G45: Volume 3: Display Register

Intel® 965G Express Chipset Family and Intel® G35 Express Chipset Graphics Controller

Programmer's Reference Manual (PRM)

January 2009

Revision 2.0a
Document Number: 321394-001

Technical queries: ilg@linux.intel.com

www.intellinuxgraphics.org

You are not obligated to provide Intel with comments or suggestions regarding this document. However, should you provide Intel with comments or suggestions for the modification, correction, improvement, or enhancement of: 9a) this document; or (b) Intel products, which may embody this document, you grant to Intel a non-exclusive, irrevocable, worldwide, royalty-free license, with the right to sublicense Intel's licensees and customers, under Recipient intellectual property rights, to use and disclose such comments and suggestions in any manner Intel chooses and to display, perform, copy, make, have made, use, sell, and otherwise dispose of Intel's and its sublicensee's products embodying such comments and suggestions in any manner and via any media Intel chooses, without reference to the source.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® 965 Express Chipset Family and Intel® G35 Express Chipset may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

I2C is a two-wire communications bus/protocol developed by Philips. SMBus is a subset of the I2C bus/protocol and was developed by Intel. Implementations of the I2C bus/protocol may require licenses from various entities, including Philips Electronics N.V. and North American Philips Corporation.

Intel and the Intel are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Contents

Figures

Tables

Revision History

Document Number	Revision Number	Description	Revision Date
24517	1.0c	Initial release.	January 2008
321394-001	2.0a	Cantiga release	January 2009

§§

1 Introduction

This Programmer's Reference Manual (PRM) describes the architectural behavior and programming environment of the Intel® 965 Express Chipset family and Intel® G35 and G45 Express Chipset GMCH graphics devices (see Table 1-1). The GMCH's Graphics Controller (GC) contains an extensive set of registers and instructions for configuration, 2D, 3D, and Video systems. The PRM describes the register, instruction, and memory interfaces and the device behaviors as controlled and observed through those interfaces. The PRM also describes the registers and instructions and provides detailed bit/field descriptions.

Note: The term "GenX" is used throughout the PRM to refer to the family of graphics devices. The devices listed in Table 1-1 are GenX devices.

Table 1-1. Supported Chipsets

Chipset Family Name	Device Name	Device Tag
Intel® Q965 Chipset Intel® Q963 Chipset Intel® G965 Chipset	82Q965 GMCH 82Q963 GMCH 82G965 GMCH	[DevBW]
Intel® G35 Chipset	82G35 GMCH	[DevBW-E]
Intel® GM965 Chipset Intel® GME965 Chipset	GM965 GMCH GME965 GMCH	[DevCL]
Intel® GM45 Chipset	GM45 GMCH GS45 GMCH GL40 GMCH	[DevCTG]
Intel® G41 Chipset Intel® G45 Chipset Intel® G43 Chipset Intel® G54 Chipset Intel® Q43 Chipset Intel® Q45 Chipset	GM41 GMCH GM43 GMCH GM54 GMCH Q43 GMCH Q45 GMCH	[DevEL]

NOTES:

1. Unless otherwise specified, the information in this document applies to all of the devices mentioned in Table 1-1. For Information that does not apply to all devices, the Device Tag is used.
2. Throughout the PRM, references to "All" in a project field refters to all devices in Table 1-1.
3. Throughout the PRM, references to [DevBW] apply to both [DevBW] and [DevBW-E]. [DevBW-E] is referenced specifically for information that is [DevBW-E] only.
4. Stepping info is sometimes appended to the device tag (e.g., [DevBW-C]). Information without any device tagging is applicable to all devices/steppings.

The PRM is intended for hardware, software, and firmware designers who seek to implement or use the graphic functions of the 965 Express Chipset family, G35, and G45 Express Chipset. Familiarity with 2D and 3D graphics programming is assumed.

The Programmer's Reference Manual is organized into four volumes:

- **PRM, Volume 1a and Volume 1b: Graphics Core**
  Volume 1 covers the overall Graphics Processing Unit (GPU), without much detail on 3D, Media, or the core subsystem. Topics include the command streamer, context switching, and memory access (including tiling). The Memory Data Formats can also be found in this volume.

  The volume also contains a chapter on the Graphics Processing Engine (GPE). The GPE is a collective term for 3D, Media, the subsystem, and the parts of the memory interface that are used by these units. Display, blitter and their memory interfaces are *not* included in the GPE.
- ***PRM, Volume 2; 3D/Media***
  Volume 2 covers the 3D and Media pipelines in detail. This volume is where details for all of the "fixed functions" are covered, including commands processed by the pipelines, fixed-function state structures, and a definition of the inputs (payloads) and outputs of the threads spawned by these units.

  This volume also covers the single Media Fixed Function, VLD. It describes how to initiate generic threads using the thread spawner (TS). It is generic threads which will be used for doing the majority of media functions. Programmable kernels will handle the algorithms for media functions such IDCT, Motion Compensation, and even Motion Estimation (used for encoding MPEG streams).
- ***PRM, Volume 3: Display Registers***
  Volume 3 describes the control registers for the display. The overlay registers and VGA registers are also covered in this volume.
- ***PRM, Volume 4: Subsystem and Cores***
  Volume 4 describes the GMCH programmable cores, or EUs, and the "shared functions", which are shared by more than one EU and perform functions such as I/O and complex math functions.

  The shared functions consist of the sampler, extended math unit, data port (the interface to memory for 3D and media), Unified Return Buffer (URB), and the Message Gateway which is used by EU threads to signal each other. The EUs use messages to send data to and receive data from the subsystem; the messages are described along with the shared functions, although the generic message send EU instruction is described with the rest of the instructions in the Instruction Set Architecture (ISA) chapters.

  This latter part of this volume describes the GMCH core, or EU, and the associated instructions that are used to program it. The instruction descriptions make up what is referred to as an Instruction Set Architecture, or ISA. The ISA describes all of the instructions that the GMCH core can execute, along with the registers that are used to store local data.

Note: The chipset PCI Configuration registers are not part of this PRM.

1.1 Notations and Conventions

1.1.1 Reserved Bits and Software Compatibility

In many register, instruction and memory layout descriptions, certain bits are marked as “Reserved”. When bits are marked as reserved, it is essential for compatibility with future devices that software treat these bits as having a future, though unknown, effect. The behavior of reserved bits should be regarded as not only undefined, but unpredictable. Software should follow these guidelines in dealing with reserved bits:

- Do not depend on the states of any reserved bits when testing values of registers that contain such bits. Mask out the reserved bits before testing. Do not depend on the states of any reserved bits when storing to instruction or to a register.
- When loading a register or formatting an instruction, always load the reserved bits with the values indicated in the documentation, if any, or reload them with the values previously read from the register.

1.2 Terminology

Term	Abbr.	Definition
3D Pipeline	—	One of the two pipelines supported in the GPE. The 3D pipeline is a set of fixed-function units arranged in a pipelined fashion, which process 3D-related commands by spawning EU threads. Typically this processing includes rendering primitives. See *3D Pipeline*.
Adjacency	—	One can consider a single line object as existing in a strip of connected lines. The neighboring line objects are called “adjacent objects”, with the non-shared endpoints called the “adjacent vertices.” The same concept can be applied to a single triangle object, considering it as existing in a mesh of connected triangles. Each triangle shares edges with three other adjacent triangles, each defined by an non-shared adjacent vertex. Knowledge of these adjacent objects/vertices is required by some object processing algorithms (e.g., silhouette edge detection). See *3D Pipeline*.
Application IP	AIP	Application Instruction Pointer. This is part of the control registers for exception handling for a thread. Upon an exception, hardware moves the current IP into this register and then jumps to SIP.
Architectural Register File	ARF	A collection of architecturally visible registers for a thread such as address registers, accumulator, flags, notification registers, IP, null, etc. ARF should not be mistaken as just the address registers.
Array of Cores	—	Refers to a group of GenX EUs, which are physically organized in two or more rows. The fact that the EUs are arranged in an array is (to a great extent) transparent to CPU software or EU kernels.

Term	Abbr.	Definition
Binding Table	—	Memory-resident list of pointers to surface state blocks (also in memory).
Binding Table Pointer	BTP	Pointer to a binding table, specified as an offset from the Surface State Base Address register.
Bypass Mode	—	Mode where a given fixed function unit is disabled and forwards data down the pipeline unchanged. Not supported by all FF units.
Byte	B	A numerical data type of 8 bits, B represents a signed byte integer.
Child Thread	—	A branch-node or a leaf-node thread that is created by another thread. It is a kind of thread associated with the media fixed function pipeline. A child thread is originated from a thread (the parent) executing on an EU and forwarded to the Thread Dispatcher by the TS unit. A child thread may or may not have child threads depending on whether it is a branch-node or a leaf-node thread. All pre-allocated resources such as URB and scratch memory for a child thread are managed by its parent thread.
Clip Space	—	A 4-dimensional coordinate system within which a clipping frustum is defined. Object positions are projected from Clip Space to NDC space via “perspective divide” by the W coordinate, and then viewport mapped into Screen Space
Clipper	—	3D fixed function unit that removes invisible portions of the drawing sequence by discarding (culling) primitives or by “replacing” primitives with one or more primitives that replicate only the visible portion of the original primitive.
Color Calculator	CC	Part of the Data Port shared function, the color calculator performs fixed-function pixel operations (e.g., blending) prior to writing a result pixel into the render cache.
Command	—	Directive fetched from a ring buffer in memory by the Command Streamer and routed down a pipeline. Should not be confused with instructions which are fetched by the instruction cache subsystem and executed on an EU.
Command Streamer	CS or CSI	Functional unit of the Graphics Processing Engine that fetches commands, parses them and routes them to the appropriate pipeline.
Constant URB Entry	CURBE	A UE that contains “constant” data for use by various stages of the pipeline.
Control Register	CR	The read-write registers are used for thread mode control and exception handling for a thread.
Degenerate Object	—	Object that is invisible due to coincident vertices or because does not intersect any sample points (usually due to being tiny or a very thin sliver).
Destination	—	Describes an output or write operand.
Destination Size	—	The number of data elements in the destination of a GenX SIMD instruction.

Term	Abbr.	Definition
Destination Width	—	The size of each of (possibly) many elements of the destination of a GenX SIMD instruction.
Double Quad word (DQword)	DQ	A fundamental data type, DQ represents 16 bytes.
Double word (DWord)	D or DW	A fundamental data type, D or DW represents 4 bytes.
Drawing Rectangle	--	A screen-space rectangle within which 3D primitives are rendered. An objects screen-space positions are relative to the Drawing Rectangle origin. See *Strips and Fans*.
End of Block	EOB	A 1-bit flag in the non-zero DCT coefficient data structure indicating the end of an 8x8 block in a DCT coefficient data buffer.
End Of Thread	EOT	A message sideband signal on the Output message bus signifying that the message requester thread is terminated. A thread must have at least one SEND instruction with the EOT bit in the message descriptor field set in order to properly terminate.
Exception	—	Type of (normally rare) interruption to EU execution of a thread's instructions. An exception occurrence causes the EU thread to begin executing the System Routine which is designed to handle exceptions.
Execution Channel	—	GenX EU instructions typically operate on multiple data values in parallel (i.e., in “SIMD” fashion). The data is processed in parallel “execution channels” (e.g., a SIMD8 instruction uses 8 execution channels to perform 8 operations in parallel).
Execution Size	ExecSize	Execution Size indicates the number of data elements processed by a GENX SIMD instruction. It is one of the GENX instruction fields and can be changed per instruction.
Execution Unit	EU	Execution Unit. An EU is a multi-threaded processor within the GENX multi-processor system. Each EU is a fully-capable processor containing instruction fetch and decode, register files, source operand swizzle and SIMD ALU, etc. An EU is also referred to as a GENX Core.
Execution Unit Identifier	EUID	The 4-bit field within a thread state register (SR0) that identifies the row and column location of the EU a thread is located. A thread can be uniquely identified by the EUID and TID.
Execution Width	ExecWidth	The width of each of several data elements that may be processed by a single GenX SIMD instruction.
Extended Math Unit	EM	A Shared Function that performs more complex math operations on behalf of several EUs.
FF Unit	—	A Fixed-Function Unit is the hardware component of a 3D Pipeline Stage. A FF Unit typically has a unique FF ID associated with it.

Term	Abbr.	Definition
Fixed Function	FF	Function of the pipeline that is performed by dedicated (vs. programmable) hardware.
Fixed Function ID	FFID	Unique identifier for a fixed function unit.
FLT_MAX	fmax	The magnitude of the maximum representable single precision floating number according to IEEE-754 standard. FLT_MAX has an exponent of 0xFE and a mantissa of all one's.
Gateway	GW	See Message Gateway.
GENX Core	—	Alternative name for an EU in the GENX multi-processor system.
General Register File	GRF	Large read/write register file shared by all the EUs for operand sources and destinations. This is the most commonly used read-write register space organized as an array of 256-bit registers for a thread.
General State Base Address	—	The Graphics Address of a block of memory-resident "state data", which includes state blocks, scratch space, constant buffers and kernel programs. The contents of this memory block are referenced via offsets from the contents of the General State Base Address register. See *Graphics Processing Engine*.
Geometry Shader	GS	Fixed-function unit between the vertex shader and the clipper that (if enabled) dispatches "geometry shader" threads on its input primitives. Application-supplied geometry shaders normally expand each input primitive into several output primitives in order to perform 3D modeling algorithms such as fur/fins. See *Geometry Shader*.
Graphics Address	—	The GPE virtual address of some memory-resident object. This virtual address gets mapped by a GTT or PGTT to a physical memory address. Note that many memory-resident objects are referenced not with Graphics Addresses, but instead with offsets from a "base address register".
Graphics Processing Engine	GPE	Collective name for the Subsystem, the 3D and Media pipelines, and the Command Streamer.
Guardband	GB	Region that may be clipped against to make sure objects do not exceed the limitations of the renderer's coordinate space.
Horizontal Stride	HorzStride	The distance in element-sized units between adjacent elements of a GenX region-based GRF access.
Immediate floating point vector	VF	A numerical data type of 32 bits, an immediate floating point vector of type VF contains 4 floating point elements with 8 bits each. The 8-bit floating point element contains a sign field, a 3-bit exponent field and a 4-bit mantissa field. It may be used to specify the type of an immediate operand in an instruction.

Term	Abbr.	Definition
Immediate integer vector	V	A numerical data type of 32 bits, an immediate integer vector of type V contains 8 signed integer elements with 4-bit each. The 4-bit integer element is in 2's complement form. It may be used to specify the type of an immediate operand in an instruction.
Index Buffer	IB	Buffer in memory containing vertex indices.
In-loop Deblocking Filter	ILDB	The deblocking filter operation in the decoding loop. It is a stage after MC in the video decoding pipe.
Instance	—	In the context of the VF unit, an instance is one of a sequence of sets of similar primitive data. Each set has identical vertex data but may have unique instance data that differentiates it from other sets in the sequence.
Instruction	—	Data in memory directing an EU operation. Instructions are fetched from memory, stored in a cache and executed on one or more GenX cores. Not to be confused with commands which are fetched and parsed by the command streamer and dispatched down the 3D or Media pipeline.
Instruction Pointer	IP	The address (really an offset) of the instruction currently being fetched by an EU. Each EU has its own IP.
Instruction Set Architecture	ISA	The GENX ISA describes the instructions supported by a GENX EU.
Instruction State Cache	ISC	On-chip memory that holds recently-used instructions and state variable values.
Interface Descriptor	—	Media analog of a State Descriptor.
Intermediate Z	IZ	Completion of the Z (depth) test at the front end of the Windower/Masker unit when certain conditions are met (no alpha, no pixel-shader computed Z values, etc.)
Inverse Discrete Cosine Transform	IDCT	The stage in the video decoding pipe between IQ and MC
Inverse Quantization	IQ	A stage in the video decoding pipe between IS and IDCT.
Inverse Scan	IS	A stage in the video decoding pipe between VLD and IQ. In this stage, a sequence of none-zero DCT coefficients are converted into a block (e.g. an 8x8 block) of coefficients. VFE unit has fixed functions to support IS for both MPEG-2 and WMV.
Jitter	—	Just-in-time compiler.
Kernel	—	A sequence of GenX instructions that is logically part of the driver or generated by the jitter. Differentiated from a Shader which is an application supplied program that is translated by the jitter to GenX instructions.
Least Significant Bit	LSB	Least Significant Bit
MathBox	—	See Extended Math Unit

Term	Abbr.	Definition
Media	—	Term for operations such as video decode and encode that are normally performed by the Media pipeline.
Media Pipeline	—	Fixed function stages dedicated to media and “generic” processing, sometimes referred to as the generic pipeline.
Message	—	Messages are data packages transmitted from a thread to another thread, another shared function or another fixed function. Message passing is the primary communication mechanism of GENX architecture.
Message Gateway	—	Shared function that enables thread-to-thread message communication/synchronization used solely by the Media pipeline.
Message Register File	MRF	Write-only registers used by EUs to assemble messages prior to sending and as the operand of a send instruction.
Most Significant Bit	MSB	Most Significant Bit
Motion Compensation	MC	Part of the video decoding pipe.
Motion Picture Expert Group	MPEG	MPEG is the international standard body JTC1/SC29/WG11 under ISO/IEC that has defined audio and video compression standards such as MPEG-1, MPEG-2, and MPEG-4, etc.
Motion Vector Field Selection	MVFS	A four-bit field selecting reference fields for the motion vectors of the current macroblock.
Multi Render Targets	MRT	Multiple independent surfaces that may be the target of a sequence of 3D or Media commands that use the same surface state.
Normalized Device Coordinates	NDC	Clip Space Coordinates that have been divided by the Clip Space “W” component.
Object	—	A single triangle, line or point.
Open GL	OGL	A Graphics API specification associated with Linux.
Out-of-loop De-Blocking Filter	OLDB	The de-blocking filter operation outside the decoding loop.
Out-of-loop De-Ringing Filter	OLDR	The de-ringing filter operation outside the decoding loop.
Parent Thread	—	A thread corresponding to a root-node or a branch-node in thread generation hierarchy. A parent thread may be a root thread or a child thread depending on its position in the thread generation hierarchy.
Pipeline Stage	—	A abstracted element of the 3D pipeline, providing functions performed by a combination of the corresponding hardware FF unit and the threads spawned by that FF unit.
Pipelined State Pointers	PSP	Pointers to state blocks in memory that are passed down the pipeline.

Term	Abbr.	Definition
Pixel Shader	PS	Shader that is supplied by the application, translated by the jitter and is dispatched to the EU by the Windower (conceptually) once per pixel.
Point	—	A drawing object characterized only by position coordinates and width.
Primitive	—	Synonym for object: triangle, rectangle, line or point.
Primitive Topology	—	A composite primitive such as a triangle strip, or line list. Also includes the objects triangle, line and point as degenerate cases.
Provoking Vertex	—	The vertex of a primitive topology from which vertex attributes that are constant across the primitive are taken.
Quad Quad word (QQword)	QQ	A fundamental data type, QQ represents 32 bytes.
Quad Word (QWord)	QW	A fundamental data type, QW represents 8 bytes.
Rasterization	—	Conversion of an object represented by vertices into the set of pixels that make up the object.
Region-based addressing	—	Collective term for the register addressing modes available in the EU instruction set that permit discontiguous register data to be fetched and used as a single operand.
Render Cache	RC	Cache in which pixel color and depth information is written prior to being written to memory, and where prior pixel destination attributes are read in preparation for blending and Z test.
Render Target	RT	A destination surface in memory where render results are written.
Render Target Array Index	—	Selector of which of several render targets the current operation is targeting.
Root Thread	—	A root-node thread. A thread corresponds to a root-node in a thread generation hierarchy. It is a kind of thread associated with the media fixed function pipeline. A root thread is originated from the VFE unit and forwarded to the Thread Dispatcher by the TS unit. A root thread may or may not have child threads. A root thread may have scratch memory managed by TS. A root thread with children has its URB resource managed by the VFE.
Sampler	—	Shared function that samples textures and reads data from buffers on behalf of EU programs.
Scratch Space	—	Memory allocated to the subsystem that is used by EU threads for data storage that exceeds their register allocation, persistent storage, storage of mask stack entries beyond the first 16, etc.
Shader	—	A GenX program that is supplied by the application in an high level shader language, and translated to GenX instructions by the jitter.

Term	Abbr.	Definition
Shared Function	SF	Function unit that is shared by EUs. EUs send messages to shared functions; they consume the data and may return a result. The Sampler, Data Port and Extended Math unit are all shared functions.
Shared Function ID	SFID	Unique identifier used by kernels and shaders to target shared functions and to identify their returned messages.
Single Instruction Multiple Data	SIMD	The term SIMD can be used to describe the kind of parallel processing architecture that exploits data parallelism at instruction level. It can also be used to describe the instructions in such architecture.
Source	—	Describes an input or read operand
Spawn	—	To initiate a thread for execution on an EU. Done by the thread spawner as well as most FF units in the 3D pipeline.
Sprite Point	—	Point object using full range texture coordinates. Points that are not sprite points use the texture coordinates of the point's center across the entire point object.
State Descriptor	—	Blocks in memory that describe the state associated with a particular FF, including its associated kernel pointer, kernel resource allowances, and a pointer to its surface state.
State Register	SR	The read-only registers containing the state information of the current thread, including the EUID/TID, Dispatcher Mask, and System IP.
State Variable	SV	An individual state element that can be varied to change the way given primitives are rendered or media objects processed. On GenX state variables persist only in memory and are cached as needed by rendering/processing operations except for a small amount of non-pipelined state.
Stream Output	—	A term for writing the output of a FF unit directly to a memory buffer instead of, or in addition to, the output passing to the next FF unit in the pipeline. Currently only supported for the Geometry Shader (GS) FF unit.
Strips and Fans	SF	Fixed function unit whose main function is to decompose primitive topologies such as strips and fans into primitives or objects.
Sub-Register	—	Subfield of a SIMD register. A SIMD register is an aligned fixed size register for a register file or a register type. For example, a GRF register, *r2*, is 256-bit wide, 256-bit aligned register. A sub-register, *r2.3:d*, is the fourth dword of GRF register *r2*.
Subsystem	—	The GenX name given to the resources shared by the FF units, including shared functions and EUs.
Surface	—	A rendering operand or destination, including textures, buffers, and render targets.
Surface State	—	State associated with a render surface including
Surface State Base Pointer	—	Base address used when referencing binding table and surface state data.

Term	Abbr.	Definition
Synchronized Root Thread	—	A root thread that is dispatched by TS upon a 'dispatch root thread' message.
System IP	SIP	There is one global System IP register for all the threads. From a thread's point of view, this is a virtual read-only register. Upon an exception, hardware performs some bookkeeping and then jumps to SIP.
System Routine	—	Sequence of GenX instructions that handles exceptions. SIP is programmed to point to this routine, and all threads encountering an exception will call it.
Thread	—	An instance of a kernel program executed on an EU. The life cycle for a thread starts from the executing the first instruction after being dispatched from Thread Dispatcher to an EU to the execution of the last instruction – a send instruction with EOT that signals the thread termination. Threads in GENX system may be independent from each other or communicate with each other through Message Gateway share function.
Thread Dispatcher	TD	Functional unit that arbitrates thread initiation requests from Fixed Functions units and instantiates the threads on EUs.
Thread Identifier	TID	The field within a thread state register (SR0) that identifies which thread slots on an EU a thread occupies. A thread can be uniquely identified by the EUID and TID.
Thread Payload	—	Prior to a thread starting execution, some amount of data will be pre-loaded in to the thread's GRF (starting at r0). This data is typically a combination of control information provided by the spawning entity (FF Unit) and data read from the URB.
Thread Spawner	TS	The second and the last fixed function stage of the media pipeline that initiates new threads on behalf of generic/media processing.
Topology	—	See Primitive Topology.
Unified Return Buffer	URB	The on-chip memory managed/shared by GENX Fixed Functions in order for a thread to return data that will be consumed either by a Fixed Function or other threads.
Unsigned Byte integer	UB	A numerical data type of 8 bits.
Unsigned Double Word integer	UD	A numerical data type of 32 bits. It may be used to specify the type of an operand in an instruction.
Unsigned Word integer	UW	A numerical data type of 16 bits. It may be used to specify the type of an operand in an instruction.
Unsynchronized Root Thread	—	A root thread that is automatically dispatched by TS.
URB Dereference	—	See URB Reference
URB Entry	UE	URB Entry: A logical entity stored in the URB (such as a vertex), referenced via a URB Handle.

Term	Abbr.	Definition
URB Entry Allocation Size	—	Number of URB entries allocated to a Fixed Function unit.
URB Fence	Fence	Virtual, movable boundaries between the URB regions owned by each FF unit.
URB Handle	—	A unique identifier for a URB entry that is passed down a pipeline.
URB Reference	—	For the most part, data is passed down the fixed function pipeline in an indirect fashion. The data is typically stored in the URB and accessed via a URB handle. When a pipeline stage passes the handle of a URB data entry to a downstream stage, it is said to make a URB reference. Note that there may be several references to the same URB data entry in the pipeline at any given time. When a downstream stage accesses the URB data entry via a URB handle, it is said to “dereference” the URB data entry. When there are no longer any references to a URB data entry within the pipeline, the URB storage can be reclaimed.
Variable Length Decode	VLD	The first stage of the video decoding pipe that consists mainly of bit-wide operations. GENX supports hardware VLD acceleration in the VFE fixed function stage.
Vertex Buffer	VB	Buffer in memory containing vertex attributes.
Vertex Cache	VC	Cache of Vertex URB Entry (VUE) handles tagged with vertex indices. See the VS chapter for details on this cache.
Vertex Fetcher	VF	The first FF unit in the 3D pipeline responsible for fetching vertex data from memory. Sometimes referred to as the Vertex Formatter.
Vertex Header	—	Vertex data required for every vertex appearing at the beginning of a Vertex URB Entry.
Vertex ID	—	Unique ID for each vertex that can optionally be included in vertex attribute data sent down the pipeline and used by kernel/shader threads.
Vertex Index	—	Offset (in vertex-sized units) of a given vertex in a vertex buffer. Available in the VF and VS units for debugging purposes. Not unique per vertex instance.
Vertex Sequence Number	—	Unique ID for each vertex sent down the south bus that may be used to identify vertices for debugging purposes.
Vertex Shader	VS	An API-supplied program that calculates vertex attributes. Also refers to the FF unit that dispatches threads to “shade” (calculate attributes for) vertices.
Vertex URB Entry	VUE	A URB entry that contains data for a specific vertex.
Vertical Stride	VertStride	The distance in element-sized units between 2 vertically-adjacent elements of a GenX region-based GRF access.
Video Front End	VFE	The first fixed function in the GENX generic pipeline; performs fixed-function media operations.

Term	Abbr.	Definition
Viewport	VP	Post-clipped geometry is mapped to a rectangular region of the bound rendertarget(s). This rectangular region is called a viewport. Typically, the viewport is set to the full extent of the rendertarget(s), but any subregion can be used as well.
Windower IZ	WIZ	Term for Windower/Masker that encapsulates its early (“intermediate”) depth test function.
Windower/Masker	WM	Fixed function triangle/line rasterizer.
Word	W	A numerical data type of 16 bits, W represents a signed word integer.

2 Display Registers

2.1 Introduction and Register Summary

This chapter contains the register descriptions for the display portion of a family of integrated graphics devices. These registers do vary by devices within the family of devices so special attention needs to be paid to which devices use which registers and register fields.

Different devices within the family may add, modify, or delete registers or register fields relative to another device in the same family based on the supported functions of that device. This document covers both desktop and mobile products.

The following table contains the sections break down where the register information is contained within this chapter:

Address Range	Description
30000h–3FFFFh	Overlay Registers
05000h-05FFFh	GMBUS and I/O Control
06000h-06FFFh	Display Clocks and Clock Gating
0A000h–0AFFFh	Display Palette A/B Registers
60000h–600FFh	Display Pipeline A
61000h–610FFh	Display Pipeline B
61100h–611FFh	Display Port Control Registers
61200h-612FFh	Panel Fitting/Power Sequencing/LVDS Control
62000h-62FFFh	Reserved.
64000h-64FFFh	DisplayPort Registers
70000h–7FFFFh	Display Pipeline, Display Planes, Cursor Planes, and VGA Control Registers

2.1.1 Panel Control Register Summary ([DevCL] and [DevCTG] Only)

Address	Function	Name	Normal Access	Panel Write Protect
61180h	Port Control	LVDS Port Control	RW	Yes
61200h	Panel Power Sequence	Panel power status	RO	NA
61204h		Panel Power Control	RW	No
61208h		Panel power on sequencing delay	RW	Yes
6120Ch		Panel power off sequencing delay	RW	Yes
61210h		Panel power cycle Delay and Reference	RW	Yes
61254h	Backlight Control	Backlight PWM Control	RW	No
61260h	BLM	Image BLM Control	RW	No
61270h – 61284h		BLM Thresholds	RW	No
61290h–612A4h		BLM Accumulators	RO	NA

2.1.2 Display Pipe and Plane Control Registers

Address Range	Description
70000h – 70027h	Display Pipeline A Control
70028h – 7007Fh	Reserved
70080h – 7009Fh	Display Cursor Plane Registers A
700A0h – 700BFh	Reserved
700C0h – 70FFFh	Reserved
700C0h – 700DFh	Display Cursor Plane Registers B
700E0h – 70FFFh	Reserved
71000h – 7100Ch	Display Pipeline B Control
71010h – 7107Fh	Reserved
70180h – 7018Ch	Display A Plane Control
71090h – 7117Fh	Reserved
71180h – 7119Bh	Display B/Sprite Plane Control
711D0h – 713FFh	Reserved
71400h – 7140Fh	Video BIOS Registers
72010h – 720FFh	Reserved
72100h – 7217Fh	Reserved
72180h – 7219Fh	Display C/Sprite Plane Control
721D0h – 721F7h	Display C Color Adjustment

Address Range	Description
72000h – 72FFFh	Reserved

2.1.3 Terminology

Description	Software Use	Should be implemented as
Reserved write as zero.	Software must always write a zero to these bits. This allows new features to be added using these bits that will be disabled when using old software and as the default case.	These are read-only bits that always read as zeros or r/w bits that are default to zero.
Reserved write as one.	Software must always write a one to these bits. This allows new features to be added using these bits that will be disabled when using old software and as the default case.	
Reserved for BIOS Do not change	Driver access to these bits must read these bits that have been set through an initialization operation before writing this register so that the bits can remain unchanged.	According to each specific bit
Reserved for Video BIOS	These register bits will be used only by video BIOS and drivers should not change them.	These are read/write bits that have no hardware function. They are intended for use by the video BIOS for storage.
Reserved for Compatibility	For functions that are no longer needed these bits had old use, but now does nothing. New software should use the new method.	Read/write bits that have no functions.
Use for compatibility only	Under specific conditions, these bits functions as in the old part, new software should use the new method.	According to each specific bit
Read-Only	This bit is read-only. The read value is determined by hardware. Writes to this bit have no effect.	According to each specific bit. The bit value is determined by hardware and not affected by register writes to the actual bit.
Reserved read-only	Don't assume a value for these bits. Writes have no effect.	These bits should read as zero.
Reserved read-only write as zero	Don't assume a value for these bits, always write a zero.	These bits should read as zero.
Read/Clear	This bit can be read and writes to it with a one cause the bit to clear.	Hardware events cause the bit to be set and the bit will be cleared on a write operation where the corresponding bit has a one for a value.
Read/Write	This bit can be read or written.	

Description	Software Use	Should be implemented as
Double Buffered	Write when desired	Takes effect only after a particular event such as a VBLANK.

2.1.4 Register Protection for Panel Protection

2.2 Display Mode Set Sequence

2.2.1 Pipe register double-buffering

Pipe config registers (0x70008 0x71008):

Contain pipe enable, double wide, gamma mode, interlaced mode, plane enable overrides, frame start delay, force border.

Pipe timing registers (60000/4/8/C/10/14/1C 61000/4/8/C/10/14/1C):

Contain horizontal and vertical total, active, blank start/end, sync start/end, source size.

	Pipe Register Double-Buffering **(double-buffering to start of vertical blank)**
Pipe config register	Updates at Vblank, or updates if pipe completely off, or updates if VGA native and enabled.
Pipe timing registers	Updates at Vblank after pipe config enable written 0 to 1, or updates if DPLL off or pipe completely off and pipe config register written.

Care must be taken when disabling the pipe. The pipe will not completely turn off until the start of vertical blank after the pipe enable was written to 0. If there is no vertical blank the pipe will not be able to completely turn off. Double-buffering is bypassed if VGA native display is enabled (0x71400 bits 31,25,24 set to 0), allowing pipe to turn off immediately.

Enabling pipe always occurs instantly.

2.2.2 Mode Switch Programming Sequences

General rules to follow:

- DPLL must be enabled and warmed up before pipe or ports are enabled.
- DPLL must be kept enabled until ports are disabled and pipe is completely off.
- DPLL frequency must not be changed until ports are disabled and pipe is completely off, except when in native VGA where SR01 25/28 MHz select can be changed.
- Planes must be disabled before pipe is disabled or pipe timings changed.
- Panelfitter must be enabled or disabled only when pipe is completely off.
- On Gen3 set port multiply when enabling a SDVO port.
- On Gen3.5 and GenX set port multiply when programming the DPLL.
- The internal TV and CRT ports can be left on during a mode switch if DPLL is not touched.
- Ports can be freely enabled or disabled on a running pipe, except when port multiply needs to be changed.

Wait for Vblank (using pipe status register or MI_WAIT_FOR_EVENT, not VGA status register) is the best way to find when planes have completely turned off for pre-Gen3. On pre-Gen3 only planes are double-buffered, not pipe, so wait for planes to turn off then disable pipe.

Wait for vertical counter to stop (by reading the scanline count register over multiple line times) is the best way to find when pipe is completely off for Gen3 and Gen3.5. Pipe and planes are double-buffered on Gen3 onwards, so wait after disabling pipe.

Wait for pipe off status (using pipe config register bit 30) is the best way to find when pipe is completely off.

Table 2-1. Mode Switch Sequences

Enable sequence
Pipe A must be completely off at this point Write PIPEACONF bits[19:18] = 00 Write DSPACNTR bit[31] = 1 Write DSPASURF = 0x00000000 Write DSPACNTR bit[31] = 0 Write DSPASURF = 0x00000000 Restore PIPEACONF bits[19:18] to original value Program DPLL Enable DPLL **Wait for DPLL warmup 150us** (Wait ensures clock is running smoothly before enabling pipe) Program pipe timings (Can be done before DPLL programming) Enable panelfitter as needed (Can be done before DPLL and/or pipe timings programming) Enable pipe Enable planes (VGA or hires) Enable ports
Disable sequence
Disable sDVO ports' stall input by clearing 0x61140 and 0x61160 bit 29 to 0 (port enable not changed) Program sDVO ADD device Disable ports Disable planes (VGA or hires) Disable pipe **Disable VGA display in 0x71400 bit 31** (Disable VGA display done after disable pipe to allow pipe to turn off when no vblank is available in native VGA mode) If GenX { **Wait for pipe off status** } (Wait ensures planes and pipe have completely turned off prior to disabling panelfitter then DPLL) Disable panelfitter Disable DPLL
Pipe timings change or change between VGA native or VGA center/upperleft or Hires
Use complete disable sequence followed by complete enable sequence with new mode programmings.

VGA native timings change
Program VGA and SR01 25/28 MHz select registers as needed.

Note: On Cantiga [DevCTG] when using Self Refresh, and one pipe is enabled and displaying, and a second pipe is being enabled or disabled, the workaround sequence below needs to be followed. This does not need to done when both pipes are being enabled or disabled at the same time.

When turnning on a second pipe:

1. Disable Self Refresh (register 0x20E0 bit 15 set to 0).
2. Wait for vblank from first pipe.
3. Enable second pipe using the mode switch enable sequence as normal.

When turning off a second pipe:

4. Disable second pipe using the mode switch disable sequence as normal.
5. Anytime after the step in the disable sequence where there is a wait for pipe off status, re-enable Self Refresh if desired (register 0x20E0 bit 15 set to 1).

2.3 Display Power Down/Up Register Access Sequence

2.3.1 Power Up/Down for CRT-like Display Devices

For products driving a display or transcoder that require changing display timings for a mode change, the programming sequence outlined below should be used during a power up or power down sequence. Examples of this type of display are a CRT device, or one that uses an external scaler but not fixed timings (such as an external TV encoder). Only registers that are double buffered should be updated while the display pipe is enabled in order to avoid screen glitches.

Power down sequence:

1. Turn off the port(s) assigned to that pipe.
2. Turn off display planes that are assigned to the display pipe by writing the enable bit with a zero.
3. Turn off the display pipe (will disable at the next VBLANK).

Power up sequence:

4. Program the pipe timing, source size registers and DPLL register values to the desired mode values.
5. Configure and turn on the planes that are to be assigned to that pipe.
6. Turn on the port(s) assigned to that pipe.

2.3.2 Power Up/Down for Integrated and External Panel Scaler Driven Display Devices

For products driving an integrated panel or a device that operates through a scaler function using fixed timings, the sequence outlined below should be used to power down or up. The screen does not have to be powered down during this mode change operation.

Power down sequence:

1. Turn off the port(s) assigned to that pipe.
2. Turn off display planes that are assigned to the target display pipe (VGA display disabled if VGA mode).
3. Turn off the display pipe (will disable at the next VBLANK).

Power up sequence:

4. Program the pipe timing, source size registers and DPLL register values to the desired mode values.
5. Configure and turn on the plane engines that are to be assigned to that pipe (VGA display enabled if going to a VGA mode, centering, pixel doubling or panel fitting appropriately enabled).
6. Turn on the port(s) assigned to that pipe.

2.4 GMBUS and I/O Control Registers (05000h–05FFFh)

2.4.1 GPIO Pin Usage (By Functions)

GPIO pins allow the support of simple query and control functions such as DDC and I^2C interface protocols. GPIO pins exist in pairs (for the most part) and provide a mechanism to control external devices through a register programming interface. GPIO pins can be set to a level or the value of the pin can be read. This allows for a “bit banging” version of an I2C interface to be implemented. An additional function of using the GMBUS engine to run the I2C protocols is also allowed. Refer to the EDS for GPIO signal descriptions. Refer to the *Philips I2C-BUS SPECIFICATION version 2.1* for a description of the I2C bus and protocol.

The number and names of the GPIO pins vary from device type to device type. Some of the GPIO pins will be muxed with other functions and are only available when the other function is not being used. The following subsections describe the GPIO pin to register mapping for the various devices. OEMs have the ability to remap these functions onto other pins as long as the hardware limitations are observed.

2.4.2 GPIO Pin Usage (By Device)

Port	Pin Use (Name)	GMBUS Use	sDVO Use	Internal Pullup	I^2C	Device	Description
7	TVDCONSEL1	Yes	No	No	Yes	All with integrated TV	Used for interface to glue logic generating D-connector HV levels. Default value is tri-stated.
	TVDCONSEL0				Yes		
6	Reserved	No	No	No			
	Reserved						
5	Reserved	No	No	No	No		Reserved
	Reserved				No		Reserved
4	Reserved	Yes	Yes	No – weak pulldown on reset	Yes	All	Reserved
	Reserved				Yes		
3	Reserved	No	No	No	No		Reserved
	Reserved				No		Reserved
2	LVDS/DPD DDC Data (DDCPDATA)	Yes	No	No – weak pulldown on reset	Yes	All	DDC for Digital Display connection via the integrated LVDS or DP port D.
	LVDS/DPD DDC Clock (DDCPCLK)				Yes		

Port	Pin Use (Name)	GMBUS Use	sDVO Use	Internal Pullup	I^2C	Device	Description
1	I2C Data (LCLKCTRLB)	Yes	No	No	Yes	All	For control of SSC clock generator devices on motherboard. Support can be optionally I2C or control level.
	I2C Clock (LCLKCTRLA)				Yes		
0	DAC DDC Data (DDCADATA)	Yes	No	No	Yes	All	DDC for Analog monitor (VGA) connection. This cannot be shared with other DDC or I2C pairs due to legacy monitor issues.
	DAC DDC Clock (DDCACLK)				Yes		

2.4.3 GPIO Pin Glue Logic

The GPIO output can be converted to alternate levels through use of external logic on the board. This section describes conversion logic that will be implemented on the board.

2.4.3.1 D-Connector

The D-connector requires multiple voltage levels to select modes on a display device. The following table describes the translation as done by the board glue logic from the GPIO pins to the video modes. When the GPIO pins are in their default, tri-stated state, the D-connector default value will drive zeros on all three lines, indicating 480i at 4:3.

GPIO Pins			**D-Connector Lines**			
TVDCONSEL0	**TVDCONSEL1**	**TVDCONSEL2**	**Line 1** **Voltage (V)**	**Line 2** **Voltage (V)**	**Line 3** **Voltage (V)**	**Video Format**
0	0	0	5	0	5	1080i, 16:9
0	0	1	2.2	5	5	720p, 16:9
0	1	0	0	0	5	480i, 16:9
0	1	1	0	5	5	480p, 16:9
1	0	0	0	0	2.2	480i, 4:3L
1	0	1	0	5	2.2	480p, 4:3L
1	1	0	0	0	0	480i, 4:3
1	1	1	0	5	0	480p, 4:3

2.4.4 GPIO Control Registers

The number of registers and their usage may change with each product.

Address offset:	05000h–05003h	Reserved
	05004h–05007h	Reserved
	05008h–0500Ch	Reserved
	05010h–05013h	GPIOCTL_0
	05014h–05017h	GPIOCTL_1
	05018h–0501Bh,	GPIOCTL_2
	0501Ch–0501Fh,	GPIOCTL_3
	05020h–05023h,	GPIOCTL_4
	05024h–05027h	GPIOCTL_5
	05028h–0502Bh	GPIOCTL_6
	0502Ch–0502Fh	GPIOCTL_7
Default value:	00h, 00h, 000U1000b, 000U1000b	
Normal Access:	Read / Write	
Size:	32 bit	

These registers define the control of the sets of the so called "general purpose" I/O pins. Each register controls a pair of pins that while can be used for general purpose control, most are designated for specific functions according to the requirements of the device and the system that the device finds itself in. Each pin of the two-pin pair is designated as a clock or data for descriptive purposes. See the table at the beginning of this section to determine for each product which pins/registers are supported and their intended functions. **Board design variations are possible and would affect the usage of these pins.**

GMBUS port 5 is reserved for LT visual status indication and is controlled by hardware only. It is not accessible or programmable through its associated GPIO register which should be considered reserved.

Bit	Description
31:13	Reserved
12	**GPIO_Data In—RO:** This is the value that is sampled on the GPIO_Data pin as an input. This input is synchronized to the Core Clock domain. Because the default setting is this buffer is an input, this bit is undefined at reset.
11	**GPIO Data Value—R/W:** This is the value that should be place on the GPIO Data pin as an output. This value is only written into the register if **GPIO DATA MASK** is also asserted. The value will appear on the pin if this data value is actually written to this register and the **GPIO Data DIRECTION VALUE** contains a value that will configure the pin as an output. **Default = 1.** The GPIO default clock data value is programmed to '1' in hardware. The hardware drives a default of '1' since the I2C interface defaults to a '1'. (this mimics the I2C external pull-ups on the bus)
10	**GPIO Data Mask—WO:** This is a mask bit to determine whether the **GPIO DATA VALUE** bit should be written into the register. This value is not stored and when read returns 0. 0 = Do NOT write GPIO Data Value bit (default). 1 = Write GPIO Data Value bit.

Bit	Description
9	**GPIO Data Direction Value—R/W:** This is the value that should be used to define the output enable of the GPIO Data pin. This value is only written into the register if **GPIO Data DIRECTION MASK** is also asserted. The value that will appear on the pin is defined by what is in the register for the **GPIO DATA VALUE** bit. 0 = Pin is configured as an input (default) 1 = Pin is configured as an output.
8	**GPIO Data Direction Mask—WO:** This is a mask bit to determine whether the **GPIO DIRECTION VALUE** bit should be written into the register. This value is not stored and when read always returns 0. 0 = Do NOT write GPIO Data Direction Value bit (default). 1 = Write GPIO Data Direction Value bit.
7:5	Reserved : MBZ
4	**GPIO Clock Data In—RO:** This is the value that is sampled on the GPIO Clock pin as an input. This input is synchronized to the Core Clock domain. Because the default setting is this buffer is an input, this bit is undefined at reset.
3	**GPIO Clock Data Value—R/W:** This is the value that should be place on the GPIO Clk pin as an output. This value is only written into the register if **GPIO Clock DATA MASK** is also asserted. The value will appear on the pin if this data value is actually written to this register and the **GPIO Clock DIRECTION VALUE** contains a value that will configure the pin as an output. **Default = 1.** The GPIO default clock data value is programmed to '1' in hardware. The hardware drives a default of '1' since the I2C interface defaults to a '1'. (this mimics the I2C external pull-ups on the bus)
2	**GPIO Clock Data Mask—WO:** This is a mask bit to determine whether the **GPIO Clock DATA VALUE** bit should be written into the register. This value is not stored and when read always returns 0. 0 = Do NOT write GPIO Clock Data Value bit (default). 1 = Write GPIO Clock Data Value bit.
1	**GPIO Clock Direction Value—R/W:** This is the value that should be used to define the output enable of the GPIO Clock pin. This value is only written into the register if **GPIO Clock DIRECTION MASK** is also asserted. The value that will appear on the pin is defined by what is in the register for the **GPIO Clock DATA VALUE** bit. 0 = Pin is configured as an input and the output driver is set to tri-state (default) 1 = Pin is configured as an output.
0	**GPIO Clock Direction Mask—WO:** This is a mask bit to determine whether the **GPIO Clock DIRECTION VALUE** bit should be written into the register. This value is not stored and when read returns 0. 0 = Do NOT update the GPIO Clock Direction Value bit on a write (default). 1 = Update the GPIO Clock Direction Value bit. on a write operation to this register.

2.4.5 GMBUS Controller Programming Interface

The GMBUS (Graphic Management Bus) can be used to indirectly access/control devices connected to a GMBUS bus as an alternate to bit-wise programming via software.

The GMBUS interface is I^{2}C compatible. The basic features are listed as follow:

1. Works as the master of a single master bus.
2. The bus clock frequency is selectable by software to be 50KHz, 100KHz, 400KHz , and 1MHz
3. The GMBUS controller can be attached to the selected GPIO pin pairs.
4. 7 or 10-Bit Slave Address and 8- or 16-bit index.
5. Hardware byte counter to track the data transmissions/reception
6. Timing source from 250MHz ungated PCI-Express clock.
7. There is a double buffered data register and a 9 bit counter to support 0 byte to 256 byte transfers.
8. The slave device can cause a stall by pulling down the clock line (Slave Stall), or delay the slave acknowledge response.
9. The master controller detects and reports time out conditions for a stall from a slave device or delayed or missing slave acknowledge.
10. Interrupt may optionally be generated from a GMBUS Timeout error.

The byte counter register is a read/write register, and in receiving mode, is used to track the data bytes received. There is a status register to indicate the error condition, data buffer busy, time out, and data complete acknowledgement.

Note: There is no support for ring buffer based operation of GMBUS. The GMBUS is controlled by a set of memory mapped IO registers. Status is reported through the GMBUS status register.

2.4.6 GMBUS0—GMBUS Clock / Port Select

Address Offset:	5100h
Default Value:	0000 0000h
Normal Access:	Read/Write
Size:	32 bits
Double Buffered:	No

The GMBUS0 register will set the clock rate of the serial bus and the device the controller is connected to. The clock rate options are 50 KHz, 100 KHz, 400 KHz, and 1MHz. This register should be set before the first data valid bit is set, because it will be read only at the very first data valid bit, and not read during the period of the transmission until stop is issued and next first data valid bit is set.

Bit	Description
31:11	Reserved
10:8	**GMBUS Rate Select:** These two bits select the rate that the GMBUS will run at. It also defines the AC timing parameters used. It should only be changed when between transfers when the GMBUS is idle. 1xx = **Reserved.** 000 = 100 KHz 001 = 50 KHz 010 = 400 KHz 011 = 1 MHz for SDVO
7	**Hold Time extension:** This bit selects the hold time on the data line driven from the GMCH. 0 = Hold time of 0ns 1 = Hold time of 300 ns
6:3	Reserved
2:0	**Pin Pair Select:** This field selects an GMBUS pin pair for use in the GMBUS communication. Use the table above to determine which pin pairs are available for a particular device and the intended function of that pin pair. Note that it is not a straight forward mapping of port numbers to pair select numbers. 000 = None (disabled) 001 = Dedicated Control/GMBUS Pins [DevCL] – LCTRCLKA, LCTRLCLKB SSC Clock Device 010 = Dedicated Analog Monitor DDC Pins (DDC1DATA, DDC1CLK) 011 = [DevCL] - Integrated Digital Panel DDC Pins, LVDS or DP D 100 = Reserved 101 = Reserved 110 = Reserved 111 = D connector control signals

2.4.7 GMBUS1—GMBUS Command/Status

Address Offset:	5104h
Default Value:	0000 0000h
Normal Access:	Read/Write (Write Protect except bit 31)
Size:	32 bits
Double Buffered:	no

This register lets the software indicate to the GMBUS controller the slave device address, register index, and indicate when the data write is complete.

When the **SW_CLR_INT** bit is asserted, all writes to the GMBUS2, GMBUS3, and GMBUS4 registers are discarded. The GMBUS1 register writes to any other bit except the **SW_CLR_INT** are also lost. Reads to these registers always work normally regardless of the state of the **SW_CLR_INT** bit.

Bit	Description
31	**Software Clear Interrupt (SW_CLR_INT):** This bit must be clear for normal operation. Setting the bit , then clearing it acts as local reset to the GMBUS controller. This bit is commonly used by software to clear a BUS_ERROR when a slave device delivers a NACK. 0 = If this bit is written as a zero when its current state is a one, will clear the **HW_RDY** bit and allows register writes to be accepted to the GMBUS registers (Write Protect Off). This bit is cleared to zero when an event causes the HW_RDY bit transition to occur. 1 = Asserted by software after servicing the GMBUS interrupt. Setting this bit causes the INT status bit to be cleared. Setting (1) this bit also asserts the **HW_RDY** bit (until this bit is written with a 0). When this bit is set, no writes to GMBUS registers will cause the contents to change with the exception of this bit which can be written.
30	**Software Ready (SW_RDY):** Data handshake bit used in conjunction with **HW_RDY** bit. 0 = De-asserted via the assertion event for **HW_RDY** bit 1 = When asserted by software, results in de-assertion of **HW_RDY** bit
29	**Enable Timeout (ENT)** Enables timeout for slave response. When this bit is enabled and the slave device response has exceeded the timeout period, the GMBUS Slave Stall Timeout Error interrupt bit is set. 0 = disable timeout counter 1 = enable timeout counter
28	Reserved

Bit	Description
27:25	**Bus Cycle Select** 000 = No GMBUS cycle is generated. 001 = GMBUS cycle is generated without an INDEX, with no STOP, and ends with a WAIT 010 = Reserved 011 = GMBUS cycle is generated with an INDEX, with no STOP, and ends with a WAIT 100 = Generates a STOP if currently in a WAIT or after the completion of the current byte if active. 101 = GMBUS cycle is generated without an INDEX and with a STOP 110 = Reserved 111 = GMBUS cycle is generated with an INDEX and with a STOP GMBUS cycle will always consist of a START followed by Slave Address, followed by an optional read or write data phase. A read cycle with an index will consist of a START followed by a Slave Address a WRITE indication and the INDEX and then a RESTART with a Slave Address and an optional read data phase. The GMBUS cycle will terminate either with a STOP or by entering a wait state. The WAIT state is exited by generating a STOP or by starting another GMBUS cycle. **This can only cause a STOP to be generated if a GMBUS cycle is generated, the GMBUS is currently in a data phase, or it is in a WAIT phase:** Note that the three bits can be decoded as follows: 27 = STOP generated 26 = INDEX used 25 = cycle ends in a WAIT
24:16	**Total Byte Count** (9-bits). This determines the total number of bytes to be transferred during the DATA phase of a GMBUS cycle. The DATA phase can be prematurely terminated by generating a STOP while in the DATA phase (see Bus Cycle Select). Do not change the value of this field during GMBUS cycles transactions.
15:8	**8-bit GMBUS Slave Register Index (INDEX):** This field specifies the 8-bits of index to be used for the generated bus write transaction or the index used for the WRITE portion of the WRITE/READ pair. It only has an effect if the enable Index bit is set. Do not change this field during a GMBUS transaction.
7:1	**7-bit GMBUS Slave Address (SADDR):** When a GMBUS cycle is to be generated using the Bus Cycle Select field, this field specifies the value of the slave address that is to be sent out. For use with 10-bit slave address devices, set this value to 11110XXb (where the last two bits (xx) are the two MSBs of the 10-bit address) and the slave direction bit to a write. This is followed by the first data byte being the 8 LSBs of the 10-bit slave address. Special Slave Addresses 0000 000R = General Call Address 0000 000W = Start byte 0000 001x = CBUS Address 0000 010x = Reserved 0000 011x = Reserved 0000 1xxx = Reserved 1111 1xxx = Reserved 1111 0xxx = 10-Bit addressing
0	**Slave Direction Bit:** When a GMBUS cycle is to be generated based on the Bus Cycle Select, this bit determines if the operation will be a read or a write. A read operation with the index enabled will perform a write with just the index followed by a re-start and a read. 1 = Indicates that a Read from the slave device operation is to be performed. 0 = Indicates that a Write to slave device operation is to be performed.

2.4.8 GMBUS2—GMBUS Status Register

Address Offset:	05108h
Default Value:	0000 0800h
Normal Access:	Read/Write (Write Protect)
Size:	32 bits
Double Buffered:	No

Bit	Description
31:16	Reserved
15	**INUSE** 0 = read operation that contains a zero in this bit position indicates that the GMBUS engine is now acquired and the subsequent reads of this register will now have this bit set. Writing a 0 to this bit has no effect. 1 = read operation that contains a one for this bit indicates that the GMBUS is currently allocated to someone else and "In use". Once set, a write of a 1 to this bit indicates that the software has relinquished the GMBUS resource and will reset the value of this bit to a 0. Software wishing to arbitrate for the GMBUS resource can poll this bit until it reads a zero and will then own usage of the GMBUS controller. This bit has no effect on the hardware, and is only used as semaphore among various independent software threads that don't know how to synchronize their use of this resource that may need to use the GMBUS logic. Writing a one to this bit is software's indication that the software use of this resource is now terminated and it is available for other clients.
14	**Hardware Wait Phase (HW_WAIT_PHASE):** Read-Only 0 = The GMBUS engine is not in a wait phase. 1 = Set when GMBUS engine is in wait phase. Wait phase is entered at the end of the current transaction when that transaction is selected not to terminate with a STOP. Once in a WAIT_PHASE, the software can now choose to generate a STOP cycle or a repeated start (RESTART) cycle followed by another GMBUS transaction on the GMBUS.
13	**Slave Stall Timeout Error** Read-Only. This bit indicates that a slave stall timeout has occurred. It is tied to the Enable Timeout (ENT) bit. 0 = No slave timeout has occurred. 1 = A slave acknowledge timeout has occurred
12	**GMBUS Interrupt Status** Read-Only. This bit indicates that an event that causes a GMBUS interrupt has occurred. 0 = The conditions that could cause a GMBUS interrupt have not occurred or this bit has been cleared by software assertion of the **SW_CLR_INT** bit. 1 = GMBUS interrupt event occurred. This interrupt must have been one of the types enabled in the GMBUS4 register.

Bit	Description
11	**Hardware Ready (HW_RDY):** Read-Only. This provides a method of detecting when the current software client routine can proceed with the next step in a sequence of GMBUS operations. This data handshake bit is used in conjunction with the **SW_RDY** bit. When this bit is changed to asserted by the GMBUS controller, it results in the de-assertion of the **SW_RDY** bit. 0 **=** Condition required for assertion has not occurred or when this bit was a one and: 0 **SW_RDY** bit has been asserted. 1 During a GMBUS read transaction, after the each read of the data register. 2 During a GMBUS write transaction, after each write of the data register. 3 **SW_CLR_INT** bit has been cleared. 1 = This bit is asserted under the following conditions: a) After a reset or when the transaction is aborted by the setting of the **SW_CLR_INT** bit. b) When an active GMBUS cycle has terminated with a STOP. c) When during a GMBUS write transaction, the data register needs and can accept another four bytes of data. d) During a GMBUS read transaction, this bit is asserted when the data register has four bytes of new data or the read transaction DATA phase is complete and the data register contains the last few bytes of the read data. This bit resumes to normal operation when the **SW_CLR_INT** bit is written to a 0.
10	**NAK Indicator (was previously called Slave Acknowledge Timeout Error SATOER):** Read-Only. 0 = No bus error has been detected or **SW_CLR_INT** has been written as a zero since the last bus error. 1 = Set by hardware if any expected device acknowledge is not received from the slave within the timeout.
9	**GMBUS Active (GA):** Read-Only. This is a status bit that indicates whether the GMBUS controller is in an IDLE state or not. 0 = The GMBUS controller is currently IDLE. 1 = This indicates that the bus is in START, ADDRESS, INDEX, DATA, WAIT, or STOP Phase. Set when GMBUS hardware is not IDLE.
8:0	**Current Byte Count:** Read-Only. Can be used to determine the number of bytes currently transmitted/received by the GMBUS controller hardware. Set to zero at the start of a GMBUS transaction data transfer and incremented after the completion of each byte of the data phase. Note that because reads have internal storage, the byte count on a read operation may be ahead of the data that has been accepted from the data register.

2.4.9 GMBUS3—GMBUS Data Buffer

Address Offset:	0510Ch
Default Value:	0000 0000h
Normal Access:	Read/Write (Write Protect)
Size:	32 bits

This is data read/write register. This register is double buffered. Bit 0 is the first bit sent or read, bit 7 is the 8th bit sent or read, all the way through bit 31 being the 32nd bit sent or read. For GMBUS write operations with a non-zero byte count, this register should be written with the data before the GMBUS cycle is initiated. For byte counts that are greater than four bytes, this register will be written with subsequent data only after the HW_RDY status bit is set indicating that the register is now ready for additional data. For GMBUS read operations, software should wait until the HW_RDY bit indicates that the register contains the next set of valid read data before reading this register.

Bit	Description
31:24	**Data Byte 3:**
23:16	**Data Byte 2:**
15:8	**Data Byte 1:**
7:0	**Data Byte 0:**

2.4.10 GMBUS4—GMBUS Interrupt Mask

Address Offset:	05110h
Default Value:	0000 0000h
Normal Access:	Read/Write
Size:	32 bits

Bit	Description
31:5	Reserved
4:0	**Interrupt Mask:** This field specifies which GMBUS interrupts events may contribute to the setting of gmbus interrupt status bit in second level interrupt status register PIPEASTAT. Bit 4: GMBUS Slave stall timeout Bit 3: GMBUS NAK Bit 2: GMBUS Idle Bit 1: Hardware wait (GMBUS cycle without a stop has completed) Bit 0: Hardware ready (Data has been transferred) 0 = Disable this type of GMBUS interrupt 1 = Enable this type of GMBUS interrupt

2.4.11 GMBUS5—2 Byte Index Register

Address Offset:	05120h
Default Value:	0000h
Normal Access:	Read/Write
Size:	32 bits
Double Buffered:	no

This register provides a method for the software indicate to the GMBUS controller the 2 byte device index.

Bit	Description
31	**2 Byte Index Enable:** When this bit is asserted (1), then bits 15:00 are used as the index. Bits 15:8 are used in the first byte which is the most significant index bits. The slave index in the GMBUS1<15:8> are ignored. Bits 7:0 are used in the second byte which is the least significant index bits.
30:16	Reserved
15:00	**2 Byte Slave Index:** This is the 2 byte index used in all GMBUS accesses when bit 31 is asserted (1).

2.5 Display Clock Control Registers (06000h–06FFFh)

The registers described in this section are used across products. However, slight changes may be present in some registers (i.e., for features added or removed), or some registers may be removed entirely.

The following list of registers may contain one or more differences between products. This table is provided for convenience only, please check each instruction separately to determine its exact impact on a specific product implementation.

[DevCL] When one or more display pipes are enabled, the bit 31 Enable HPLL off during Self Refresh should be disabled before accessing the 6XXXh MMIO register address space. Software must follow these steps:

1. Disable bit 31 Enable HPLL off during Self Refresh (if enabled and one display pipe is enabled)
2. Wait for next vblank (switch from hrawclk back to cdclk will occur)
3. Access the 6XXXh address space as needed
4. Re-enable Enable HPLL off during Self Refresh bit 31

Note that the wait on next vblank step requires an enabled display pipe.

Table 2-2. Simultaneous Display Capabilities on a Single Display Pipe

	Integrated LVDS*	Integrated TV*	CRT
Integrated LVDS*		No (1)	No (3)

Reserved	No (2, 3)	No (1)	Yes
Reserved	No (2, 3)	No (1)	Yes
Reserved	No (1, 2, 3)	No (1)	No (1)
Reserved	No (2)	No (1)	Yes
Reserved	No (2,3)	No (1)	No (4)
Reserved	No (2)	No (1)	No (4)
Integrated TV*	No (1)		No (1)
CRT	No (3)	No (1)	

NOTES:

1. Down config only

* Not available on UMA [DevBW]

Green = LVDS ([DevCL] only)

Pixel Data Rate	Dot Clock	Dual Channel?	External Clock	Data Clock Rate	Multiplier
20-25MHz	100-125MHz	NO	100-125MHz	1.0-1.25GHz	5x
25-50MHz	100-200MHz	NO	100-200MHz	1.0-2.0GHz	4x
50-100MHz	100-200MHz	NO	100-200MHz	1.0-2.0GHz	2x
100-270MHz	100-270MHz	NO	100-270MHz	1.0-2.7GHz	1x
25-112MHz	25-112MHz	NO	25-112MHz	175-784MHz	1x
80-224MHz	80-224MHz	NO	80-224MHz	280-784MHz	1x

Display Modes	Display Clock Frequency Range (MHz)
CRT DAC	20-400
Reserved	100-270
LVDS (Single Channel)	20-112
LVDS (Dual Channel)	80-224
TV-Out on Serial DVO [DevCL]	100-200

The PLL frequency selection must be done such that the internal VCO frequency is within its limits. The PLL Frequency is based on the selected register and the following formula. The post divider register value limits are different for Serial-DVO and LVDS modes.

Reference Frequency:
96MHz or 100MHz for LVDS.

DotClk_Frequency = (ReferenceFrequency * (5* (M1+2)+(M2+2)) / (N+2))/ (P1* P2)

Item	Units	Range	Notes
Dot Clock	Frequency	20-400	MHz (Combining ALL modes)
VCO	Frequency	1400-2800	MHz
N – Counter	Value	3-8	
M – Counter	Value	70-120	M=5*(M1+2)+(M2+2)
M1 and M2		M1 > M2	
M1	Value	10-20	
M2	Value	5-9	
P-Div	Value	5-80	Combined P1 and P2 for sDVO/DAC mode
P-Div	Value	7-98]	Combined P1 and P2 for LVDS mode
P1-Div	Value	1-8	All modes

2.5.1 VGA0—VGA 0 Divisor Register (100.8 MHz dot clk, 25.175 MHz pixel rate)

Address Offset:	06000h–06003h
Default Value:	00031108h
Normal Attribute:	R/W
Size:	32 bits

This register defines one of the two standard VGA frequencies that can be selected from the VGA register clock control bits for use in VGA Native modes. These values default to the proper VGA standard native frequency using the default 96MHz reference clock.

Bit	Description
31:22	Reserved: Write as zero.
21:16	**VGA0 N-Divisor:** N-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.
15:14	Reserved: Write as zero.
13:8	**VGA0 M1-Divisor:** M-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.
7:6	Reserved: MBZ
5:0	**VGA0 M2-Divisor:** M-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.

2.5.2 VGA1 —VGA 1 Divisor Register (113.280 MHz dot clk, 28.322 MHz pixel rate)

Address Offset:	06004h–06007h
Default Value:	00031406h
Normal Attribute:	R/W
Size:	32 bits

This register defines one of the two standard VGA frequencies that can be selected from the VGA register clock control bits for use in VGA Native modes. These values default to the proper VGA native frequency using the default 96MHz reference clock. These registers are only used for native mode VGA and not for centered or panel fitted VGA.

Bit	Description
31:22	Reserved: Write as zero.
21:16	**VGA 1 N-Divisor:** N-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.
15:14	Reserved: Write as zero.
13:8	**VGA0 M1-Divisor:** M-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.
7:6	Reserved: Write as zero.
5:0	**VGA 1 M2-Divisor:** M-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.

2.5.3 VGA_PD—VGA Post Divisor Values

Address Offset:	06010h–06013h
Default Value:	00020002h
Normal Attributes:	R/W
Size:	32 bits

Bit	Description
31:26	Reserved: Write as zero.
25:24	**VGA1 P2 Clock Divide:** 00 = Divide by 10. This is used when Dot Clock =< 270MHz in sDVO or DAC modes 01 = Divide by 5. This is used when Dot Clock > 270MHz in sDVO or DAC modes 10 = Reserved 11 = Reserved
23:16	**VGA1P1 Post Divisor:** Writes to this byte cause the staging register contents to be written into the active register when in the VGA mode of operation. This will also occur when the VGA MSR register is written. 00000001b = Divide by one 00000010b = Divide by two 00000100b = Divide by three 00001000b = Divide by four 00010000b = Divide by five 00100000b = Divide by six 01000000b = Divide by seven 10000000b = Divide by Eight All other values are illegal and should not be used
15_10	Reserved: Write as zero.
9:8	**VGA0 P2 Clock Divide:** 00 = Divide by 10. This is used when Dot Clock =< 270MHz in sDVO or DAC modes 01 = Divide by 5. This is used when Dot Clock > 270MHz in sDVO or DAC modes 10 = Reserved 11 = Reserved
7:0	**VGA0 P1 Post Divisor:** Writes to this byte cause the staging register contents to be written into the active register when in the VGA mode of operation. This will also occur when the VGA MSR register is written. 00000001b = Divide by one 00000010b = Divide by two 00000100b = Divide by three 00001000b = Divide by four 00010000b = Divide by five 00100000b = Divide by six 01000000b = Divide by seven 10000000b = Divide by Eight All other values are illegal and should not be used

2.5.4 DPLLA_CTRL—DPLL A Control Register

Address Offset:	6014h–06017h
Default Value:	04020C00h
Normal Attribute:	R/W
Size:	32 bits

Bit	Description
31	**DPLL VCO Enable: This bit will enable or disable the PLL VCO. Pipe A** palette accesses require that the DPLL for that pipe be running. Disabling the PLL will cause the display clock to stop and the display pipe to be disabled. 0 = DPLL is disabled in its lowest power state (default) 1 = DPLL is enabled and operational
30	**DPLLA Serial DVO High Speed IO clock Enable** 0 = High Speed IO Clock Disabled **(default)** 1 = High Speed IO Clock Enabled
29	Reserved : Write as zero.
28	**VGA Mode Disable:** When in native VGA modes, writes to the VGA MSR register causes the value in the selected (by MSR bits) VGA clock control register to be loaded into the active register. This allows the VGA clock select to select the pixel frequency between the two standard VGA pixel frequencies. 0 = VGA MSR<3:2> Clock Control bits select DPLL A Frequency 1 = Disable VGA Control
27:26	**DPLLA Mode Select :** Configure the DPLLA for various supported Display Modes 00 = Reserved 01 = DPLLA in DAC/Serial DVO/UDI/Integrated TV mode 10 = DPLLA in LVDS mode (Mobile devices ONLY) otherwise RESERVED 11 = Reserved
25:24	**FPA0/FPA1 P2 Clock Divide:** 00 = Divide by 10. This is used when Dot Clock =< 270MHz in DAC mode 01 = Divide by 5. This is used when Dot Clock >270MHz 10 = Reserved 11 = Reserved **For DPLLA in LVDS mode, BITS(27:26)=10** 00 = Divide by 14. This is used in Single-Channel LVDS 01 = Divide by 7. This is used in Dual-Channel LVDS 10 = Reserved 11 = Reserved

Bit	Description
23:16	**FPA0 / FPA1 P1 Post Divisor:** Writes to this byte cause the staging register contents to be written into the active register when in the VGA mode of operation. This will also occur when the VGA MSR register is written. 00000001b = Divide by one 00000010b = Divide by two 00000100b = Divide by three 00001000b = Divide by four 00010000b = Divide by five 00100000b = Divide by six 01000000b = Divide by seven 10000000b = Divide by Eight All other values are illegal and should not be used
15	Reserved: Write as zero
14:13	**PLL Reference Input Select:** The PLL reference should be selected based on the display device that is being driven. The standard reference clock is used for CRT modes using the analog display port or LCD panels for both the sDVO connected transmitter or the integrated LVDS. TV Clock in should be selected when driving an sDVO connected TV encoder. 00 = DREFCLK (default is 96 MHz) 01 = Reserved 10 = SDVO TVCLKIN 11 = Spread spectrum input clock
12:9	**Parallel to Serial Load Pulse phase selection:** Programmable select bits to choose the relative phase of the high speed (10X) DPLL clock used for generating the parallel to serial load pulse for digital display port on PCIe. The relative phase is the number of flop delays (phase 0 represents 1 flop delay) of the 1X parallel data synchronization signal in the 10X clock domain. The earliest selectable clock phase is 4. A phase selection of 10 or greater simply extends the flop delay count to sample delayed data. 0100 = use clock phase-4 0101 = use clock phase-5 **0110 = use clock phase-6 (Default value)** 0111 = use clock phase-7 1000 = use clock phase-8 1001 = use clock phase-9 1010 = use clock phase-10 1011 = use clock phase-11 1100 = use clock phase-12 1101 = use clock phase-13 Phases 0 through 3 are not available for Load Pulse selection. **[DevCL]** The following programming is recommended for DevCL based on PV timing analysis: **1101 – use clock phase-13**

Bit	Description
8:0	[DevBW, DevCL, Reserved [DevCTG] Display Rate Select: This bit is only available when bits 17:16 of the PIPEACONF register are 00. When the VGA MSR clock control bits do not control the frequency selection (VGA display disabled), this bit selects between the two alternate frequencies select register values. The change of frequencies is triggered by a write to this register followed by the start of VBLANK for this display pipe. This is used to dynamically select between the high and low frequencies for the sync lock mode of operation. The frequency switch should occur without any visible interruption on the display device. When the PLL is disabled, updates to this bit take effect immediately. 0 = Select the clock rate programmed in FPA0 **1 = Select the clock rate programmed in FPA1**
7:0	[DevBW, DevCL] Reserved [DevCTG] FPA1 P1 Post Divisor: Writes to this byte cause the staging register contents to be written into the active register when in the VGA mode of operation. This will also occur when the VGA MSR register is written. 00000001b - Divide by one 00000010b - Divide by two 00000100b - Divide by three 00001000b - Divide by four 00010000b - Divide by five 00100000b - Divide by six 01000000b - Divide by seven 10000000b - Divide by Eight **All other values are illegal and should not be used**

2.5.5 DPLLB_CTRL—DPLL B Control Registers

Address Offset:	06018h–0601Bh
Default Value:	04020C00h
Normal Attribute:	R/W
Size:	32 bits

Bit	Description
31	**DPLLB VCO Enable: This bit will enable or disable the PLL VCO.** If the PLL is disabled, the display clock will stop and the display pipe will be disabled. 0 = DPLL is disabled in its lowest power state (default) 1 = DPLL is enabled and operational
30	**DPLLB Serial DVO High Speed IO clock Enable** 0 = High Speed IO Clock Disabled **(default)** 1 = High Speed IO Clock Enabled
29	Reserved : Write as zero.
28	**VGA Native Mode Disable:** VGA native mode uses two frequencies selected by a VGA register field. The two frequencies are programmed in the VGA0 and VGA1 registers and selected by the VGA register bits. Writes to the VGA MSR register causes the value in the selected (by MSR bits) VGA clock control register to be loaded into the active register and thereby select the proper VGA frequency. Setting this bit to a zero only makes sense when the VGA disable bit is not set. 0 = VGA MSR<3:2> Clock Control bits select DPLL B Frequency 1 = Disable Native mode VGA Control (High Res or VGA centered)
27:26	**DPLLB Mode Select :** Configure the DPLLB for various supported Display Modes 00 = Reserved 01 = DPLLB in DAC/Serial DVO/UDI/Integrated TV mode 10 = DPLLB in LVDS mode (Mobile devices ONLY) otherwise RESERVED 11 = Reserved
25:24	**FPB0/FPB1 P2 Clock Divide:** **For DPLLB in Serial DVO or DAC mode , BITS(27:26)=01** 00 = Divide by 10. This is used when Dot Clock =< 270MHz in DAC mode 01 = Divide by 5. This is used when Dot Clock >270MHz 10 = Reserved 11 = Reserved **For DPLLB in LVDS mode, BITS(27:26)=10** 00 = Divide by 14. This is used in Single-Channel LVDS 01 = Divide by 7. This is used in Dual-Channel LVDS 10 = Reserved 11 = Reserved

Bit	Description
23:16	**FPB0 / FPB1 P1 Post Divisor:** Writes to this byte cause the staging register contents to be written into the active register when in the VGA mode of operation. This will also occur when the VGA MSR register is written. 00000001b = Divide by one 00000010b = Divide by two 00000100b = Divide by three 00001000b = Divide by four 00010000b = Divide by five 00100000b = Divide by six 01000000b = Divide by seven 10000000b = Divide by Eight All other values are illegal and should not be used
15	Reserved : Write as zero.
14:13	**PLL Reference Input Select:** The PLL reference should be selected based on the display device that is being driven. The standard reference clock is used for CRT modes using the analog display port or LCD panels for both the sDVO connected transmitter or the integrated LVDS. Spread spectrum should only be selected when driving the internal LVDS. One of the TV Clock in references should be selected when driving a DVO connected TV encoder. 00 = DREFCLK (default 96 MHz) for DAC/Serial-DVO 01 = Reserved 10 = SDVO TVCLKIN 11 = Spread spectrum input clock (mobile devices only)
12:9	**Parallel to Serial Load Pulse phase selection:** Programmable select bits to choose the relative phase of the high speed (10X) DPLL clock used for generating the parallel to serial load pulse for digital display port on PCIe. The relative phase is the number of flop delays (phase 0 represents 1 flop delay) of the 1X parallel data synchronization signal in the 10X clock domain. The earliest selectable clock phase is 4. A phase selection of 10 or greater simply extends the flop delay count to sample delayed data. 0100 = use clock phase-4 0101 = use clock phase-5 **0110 = use clock phase-6 (Default value)** 0111 = use clock phase-7 1000 = use clock phase-8 1001 = use clock phase-9 1010 = use clock phase-10 1011 = use clock phase-11 1100 = use clock phase-12 1101 = use clock phase-13 Phases 0 through 3 are not available for Load Pulse selection. **[DevCL]** The following programming is recommended for DevCL based on PV timing analysis: **0101 – use clock phase-5**

Bit	Description
8	[DevBW, DevCL] Reserved [DevCTG] Display Rate Select: This bit is only available when bits 17:16 of the PIPEBCONF register are 00. When the VGA MSR clock control bits do not control the frequency selection (VGA display disabled), this bit selects between the two alternate frequencies select register values. The change of frequencies is triggered by a write to this register followed by the start of VBLANK for this display pipe. This is used to dynamically select between the high and low frequencies for the sync lock mode of operation. The frequency switch should occur without any visible interruption on the display device. When the PLL is disabled, updates to this bit take effect immediately. 0 = Select the clock rate programmed in FPB0 **1 = Select the clock rate programmed in FPB1**
7:0	[DevBW, DevCL] Reserved [DevCTG] FPB1 P1 Post Divisor: Writes to this byte cause the staging register contents to be written into the active register when in the VGA mode of operation. This will also occur when the VGA MSR register is written. 00000001b - Divide by one 00000010b - Divide by two 00000100b - Divide by three 00001000b - Divide by four 00010000b - Divide by five 00100000b - Divide by six 01000000b - Divide by seven 10000000b - Divide by Eight **All other values are illegal and should not be used**

2.5.6 FPA0—DPLL A Divisor Register 0

Address Offset: 06040h–06043h
Default Value: 00031108h
Normal Attribute: R/W
Size: 32 bit

Bit	Description
31:28	Reserved: Write as zero.
27	[DevBW, DevCL] Reserved: Write as zero. [DevCTG]Frequency doubler clock enable: this bit enables/ disables the frequency doubler clock. When the VCO clock to the doubler is disabled, the circuit does not dissipate power and its output clock is not available 0 = disables clock of frequency doubler (default) 1 = enables clock of frequency doubler
26:24	[DevBW, DevCL] Reserved: Write as zero. [DevCTG] Raw display clock duty cycle select: controls the duty cycle of the display 2x raw clock (using three bits for future expandability) 000 – display 2x raw-clock high-time will be between 3 and 4 DPLL VCO periods 001 – display 2x raw-clock high-time will be between 4 and 5 DPLL VCO periods 010 – display 2x raw-clock high-time will be between 5 and 6 DPLL VCO periods 011 – display 2x raw-clock high-time will be between 6 and 7 DPLL VCO periods 1xx - Undefined These bits determine the high time for the display 2x raw clock (which may optionally clock the display pipes during SR). The appropriate high-time needs to be selected so the display 2x raw clock can meet timing requirements in the 2D pipe. The default values of these bits (see below) will be determined by the DPLL VCO frequency. S/W should normally not change the default values (changing them might cause timing violations in the display core).These bits will be changed in test mode only DPLL VCO (Mhz) — Default bit settings 1600-1800 — 000 1900-2400 — 001 2500-3000 — 010 3100-3600 — 011
23:22	Reserved: Write as zero.
21:16	FPA0 N-Divisor: N-Divisor value calculated for the desired output frequency. The register value is programmed two less than the actual divisor.

Bit	Description
15:14	Reserved: Write as zero.
13:8	FPA0 M1-Divisor: M-Divisor value calculated for the desired output frequency. The register value is programmed to two less than the actual divisor.
7:6	Reserved: MBZ
5:0	FPA0 M2-Divisor: M-Divisor value calculated for the desired output frequency. . The register value is programmed two less than the actual divisor.

2.5.7 FPA1—DPLL A Divisor Register 1

Address Offset: 06044h–06047h
Default Value: 00031108h
Normal Attribute: R/W
Size: 32 bits

Bit	Description
31:27	Reserved: Write as zero.
26:24	[DevBW, DevCL]Reserved: Write as zero. [DevCTG] Raw display clock duty cycle select: controls the duty cycle of the display 2x raw clock (using three bits for future expandability) 000 – display 2x raw-clock high-time will be between 3 and 4 DPLL VCO periods 001 – display 2x raw-clock high-time will be between 4 and 5 DPLL VCO periods 010 – display 2x raw-clock high-time will be between 5 and 6 DPLL VCO periods 011 – display 2x raw-clock high-time will be between 6 and 7 DPLL VCO periods 1xx - Undefined These bits determine the high time for the display 2x raw clock (which may optionally clock the display pipes during SR). The appropriate high-time needs to be selected so the display 2x raw clock can meet timing requirements in the 2D pipe. The default values of these bits (see below) will be determined by the DPLL VCO frequency. S/W should normally not change the default values (changing them might cause timing violations in the display core).These bits will be changed in test mode only DPLL VCO (Mhz) — Default bit settings 1600-1800 — 000 1900-2400 — 001 2500-3000 — 010 3100-3600 — 011
23:22	Reserved: Write as zero.
21:16	FPA1 N-Divisor: N-Divisor value calculated for the desired output frequency. . The register value is two less than the actual divisor.
15:14	Reserved: Write as zero.
13:8	FPA1 M1-Divisor: M-Divisor value calculated for the desired output frequency. . The register value is two less than the actual divisor.
7:6	Reserved: MBZ

Bit	Description
5:0	FPA1 M2-Divisor: M-Divisor value calculated for the desired output frequency. . The register value is two less than the actual divisor.

2.5.8 FPB0—DPLL B Divisor Register

Address Offset:	06048h–0604Bh
Default Value:	00031108h
Normal Attribute:	R/W
Size:	32 bits

Bit	Description
31:28	Reserved: Write as zero.
27	[DevBW, DevCL] Reserved: Write as zero. [DevCTG]Frequency doubler clock enable: this bit enables/ disables the frequency doubler clock. When the VCO clock to the doubler is disabled, the circuit does not dissipate power and its output clock is not available 0 = disables clock of frequency doubler (default) 1 = enables clock of frequency doubler
26:24	[DevBW, DevCL] Reserved: Write as zero. [DevCTG] Raw display clock duty cycle select: controls the duty cycle of the display 2x raw clock (using three bits for future expandability) 000 – display 2x raw-clock high-time will be between 3 and 4 DPLL VCO periods 001 – display 2x raw-clock high-time will be between 4 and 5 DPLL VCO periods 010 – display 2x raw-clock high-time will be between 5 and 6 DPLL VCO periods 011 – display 2x raw-clock high-time will be between 6 and 7 DPLL VCO periods 1xx - Undefined These bits determine the high time for the display 2x raw clock (which may optionally clock the display pipes during SR). The appropriate high-time needs to be selected so the display 2x raw clock can meet timing requirements in the 2D pipe. The default values of these bits (see below) will be determined by the DPLL VCO frequency. S/W should normally not change the default values (changing them might cause timing violations in the display core).These bits will be changed in test mode only DPLL VCO (Mhz) — Default bit settings 1600-1800 — 000 1900-2400 — 001 2500-3000 — 010 3100-3600 — 011
23:22	Reserved: Write as zero.

Bit	Description
21:16	FPB0 N-Divisor: N-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.
15:14	Reserved: Write as zero.
13:8	FPB0 M1-Divisor: M-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.
7:6	Reserved: MBZ
31:28	Reserved: Write as zero.

2.5.9 FPB1—DPLL B Divisor Register 1

Address Offset:	0604Ch–0604Fh
Default Value:	00031108h
Normal Attribute:	R/W
Size:	32 bits

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:27</td><td><b>Reserved: Write as zero.</b></td></tr>
<tr><td>26:24</td><td>[DevBW, DevCL] Reserved: Write as zero.
[DevCTG] Raw display clock duty cycle select: controls the duty cycle of the display 2x raw clock (using three bits for future expandability)
000 – display 2x raw-clock high-time will be between 3 and 4 DPLL VCO periods
001 – display 2x raw-clock high-time will be between 4 and 5 DPLL VCO periods
010 – display 2x raw-clock high-time will be between 5 and 6 DPLL VCO periods
011 – display 2x raw-clock high-time will be between 6 and 7 DPLL VCO periods
1xx - Undefined
These bits determine the high time for the display 2x raw clock (which may optionally clock the display pipes during SR). The appropriate high-time needs to be selected so the display 2x raw clock can meet timing requirements in the 2D pipe. The default values of these bits (see below) will be determined by the DPLL VCO frequency. S/W should normally not change the default values (changing them might cause timing violations in the display core).These bits will be changed in test mode only
<table>
<tr><td>DPLL VCO (Mhz)</td><td>Default bit settings</td></tr>
<tr><td>1600-1800</td><td>000</td></tr>
<tr><td>1900-2400</td><td>001</td></tr>
<tr><td>2500-3000</td><td>010</td></tr>
<tr><td>3100-3600</td><td>011</td></tr>
</table></td></tr>
<tr><td>23:22</td><td>Reserved: Write as zero.</td></tr>
<tr><td>21:16</td><td>FPB1 N-Divisor: N-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.</td></tr>
<tr><td>15:14</td><td>Reserved: Write as zero.</td></tr>
<tr><td>13:8</td><td>FPB1 M1-Divisor: M-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.</td></tr>
</table>

Bit	Description
7:6	Reserved: MBZ
	FPB1 M2-Divisor: M-Divisor value calculated for the desired output frequency. The register value is two less than the actual divisor.

2.5.10 DPLL_TEST—DPLLA and DPLLB Test Register

Address Offset:	0606Ch-0606Fh
Default Value:	00010001h
Normal Attribute:	R/W
Size:	32 bits

Bit	Description
31:24	Reserved: Write as zero.
23:22	**DPLLB Pre-Divider select for 2GHz SDVO clock: DPLL pins OCPDPLL2GDIVSEL[1:0]** 00 = Divide by 1 (Selects all non-2GHz CTM modes) 01 = Divide by 2 (Selects 200/2GHz CTM) 10 = Divide by 4 (Selects 400/2GHz CTM) 11 = Reserved
21:20	**DPLLB Post Divider P2 select for Feedback core clock (DPLLLKCLKP):** Valid **ONLY** when DPLL pin OCPDPLLFDBKSEL is set to 1. DPLL pins OCPDFDBKP2DVSEL[1:0] **In Serial DVO or DAC mode** X0 = Clockout is forced to Gnd X1 = Divide by 10. All others = Reserved **In LVDS mode** X0 = Clockout is forced to Gnd X1 = Divide by 14. All others = Reserved
19	**DPLLB Input N-divider Bypass select:** DPLL pin OCPDPLLNDVBYPASS 0 = DPLL uses output of N-divider (Default) 1 = Output of N-Divider is By-passed
18	**DPLLB M-divider Bypass select:** DPLL pin OCPDPLLMDVBYPASS 0 = DPLL uses output of M-divider (Default) 1 = Output of M-Divider is By-passed

Bit	Description
17	**DPLLB Feedback clock select:** DPLL pin OCPDPLLFDBKSEL 0 = DPLL locks on internal Feedback (default) 1 = DPLL locks on feedback from Core (DPLL pin DLKFBCLK**)**
16	**Enable DPLLB Input Clock Buffer:** DPLL pin OCPDPLLBUFFEN 0 = Disables the Differential Input clock buffer 1 = Enables the Differential Input clock buffer (Default)
15:6	Reserved: Write as zero.
5:4	**DPLLA Post Divider P2 select for Feedback core clock (DPLLLKCLKP):** Valid **ONLY** when DPLL pin OCPDPLLFDBKSEL is set to 1. DPLL pins OCPDFDBKP2DVSEL[1:0] **In Serial DVO or DAC mode** X0 = Clockout is forced to Gnd X1 = Divide by 10. All others = Reserved **In LVDS mode** X0 = Clockout is forced to Gnd X1 = Divide by 14. All others = Reserved
3	**DPLLA Input N-divider Bypass select**. DPLL pin OCPDPLLNDVBYPASS 0 = DPLL uses output of N-divider (Default) 1 = Output of N-Divider is By-passed
2	**DPLLA M-divider Bypass select:** DPLL pin OCPDPLLMDVBYPASS 0 = DPLL uses output of M-divider (Default) 1 = Output of M-Divider is By-passed
1	**DPLLA Feedback clock select:** DPLL pin OCPDPLLFDBKSEL 0 = DPLL locks on internal Feedback (default) 1 = DPLL locks on feedback from Core (DPLL pin DLKFBCLK**)**
0	**Enable DPLLA Input Clock Buffer:** DPLL pin OCPDPLLBUFFEN 0 = Disables the Differential Input clock buffer 1 = Enables the Differential Input clock buffer (Default)

2.5.11 DEUC—Dynamic EU Control ([DevCL] only)

Address Offset:	6214–6215h
Default Value:	00h
Access:	R/W/L;
Size:	16 bits

This register generally provides control on whether 0, 1, or 2 eus should be gated dynamically. Lock control is provided by TCO[7].

Bit	Access	Default Value	Description
15:13	RW		**EU disable on ME clock throttling:** Specifies as a percentage the minimum throttling level of ep_2xclk, ep_clk, arc_clk that must occur to allow all EU's to be enabled if EU can be disabled based on ME activity. 000 = 0% - 12.4% 001 = 12.5% - 24.9% 010 = 25% - 37.4% 011 = 37.5% - 49.9% 100 = 50% - 62.4% 101 = 62.5% - 74.9% 110 = 75% - 87.4% 111 = 87.5% - 100%
12:6	RO	0h	Reserved
5	RO		**Dynamic EU Control Status** 0 = selected EUs are are not idle 1 = selected EUs are completely idle
4	RW	0h	**SW eu Control Enable** 0 = SW does not affect DEUC mode actions 1 = SW assertion of this bit does affect DEUC mode actions
3	RWL	0h	**Thermal sensor trip eu Control enable** 0 = Thermal sensor trip does not affect DEUC mode actions 1 = Thermal sensor trip does affect DEUC mode actions
2	RWL	0h	**ME on eu control enable** 0 = ME on does not affect DEUC mode actions 1 = ME on does affect DEUC mode actions
1:0	RW	0h	**DEUC Mode (Dynamic EU Control Mode)** 00 = no EUs are affected is on 01 = No more than 1 EU is affected (if one EU is already disabled, this setting not cause another to be disabled). If software control has been enabled with this mode, then one EU will be affected. 10 = At least 1 EU is affected (if one EU is already disabled, this setting will cause another to be disabled. If the reason for one already being disabled goes away, this setting will leave one disabled) 11 = 2 EUS are affected

2.5.12 DEUC—Dynamic EU Control (not for EDS publication ([DevCL] only)

Address Offset:	6214–6215h
Default Value:	00h
Access:	R/W/L;
Size:	16 bits

This register generally provides control on whether 0, 1, or 2 eus should be gated dynamically. Lock control is provided by TCO[7].

Bit	Access	Default Value	Description
15:13	RW		**EU disable on ME clock throttling:** Specifies as a percentage the minimum throttling level of ep_2xclk, ep_clk, arc_clk that must occur to allow all EU's to be enabled if EU can be disabled based on ME activity. 000 = 0% - 12.4% 001 = 12.5% - 24.9% 010 = 25% - 37.4% 011 = 37.5% - 49.9% 100 = 50% - 62.4% 101 = 62.5% - 74.9% 110 = 75% - 87.4% 111 = 87.5% - 100%
12:6	RO	0h	**Reserved (RSVD)**
5	RO		**Dynamic EU Control Status** 0 = selected EUs are are not idle 1 = selected EUs are completely idle
4	RW	0h	**SW eu Control Enable** 0 = SW does not affect DEUC mode actions 1 = SW assertion of this bit does affect DEUC mode actions
3	RWL	0h	**Thermal sensor trip eu Control enable** 0 = Thermal sensor trip does not affect DEUC mode actions 1 = Thermal sensor trip does affect DEUC mode actions
2	RWL	0h	**ME on eu control enable** 0 = ME on does not affect DEUC mode actions 1 = ME on does affect DEUC mode actions

Bit	Access	Default Value	Description
1:0	RW	0h	**DEUC Mode (Dynamic EU Control Mode)** 00 = no EUs are affected is on 01 = No more than 1 EU is affected (if one EU is already disabled, this setting not cause another to be disabled). If software control has been enabled with this mode, then one EU will be affected. 10 = At least 1 EU is affected (if one EU is already disabled, this setting will cause another to be disabled. If the reason for one already being disabled goes away, this setting will leave one disabled) 11 = 2 EUS are affected

2.6 Display Palette Registers (0A000h–AFFFh)

Display palettes provide a method for converting index data values to color values for VGA and 8-bpp indexed display modes. It also provides methods to gamma correct the true color data that was derived from either 16- or 32-bpp display modes. Accesses to the palette entries require that the core display clock is running at the time of the update. All devices support a Display Palette associated with a particular Display Pipe.

The Display Palettes can be accessed through two methods and operate in one of two modes. There are 256 8-bit entries per color channel in the palette in the 8-bit mode. In the 8-bit mode, each DWord write will load each of the three channels with a single byte. These entries are used for graphics color translation in indexed modes and gamma correction in true and high color modes.

The palette is also accessible via the VGA palette register I/O addresses and method when enabled through the VGA control bits. For VGA palette accesses through the VGA palette register I/O addresses, the palette can look as though there are only 6 bits per color component. When using the palette for VGA, it should be set to the 8-bit mode. The palette entries are always accessed as 24 bits within a DWord. Byte or word writes are not allowed, they must always be DWord. Accesses to the palette must be done with the DPLL for that pipe and display core enabled.

2.6.1 DPALETTE_A—Pipe A Display Palette

Address Offset: 0A000h–0A3FFh
Default: UUh
Normal Attributes: R/W (DWORD only)

Table 2-3. 8-Bit Mode

31 24	23 16	15 8	7 0
Reserved	Red Palette Entry	Green Palette Entry	Blue Palette Entry

Bit	Description
31:24	Reserved: Read-Only.
23:16	**Pipe A Red Palette Entry:**
15:8	**Pipe A Green Palette Entry:**
7:0	**Pipe A Blue Palette Entry:**

2.6.1.1 10-bit Programming Notes:

The 10-bit gamma correction curve is represented by specifying a set of reference points spaced equally along the curve. Red, Green, and Blue each have 129 reference points. The first 128 reference points are stored in the palette RAM, and the final value is stored in the GCMAX register. The first 128 reference points are 16 bits represented in a 10.6 format with 10 integer and 6 fractional bits. The final reference points are 17 bits represented in a 11.6 format with 11 integer and 6 fractional bits.

The appropriate reference point pairs (adjacent) are selected for each color, and the output is interpolated between these two reference point values.

To program the gamma correction reference points calculate the desired gamma curve for inputs from 0 to 1024.

Every 8th point on the curve (0,8,16...1016,1024) becomes a reference point. Convert the gamma value to the 10.6 format. The first 128 reference points are saved to the palette RAM, where the odd DWords contain the lower 8 bits of the reference point value, and the even DWords contain the upper 8 bits of the reference point value. The final 129th reference point is saved in the GCMAX register in 11.6 format..

Example equation for gamma curve of 2.2:

For (X = 0..1024) { gamma = [(X / 1024) ^ 2.2] * 1024 }

The curve is assumed to be flat or increasing, never decreasing.

The start and end reference points are not fixed values.

Figure 2-1. 10-bit Gamma Correction Curve

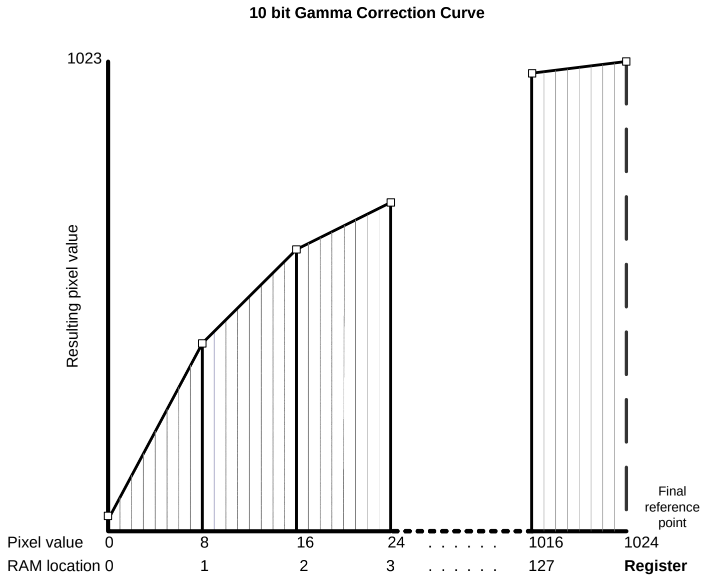


Table 2-4. 10-bit Mode (Even DWord)

31							24	23			22	21			16
Reserved								Red Base[1:0]				Red Fractional Address			
15							8	7							0
Green Base[1:0]				Green Fractional Address				Blue Base[1:0]				Blue Fractional Address			

Table 2-5. 10-bit Mode (Odd DWord)

31							24	23							16
Reserved								Red Base [9:2]							
15							8	7							0
Green Base[9:2]								Blue Base [9:2]							

2.6.2 DPALETTE_B—Pipe B Display Palette

Address Offset:	0A800h–0ABFFh
Default:	UUh
Normal Attributes:	R/W (DWORD only)

8-Bit Mode

Bit	Description
31:24	Reserved: Read-Only.
23:16	**Pipe B Red Palette Entry:**
15:8	**Pipe B Green Palette Entry:**
7:0	**Pipe B Blue Palette Entry:**

See DPALETTE_A for 10-bit Mode definitions.

2.7 Display Pipeline / Port Registers (60000h–6FFFFh)

2.7.1 Display Pipeline A

2.7.1.1 HTOTAL_A—Pipe A Horizontal Total Register

Address Offset: 60000h–60003h
Default Value: 00000000h
Normal Access: R/W

Bit	Description
31:29	Reserved: Write as zero.
28:16	**Pipe A Horizontal Total Display Clocks:** This 13-bit field provides Horizontal Total up to 8192 pixels encompassing the Horizontal Active Display period, front/back border and retrace period. Any pending event (HSYNC, ACTIVE, HBLANK) is reset at HTOTAL and the programmed sequence begins again. This field is programmed to the number of clocks desired minus one. This number of clocks needs to be a multiple of two when driving data out the digital port out the LVDS port in two channel mode. This value should always be equal or greater to the sum of the horizontal active and the horizontal blank, and border region sizes.
15: 12	Reserved: Write as zero.
11: 0	**Pipe A Horizontal Active Display Pixels:** This 12-bit field provides Horizontal Active Display resolutions up to 4096 pixels. Note that the first horizontal active display pixel is considered pixel number 0. The value programmed should be the (active pixels/line – 1). The number of active pixels will be limited to multiples of two pixels when driving the integrated LVDS port in two channel mode. For proper results during VGA centering mode this value needs to be large enough to fit the largest VGA mode supported, this should be at least 720/1440 pixels for standard VGA type modes or 640/1280 pixels if the nine-dot disable bit in the VGA control register is set. When using the internal panel fitting logic, the minimum horizontal size allowed will be three pixels.

2.7.1.2 HBLANK_A—Pipe A Horizontal Blank Register

Address Offset: 60004h–60007h
Default Value: 00000000h
Normal Access: R/W

Bit	Description
31:29	Reserved: Read-Only.
28:16	**Pipe A Horizontal Blank End:** This 13-bit field specifies the position of Horizontal Blank End expressed in terms of the absolute pixel number relative to the horizontal active display start. The value programmed should be the HBLANK End pixel position, where the first active pixel is considered position 0; the second active pixel is considered position 1, etc. Horizontal blank ending at the same point as the horizontal total indicates that there is no left hand border area. HBLANK size has a minimum value of 32 clocks. The number of clocks within blank needs to be a multiple of two when driving data out LVDS in two channel mode. The value loaded in the register would be equal to RightBorder+Active+HBlank-1. If this pipe is connected to the TVout port or Panel Fitter 2 the border must be zero. In that case this register is programmed to the same value as the HTOTAL register.
15: 13	Reserved: Read-Only.
12:0	**Pipe A Horizontal Blank Start:** This 13-bit field specifies the Horizontal Blank Start position expressed in terms of the absolute pixel number relative to the horizontal active display start. The value programmed should be the HBLANK Start pixel position, where the first active pixel is considered position 0; the second active pixel is considered position 1, etc. The number of clocks for both left and right borders need to be a multiple of two when driving data out the LVDS port in two channel mode. Horizontal blank should only start after the end of the horizontal active region. The value loaded in the register would be equal to RightBorder+Active-1. If this pipe is connected to the TVout port or Panel Fitter 2 the border must be zero. In that case this register is programmed to the same value as the HACTIVE register.

2.7.1.3 HSYNC_A—Pipe A Horizontal Sync Register

Address Offset: 60008h–6000Bh
Default Value: 00000000h
Normal Access: R/W

Bit	Description
31:29	Reserved: Write as zero.
28:16	**Pipe A Horizontal Sync End:** This 13-bit field specifies the horizontal Sync End position expressed in terms of the absolute pixel number relative to the horizontal active display start. The value programmed should be the HSYNC End pixel position, where the first active pixel is considered position 0; the second active pixel is considered position 1, etc. The number of clocks in the sync period needs to be a multiple of two when driving data out the LVDS port in two channel mode. This value should be greater than the horizontal sync start position and would be loaded with the Active+RightBorder+FrontPorch+Sync-1.
15: 13	Reserved: Read-Only.
12:0	**Pipe A Horizontal Sync Start:** This 13-bit field specifies the horizontal Sync Start position expressed in terms of the absolute pixel number relative to the horizontal active display start. The value programmed should be the HSYNC Start pixel position, where the first active pixel is considered position 0; the second active pixel is considered position 1, etc. Note that when HSYNC Start is programmed equal to HBLANK Start, both HSYNC and HBLANK will be asserted on the same pixel clock. It should never be programmed to less than HBLANK start. The number of cycles from the beginning of the line needs to be a multiple of two when driving data out the LVDS port in two channel mode. This register should not be less than the horizontal active end. This register should be loaded with the Active+RightBorder+FrontPorch-1.

2.7.1.4 VTOTAL_A—Pipe A Vertical Total Register

Address Offset: 6000Ch–6000Fh
Default Value: 00000000h
Normal Access: R/W

Bit	Description
31:29	Reserved: Read-Only.
28:16	**Pipe A Vertical Total Display Lines:** This 13-bit field provides Vertical Total up to 8192 lines encompassing the Vertical Active Display Lines, top/bottom border and retrace period. The value programmed should be the number of lines required minus one. Vertical total needs to be large enough to be greater than the sum of the vertical active, vertical border, and the vertical blank regions. The vertical counter is incremented on the leading edge of the horizontal sync. For interlaced display modes, this indicates the total number of lines in both fields. In interlaced modes, hardware automatically divides this number by 2 to get the number of lines in each field.
15:12	Reserved: Read-Only.
11:0	**Pipe A Vertical Active Display Lines:** This 12-bit field provides vertical active display resolutions up to 4096 lines. It should be programmed with the desired number of lines minus one. When using the internal panel fitting logic, the minimum vertical active area must be three lines. For interlaced display modes, this indicates the total number of lines in both fields. In interlaced modes, hardware automatically divides this number by 2 to get the number of lines in each field.

2.7.1.5 VBLANK_A—Pipe A Vertical Blank Register

Address Offset: 60010h–60013h
Default Value: 00000000h
Normal Access: R/W

Bit	Description
31:29	Reserved: Read-Only.
28:16	**Pipe A Vertical Blank End:** This 13-bit field specifies the Vertical Blank End position expressed in terms of the absolute Line number relative to the vertical active display start. The value programmed should be the VBLANK End line position, where the first active line is considered line 0, the second active line is considered line 1, etc. The end of vertical blank should be after the start of vertical blank and before or equal to the vertical total. This register should be loaded with the Vactive+BottomBorder+VBlank-1. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical blank end in each field. It does not count the two half lines that get added when operating in modes with half lines. If this pipe is connected to the TVout port or Panel Fitter 2 the border must be zero. In that case this register is programmed to the same value as the VTOTAL register.
15:13	Reserved: Read-Only.
12: 0	**Pipe A Vertical Blank Start:** This 13-bit field specifies the Vertical Blank Start expressed in terms of the absolute line number relative to the vertical active display start. The value programmed should be the VBLANK Start line position, where the first active line is considered line 0, the second active line is considered line 1, etc. Minimum vertical blank size is required to be at least three lines. Blank should start after the end of active. This register is loaded with the Vactive+BottomBorder-1. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical blank start in each field. It does not count the two half lines that get added when operating in modes with half lines. If this pipe is connected to the TVout port or Panel Fitter 2 the border must be zero. In that case this register is programmed to the same value as the VACTIVE register.

2.7.1.6 VSYNC_A—Pipe A Vertical Sync Register

Address Offset:	60014h–60017h
Default Value:	00000000h
Normal Access:	R/W

Bit	Description
31:29	Reserved: Read-Only.
28:16	**Pipe A Vertical Sync End:** This 13-bit field specifies the Vertical Sync End position expressed in terms of the absolute Line number relative to the vertical active display start. The value programmed should be the VSYNC End line position, where the first active line is considered line 0, the second active line is considered line 1, etc. This register should be loaded with Vactive+BottomBorder+FrontPorch+Sync-1. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical sync end in each field. It does not count the two half lines that get added when operating in modes with half lines.
15: 13	Reserved: Read-Only.
12:0	**Pipe A Vertical Sync Start:** This 13-bit field specifies the Vertical Sync Start position expressed in terms of the absolute line number relative to the vertical active display start. The value programmed should be the VSYNC Start line position, where the first active line is considered line 0, the second active line is considered line 1, etc. This register would be loaded with Vactive+BottomBorder+FrontPorch-1. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical sync start in each field. It does not count the two half lines that get added when operating in modes with half lines.

2.7.1.7 PIPEASRC—Pipe A Source Image Size

Address Offset: 6001Ch–6001Fh
Default Value: 00000000h
Normal Access: Read/Write

Bit	Description
31:28	Reserved: Write as zero
27:16	**Pipe A Horizontal Source Image Size:** This 12-bit field specifies Horizontal source image size up to 4096. This determines the size of the image created by the display planes sent to the blender. The value programmed should be the source image size minus one. The actual source size must be two times the programmed value in the "pixel multiply mode. It must represent a size that is a multiple of two (even numbers) when driving the LVDS port in two channel mode. This implies that for this mode, the value programmed will always be an odd number. Except in the case of panel fitting internal or in an external device, this register field would be programmed to a value identical to the horizontal active. This is the only register of the timing registers that is allowed to be programmed while the pipe is enabled.
15: 12	Reserved: Write as zero
11: 0	**Pipe A Vertical Source Image Size:** This 12-bit field specifies the vertical source image size up to 4096 lines. This determines the size of the image created by the display planes sent to the blender. The value programmed should be the source image size minus one. Note that the actual number of lines needs to be at least twice the planes programmed value when in the pixel multiply mode. Except in the case of panel fitting internal or in an external device, this register field would be programmed to a value identical to the vertical active. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical source image size in each field.

2.7.1.8 BCLRPAT_A— Pipe A Border Color Pattern Register

Address Offset: 60020h–60023h
Default Value: 00000000h
Normal Access: Read/Write
Size: 32 bits

This register value determines what color should be sent to the display in the border region, the space between the end of active and the beginning of blank and the end of blank and the beginning of active.

Bit	Description
31:24	Reserved
23:16	**Pipe A Border Red Channel Value:**
15:8	**Pipe A Border Green Channel Value:**
7:0	**Pipe A border Blue Channel Value:**

2.7.1.9 VSYNCSHIFT_A— Vertical Sync Shift Register

Address Offset:	60028h–6002Bh
Default Value:	00000000h
Normal Access:	Read/Write
Size:	32 bits

Bit	Description
31:13	Reserved: Write as zero.
12:0	**Pipe A Second Field Vertical Sync Shift:** This value specifies the vertical sync alignment for the start of the interlaced second field expressed in terms of the absolute pixel number relative to the horizontal active display start. This value will only be used if the PIPEACONF is programmed to an interlaced mode using vsync shift. Otherwise a legacy value of floor[htotal / 2] will be used. Typically, the interlaced second field vertical sync should start one pixel after the point halfway between successive horizontal syncs, so the value of this register should be programmed to: (horizontal sync start - floor[horizontal total / 2]) (use the actual horizontal sync start and horizontal total values and not the minus one values programmed into registers). This vertical sync shift only occurs during the interlaced second field. In all other cases the vertical sync start position is aligned with horizontal sync start.

2.7.2 Display Pipeline B

2.7.2.1 HTOTAL_B—Pipe B Horizontal Total Register

Address Offset: 61000h–61003h
Default Value: 00000000h
Normal Access: R/W (Write Protect by Panel Power Sequencer on [DevCL])

Bit	Description
31:29	Reserved: Write as zero.
28:16	**Pipe B Horizontal Total Display:** See pipe A description.
15:12	**Reserved:** Write as zero.
11:0	**Pipe B Horizontal Active Display:** See pipe A description

2.7.2.2 HBLANK_B—Pipe B Horizontal Blank Register

Address Offset: 61004h–61007h
Default Value: 00000000h
Normal Access: R/W (Write Protect by Panel Power Sequencer on [DevCL])

Bit	Description
31:29	Reserved. Write as zero.
28:16	**Pipe B Horizontal Blank End:** See pipe A description
15:12	Reserved: Write as zero.
12:0	**Pipe B Horizontal Blank Start:** See pipe A description.

2.7.2.3 HSYNC_B—Pipe B Horizontal Sync Register

Address Offset: 61008h–6100Bh
Default Value: 00000000h
Normal Access: R/W (Write Protect by Panel Power Sequencer on [DevCL])

Bit	Description
31:29	Reserved: Write as zero.
28: 16	**Pipe B Horizontal Sync End:** See pipe A description.
15:12	Reserved: Write as zero.
12:0	**Pipe B Horizontal Sync Start:** See pipe A description

2.7.2.4 VTOTAL_B—Pipe B Vertical Total Register

Address Offset: 6100Ch–6100Fh
Default Value: 00000000h
Normal Access: R/W (Write Protect by Panel Power Sequencer on [DevCL])

Bit	Description
31:29	Reserved: Write as zero.
28:16	**Pipe B Vertical Total Display:** See pipe A description.
15:11	Reserved: Write as zero.
11:0	**Pipe B Vertical Active Display:** See pipe A description.

2.7.2.5 VBLANK_B—Pipe B Vertical Blank Register

Address Offset: 61010h–61013h
Default Value: 00000000h
Normal Access: R/W (Write Protect by Panel Power Sequencer on [DevCL])

Bit	Description
31:29	Reserved: Write as zero.
28:16	**Pipe B Vertical Blank End:** See pipe A description.
15:12	**Reserved:** Write as zero.
12:0	**Pipe B Vertical Blank Start:** See pipe A description.

2.7.2.6 VSYNC_B—Pipe B Vertical Sync Register

Address Offset: 61014h–61017h
Default Value: 00000000h
Normal Access: R/W (Write Protect by Panel Power Sequencer on [DevCL])

Bit	Description
31:29	Reserved: Write as zero.
28:16	**Pipe B Vertical Sync End:** See pipe A description.
15:12	Reserved: Write as zero.
12:0	**Pipe B Vertical Sync Start:** See pipe A description.

2.7.2.7 PIPEBSRC—Pipe B Source Image Size

Address Offset: 6101Ch–6101Fh
Default Value: 00000000h
Normal Access: Read/Write

Bit	Description
31:28	Reserved: Write as zero
27:16	**Pipe B Horizontal Source Image Size:** See pipe A description.
15:11	Reserved: Write as zero
11:0	**Pipe B Vertical Source Image Size:** See pipe A description.

2.7.2.8 BCLRPAT_B—Pipe B Border Color Pattern Register

Address Offset: 61020h–61023h
Default Value: 00000000h
Normal Access: Read/Write

This register determines the color sent during the border region, the periods between the end of blank and the start of active and the end of active and the start of blank. Also same color will be sent during pseudo border period. VGA border color is determined by the VGA border “overscan” color register.

Bit	Description
31:24	Reserved
23:16	**Pipe B Red channel color value:**
15:8	**Pipe B Green channel color value:**
7:0	**Pipe B Blue channel color value:**

2.7.2.9 VSYNCSHIFT_B— Vertical Sync Shift Register

Address Offset: 61028h–6102Bh
Default Value: 00000000h
Normal Access: Read/Write

Bit	Description
31:13	Reserved: Write as zero.
12:0	**Pipe B Second Field Vertical Sync Shift:** This value specifies the vertical sync alignment for the start of the interlaced second field expressed in terms of the absolute pixel number relative to the horizontal active display start. This value will only be used if the PIPEBCONF is programmed to an interlaced mode using vsync shift. Otherwise a legacy value of floor[htotal / 2] will be used. Typically, the interlaced second field vertical sync should start one pixel after the point halfway between successive horizontal syncs, so the value of this register should be programmed to: (horizontal sync start - floor[horizontal total / 2]) (use the actual horizontal sync start and horizontal total values and not the minus one values programmed into registers). This vertical sync shift only occurs during the interlaced second field. In all other cases the vertical sync start position is aligned with horizontal sync start.

2.8 Display Port Control

2.8.1 ADPA—Analog Display Port Register

Address Offset: 61100h–61103h
Default Value: 00000000h
Normal Access: Read/Write

Bit	Description
31	**ADPA (Analog Display Port A) Enable:** This bit enables or disables the DAC output. It has no effect on the horizontal or vertical sync outputs. 1 = Enable. This bit enables analog port DAC 0 = Disable the DAC and go to a low power state. [DevBW] When port multiply is not programmed to 1X and CRT is the only port enabled on a pipe, it should be enabled as follows: 1) Set bit #31 to '1' to enable the DAC and bits #[11:10] to '11' to disable hsync/vsync CRT output. 2) Wait for Vsync. 3) Clear bits #[11:10] to '00'
30	**Pipe Select:** Determines which display pipe will feed this DAC port. This only applies to dual display pipe devices. It is reserved in all other devices. 0 = Pipe A 1 = Pipe B
29:16	Reserved: Software must preserve the contents of these bits.

Bit	Description
15	**ADPA HSYNC and VSYNC Polarity Select:** VGA VSYNC and HSYNC polarity bits are ignored on this port when this bit is clear. This should only be set if this port is used for VGA display in the VGA native mode. 1 = Use VGA registers to select VSYNC and HSYNC Polarities. (VGA Native Mode) 0 = Use bits 4 and 3 of this register (ADPA) to select VSYNC and HSYNC Polarities.
14	Reserved: Software must preserve the contents of this field.
13:12	Reserved
11:10	**Monitor DPMS:** (for CRT port) When the graphics device is in D0, these bits force the HSYNC/VSYNC output signal to the state specified in the polarity selection or allow the standard timing generated syncs to continue. These bits enable proper handling of DPMS monitors that support the D1 or D2 states during display device power management by toggling sync signals required. This field should always be loaded with the display device D state during display power management operations. 00 = Monitor in D0 Monitor On. (will not affect sync pulses) 01 = Monitor in D2. Monitor Suspend (HSYNC pulses, VSYNC does not.) 10 = Monitor in D1. Monitor Standby (VSYNC pulses, HSYNC does not) 11 = Monitor in D3. Monitor Off (Neither HSYNC nor VSYNC pulses)
9:5	Reserved : Software must preserve the contents of these bits.
4	**VSYNC Polarity Control:** The output VSYNC polarity is controlled either by the VGA control bits or this bit when in VGA modes. This is used to implement display modes that require inverted polarity syncs and to set the disabled state of the VSYNC signal. 1 = Active high. 0 = Active low.
3	**HSYNC Polarity Control:** According to the ADPA polarity select bit the HSYNC polarity is controlled by either the VGA sync polarity register bit or this bit. 1 = Active high. 0 = Active low.
2:0	Reserved: Software must preserve the contents of these bits.

2.8.2 LVDS—Digital Display Port Control ([DevCL and DevCTG])

Address offset:	61180h–61183h
Default:	4000 0000h
Normal Access:	Read/Write (Write Protect by Panel Power Sequencer on [DevCL])

Bit	Description
31	**LVDS Port Enable:** When disabled the LVDS port is inactive and in its low power state. Enabling the LVDS port changes the way that the PLL for this pipe is programmed. This bit must be set before the display PLL is enabled and the port is power sequenced on using the panel power sequencing logic. 0 = The port is disabled and all LVDS pairs are powered down. 1 = The port is enabled (port must be enabled before powering up a connected panel)
30	**LVDS Port Pipe Assign** 0 = The port gets data from pipe A 1 = The port gets data from pipe B (default)
29:26	Reserved
25	**Dither Enable:** This bit enables or disables (bypassing) 8-6-bit color dithering function. The usage of this bit would be on for 18-bpp panels and off for 24-bpp panels. 0 = disabled 1 = enabled
24	**Data Format Select:** Combined with the other control bits it selects the LVDS data format. Other control bits in this register determine if two channel is enabled and 18 or 24-bit color is enabled. 0 = 1x18.0, 2x18.0, 1x24.0 or 2x24.0 1 = 1x24.1 or 2x24.1
23	**LE Control Enable:** This bit is used when the second channel control signal field indicates that we are using the LE instead of HS and the two channel mode is enabled. In single channel mode, this bit has no effect. 0 = Send 0 on second channel HS (B2<2>) 1 = Send 1 on second channel HS
22	**LF Control Enable:** This bit is used when the second channel control signal field indicates that we are using the LF instead of VS and two channel mode is enabled. In single channel mode, this bit has no effect. 0 = Send 0 on second channel VS (B2<3>) 1 = Send 1 on second channel VS
21	**VSYNC Polarity:** This controls the polarity of the VSYNC indicator that is sent over the LVDS connection. Panels may require one or the other polarity or work with either polarity. 0 = No inversion (1=active) 1 = Invert the sense (0=active)

Bit	Description
20	**HSYNC Polarity (LP Invert):** This controls the polarity of the HSYNC indicator that is sent over the LVDS connection. Panels may require one or the other polarity or work with either polarity. 0 = No inversion (1=active) 1 = Invert the sense (0=active)
19	**DE invert:** This controls the polarity of the DE indicator that is sent over the LVDS connection. 0 = No inversion of DE (1=active) 1 = Invert the sense of DE (0=active)
18:17	**Second Channel Control Signals:** This bit only applies to the two channel modes of operation it has no effect in single channel modes. 00 = Send DE, HS, VS on second channel if enabled 01 = Reserved 10 = Do not send DE, HS, VS on second channel use zero instead 11 = Use DE=0, HS=LE, VS=LF on second channel
16	**Channel Reserved Bits** 0 = Send 0 for the channel reserved bits 1 = Send duplicate data bit for reserved bits
15	**LVDS Border Enable:** This selects whether the border data should be included in the active display data sent to the panel. Border should be used when in VGA centered (un-scaled) mode or when scaling a 4:3 source image to a wide screen panel (typical 16:9). 0 = Border to the LVDS transmitter is disabled. DE (Display Enable) is used. 1 = Border to the LVDS transmitter is enabled. Blank# is used as DE for the panel.
14:11	Reserved
10	**Buffer Power Down State:** This bit selects the state of the LVDS buffers during a powered down state caused by the power sequence logic power down. This selection will be made based on the connected panel requirements. 0 = Zero Volts (Driven on both lines of the pairs) 1 = Tri-State (High impedance state)
9:8	**ClkA, A0, A1, A2 Control:** This field controls the A0-A2 data pairs and CLKA. It sets the highest level of activity that is allowed on these lines when the panel is powered on. Power sequencing for LVDS connected panels overrides the control. When the power sequencer is in the power down mode all signals are in the power down state. 00 = Power Down all A channel signals including A3 (0V) 01 = Power up – A0, A1, A2 Data bits forced to 0,Timing active, Clock Active 10 = Reserved 11 = Power up – Data lines and clock active

Bit	Description
7:6	**Eight bit color channel A3, (B3) Control:** This field can control both the A3 and B3 data pairs. Enabling those pairs indicates the selection of 8-bit per color channel mode. It sets the highest level of activity that is allowed on these lines when the panel is powered on. The A3 pair will only be powered up if both this field and the A0, A1, A2, CLKA field indicates that the pair should be powered up and will only be active if both indicate that it should be active. The B3 pair will only be powered up if both this field and the B0, B1, B2, (B3) field indicates that the pair should be powered up and will only be active if both indicate that it should be active. Power sequencing for LVDS connected panels overrides the control. When the power sequencer is in the power down mode all signals are in the power down state. 00 = Power Down all signals A3, B3 (common mode) 01 = Power up – A3, (B3) Data (pixel data not control) lines forced to 0 output 10 = Reserved 11 = Power up – A3, (B3) Data lines active
5:4	**Two channel mode ClkB Control:** When in two channel mode, this field controls the CLKB pair. It sets the highest level of activity that is allowed on these lines when the panel is powered on. The CLKB pair should only be powered up if the B0, B1, B2, (B3) field indicates that the second channel should be powered up and will only be active if both indicate that it should be active. Power sequencing for LVDS connected panels overrides the control. 00 = Power Down CLKB (common mode) 01 = Power up – CLKB Forced to 0 10 = Reserved 11 = Power up – Clock B active
3:2	**Two channel mode B0, B1, B2 Control:** This field controls both the set B0-B2 data pairs. It sets the highest level of activity that is allowed on these lines when the panel is powered on. Power sequencing for LVDS connected panels overrides the control. During single channel operation (1x18.0), these bits need to be both zero. Two channel operation is selected by setting them to ones. Note that the second clock can be optionally enabled or disabled by the two channel mode ClkB control field. 00 = Power Down all signals including B3 and CLKB 01 = Power up – B0, B1, B2, Data lines forced to 0, timing is active 10 = Reserved 11 = Power up – Data lines active (color and timing)
1:0	Reserved

2.9 Panel Registers

2.9.1 Panel Power Sequencing Registers ([DevBW], [DevCL], [DevCTG])

2.9.1.1 PP_STATUS—Panel Power Status Register ([DevCL])

Address offset:	61200-61203h
After Reset:	08000000h
Normal Access:	Read-Only

Bit	Description
31	**Panel Power On Status** 0 = Indicates that the panel power down sequencing has completed. A power cycle delay may be currently active. It is safe and allowed to program pipe timing and DPLL registers. If this bit is not a zero, it activates the register write protect and writes to those registers will be ignored unless the write protect key value is set in the panel sequencing control register. 1 = In conjunction with bits Power Sequence Progress field and Power Cycle Delay Active, this bit set to a one indicates that the panel is currently powered up or is currently in the power down sequence and it is unsafe to change the pipe timing and DPLL registers for the pipe that is assigned to the LVDS output. If the LVDS port is selected as the target for the panel control, Software is responsible for enabling the LCD display by writing a "1" to the port enable bit only after all pipe timing, DPLL registers are properly programmed, and the PLL has locked to the reference signal. This bit is cleared (set to "0") only after the panel power down sequencing is completed.
30	**Require Asset Status:** This bit indicates the status of programming of the display PLL and the selected display port. This a power on cycle will not be allowed unless this status indicates that the required assets are programmed and ready for use. 0 = All required assets are not properly programmed. 1 = All required assets are ready for the driving of a panel. The following conditions determine that the assets are ready: 1) Display Pipe PLL Enabled and frequency locked (bit-31 of DPLL Control Register for the pipe attached to the LVDS port). 2) Display Pipe Enabled (bit-31 of PIPECONF—Pipe Configuration Register. For the pipe attached to the LVDS port) 3) LVDS Port is Programmed Enabled
29:28	**Power Sequence Progress** 00 = Indicates that the panel is not in a power sequence 01 = Indicates that the panel is in a power up sequence (may include power cycle delay) 10 = Indicates that the panel is in a power down sequence 11 = Reserved

Bit	Description
27	**Power Cycle Delay Active:** Power cycle delays occur after a panel power down sequence or after a hardware reset. On reset, a power cycle delay will occur using the default value for the timing. 0 = A power cycle delay is not currently active 1 = A power cycle delay (T4) is currently active
26:4	Reserved
3:0	**Internal Sequence State (For test/debug)** 0000 = Power Off Idle (S0.0) 0001 = Power Off, Wait for cycle delay (S0.1) 0010 = Power Off (S0.2) 0011 = Power Off (S0.3) 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = Reserved 1000 = Power On Idle (S1.0) 1001 = Power On, (S1.1) 1010 = Power On, (S1.2) 1011 = Power On, Wait for cycle delay (S1.3) 1100 = Reserved 1101 = Reserved 1110 = Reserved 1111 =– Reset

2.9.1.2 PP_CONTROL—Panel Power Control Register ([DevCL])

Address offset: 61204-61207h
Default: 000000000h
Normal Access: Read/Write

Bit	Description
31:16	**Write Protect Key** ABCD – Write protect off When this field is programmed to anything except the write protect off setting and the panel is either powered up or in the process of a power up sequence, a set of registers involved in generation of panel timing or control become write-protected. Any write cycles to those write-protected registers, while they will complete as normal, will not change the value of the register when write-protected. When this register field contains the write protect off key value, write protect will be unconditionally disabled. In situations where the LVDS port is unused, the port should remain powered down and the write protect will be inactive. This field in normal operation should be left to all zeros and never programmed with the key value. It exists only to allow testing and workarounds. **List of Write-protected registers:** (LVDS and Panel sequencing Registers): LVDS – Digital Display Port Control – Address: 61180h–61183h Panel power on sequencing delays - Address: 61208-6120Bh Panel power off sequencing delays – Address: 6120Ch – 6120Fh Panel power cycle delay and Reference Divisor – Address: 61210h – 61213 (DPLL registers): DPLL Control Registers FPB0—DPLL Divisor Register FPB1—DPLL Divisor Register 1 (Display Pipe timing registers except source size) HTOTAL—Horizontal Total Register HBLANK—Horizontal Blank Register HSYNC_—Horizontal Sync Register VTOTAL_—Vertical Total Register VBLANK_—Vertical Blank Register **VSYNC_—Vertical Sync Register**
15:3	Reserved

Bit	Description
2	Backlight Enable [DevCTG]: Enabling this bit enables the panel backlight if the embedded panel is DisplayPort, as indicated in bits 31:30 of the panel power on sequencing. Software must enable this bit after training the link, and disable it when disabling the panel power state target. 0 = Backlight disabled 1 = Backlight enabled **[DevCL] Reserved**
1	**Power Down on Reset:** Enabling this bit causes the panel to power down when a reset warning comes to the GMCH from the ICH. When system reset is initiated, the LVDS automatically begins the panel power down sequence. If the panel is not on during a reset event, this bit is ignored. 0 = Do not run panel power down sequence when reset is detected 1 = Run panel power down sequence when system is reset
0	**Power State Target:** Writing this bit can occur any time, it will only be used at the completion of any current power cycle. 0 = The panel power state target is off, if the panel is either on or in a power on sequence, a power off sequence is started as soon as the panel reaches the power on state. This may include a power cycle delay. If the panel is currently off, there is no change of the power state or sequencing done. 1= The panel power state target is on, if the panel is in either the off state or a power off sequence, if all pre-conditions are met, a power on sequence is started as soon as the panel reaches the power off state. This may include a power cycle delay. If the panel is currently off, there is no change of the power state or sequencing done. While the panel is on or in a power on sequence, the register write lock will be enabled.

2.9.1.3 PP_ON_DELAYS—Panel Power on Sequencing Delays ([DevCL])

Address offset:	61208-6120Bh
Default:	00000000h
Normal Access:	Read/Write (Write Protect by Panel Power Sequencer on [DevCL]

Bit	Description
31:30	**Panel Control port select:** These bits define to which port the LCD panel is connected. This is used for automatic control of the panel power. If the selected port is disabled or if the port is not on pipe-B, then, the power sequence will not allow a panel power up. 00 = Panel is connected to the LVDS display port 01 = Reserved 10 = Reserved 11 = Reserved The selection of non-existent ports is not allowed (such as DVO A). This programming will disable panel power sequencing logic.
29	Reserved
28:16	**Power up delay:** Programmable value of panel power sequencing delay during panel power up. This provides the time delay for the T1+T2 time sequence. The time unit used is the 100us timer.
15:13	Reserved
12:0	**Power on to Backlight enable delay:** Programmable value of panel power sequencing delay during panel power up. This provides the time delay for the T5 time sequence. The time unit used is the 100us timer.

2.9.1.4 PP_OFF_DELAYS—Panel Power off Sequencing Delays ([DevCL])

Address offset:	6120Ch – 6120Fh
Default:	00000000h
Normal Access:	Read/Write (Write Protect by Panel Power Sequencer on Mobile products

Bit	Description
31:29	Reserved
28:16	**Power Down delay:** Programmable value of panel power sequencing delay during power up. This provides the time delay for the T3 time sequence. The time unit used is the 100us timer.
15:13	Reserved
12:0	**Power Backlight off to power down delay:** Programmable value of panel power sequencing delay during power down. This provides the time delay for the Tx time sequence. The time unit used is the 100us timer.

2.9.1.5 PP_DIVISOR—Panel Power Cycle Delay and Reference Divisor ([DevCL])

Address offset:	61210h – 61213
Default:	00270F04h
Normal Access:	Read/Write (Write Protect by Panel Power Sequencer on Mobile Products)

This register selects the reference divisor and controls how long the panel must remain in a power off condition once powered down. This has a default value that allows a timer to initiate directly after device reset. If the panel limits how fast we may sequence from up to down to up again. Typically this is .5-1.5 sec. but limited to 400ms in the SPWG specification. This register forces the panel to stay off for a programmed duration. Special care is needed around reset and D3 cold situations to conform to power cycle delay specifications.

Bit	Description
31:8	**Reference divider:** This field provides the value of the divider used for the creation of the panel timer reference clock. The output of the divider is used as the fastest of the three time bases (100us) for all other timers. The other time bases are divided from this frequency. The value of zero should not be used. When it is desired to divide by N, the actual value to be programmed is (N/2)-1. The value should be (100*RefinMHz/2)-1. The default value assumes the default value for the display core clock that is for [DevCL] a 200MHz reference value. The following are examples for other memory speeds. **Display Core Frequency** **Value of Field** 233MHz 2D81h 200MHz 270Fh 133MHz 19F9h
7:5	Reserved
4:0	**Power Cycle Delay:** Programmable value of time panel must remain in a powered down state after powering down. For devices coming out of reset, the default values will define how much time must pass before a power on sequence can be started. This field uses the .1 S time base unit from the divider. If the panel power on sequence is attempted during this delay, the power on sequence will commence once the power cycle delay is complete. Writing a value of 0 selects no delay or is used to abort the delay if it is active. During the initial power up reset, a D3 cold power cycle, or a user instigated system reset, the timer will be set to the default value and the count down will begin after the de-assertion of reset. Writing this field to a zero while the count is active will abort this portion of the sequence. This corresponds to the T4 of the SPWG specification. Note: Even if the panel is not enabled, the T4 count happens after reset. This register needs to be programmed to a "+1" value. For instance for meeting the SPWG specification of 400mS, program 5 to achieve at least 400mS delay prior to powerup.

2.9.2 Panel Fitting Registers

2.9.2.1 PFIT_CONTROL—Panel Fitting Controls

Address offset:	61230h
Default:	20000000h
Normal Access:	Read/Write

Bit	Description
31	**Panel Fitting Enabled:** Disables the panel fitting function by forcing pixels to bypass. Panel fitting must be disabled when running VGA native modes or interlaced modes on the same pipe. Panel fitting should be enabled or disabled before the pipe is enabled. 0 = Bypass the panel fitting (1:1 ratio) 1 = Enable panel fitting (Ratios include 1:1)
30:29	**Pipe Select:** Indicates the pipe attached to the panel fitter 00 = Panel fitter is attached to Display Pipe A. 01 = Panel fitter is attached to Display Pipe B. **This is the default after reset**. 10 = Reserved for pipe C 11 = Reserved for pipe D
28:26	**Scaling Mode** 000 = Auto-scale (source and destination should have the same aspect ratios) 001 = Programmed scaling: Values in register 61234h will be used for horizontal and vertical scaling factors 010 = Pillarbox (example: 4:3 to 16:9 auto conversion) use only when destination has wider aspect ratio than source 011 = Letterbox (example: 16:9 to 4:3 auto conversion) use only when destination has taller aspect ratio than source 1xx = Reserved
25:24	**Filter Coefficient Select:** Selects the set of predefined filter coefficients to use for panel fitting 00 = Fuzzy filtering 01 = Crisp edge enhancing filtering 10 = Median between fuzzy and crisp filtering 11 = Reserved
23	**Debug: Force Two Line Mode**
22	**Debug: Force Three Pixel Mode when in Two Line Mode**
21:19	**Debug: Create extra stalls in VGA mode** 000 = No stall 001 = 33% stall 010 = 50% stall 011 = 66% stall 100 = 75% stall 101 = 80% stall 110 = 90% stall 111 = Reserved
18:5	Reserved**:**
4	Reserved : write as zero
3:0	Reserved

2.9.2.2 PFIT_PGM_RATIOS—Programmed Panel Fitting Ratios

Address offset:	61234h
Default:	00000000h
Normal Access:	Read/Write

When programmed scaling mode (Panel Fitting Controls 28:26 = "001") is selected, this determines the vertical and horizontal ratios used for panel fitting scaling. The values should be based on the source sizes and active sizes programmed into the pipe timing registers. The values written into the register should be rounded to the proper number of bits for the best precision. The value programmed should be [(source size register value +1) / (active size register value + 1)]

When programmed scaling mode is not selected, read back of this register gives the auto-generated vertical and horizontal scaling ratios used for panel fitting scaling. Register writes will be ignored. The ratios are calculated each VBLANK. When in HiRes modes, the values are based on the source sizes and active sizes programmed into the pipe timing registers. When in VGA modes it is determined by the VGA source sizes calculated by the VGA and active sizes from the pipe timing registers. VGA source sizes may have invalid values due to mode change transitions. These will eventually be correct when the mode change is complete. The value read is internally generated [(source size register value +1) / (active size register value + 1)]

For each register field the MSB is the 1 bit integer value and the lower 12 bits are the fractional value. A value of 1.0 will indicate 1:1 scaling. A value greater than 1.0 will indicate downscaling. A value less than 1.0 will indicate upscaling. The vertical and horizontal ratios are usually identical, except for when source and active aspect ratios differ.

Bit	Description
31:29	Reserved : Reads as zeros
28:16	**Panel fitting vertical ratio:** Vertical scaling ratio for panel fitting.
15:13	Reserved – Reads as zeros
12:0	**Panel fitting horizontal ratio:** Horizontal scaling ratio for panel fitting.

2.9.2.3 Reserved (Used to be Auto Scaling Ratios Readback)

Address offset:	61238h
Default:	00000000h
Normal Access:	Read-Only

Bit	Description
31:0	Reserved

2.9.2.4 Reserved (Used to be Scaling Initial Phase)

Address offset: 6123Ch
Default: 01000100h
Normal Access: Read/Write

Bit	Description
31:0	Reserved

2.9.3 Backlight Control and Modulation Histogram Registers ([DevCL])

2.9.3.1 BLC_PWM_CTL2—Backlight PWM Control Register 2 ([DevCL])

Address offset: 61250h
Default: 00000000h
Normal Access: Read/Write
Double Buffered: No

Bit	Description
31	**PWM Enable:** This bit enables the PWM counter logic 0 = PWM disabled (drives 0 always) 1 = PWM enabled
30	**BLM Legacy Mode** 0 = PWM Duty Cycle is derived from the Backlight Duty Cycle only 1 = PWM Duty Cycle is a combination of Backlight Duty Cycle and Legacy Backlight Control (LBPC) Note: ‘1’ implies the Duty Cycle = if(BPC[7:0] <> xFF) then BPCR[15:0] * BPC[7:0] Else BPCR[15:0]
29	**PWM Pipe assignment:** This bit assigns PWM to a pipe. The PWM counter will run off of this pipe’s PLL. The PWM function must be disabled in order to change the value of this field. 0 = Pipe A 1 = Pipe B
28	**Backlight Polarity.** This field controls the polarity of the PWM signal. 0 = Active High 1 = Active Low
27	Reserved

Bit	Description
26	**Phase-In Interrupt Status:** This bit will be set by hardware when a Phase-In interrupt has occurred. Software will clear this bit by writing a '1', which will reset the interrupt generation. [DevCL-A,B] **Reserved**
25	**Phase In Enable:** Setting this bit enables a PWM phase in based on the programming of the Phase In registers below. This bit clears itself when the phase in is completed.
24	**Phase In Interrupt Enable:** Setting this bit enables an interrupt to be generated when the PWM phase in is completed.
23:16	**Phase In time base:** This field determines the number of VBLANK events that pass before one increment occurs. 0 = invalid 1 = 1 vblank 2 = 2 vblanks etc.
15:8	**Phase In Count:** This field determines the number of increment events in this phase in. Writes to this register should only occur when hardware-phase-ins are disabled. Reads to this register can occur any time, where the value in this field indicates the number of increment events remaining to fully apply a phase-in request as hardware automatically decrements this value. A value of 0 is invalid.
7:0	**Phase In Increment:** This field indicates the amount to adjust the PWM duty cycle register on each increment event. This is a two's complement number.

2.9.3.2 BLC_PWM_CTL—Backlight PWM Control Register ([DevCL])

Address offset:	61254h
Default:	00000000h
Normal Access:	Read/Write

Bit	Description
31:16	**Backlight Modulation Frequency.** This field determines the number of time base events in total for a complete cycle of modulated backlight control. This field is normally set once during initialization based on the frequency of the clock that is being used and the desired PWM frequency. This value represents the period of the PWM stream in display core clocks multiplied by 128.
15:0	**Backlight Duty Cycle.** This field determines the number of time base events for the active portion of the PWM backlight control. This should never be larger than the frequency field. A value of zero will turn the backlight off. A value equal to the backlight modulation frequency field will be full on. This field gets updated when it is desired to change the intensity of the backlight, it will take effect at the end of the current PWM cycle. This value represents the active time of the PWM stream in display core clock periods multiplied by 128.

2.9.3.3 BLM_HIST_CTL—Image Enhancement Histogram Control Register ([DevCL]

Address offset:	61260h
Default:	00000000h
Normal Access:	Read/Write
Double buffer:	No

Bit	Description
31	**Image Enhancement Histogram Enabled:** This bit enables the Image Enhancement histogram logic to collect data. 0 = Image histogram is disabled 1 = The Image histogram is enabled. When this bit is changed from a zero to a one, histogram calculations will begin after the next VBLANK of the assigned pipe.
30	**Image Enhancement Modification Table Enabled:** This bit enables the Image Enhancement modification table. 0 = disabled 1 = enabled. When this bit is changed from a zero to a one, modifications begin after the next VBLANK of the assigned pipe.
29	**Image Enhancement Pipe assignment:** This bit assigns the IE function to a pipe. IE events will be synchronized to the VBLANK of the selected pipe. The IE function must be disabled in order to change the value of this field. 0 = Pipe A 1 = Pipe B
28:25	Reserved : Always write as 0s.
24	**Histogram Mode Select:** 0 = YUV Luma Mode 1 = HSV Intensity Mode - Reserved on [DevCL]
23:16	**Sync to Phase In Count:** This field indicates the phase in count number on which the Image Enhancement table will be loaded if the Sync to Phase in is enabled.
15	Reserved : Always write as 0.
14:13	**Enhancement mode:** 00 = Direct look up mode 01 = Additive mode 10 = Multiplicative mode - Reserved on [DevCL] 11 = Reserved
12	**Sync to Phase In:** Setting this bit enables the double buffered registers to be loaded on the phase in count value specified instead of the next vblank.
11	**Bin Register Function Select:** This field indicates what data is being written to or read from the bin data register. 0 = Bin Threshold Count. A read from the bin data register returns that bin's threshold value from the most recent vblank load event (guardband threshold trip). Valid range for the Bin Index is 0 to 31. 1 = Bin Image Enhancement Value. Valid range for the Bin Index is 0 to 32

Bit	Description
10:7	Reserved : Always write as 0's.
6:0	**Bin Register Index:** This field indicates the bin number whose data can be accessed through the bin data register. This value is automatically incremented by a read or a write to the bin data register if the busy bit is not set.

2.9.3.4 Image Enhancement Bin Data Register ([DevCL])

Address offset:	61264h
Default:	00000000h
Normal Access:	Depends on function)
Double buffer:	Depends on function

Writes to this address are steered to the correct register by programming the Bin Register Function Select and the Bin Register Index.

Function 0 usage (Threshold Count) this Function is Read-Only

Bit	Description
31	**Busy Bit:** If set , the engine is busy, the rest of the register is undefined. If clear, the register contains valid data.
30:22	Reserved
21:0	**Bin Count:** The total number of pixels in this bin, value is updated at the start of each vblank.

Function 1 usage (Image Enhancement) this Function is Read/Write

Bit	Description
31:10	Reserved
9:0	**Image Bin Correction Factor** The correction value for this bin, writes to this register are double buffered on the next vblank if in normal mode, or on the phase in Sync event frame if it is enabled. The value written here is the 10bit corrected channel value for the lowest point of the bin.

For Additive mode the data format is a two’s complement number. For multiply mode the correction factor is a 1.9 fixed point number is the range 0.0 to 1.9999. (ex, 1.000 = 200h, 1.5 = 300h, 0.5 -= 100h)

2.9.3.5 Histogram Threshold Guardband Register ([DevCL])

Address offset:	61268h
Default:	00000000h
Normal Access:	Read/Write
Double buffer:	Yes

Bit	Description
31	**Histogram Interrupt enable:** 0 = Disabled 1 = Enabled. This generates a histogram interrupt once a Histogram event occurs.
30	**Histogram Event status:** When a Histogram event has occured, this will get set by the hardware. For any more Histogram events to occur, the software needs to clear this bit by writing a '1'. The default state for this bit is '0'. 0 = Histogram event has not occurred. 1 = Histogram event has occurred. **[DevCL-A,B]: Read-Only**
29:22	**Guardband Interrupt Delay:** An interrupt is generated after this many consecutive frames of the guardband threshold being surpassed. This value is double buffered on start of vblank. A value of 0 is invalid.
21:0	**Threshold Guardband:** This value is used to determine the guardband for the threshold interrupt generation. This single value is used for all the segments. This value is double buffered on start of vblank

2.10 Display and Cursor Registers (70000h–7FFFFh)

These registers are memory mapped and accessible through normal 32 bit, 16 bit, or 8-bit accesses.

2.10.1 Display Pipeline A

2.10.1.1 PIPEA_DSL—Pipe A Display Scan Line

Memory Offset Address:	70000h
Default:	00000000h
Normal Access:	Read-Only

This register enables the read back of the display pipe vertical “line counter”. The display line value is from the display pipe A timing generator and is reset to zero at the beginning of a scan. The value increments at the leading edge of HSYNC and can be safely read any time. For normal operation, scan line zero is the first active line of the display. When in VGA centering mode, the scan line 0 is the 1st active scan line of the pseudo border not the centered active VGA image. In interlaced display timings, the scan line counter provides the current line in the field. One field will have a total number of lines that is one greater than the other field.

Programming Note: In order to cause the scan line logic to report the correct Line Counter value, the corresponding Display Pipeline timing registers must be programmed to valid, non-zero (e.g., 640x480 @ 60Hz) values before enabling the Pipe or programming VGA timing and enabling native VGA.

Bit	Descriptions
31	**[DevCTG] Current Field:** Provides read back of the current field being displayed on display pipe A. Non-TV mode: 0 = first field (odd field) 1 = second field (even field) TV mode: 1 = first field (odd field) 0 = second field (even field) **[DevBW] and [DevCL] Reserved:** Read only.
30:13	Reserved: Read-only.
12:0	**Line Counter for Display [12:0]:** Provides read back of the display pipe A vertical line counter. This is an indication of the current display scan line to be used by software to synchronize with the display.

2.10.1.2 PIPEA_SLC—Pipe A Display Scan Line Count Range Compare

Memory Offset Address:	70004h
Default:	00000000h
Normal Access:	Read-Only

This register can be written via the command stream processor using the MI_LOAD_SCAN_LINES_INCL or MI_LOAD_SCAN_LINES_EXCL commands. They can safely be accessed at any time.

The Top and Bottom Line Count Compare registers are compared with the display line values from display A timing generator. The Top compare register operator is a less than or equal, while the Bottom compare register operator is a greater than or equal. The results of these 2 comparisons are communicated to the command stream controller for generating interrupts, status, and command stream flow control ("wait for within range" and "wait for not within range"). For range check, the value programmed should be the (**desired value – 1)**, so for line 0, the value programmed is VTOTAL, and for line 1, the value programmed is 0.

Bit	Descriptions
31	**Inclusive / Exclusive:** 1 = Inclusive: within the range. 0 = Exclusive: outside of the range.
30:29	Reserved: Read-only.
28:16	[DevCTG] Start Scan Line Number: This field specifies the starting scan line number of the Scan Line Window. Format = U16 in scan lines, where scan line 0 is the first line of the display frame. Range = [0,Display Buffer height in lines-1]. [DevBW] and [DevCL] End Scan Line Number: This field specifies the ending scan line number of the Scan Line Window. Format = U16 in scan lines, where scan line 0 is the first line of the display frame. Range = [0, Display Buffer height in lines-1].

Bit	Descriptions
15:13	**Reserved: Read only.**
12:0	[DevCTG] End Scan Line Number: This field specifies the ending scan line number of the Scan Line Window. Format = U16 in scan lines, where scan line 0 is the first line of the display frame. Range = [0, Display Buffer height in lines-1]. [DevBW] and [DevCL] Start Scan Line Number: This field specifies the starting scan line number of the Scan Line Window. Format = U16 in scan lines, where scan line 0 is the first line of the display frame. **Range = [0,Display Buffer height in lines-1].**

2.10.1.3 PIPEACONF—Pipe A Configuration Register

Memory Offset Address: 70008h
Default: 00000000h
Normal Access: Read/Write double buffered

Bit	Descriptions
31	**Pipe A Enable:** Setting this bit to the value of one, turns on pipe A. This must be done before any planes are enabled on this pipe. Changing it to a zero should only be done when all planes that are assigned to this pipe have been disabled. Turning the pipe enable bit off disables the timing generator in this pipe. Plane disable occurs after the next VBLANK event after the plane is disabled. Synchronization pulses to the display are not maintained if the timing generator is disabled. Power consumption will be at its lowest state when disabled. A separate bit controls the DPLL enable for this pipe. Pipe timing registers should contain valid values before this bit is enabled. 0 = Disable 1 = Enable
30	**Pipe State:** This bit indicates the actual state of the pipe. Since there can be some delay between disabling the pipe and the pipe actually shutting off, this bit indicates the true current state of the pipe. 0 = Disabled 1 = Enabled
29	Reserved
28:27	**Frame Start Delay:** Used to delay the frame start signal that is sent to the display planes. Normal operation uses the default 00 value and test modes can use the delayed frame start to shorten the test time. Care must be taken to insure that there are enough lines during VBLANK to support this setting. 00 = Frame Start occurs on the first HBLANK after the start of VBLANK 01 = Frame Start occurs on the second HBLANK after the start of VBLANK 10 = Frame Start occurs on the third HBLANK after the start of VBLANK 11 = Frame Start occurs on the forth HBLANK after the start of VBLANK
26	Reserved
25	**FORCE_BORDER**: 0 = Normal Operation 1 = Color information is ignored and border color is substituted during active region
24	**Pipe A Gamma Unit Mode:** This bit selects which mode the pipe gamma correction logic works in. In the palette mode, it behaves as a 3X256x8 RAM lookup. VGA and indexed mode operation should use the palette in 8-bit mode. In the 10-bit gamma mode, it will act as a piecewise linear interpolation. Other gamma units such as in the overlay or sprite are unaffected by this bit. 0 = 8-bit Palette Mode 1 = 10-bit Gamma Mode

Bit	Descriptions
23:21	**Interlaced Mode** These bits are used for software control of interlaced behavior. They are updated immediately if the pipe is off, or in the vertical blank after programming if pipe is enabled. 0xx = Progressive 100 = Interlaced embedded panel using programmable vertical sync shift. 101 = Interlaced using vertical sync shift. Backup option to 110 setting.110 = Interlaced with VSYNC/HSYNC Field Indication using legacy vertical sync shift. Used for SDVO. 111 = Interlaced with Field 0 Only using legacy vertical sync shift. Not used Note: VGA display modes, sDVO line stall, and Panel fitting do not work while in interlaced modes Setting the Interlaced embedded panel mode causes hardware to automatically modify the output to match the specifications of panels that support interlaced mode.
20	Reserved
19	**Display / Overlay Planes Off:** This bit when set will cause all enabled Display and overlay planes that are assigned to this pipe to be disabled by overriding the current setting of the plane enable bit, at the next VBLANK. Timing signals continue as they were but the screen becomes blank. Setting the bit back to a zero will then allow the display and overlay planes to resume on the following VBLANK. 0 = Normal Operation 1 = Planes assigned to this pipe are disabled.
18	**Cursor Planes Off:** This bit when set will cause all enabled cursor planes that are assigned to this pipe to be disabled by overriding the current setting of the plane enable bit, at the next VBLANK. Timing signals continue as they were but the cursor(s) no longer appear on the screen. Setting the bit back to a zero will then allow the cursor planes to resume on the following VBLANK. 0 = Normal Operation 1 = Planes assigned to this pipe are disabled.
17:16	**Refresh Rate CxSR Mode Association** These bits select how refresh rates are tied to big FIFO mode on pipe A. When they are set to anything other than 00, bits 23:21 of this register must be programmed to 0xx. Switching between 01 and 10 settings directly is not allowed. Software must program this field to 00 before switching. Software is responsible for enabling this mode only for integrated dispay panels that support corresponding mode. 00 – **Default** – no dynamic refresh rate change enabled. Software control only. 01 – Progressive-to-progressive refresh rate change enabled and tied to big FIFO mode. Pixel clock values set in FPA0/FPA1 settings in the DPLLA control register and FPA0/FPA1 divider registers. FPA0 is tied to non-big-FIFO mode 10 – Progressive-to-interlaced refresh rate change enabled and tied to big FIFO mode. Pixel clock value does not change in this case. Scaling must be disabled in this mode. Uses programmable VS shift 11 – Reserved
15:10	Reserved: Write as zero

Bit	Descriptions
9:8	**Reserved: MBZ**
7:5	Bits Per Color [DevCTG]: This field selects the number of bits per color sent to a receiver device connected to this port. Color format takes place on the Vblank after being written. Color format change can be done independent of a pixel clock change in DisplayPort. Selecting a pixel color depth higher or lower than the pixel color depth of the frame buffer results in dithering the output stream. For further details on Display Port fixed frequency programming to accommodate these formats refer to “DP Frequency Programming” in DPLL section of Bspec. 000 = 8 bits per color (Default) 001 = 10 bits per color 010 = 6 bits per color 011 = RESERVED **1xx = RESERVED**
4	Dithering enable [DevCTG]: This bit enables dithering for DisplayPort 6bpc or 8bpc modes 0 – Dithering disabled (Default) **1 – Dithering enabled**
3:2	Dithering type [DevCTG]: This bit selects dithering type for DisplayPort 6bpc or 8bpc modes 00 - Spatial only (default) 01- Spatio-Temporal 1 10- Spatio-Temporal 2 (testmode) **11- Temporal only (testmode)**
1	DDA reset [DevCTG]: 0 – Do not reset DDA **1 – Reset DDA every 8th display frame**
0	Reserved: Write as zero

2.10.1.4 PIPEAGCMAXRED—Pipe A Gamma Correction Max Red

Memory Offset Address: 70010h
Default: 00010000h
Normal Access: Read/Write

Bit	Descriptions
31:17	Reserved
16:0	**Max Red Gamma Correction Point:** 129th reference point for red channel of the pipe piecewise linear gamma correction. The value should always be programmed to be less than or equal to 1024.0. Format: 11.6 Default: 0x10000

2.10.1.5 PIPEAGCMAXGREEN—Pipe A Gamma Correction Max Green

Memory Offset Address: 70014h
Default: 00010000h
Normal Access: Read/Write

Bit	Descriptions
31:17	Reserved
16:0	**Max Green Gamma Correction Point:** 129th reference point for green channel of the pipe piecewise linear gamma correction. The value should always be programmed to be less than or equal to 1024.0. Format: 11.6 Default: 0x10000

2.10.1.6 PIPEAGCMAXBLUE—Pipe A Gamma Correction Max Blue

Memory Offset Address: 70018h
Default: 00010000h
Normal Access: Read/Write

Bit	Descriptions
31:17	Reserved
16:0	**Max Blue Gamma Correction Point:** 129th reference point for blue channel of the pipe piecewise linear gamma correction. The value should always be programmed to be less than or equal to 1024.0. Format: 11.6 Default: 0x10000

2.10.1.7 PIPEASTAT—Pipe A Display Status

Memory Offset Address:	70024h
Default:	00000000h
Normal Access:	Read/Write

This register is the second level of a two level interrupt and status scheme. A single bit in the first line interrupt status register represents the state of this register which is equal to the AND of a status bits with their corresponding enable bits OR'ed together. First line interrupt status bits can cause interrupts or writes of the status register to cacheable memory. Bits in this register indicate the status of the display pipe A and can cause interrupt status bit changes in the first level interrupt and status register. Status bits in this register as 'sticky" and once they are set will be cleared by writing a one to that bit. A write of a zero will not have an effect on the corresponding Interrupt status bit. The corresponding enable bits will determine if the interrupt status bit should be used in the first line interrupt status register. When an interrupt occurs, the first line interrupt register indicates the second line source of the interrupt. Reading the second line register will determine the precise source for the interrupt.

Programming:

1. Prior to clearing a Display Pipe-sourced interrupt (e.g., Display Pipe A VBLANK) in the IIR, the corresponding interrupt (source) status in the PIPEASTAT or PIPEBSTAT register (e.g., Pipe A VBLANK Interrupt Status bit of PIPEASTAT) must first be cleared. Note that clearing these status bits requires writing a '1' to the appropriate bit position.

Bit	Descriptions
31	**FIFO A Under-run Status:** Set when a pipe A FIFO under-run occurs, cleared by a write of a 1. An underrun has occurred on an attempt to pop an empty FIFO. This does not feed into the first line interrupt status register. This will occur naturally during mode changes, to be useful, it should be cleared after a mode change has occurred. This bit is only valid after Pipe A has been completely configured. 1 = FIFO A Underflow occurred 0 = FIFO A Underflow did not occur
30	Reserved**:**
29	**CRC Error Enable:** This will enable the consideration of the CRC error status bit in the first line interrupt/status logic. 0 = CRC Error Detect Disabled 1 = CRC Error Detect Enabled
28	**CRC Done Enable:** This will enable the consideration of the CRC error status bit in the first line interrupt/status logic. 0 = CRC Done Detect Disabled 1 = CRC Done Detect Enabled
27	**GMBUS Event Enable:** This will enable the use of the GMBUS interrupt status bit in the first line interrupt/status logic. 0 = No GMBUS event enabled 1 = GMBUS event enabled
26	Reserved: Write as zero

Bit	Descriptions
25	**Vertical Sync Interrupt Enable:** This will enable the consideration of the vertical sync interrupt status bit in the first line interrupt logic. 0 = Vertical Sync Interrupt/Status Disabled 1 = Vertical Sync Interrupt/Status Enabled
24	**Display Line Compare Enable:** This will enable the consideration of the line compare interrupt status bit in the first line interrupt/status logic. 0 = Display Line Compare Interrupt/Status Disabled 1 = Display Line Compare Interrupt/Status Enabled
23	**Reserved.**
22	**[DevCL]: Legacy BLC Event Enable:** This will enable writes to the PCI Backlight Control Register to cause and the display A event status to be set and an Interrupt if Display A Event interrupt is enabled. 0 = No BLC Event enabled 1 = BLC Event enabled
21	**Odd Field Interrupt Event Enable:** This bit should only be used when this pipe is in an interlaced display timing. 0 = Odd Field Event disable 1 = Odd Field Event enable
20	**Even Field Interrupt Event Enable:** This bit should only be used when this pipe is in an interlaced display timing. 0 = Even field Event disable 1 = Even field Event enable
19	Reserved: Write as zero.
18	**Start of Vertical Blank Interrupt Enable:** This will enable the consideration of the start of vertical blank interrupt status bit in the first line interrupt/status logic. 0 = Start of Vertical Blank Interrupt/Status Disabled 1 = Start of Vertical Blank Interrupt/Status Enabled
17	**Vertical Blank Interrupt Enable:** This will enable the consideration of the vertical blank interrupt status bit in the first line interrupt/status logic. 0 = Vertical Blank Interrupt/Status Disabled 1 = Vertical Blank Interrupt/Status Enabled
16	**Overlay Registers Updated Enable:** 0 = Overlay Registers have been updated during Vertical Blank Status Disabled 1 = Overlay Registers have been updated during Vertical Blank Status Enabled
15	Reserved: MBZ
14	Reserved

Bit	Descriptions
13	**CRC Error Interrupt Status:** This sticky status bit is set when a Pipe A CRC error is detected. It is cleared by a write of a one. For this bit to be meaningful, the pipe and pixel clock should be enabled and running. 0 = No CRC error has occurred 1 = CRC Error Detected
12	**CRC Done Interrupt Status:** This sticky status bit is set when Pipe A CRC calculation and compare are complete. It is cleared by a write of a one. For this bit to be meaningful, the pipe and pixel clock should be enabled and running. 0 = CRC Not Done 1 = CRC Done
11	**GMBUS Interrupt Status:** This status bit will be set on a GMBUS event. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. 0 = GMBUS event has not occurred 1 = GMBUS event has occurred
10	Reserved
9	**Vertical Sync Interrupt Status:** This bit provides a sticky status that is set when a pipe A vertical sync occurs, cleared by a write of a 1. For interlaced timing modes, this occurs once per field, when in progressive, it occurs once per frame. For this bit to be meaningful, the pipe and pixel clock should be enabled and running. 0 = Vertical Sync has not occurred 1 = Vertical Sync has occurred
8	**Display Line Compare Interrupt Status:** Set when a pipe A compare match occurs, cleared by a write of a 1. 0 = Display Line Compare has not been satisfied 1 = Display Line Compare has been satisfied
7	**Reserved.**
6	**[DevCL]: Legacy BLC Event Status:** This status bit indicates that a write to the PCI Backlight Control Register (LBPC) has occurred. Software must clear this bit in order to detect subsequent writes, for example while servicing the Event Interrupt. This bit is cleared when a write to this register occurs with this bit as a one. Writes with this bit as a zero has no effect on the value of the bit. 0 = No BLC write detected 1 = A BLC write was detected
5	**Odd Field Interrupt Status:** This status bit will be set on a Odd field VBLANK event. This bit should only be used when this pipe is in an interlaced display timing. For synchronization with register updates, the actual event will occur one line after the start of VBLANK. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. Note: This bit will not be set when pipe is in “Interlaced with Field 0 Only using legacy vertical sync shift” mode. 0 = Odd Field Vertical Blank has not occurred 1 = Odd Field Vertical Blank has occurred

Bit	Descriptions
4	**Even Field Interrupt Status:** This status bit will be set on a even field VBLANK event. This bit should only be used when this pipe is in an interlaced display timing. For synchronization with register updates, the actual event will occur one line after the start of VBLANK. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. Note: This bit will not be set when pipe is in "Interlaced with Field 0 Only using legacy vertical sync shift" mode. 0 = Even Field Vertical Blank has not occurred 1 = Even Field Vertical Blank has occurred
3	Reserved
2	**Start of Vertical Blank Interrupt Status:** This status bit will be set at the beginning of a VBLANK event. At this point, the double buffered display registers flip, taking their new values. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. 0 = Start of Vertical Blank has not occurred 1 = Start of Vertical Blank has occurred
1	**Vertical Blank Interrupt Status:** This status bit will be set on a VBLANK event, when the frame start occurs. The display registers are updated at the start of vertical blank, but the new register data is not utilized by the display pipeline until the point in the vertical blank period when the frame start occurs, which is the event that triggers this bit. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. 0 = Vertical Blank has not occurred 1 = Vertical Blank has occurred
0	[DevBW] and [DevCL] Overlay Registers Updated Interrupt Status: This is not a pipe A event. It exists in this register for compatibility reasons only. The bit is set when an overlay register update completes, cleared by a write of a 1. 0 = Overlay Registers update has not occurred 1 = Overlay Registers update has occurred.

2.10.1.8 DSPARB—Display Arbitration Control

Memory Offset Address: 70030h
Default: 00001D9Ch (FIFO Sizes A=28, B=31, C=37)
Normal Access: Read/Write

Notes:

Each active display plane A, B, or C requires a FIFO to cover for memory latency. The FIFOs all come from a single RAM that is divided into areas for each display plane. The amount of the RAM used by each display plane is defined by this register. The two fields in the register split the display RAM into three portions, allocated between display planes A, B, and C. This register is double buffered and updated on the leading edge of Vertical Blank of the pipe that the planes are assigned to. This register should only be changed when a single pipe is enabled or if all of the Display A, B, C planes are disabled. It takes effect on the next VBLANK for whichever pipe is currently active. Each display plane needs a minimum FIFO size that is at least MaxLatencyForPlane * PixelRate * PixelSize + 512. All values should be rounded up to the next unit of 64B.

A special C3 mode can occur when a single display (of Display A and Display B) is active and the overlay and Display C are disabled. In that mode, when C3 is entered, the values in the BSTART and CSTART fields are ignored and the entire RAM is allocated to the single active display plane.

Notes: [DevBW] The control granularity of FIFO size is 64-bytes and the total size of the RAM is 384*16 bytes making TOTALSIZE equal to 96. The range of values for CSTART and BSTART is 0-95.

Notes: [DevCL and DevCTG] The control granularity of FIFO size is 64-bytes and the total size of the RAM is 512*16 bytes making TOTALSIZE equal to 128. The range of values for CSTART and BSTART is 0-127.

Notes: [DevCTG] The entire register is reserved. Hardware controls the FIFO sizing automatically.

The display dot clock frequency or pixel rate must not exceed 90% of the core display clock. When a primary plane is enabled with 64bpp format and sprite is also enabled on the same pipe, the dot clock frequency or pixel rate must be less than 80% of the core display clock.

Bit	Descriptions
31:14	Reserved: Write as zero.
13:7	**CSTART:** This field selects the end of the ram used for display B and the start of the RAM for display C. If display B is unused, this field can be set to the same value as BSTART. If display C is unused, this field can be set to TOTALSIZE-1. It must be programmed to a number greater than or equal to the value in BSTART and less than the total size of the RAM (TOTALSIZE). The size of the display B FIFO will be (CSTART-BSTART)*64. The size of the display C FIFO will be (TOTALSIZE-CSTART-1) *64 bytes.
6:0	**BSTART:** This field selects the end of the ram used for display A and the start of the RAM for display B. If display A is unused, this field can be set to zero. The value should never exceed the size of the RAM (TOTALSIZE). The size of the display A FIFO will be (BSTART)*64 bytes.

2.10.1.9 FW1—Display FIFO Watermark Control 1

Memory Offset Address:	70034h
Default:	(3F8F0F0Fh for [DevCL]) (1F8F0F0Fh for [DevBW])
Normal Access:	Read/Write

These control values only apply to high-resolution (non-VGA) modes of operation. The hardware depends on these registers being set properly since it is possible to set the watermarks to states causing starvation of the sync FIFO.

Bit	Description
31:23	**Display FIFO Self Refresh Watermark**. This register defines the value of the watermark used by the Display streamer in case the CPU is in C2/C3/C4 and the memory has entered self refresh. Number in 64Bs of space in FIFO above which the Display Stream will generate requests to Memory. **Programming Note [DevCL]:** When calculating watermark values for 15/16bpp display formats, assume 32bpp for purposes of calculation using the high priority bandwidth analysis spreadsheet.
22	**Reserved**: MBZ
21:16	**Cursor B FIFO Watermark**. Number in 64Bs of space in the Cursor B FIFO above which the Cursor B Stream will generate requests to Memory. Always program to 8.
15	Reserved: MBZ
14:8	**Display Plane B FIFO Watermark:** Number in 64Bs of space in FIFO above which the Display B Stream will generate requests to Memory. Always program to 8.
7	Reserved: MBZ
6:0	**Display Plane A FIFO Watermark:** Number in 64Bs of space in FIFO above which the Display A Stream will generate requests to Memory. Always program to 8.

2.10.1.10 FW2—Display FIFO Watermark Control 2

Memory Offset Address:	70038h
Default:	00 00 0F 0Fh
Normal Access:	Read/Write

These control values only apply to high-resolution (non-VGA) modes of operation. The hardware depends on these registers being set properly since it is possible to set the watermarks to states causing starvation of the sync FIFO.

Bit	Description
31	**[DevCTG] FBC SR Watermark Enable: Enables the FBC watermarks to be used in the fetch calculation** **0: disabled** **1: enabled**
30:28	**[DevCTG] FBC SR Watermark: Number of equivalent lines of the primary display SR watermark**
27:24	**[DevCTG] FBC SR HPLL Watermark: Number of equivalent lines of the primary display SR HPLL watermark.**
23	**Reserved: MBZ**
22:16	**[DevBLKC] and [DevCTG] Display Plane Sprite B FIFO Watermark Number in 64Bs of space in FIFO above which the Display Sprite B Stream will generate requests to Memory (Value should be as recommended in the high priority bandwidth analysis spreadsheet).**
15:14	**Reserved: MBZ**
13:8	Cursor A FIFO Watermark. Number in 64Bs of space in the Cursor A FIFO above which the Cursor A Stream will generate requests to Memory (Value should be as recommended in the high priority bandwidth analysis spreadsheet). **[DevBW] and [DevCL] – Always program to 8.**
7	**Reserved: MBZ**

Bit	Description
6:0	[DevBLKC] and [DevCTG] Display Plane Sprite A FIFO Watermark. Number in 64Bs of space in FIFO above which the Display Sprite A Stream will generate requests to Memory [DevBW] and [DevCL] Display Plane C FIFO Watermark. Number in 64Bs of space in FIFO above which the Display C Stream will generate requests to Memory (Value should be as recommended in the high priority bandwidth analysis spreadsheet). **[DevBW] and [DevCL] – Always program to 8.**

2.10.1.11 PipeAFrameHigh— Pipe A Frame Count High [DevCL]

Memory Offset Address: 70040h
Default: 00000000h
Normal Access: Read-Only (Requires that this pipe's PLL is running)

Bit	Descriptions
31:16	Reserved:
15:0	**Pipe A Frame Count High**. This field provides the most significant 16-bits of the free running frame counter for this display pipe. When combined with the low 8-bits of the frame counter, it allows the frame count to wrap only once per 16M frames. The counter is reset to zero when the device is reset or when the pipe transitions from off to on. It is incremented when the low frame counter rolls over to zero. This counter wraps, when the maximum count is reached, the next count value will be zero. When combined with the least significant bits of the frame counter forms a 24-bit value that indicates the number of frames since the pipe was enabled. **This register should only be read if the display PLL for this display pipe is running. The hardware does not attempt to synchronize this value with the read of the least significant bits. Software must take the appropriate actions when it is desired to form a full frame count value by combining the two portions of the frame counter.** The following example is a possible method: Read the Frame count high → High1 Read the Frame count low → Low 1, Pixel1 Read the Frame count high → High2 Read the Frame count low → Low 2, Pixel2 Read the Frame count high → High3 If (Both versions of the frame count high are equal (High1 = High2)) then: Frame# ← High1 | Low1 Line# ← int(Pixel1/Htotal) Pixel# ← Pixel1 – (Line#*Htotal) ElseIf (High2 = High3) then: Frame# ← High2 | Low2 Line# ← int(Pixel2/Htotal) Pixel# ← Pixel2 – (Line#*Htotal) Else: Error, Unable to acquire frame number – Indicates that the above register read sequence takes more than 256 display frames. EndIf

2.10.1.12 PIPEA_FRMCOUNT—Pipe A Frame Counter [DevCTG, DevELK]

PIPEA_FRMCOUNT—Pipe A Frame Counter	
Register Type:	MMIO
Address Offset:	70040h
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32

Bit	Description
31:0	**Pipe_Frame_Counter** Project: All Format: Provides read back of the display pipe frame counter. This counter increments on every start of vertical blank and rolls over back to 0 after 2^32 frames.

2.10.1.13 PipeAFramePixel— Pipe A Frame Count Low and Pixel Count [DevCL]

Memory Offset Address:	70044h
Default:	00000000h
Normal Access:	Read-Only (Requires that this pipe's PLL be running)

Bit	Descriptions
31:24	**Pipe A Frame Count Low**. This field provides the least significant 8-bits of the frame counter for this display pipe. It is reset when the device is reset or when the pipe transitions from off to on. It is incremented at the beginning of horizontal active of the first line of vertical active. This counter wraps, when the maximum count is reached, the next count value will be zero.
23:0	**Pipe A Pixel Count**. This field provides a pixel counter for the display pipe. It is reset at the beginning of horizontal active of the first line of vertical active and is incremented every pipe pixel clock. It will increment to a maximum of the number of clocks in a frame minus one (Htotal*Vtotal-1). To determine the line that the pixel count represents, the value is divided by the horizontal total for this pipe. The pixel count and the frame count low operate together, if a display frame has for example 64,000 clocks and we are on frame 5, this register value would read Frame=5 Pixel=63,999 on the last pixel of frame five, one display clock later it would read Frame=6 Pixel = 0. Pixel Count tracks pixels at the output. Even in the most basic cases, the actual fetch of the source data for that pixel may occur a significant amount of time before the pixel makes it to the output. In addition, there are cases where there is a difference between pixels in the source data and the output. These cases include: - Panel fit images - Integrated TV using overscan scaler - Modes that use line stall (i.e. External LVDS/TMDS scalers) - Display Plane 2X modes - Various methods of display rotation **This field is only to be used for display modes with 16M or less clocks per frame. This is not to be used in VGA display operation.**

2.10.1.14 PIPEA_FLIPCOUNT—Pipe A Flip Counter [DevCTG, DevELK]

PIPEA_FLIPCOUNT—Pipe A Flip Counter	
Register Type: MMIO **Address Offset:** 70044h **Project:** All **Default Value:** 00000000h **Access:** Read Only **Size (in bits):** 32	
Bit	**Description**
31:0	**Pipe_Flip_Counter** Project: All Format: Provides read back of the display pipe flip counter. This counter increments on each flip of the surface of the primary plane on this pipe. This includes command streamer asynchronous and synchronous flips and any MMIO writes to the primary plane surface address. It rolls over back to 0 after 2^32 flips.

2.10.1.15 PIPEA_FRMTIMESTAMP—Pipe A Frame Time Stamp [DevCTG, DevELK]

PIPEA_FRMTIMESTAMP—Pipe A Frame Time Stamp	
Register Type: MMIO **Address Offset:** 70048h **Project:** All **Default Value:** 00000000h **Access:** Read Only **Size (in bits):** 32	
Bit	**Description**
31:0	**Pipe_Frame_Time_Stamp** Project: All Format: Provides read back of the display pipe frame time stamp. The time stamp value is sampled at every start of vertical blank. The TIMESTAMP register has information on the time stamp value.

2.10.1.16 PIPEA_FLIPTIMESTAMP—Pipe A Flip Time Stamp [DevCTG, DevELK]

PIPEA_FLIPTIMESTAMP—Pipe A Flip Time Stamp	
Register Type:	MMIO
Address Offset:	7004Ch
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32

Bit	Description
31:0	**Pipe_Flip_Time_Stamp** Project: All Format: Provides read back of the display pipe flip time stamp. The time stamp value is sampled on each flip of the surface of the primary plane on this pipe. This includes command streamer asynchronous and synchronous flips and any MMIO writes to the primary plane surface address. The TIMESTAMP register has information on the time stamp value.

2.10.1.17 PipeAGMCHDataM— Pipe A Data M value [DevCTG]

Memory Offset Address:	70050h
Default:	00000000h
Normal Access:	Read/Write

Bit	Descriptions
31	Reserved. Must be zero
30:25	TU Size. This field is the size of the transfer unit for DP, minus one. Default value to program = 111111 (TU size of 64)
24	Reserved. Must be zero
23:0	Pipe A Data M value. This field is the m value for internal use of the DDA. Calculation of this value is as follows: Data m/n = dot clock * bytes per pixel / ls_clk * # of lanes Please note that in the DisplayPort specification, dot clock is referred to as strm_clk

2.10.1.18 PipeAGMCHDataN— Pipe A Data N value [DevCTG]

Memory Offset Address: 70054h
Default: 00000000h
Normal Access: Read/Write

Bit	Descriptions
31:24	Reserved. Must be zero
23:0	Pipe A Data N value. This field is the m value for internal use of the DDA. Calculation of this value is as follows: Data m/n = dot clock * bytes per pixel / ls_clk * # of lanes Please note that in the DisplayPort specification, dot clock is referred to as strm_clk

2.10.1.19 PipeADPLinkM— Pipe A Link M value [DevCTG]

Memory Offset Address: 70060h
Default: 00000000h
Normal Access: Read/Write

Bit	Descriptions
31:24	Reserved. Must be zero
23:0	Pipe A Link M value. This field is the m value for external transmission in the Main Stream Attributes. Calculation of this value is as follows: Link m/n = pixel clk / ls_clk Please note that in the DisplayPort specification, pixel clk is referred to as strm_clk

2.10.1.20 PipeADPLinkN— Pipe A Link N value [DevCTG]

Memory Offset Address: 70064h
Default: 00000000h
Normal Access: Read/Write

Bit	Descriptions
31:24	Reserved. Must be zero
23:0	Pipe A Data N value. This field is the m value for external transmission in the Main Stream Attributes and VB-ID. Calculation of this value is as follows (to be filled in): Link m/n = pixel clk / ls_clk Please note that in the DisplayPort specification, pixel clk is referred to as strm_clk

2.10.2 Cursor A Plane Control Registers

The hardware cursor registers are memory mapped and accessible through 32 bit, 16 bit, or 8-bit accesses. They are all double-buffered, including the palette registers. Writes to cursor registers are performed to a holding register. The actual register update will occur based on the associated pipe's Vertical Blank signal only after a write cycle to the base address register (setting the trigger) has occurred. Writes to any register other than the trigger register will disable an active trigger if that occurs before the vertical blank.

2.10.2.1 CURACNTR—Cursor A Control Register

This register, and all other cursor registers will remain in their holding register (readable) after a write. The holding registers are transferred into the active registers on the asserting edge of Vertical Blank only after a write cycle to the base address register has completed.

Memory Offset Address:	70080h
Default:	00000000h
Normal Access:	Read/Write

Bit	Description
31:30	Reserved: Write as zero.
29:28	**Pip Select:** A state machine handles the synchronization of the switch to both vertical blank signals. So as far as the software is concerned, when both display pipes are being used, it can be switched at any time; the hardware will synchronize the switch. [DevBW] and [DevCL] Pipe Select: A state machine handles the synchronization of the switch to both vertical blank signals. So as far as the software is concerned, when both display pipes are being used, it can be switched at any time; the hardware will synchronize the switch. 00 = HW cursor is attached to Display Pipe A. This is the default after reset. 01 = HW cursor is attached to Display Pipe B. 10 = Reserved for pipe C 11 = Reserved for pipe D [DevCTG] Reserved: Write as zero.
27	**Popup Cursor Enabled**. This bit should be turned on when using Cursor A as a popup cursor. When in popup mode, hardware interprets the cursor base address as a physical address instead of a graphics address. 0 = Cursor A is hi-res 1 = Cursor A is popup
26	**Cursor Gamma Enable:** This bit only has an effect when using the cursor in a non-VGA mode. In VGA pop-up operation, the cursor data will always bypass the gamma (palette) unit. 0 = Cursor pixel data bypasses gamma correction or palette (default). 1 = Cursor pixel data is gamma to be corrected in the pipe.
25:16	Reserved: Write as zero
15	**180° Rotation:** This mode causes the cursor to be rotated 180°. In addition to setting this bit, software must also set the base address to the lower right corner of the unrotated image. Only 32 bits per pixel cursors can be rotated. This field must be zero when the cursor format is 2 bits per pixel. 0 = No rotation 1 = 180° Rotation of 32-bit per pixel cursors

Bit	Description
14:6	Reserved
5	**Cursor Mode Select:** See following table.
4:3	Reserved
2:0	**Cursor Mode Select:** These three bits together with bit 5 select the mode for cursor as shown in the following table.

Table 2-6. Cursor Mode Select

Bit 5	Bits 2:0	Mode
0	000	Cursor is disabled. This is the default after reset. When the cursor register value changes from enabled to disabled, the cursor will stop fetching data at the following VBLANK event. The cursor enable can be overridden by the pipe cursor disable bit. The value of these bits do not change when disabled by the pipe cursor disable bit.
0	001	Reserved
0	010	128x128 32bpp XRGB (not available for VGA use)
0	011	256x256 32bpp XRGB (not available for VGA use)
0	100	64 x 64 2bpp 3-color and transparency mode
0	101	64 x 64 2bpp AND/XOR 2-plane mode
0	110	64 x 64 2bpp 4-color mode
0	111	64 x 64 32bpp AND/XOR (not available for VGA use) For each pixel: Least Significant Three Bytes Provides cursor RGB 888 color information Most Significant Byte: All Ones: Opaque show the cursor color All Zeros: Transparent (color must also equal zero) - Other: Invert the underlying display pixel data (ignore the color)
1	000	Reserved
1	001	Reserved
1	010	128x128 32bpp ARGB (not available for VGA use)
1	011	256x256 32bpp ARGB (not available for VGA use)
1	100	Reserved
1	101	Reserved
1	110	Reserved
1	111	64 x 64 32bpp ARGB (not available for VGA use)

2.10.2.2 CURABASE—Cursor A Base Address Register

Memory Offset Address: 70084h
Default: 00000000h
Normal Access: Read/Write

This register specifies the graphics memory address at which the cursor image data is located. Writes to this register acts like a trigger that enables atomic updates of the cursor registers. When updating the cursor registers, this register should be written last in the sequence. This register should be written even if the actual contents did not change to allow the holding registers to move to the active registers on the next VBLANK.

For legacy cursor modes, this register is sufficient to specify the address of the entire cursor. For ARGB modes, this register specifies the address of the first page of the cursor data.

Bit	Description
31:4	**Cursor Base Address**. This field specifies bits 31:4 of the graphics address of the base of the cursor. If the cursor is a popup, this field specifies bits 31:4 of the physical address of the base of the cursor, and bits 35:32 of the address are specified in the LSBs of this register. Popup cursor mode is selected within the CURACNTR register. The cursor surface address must be 4K byte aligned. The cursor must be in linear memory, it cannot be tiled. When performing 180° rotation, this offset must be the difference between the last pixel of the last line of the cursor data in its unrotated orientation and the cursor surface address. A write to this register also acts as a trigger event to force the update of active registers from the staging registers on the next display event. Each cursor register is double-buffered. The CPU writes to a set of holding registers. The active registers are updated from the holding registers following the leading edge of the vertical blank pulse. The update is postponed until the next vblank if a write cycle is active to any of the cursor registers at the time of the vblank. The update is also postponed if a write sequence is in progress. It is assumed that if the cursor mode is changed, the cursor image will also be changed. To prevent the cursor from appearing when it is only partially programmed, the active registers will not be updated until both the cursor control and base address registers have been programmed. If the cursor control register is written, the cursor base address must also be written before the change will be effective. However, the base address register may be changed (e.g., to change the shape of the cursor) without also writing to the control register. If both are to be written, the control register must be written first.
3:0	**Popup Cursor Base Address MSBs**. This field specifies bits 35:32 of the popup cursor physical address. If popup mode is not selected, this field is ignored.

2.10.2.3 CURAPOS—Cursor A Position Register

Memory Offset Address: 70088h
Default: 00000000h
Normal Access: Read/Write

This register specifies the screen position of the cursor. The origin of the cursor position is always the upper left corner of the active image for the display pipe that the cursor is assigned. This register can be loaded atomically (requires that the base address be written) and is double buffered.

Bit	Description
31	**Cursor Y-Position Sign Bit:** This bit provides the sign bit of a signed 13-bit value that specifies the horizontal position of cursor. (Default is 0.) For normal high resolution display modes, the cursor must have at least a single pixel positioned over the active screen. For use as a VGA Popup, the entire cursor must be positioned over the active area of the VGA image.
30:28	Reserved: Write as zero.
27:16	**Cursor Y-Position Magnitude Bits 11:0:** This register provides the magnitude bits of a signed 12-bit value that specifies the vertical position of cursor. The sign bit of this value is provided by bit 31of this register. (Default is 0.) For use as a VGA Popup, the entire cursor must be positioned over the active area of the VGA image. Enabling the border in VGA (VGA Border Enable bit in the VGA Config register) includes the border in what is considered the “active area”. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation.
15	**Cursor X-Position Sign Bit:** This bit provides the sign bit of a signed 13-bit value that specifies the horizontal position of cursor. (Default is 0.) For normal high resolution display modes, the cursor must have at least a single pixel positioned over the active screen. For use as a VGA Popup, the entire cursor must be positioned over the active area of the VGA image. Enabling the border in VGA (VGA Border Enable bit in the VGA Config register) includes the border in what is considered the “active area”.
14:12	Reserved: Write as zero.
11:0	**Cursor X-Position Magnitude Bits 11:0:** These 12 bits provide the signed 13-bit value that specifies the horizontal position of cursor. The sign bit is provided by bit 15 of this register. (Default is 0.) When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the end of the active video area in the unrotated orientation.

2.10.2.4 CURARESV—Cursor A (Reserved)

Memory Offset Address: 7008Ch
Default: 00000000h
Normal Access: Read-Only

2.10.2.5 CURAPALET[0:3]—Cursor A Palette registers (4 Registers)

Memory Offset Address: 70090–7009Fh
CURAPALET0: 70090–70093h
CURAPALET1: 70094–70097h
CURAPALET2: 70098–7009Bh
CURAPALET3: 7009C–7009Fh
Default: 00000000h
Normal Access: Read/Write

These palette registers can be accessed through this MMIO interface register locations combined with an enable bit. This is the preferred method. The cursor palette provides color information when using one of the indexed modes. The two-bit index selects one of the four colors or two of the colors when in the AND/XOR cursor mode. The cursor palette provides color information when using one of the indexed modes. The two-bit index selects one of the four colors or two of the colors when in the AND/XOR cursor mode.

The table below describes the palette usage for different cursor modes and indexes.

Index	2 color	3color	4color
00	palette 0	palette 0	palette 0
01	palette 1	palette 1	palette 1
10	transparent	transparent	palette 2
11	invert destination	palette 3	palette 3

Bit	Descriptions
31:24	Reserved: Write as zero.
23:16	**Red or Y Value:** These registers specify the cursor palette. RGB data is full range unsigned numbers. YUV data will be unsigned for the Y and excess 128 notation for the UV values. The data can be pre-gamma corrected and bypass the gamma correction logic or passed through the gamma corrector.
15:8	**Green or U Value:**
7:0	**Blue or V Value:**

2.10.2.6 CURASURFLIVE—Cursor A Live Surface Base Address Register [DevCTG-B]

Memory Offset Address:	700ACh
Default:	00000000h
Normal Access:	Read Only

Bit	Descriptions
31:0	Cursor A Live Surface Base Address. This gives the live value of the surface base address as being currently used for the plane.

2.10.3 Cursor B Plane Control Registers

2.10.3.1 CURBCNTR—Cursor B Control Register

Memory Offset Address:	700C0h
Default:	00000000h
Normal Access:	Read/Write

The hardware cursor registers are memory mapped and accessible through 32 bit, 16 bit, or 8-bit accesses. They are all, including the palette registers double buffered. Writes to cursor registers are done to a holding register. The actual register update will occur based on the assigned pipes VBLANK. It is recommended that the base register be accessed through a 32-bit write only. To update all cursor registers atomically, a sequence that ends with a base address register write should be used.

Bit	Descriptions
31:30	Reserved: Write as zero.
29:28	**Pipe Select:** A state machine handles the synchronization of the switch to both vertical blank signals. So as far as the software is concerned, when both display pipes are being used, it can be switched at any time; the hardware will synchronize the switch. 00 = HW cursor is attached to Display Pipe A. This is the default after reset. 01 = HW cursor is attached to Display Pipe B. 10 = Reserved for to Display Pipe C. 11 = Reserved for to Display Pipe D.
27	Reserved: Write as zero.
26	**Cursor Gamma Enable:** 0 = Cursor pixel data bypasses gamma correction (default). 1 = Cursor pixel data is gamma to be corrected.
25:16	Reserved
15	**180° Rotation:** This mode causes the cursor to be rotated 180°. In addition to setting this bit, software must also set the base address to the lower right corner of the unrotated image. Only 32 bits per pixel cursors can be rotated. This field must be zero when the cursor format is 2 bits per pixel. 0 = No rotation 1 = 180° Rotation of 32-bit per pixel cursors
14:6	Reserved: Write as zero
5	**Cursor Mode Select:** See following table.
4:3	Reserved
2:0	**Cursor Mode Select:** These three bits together with bit 5 select the mode for cursor as shown in the following table.

Table 2-7. Cursor Mode Select

Bit 5	Bits 2:0	Mode
0	000	**Cursor is disabled:** This is the default after reset. When the cursor register value changes from enabled to disabled, the cursor will stop fetching data at the following VBLANK event. The cursor enable can be overridden by the pipe cursor disable bit. The value of these bits does not change when disabled by the pipe cursor disable bit.
0	001	Reserved
0	010	128x128 32bpp XRGB (not available for VGA use)
0	011	256x256 32bpp XRGB (not available for VGA use)
0	100	64 x 64 2bpp 3-color and transparency mode
0	101	64 x 64 2bpp AND/XOR 2-plane mode
0	110	64 x 64 2bpp 4-color mode
0	111	64 x 64 32bpp AND/XOR (not available for VGA use) For each cursor pixel: Least Significant Three Bytes Provides cursor RGB 888 color information Most Significant Byte: All Ones: Opaque show the cursor color All Zeros: Transparent (color must also equal zero) - Other: Invert the underlying display pixel data (ignore the color)
1	000	Reserved
1	001	Reserved
1	010	128x128 32bpp ARGB (not available for VGA use)
1	011	256x256 32bpp ARGB (not available for VGA use)
1	100	Reserved
1	101	Reserved
1	110	Reserved
1	111	64 x 64 32bpp ARGB

2.10.3.2 CURBBASE—Cursor B Base Address Register

Memory Offset Address: 700C4h
Default: 00000000h
Normal Access: Read/Write

This register specifies the memory address at which the cursor data is located. Writes to this register should be done with 32-bit accesses and acts as a trigger to atomically update the cursor register set. For legacy cursor modes, this register is sufficient to specify the address of the entire cursor. The address is the graphics address. For ARGB modes, this register specifies the address of the first page of the cursor data.

Bit	Description
31:0	**Cursor Base Address:** This register specifies the graphics address of the entire cursor. It also acts as a trigger event to force the update of active registers on the next display event. The cursor surface address must be 4K byte aligned. The cursor must be in linear memory, it cannot be tiled. When performing 180° rotation, this offset must be the difference between the last pixel of the last line of the cursor data in its unrotated orientation and the cursor surface address. A write to this register also acts as a trigger event to force the update of active registers from the staging registers on the next display event. Each cursor register is double-buffered. The CPU writes to a set of holding registers. The active registers are updated from the holding registers following the leading edge of the vertical blank pulse. The update is postponed until the next vblank if a write cycle is active to any of the cursor registers at the time of the vblank. The update is also postponed if a write sequence is in progress. It is assumed that if the cursor mode is changed, the cursor image will also be changed. To prevent the cursor from appearing when it is only partially programmed, the active registers will not be updated until both the cursor control and base address registers have been programmed. If the cursor control register is written, the cursor base address must also be written before the change will be effective. However, the base address register may be changed (e.g., to change the shape of the cursor) without also writing to the control register. If both are to be written, the control register must be written first.

2.10.3.3 CURBPOS—Cursor B Position Register

Memory Offset Address: 700C8h
Default: 00000000h
Normal Access: Read/Write

This register specifies the screen position of the cursor. The origin of the cursor position is always the upper left corner of the active image for the display pipe that the cursor is assigned. This register can be loaded atomically and is double buffered. The load register is transferred into the active register on the leading edge of Vertical Blank of the pipe cursor is currently assigned after the trigger has been set.

Bit	Descriptions
31	**Cursor Y-Position Sign Bit:** This bit provides the sign bit of a signed 13-bit value that specifies the horizontal position of cursor. (default is 0).). For normal high resolution display modes, the cursor must have at least a single pixel positioned over the active screen.
30:28	Reserved: Write as zero.
27:16	**Cursor Y-Position Magnitude Bits 11:0:** This register provides the magnitude bits of a signed 13-bit value that specifies the vertical position of cursor. The sign bit of this value is provided by bit 31 of this register. (default is 0) When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation.
15	**Cursor X-Position Sign Bit:** This bit provides the sign bit of a signed 13-bit value that specifies the horizontal position of cursor. (default is 0).). For normal high resolution display modes, the cursor must have at least a single pixel positioned over the active screen.
14:12	Reserved: Write as zero.
11:0	**Cursor X-Position Magnitude Bits 11:0:** These 12 bits provide the signed 13-bit value that specifies the horizontal position of cursor. The sign bit is provided by bit 15 of this register. (default is 0) When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation.

2.10.3.4 CURBRESV—Cursor B Reserved

Memory Offset Address: 700CCh
Default: 00000000h
Normal Access: Read

2.10.3.5 CURB PALET[0:3]—Cursor B Palette Rgisters (4 Registers)

Memory Offset Address: 700D0–700DCh
CURBPALET0: 700D0–700D3h
CURBPALET1: 700D4–700D7h
CURBPALET2: 700D8–700DBh
CURBPALET3: 700DC–700DFh
Default: 00000000h
Normal Access: Read/Write

These palette registers can be accessed through this MMIO interface or through a legacy mode using the VGA palette register locations combined with an enable bit. This is the preferred method. The cursor palette provides color information when using one of the indexed modes. In the two-bit AND/XOR cursor modes, the two-bit index selects one of the four colors or two of the colors when in the mode. RGB data is full range unsigned numbers. YUV data will be unsigned for the Y and excess 128 notation for the UV values. The data can be pre-gamma corrected and bypass the gamma correction logic or passed through the gamma corrector.

Bit	Descriptions
31:30	Reserved: Write as zero.
23:16	**Red or Y:**
15:8	**Green or U Value:**
7:0	**Blue or V Value:**

2.10.3.6 CURBSURFLIVE—Cursor B Live Surface Base Address Register [DevCTG-B]

Memory Offset Address:	700ECh
Default:	00000000h
Normal Access:	Read Only

Bit	Descriptions
31:0	Cursor B Live Surface Base Address. This gives the live value of the surface base address as being currently used for the plane.

2.10.4 Display Pipeline B

2.10.4.1 PIPEB_DSL—Display Scan Line

Memory Address Offset:	71000h
Default:	00h
Normal Access:	Read-Only

This register enables the read back of the display pipe vertical “line counter”. The display line value is from the display pipe B timing generator and is reset to zero at the beginning of a scan. The value increments at the leading edge of HSYNC and can be safely read any time. For normal operation, scan line zero is the first active line of the display. When in VGA centering mode, the scan line 0 is the 1st active scan line of the pseudo border not the centered active VGA image display area. In interlaced display timings, the scan line counter provides the current line in the field. One field will have a total number of lines that is one greater than the other field.

Programming Note: In order to cause the scan line logic to report the correct Line Counter value, the corresponding Display Pipeline timing registers must be programmed to valid, non-zero (e.g., 640x480 @ 60Hz) values before enabling the Pipe or programming VGA timing and enabling native VGA.

Bit	Description
31	[DevCTG] Current Field: Provides read back of the current field being displayed on display pipe B. Non-TV mode: 0 = first field (odd field) 1 = second field (even field) TV mode: 1 = first field (odd field) 0 = second field (even field) **[DevBW and DevCL] Reserved: Read only.**
30:13	Reserved: Read-only.
12: 0	**Pipe B Display Line Counter:** This register enables the read back of the display vertical “line counter”. The display line values are from the pipe B timing generator. They change at the leading edge of HSYNC, and can be safely read at any time.

2.10.4.2 PIPEB_SLC—Pipe B Display Scan Line Range Compare Register

Memory Address Offset: 71004h
Default: 00h
Normal Access: Read-Only

The Start and End Line Count Compare registers are compared with the display line values from the timing generator. They change at the leading edge of HSYNC. They can safely be accessed at any time. The End compare register operator is a less than or equal, while the Start compare register operator is a greater than or equal. The results of these 2 comparisons are communicated to the command stream controller for generating interrupts, status, and command stream flow control ("wait for within range" and "wait for not within range").

For range check, the value programmed should be the desired value - 1. So for line 0, the value programmed is VTOTAL and for line 1, the value programmed is 0.

This register can be written via the command stream processor using the MI_LOAD_SCAN_LINES_INCL or MI_LOAD_SCAN_LINES_EXCL commands.

Bit	Description
31	**Inclusive/Exclusive:** 1 = Inclusive Within Range, 0 = Exclusive Out of Range
30:29	Reserved: Read-only.
28:16	**End Scan Line Number:** This field specifies the ending scan line number of the Scan Line Window. Format = U16 in scan lines, where scan line 0 is the first line of the display frame. Range = [0, Display Buffer height in lines-1].
15:13	Reserved: Read-only.
12:0	[DevCTG] End Scan Line Number: This field specifies the ending scan line number of the Scan Line Window. Format = U16 in scan lines, where scan line 0 is the first line of the display frame. Range = [0, Display Buffer height in lines-1]. [DevBW] and [DevCL] Start Scan Line Number: This field specifies the starting scan line number of the Scan Line Window. Format = U16 in scan lines, where scan line 0 is the first line of the display frame. Range = [0,Display Buffer height in lines-1].

2.10.4.3 PIPEBCONF—Pipe B Configuration Register

Memory Offset Address: 71008h
Default: 00000000h
Normal Access: Read/Write double buffered

Bit	Descriptions
31	**Pipe B Enable:** Setting this bit to the value of one, turns on pipe B. This must be done before any planes are enabled on this pipe. Changing it to a zero should only be done when all planes that are assigned to this pipe have been disabled. Turning the pipe enable bit off disables the timing generator in this pipe. Plane disable occurs after the next VBLANK event after the plane is disabled. Synchronization pulses to the display are not maintained if the timing generator is disabled. Power consumption will be at its lowest state when disabled. A separate bit controls the DPLL enable for this pipe. Pipe timing registers should contain valid values before this bit is enabled. Disabling the Pipe and changing the timing registers and re-enabling the pipe before the next VBLANK will cause the mode change to occur at the end of the current frame. This requires no wait on the software's part. On the other hand, if this is the disabling of the pipe, that does require a software wait for VBLANK to occur. Synchronization pulses to the display are not maintained if the timing generator is disabled. Power consumption is at its lowest state. 1 = Enable 0 = Disable
30	**Pipe State:** This bit indicates the actual state of the pipe. Since there can be some delay between disabling the pipe and the pipe actually shutting off, this bit indicates the true current state of the pipe. 0 = Disabled 1 = Enabled
29	Reserved: Write as zero.
28:27	**Frame Start Delay:** Used to delay the frame start signal that is sent to the display planes. Normal operation uses the default 00 value and test modes can use the delayed frame start to shorten the test time. This would be set to 00 for normal operation. 00 = Frame Start occurs on the first HBLANK after the start of VBLANK 01 = Frame Start occurs on the second HBLANK after the start of VBLANK 10 = Frame Start occurs on the third HBLANK after the start of VBLANK 11 = Frame Start occurs on the forth HBLANK after the start of VBLANK
26	Reserved: Write as zero.
25	**FORCE_BORDER**: 0 = Normal Operation 1 = Color information is ignored and border color is substituted during active region
24	**Pipe B Gamma Unit Mode**. This bit selects which mode the pipe gamma correction logic works in. In the palette mode, it behaves as a 3X256x8 RAM lookup. VGA and indexed mode operation should use the palette in 8-bit mode. In the 10-bit gamma mode, it will act as a piecewise linear interpolation. Other gamma units such as in the overlay and sprite are unaffected by this bit. 0 = 8-bit Palette Mode 1 = 10-bit Gamma Mode

Bit	Descriptions
23:21	**Interlaced Mode** These bits are used for software control of interlaced behavior. They are updated immediately if the pipe is off, or in the vertical blank after programming if pipe is enabled. 0xx = Progressive 100 = Interlaced embedded panel using programmable vertical sync shift 101 = Interlaced using vertical sync shift. Backup option to setting 110. 110 = Interlaced with VSYNC/HSYNC Field Indication using legacy vertical sync shift. Used for SDVO. 111 = Interlaced with Field 0 Only using legacy vertical sync shift. Not used. Note: VGA display modes, sDVO line stall, and Panel fitting do not work while in interlaced modes Setting the Interlaced embedded panel mode causes hardware to automatically modify the output to match the specifications of panels that support interlaced mode.
20	Reserved: Write as zero
19	**Display / Overlay Planes Off**. This bit when set will cause all enabled Display and overlay planes that are assigned to this pipe to be disabled by overriding the current setting of the plane enable bit, at the next VBLANK. Timing signals continue as they were but the screen becomes blank. Setting the bit back to a zero will then allow the display and overlay planes to resume on the following VBLANK. 0 = Normal Operation 1 = Planes assigned to this pipe are disabled.
18	**Cursor Planes Off**. This bit when set will cause all enabled cursor planes that are assigned to this pipe to be disabled by overriding the current setting of the plane enable bit, at the next VBLANK. Timing signals continue as they were but the screen becomes blank. Setting the bit back to a zero will then allow the cursor planes to resume on the following VBLANK. 0 = Normal Operation 1 = Planes assigned to this pipe are disabled.
17:16	**Refresh Rate CxSR Mode Association** These bits select how refresh rates are tied to CxSR on pipe B. When they are set to anything other than 00, bits 23:21 of this register must be programmed to 0xx. Switching between 01 and 10 settings directly is not allowed. Software must program this field to 00 before switching. Software is responsible for enabling this mode only for integrated dispay panels that support corresponding mode. 00 – **Default** – no dynamic refresh rate change enabled. Software control only. 01 – Progressive-to-progressive refresh rate change enabled and tied to CxSR. Pixel clock values set in FPB0/FPB1 settings in the DPLLB control register and FPB0/FPB1 divider registers. 10 – Progressive-to-interlaced refresh rate change enabled and tied to CxSR. Pixel clock value does not change in this case. Scaling must be disabled in this mode. Uses programmable VS shift 11 – **Reserved**
15:10	Reserved: MBZ.
9:8	Reserved: MBZ.

Bit	Descriptions
7:5	Bits Per Color[DevCTG]: This field selects the number of bits per color sent to a receiver device connected to this port. Color format takes place on the Vblank after being written. Color format change can be done independent of a pixel clock change. Selecting a pixel color depth higher or lower than the pixel color depth of the frame buffer results in dithering the output stream. For further details on Display Port fixed frequency programming to accommodate these formats refer to "DP Frequency Programming" in DPLL section of Bspec. 000 = 8 bits per color (Default) 001 = 10 bits per color 010 = 6 bits per color 011 = RESERVED 1xx = RESERVED
4	Dithering enable [DevCTG]: This bit enables dithering for DisplayPort 6bpc or 8bpc modes 0 – Dithering disabled (Default) 1 – Dithering enabled
3:2	Dithering type [DevCTG]: This bit selects dithering type for DisplayPort 6bpc or 8bpc modes 00 - Spatial only (default) 01- Spatio-Temporal 1 10- Spatio-Temporal 2 (testmode) 11- Temporal only (testmode)
1	DDA reset [DevCTG]: 0 – Do not reset DDA 1 – Reset DDA every 8th display frame
0	**Reserved: Write as zero**

2.10.4.4 PIPEBGCMAXRED—Pipe B Gamma Correction Max Red

Memory Offset Address: 71010h
Default: 00010000h
Normal Access: Read/Write

Bit	Descriptions
31:17	Reserved
16:0	**Max Red Gamma Correction Point**. 129th reference point for red channel of the pipe piecewise linear gamma correction. The value should always be programmed to be less than or equal to 1024.0. Format: 11.6 Default: 0x10000

2.10.4.5 PIPEBGCMAXGREEN—Pipe B Gamma Correction Max Green

Memory Offset Address: 71014h
Default: 00010000h
Normal Access: Read/Write

Bit	Descriptions
31:17	Reserved
16:0	**Max Green Gamma Correction Point**. 129th reference point for green channel of the pipe piecewise linear gamma correction. The value should always be programmed to be less than or equal to 1024.0. Format: 11.6 Default: 0x10000

2.10.4.6 PIPEBGCMAXBLUE—Pipe B Gamma Correction Max Blue

Memory Offset Address: 71018h
Default: 00010000h
Normal Access: Read/Write

Bit	Descriptions
31:17	Reserved
16:0	**Max Blue Gamma Correction Point**. 129th reference point for blue channel of the pipe piecewise linear gamma correction. The value should always be programmed to be less than or equal to 1024.0. Format: 11.6 Default: 0x10000

2.10.4.7 PIPEBSTAT—Pipe B Status

Memory Offset Address:	71024h
Default:	00000000h
Normal Access:	Read/Write

Programming:

Prior to clearing a Display Pipe-sourced interrupt (e.g., Display Pipe A VBLANK) in the IIR, the corresponding interrupt (source) status in the PIPEASTAT or PIPEBSTAT register (e.g., Pipe A VBLANK Interrupt Status bit of PIPEASTAT) must first be cleared. Note that clearing these status bits requires writing a '1' to the appropriate bit position.

Bit	Descriptions
31	**Pipe B Underflow Status:** This bit is set when an underflow occurs at the display pipe B. It is cleared by writing a one to this bit. This event will occur naturally during mode changes, to be effective, it should be cleared after a mode change. This bit is only valid after Pipe B has been completely configured. 1 = FIFO B Underflow occurred 0 = FIFO B Underflow did not occur
30	Reserved: Write as zero.
29	**CRC Error Enable:** This will enable the consideration of the CRC error status bit in the first line interrupt/status logic. 0 = CRC Error Detect Disabled 1 = CRC Error Detect Enabled
28	**CRC Done Enable:** This will enable the consideration of the CRC done status bit in the first line interrupt/status logic. 0 = CRC Done Detect Disabled 1 = CRC Done Detect Enabled
27:26	Reserved: Write as zero.
25	**Vertical Sync Interrupt Enable:** 0 = Vertical Sync Interrupt/Status Disabled 1 = Vertical Sync Interrupt/Status Enabled
24	**Display Line Compare Enable:** 0 = Pipe B Display Line Compare Status Report Disabled 1 = Pipe B Display Line Compare Status report Enabled
23	**[DevCL]: BLM Event Enable**. This interrupt is generated by the image brightness segment comparators. Which segment cause an interrupt are controlled by the BLM Histogram control register. 0 = No BLM event enabled 1 = BLM event enabled

Bit	Descriptions
22	**[DevCL]: Legacy BLC Event Enable**. This will enable writes to the PCI Backlight Control Register to cause the display B event status to be set and an Interrupt if Display B Event interrupt is enabled. 0 = No BLC Event enabled 1 = BLC Event enabled
21	**Odd Field Interrupt Event Enable**. This bit should only be used when this pipe is in an interlaced display timing. 0 = Odd Field Event disable 1 = Odd Field Event enable
20	**Even Field Interrupt Event Enable**. This bit should only be used when this pipe is in an interlaced display timing. 0 = Even field Event disable 1 = Even field Event enable
19	Reserved: Read-only as zero
18	**Start of Vertical Blank Interrupt Enable:** This will enable the consideration of the start of vertical blank interrupt status bit in the first line interrupt/status logic. 0 = Start of Vertical Blank Interrupt/Status Disabled 1 = Start of Vertical Blank Interrupt/Status Enabled
17	**Vertical Blank Interrupt Enable:** This will enable the consideration of the vertical blank interrupt status bit in the first line interrupt/status logic. 0 = Vertical Blank Interrupt/Status Disabled 1 = Vertical Blank Interrupt/Status Enabled
16	Reserved: Write as zero.
15:14	Reserved: Read-only.
13	**CRC Error Status:** This bit is set when a Pipe B CRC error is detected. It is cleared by a write of a one. 0 = No CRC Error 1 = CRC Error detected
12	**CRC Done Interrupt Status:** This bit is set when Pipe B CRC calculation and compare are complete. It is cleared by a write of a one. 0 = CRC Not Done 1 = CRC Done
11:10	Reserved: Read-only, write as zero
9	**Pipe B Vertical Sync Status:** 0 = Vertical Sync not asserted 1 = Vertical Sync asserted
8	**Pipe B Display Line Compare Status:** This bit is cleared when a write to this register occurs with this bit as a one. Writes with this bit as a zero has no effect on the value of the bit. 0 = Display Line Compare Status not asserted 1 = Display Line Compare Status asserted

Bit	Descriptions
7	**[DevCL]: BLM Image Brightness Status**. This bit is cleared when a write to this register occurs with this bit as a one. Writes with this bit as a zero has no effect on the value of the bit. 0 = DPST Interrupt has not occurred on pipe B 1 = DPST Interrupt has occurred on pipe B
6	**[DevCL]: Legacy BLC Event Status**. This status bit indicates that a write to the PCI Backlight Control Register (LBPC) has occurred. Software must clear this bit in order to detect subsequent writes, for example while servicing the Event Interrupt. This bit is cleared when a write to this register occurs with this bit as a one. Writes with this bit as a zero has no effect on the value of the bit. 0 = No BLC write detected 1 = A BLC write was detected
5	**Odd Field Interrupt Status**. This status bit will be set on a Odd field VBLANK event. This bit should only be used when this pipe is in an interlaced display timing. For synchronization with register updates, the actual event will occur one line after the start of VBLANK. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. Note: This bit will not be set when pipe is in “Interlaced with Field 0 Only using legacy vertical sync shift” mode. 0 = Odd Field Vertical Blank has not occurred 1 = Odd Field Vertical Blank has occurred
4	**Even Field Interrupt Status**. This status bit will be set on a even filed VBLANK event. This bit should only be used when this pipe is in an interlaced display timing. For synchronization with register updates, the actual event will occur one line after the start of VBLANK. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. Note: This bit will not be set when pipe is in “Interlaced with Field 0 Only using legacy vertical sync shift” mode. 0 = Even Field Vertical Blank has not occurred 1 = Even Field Vertical Blank has occurred
3	Reserved: Read-only, write as zero
2	**Start of Vertical Blank Interrupt Status:** This status bit will be set at the beginning of a VBLANK event. At this point, the double buffered display registers flip, taking their new values. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. 0 = Start of Vertical Blank has not occurred 1 = Start of Vertical Blank has occurred
1	**Vertical Blank Interrupt Status:** This status bit will be set on a VBLANK event, when the frame start occurs. The display registers are updated at the start of vertical blank, but the new register data is not utilized by the display pipeline until the point in the vertical blank period when the frame start occurs, which is the event that triggers this bit. To use this bit in a polling manner, clear the bit by writing a one to it followed by the polling loop waiting for it to become set. 0 = Vertical Blank has not occurred 1 = Vertical Blank has occurred
0	Reserved: Read-only, write as zero

2.10.4.8 PipeBFrameHigh— Pipe B Frame Count High [DevCL]

Memory Offset Address:	71040h
Default:	00000000h
Normal Access:	Read-Only

Bit	Descriptions
31:16	Reserved
15:0	**Pipe B Frame Count High** See PipeAFrameHigh description.

2.10.4.9 PIPEB_FRMCOUNT—Pipe B Frame Counter [DevCTG, DevELK]

PIPEB_FRMCOUNT—Pipe B Frame Counter	
Register Type:	MMIO
Address Offset:	71040h
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32

Bit	Description
31:0	**Pipe_Frame_Counter** Project: All Format: See Pipe A description

2.10.4.10 PipeBFramePixel — Pipe B Frame Count Low and Pixel Count [DevCL]

Memory Offset Address:	71044h
Default:	00000000h
Normal Access:	Read-Only

Bit	Descriptions
31:24	**Frame Count Low**. See PipeAFramePixel description.
23:0	**Pixel Count**. See PipeAFramePixel description.

2.10.4.11 PIPEB_FLIPCOUNT—Pipe B Flip Counter [DevCTG, DevELK]

PIPEB_FLIPCOUNT—Pipe B Flip Counter

Register Type:	MMIO
Address Offset:	71044h
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32

Bit	Description
31:0	**Pipe_Flip_Counter** Project: All Format: See Pipe A description

2.10.4.12 PIPEB_FRMTIMESTAMP—Pipe B Frame Time Stamp [DevCTG, DevELK]

PIPEB_FRMTIMESTAMP—Pipe B Frame Time Stamp

Register Type:	MMIO
Address Offset:	71048h
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32

Bit	Description
31:0	**Pipe_Frame_Time_Stamp** Project: All Format: See Pipe A description

2.10.4.13 PIPEB_FLIPTIMESTAMP—Pipe B Flip Time Stamp [DevCTG, DevELK]

PIPEB_FLIPTIMESTAMP—Pipe B Flip Time Stamp	
Register Type:	MMIO
Address Offset:	7104Ch
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32

Bit	Description
31:0	**Pipe_Flip_Time_Stamp** Project: All Format: See Pipe A description

2.10.4.14 PipeBGMCHDataM— Pipe B Data M value [DevCTG]

Memory Offset Address:	71050h
Default:	00000000h
Normal Access:	Read/Write

Bit	Descriptions
31	Reserved. Must be zero
30:25	TU Size. This field is the size of the transfer unit for DP, minus one. Default value to program = 111111 (TU size of 64)
24	Reserved. Must be zero
23:0	Pipe B Data M value. This field is the m value for internal use of the DDA. Calculation of this value is as follows: Data m/n = dot clock * bytes per pixel / ls_clk * # of lanes Please note that in the DisplayPort specification, dot clock is referred to as strm_clk

2.10.4.15 PipeBGMCHDataN— Pipe B Data N value [DevCTG]

Memory Offset Address: 71054h
Default: 00000000h
Normal Access: Read/Write

Bit	Descriptions
31:24	Reserved. Must be zero
23:0	Pipe B Data N value. This field is the m value for internal use of the DDA. Calculation of this value is as follows: Data m/n = dot clock * bytes per pixel / ls_clk * # of lanes Please note that in the DisplayPort specification, dot clock is referred to as strm_clk

2.10.4.16 PipeBDPLinkM— Pipe B Link M value [DevCTG]

Memory Offset Address: 71060h
Default: 00000000h
Normal Access: Read/Write

Bit	Descriptions
31:24	Reserved. Must be zero
23:0	Pipe B Link M value. This field is the m value for external transmission in the Main Stream Attributes. Calculation of this value is as follows: Link m/n = pixel clk / ls_clk Please note that in the DisplayPort specification, pixel clk is referred to as strm_clk

2.10.4.17 PipeBDPLinkN— Pipe B Link N value [DevCTG]

Memory Offset Address: 71064h
Default: 00000000h
Normal Access: Read/Write

Bit	Descriptions
31:24	Reserved. Must be zero
23:0	Pipe B Data N value. This field is the m value for external transmission in the Main Stream Attributes and VB-ID. Calculation of this value is as follows (to be filled in): Link m/n = pixel clk / ls_clk Please note that in the DisplayPort specification, pixel clk is referred to as strm_clk

2.10.5 Display A (Primary) Plane Control

2.10.5.1 DSPACNTR—Display A Plane Control Register

Memory Offset Address: 70180h
Default: 00000000h
Normal Access: Read/Write Double buffered

Note: The active set of registers will be updated on the VBlank (of the currently selected pipe) after the “trigger” register (the Start Address register or the Control register when plane enable bit transitioning from a zero to a one) is written – thus providing an atomic update of all display controls. If the currently selected pipe is disabled, the update is immediate.

Bit	Descriptions
31	**Display Plane A (Primary A) Enable:** When this bit is set, the primary plane will generate pixels for display. When set to zero, display plane A memory fetches cease and display is blanked (from this plane) at the next VBLANK event from the pipe that display A is assigned. The display pipe must be enabled to enable this plane. There is an override for the enable of this plane in the Pipe Configuration register. 1 = Enable 0 = Disable
30	**Display A Gamma Enable:** This bit should only be changed after the plane has been disabled. It controls the bypassing of the display pipe gamma unit for this display plane's pixel data only. For 8-bit indexed display data, this bit should be set to a one. 0 = Display A pixel data bypasses the display pipe gamma correction logic (default). 1 = Display A pixel data is gamma corrected in the display pipe gamma correction logic.

Bit	Descriptions
29:26	**Display A Source Pixel Format:** These bits should only be changed after the plane has been disabled. Pixel formats with an alpha channel (8:8:8:8) should not use source keying. Pixel format of 8-bit indexed uses the palette. Before entering the blender, each source format is converted to 10 bits per pixel (details are described in the intermediate precision for the blender section of the Display Functions chapter). 000x = Reserved. 0010 = 8-bpp Indexed. 0011 = Reserved. 0100 = Reserved. 0101 = 16-bit BGRX (5:6:5:0) pixel format (XGA compatible). 0110 = 32-bit BGRX (8:8:8:8) pixel format. Ignore alpha. 0111 = Reserved. 1000 = 32-bit RGBX (10:10:10:2) pixel format. Ignore alpha. 1001 = Reserved. 1010 = BGRX 10:10:10:2 1011 = Reserved. 1100 = 64-bit RGBX (16:16:16:16) 16-bit floating point pixel format. Ignore alpha. 1101 = Reserved. 1110 = 32-bit RGBX (8:8:8:8) pixel format. Ignore alpha. 1111 = Reserved.
25:24	**Display A Pipe Select:** This is read-only and selects the display pipe that this plane is assigned to. It is hardwired to pipe A. 00 = Select Pipe A (**default: cannot be changed**) 01 = Select Pipe B 10 = Reserved for pipe C 11 = Reserved for pipe D
23	**Key Window Enable**. This bit applies only to devices with a display plane C. This bit is set to one when the color key is used as a destination key for display C. Display plane C must be enabled on the same pipe and its Z-order should be programmed to be behind display A for this to be set to a one. 0 = Source Key applies to entire display plane A 1 = Source Key applies to only pixels within the intersection between Display A and Display C
22	**Key Enable**. This bit enables source keying for display A. Source keying allows a plane that is behind (below) this plane to show through where the display A key matches the display A data. This function is overloaded to provide display C destination keying when combined with the key window enable bit. Setting this bit is not allowed when the display pixel format includes an alpha channel. 0 = Source key is disabled 1 = Source key is enabled
21:20	**Pixel Multiply:** This cause the display plane to duplicate lines and pixels sent to the assigned pipe. In the pixel multiply mode, the horizontal pixels are doubled and lines are sent twice. Asynchronous flips are not used in this mode. **Programming Notes:** • Asynchronous flips are not permitted when pixel multiply is enabled. 00 = No duplication 01 = Line/pixel Doubling 10 = Reserved 11 = Pixel Doubling only
19	Reserved: Software must preserve the contents of this bit.
18	Reserved: Write as zero

Bit	Descriptions
17:16	Reserved: Software must preserve the contents of this bit.
15	**180° Display Rotation:** This mode causes the display plane to be rotated 180°. In addition to setting this bit, software must also set the base address to the lower right corner of the unrotated image. [DevCL] Do not enable 180° rotation together with Frame Buffer Compression0 = No rotation 1 = 180° rotation
14	**[DevCTG] Display A Trickle Feed Enable:** 0 = Trickle Feed Enabled - Display A data requests are sent whenever there is space in the Display Data Buffer. 1 = Trickle Feed Disabled - Display A data requests are sent in bursts. Note: On mobile products this bit will be ignored such that Trickle Feed is always disabled. **[DevELK] Must always be programmed disabled** **[DevBW] and [DevCL]: Reserved**
13	**[DevCTG] Display A Data Buffer Partitioning Control:** 0 = Display A Data Buffer will encompass Sprite A buffer space when Sprite A is disabled. 1 = Display A Data Buffer will not use Sprite A buffer space when Sprite A is disabled. Note: When in C3xR Max FIFO mode, this bit will be ignored. **[DevBW] and [DevCL]: Reserved**
12:11	Reserved
10	**Tiled Surface**. This bit indicates that the display A surface data is in tiled memory. The tile pitch is specified in bytes in the DSPASTRIDE register. Only X tiling is supported for display surfaces. When this bit is set, it affects the hardware interpretation of the DSPATILEOFF, DSPALINOFF, and DSPASURF registers. 0 = Display A surface uses linear memory 1 = Display A surface uses X-tiled memory

Bit	Descriptions
9	**[DevCTG] Asynchronous Surface Address Update Enable:** This bit will enable asynchronous updates of the surface address when written by MMIO. The surface address will change with the next TLB request or when start of vertical blank is reached. Updates during vertical blank may not complete until after the first few active lines are displayed. **Restrictions:** No command streamer initiated surface address updates are allowed when this bit is enabled. Only one asynchronous update may be made per frame. Must wait for vertical blank before again writing the surface address register. 0 = DSPASURF MMIO writes will update synchronous to start of vertical blank (default) 1 = DSPASURF MMIO writes will update asynchronously [DevBW] and [DevCL] Reserved: **Write as zero**
8	**Reserved: Write as zero**

2.10.5.2 DSPALINOFF—Display A Linear Offset Register

Memory Offset Address: 70184h
Default: 00000000h
Normal Access: Read/Write Double buffered

This register specifies the panning for the display surface. The surface base address is specified in the DSPASURF register, and this register is used to describe an offset from that base address. Bit 10 of DSPACNTR specifies whether the display A surface is in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the contents of this register are ignored.

This register can be written directly through software or by load register immediate command packets in the command stream.

This register is double buffered by VSYNC only. A change to this register will take effect on the next vsync following the write.

Bit	Descriptions
31:0	**Display A Offset:** This register provides the panning offset into the display A plane. This value is added to the surface address to get the graphics address of the first pixel to be displayed. This offset must be at least pixel aligned. This offset is the difference between the address of the upper left pixel to be displayed and the display surface address. When performing 180° rotation, this offset must be the difference between the last pixel of the last line of the display data in its unrotated orientation and the display surface address.

2.10.5.3 DSPASTRIDE—Display A Stride Register

Memory Offset Address: 70188h
Default: 00000000h
Normal Access: Read/Write Double Buffered

Bit	Descriptions
31:0	**Display A Stride:** This is the stride for display A in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. This value is used to determine the line to line increment for the display. This register is updated either through a command packet passed through the command stream or writes to this register. When it is desired to update both this and the start register, the stride register must be written first because the write to the start register is the trigger that causes the update of both registers on the next VBLANK event. When using tiled memory, the actual memory buffer stride is limited to a maximum of 16K bytes. The display stride must be power of 2 when doing Asynch Flips. The display stride must be 8KB or greater when doing Asynch Flips together with 180 rotation. The value in this register is updated through the command streamer during a synchronous flip.

2.10.5.4 DSPARESV—Display A Reserved

Memory Offset Address: 7018Ch–7018Fh
Default: 00000000h
Normal Access: Read-Only

2.10.5.5 DSPARESV—Display A Reserved

Memory Offset Address: 70190h
Default: 00000000h
Normal Access: Read/Write Double Buffered

2.10.5.6 DSPAKEYVAL—Sprite Color Key Value Register

Memory Offset Address: 70194h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the key color to be used with the mask bits to determine if the display source data matches the key. This register will only have an effect when the display color key is enabled. The overlay destination key value is used for overlay keying when Display A is being used as a primary display with overlay destination keying enabled. This key can be used as a Display C destination key onto Display A.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	**Red Key Value:** Specifies the color key value for the sprite red/Cr channel.
15:8	**Green Key Value:** Specifies the color key value for the sprite green/Y channel.
7:0	**Blue Key Value:** Specifies the color key value for the sprite blue/Cb channel.

2.10.5.7 DSPAKEYMSK—Sprite Color Key Mask Register

Memory Offset Address: 70198h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the key mask to be used with the color value bits to determine if the display source data matches the key when enabled. A zero bit in the mask indicates that the corresponding bit match failure should be ignored when determining if the pixel matches.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	**Red mask Value:** Specifies the color key mask for the sprite red/Cr channel.
15:8	**Green mask Value:** Specifies the color key mask for the sprite green/Y channel.
7:0	**Blue mask Value:** Specifies the color key mask for the sprite blue/Cb channel.

2.10.5.8 DSPASURF—Display A Surface Base Address Register

Memory Offset Address: 7019Ch
Default: 00000000h
Normal Access: Read/Write Double buffered

Writing to this register triggers the display plane flip. When it is desired to change multiple display A registers, this register should be written last as a write to this register will cause all new register values to take effect.

Bit	Descriptions
31:29	Reserved
28:12	**Display A Surface Base Address**. This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DSPATILEOFF register. When the surface is in linear memory, panning is specified using a linear offset in the DSPALINOFF register. This address must be 4K aligned. When performing asynchronous flips and the display surface is in tiled memory, this address must be 256K aligned. This register can be written directly through software or by command packets in the command stream. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. This address must be 128K aligned for linear memory.
11:0	Reserved

2.10.5.9 DSPATILEOFF—Display A Tiled Offset Register

Memory Offset Address: 701A4h
Default: 00000000h
Normal Access: Read/Write Double buffered

This register specifies the panning for the display surface. The surface base address is specified in the DSPASURF register, and this register is used to describe an offset from that base address. Bit 10 of DSPACNTR specifies whether the display A surface is in linear or tiled memory. When the surface is in linear memory, the offset is specified in the DSPALINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

This register can be written directly through software or by load register immediate command packets in the command stream.

This register is double buffered by VBLANK only. A change to this register will take effect on the next vblank following the write.

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	**Plane Start Y-Position:** These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation.
15:12	Reserved: Write as zero
11:0	**Plane Start X-Position:** These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation. When display stride is 16KB and doing Asynch Flips, do not program the offset to give pans of 7680 to 8191 bytes.

2.10.5.10 DSPASURFLIVE—Display A Live Surface Base Address Register [DevCTG-B]

Memory Offset Address: 701ACh
Default: 00000000h
Normal Access: Read Only

Bit	Descriptions
31:0	Display A Live Surface Base Address. This gives the live value of the surface base address as being currently used for the plane.

2.10.5.11 DSPAFLPQSTAT—Flip Queue Status Register

Memory Offset Address: 70200h
Default: 00000800h
Normal Access: Read/Write with Read-Only fields

Bit	Descriptions
31:16	Reserved: Write as zero (RO)
15:8	**Queue Free Entry Count (RO)**. This value indicates the number of free entries in the queue at the time that the register was read. The total number of entries in the queue is the sum of the occupied entry count and the free entry count.
7:0	**Queue Occupied Entry Count (RO)**. This value indicates the number of occupied entries in the queue at the time that the register was read. The total number of entries in the queue is the sum of the occupied entry count and the free entry count.

2.10.5.12 Chicken Bit Register

Memory Offset Address: 70400h
Default: 00000000h
Normal Access: Read Write

Bit	Descriptions
31:29	**Reserved:**
28:26	**VGA OOO queue depth** **0xx = Disable out-of-order stall logic for VGA** **100 = Enable out-of-order VGA stall depth of 64** **101 = Enable out-of-order VGA stall depth of 48** **110 = Enable out-of-order VGA stall depth of 32** **111 = Enable out-of-order VGA stall depth of 16**
25	**PLLB Safe shutdown override: This bit disables the dependency for pipe B to be disabled before the PLL is shut down**
24	**PLLA Safe shutdown override: This bit disables the dependency for pipe A to be disabled before the PLL is shut down**
23:22	**Reserved**
21	**Elpin 409 Select: This bit is used to select one of the elpin 409 bug fixes** **0 = vrd_ci_rreq** **1 = count comparator**
20	**Reserved**

Bit	Descriptions
19	CR12 Write Counter Reset 0 = Disable CR12 write counter reset 1 = Enable CR12 write counter reset
18	Reserved
17	Monitor Detection: This bit is used to test the monitor detection. Do not program unless directed.
16	Invert DPO Field: Invert DPO interlaced field output. This bit is used to invert the field sense input to the planes from DPO.
15:12	Reserved
11	VGA Stall: Stall native mode VGA when frequency is over 50 MHz. This bit is only used during VGA native mode.
10	Reserved: [DevCL] Disable Crestline C0 fix for requests performance: On Crestline C0 and later steppings, this bit must always be programmed to "1" to prevent async flip failures.
9	Pixel Size: This bit changes the VGA pixel width and height calculations.
8	Immediate Asynchronous Flips: This bit causes asynchronous flips to complete immediately upon the start of the vertical blank period. When enabling this feature, frame start should also be moved to the end of the vertical blank period by setting the frame start position bit.
7	Pipe B Frame Start Position: This bit changes the position of frame start on pipe B. This feature is used in conjunction with the immediate asynchronous flips bit to enable fast asynchronous flips during vertical blanking. 0 = frame start occurs at start of the vertical blank period 1 = frame start occurs at end of the vertical blank period Default: 1, causing frame start to occur at the end of vertical blank
6	Pipe B Palette Write Enable: Disables anti-collision logic in the palette during non-blanking periods on pipe B.
5	Pipe A Palette Write Enable: Disables anti-collision logic in the palette during non-blanking periods on pipe A
4	Reserved:

Bit	Descriptions
3	**Pipe A Frame Start Position: This bit changes the position of frame start on pipe A. This feature is used in conjunction with the immediate asynchronous flips bit to enable fast asynchronous flips during vertical blanking.** **0 = frame start occurs at start of the vertical blank period** **1 = frame start occurs at end of the vertical blank period** **Default: 1, causing frame start to occur at the end of vertical blank**
2:0	**Reserved:**

2.10.6 Display B (Second Primary or Sprite) Control

All Display B/Sprite registers are double buffered. The active set of registers will be updated on the VBlank (of the currently selected pipe) after the "trigger" register (the Start Address register or the Control register when plane enable bit transitioning from a zero to a one) is written – thus providing an atomic update of all display controls. If the currently selected pipe is disabled, the update is immediate.

2.10.6.1 DSPBCNTR—Display B/Sprite Plane Control Register

Memory Offset Address:	71180h
Default:	01000000h
Normal Access:	Read/Write Double Buffered

The active set of registers will be updated on the VBlank (of the currently selected pipe) after the "trigger" register (the Start Address register or the Control register when plane enable bit transitioning from a zero to a one) is written – thus providing an atomic update of all display controls. If the currently selected pipe is disabled, the update is immediate.

Bit	Descriptions
31	**Display B/Sprite (Primary B) Enable:** This bit will enable or disable the display B/sprite. When this bit is set, the plane will generate pixels for display. When set to zero, memory fetches cease and display is blanked (from this plane) at the next VBLANK event from the pipe that this plane is assigned. At least one of the display pipes must be enabled to enable this plane. There is an override for the enable of this plane in the Pipe Configuration register. 1 = Enable 0 = Disable
30	**Display B/Sprite Gamma Enable:** This bit should only be changed after the plane has been disabled. It controls the bypassing of the display pipe gamma unit for this display plane pixel data only. For 8-bit indexed display data, this bit should be set to a one. 0 = Display B pixel data bypasses the pipe gamma correction logic (default). 1 = Display B pixel data is gamma corrected in the pipe gamma correction logic

Bit	Descriptions
29:26	**Display B Source Pixel Format:** This field selects the pixel format for the sprite/display B. Pixel formats with an alpha channel (8:8:8:8) should not use source keying. Before entering the blender, each source format is converted to 10 bits per pixel (details are described in the intermediate precision for the blender section of the Display Functions chapter). 000x = Reserved. 0010 = 8-bpp Indexed. 0011 = Reserved. 0100 = Reserved. 0101 = 16-bit BGRX (5:6:5:0) pixel format (XGA compatible). 0110 = 32-bit BGRX (8:8:8:8) pixel format. Ignore alpha. 0111 = Reserved. 1000 = 32-bit RGBX (10:10:10:2) pixel format. Ignore alpha. 1001 = Reserved. 1010 = BGRX 10:10:10:2 1011 = Reserved. 1100 = 64-bit RGBX (16:16:16:16) 16-bit floating point pixel format. Ignore alpha. 1101 = Reserved. 1110 = 32-bit RGBX (8:8:8:8) pixel format. Ignore alpha. 1111 = Reserved.
25:24	**Display B/Sprite Pipe Select:** This is read-only and selects the display pipe that this plane is assigned to. It is hardwired to pipe B. 00 = Select Pipe A 01 = Select Pipe B (**default: cannot be changed**) 10 = Reserved for pipe C 11 = Reserved for pipe D
23	**Key Window Enable: This applies only to devices with a Display Plane C**. It determines what area of the screen the source key compare should be applied. This bit is set to one when the color key is used as a destination key for display C. Display plane C must be enabled on the same pipe and display A should not be enabled on this pipe for this to be used. The function is only effective when display C is enabled and defined by Z-order to be behind display B. 0 = If keying is enabled, it applies to the entire display B plane 1 = If keying is enabled, it applies only to the intersection between display B and display C
22	**Source Key Enable:** When used as a sprite or a secondary this enables source color keying. Sprite pixel values that match the key will become transparent. Source keying allows a plane that is behind (below) this plane to show through where the display B data matches the display B key. This function is overloaded to provide display C destination keying when combined with the key window enable bit.. Setting this bit is not allowed when the display pixel format includes an alpha channel. 0 = Sprite source key is disabled (default) 1 = Sprite source key is enabled.

Bit	Descriptions
21:20	**Pixel Multiply:** This cause the display plane to duplicate lines and pixels sent to the assigned pipe. In the line/pixel doubling mode, the horizontal pixels are doubled and lines are sent twice. Asynchronous flips are not used in this mode. **Programming Notes:** • Asynchronous flips are not permitted when pixel multiply is enabled. 00 = No duplication 01 = Line/pixel Doubling 10 = Reserved 11 = Pixel Doubling only
19:16	**Reserved:** Write as zero
15	**180° Display Rotation:** This mode causes the display plane to be rotated 180°. In addition to setting this bit, software must also set the base address to the lower right corner of the unrotated image. [DevCL] Do not enable 180° rotation together with Frame Buffer Compression 0 = No rotation 1 = 180° rotation
14	[DevCTG] Display B Trickle Feed Enable: 0 = Trickle Feed Enabled - Display B data requests are sent whenever there is space in the Display Data Buffer. 1 = Trickle Feed Disabled - Display B data requests are sent in bursts. Note: On mobile products this bit will be ignored such that Trickle Feed is always disabled. [DevELK] Must always be programmed disabled **[DevBW] and [DevCL]: Reserved**

13	[DevCTG] Display B Data Buffer Partitioning Control: 0 = Display B Data Buffer will encompass Sprite B buffer space when Sprite B is disabled. 1 = Display B Data Buffer will not use Sprite B buffer space when Sprite B is disabled. Note: When in C3xR Max FIFO mode, this bit will be ignored. **[DevBW] and [DevCL]: Reserved**
12:11	**Reserved**
10	Tiled Surface This bit indicates that the display B surface data is in tiled memory. The tile pitch is specified in bytes in the DSPBSTRIDE register. Only X tiling is supported for display surfaces. When this bit is set, it affects the hardware interpretation of the DSPBLINOFF, DSPBTILEOFF, and DSPBSURF registers. 0 = Display B surface uses linear memory **1 = Display B surface uses X-tiled memory**
9	[DevCTG] Asynchronous Surface Address Update Enable: This bit will enable asynchronous updates of the surface address when written by MMIO. The surface address will change with the next TLB request or when start of vertical blank is reached. Updates during vertical blank may not complete until after the first few active lines are displayed. Restrictions: No command streamer initiated surface address updates are allowed when this bit is enabled. Only one asynchronous update may be made per frame. Must wait for vertical blank before again writing the surface address register. 0 = DSPBSURF MMIO writes will update synchronous to start of vertical blank (default) 1 = DSPBSURF MMIO writes will update asynchronously **[DevBW] and [DevCL] Reserved: Write as zero**
8:1	**Reserved: Write as zero**
0	**Reserved: Write as zero**

2.10.6.2 DSPBLINOFFSET —Display B/Sprite Linear Offset Register

Memory Offset Address:	71184h
Default:	00000000h
Normal Access:	Read/Write Double Buffered

This register specifies the panning for the display surface. The surface base address is specified in the DSPBSURF register, and this register is used to describe an offset from that base address. Bit 10 of DSPBCNTR specifies whether the display B surface is in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the contents of this register are ignored.

This register can be written directly through software or by load register immediate command packets in the command stream.

This register is double buffered by VSYNC only. A change to this register will take effect on the next vsync following the write.

Bit	Descriptions
31:0	**Display B Offset:** This register provides the panning offset into the display B plane. This value is add to the surface address to get the graphics address of the first pixel to be displayed. This offset must b at least pixel aligned. This offset is the difference between the address of the upper left pixel to be displayed and the display surface address. When performing 180° rotation, this offset must be the difference between the last pixel of the last line of the display data in its unrotated orientation and the display surface address.

2.10.6.3 DSPBSTRIDE—Display B/Sprite Stride Register

Memory Offset Address:	71188h
Default:	00000000h
Normal Access:	Read/Write Double Buffered

Bit	Descriptions
31:0	**Display B/Sprite Stride:** This is the stride for display B/Sprite in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. The maximum value for this register is fixed. This register is updated through a command packet passed through the command stream or writes to this register. When it is desired to update both this and the start register, the stride register must be written first because the write to the start register is the trigger that causes the update of both registers on the next VBLANK event. When using tiled memory, the actual memory buffer stride is limited to a maximum of 16K bytes. The display stride must be power of 2 when doing Asynch Flips. The display stride must be 8KB or greater when doing Asynch Flips together with 180 rotation. The value in this register is updated through the command streamer during a synchronous flip.

2.10.6.4 DSPBKEYVAL—Sprite Color Key Value Register

Memory Offset Address: 71194h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite color key is enabled. The overlay destination key value is used for overlay keying when Display B is being used as a secondary display with overlay destination keying enabled.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	**Red Key Value:** Specifies the color key value for the sprite red/Cr channel.
15:8	**Green Key Value:** Specifies the color key value for the sprite green/Y channel.
7:0	**Blue Key Value:** Specifies the color key value for the sprite blue/Cb channel.

2.10.6.5 DSPBKEYMSK—Sprite Color Key Mask Register

Memory Offset Address: 71198h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the key mask to be used with the color value bits to determine if the sprite source data matches the key when enabled. A zero bit in the mask indicates that the corresponding bit match failure should be ignored when determining if the pixel matches.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	**Red Mask Value:** Specifies the color key mask for the sprite red/Cr channel.
15:8	**Green Mask Value:** Specifies the color key mask for the sprite green/Y channel.
7:0	**Blue Mask Value:** Specifies the color key mask for the sprite blue/Cb channel.

2.10.6.6 DSPBSURF—Display B Surface Address Register

Memory Offset Address:	7119Ch
Default:	00000000h
Normal Access:	Read/Write Double buffered

Writing to this register triggers the display plane flip. When it is desired to change multiple display B registers, this register should be written last as a write to this register will cause all new register values to take effect.

Bit	Descriptions
31:29	Reserved
28:12	**Display B Surface Base Address:** This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DSPBTILEOFF register. When the surface is in linear memory, panning is specified using a linear offset in the DSPBLINOFF register. This address must be 4K aligned. When performing asynchronous flips and the display surface is in tiled memory, this address must be 256K aligned. This register can be written directly through software or by command packets in the command stream. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. This address must be 128K aligned for linear memory.
11:0	Reserved

2.10.6.7 DSPBTILEOFF—Display B Tiled Offset Register

Memory Offset Address:	711A4h
Default:	00000000h
Normal Access:	Read/Write Double buffered

This register specifies the panning for the display surface. The surface base address is specified in the DSPBSURF register, and this register is used to describe an offset from that base address. Bit 10 of DSPBCNTR specifies whether the display B surface is in linear or tiled memory. When the surface is in linear memory, the offset is specified in the DSPBLINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

This register can be written directly through software or by load register immediate command packets in the command stream.

This register is double buffered by VBLANK only. A change to this register will take effect on the next vblank following the write.

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	**Plane Start Y-Position:** These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation.
15:12	Reserved: Write as zero
11:0	**Plane Start X-Position:** These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation. When display stride is 16KB and doing Asynch Flips, do not program the offset to give pans of 7680 to 8191 bytes.

2.10.6.8 DSPBFLPQSTAT—Flip Queue Status Register

Memory Offset Address:	71200h
Default:	00000800h
Normal Access:	Read/Write with Read-Only fields

Bit	Descriptions
31:16	Reserved: Write as zero (RO)
15:8	**Queue Free Entry Count (RO):** This value indicates the number of free entries in the queue at the time that the register was read. The total number of entries in the queue is the sum of the occupied entry count and the free entry count.
7:0	**Queue Occupied Entry Count (RO):** This value indicates the number of occupied entries in the queue at the time that the register was read. The total number of entries in the queue is the sum of the occupied entry count and the free entry count.

2.10.7 Video BIOS Registers (71400h- 714FFh)

2.10.7.1 VGACNTRL—VGA Display Plane Control Register

Memory Offset Address:	71400h
Default:	00000000h
Normal Access:	Read/Write

This register provides support for VGA compatibility modes. This register is used by video BIOS only.

Bit	Descriptions
31	**VGA Display Disable:** This bit will disable the VGA compatible display mode. It has no effect on VGA register or A0000-BFFFF memory aperture accesses which are controlled by the PCI configuration and VGA register settings. VGA display should only be enabled if all display planes other than VGA are disabled. After enabling the VGA, most display planes need to stay disabled, only the VGA popup (cursor A) can be enabled. VGA 132 Column text mode is not supported. 0 = VGA Display Enabled 1 = VGA Display Disabled
30	**VGA / Pop-up 2X Centered Mode Scaling**. When this bit is set to a one, the VGA and pop-up data is scaled using pixel doubling in both the horizontal and vertical direction for use on un-scaled flat panel displays. Setting this bit allows the VGA to run at higher dot clock frequencies and creates a larger (4x the size) image for better quality on larger displays. It is intended for use in one of the centering modes when not using the internal panel fitting. Do not use it for native VGA modes or when internal panel fitting is used to scale VGA. In the situations where it is used, for 1280 wide or larger panels this bit should be set. For exactly 1280 wide panels, the Nine-dot disable bit should also be set. This operation is in addition to the VGA functions that double the pixels and lines. 0 = VGA display is normal size 1 = VGA and VGA popup data is doubled in the horizontal and vertical direction.
29	**VGA Pipe Select:** This bit only applies to devices with dual pipe support. For devices with a single display pipe, this bit will be ignored. For dual pipe devices, this bit determines which pipe is to receive the VGA display data. This must be changed only when the VGA display is in the disabled state via the VGA display disable bit or during the write to enable VGA display. 0 = Selects Assigns the VGA display to Pipe A 1 = Selects Assigns the VGA display to Pipe B
28:27	Reserved: Software must preserve the contents of these bits.
26	**VGA Border Enable:** This bit determines if the VGA border areas during VGA centering modes are included in the active display area and do or do not appear on integrated TV encoder output and devices that use centering such as on DVO connected flat panel, TV displays, or integrated panels. For use with the internal panel fitting logic, the border if enabled will be scaled along with the pixel data. Setting this bit allows the popup to be positioned overlapping the border area of the image. 0 = VGA Border areas are not included in the image size calculations for centering only active area. 1 = VGA Border areas are enabled and is passed to the display pipe for display and used in the image size calculation for centering modes

Bit	Descriptions
25:24	**VGA Centering Enable:** VGA centering modes use the pipe timing generators to determine the actual display timings. This would normally correspond to the display panel size and timings. The VGA registers determine the centered VGA image height and width. The VGA border may or may not be considered in the calculation selected by the VGA Border Enable bit. For a proper image, the VGA image size should not exceed the pipe timing generator active rectangle. When using the internal panel fitting logic, the horizontal image size needs to be less than or equal to 2048 pixels to generate a proper image. The VGA image will either be centered within the pipe timing rectangle or appear in the upper left corner. Upper left corner centered mode is generally used for external panel scaling where the DVO stall signal is used and is always used for internal panel fitting operation. When panel fitter is enabled on the same pipe as VGA this register setting is ignored and upper left corner centered mode is always selected. When centering is disabled, the VGA CRTC registers determine the display timing compatible with legacy VGA devices for driving CRT like devices. 00 = VGA centering is disabled, VGA operates in Native VGA mode or when driving integrated TV (even when using integrated TV for hires mode). 01 = Reserved, invalid setting. 10 = VGA centering is enabled, VGA image appears in the upper left corner of the larger rectangle 11 = Reserved, invalid setting.
23	**VGA Palette Read Select:** This bit only applies to dual display pipe devices and determines which palette VGA palette read accesses will occur from. 0 = VGA palette reads will access Palette A (default). 1 = VGA palette reads will access Palette B VGA palette reads are reads from I/O address 0x3c9.
22	**VGA Palette A Write Disable:** This determines which palette the VGA palette writes will have as a destination. One or both palettes can be the destination. If both are disabled, writes will not affect the palette RAM contents. 0 = VGA palette writes will update Palette A (default). 1 = VGA palette writes will not update Palette A VGA palette writes are writes to I/O address 0x3C9h.
21	***Dual Pipe* VGA Palette B Write Disable:** This determines which palette the VGA palette writes will have as a destination. One or both palettes can be the destination. If both are disabled, writes will not affect the palette RAM contents. 0 = VGA palette writes will update Palette B (default). 1 = VGA palette writes will not update Palette B VGA palette writes are writes to I/O address 0x3C9h.
20	**Legacy VGA 8-Bit Palette Enable:** This bit only affects reads and writes to the palette through VGA I/O addresses. In the 6-bit mode, the 8-bits of data are shifted up two bits on the write (upper two bits are lost) and shifted two bits down on the read. It provides backward compatibility for original VGA programs (in its default state) as well as VESA VBE support for 8-bit palette. It does not affect palette accesses through the palette register MMIO path. 0 = 6-bit DAC (default). 1 = 8-bit DAC.
19	**Palette Bypass:** 0 = Pass VGA data through the palette for translation (Normal Operation) 1 = Bypass the palette for allowing testing without loading palette both VGA and popup data will bypass the palette in this mode.

Bit	Descriptions
18	**Nine Dot Disable** Prevents DOS applications from setting the VGA display into a real 9-dot per character operation mode, instead the device emulates that using 8-dots per character. This is intended to provide VGA compatibility on DVI type connectors and integrated panels where there would otherwise not be room for the 720 horizontal pixels or 1440 pixels when horizontally doubled. The VGA register bit SR01<0> functionality is disabled. VGA panning control handles the pseudo 9-dot mode when both this bit is set and SR01<0> is clear. 0 = Enable use of 9-dot enable bit in VGA registers 1 = Ignore the 9-dot per character bit and always use 8
17	Reserved
16:8	Reserved: Software must preserve the contents of these bits.
7:6	**Blink Duty Cycle:** Controls the VGA text mode blink duty cycle relative to the cursor blink duty cycle. 00 = 100% Duty Cycle, Full Cursor Rate (Default) 01 = 25% Duty Cycle, ½ Cursor Rate 10 = 50% Duty Cycle, ½ Cursor Rate 11 = 75% Duty Cycle, ½ Cursor Rate
5:0	**VSYNC Blink Rate:** Controls the VGA blink rate in terms of the number of VSYNCs per on/off cycle. These bits are programmed with the (VSYNCs/cycle)/2-1. The proper programming of this register is determined by the VSYNC rate that the display requires when in a VGA display mode.

2.10.7.2 SWFxx—Software Flag Registers

Memory Offset Address:	SWF00 = 70410h
	SWF01 = 70414h
	SWF02 = 70418h
	SWF03 = 7041Ch
	SWF04 = 70420h
	SWF05 = 70424h
	SWF06 = 70428h
	SWF07 = 7042Ch
	SWF08 = 70430h
	SWF09 = 70434h
	SWF0A = 70438h
	SWF0B = 7043Ch
	SWF0C = 70440h
	SWF0D = 70444h
	SWF0E = 70448h
	SWF0F = 7044Ch
	SWF10 = 71410h
	SWF11 = 71414h
	SWF12 = 71418h
	SWF13 = 7141Ch
	SWF14 = 71420h
	SWF15 = 71424h
	SWF16 = 71428h
	SWF17 = 7142Ch
	SWF18 = 71430h
	SWF19 = 71434h
	SWF1A = 71438h
	SWF1B = 7143Ch
	SWF1C = 71440h
	SWF1D = 71444h
	SWF1E = 71448h
	SWF1F = 7144Ch
	SWF30 = 72414h
	SWF31 = 72418h
	SWF32 = 7241Ch
Default:	00000000h
Normal Access:	Read/Write

These 32-bit registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by the software architecture.

Bit	Descriptions
31:0	Reserved for Video BIOS and Drivers

2.10.8 Display C (Sprite or Second Overlay) Control

All of the basic control Display C/Sprite registers are double buffered. The active set is updated after the trigger register is written followed by a VBLANK event on the pipe that the Display C/Sprite is assigned. The Display C color adjustment registers are not double buffered and take effect immediately.

2.10.8.1 DSPCCNTR—Display C Sprite Control Register

Memory Offset Address:	72180h
Default:	00000000h
Normal Access:	Read/Write Double Buffered

The active set of basic control registers will be updated on the VBlank (of the currently selected pipe) after the "trigger" register (the Start Address register or the Control register when plane enable bit transitioning from a zero to a one) is written – thus providing an atomic update of all display controls with the exception of the Display C color control registers. If the currently selected pipe is disabled, the update is immediate.

Bit	Descriptions
31	**Display C/Sprite Enable:** This bit will enable or disable the display C/sprite. When this bit is set, the plane will generate pixels for display to be combined by the blender for the target pipe. When set to zero, memory fetches cease and display is blanked (from this plane) at the next VBLANK event from the pipe that this plane is assigned. At least one of the display pipes must be enabled to enable this plane. There is an override for the enable of this plane in the Pipe Configuration register. 0 = Disable 1 = Enable
30	**Display C/Sprite Gamma Enable:** There are two gamma adjustments possible in the display C data path. This bit controls the gamma correction in the display pipe not the gamma control in this plane. It affects only the pixel data from this display plane. For pixel format of 8-bit indexed, this bit should be set to a one. Gamma correction logic that is contained in the display C logic is disabled by loading the default values into those registers. 0 = Display C pixel data bypasses the display pipe gamma correction logic (default). 1 = Display C pixel data is gamma corrected in the pipe gamma correction logic

Bit	Descriptions
29:26	**Display C Source Pixel Format:** This field selects the pixel format for the sprite/display C. Pixel formats with an alpha channel should not use source keying. Before entering the blender, each source format is converted to 10 bits per pixel (details are described in the intermediate precision for the blender section of the Display Functions chapter). 0000 = YUV 4:2:2 packed (see byte order below). 0001 = Reserved 0010 = 8-bpp Indexed. 0011 = Reserved. 0100 = Reserved. 0101 = 16-bit BGRX (5:6:5:0) pixel format (XGA compatible). 0110 = 32-bit BGRX (8:8:8:8) pixel format. Ignore alpha. 0111 = 32-bit BGRA (8:8:8:8) pixel format with pre-multiplied alpha channel. 1000 = 32-bit RGBX (10:10:10:2) pixel format. Ignore alpha. 1001 = 32-bit RGBA (10:10:10:2) pixel format 1010 = Reserved. 1011 = Reserved. 1100 = Reserved. 1101 = Reserved. 1110 = 32-bit RGBX (8:8:8:8) pixel format. Ignore alpha. 1111 = 32-bit RGBA (8:8:8:8)
25:24	**Display C/Sprite Pipe Select:** This selects the display pipe that this plane is assigned to. This bit can change when the sprite is active and causes a flip to the other display pipe. The position and size is still required to fit within the pipe source rectangle. The synchronization is handled in the hardware. 00 = Select Pipe A 01 = Select Pipe B 10 = Reserved for pipe C 11 = Reserved for pipe D
23	Reserved
22	**Sprite Source Key Enable:** When used as a sprite in the 16/32-bpp modes without alpha this enables source color keying. Sprite pixel values that match (within range) the key will become transparent. Setting this bit is not allowed when the display C pixel format includes an alpha channel. **[DevBW] Erratum:** This bit must always be set to 0 when display C pixel format is YUV 0 = Sprite source key is disabled (default) 1 = Sprite source key is enabled.
21:20	**Pixel Multiply:** This cause the display plane to duplicate lines and pixels sent to the assigned pipe. In the line/pixel doubling mode, the horizontal pixels are doubled and lines are sent twice. This is a method of scaling the source image by two (both H and V). 00 = No line/Pixel duplication 01 = Line/Pixel Doubling 10 = Reserved 11 = Pixel Doubling only

Bit	Descriptions
19	**Color Conversion Disabled:** This bit enables or disables the color conversion logic. Color conversion is intended to be used with the formats that support YUV formats such as the YUV 4:2:2 packed format and x:8:8:8 and 8:8:8:8 formats. Formats such as RGB5:5:5 and 5:6:5 do not have YUV versions. 0 = Pixel data is sent through the conversion logic (only applies to YUV formats) 1 = Pixel data is not sent through the YUV->RGB conversion logic.
18	**YUV Format:** This bit specifies the source YUV format for the YUV to RGB color conversion operation. This field is ignored when source data is RGB. 0 = ITU-R Recommendation BT.601 1 = ITU-R Recommendation BT.709
17:16	**YUV byte Order:** This field is used to select the byte order when using YUV 4:2:2 data formats. For other formats, this field is ignored. 00 = YUYV 01 = UYVY 10 = YVYU 11 = VYUY
15	**180° Display Rotation:** This mode causes the display plane to be rotated 180°. In addition to setting this bit, software must also set the base address to the lower right corner of the unrotated image and calculate the x, y offset as relative to the lower right corner. 0 = No rotation 1 = 180° rotation
14:11	Reserved
10	**Tiled Surface:** This bit indicates that the display C surface data is in tiled memory. The tile pitch is specified in bytes in the DSPCSTRIDE register. Only X tiling is supported for display surfaces. When this bit is set, it affects the hardware interpretation of the DSPCTILEOFF, DSPCLINOFF, and DSPCSURF registers. 0 = Display C surface uses linear memory 1 = Display C surface uses X-tiled memory
9:3	**Reserved:** Write as zero
2	**Display C Bottom:** This bit will force the display C plane to be on the bottom of the Z order. 0 = Display C Z order is determined by the other control bits 1 = Display C is forced to be on the bottom of the Z order.
1:0	Reserved

2.10.8.2 DSPCLINOFF —Display C/Sprite Linear Offset Register

Memory Offset Address: 72184h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the panning for the display surface. The surface base address is specified in the DSPCSURF register, and this register is used to describe an offset from that base address. Bit 10 of DSPCCNTR specifies whether the display A surface is in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the contents of this register are ignored.

This register can be written directly through software or by load register immediate command packets in the command stream.

This register is double buffered by VBLANK only. A change to this register will take effect on the next vblank following the write.

Bit	Descriptions
31:0	**Display C Offset:** This register provides the panning offset into the display C plane. This value is added to the surface address to get the graphics address of the first pixel to be displayed. This offset must be at least pixel aligned. This offset is the difference between the address of the upper left pixel to be displayed and the display surface address. When performing 180° rotation, this offset must be the difference between the last pixel of the last line of the display data in its unrotated orientation and the display surface address.

2.10.8.3 DSPCSTRIDE—Display C/Sprite Stride Register

Memory Offset Address: 72188h
Default: 00000000h
Normal Access: Read/Write Double Buffered

Bit	Descriptions
31:0	**Display C/Sprite Stride:** This is the stride for display C/Sprite in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. This register is updated through a command packet passed through the command stream or writes to this register. When it is desired to update both this and the start register, the stride register must be written first because the write to the start register is the trigger that causes the update of both registers on the next VBLANK event. When using tiled memory, the actual memory buffer stride is limited to a maximum of 16K bytes.

2.10.8.4 DSPCPOS—Sprite Position Register

Memory Offset Address: 7218Ch
Default: 00000000h
Normal Access: Read/Write Double Buffered

These registers specify the screen position and size of the sprite. This register is double buffered. The load register is transferred into the active register on the asserting edge of Vertical Blank for the pipe that the display is assigned. When using the sprite as a secondary display, this should be set to the entire display rectangle.

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	**SpriteY-Position:** These 12 bits specify the vertical position in lines of the sprite (upper left corner) relative to the beginning of the active video area. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.
15:12	Reserved: Write as zero
11:0	**Sprite X-Position:** These 12 bits specify the horizontal position in pixels of the sprite (upper left corner) relative the beginning of the active video area. When performing 180° rotation, this field specifies the horizontal position of the original lower right corner relative to the original end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.

2.10.8.5 DSPCSIZE—Sprite Height and Width Register

Memory Offset Address: 72190h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the height and width of the sprite in pixels and lines. The rectangle defined by the size and position should never exceed the boundaries of the display rectangle that the sprite is assigned to.

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	**Sprite Height:** This register field is used to specify the height of the sprite in lines. The value in the register is the height minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.
15:12	Reserved: Write as zero
11:0	**Sprite Width:** This register field is used to specify the width of the sprite in pixels. This does not have to be the same as the stride but should be less than or equal to the stride (converted to pixels). The value in the register is the width minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. The sprite width is limited to even values when YUV source pixel format is used, or Pixel Multiply is set to Line/Pixel doubling or Pixel doubling only (actual width, not the width minus one value).

2.10.8.6 DSPCKEYMINVAL—Sprite Color Key Min Value Register

Memory Offset Address: 72194h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite color key is enabled. The unused bits of the 5:5:5 or 5:6:5 formats must be filled with duplicates of the three or two MSBs of the pixel value.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	**Red Key Min Value:** Specifies the color key minimum value for the sprite red/Cr channel.
15:8	**Green Key Min Value:** Specifies the color key minimum value for the sprite green/Y channel.
7:0	**Blue Key Min Value:** Specifies the color key minimum value for the sprite blue/Cb channel.

2.10.8.7 DSPCKEYMSK—Sprite Color Key Mask Register

Memory Offset Address: 72198h
Default: 00000000h
Normal Access: Read/Write Double Buffered

Bit	Descriptions
31:3	Reserved: Write as zero
2	**Red Channel Enable:** Specifies the source color key enable for the red/Cr channel.
1	**Green Channel Enable:** Specifies the source color key enable for the green/Y channel.
0	**Blue Channel Enable:** Specifies the source color key enable for the blue/Cb channel

2.10.8.8 DSPCSURF—Display C Surface Address Register

Memory Offset Address:	7219Ch
Default:	00000000h
Normal Access:	Read/Write Double buffered

Writing to this register triggers the display plane flip. When it is desired to change multiple display C registers, this register should be written last as a write to this register will cause all new register values to take effect.

Bit	Descriptions
31:29	Reserved
28:12	**Display C Surface Base Address:** This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DSPCTILEOFF register. When the surface is in linear memory, panning is specified using a linear offset in the DSPCLINOFF register. This address must be 4K aligned. This register can be written directly through software or by command packets in the command stream. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. The value in this register is updated through the command streamer during synchronous flips. This address must be 128K aligned for linear memory.
11:0	Reserved

2.10.8.9 DSPCKEYMAXVAL—Sprite Color Key Max Value Register

Memory Offset Address:	721A0h
Default:	00000000h
Normal Access:	Read/Write Double Buffered

This register specifies the key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite color key is enabled.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	**Red Key Max Value:** Specifies the color key value for the sprite red/Cr channel.
15:8	**Green Key Max Value:** Specifies the color key value for the sprite green/Y channel.
7:0	**Blue Key Max Value:** Specifies the color key value for the sprite blue/Cb channel.

2.10.8.10 DSPCTILEOFF—Display C Tiled Offset Register

Memory Offset Address: 721A4h
Default: 00000000h
Normal Access: Read/Write Double buffered

This register specifies the panning for the display surface. The surface base address is specified in the DSPCSURF register, and this register is used to describe an offset from that base address. Bit 10 of DSPCCNTR specifies whether the display A surface is in linear or tiled memory. When the surface is in linear memory, the offset is specified in the DSPCLINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

This register can be written directly through software or by load register immediate command packets in the command stream.

This register is double buffered by VBLANK only. A change to this register will take effect on the next vblank following the write.

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	**Plane Start Y-Position:** These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation.
15:12	Reserved: Write as zero
11:0	**Plane Start X-Position:** These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation.

2.10.8.11 DSPCFLPQSTAT—Flip Queue Status Register

Memory Offset Address: 72200h
Default: 00000800h
Normal Access: Read/Write with Read-Only fields

Bit	Descriptions
31:16	Reserved: Write as zero (RO)
15:8	**Queue Free Entry Count (RO):** This value indicates the number of free entries in the queue at the time that the register was read. The total number of entries in the queue is the sum of the occupied entry count and the free entry count.
7:0	**Queue Occupied Entry Count (RO):** This value indicates the number of occupied entries in the queue at the time that the register was read. The total number of entries in the queue is the sum of the occupied entry count and the free entry count.

2.10.9 Display C Color Adjustment

These functions provide mechanisms for control of image colors generated from Display C sources. These functions are mainly intended for use for YUV color format sources. Adjustments are made before the YUV to RGB conversion. They take effect even when the Conversion Bypass bit is set. For display source input data in RGB format, software must set all the color correction registers to their default values (equivalent to a bypass mode). **These registers are not double buffered, they take effect immediately after loading.**

2.10.9.1 DCLRC0—Display C Color Correction 0 Register

Memory Address:	721D0h
Default Value:	01000000h
Normal Access:	RW
Size:	32 bits

Bit	Description
31:27	Reserved
26:18	**Contrast:** Contrast adjustment applies to YUV data. The Y channel is multiplied by the value contained in the register field. This signed fixed-point number is in 3i.6f format with the first 3 MSBs as the integer value and the last 6 LSBs as the fraction value. The allowed contrast value ranges from 0 to 7.53125 decimal. Bypassing Contrast, for YUV modes and for source data in RGB format, is accomplished by programming this field to a field value that represents 1.0 decimal or 001.000000 binary .
17:8	Reserved
7:0	**Brightness:** This field provides the brightness adjustment with a 8-bit 2's complement value ranging [-128, +127]. This value is added to the Y value after contrast multiply and before YUV to RGB conversion. A value of zero disables this adjustment affect. This 8-bit signed value provides half of the achievable brightness adjustment dynamic range. A full range brightness value would have a programmable range of [-255, +255]. Bypassing Brightness for YUV formats and for source data in RGB format is accomplished by programming this field to 0.

2.10.9.2 DCLRC1—Display C Color Correction 1 Register

Memory Address:	721D4h
Default Value:	00000080h
Normal Access:	RW
Size:	32 bits

The sum of the absolute value of SH_SIN and SH_COS must be limited to less than 8.

ABS(SH_SIN) + ABS(SH_COS) < 8

Bit	Description
31:27	Reserved
26:16	**Saturation and Hue SIN (SH_SIN):** This 11-bit signed fixed-point number is in 2's complement (s3i.7f) format with the MSB as the sign, next 3 MSBs as the integer value and the last 7 LSBs as the fraction value. This field can be used in two modes. When full range YUV data is operated on, this field contains the saturation value. When the range-limited YCbCr data is used, software should program this field with the product of the saturation multiplier value multiplied by the CbCr range scale factor (=128/112). Similar to the contrast field, there is no limit for saturation reduction – saturation = 0 means all pixels become the same value. However, increasing contrast can only be increased by a factor less than 8. For example, the largest contrast with value of 0x7.7F can bring input range [0, 32] to a full display color range of [0, 255]. Bypassing Hue, even for source data in RGB format, is accomplished by programming this field to 0.0.
15:10	Reserved
9:0	**Saturation and Hue COS (SH_COS):** This unsigned fixed-point number is in 3i.7f format with the first 3 MSBs be the integer value and the last 7 LSBs be the fraction value. This field can be used in two modes. When full range YUV data is operated on, this field contains the saturation value. When the range-limited YCbCr data is used, software should program this field with the product of the saturation multiplier value multiplied by the CbCr range scale factor (=128/112). Similar to the contrast field, there is no limit for saturation reduction – saturation = 0 means all pixels become the same value. However, increasing contrast can only be increased by a factor less than 8. For example, the largest contrast with value of 0x7.7F can bring input range [0, 32] to a full display color range of [0, 255]. Bypassing Saturation, even for source data in RGB format, is accomplished by programming this field to 1.0.

2.10.9.3 GAMC[5:0]— Display C Gamma Correction Registers

Memory Address:	721E0h – 721F7h GAMC5: 721E0h–721E3h GAMC4: 721E4h–721E7h GAMC3: 721E8h–721EBh GAMC2: 721ECh–721EFh GAMC1: 721F0h–721F3h GAMC0: 721F4h–721F7h

Default Value:	Linear R/G/B ramp
	GAMC5 = C0C0C0h
	GAMC4 = 808080h
	GAMC3 = 404040h
	GAMC2 = 202020h
	GAMC1 = 101010h
	GAMC0 = 080808h

Normal Access:	R/W
Size:	6 x 32 bits

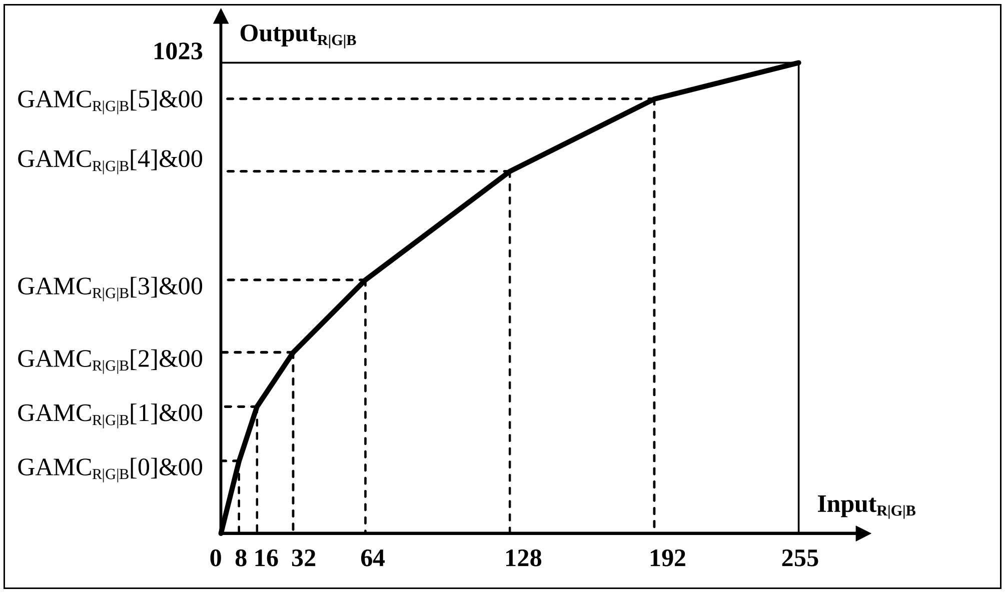
These registers are used to determine the characteristics of the gamma correction for the display C pixel data pre-blending. Additional gamma correction can be done in the display pipe gamma if desired. The pixels input to the gamma correction are 8-bit per channel pixels, and the output of the gamma correction is 10 bit per channel pixels. The gamma curve is represented by specifying a set of points along the curve. Each register has 32 bits, which are written to and read from together when accessed by the software. They are the six individual breakpoints on a logarithmically spaced color intensity space as shown in the following figure. The 8-bit values in the register are extended to 10 bit values in hardware by concatenating two zeroes onto the LSBs. The two end points (0 and 1023) have fixed values 0 and 1023, respectively. The appropriate Gamma breakpoint pairs (adjacent) are selected for each color component (Red, Green and Blue), and the output is interpolated between these two breakpoint values. The Gamma Correction registers (GAMC0 to GAMC5) are not double-buffered. They should be updated when the sprite is off. Otherwise, screen artifacts may show.

When the output from sprite is set in YUV format by programming CSC bypass, normally software should also bypass this gamma unit. However, since this gamma unit can also be viewed as a nonlinear transformation, it can be used, for whatever reason, in YUV output mode. In this case, the mapping of the three sets of piecewise linear map are as the following:

- Red to Cr (also called V)
- Green to Y
- Blue to Cb (also called U)

Bit	Description
31:24	Reserved
23:16	**Red (V/Cr):**
15:8	**Green (Y):**
7:0	**Blue (U/Cb):**

Figure 2-2. Programming of the Piecewise-linear Estimation of Gamma Correction Curve



2.10.10 Video Sprite A/B Control [DevCTG]

Two video sprites are provided for the display of YUV422 video content. These sprite planes provide windowing and keying functions as well as color space conversion from YUV to RGB. Each sprite plane is attached to only one of the pipes, Video Sprite A to pipe A and Video Sprite B to pipe B. Apart from the pipe assignments, the functionality is identical.

2.10.10.1 DVSACNTR—Video Sprite A Control Register [DevCTG]

Memory Offset Address: 72180h
Default: 00000000h
Normal Access: Read/Write Double Buffered

The active set of basic control registers will be updated on the VBlank after the “trigger” register (the Surface Address register or the Control register when plane enable bit transitioning from a zero to a one) is written – thus providing an atomic update of all display controls with the exception of the sprite gamma, size and position registers.

Bit	Descriptions
31	Video Sprite A Enable: This bit will enable or disable the sprite. When this bit is set, the plane will generate pixels for display to be combined by the blender for pipe A. When set to zero, memory fetches cease and display is blanked (from this plane) at the next VBLANK event from the pipe that this plane is assigned. Pipe A must be enabled to enable this plane. There is an override for the enable of this plane in the Pipe Configuration register. 1 = Enable 0 = Disable

Bit	Descriptions
30	Video Sprite A Gamma Enable: There are two gamma adjustments possible in the video sprite data path. This bit controls the gamma correction in the display pipe not the gamma control in this plane. It affects only the pixel data from this display plane. If display plane A is set to 8bpp this function is not available. Gamma correction logic that is contained in the video sprite is disabled by loading the default values into those registers. 0 = Sprite pixel data bypasses the display pipe gamma correction logic (default). 1 = Sprite pixel data is gamma corrected in the pipe gamma correction logic
29	Reserved: Write as zero
28	YUV Bypass Excess-512 Format Conversion: 1 = Enable excess-512 conversion 0 = Disable excess-512 conversion
27	Range Correction Disable: Setting this bit disables the YUV range correction logic. Normally the range compressed YUV is expanded to full range RGB, setting this bit will generate range compressed RGB. This bit should also be used if full range YUV source material is used. This bit has no effect on RGB source formats. 0 = Range correction enabled (default) 1 = No range correction.
26	Video Sprite Source Pixel Format (Testmode): This field selects the pixel format for the sprite. Before entering the blender, each source format is converted to 10 bits per pixel (details are described in the intermediate precision for the blender section of the Display Functions chapter). 0 = YUV 4:2:2 packed (YUYV) 1 = 32-bit BGRX (8:8:8:8) pixel format. Ignore alpha. This is a testmode only setting.
25:23	Reserved: Write as zeroes
22	Sprite Source Key Enable: This bit enables source color keying. Sprite pixel values that match (within range) the key will become transparent. Source key can not be enabled if destination key is enabled. 0 = Sprite source key is disabled (default) 1 = Sprite source key is enabled.
21:20	Pixel Multiply: This cause the display plane to duplicate lines and pixels sent to the assigned pipe. In the line/pixel doubling mode, the horizontal pixels are doubled and lines are sent twice. This is a method of scaling the source image by two (both H and V). 00 = No line/Pixel duplication 01 = Line/Pixel Doubling 10 = Reserved 11 = Pixel Doubling only
19	Color Conversion Disabled: This bit enables or disables the color conversion logic. Color conversion is intended to be used with the formats that support YUV format. 0 = Pixel data is sent through the conversion logic (only applies to YUV formats) 1 = Pixel data is not sent through the YUV->RGB conversion logic.

Bit	Descriptions
18	YUV Format: This bit specifies the source YUV format for the YUV to RGB color conversion operation. This field is ignored when source data is RGB. 0 = ITU-R Recommendation BT.601 1 = ITU-R Recommendation BT.709
17:16	YUV byte Order [DevCTG-B]: This field is used to select the byte order when using YUV 4:2:2 data formats. For other formats, this field is ignored. 00 = YUYV 01 = UYVY 10 = YVYU 11 = VYUY
15	180° Display Rotation: This mode causes the display plane to be rotated 180°. In addition to setting this bit, software must also set the base address to the lower right corner of the unrotated image and calculate the x, y offset as relative to the lower right corner. 0 = No rotation 1 = 180° rotation
14:11	Reserved
10	Tiled Surface: This bit indicates that the surface data is in tiled memory. The tile pitch is specified in bytes in the DVSASTRIDE register. Only X tiling is supported for display surfaces. When this bit is set, it affects the hardware interpretation of the DVSASTART and DVSASURFADDR registers. 0 = Linear memory 1 = Tiled memory
9:0	Reserved: Write as zero
2	Sprite Destination Key: This bit enables the destination key function. If the display A pixel for this location matches the key value in DVSAKEYVAL the sprite pixel is used, otherwise the pixel is passed throught the blender unmodified. Destination Key can not be enabled if source key is enabled. 0 = Destination Key is disabled 1 = Destination Key is enabled
1:0	Reserved: Write as zero

2.10.10.2 DVSALINOFF — Video Sprite A Linear Offset Register [DevCTG]

Memory Offset Address:	72184h
Default:	00000000h
Normal Access:	Read/Write Double Buffered

This register specifies the panning for the display surface in linear memory. The surface base address is specified in the DVSASURFADDR register, and this register is used to describe an offset from that base address. Bit 10 of DVSACNTR specifies whether the display surface is in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the contents of this register are ignored.

This register is double buffered by VBLANK only. A change to this register will take affect on the next vblank following the write.

Bit	Descriptions
31:0	Video Sprite A Offset: This register provides the panning offset into the sprite plane. This value is added to the surface address to get the graphics address of the first pixel to be displayed. This offset must be at least pixel aligned. This offset is the difference between the address of the upper left pixel to be displayed and the display surface address. When performing 180° rotation, this offset must be the difference between the last pixel of the last line of the display data in its unrotated orientation and the display surface address.

2.10.10.3 DVSASTRIDE— Video Sprite A Stride Register [DevCTG]

Memory Offset Address: 72188h
Default: 00000000h
Normal Access: Read/Write Double Buffered

Bit	Descriptions
31:0	Video Sprite A Stride: This is the stride for the sprite in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. This register is can be updated through a command packet passed through the command stream or writes to this register. When it is desired to update both this and the surface address register, the stride register must be written first because the write to the surface address register is the trigger that causes the update of both registers on the next VBLANK event. When using tiled memory, the actual memory buffer stride is limited to a maximum of 16K bytes.

2.10.10.4 DVSAPOS—Video Sprite A Position Register [DevCTG]

Memory Offset Address: 7218Ch
Default: 00000000h
Normal Access: Read/Write Double Buffered

These registers specify the screen position and size of the sprite. This register is double buffered. The load register is transferred into the active register on the asserting edge of Vertical Blank for the pipe that the display is assigned.   If the sprite is in 2x pixel multiply mode, the coordinates written here will be doubled by hardware. Software must take care that the sprite does not extend out of the display active area. ie. Xposition + Xsize =< Xsrcsize

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	Sprite Y-Position: These 12 bits specify the vertical position in lines of the sprite (upper left corner) relative to the beginning of the active video area. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.
15:12	Reserved: Write as zero

Bit	Descriptions
11:0	Sprite X-Position: These 12 bits specify the horizontal position in pixels of the sprite (upper left corner) relative the beginning of the active video area. When performing 180° rotation, this field specifies the horizontal position of the original lower right corner relative to the original end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.

2.10.10.5 DVSASIZE—Video Sprite A Height and Width Register [DevCTG]

Memory Offset Address:	72190h
Default:	00000000h
Normal Access:	Read/Write Double Buffered

This register specifies the height and width of the sprite in pixels and lines. The rectangle defined by the size and position should never exceed the boundaries of the display rectangle that the sprite is assigned to. If the sprite is in 2x pixel multiply mode, the size written here will be doubled by hardware. Software must take care that the sprite does not extend out of the display active area. ie. Xposition + Xsize =< Xsrcsize

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	Sprite Height: This register field is used to specify the height of the sprite in lines. The value in the register is the height minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. [DevCTG-B] The height must be even when sprite scaling is enabled and the pipe is interlaced. That means the programmed value must be odd.
15:12	Reserved: Write as zero
11:0	Sprite Width: This register field is used to specify the width of the sprite in pixels. This does not have to be the same as the stride but should be less than or equal to the stride (converted to pixels). The value in the register is the width minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. The sprite width is limited to even values when YUV source pixel format is used, or Pixel Multiply is set to Line/Pixel doubling or Pixel doubling only (actual width, not the width minus one value).

2.10.10.6 DVSAKEYVAL—Video Sprite A Color Key Value Register [DevCTG]

Memory Offset Address:	72194h
Default:	00000000h
Normal Access:	Read/Write Double Buffered

This register specifies the key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite color key is enabled. In source key mode with YUV data formats this value is the minimum value for the range compare.

In source key mode with RGB data formats or in destination key mode this value is the compare value.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	V Source Key Min Value/R Source/Dest Key Value: Specifies the color key (minimum) value for the sprite V channel source key or the Red channel source or destination key compare value.
15:8	Y Source Key Min Value/G Source/Dest Key Value: Specifies the color key (minimum) value for the sprite Y channel source key or the Green channel source or destination key compare value..
7:0	U Source Key Min Value/B Source/Dest Key Value: Specifies the color key (minimum) value for the sprite U channel source key or the Blue channel source or destination key compare value..

2.10.10.7 DVSAKEYMSK—Video Sprite A Color Key Mask Register [DevCTG]

Memory Offset Address: 72198h
Default: 00000000h
Normal Access: Read/Write Double Buffered

For source key this register specifies which channels to perform range checking on.

For destination key this register specifies the key mask to be used with the color value bits to determine if the display source data matches the key when enabled. A zero bit in the mask indicates that the corresponding bit match failure should be ignored when determining if the pixel matches.

Note that source key and destination key are mutually exclusive modes of operation, they can not be used simultaneously. For the function that is not enabled, the associated bits in this register should be programmed to zeroes.

Bit	Descriptions
31:27	Reserved: Write as zero
26	V/R Channel Source Key Enable: Specifies the source color key enable for the V/Red channel.
25	Y/G Channel Source Key Enable: Specifies the source color key enable for the Y/Green channel.
24	U/B Channel Source Key Enable: Specifies the source color key enable for the U/Blue channel
23:16	R mask Dest Key Value: Specifies the destination color key mask for the sprite R channel.
15:8	G mask Dest Key Value: Specifies the destination color key mask for the sprite G channel.
7:0	B mask Dest Key Value: Specifies the destination color key mask for the sprite B channel.

2.10.10.8 DVSASURF—Video Sprite A Surface Address Register [DevCTG]

Memory Offset Address: 7219Ch
Default: 00000000h

Normal Access: Read/Write Double buffered

Writing to this register triggers the display plane flip. When it is desired to change multiple sprite registers, this register should be written last as a write to this register will cause all new register values to take effect.

Bit	Descriptions
31:29	Reserved
28:12	Video Sprite A Surface Base Address: This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DVSASTART register. When the surface is in linear memory, panning is specified using a linear offset in the DVSASTART register. This address must be 4K aligned. This register can be written directly through software or by command packets in the command stream. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. The value in this register is updated through the command streamer during synchronous flips.
11:0	Reserved

2.10.10.9 DVSAKEYMAXVAL—Video Sprite A Color Key Max Value Register [DevCTG]

Memory Offset Address: 721A0h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite source color key is enabled.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	V Key Max Value: Specifies the color key value for the sprite V channel.
15:8	Y Key Max Value: Specifies the color key value for the sprite Y channel.
7:0	U Key Max Value: Specifies the color key value for the sprite U channel.

2.10.10.10 DVSATILEOFF—Video Sprite A Tiled Offset Register [DevCTG]

Memory Offset Address: 721A4h
Default: 00000000h
Normal Access: Read/Write Double buffered

This register specifies the panning for the display surface in tiled memory. The surface base address is specified in the DVSASURFADDR register, and this register is used to describe an offset from that base address. Bit 10 of DVSACNTR specifies whether the surface is in linear or tiled memory. When the surface is in linear memory, the offset is specified in the DVSALINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

This register is double buffered by VBLANK only. A change to this register will take affect on the next vblank following the write.

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	Plane Start Y-Position: These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation.
15:12	Reserved: Write as zero
11:0	Plane Start X-Position: These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation.

2.10.10.11 DVSASURFLIVE—Video Sprite A Live Surface Base Address Register [DevCTG-B]

Memory Offset Address:	721ACh
Default:	00000000h
Normal Access:	Read Only

Bit	Descriptions
31:0	Video Sprite A Live Surface Base Address. This gives the live value of the surface base address as being currently used for the plane.

2.10.10.12 Video Sprite A Gamma Correction Registers -- GAMC[5:0] [DevCTG]

Memory Address:	721E0h – 721F7h GAMC5: 721E0h–721E3h GAMC4: 721E4h–721E7h GAMC3: 721E8h–721EBh GAMC2: 721ECh–721EFh GAMC1: 721F0h–721F3h GAMC0: 721F4h–721F7h

Default Value:	Linear R/G/B ramp
	GAMC5 = C0C0C0h
	GAMC4 = 808080h
	GAMC3 = 404040h
	GAMC2 = 202020h
	GAMC1 = 101010h
	GAMC0 = 080808h

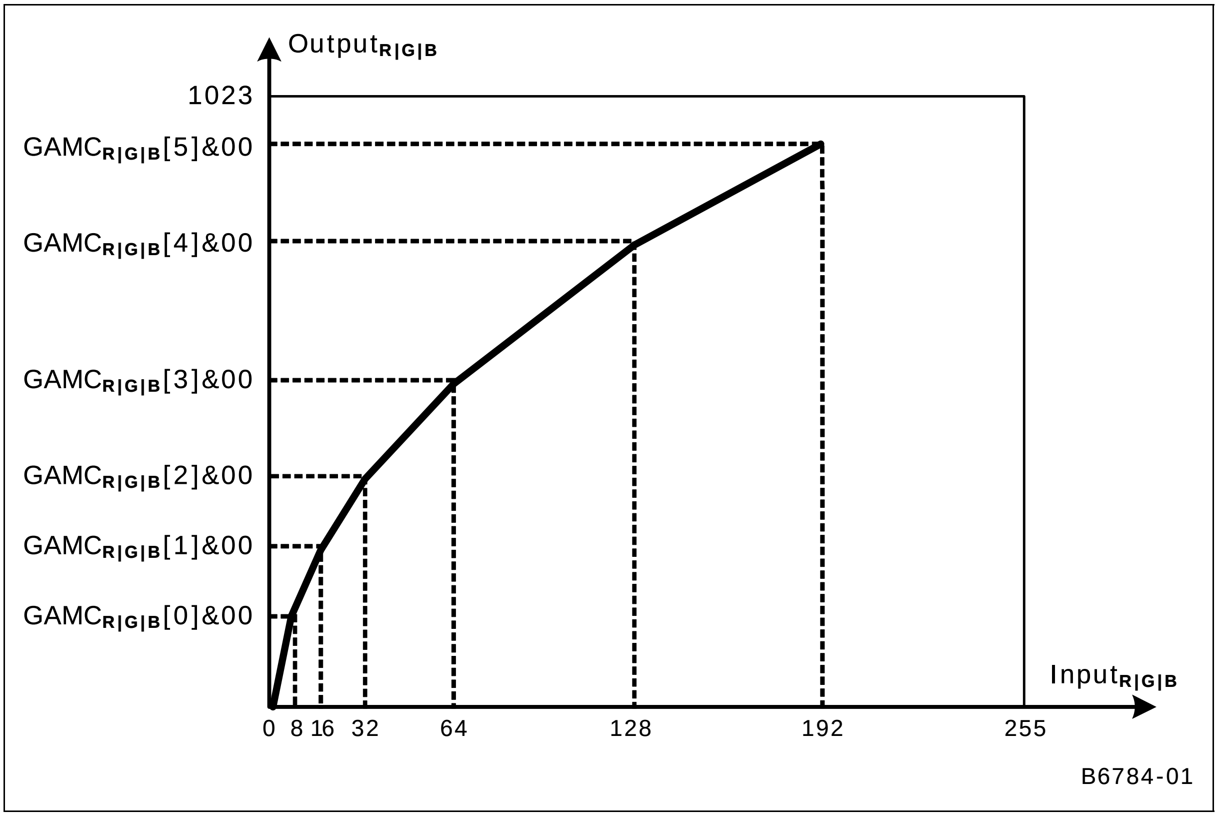
Normal Access:	R/W
Size:	6 x 32 bits

These registers are used to determine the characteristics of the gamma correction for the sprite pixel data pre-blending. Additional gamma correction can be done in the display pipe gamma if desired. The pixels input to the gamma correction are 8 bit per channel pixels, and the output of the gamma correction is 10 bit per channel pixels. The gamma curve is represented by specifying a set of points along the curve. Each register has 32 bits, which are written to and read from together when accessed by the software. They are the six individual breakpoints on a logarithmically spaced color intensity space as shown in the following figure. The 8 bit values in the register are extended to 10 bit values in hardware by concatenating two zeroes onto the LSBs. The two end points (0 and 1023) have fixed values 0 and 1023, respectively. The appropriate Gamma breakpoint pairs (adjacent) are selected for each color component (Red, Green and Blue), and the output is interpolated between these two breakpoint values. The Gamma Correction registers (GAMC0 to GAMC5) are not double-buffered. They should be updated when the sprite is off. Otherwise, screen artifacts may show.

When the output from sprite is set in YUV format by programming CSC bypass, the Gamma Unit will also be bypassed.

Bit	Description
31:24	Reserved:
23:16	Red:
15:8	Green:
7:0	Blue:

Figure 2-3. Programming of the Piecewise-linear Estimation of Gamma Correction Curve



2.10.10.13 DVSASCALE—Video Sprite A Scaler Control [DevCTG-B]

Memory Offset Address:	72204h
Default:	00000000h
Normal Access:	Read/Write Double Buffered

Double buffered to update on the vertical blank after a surface address write.

The sprite height and width values from DVSASIZE give the destination (output to pipe) size of the sprite. Source and destination sizes must be 3x3 (3x6 when interlacing) or greater when scaling is enabled. The scaling function will scale the sprite image from the source size to the destination size. Upscaling of any amount is allowed. Downscaling up to 16X (source/destination) is allowed. Downscaling of 16x while interlacing is only allowed together with tiled memory, not with linear memory. Downscaling greater than 2X will involve decimation. Downscaling increases memory bandwidth requirements. Horizontal downscaling limits the maximum dot clock allowed as percent of cdclk.

Rules to calculate the allowed dot clock:

Start with maximum dot clock 90% of cdclk. (There is a separate requirement that planes using 64bpp formats can not be enabled with dot clock >80% of cdclk when sprite is enabled on the same pipe)

Subtract 10% more per horizontal decimation step (decimation steps at 2x, 4x, 8x, and 16x downscale).

Subtract 10% more if sprite is using the RGB data format.

Subtract 10% more if sprite scaling is enabled on the other pipe.

Then divide that by downscale amount within each decimation step.

The result is the maximum allowed dot clock as percent of cdclk frequency.

Example:

Scale factor	Decimation amount	YUV single pipe dot clock %	YUV dual pipe dot clock %	RGB single pipe dot clock %	RGB dual pipe dot clock %	Comment
1	1	90	80	80	70	No scaling
1.5	1	60	53	53	46	
1.99	1	45	40	40	35	Max downscale before decimation starts
2	2	80	70	70	60	
3	2	53	46	46	40	
3.99	2	40	35	35	30	
4	4	70	60	60	50	
6	4	46	40	40	33	
7.99	4	35	30	30	25	
8	8	60	50	50	40	
12	8	40	33	33	26	
15.99	8	30	25	25	20	Worst case dot clock
16	16	50	40	40	30	Max downscaling allowed

Bit	Descriptions
31	Scaling Enable: Enables the scaling function. Source width can be no more than 4k bytes. For best picture quality, disable when scaling is not required (1:1 scale factor). 0: Off 1: On
30:29	Filter Control: Filter selection 00: Medium (Default) 01: Sharp 10: Soft 11: Reserved
28	Even/Odd Field Offset: Select the vertical offset of the filtered data. Software is responsible for updating this to match the surface data. 0: Vertical initial phase of 0 1: Vertical initial phase of 0.5
27	Even/Odd field Enable: Enable adjustment of the vertical offset of the filtered data. 0: Off (Vertical initial phase is 1/2 the scale factor) 1: On (Vertical initial phase is selected by the Even/Off Field Offset bit)
26:16	Source Width: The horizontal size of the source image to be scaled in pixels. Max number of pixels is 2048; minimum is 3. The value programmed is one less than the number of pixels. Source width can be no more than 4k bytes, counting from a 64 byte alignment.
15:11	Reserved: MBZ
10:0	Source Height: The vertical size of the source image to be scaled in lines. If the source is a field, this is the number of lines in the field. Max number of lines is 2048; minimum is 3. The value programmed is one less than the number of lines. [DevCTG-B] The height must be even when sprite scaling is enabled and the pipe is interlaced. That means the programmed value must be odd.

2.10.10.14 DVSBCNTR—Video Sprite B Control Register [DevCTG]

Memory Offset Address: 73180h
Default: 00000000h
Normal Access: Read/Write Double Buffered

The active set of basic control registers will be updated on the VBlank after the “trigger” register (the Surface Address register or the Control register when plane enable bit transitioning from a zero to a one) is written – thus providing an atomic update of all display controls with the exception of the sprite gamma, size and position registers.

Bit	Descriptions
31	Video Sprite B Enable: This bit will enable or disable the sprite. When this bit is set, the plane will generate pixels for display to be combined by the blender for pipe B. When set to zero, memory fetches cease and display is blanked (from this plane) at the next VBLANK event from the pipe that this plane is assigned. Pipe B must be enabled to enable this plane. There is an override for the enable of this plane in the Pipe Configuration register. 1 = Enable 0 = Disable
30	Video Sprite B Gamma Enable: There are two gamma adjustments possible in the video sprite data path. This bit controls the gamma correction in the display pipe not the gamma control in this plane. It affects only the pixel data from this display plane. If display plane B is set to 8bpp this function is not available. Gamma correction logic that is contained in the video sprite is disabled by loading the default values into those registers. 0 = Sprite pixel data bypasses the display pipe gamma correction logic (default). 1 = Sprite pixel data is gamma corrected in the pipe gamma correction logic
29	Reserved: Write as zero
28	YUV Bypass Excess-512 Format Conversion: 1 = Enable excess-512 conversion 0 = Disable excess-512 conversion
27	Range Correction Disable: Setting this bit disables the YUV range correction logic. Normally the range compressed YUV is expanded to full range RGB, setting this bit will generate range compressed RGB. This bit should also be used if full range YUV source material is used. This bit has no effect on RGB source formats. 0 = Range correction enabled (default) 1 = No range correction.
26	Video Sprite Source Pixel Format (Testmode): This field selects the pixel format for the sprite. Before entering the blender, each source format is converted to 10 bits per pixel (details are described in the intermediate precision for the blender section of the Display Functions chapter). 0 = YUV 4:2:2 packed (YUYV) 1 = 32-bit BGRX (8:8:8:8) pixel format. Ignore alpha. This is a testmode only setting.
25:23	Reserved: Write as zeroes

Bit	Descriptions
22	Sprite Source Key Enable: This bit enables source color keying. Sprite pixel values that match (within range) the key will become transparent. Source key can not be enabled if destination key is enabled. 0 = Sprite source key is disabled (default) 1 = Sprite source key is enabled.
21:20	Pixel Multiply: This cause the display plane to duplicate lines and pixels sent to the assigned pipe. In the line/pixel doubling mode, the horizontal pixels are doubled and lines are sent twice. This is a method of scaling the source image by two (both H and V). 00 = No line/Pixel duplication 01 = Line/Pixel Doubling 10 = Reserved 11 = Pixel Doubling only
19	Color Conversion Disabled: This bit enables or disables the color conversion logic. Color conversion is intended to be used with the formats that support YUV format. 0 = Pixel data is sent through the conversion logic (only applies to YUV formats) 1 = Pixel data is not sent through the YUV->RGB conversion logic.
18	YUV Format: This bit specifies the source YUV format for the YUV to RGB color conversion operation. This field is ignored when source data is RGB. 0 = ITU-R Recommendation BT.601 1 = ITU-R Recommendation BT.709
17:16	YUV byte Order [DevCTG-B]: This field is used to select the byte order when using YUV 4:2:2 data formats. For other formats, this field is ignored. 00 = YUYV 01 = UYVY 10 = YVYU 11 = VYUY
15	180° Display Rotation: This mode causes the display plane to be rotated 180°. In addition to setting this bit, software must also set the base address to the lower right corner of the unrotated image and calculate the x, y offset as relative to the lower right corner. 0 = No rotation 1 = 180° rotation
14:11	Reserved
10	Tiled Surface: This bit indicates that the surface data is in tiled memory. The tile pitch is specified in bytes in the DVSBSTRIDE register. Only X tiling is supported for display surfaces. When this bit is set, it affects the hardware interpretation of the DVSBSTART and DVSBSURFADDR registers. 0 = Linear memory 1 = Tiled memory
9:0	Reserved: Write as zero

Bit	Descriptions
2	Sprite Destination Key: This bit enables the destination key function. If the display B pixel for this location matches the key value in DVSBKEYVAL the sprite pixel is used, otherwise the pixel is passed throught the blender unmodified. Destination Key can not be enabled if source key is enabled. 0 = Destination Key is disabled 1 = Destination Key is enabled
1:0	Reserved: Write as zero

2.10.10.15 DVSBLINOFF — Video Sprite B Linear Offset Register [DevCTG]

Memory Offset Address: 73184h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the panning for the display surface in linear memory. The surface base address is specified in the DVSBSURFADDR register, and this register is used to describe an offset from that base address. Bit 10 of DVSBCNTR specifies whether the display surface is in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the contents of this register are ignored.

This register is double buffered by VBLANK only. A change to this register will take affect on the next vblank following the write.

Bit	Descriptions
31:0	Video Sprite B Offset: This register provides the panning offset into the sprite plane. This value is added to the surface address to get the graphics address of the first pixel to be displayed. This offset must be at least pixel aligned. This offset is the difference between the address of the upper left pixel to be displayed and the display surface address. When performing 180° rotation, this offset must be the difference between the last pixel of the last line of the display data in its unrotated orientation and the display surface address.

2.10.10.16 DVSBSTRIDE— Video Sprite B Stride Register [DevCTG]

Memory Offset Address: 73188h
Default: 00000000h
Normal Access: Read/Write Double Buffered

Bit	Descriptions
31:0	Video Sprite B Stride: This is the stride for the sprite in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. This register is can be updated through a command packet passed through the command stream or writes to this register. When it is desired to update both this and the surface address register, the stride register must be written first because the write to the surface address register is the trigger that causes the update of both registers on the next VBLANK event. When using tiled memory, the actual memory buffer stride is limited to a maximum of 16K bytes.

2.10.10.17 DVSBPOS—Video Sprite B Position Register [DevCTG]

Memory Offset Address: 7318Ch
Default: 00000000h
Normal Access: Read/Write Double Buffered

These registers specify the screen position and size of the sprite. This register is double buffered. The load register is transferred into the active register on the asserting edge of Vertical Blank for the pipe that the display is assigned. If the sprite is in 2x pixel multiply mode, the coordinates written here will be doubled by hardware. Software must take care that the sprite does not extend out of the display active area. ie. Xposition + Xsize =< Xsrcsize

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	Sprite Y-Position: These 12 bits specify the vertical position in lines of the sprite (upper left corner) relative to the beginning of the active video area. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.
15:12	Reserved: Write as zero
11:0	Sprite X-Position: These 12 bits specify the horizontal position in pixels of the sprite (upper left corner) relative the beginning of the active video area. When performing 180° rotation, this field specifies the horizontal position of the original lower right corner relative to the original end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.

2.10.10.18 DVSBSIZE—Video Sprite B Height and Width Register [DevCTG]

Memory Offset Address: 73190h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the height and width of the sprite in pixels and lines. The rectangle defined by the size and position should never exceed the boundaries of the display rectangle that the sprite is assigned to. If the sprite is in 2x pixel multiply mode, the size written here will be doubled by hardware. Software must take care that the sprite does not extend out of the display active area. ie. Xposition + Xsize =< Xsrcsize

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	Sprite Height: This register field is used to specify the height of the sprite in lines. The value in the register is the height minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. [DevCTG-B] The height must be even when sprite scaling is enabled and the pipe is interlaced. That means the programmed value must be odd.
15:12	Reserved: Write as zero

Bit	Descriptions
11:0	Sprite Width: This register field is used to specify the width of the sprite in pixels. This does not have to be the same as the stride but should be less than or equal to the stride (converted to pixels). The value in the register is the width minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. The sprite width is limited to even values when YUV source pixel format is used, or Pixel Multiply is set to Line/Pixel doubling or Pixel doubling only (actual width, not the width minus one value).

2.10.10.19 DVSBKEYVAL—Video Sprite B Color Key Value Register [DevCTG]

Memory Offset Address: 73194h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite color key is enabled. In source key mode with YUV data formats this value is the minimum value for the range compare. In source key mode with RGB data formats or in destination key mode this value is the compare value.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	V Source Key Min Value/R Source/Dest Key Value: Specifies the color key (minimum) value for the sprite V channel source key or the Red channel source or destination key compare value.
15:8	Y Source Key Min Value/G Source/Dest Key Value: Specifies the color key (minimum) value for the sprite Y channel source key or the Green channel source or destination key compare value..
7:0	U Source Key Min Value/B Source/Dest Key Value: Specifies the color key (minimum) value for the sprite U channel source key or the Blue channel source or destination key compare value..

2.10.10.20 DVSBKEYMSK—Video Sprite B Color Key Mask Register [DevCTG]

Memory Offset Address: 73198h
Default: 00000000h
Normal Access: Read/Write Double Buffered

For source key this register specifies which channels to perform range checking on.

For destination key this register specifies the key mask to be used with the color value bits to determine if the display source data matches the key when enabled. A zero bit in the mask indicates that the corresponding bit match failure should be ignored when determining if the pixel matches.

Note that source key and destination key are mutually exclusive modes of operation, they can not be used simultaneously. For the function that is not enabled, the associated bits in this register should be programmed to zeroes.

Bit	Descriptions
31:27	Reserved: Write as zero
26	V/R Channel Source Key Enable: Specifies the source color key enable for the V/Red channel.
25	Y/G Channel Source Key Enable: Specifies the source color key enable for the Y/Green channel.
24	U/B Channel Source Key Enable: Specifies the source color key enable for the U/Blue channel
23:16	R mask Dest Key Value: Specifies the destination color key mask for the sprite R channel.
15:8	G mask Dest Key Value: Specifies the destination color key mask for the sprite G channel.
7:0	B mask Dest Key Value: Specifies the destination color key mask for the sprite B channel.

2.10.10.21 DVSBSURF—Video Sprite B Surface Address Register [DevCTG]

Memory Offset Address: 7319Ch
Default: 00000000h
Normal Access: Read/Write Double buffered

Writing to this register triggers the display plane flip. When it is desired to change multiple sprite registers, this register should be written last as a write to this register will cause all new register values to take effect.

Bit	Descriptions
31:29	Reserved
28:12	Video Sprite B Surface Base Address: This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DVSBSTART register. When the surface is in linear memory, panning is specified using a linear offset in the DVSBSTART register. This address must be 4K aligned. This register can be written directly through software or by command packets in the command stream. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. The value in this register is updated through the command streamer during synchronous flips.
11:0	Reserved

2.10.10.22 DVSBKEYMAXVAL—Video Sprite B Color Key Max Value Register [DevCTG]

Memory Offset Address: 731A0h
Default: 00000000h
Normal Access: Read/Write Double Buffered

This register specifies the key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite source color key is enabled.

Bit	Descriptions
31:24	Reserved: Write as zero
23:16	V Key Max Value: Specifies the color key value for the sprite V channel.
15:8	Y Key Max Value: Specifies the color key value for the sprite Y channel.
7:0	U Key Max Value: Specifies the color key value for the sprite U channel.

2.10.10.23 DVSBTILEOFF—Video Sprite B Tiled Offset Register [DevCTG]

Memory Offset Address: 731A4h
Default: 00000000h
Normal Access: Read/Write Double buffered

This register specifies the panning for the display surface in tiled memory. The surface base address is specified in the DVSBSURFADDR register, and this register is used to describe an offset from that base address. Bit 10 of DVSBCNTR specifies whether the surface is in linear or tiled memory. When the surface is in linear memory, the offset is specified in the DVSBLINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

This register is double buffered by VBLANK only. A change to this register will take affect on the next vblank following the write.

Bit	Descriptions
31:28	Reserved: Write as zero
27:16	Plane Start Y-Position: These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation.
15:12	Reserved: Write as zero
11:0	Plane Start X-Position: These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation.

2.10.10.24 DVSBSURFLIVE—Video Sprite B Live Surface Base Address Register [DevCTG-B]

Memory Offset Address:	731ACh
Default:	00000000h
Normal Access:	Read Only

Bit	Descriptions
31:0	Video Sprite B Live Surface Base Address. This gives the live value of the surface base address as being currently used for the plane.

2.10.10.25 Video Sprite B Gamma Correction Registers -- GAMC[5:0] [DevCTG]

Memory Address:	731E0h – 731F7h GAMC5: 731E0h–731E3h GAMC4: 731E4h–731E7h GAMC3: 731E8h–731EBh GAMC2: 731ECh–731EFh GAMC1: 731F0h–731F3h GAMC0: 731F4h–731F7h

Default Value:	Linear R/G/B ramp
	GAMC5 = C0C0C0h
	GAMC4 = 808080h
	GAMC3 = 404040h
	GAMC2 = 202020h
	GAMC1 = 101010h
	GAMC0 = 080808h

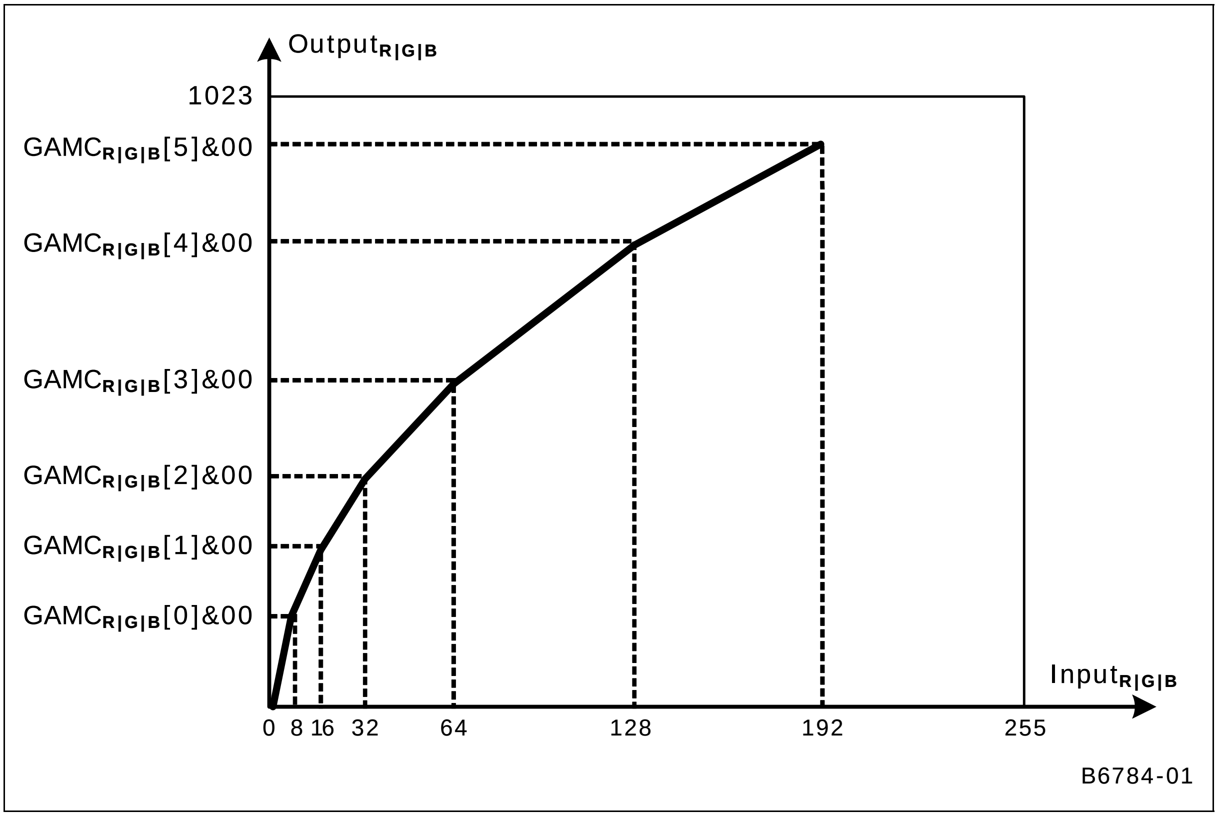
Normal Access:	R/W
Size:	6 x 32 bits

These registers are used to determine the characteristics of the gamma correction for the sprite pixel data pre-blending. Additional gamma correction can be done in the display pipe gamma if desired. The pixels input to the gamma correction are 8 bit per channel pixels, and the output of the gamma correction is 10 bit per channel pixels. The gamma curve is represented by specifying a set of points along the curve. Each register has 32 bits, which are written to and read from together when accessed by the software. They are the six individual breakpoints on a logarithmically spaced color intensity space as shown in the following figure. The 8 bit values in the register are extended to 10 bit values in hardware by concatenating two zeroes onto the LSBs. The two end points (0 and 1023) have fixed values 0 and 1023, respectively. The appropriate Gamma breakpoint pairs (adjacent) are selected for each color component (Red, Green and Blue), and the output is interpolated between these two breakpoint values. The Gamma Correction registers (GAMC0 to GAMC5) are not double-buffered. They should be updated when the sprite is off. Otherwise, screen artifacts may show.

When the output from sprite is set in YUV format by programming CSC bypass, the Gamma Unit will also be bypassed.

Bit	Description
31:24	Reserved:
23:16	Red:
15:8	Green:
7:0	Blue:

Figure 2-4. Programming of the Piecewise-linear Estimation of Gamma Correction Curve



2.10.10.26 DVSBSCALE—Video Sprite B Scaler Control [DevCTG-B]

Memory Offset Address: 73204h
Default: 00000000h
Normal Access: Read/Write Double Buffered

Double buffered to update on the vertical blank after a surface address write.

The sprite height and width values from DVSBSIZE give the destination (output to pipe) size of the sprite. Source and destination sizes must be 3x3 (3x6 when interlacing) or greater when scaling is enabled. The scaling function will scale the sprite image from the source size to the destination size. Upscaling of any amount is allowed. Downscaling up to 16X (source/destination) is allowed. Downscaling of 16x while interlacing is only allowed together with tiled memory, not with linear memory. Downscaling greater than 2X will involve decimation. Downscaling increases memory bandwidth requirements. Horizontal downscaling limits the maximum dot clock allowed as percent of cdclk. See DVSASCALE for rules to calculate the allowed dot clock.

Bit	Descriptions
31	Scaling Enable: Enables the scaling function. Source width can be no more than 4k bytes. For best picture quality, disable when scaling is not required (1:1 scale factor). 0: Off 1: On
30:29	Filter Control: Filter selection 00: Medium (Default) 01: Sharp 10: Soft 11: Reserved
28	Even/Odd Field Offset: Select the vertical offset of the filtered data. Software is responsible for updating this to match the surface data. 0: Vertical initial phase of 0 1: Vertical initial phase of 0.5
27	Even/Odd field Enable: Enable adjustment of the vertical offset of the filtered data. 0: Off (Vertical initial phase is 1/2 the scale factor) 1: On (Vertical initial phase is selected by the Even/Off Field Offset bit)
26:16	Source Width: The horizontal size of the source image to be scaled in pixels. Max number of pixels is 2048; minimum is 3. The value programmed is one less than the number of pixels. Source width can be no more than 4k bytes, counting from a 64 byte alignment.
15:11	Reserved: MBZ
10:0	Source Height: The vertical size of the source image to be scaled in lines. If the source is a field, this is the number of lines in the field. Max number of lines is 2048; minimum is 3. The value programmed is one less than the number of lines. [DevCTG-B] The height must be even when sprite scaling is enabled and the pipe is interlaced. That means the programmed value must be odd.

2.10.11 Performance Counters

The performance counter hardware provides a method for software to monitor memory latency and hardware FIFO statistics for the purpose of optimizing memory accesses for the display planes.

2.10.11.1 PCSRC—Performance Counter Source Register

Memory Address Offset:	73000h
Default Value:	00000000h
Normal Access:	RW
Size:	32 bits

Bit	Description
31	**Performance Counter Enable:** This bit enables the performance counter. 0 = Performance counter is disabled 1 = Performance counter is enabled.
30	**Interrupt Enable:** This bit enables/disables an interrupt when the threshold value programmed in the Performance Counter Source register matches the value of the performance counter. 0 = Interrupt disabled 1 = Interrupt enabled
29	**Reset Counter:** This bit indicates when the counter will be reset. 1 = Reset after each frame, summing all events in the frame 0 = Reset after each event within the frame
28	**Max Or Min:** This bit tells whether the stored counter value for an event is the maximum or the minimum value. The previous value is used to do the compare. 0 = Stored value is the maximum latency 1 = Stored value is the minimum latency

Bit	Description
27:23	**Source For Performance Counter:** These bits indicate the source for the performance counter. 00000 = Overlay Register Request Latency 00001 = VGA Font Request Latency 00010 = VGA Character Request Latency 00011 = Display A FIFO Status 00100 = Display B FIFO Status 00101 = Display C FIFO Status 00110 = Cursor A FIFO Status 00111 = Cursor B FIFO Status 01000 = Display Steamer A TLB Latency 01001 = Display Streamer B TLB Latency 01010 = Display Streamer C TLB Latency 01011 = Cursor Streamer A TLB Latency 01100 = Cursor Streamer B TLB Latency 01101 = Overlay Streamer TLB Latency 01110 = Display Steamer A Request Latency 01111 = Display Streamer B Request Latency 10000 = Display Streamer C Request Latency 10001 = Cursor Streamer A Request Latency 10010 = Cursor Streamer B Request Latency 10011 = Overlay Streamer Request Latency 10100 = Display A Command Request Latency 10101 = Display B Command Request Latency 10110 = Display C Command Request Latency 10111 = Cursor A Command Request Latency 11000 = Cursor B Command Request Latency 11001 = Overlay Command Request Latency 11010 = Reserved 11011 = Reserved
22:16	Reserved: Write as zero.
15:0	**Performance Counter Threshold Value:** This value is used to compare against the performance counter. If the performance counter matches this value, an interrupt is generated if the interrupt bit is enabled. When the source selected is DDB FIFO status, the threshold value is used to program the value needed to monitor in the DDB FIFO. No interrupt is generated in this condition.

2.10.11.2 PCSTAT—Performance Counter Status Register

Memory Address Offset:	73004h
Default Value:	00000000h
Normal Access:	RO
Size:	32 bits

Bit	Description
31	**Overflow:** This bit indicates weather the 16-bit counter overflowed or not. 0 = Counter is valid 1 = Counter is invalid since it overflowed
30	**Reset Counter:** This bit indicates when the counter will be reset. 1 = Reset after each frame, sum of all event in the frame 0 = Reset after each event within the frame
29	**Max Or Min:** This bit tells whether the stored counter value for an event is the maximum or the minimum value of the previous event. 0 = Stored value is the maximum latency 1 = Stored value is the minimum latency
28:24	**Source For Performance Counter:** These bits indicate the source for the performance counter. 00000 = Overlay Register Request Latency 00001 = VGA Font Request Latency 00010 = VGA Character Request Latency 00011 = Display A FIFO Status 00100 = Display B FIFO Status 00101 = Display C FIFO Status 00110 = Cursor A FIFO Status 00111 = Cursor B FIFO Status 01000 = Display Steamer A TLB Latency 01001 = Display Streamer B TLB Latency 01010 = Display Streamer C TLB Latency 01011 = Cursor Streamer A TLB Latency 01100 = Cursor Streamer B TLB Latency 01101 = Overlay Streamer TLB Latency 01110 = Display Steamer A Request Latency 01111 = Display Streamer B Request Latency 10000 = Display Streamer C Request Latency 10001 = Cursor Streamer A Request Latency 10010 = Cursor Streamer B Request Latency 10011 = Overlay Streamer Request Latency 10100 = Display A Command Request Latency 10101 = Display B Command Request Latency 10110 = Display C Command Request Latency 10111 = Cursor A Command Request Latency 11000 = Cursor B Command Request Latency 11001 = Overlay Command Request Latency 11010 = Reserved 11011 = Reserved
23:16	Reserved: Write as zero.
15:0	**Performance Counter Value:** This is the value of the performance counter for the source indicated in the source field.

3 Overlay Registers

3.1 Introduction and Register Summary

This chapter contains the register descriptions for the overlay portion of a family of integrated graphics devices. These registers do vary by device within the family of devices so special attention needs to be paid to which devices use which registers and register fields.

Different devices within the family may add, modify, or delete registers or register fields relative to another device in the same family based on the supported functions of that device. Additional information on the use and programming of these registers can be found in the display chapter. This document covers both desktop and mobile products.

The following table contains the sections break down where the register information is contained within this chapter:

Address Range	Description
30000h–3FFFFh	Overlay Registers

3.1.1 Terminology

Description	Software Use	Should be implemented as
Reserved write as zero.	Software must always write a zero to these bits. This allows new features to be added using these bits that will be disabled when using old software and as the default case.	These are read-only bits that always read as zeros or r/w bits that are default to zero.
Reserved write as one.	Software must always write a one to these bits. This allows new features to be added using these bits that will be disabled when using old software and as the default case.	
Reserved for BIOS Do not change	Driver access to these bits must read these bits that have been set through an initialization operation before writing this register so that the bits can remain unchanged.	According to each specific bit
Reserved for Video BIOS	These register bits will be used only by video BIOS and drivers should not change them.	These are read/write bits that have no hardware function. They are intended for use by the video BIOS for storage.
Reserved for Compatibility	For functions that are no longer needed these bits had old use, but now does nothing. New software should use the new method.	Read/write bits that have no functions.

Description	Software Use	Should be implemented as
Use for compatibility only	Under specific conditions, these bits functions as in the old part, new software should use the new method.	According to each specific bit
Read-Only	This bit is read-only. The read value is determined by hardware. Writes to this bit have no effect.	According to each specific bit. The bit value is determined by hardware and not affected by register writes to the actual bit.
Reserved read-only	Don't assume a value for these bits. Writes have no effect.	These bits should read as zero.
Reserved read-only write as zero	Don't assume a value for these bits, always write a zero.	These bits should read as zero.
Read/Clear	This bit can be read and writes to it with a one cause the bit to clear.	Hardware events cause the bit to be set and the bit will be cleared on a write operation where the corresponding bit has a one for a value.
Read/Write	This bit can be read or written.	
Double Buffered	Write when desired	Takes effect only after a particular event such as a VBLANK.

3.1.2 Register Protection in a Trusted Environment

With the exception of Type 5 registers, the register protection defined via the trusted type only has an affect if the trusted environment has been established and the current state of the trusted register lock is "hard lock". Access restrictions apply to all un-trusted clients that may include the CPU, command stream, and real mode access. Additional conditions apply to each register or register field that affect locking.

There are five basic types of registers that are listed in the following table.

Register Type	Access Restrictions
Type 1	Normal access from either BAR or trusted or un-trusted rings. The trusted BAR access only exists when the trusted environment has been established. Normal access from trusted and untrusted rings. Registers that don't indicate the trusted type are Type 1.
Type 2	Normal read or write access from trusted BAR or trusted rings. Normal read operations from un-trusted BAR, and write operations to specific fields/field combinations not allowed, have a different effect, or have no impact. This is based on a set of conditions being met. The details of the fields and their behaviors are listed in the register descriptions.
Type 3	Normal access from trusted BAR and trusted rings. Read-only from un-trusted BAR. Read side effects still occur on the read operation.
Type 4	Normal access from trusted BAR and trusted rings and not accessible from un-trusted BAR when conditions are met. Untrusted access reads result in a zero value, writes complete with no effect. This register reverts to a type 1 access when the lock bits are not in lock state.

Register Type	Access Restrictions
Type 5	Access from trusted BAR or trusted rings only. Never accessible from the untrusted aperture or command ring. Untrusted access reads result in a zero value, writes complete with no effect. When the trusted environment is not established, there is no access allowed to these registers. Note that these register type access restrictions exist regardless of the state of the lock bits.

3.2 Overlay Register Definition

3.2.1 Introduction to Overlay Registers

The registers detailed in this chapter are used across products. However, slight changes may be present in some registers (i.e., for features added or removed), or, alternatively, some registers may be removed entirely. This section contains the description of the registers that control the Overlay hardware in the graphics controller. The registers include:

- Overlay Control registers
- Overlay Filter Coefficient registers
- Overlay Gamma and color control registers
- Overlay Scaling registers
- Overlay Source and destination sizes
- Overlay keying values

One hardware overlay is implemented in the graphics device. Note that most of the Overlay registers are indirectly written by first setting up a buffer in memory and then instructing the graphics controller to update the on-chip registers from this buffer. Software can invoke the update process by either writing to the OVADD (with restrictions) register or by issuing an Overlay Flip instruction (MI_OVERLAY_FLIP).

The Overlay registers can be used for overlay gamma correction and color adjustment. They are read/written directly by MMIO (memory mapped I/O) without going through the buffer in memory.

The register categories are listed in the Overlay Register Categories table.

Figure 3-1. Overlay Register Memory Map

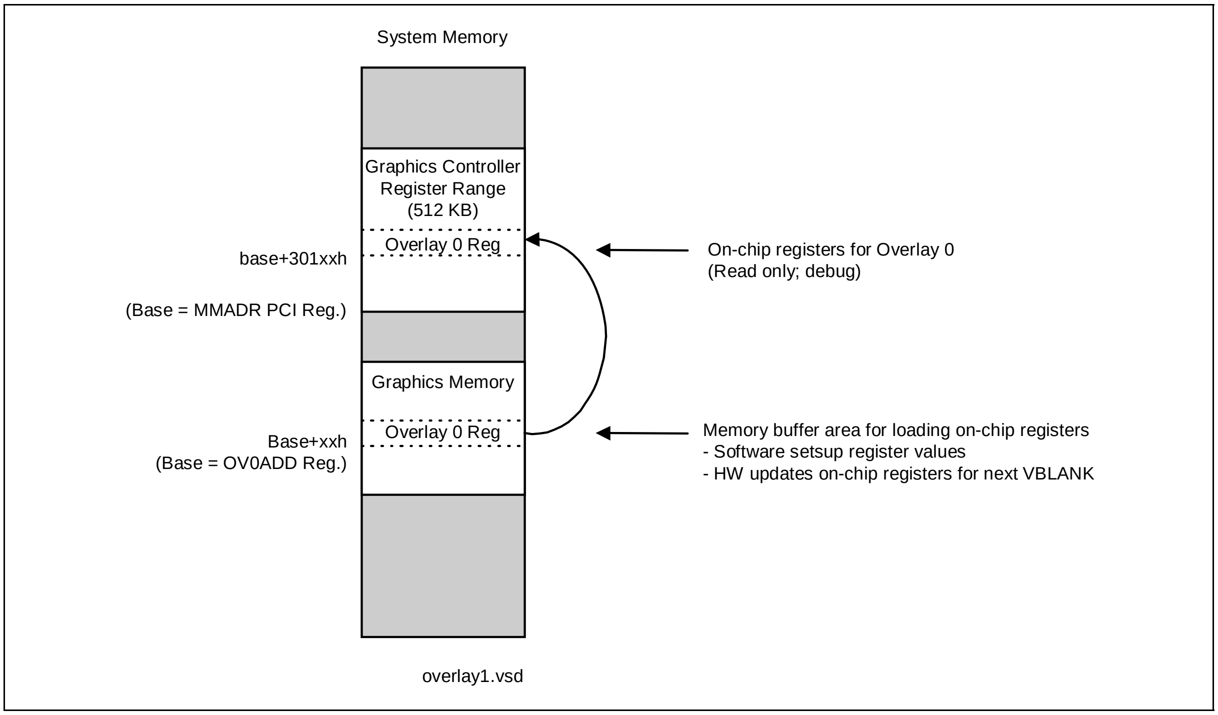

System Memory
Graphics Controller Register Range (512 KB)
Overlay 0 Reg
base+301xxh
(Base = MMADR PCI Reg.)
On-chip registers for Overlay 0
(Read only; debug)
Graphics Memory
Overlay 0 Reg
Base+xxh
(Base = OV0ADD Reg.)
Memory buffer area for loading on-chip registers
- Software setsup register values
- HW updates on-chip registers for next VBLANK
overlay1.vsd

Table 3-1. Overlay Register Categories

Memory Address Offset	Register / Instruction Category	MMI (r/w)	Command Streamer (W)	Load from Memory	Comment
30000h–30003h	Overlay Control Register Update Address (OVADD)	RW	W (See Flip Inst)	No	Provides physical memory address of buffer area used for updating on-chip registers Used to update Overlay Control registers: A write to register OVADD causes hardware to update on-chip registers on next display VBLANK.
30004h–30007h	Overlay Test Register (OTEST)	RW	No	No	Overlay test
30008h–3000Bh	Display/Overlay Status Register (DOVSTA)	RO	No	No	Overlay status bits
3000Ch–3000Fh	Display/Overlay Extended Status Register (DOVSTAEX)	RO	No	No	Overlay extended status bits

Memory Address Offset	Register / Instruction Category	MMI (r/w)	Command Streamer (W)	Load from Memory	Comment
30010h–30027h	Piecewise Linear Gamma Correction Registers (GAMMA[0:5])	RW	No	No	Six breakpoint values in the adjustment curve define a gamma correction function using a piece-wise linear approximation. The end points of the curve are fixed to the lowest and highest possible values. Each color has its own correction curve. All three colors are accessed simultaneously for each breakpoint. Breakpoints on an eight-bit resolution are at 8,16,32,64,128, and 192 respectively with 0 and 255 as the two end points.
30028h–30057h	Reserved	N/A	No	No	
30058h–30067h	Video Sync Lock Phase Registers (SYNCPH0-3)	RO	No	No	Used for software-based display sync lock to a hardware or software overlay flip.
30068h– 300FFh	Reserved	N/A	No	No	
30100h–30177h (on chip RO)	Compatible Overlay Register Sets Buffer Pointers Stride/Source Size Initial Phase Window Position/Size Scale Factor Color Correction Color Key Configuration Command	RO	No	Yes	Memory image registers: On-chip registers are not directly write-able but are loaded from a memory image or via the command stream. Software sets up buffer in memory. Software writes to OVADD Register, which provides memory buffer address location and causes hardware to read memory buffer and update on-chip registers during next appropriate Display VBLANK.
30178h–3019Fh	Reserved	N/A	No	No	
301A0h–301A7h (on chip RO)	Overlay Scaling Registers (FASTHSCALE and UVSCALEV)	RO	No	Yes	Memory buffer registers: Base_Address + (A0h to A4h)
301ACh–302FFh	Reserved	N/A	No	No	
30300h–307FFh	Overlay Filter Coefficients	No	No	Yes	Memory buffer registers: Base_Address + (200h to 600h)
30424h–30FFFh	Reserved	N/A	No	No	

3.2.1.1 Updating Overlay Registers

Most Overlay Registers are double-buffered (specifically, all those with a "Yes" in the "Load from memory" column in the above table). The write-able copy resides in memory; this memory-resident buffer is called the "Overlay Register Back Buffer", or the "memory back buffer". This allows software to initiate an update of Overlay Register values at any time, including during active overlay display. The Overlay Register Back Buffer includes Overlay Control Registers and Overlay Filter Coefficient Registers. If two sequential requests to load registers occur before the actual registers are loaded result in the last request to be honored.

Multiple display pipes are supported, where the pipe assign bit selects which display pipe VBLANK event causes the registers to be loaded except in the case where the currently assigned display pipe is currently disabled. When the pipe assign bit is changing, a two step process automatically disables the overlay from its current pipe on the VBLANK event for that pipe and enables it on the VBLANK event of the new display pipe. This means that the pipe switch can take two VBLANKs to complete.

Two conditions can cause the on-chip Overlay Registers to be updated from the Overlay Register Back Buffer:

1. A write to the OVADD register clears bit 31 of the Overlay Status Register and causes the values currently in the Overlay Register Back Buffer in memory to be loaded into the corresponding on-chip registers on the next Display VBLANK event. (VBLANK event is defined here as the active edge of the Display Vertical Blank period, including border). The Overlay Register Update Status bit is asserted after all registers are updated from memory. This method cannot be used to configure the cache during the initial enabling of the overlay or final disable. The cache must be configured to load registers. In general this method is not to be used!

2. Similarly, an Overlay Flip (or Extended Flip) Instruction, issued via the Command Stream, will cause the exact same register update sequence, upon the next Display VBLANK event.

3.2.1.2 Read Path for Overlay Registers

The on-chip overlay registers, with the exception of filter coefficient registers, are readable for state saving, debug readback, or other software purposes. The address offsets correspond to the memory offsets from the graphics base address used to read the registers.

Memory Address Offset: 30xxxh (xxx = register offset)

In normal operation, software manages the Overlay Register Back Buffer.

3.2.2 OVADD—Overlay Register Update Address Register

Memory Address Offset: 30000h–30003h
Default Value: 00000000h
Normal Access: R/W
Trusted Type: Type 1
Size: 32 bits

This register provides a graphics memory address that will be used on the next Overlay register update. This graphics memory address points to an array of Overlay registers. This register cannot be used to flip the overlay if the cache has not been configured through a command pipe flip packet.

Bit	Description
31:12	**Register Update Address:** Graphics memory address that will be used on the next Overlay register update. For forward compatibility purpose, software must ensure that the graphics memory address is at least 4K aligned (i.e., the lower 1 bits are assumed to be zero and do not exceed the proper values).
11:1	Reserved: MBZ
0	**Coefficient Flag**: This bit determines if the coefficients will be loaded in addition to the other overlay registers during an overlay flip. 0 = Load registers only, do not load coefficients 1 = Load registers and coefficients

3.2.3 OTEST—Overlay Test Register

Memory Address Offset: 30004h–30007h
Default Value: XXXXXXXX
Normal Access: R/W
Trusted Type: Type 1
Size: 32 bits

Bit	Description
31:0	Reserved: MBZ
2:0	Reserved: MBZ

3.2.4 DOVSTA—Display / Overlay Status Register

Memory Address Offset:	30008h–3000Bh
Default Value:	XXXXXXXX (values change dynamically)
Normal Access:	RO
Trusted Type:	Type 1
Size:	32 bits

This read-only register indicates status for the overlay.

Since the Overlay pipe can be assigned to either display pipe, references to display are either the pipe A timing generator or the pipe B timing generator depending on which the Overlay logic is currently slaved to.

Bit	Description
31	**Overlay Register Update Status (OVR_UPDT)**: This status bit is only applicable to the case that software writes OVADD (via either MMIO or Command Stream) to trigger hardware to load the on-chip overlay registers from the memory back buffer. This bit is cleared to zero (by hardware) when OVADD is written. It is set to one when the hardware completes loading the on-chip registers. This bit in effect acknowledges that a buffer flip has completed. When flipping the overlay from one active display pipe to the other, this occurs when the VBLANK of the target display pipe following the VBLANK of the current display pipe has occurred. 0 = Overlay Register Update register has been written, however, display VBLANK event has not yet occurred and Overlay Registers have not been loaded from memory. 1 = Overlay Register has not been updated since the last VBLANK event. This is the power-on default value.
30:22	Reserved: MBZ
21:20	**Overlay Current Buffer (OC_BUF)**: This field indicates which overlay buffer is currently being displayed. It is updated at display VBLANK. The update occurs before display VBLANK interrupt. 00 = Buffer 0 01 = Buffer 1 1x = Reserved
19	**Overlay Current Field (OC_FIELD):** This bit indicates the overlay source field currently displayed. It is updated at display VBLANK, occurring before display VBLANK interrupt. It is only valid in interleaved buffer (or Field) mode. In non-interleaved buffer (or Frame) mode, this bit is always 0. (See bit 5 of the Command Register for more on Field/Frame modes). 0 = Field 0 1 = Field 1
18	Reserved: MBZ
17	**Overlay Hardware Error (OV_ERR):** This status bit indicates that there has been an error detected during Overlay operation. It is set when the data underrun condition is detected. It's cleared by reading this register. This bit can be used in diagnostic tests to determine if there is enough memory bandwidth for a given resolution.
16	**RESERVED (was Target Field (T_FIELD))** MBZ
15	Reserved: MBZ

Bit	Description
14	**Not Active Display Pixel (NACT_PEL):** This bit indicates the Display Horizontal Blank Active state of the graphics pipe that the overlay is associated with. The Display Horizontal Blank includes the horizontal Border It is updated in real time, set by the leading edge of Overlay's display HBLANK and cleared by the trailing edge of the HBLANK. 0 = HBLANK inactive 1 = HBLANK active
13	Reserved: MBZ
12	**Not Active Display Scan Line (NACT_LINE):** This bit indicates the Display Vertical Blank Active state of the graphics pipe that the overlay is associated with. The Display Vertical Blank includes the vertical Border. This field is updated in real time, set by leading edge of display VBLANK and cleared by the trailing edge of VBLANK. 0 = VBLANK inactive 1 = VBLANK active
11:0	Reserved: MBZ

3.2.5 DOVSTAEX—Display/Overlay Extended Status Register

Memory Address Offset:	3000Ch–3000Fh
Default Value:	XXXXXXXX (values change dynamically)
Normal Access:	RO
Trusted Type:	Type 1
Size:	32 bits

This read-only register provides extended status information about the overlay. The format is RESERVED.

3.2.6 OGAMC[5:0]—Overlay Dedicated Gamma Correction Registers

Memory Address Offset:	30010h – 30027h GAMC5: 30010h–30013h GAMC4: 30014h–30017h GAMC3: 30018h–3001Bh GAMC2: 3001Ch–3001Fh GAMC1: 30020h–30023h GAMC0: 30024h–30027h

Default Value:	Linear R/G/B ramp
	GAMC5 = C0C0C0h
	GAMC4 = 808080h
	GAMC3 = 404040h
	GAMC2 = 202020h
	GAMC1 = 101010h
	GAMC0 = 080808h

Normal Access:	R/W
Trusted Type:	Type 1
Size:	5 x 32 bits

These registers are used to determine the characteristics of the gamma correction for the overlay data. The gamma correction receives 8-bit per channel pixels input, and sends out 10 bit per channel pixels to the display blender. Each register has 32 bits, which are written to and read from together when accessed by the software. They are the six individual breakpoints on a logarithmically spaced color intensity space as shown in the following figure. The two end points (0 and 255) have fixed values 0 and 255, respectively. The appropriate Gamma breakpoint pairs (adjacent) are selected for each color component (Red, Green and Blue), and the output is interpolated between these two breakpoint values. The difference between any two points should never exceed 7E hex and sequential points must be greater than or equal to the previous point. The Gamma Correction registers (GAMC0 to GAMC5) are not double-buffered. They should be updated when the overlay is off. Otherwise, video anomaly may show.

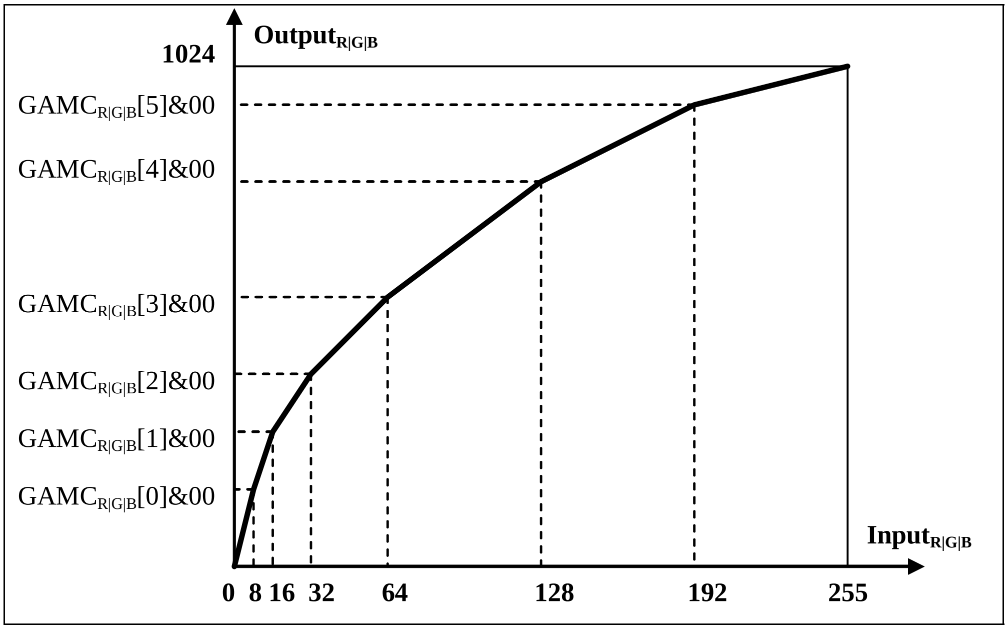
When the output from overlay is set in YUV format by programming CSC bypass, normally software should also bypass this gamma unit. However, since this gamma unit can also be viewed as a nonlinear transformation, it can be used, for whatever reason, in YUV output mode. In this case, the mapping of the three sets of piecewise linear map are as the following:

- Red to Cr (also called V)
- Green to Y
- Blue to Cb (also called U)

Errata: Overlay fails when gamma point 5 is set to 0x80.

Bit	Description
31:24	Reserved: MBZ
23:16	**Red (V/Cr)**
15:8	**Green (Y)**
7:0	**Blue (U/Cb)**

Figure 3-2. Programming of the Piecewise-linear Estimation of Gamma Correction Curve



3.2.7 Overlay Sync Lock Registers

These registers are used to implement a software involved sync lock. The four registers contain the last four flip event phases and are loaded in a round robin fashion. The valid bit indicates that the register contains valid phase information. The frame value is the frame number determines the order that the registers were loaded and the phase is the line that the flip occurred on. Display events are driven from the display pipe that the overlay is assigned to. The registers are reset to zero at power up and when the overlay is disabled.

3.2.7.1 OVRSYNCPH[0-3]— Overlay Flip Sync Lock Phase Registers

Memory Address Offset:	30058h–30067h SYNCPH0: 30058h–3005Bh SYNCPH1: 3005Ch–3005Fh SYNCPH2: 30060h–30063h SYNCPH3: 30064h–30067h
Default Value:	00h
Normal Access:	RO
Trusted Type:	Type 1
Size:	4 x 32 bits

Bit	Descriptions
31	**Data Valid:** This bit informs software that the phase information in the register is valid. It is set when a flip event occurs on the overlay (capture or software) and is cleared when the register is read. 0 = Data is not valid 1 = Data is valid
30	**Overrun:** This bit indicates that the register was updated before software had read the register and cleared the valid bit. 0 = No overrun has occurred 1 = Overrun
29:24	**Frame:** This field contains the value in the frame counter (a free running counter which counts display VBLANK events) when the flip occurred. When the frame counter reaches its maximum value, it just wraps around to zero.
23:12	Reserved: MBZ
11:0	**Phase:** This field indicates the phase in lines between the display VBLANK event and the overlay flip event.

3.2.8 Overlay Memory Image Offset Registers

Only the Register Update Address register should be written while the overlay is active. There is a debug read path to the on-chip active overlay control registers for testing overlay internal register functionality. The debug read is from the active overlay registers.

Note that the Overlay Enable bit in Overlay Command register must be 0 after power up. Default value for the rest on-chip compatible control registers are meaningless (don’t care), since it is required to load them from memory before the Overlay engine is enabled. The newly-added registers have to be defaulted to the compatible values in order for the new hardware to support legacy software, which does not update the additional registers (including control and filter coefficient registers).

Table 3-2. Overlay Memory Offset Registers

Register Mnemonic	Register Name	Memory Address Offset	On-chip Address Offset
Compatible Overlay Control Registers			
OBUF_0Y	Overlay Buffer 0 Y Pointer	00h–03h	30100h–30103h
OBUF_1Y	Overlay Buffer 1 Y Pointer	04h–07h	30104h–30107h
OBUF_0U	Overlay Buffer 0 U Pointer	08h–0Bh	30108h–3010Bh
OBUF_0V	Overlay Buffer 0 V Pointer	0Ch–0Fh	3010Ch–3010Fh
OBUF_1U	Overlay Buffer 1 U Pointer	10h–13h	30110h–30113h
OBUF_1V	Overlay Buffer 1 V Pointer	14h–17h	30114h–30117h
OSTRIDE	Overlay Stride	18h–1Bh	30118h–3011Bh
YRGB_VPH	Y/RGB Vertical Phase 0/1	1Ch–1Fh	3011Ch–3011Fh
UV_VPH	UV Vertical Phase 0/1	20h–23h	30120h–30123h
HORZ_PH	Horizontal Phase	24h–27h	30124h–30127h
INIT_PHS	Initial Phase Shift	28h–2Bh	30128h–3012Bh
DWINPOS	Destination Window Position	2Ch–2Fh	3012Ch–3012Fh
DWINSZ	Destination Window Size	30h–33h	30130h–30133h
SWIDTH	Source Width	34h–37h	30134h–30137h
SWIDTHSW	Source Width In SWORDS	38h–3Bh	30138h–3013Bh
SHEIGHT	Source Height	3Ch–3Fh	3013Ch–3013Fh
YRGBSCALE	Y/RGB Scale Factor	40h–43h	30140h–30143h
UVSCALE	U V Scale Factor	44h–47h	30144h–30147h
OCLRC0	Overlay Color Correction 0	48h–4Bh	30148h–3014Bh
OCLRC1	Overlay Color Correction 1	4Ch–4Fh	3014Ch–3014Fh
DCLRKV	Destination Color Key Value	50h–53h	30150h–30153h
DCLRKM	Destination Color Key Mask	54h–57h	30154h–30157h
SCHRKVH	Source Chroma Key Value High	58h–5Bh	30158h–3015Bh
SCHRKVL	Source Chroma Key Value Low	5Ch–5Fh	3015Ch–3015Fh
SCHRKEN	Source Chroma Key Enable	60h–63h	30160h–30163h
OCONFIG	Overlay Configuration	64h–67h	30164h–30167h
OCMD	Overlay Command	68h–6Bh	30168h–3016Bh
	Reserved	6Ch–6Fh	30168h–3016Fh
OSTART_0Y	Overlay Surface Start 0 Y Pointer	70h–73h	30170h–30173h
OSTART _1Y	Overlay Surface Start 1 Y Pointer	74h–77h	30174h–30177h
OSTART _0U	Overlay Surface Start 0 U Pointer	78h–7Bh	30178h–3017Bh
OSTART _0V	Overlay Surface Start 0 V Pointer	7Ch–7Fh	3017Ch–3017Fh
OSTART _1U	Overlay Surface Start 1 U Pointer	80h–84h	30180h–30184h
OSTART _1V	Overlay Surface Start 1 V Pointer	84h–87h	30184h–30187h
OTILEOFF_0Y	Overlay Surface 0 Y Tiled Offset	88h–8Bh	30188h–3018Bh
OTILEOFF _1Y	Overlay Surface 1 Y Tiled Offset	8Ch–8Fh	3018Ch–3018Fh

Register Mnemonic	Register Name	Memory Address Offset	On-chip Address Offset
OTILEOFF _0U	Overlay Surface 0 U Tiled Offset	90h–93h	30190h–30193h
OTILEOFF _0V	Overlay Surface 0 V Tiled Offset	94h–97h	30194h–30197h
OTILEOFF _1U	Overlay Surface 1 U Tiled Offset	98h–9Bh	30198h–3019Bh
OTILEOFF _1V	Overlay Surface 1 V Tiled Offset	9Ch–9Fh	3019Ch–3019Fh
Additional Overlay Control Registers			
FASTHSCALE	Fast Horizontal Downscale	A0h–A3h	301A0h–301A3h
UVSCALEV	UV Vertical Downscale Integer	A4h–A7h	301A4h–301A7h
	Reserved	A8h–1FFh	301C0h–302FFh
Overlay Filter Coefficient Registers			
Y_VCOEFS	Overlay Y Vertical Filter Coefficient	200h–267h	30300h–30367h
	Reserved	268h–2FFh	30368h–303FFh
Y_HCOEFS	Overlay Y Horizontal Filter Coefficients	300h–3ABh	304000h–304ABh
	Reserved	3ACh–3FFh	304ACh–3034FFh
UV_VCOEFS	Overlay UV Vertical Filter Coefficients	500h–567h	30600h–30667h
	Reserved	568h–5FFh	30668h–306FFh
UV_HCOEFS	Overlay UV Horizontal Filter Coefficients	600h–667h	30700h–30767h
	Reserved	668h–6FFh	30768h–307FFh

3.2.8.1 Overlay Buffer Start Address Pointer Registers

These registers provide overlay buffer start address pointers into graphics memory. While the start address has alignment restrictions based on pixel format, the buffers must be 64B aligned. Pixel panning on a pixel basis (for formats that support per pixel panning) is done using the start address pointer. Overlay buffers need to be 64B aligned and the stride should be a 64B multiple. In addition, buffer start address pointers should always be aligned to the natural boundaries based on the data format, as shown in the following table.

Table 3-3. Overlay Buffer Start Address Alignment Restriction

Pixel Format		Alignment	
Color	**Organization**	**Pixels**	**Bytes**
YUV4:2:2	Packed	2	4
YUV /RGB (Test Only)	Planar	1	1

3.2.8.2 OBUF_0Y—Overlay Buffer 0 Y Linear Offset Register

Memory Buffer Address Offset: 00h–03h (R/W)
On-chip Reg. Mem Addr Offset: 30100h–30103h (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

This value specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the value in this register is ignored.

Bit	Description
31:0	**Overlay Buffer 0 Y Linear Offset:** This is the offset for the Y planar or YUV packed color data. This value is added to the surface address to get the graphics address of the first pixel to be displayed. It must be pixel aligned (e.g., low order bit is zero for 16-bpp packed formats). This offset is the difference between the address of the upper left pixel to be displayed and the overlay surface address. When mirroring horizontally (X backward) this field points to first byte of the last pixel of the line.

3.2.8.3 OBUF_1Y—Overlay Buffer 1 Y Linear Offset Register

Memory Address Offset: 04h–07h (R/W)
On-chip Reg. Mem Addr Offset: 30104h–30107 (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

This value specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the value in this register is ignored.

Bit	Description
31:0	**Overlay Buffer 1 Y Linear Offset:** This is the offset for the Y planar or YUV packed color data. This value is added to the surface address to get the graphics address of the first pixel to be displayed. It must be pixel aligned (e.g., low order bit is zero for 16-bpp packed formats). This offset is the difference between the address of the upper left pixel to be displayed and the overlay surface address. When mirroring horizontally (X backward) this field points to first byte of the last pixel of the line.

3.2.8.4 OBUF_0U—Overlay Buffer 0 U Linear Offset Register

Memory Address Offset:	08h–0Bh (R/W)
On-chip Reg. Mem Addr Offset:	30108h–3010B (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

This value specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the value in this register is ignored.

Bit	Description
31:0	**Overlay Buffer 0 U Linear Offset:** This is the offset for the U planar data. This value is added to the surface address to get the graphics address of the first pixel to be displayed. This offset is the difference between the address of the upper left pixel to be displayed and the overlay surface address. When mirroring horizontally (X backward) this field points to first byte of the last pixel of the line.

3.2.8.5 OBUF_0V—Overlay Buffer 0 V Linear Offset Register

Memory Address Offset:	0Ch–0Fh (R/W)
On-chip Reg. Mem Addr Offset:	3010Ch–3010Fh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

This value specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the value in this register is ignored.

Bit	Description
31:0	**Overlay Buffer 0 V Linear Offset:** This is the offset for the V planar data. This value is added to the surface address to get the graphics address of the first pixel to be displayed. This offset is the difference between the address of the upper left pixel to be displayed and the overlay surface address. When mirroring horizontally (X backward) this field points to first byte of the last pixel of the line.

3.2.8.6 OBUF_1U—Overlay Buffer 1 U Linear Offset Register

Memory Address Offset: 10h–13h (R/W)
On-chip Reg. Mem Addr Offset: 30110h–30113 (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

This value specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the value in this register is ignored.

Bit	Description
31:0	**Overlay Buffer 1 U Linear Offset:** This is the offset for the U planar data. This value is added to the surface address to get the graphics address of the first pixel to be displayed. This offset is the difference between the address of the upper left pixel to be displayed and the overlay surface address. When mirroring horizontally (X backward) this field points to first byte of the last pixel of the line.

3.2.8.7 OBUF_1V—Overlay Buffer 1 V Linear Offset Register

Memory Address Offset: 14h–17h (R/W)
On-chip Reg. Mem Addr Offset: 30114h–30117h (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

This value specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, this field contains the byte offset of the plane data in graphics memory. When the surface is tiled, the value in this register is ignored.

Bit	Description
31:0	**Overlay Buffer 1 V Linear Offset:** This is the offset for the V planar data. This value is added to the surface address to get the graphics address of the first pixel to be displayed. This offset is the difference between the address of the upper left pixel to be displayed and the overlay surface address. When mirroring horizontally (X backward) this field points to first byte of the last pixel of the line.

3.2.8.8 OSTRIDE—Overlay Stride Register

Memory Address Offset:	18h–1Bh (R/W)
On-chip Reg. Mem Addr Offset:	30118h–3011B (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

These values represent the stride of the overlay video data buffers. A stride value determines the line-to-line increment of the buffer, which is independent of the actual width of the overlay video data that gets displayed.

Bit	Description
31:16	**Overlay UV Planar Buffer Stride (UV_STRIDE)**: Only used for YUV Planar formats and gives the U or V buffer stride in bytes. This is a two’s complement number and will be negative when Y mirroring is used with linear memory. Hardware ignores the least significant 6 bits. When using tiled memory, the stride must be positive, 512 byte aligned, and cannot exceed 2kb.
15:0	**Overlay Y Planar or YUV422 Packed Buffer Stride (YRGB_STRIDE).** Buffer (Y planar or YUV packed) stride in bytes. This is a two’s complement number and will be negative when Y mirroring is used with linear memory. Low order 6 bits must always be zero. The minimum stride is 512 bytes and the maximum is 8kb for packed mode and 4kb for linear modes. When using tiled memory, the stride must be positive.

3.2.9 Overlay Initial Phase Registers

These provide a spatial sub-pixel accurate adjustment in overlay source data coordinate. The values in the following initial phase registers are always fractional positive numbers. With 12-bit accuracy, the value ranges from 0 to (1-1/FFFh). When combined with the subtract one flags in the Initial Phase Shift register, the possible range for an initial phase is limited to:

-1 ≥ Initial Phase < 2

There are two separate vertical initial phase registers that are used in field based video display operation. YRGB_VPH is for Y/RGB buffers. UV_VPH is for UV buffers in planar formats. An additional field indicates an additional offset to be used to eliminate edge effects when panning.

Initial Phase Shift table

Table 3-4. Vertical Initial Phase (for 1:1 scaling, upper left sync)

Format	Y line #	Yip	Yip Range	Cip	Cip Range	Ratio
422	n	0.00	-.5 to +.5	0.000	-.5 to +.5	1:1
420	1p	0.00	-.5 to +.5	-0.250	-.5 to +.5	1:2
	2p	1.00	+.5 to +1.5	0.250	0.0 to +.5	1:2
420i2i	1f0	0.00	-.5 to +.5	-0.125	-.375 to .125	1:2
	2f0	1.00	.5 to 1.5	0.375	.125 to .625	1:2
	1f1	0.00	-.5 to +.5	-0.375	-.625 to -.125	1:2
	2f1	1.00	.5 to 1.5	0.125	-.125 to .375	1:2
420i2p	1f0	0.25	-.25 to .75	0.000	-.125 to +.125	1:2
	2f0	1.25	.75 to 1.75	0.500	.25 to .75	1:2
	1f1	-0.25	-.75 to +.25	-0.500	-.75 to .25	1:2
	2f1	0.75	.25 to 1.25	0.000	-.125 to .125	1:2
410	1	0.00	-.5 to .5	-0.375	-.5 to -.25	1:4
	2	1.00	.5 to 1.5	-0.125	-.25 to 0.0	1:4
	3	1.00	.5 to 1.5	0.125	0.0 to .25	1:4 adjust Y base address + 1 line
	4	1.00	.5 to 1.5	0.375	.25 to .5	1:4 adjust Y base address + 2 lines

Table 3-5. Horizontal Initial Phase (for 1:1 scaling, upper left sync)

Format	Y pixel #	Yip	Yip Range	Cip	Cip Range	Ratio
422	1	0.00	-.5 to .5	0.000	-.25 to .25	1:2
	2	1.00	.5 to 1.5	0.500	.375 to .625	1:2
420mp2	1	0.00	-.5 to .5	0.000	-.25 to .25	1:2
	2	1.00	.5 to 1.5	0.500	.25 to .75	1:2
420mp1	1	0.00	-.5 to .5	-0.250	-.5 to 0.0	1:2
	2	1.00	.5 to 1.5	0.250	0.0 to .5	1:2
410	1	0.00	-.5 to .5	-0.375	-.5 to -.25	1:4
	2	1.00	.5 to 1.5	-0.125	-.25 to 0.0	1:4
	3	2.00	1.5 to 2.5	0.125	0.0 to .25	1:4
	4	3.00	2.5 to 3.5	0.375	.25 to .5	1:4

3.2.9.1 YRGB_VPH—Y/RGB Vertical Phase Register

Memory Address Offset:	1Ch–1Fh (R/W)
On-chip Reg. Mem Addr Offset:	3011Ch–3011Fh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:20	**Y/RGB Vertical Phase for Field 1 (YRGB_VPHASE1)**: This fractional value sets the initial vertical phase for field 1. For packed formats, it applies to both YUV and RGB buffers. For planar formats, it only applies to Y buffers. When the overlay buffers are in Interleaved buffer format, this value sets the initial vertical phase for field 1 in all YUV/RGB buffers. When the overlay buffers are in Non-Interleaved format, this value is intended for field 1. The value now applies to a particular YUV/RGB buffer.
19:16	Reserved: (Software should set all the 16 bits for forward compatibility in case more accurate DDA phase is implemented.)
15:4	**Y/RGB Vertical Phase for Field 0 (YRGB_VPHASE0)**: This fractional value sets the initial vertical phase for field 0. For packed formats, it applies to YUV or RGB buffers. For planar formats, it only applies to Y buffers. When the overlay buffers are in Interleaved buffer format, this value sets the initial vertical phase for field 0 in all YUV/RGB buffers. When the overlay buffers are in Non-Interleaved format, this value is intended for field 0. The value applies to a particular YUV/RGB buffer.
3:0	Reserved: (Software should set all the 16 bits for forward compatibility in case more accurate DDA phase is implemented.)

3.2.9.2 UV_VPH—UV Vertical Phase Register

Memory Address Offset:	20h–23h (R/W)
On-chip Reg. Mem Addr Offset:	30120h–30123h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:20	**UV Vertical Phase for Field 1 (UV_VPHASE1)**: This fractional value is only used in YUV planar formats where the UV plane may have a different vertical initial phase from the Y data. When the overlay buffers are in Interleaved buffer format, this value sets the initial vertical phase for field 1 in all UV data buffers. When the overlay buffers are in Non-Interleaved format, this value is intended for buffers containing field 1 data. The value applies to a particular UV data buffer.
19:16	Reserved: (Software should set all the 16 bits for forward compatibility in case more accurate DDA phase is implemented.)
15:4	**UV Vertical Phase for Field 0 (UV_VPHASE0)**: This fractional value is only used in YUV planar formats where the UV plane may have a different vertical initial phase from the Y data. When the overlay buffers are in Interleave buffer format, this value sets the initial vertical phase for field 0 in all UV data buffers. When the overlay buffers are in Non-Interleaved format, this value is intended for buffers containing field 0 data. The value applies to a particular UV data buffer.
3:0	Reserved: (Software should set all the 16 bits for forward compatibility in case more accurate DDA phase is implemented.)

3.2.9.3 HORZ_PH—Horizontal Phase Register

Memory Address Offset: 24h–27h (R/W)
On-chip Reg. Mem Addr Offset: 30124h–30127h (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

Bit	Description
31:20	**UV Horizontal Phase (UV_HPHASE)**: Sets the initial horizontal phase for the U and V data. Only used in YUV modes.
19:16	Reserved: MBZ
15:4	**Y/RGB Horizontal Phase (YRGB_HPHASE)**: Sets the initial horizontal phase for both buffers/fields. Unlike the vertical initial phases, this does not change buffer-to-buffer or field-to-field. YUV modes use a separate initial phase for Y and UV data. This value will either be the actual initial phase or the initial phase minus one.
3:0	Reserved: MBZ

3.2.9.4 INIT_PHS—Initial Phase Shift Register

Memory Address Offset: 28h–2Bh (R/W)
On-chip Reg. Mem Addr Offset: 30128h–3012Bh (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

This register provides a method to create a negative initial phase or one with a positive integer offset. If the corresponding bits are set to all ones, the initial phase is the fractional phase register value minus one. These bits should only be set in cases where the buffer pointer is pointing to the first pixel of the line or column because it will effectively cause the first pixel to be duplicated.

Bit	Description
31:24	Reserved: MBZ
23:20	**Y0_VPP Initial Phase for Vertical Panning:** 0000 = 0 Positive fractional phase 0001 = fractional phase plus one 0010 = fractional phase plus two 0011–1110 = Reserved 1111 = fractional phase minus one (-1)
19:16	**Y1_VPP Initial Phase for Vertical Panning:**
15:12	**Y_HPP Initial Phase for Horizontal Panning:**
11:8	**UV0_VPP Initial Phase for Vertical Panning:**
7:4	**UV1_VPP Initial Phase for Vertical Panning:**
3:0	**UV_HPP Initial Phase for Horizontal Panning:**

3.2.10 Overlay Destination Window Position/Size Registers

These registers allow for the positioning of the overlay data relative to the current graphics pipe that overlay is attached to (see the overlay and display association bits in OVIMMCTL register). It allows pixel accurate positioning.

3.2.10.1 DWINPOS—Destination Window Position Register

Memory Address Offset:	2Ch–2Fh (R/W)
On-chip Reg. Mem Addr Offset:	3012Ch–3012Fh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:28	Reserved: MBZ
27:16	**Destination Vertical Top:** The vertical top line position of the overlay destination window in lines. Zero means the first active display line. When performing 180° rotation, this field specifies the vertical bottom line of the overlay destination window in its unrotated orientation. The range is [0, 4095].
15:12	Reserved: MBZ
11:0	**Destination Horizontal Left:** The horizontal left position of the overlay destination window in pixels. It determines where on the display screen the overlay window will begin. Zero means the first active pixel of the display screen. When performing 180° rotation, this field specifies the horizontal right position of the overlay destination window in its unrotated orientation. The range is [0, 4095].

3.2.10.2 DWINSZ—Destination Window Size Register

Memory Address Offset:	30h–33h (R/W)
On-chip Reg. Mem Addr Offset:	30130h–30133h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:28	Reserved: MBZ
27:16	**Destination Height:** Overlay destination window height in lines (never specifies scan lines off the active display). For devices with internal panel fitting, the single line mode requires this value to be based on the panel height instead of the source height.
15:12	Reserved: MBZ
11:0	**Destination Width**: Overlay destination window width in pixels (never specifies pixels off the active display)

3.2.11 Overlay Source Size Registers

These registers provide information to the overlay engine on what data needs to be fetched from memory. If the overlay destination window is smaller than the result of the scaled (up or down) source, it will be clipped on the right and bottom of the overlay window.

3.2.11.1 SWIDTH – Source Width Register

Memory Address Offset:	34h–37h (R/W)
On-chip Reg. Mem Addr Offset:	30134h–30137h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:27	Reserved: MBZ
26:16	**UV Pixel Source Width**: The number of valid pixels contained in a single line of planar UV source data. This field is unused in packed modes. For planar modes it's used for the U and V source width. The width for U and V sources is required to be the same. When the last pixel is reached and the complete destination window has not been filled, this pixel will be repeated until the end of the destination window. When displaying planar YUV 4:2:0 data, this field contains the number of U pixels (same as the number of V pixels), which should be one half of the Y pixels. When displaying planar YUV 4:1:0 planar data, this field contains the number of U pixels (same as the number of V pixels), which should be one quarter of the Y pixels. The UV source width cannot be more than half the Y source width. Source formats with a Y:UV ratio of less than 2:1, such as YUV 4:4:4 with a ratio of 1:1, are not supported.
15:12	Reserved: MBZ
11:0	**Y/RGB Pixel Source Width:** The number of valid pixels contained in a single line of source data. In both planar and packed modes, this is the Y source width. This field should include all contributing pixel data. When the last pixel is reached and the complete destination window has not been filled, this pixel will be repeated until the end of the destination window. For source formats of YUV 4:1:0 data, the atomic unit is four pixels For source formats of YUV 4:2:2 or YUV 4:2:0 data, the atomic unit is two pixels The starting offset within the buffer must reflect these restrictions.

3.2.11.2 SWIDTHSW—Source SWORD Width Register

Memory Address Offset: 38h–3Bh (R/W)
On-chip Reg. Mem Addr Offset: 30138h–3013Bh (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

Hardware uses values in this register to determine the number of SWORDs (32 bytes) to be fetched from the memory for each overlay source scan line. The formula below account for the memory requirment of 64byte alignments.

Bit	Description
31:24	Reserved: MBZ
23:18	**UV Source SWORD Width (UV_SWIDTHSW)**: This field sets the number of aligned SWords that contains all source U/V pixels in a single line of planar U/V source data. This field is valid only in planar modes. Minimum size is 1 SWORD. Overlay hardware uses this field to make memory request for each scan line in the planar U/V source buffers. In normal overlay horizontal display order (left to right), software should calculate this field based on the pixel-aligned buffer pointer *UVAddress* and the pixel-aligned source window width *UVSWidth* by UVSWidthSW = {{[(UVAddress + UVSWidth + 0x3F) >> 6] - [UVAddress >> 6]} << 1} In mirrored overlay horizontal display order (for right to left), software should use UVSWidthSW = {{[UVAddress >> 6] - [(UVAddress - (UVSWidth + 0x3F)) >> 6]} << 1}
17:9	Reserved: MBZ
8:2	**Y/RGB Source SWORD Width (YRGB_SWIDTHSW)**: This field sets the number of aligned SWords that contains all source Y/RGB pixels in a single source line. In planar modes, this is the Y source SWORD width. Minimum size is 1 SWORD. Overlay hardware uses this field to make memory request for each source scan line. In normal overlay horizontal display order (left to right), software should calculate this field based on the pixel-aligned buffer pointer *YAddress* and the pixel-aligned source window width *YSWidth* by YSWidthSW = {{[(YAddress + YSWidth + 0x3F) >> 6] - [YAddress >> 6]} << 1} In mirrored overlay horizontal display order (for right to left), software should use YSWidthSW = {{[YAddress >> 6] - [(YAddress - (YSWidth + 0x3F)) >> 6]} << 1}
1:0	Reserved: MBZ

3.2.11.3 SHEIGHT—Source Height Register

Memory Address Offset:	3Ch–3Fh (R/W)
On-chip Reg. Mem Addr Offset:	3013Ch–3013Fh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:26	Reserved: MBZ
25:16	**UV Source Height**: The number of lines contained in a single planar UV source data. In packed formats this is unused. For planar YUV formats it indicates the number of lines starting at and including the base address line contained in the UV source data. When the last line is reached and the complete destination window has not been filled, this line will be repeated until the end of the destination window. The maximum UV height is 1023 lines. The minimum height is the number of vertical taps being used times the vertical chroma-subsampling ratio. Height should include the line contributing to the interpolation of the last display overlay line**.**
15:11	Reserved: MBZ
10:0	**Y/RGB Source Height**: In packed formats this indicates the number of lines starting at and including the base address line contained in the source data. In planar formats, it is the number of Y lines. This is used to determine where the end of the source is in the vertical direction to handle the edge effects related to the vertical filter. When the last line is reached and the complete destination window has not been filled, this line will be repeated until the end of the destination window. The maximum height is 2047 lines. The minimum height is the number of vertical taps currently being used. A height must not be specified for lines that are completely not on the active display. Height should include the line contributing to the interpolation of the last display overlay line**.**

3.2.12 Overlay Scale Factor Registers

These registers provide the scaling information that is used to specify the amount of vertical and horizontal scaling. In the case of YUV formats, there are independent scale factors for Y and UV data to compensate for the various formats that include sub-sampled UV data. Software has to ensure that the programmed scaling factors for Y and UV data match to each other precisely. Upscaling or downscaling is selected using a zero value in the integer scale factor field.

The 12 bits fractional portion of the scaling factor corresponds to a scaling accuracy of 1/4096 of a pixel.

3.2.12.1 YRGBSCALE – Y/RGB Scale Factor Register

Memory Address Offset:	40h–43h (R/W)
On-chip Reg. Mem Addr Offset:	30140h–30143h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:20	**Y/RGB Vertical Scale Fraction (YRGB_VFRACT)**: This is a fractional positive number that represents the scale factor to be used in vertical scaling for Y/RGB components. For packed formats, it applies to all color components. For planar YUV formats, it is use for the Y (luma) component only. This field represents a value range from 0 to 1 (0 < Vertical Scale Fraction < 1). Defining a **Ver_Scale_Factor** as the ratio of source height to destination height. Ver_Scale_Factor = Source Image Height / Destination Height For upscaling, Y/RGB Vertical Scale Fraction is Ver_Scale_Factor, which is a fractional value. The integer portion should be set to zero. For downscaling, Y/RGB Vertical Scale Fraction is the fractional portion of Ver_Scale_Factor. The integer portion is in Y/RGB Vertical Downscale Integer field. For devices with integrated panel fitting, the single line mode requires that the vertical scale factor be determined by the panel height instead of the display mode's source height.
19	Reserved: MBZ
18:16	**Y/RGB Horizontal Downscale Factor Integer (YRGB_HINT)**: This field is used for horizontal down scale. A value of zero is used to indicate upscale. In the case of planar formats, it specifies the integer portion of the scale factor for Y data. This is used in backward compatible mode. Only horizontal downscaling of 2 (or less) is supported through the precision filter. For greater down scale factors, it must be done through the render path.
15	Reserved: MBZ

Bit	Description
14:3	**Y/RGB Horizontal Scale Fraction (YRGB_HFRACT)**: This is a fractional positive number that represents the scale factor to be used in horizontal scaling for Y/RGB components. For both packed formats and planar YUV formats, this field applies to all color components. This field represents a value range from 0 to 1 (0 < X Scale Fraction < 1). For the situations that are other than downscaling, we can define a **Hor_Scale_Factor** as the ratio of source width to destination width. Hor_Scale_Factor = Source Width / Destination Width. For upscaling, Y/RGB Horizontal Scale Fraction is Hor_Scale_Factor, which is a fractional value. For downscaling, Y/RGB Horizontal Scale Fraction is the fractional portion of Hor_Scale_Factor. The integer portion is in Y/RGB Horizontal Downscale Integer field. For the situation that is downscaling, we can define a **Hor_Scale_Factor** as the ratio of source width to destination width for upscaling modes: Hor_Scale_Factor = Destination Width / Source Width. For downscaling, Y/RGB Horizontal Scale Fraction is the fractional portion of Hor_Scale_Factor.
2:0	Reserved: MBZ

3.2.12.2 UVSCALE—U V Scale Factor Register

Memory Address Offset: 44h–47h (R/W)
On-chip Reg. Mem Addr Offset: 30144h–30147h (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

Bit	Description
31:20	**UV Vertical Scale Fraction (UV_VFRACT)**: This is a fractional positive number that represents the scale factor to be used in vertical scaling for UV components. This field represents a value range from 0 to 1 (0 ≤ Vertical Scale Fraction < 1). Defining a **Ver_Scale Factor** as the ratio of source height over destination height. Ver_Scale_Factor = Source Image Height / Destination Height For upscaling, UV Vertical Scale Fraction is Ver_Scale_Factor, which is a fractional number. For downscaling, UV Vertical Scale Fraction is the fractional portion of Ver_Scale_Factor. The integer portion is in UV Vertical Downscale Integer field. . For devices with integrated panel fitting, the single line mode requires that the vertical scale factor be determined by the panel height instead of the display mode's source height.
19	Reserved: MBZ
18:16	**UV Horizontal Downscale Factor Integer (UV_HINT).** This field specifies the precision filter horizontal down scale integer portion. A value of zero is used to indicate upscale. In the case of planar formats, it specifies the integer portion of the scale factor for Y data. This is used in backward compatible mode. Only horizontal downscaling of 2 (or less) is supported through the precision filter. For greater down scale factors, must be done through the render path.
15	Reserved: MBZ
14:3	**UV Horizontal Scale Fraction (UV_HFRACT):** This is a fractional positive number that represents the scale factor to be used in horizontal scaling for UV components. This field represents a value range from 0 to 1 (0 < Hor Scale Fraction < 1). Defining a **Hor_Scale Factor** as the ratio of source width over destination width Hor_Scale_Factor = Source Width / Destination Width For upscaling, UV Horizontal Scale Fraction is Hor_Scale_Factor, which is a fractional number. For downscaling, UV Horizontal Scale Fraction is the fractional portion of Hor_Scale_Factor. The integer portion is in UV Horizontal Downscale Integer field.
2:0	Reserved: MBZ

3.2.13 Overlay Color Correction Registers

Intended for use for YUV sources. Adjustments are made before the YUV to RGB conversion. They take effect even when the YUV2RGB Conversion Bypass bit in Overlay Command register is set. For overlay input data in RGB format, software must set all the color correction registers to their default values (equivalent to a bypass mode).

3.2.13.1 OCLRC0—Overlay Color Correction 0 Register

Memory Address Offset:	48h–4Bh (R/W)
On-chip Reg. Mem Addr Offset:	30148h–3014Bh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:27	Reserved: MBZ
26:18	**Contrast**: Contrast adjustment applies to YUV data. The Y channel is multiplied by the value contained in the register field. This signed fixed-point number is in 3i.6f format with the first 3 MSBs as the integer value and the last 6 LSBs as the fraction value. The allowed contrast value ranges from 0 to 7.53125 decimal. Bypassing Contrast, for YUV modes and for source data in RGB format, is accomplished by programming this field to a field value that represents 1.0 decimal or 001.000000 binary.
17:8	Reserved: MBZ
7:0	**Brightness:** This field provides the brightness adjustment with a 8-bit 2's complement value ranging [-128, +127]. This value is added to the Y value after contrast multiply and before YUV to RGB conversion. A value of zero disables this adjustment affect. This 8-bit signed value provides half of the achievable brightness adjustment dynamic range. A full range brightness value would have a programmable range of [-255, +255]. Bypassing Brightness for YUV formats and for source data in RGB format is accomplished by programming this field to 0.

3.2.13.2 OCLRC1—Overlay Color Correction 1 Register

Memory Address Offset:	4Ch–4Fh (R/W)
On-chip Reg. Mem Addr Offset:	3014Ch–3014Fh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

The sum of the absolute value of SH_SIN and SH_COS must be limited to less than 8.

ABS(SH_SIN) + ABS(SH_COS) < 8

Bit	Description
31:27	Reserved: MBZ
26:16	**Saturation and Hue SIN (SH_SIN)**: This field can be used in two modes: • When full range YUV data is operated on, this field contains the saturation value. • When the range-limited YCbCr data is used, software should program this field with the sum of the saturation multiplier value added to the CbCr range scale factor (=128/112). Similar to the contrast field, there is no limit for saturation reduction – saturation = 0 means all pixels become the same value. However, increasing contrast can only be increased by a factor less than 8. For example, the largest contrast with value of 0x7.7F can bring input range [0, 32] to a full display color range of [0, 255]. Bypassing Hue, even for source data in RGB format, is accomplished by programming this field to 0.0. Format: This 11-bit signed fixed-point number is in 2’s complement (s3i.7f) format with the MSB as the sign, next 3 MSBs as the integer value and the last 7 LSBs as the fraction value
15:10	Reserved: MBZ
9:0	**Saturation and Hue COS (SH_COS):** This field can be used in two modes: • When full range YUV data is operated on, this field contains the saturation value. • When the range-limited YCbCr data is used, software should program this field with the sum of the saturation multiplier value added to the CbCr range scale factor (=128/112). Similar to the contrast field, there is no limit for saturation reduction – saturation = 0 means all pixels become the same value. However, increasing contrast can only be increased by a factor less than 8. For example, the largest contrast with value of 0x7.7F can bring input range [0, 32] to a full display color range of [0, 255]. Bypassing Saturation, even for source data in RGB format, is accomplished by programming this field to 1.0. Format: This unsigned fixed-point number is in 3i.7f format with the first 3 MSBs be the integer value and the last 7 LSBs be the fraction value.

3.2.14 Overlay Destination Color Key Registers

These source key registers are used for YUV overlay sources only. Adjustments are made before the RGB conversion. Destination keying is based on the format of the destination surface.

3.2.14.1 DCLRKV—Destination Color Key Value Register

Memory Address Offset:	50h–53h (R/W)
On-chip Reg. Mem Addr Offset:	30150h–30153h (RO; debug path)
Default Value:	00000000h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:24	Reserved: MBZ
23:0	**Destination Color Key Value**: Destination keying causes the color of the display primary or secondary to become transparent if it is within the overlay window and matches the destination color key. The key color is specified In the format of the destination surface data (converted to 24-bits) which is limited to either the primary (Display A) or secondary display (Display B). Destination keying is used only when overlay and Display Plane A or overlay and Display Plane B are on the same pipe. If both Display A and Display B are present on the same pipe as the overlay, overlay destination keying only applies to Display A. Color Channel Value: [23:16] = Red [15:08] = Green [07:00] = Blue

Table 3-6. Destination Color Key Value Programming in Each Supported Graphics Mode

Primary/Secondary Display Mode	Color Key Programmed Value		
	Red	Green	Blue
8-bit Indexed	Indexed Color [7:0]	Indexed Color [7:0]	Indexed Color [7:0]
15-bit RGB	RedKey[4:0] <<3	GreenKey[4:0] <<3	BlueKey[4:0] <<3
16-bit RGB	RedKey[4:0] <<3	GreenKey[5:0] <<2	BlueKey[4:0] <<3
32-bit x:8:8:8	RedKey[7:0]	GreenKey[7:0]	BlueKey[7:0]

3.2.14.2 DCLRKM—Destination Color Key Mask Register

Memory Address Offset:	54h–57h (R/W)
On-chip Reg. Mem Addr Offset:	30154h–30157h (RO; debug path)
Default Value:	00000000h
Access:	see address offset above
Size:	32 bits

Bit	Description
31	**Overlay Destination Color Key Enable:** Destination keying when enabled forces the overlay surface Z order to be below the primary display. Pixels that match the key value (see above) become transparent and the overlay becomes visible at that pixel. The combination of source key enabled and destination key enabled is not a useful operational mode. 1 = destination color key is enabled 0 = destination color key is disabled
30	**Enable Constant Alpha:** Overlay constant alpha provides a way to apply an alpha value to all overlay pixels. Each pixel color channel is multiplied by the constant alpha before proceeding to the blender. This can be used to create fade out effects. This is intended for CE device use where the overlay might still be used to generate video output. 0 – Overlay Constant Alpha is disabled 1 – Overlay Constant Alpha is enabled
29:24	Reserved: MBZ
23:0	**Destination Color Key Mask:** [23:16] = Red [15:08] = Green [07:00] = Blue 0 = Bits that are active participants in the compare. 1 = Bits that are not active participants in the compare. A mask of all ones will disable the color key compare (as if all color channel values match).

Table 3-7. Destination Color Key Mask Programming in Each Supported Graphics Mode

	Color Key Mask			
Display Mode	**Red**	**Green**	**Blue**	**Note**
8-bit Indexed	11111111	11111111	00000000	Only blue channel is compared.
15-bit RGB	00000111	00000111	00000111	Compare 5 MSBs of each color channel only
16-bit RGB	00000111	00000011	00000111	Compare 5 or 6 MSBs of color channel only
32-bit RGB	00000000	00000000	00000000	Compare 8:8:8 data of the 32-bpp data Applies to all 32 bpp data formats except formats with alpha in use.

3.2.15 Overlay Source Chroma Key Registers

There is an overlay source key per overlay stream, which is used on a pixel basis. The Source comparison occurs after the horizontal zooming, but in the YUV formats before the color space conversion. Source Chroma Key is also referred as **Source Color Key**. Unlike Destination Color Key, Source Chroma Key uses a high-key value and a low-key value to specify a range. If the source data (overlay) is within the range, then the primary display is selected.

If the overlay is in RGB mode, the most significant bits are duplicated on the least significant to form 8-bit channels before the source key comparison is made. The RGB range expansion follows the rules in the following table.

Table 3-8. RGB Range Expansion Rules

Overlay Source Format	Source	Expanded Value
RGB15	R[4:0]	R[4:0 ‘ 4:2]
	G[4:0]	G[4:0 ‘ 4:2]
	B[4:0]	B[4:0 ‘ 4:2]
RGB16	R[4:0]	R[4:0 ‘ 4:2]
	G[5:0]	G[5:0 ‘ 5:4]
	B[4:0]	B[4:0 ‘ 4:2]

While it makes no sense to enable both destination and source color keying for the overlay, destination Color Key failing takes precedence over the Source Chroma Key failing. If both fail, then the primary display is selected.

3.2.15.1 SCHRKVH—Source Chroma Key Value High Register

Memory Address Offset:	58h–5Bh (R/W)
On-chip Reg. Mem Addr Offset:	30158h–3015Bh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:24	Reserved: MBZ
23:16	**Y/Green Source Key High (YG_SKEY_H):** This 8-bit field specifies the high end Y/Green value (less than or equal to) for the overlay source exclusion pixel data. When the corresponding Y/Green Source Key Mask Enables bit in SCLRKM register is set, an overlay value greater than this field fails the comparison and the overlay pixel passes. Otherwise, the overlay pixel becomes transparent. YUV format: Y value in [0,255] range. RGB format: Green value in [0,255] range.
15:8	**U/Blue Source Key High (UB_SKEY_H):** This field specifies the high end U/Blue value (less than or equal to) for the overlay source exclusion pixel data. When the corresponding U/Blue Source Key Mask Enables bit in SCLRKM register is set, an overlay value greater than this field fails the comparison and the overlay pixel passes. Otherwise, the overlay pixel becomes transparent. YUV format: U value in excess-128 format. RGB format: Blue value in [0,255] range.
7:0	**V/Red Source Key High (VR_SKEY_H):** This field specifies the high end V/Red value (less than or equal to) for the overlay source exclusion pixel data. When the corresponding V/Red Source Key Mask Enables bit in SCLRKM register is set, an overlay value greater than this field fails the comparison and the overlay pixel passes. Otherwise, the overlay pixel becomes transparent. YUV format: V value in excess-128 format. RGB format: Red value in [0,255] range.

3.2.15.2 SCHRKVL—Source Chroma Key Value Low Register

Memory Address Offset:	5Ch–5Fh (R/W)
On-chip Reg. Mem Addr Offset:	3015Ch–3015Fh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:24	Reserved: MBZ
23:16	**Y/Green Source Key Low (YG_SKEY_L):** This field specifies the low end Y/Green value (greater than or equal to) for the overlay source exclusion pixel data. When the corresponding Y/Green Source Key Mask Enables bit in SCLRKM register is set, an overlay value less than this field fails the comparison and the overlay pixel passes. Otherwise, the overlay pixel becomes transparent. YUV format: Y value in [0,255] range. RGB format: Green value in [0,255] range.
15:8	**U/Blue Source Key Low (UB_SKEY_L):** This field specifies the low end U/Blue value (greater than or equal to) for the overlay source exclusion pixel data. When the corresponding U/Blue Source Key Mask Enables bit in SCLRKM register is set, an overlay value less than this field fails the comparison and the overlay pixel passes. Otherwise, the overlay pixel becomes transparent. YUV format: U value in excess-128 format. RGB format: Blue value in [0,255] range.
7:0	**V/Red Source Key Low (VR_SKEY_L):** This field specifies the low end V/Red value (greater than or equal to) for the overlay source exclusion pixel data. When the corresponding V/Red Source Key Mask Enables bit in SCLRKM register is set, an overlay value less than this field fails the comparison and the overlay pixel passes. Otherwise, the overlay pixel becomes transparent. YUV format: V value in excess-128 format. RGB format: Red value in [0,255] range.

3.2.15.3 SCHRKEN—Source Chroma Key Enable Register

Memory Address Offset:	60h–63h (R/W)
On-chip Reg. Mem Addr Offset:	30160h–30163h (RO; debug path)
Default Value:	0000h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:27	Reserved: MBZ
26:24	**Source Chroma Key Enable**: Each bit enables one channel of source chroma key range comparison. If the bit is a one, the comparison result is used otherwise it is ignored. Bit 26 = Enables Y/Green Comparison [23:16] Bit 25 = Enables U/Blue Comparison [15:8] Bit 24 = Enables V/Red Comparison [7:0]
23:8	Reserved: MBZ
7:0	**Overlay Constant Alpha Value:** This field provides the alpha value when constant alpha is enabled. A value of FF means fully opaque and a value of zero means fully transparent. Values in between those values allow for a blending of overlay with other surfaces.

3.2.15.4 OCONFIG—Overlay Configuration Register

Memory Address Offset:	64h–67h (R/W)
On-chip Reg. Mem Addr Offset:	30164h–30167h (RO; debug path)
Default Value:	0000h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:27	Reserved: MBZ
26:19	Reserved for Slot Time: Programmed for UMA to allow for improved CPU performance.
18	**Overlay Pipe Select:** Changing this bit causes the display to take VBLANK events from both pipes (in the proper order) to complete the change. An untrusted overlay can be enabled on a trusted pipe, the Z-order will place all trusted planes above the overlay in Z-order. **Alviso/Grantsdale-G Errata GRG04**: This bit may not be toggled when BOTH pipes are enabled unless the Overlay Enable bit in the Overlay Command Register is cleared to '0'. See further description under Overlay Enable bit. 0 = Overlay is assigned to display pipe A 1 = Overlay is assigned to display pipe B
17	Reserved: MBZ
16	**Overlay Secondary Gamma Enable (GAMMA2_ENBL):** The secondary gamma refers to the gamma RAM in the graphics pipe that overlay is associated with. This field provides independent control of the secondary gamma logic for overlay pixels. When this bit is in use, software MUST make sure that its correspondence compatibility bit (bit 26 of register PIPEACONF at 70008h) is set to zero all the time. 0 = Overlay pixel data bypasses secondary gamma correction logic (default). 1 = Overlay pixel data is gamma corrected by the secondary correction logic.
15:6	Reserved: MBZ
5	**YUV2RGB Conversion Mode (CSC_MODE):** This bit sets the color space conversion mode for the overlay plane. When bit 4 (CSC_BYPASS) is set, this bit is ignored in the CSC logic since it is in bypass mode. 0 = ITU-R Recommendation BT.601 1 = ITU-R Recommendation BT.709
4	**YUV2RGB Conversion Bypass (CSC_BYPASS)**: This bit sets the overlay data path to bypass the YUV-to-RGB conversion logic. This bit should only be set when the overlay data is output to compliant external TV encoders in overlay native YUV444 format. The color adjustment controls, programmed through brightness, contrast and saturation registers, are **still active**. In CSC_BYPASS mode, software should also program the overlay gamma to a linear ramp, which is the gamma default values. The output YUV data from Overlay contains 10-bit unsigned Y value (possibly in the full 0-255 range) and 10-bit UV data in excess-512 format. The exact values depend on the mode of the Overlay Color Control Output. Converting to YCrCb (range readjustment) for 656 TV out is handled separately by the TV Out control logic. TV Out is available in [DevCL] 0 = Enable the YUV to RGB Color Space Conversion Logic (default) 1 = Bypass (Disable) the YUV to RGB Color Space Conversion Logic

Bit	Description
3	**Reserved Color Control Output Mode (CC_OUT):** Sets the output pixel resolution of the Color Control Unit (including the piecewise-linear gamma correction logic) to be either 10-bit or 8-bit. Proper rounding is applied to the pixels. CC_OUT should be set to 1 to enable correct rounding to 8-bit if the backend (including the blending unit) operates in 8-bit mode. This mode takes affect whether gamma correction is enabled or not. 0 = 10-bit output. (default) 1 = 8-bit output.
2	**Single Request Mode** 0 – Single Request disabled 1- Single Request enabled
1	Reserved: MBZ (reserved for future overlay line buffer configuration)
0	**Line Buffer Configuration (LINE_CONFIG)**: Sets the overlay line buffer configuration. Using the 2-line mode reduces the peak bandwidth demands of the overlay when down scaling. Configuring the cache for the 2-line mode increases the source line size that will fit in the line buffers. See the appendix for details of the cache configurations and the source size limits based on format and configuration. Other limits are based on available bandwidth. 0 = Two line buffers 1 = Three line buffers

3.2.15.5 OCOMD—Overlay Command Register

Memory Address Offset: 68h–6Bh (R/W)
On-chip Reg. Mem Addr Offset: 30168h–3016Bh (RO; debug path)
Default Value: 00h
Access: see address offset above
Size: 32 bits

This register and the Overlay Configuration register provide the basic programming options that the overlay engine needs to begin its work.

Bit	Description
31:20	Reserved: MBZ
19	**Tiled Surface:** This bit indicates that the overlay surface data is in tiled memory. The tile pitch is specified in bytes in the OSTRIDE register. Only X tiling is supported for overlay surfaces. When this bit is set, it affects the hardware interpretation of the OBUF and OSTART registers. 0 = Overlay surfaces use linear memory 1 = Overlay surfaces use X-tiled memory
18:17	**Mirroring**: Mirroring cause either a horizontal or vertical reversal in the displayed image. Note that when both horizontal and vertical mirroring is enabled it is a 180 degree rotation. Mirroring affects the buffer address values used. See buffer address description for details. 00 = Normal orientation 01 = Horizontal Mirroring 10 = Vertical Mirroring 11 = Both Horizontal and Vertical Mirroring (180 degree rotate)
16	Reserved: MBZ
15:14	**Packed YUV 4:2:2 Byte Order (ORDER422):** Affects the byte order for Packed YUV 4:2:2 data. This allows for the support of multiple YUV 4:2:2 formats. It must be set to zero for other data formats. 00 = Normal (in YUYV order with FOURCC code “YUY2”) 01 = UV Swap (in YVYU order) 10 = Y Swap (in UYVY order with FOURCC code “UYVY”) 11 = Y and UV swap (in VYUY order)

Bit	Description
13:10	**FORMAT: Source Format.** 0XXX = Reserved 1000 = Packed YUV 4:2:2 (Field ORDER422 in this register determines the byte order of this format) 1001 = Reserved 101X = Reserved 1100 = Planar YUV 4:2:0 (MPEG-1 or 2 based on UV initial phase) 1101 = Planar YUV 4:2:2 (e.g., for DCT-domain downsampled MPEG2 video) 1110 = Planar YUV 4:1:0 or YUV 4:1:1 planar 1111 = Reserved In all YUV formats, UV source values are in excess 128 format, in which value 128 stands for intensity of 0 for the color components.
9	Reserved: (
8	Reserved: MBZ (Enabling Autoflip (AUTOFLIP):)
7	**Field-Synchronized Flip Enable (TVSYNCFLIP_ENBL)**: This bit enables the overlay flip that is synchronized with the display field. 0 = Normal Overlay Flip (standard). The TVSYNCFLIP_PARITY bit is ignored. 1 = Field-Synchronized Overlay Flip
6	Reserved: MBZ
5	**Buffer Display/Flip Type (BUF_TYPE)**: This bit affects the buffer addressing used for buffer display and the use of the initial vertical phase. Frame mode starts addressing at the value contained in the buffer address register and increments/Decrements by stride as it advances from line to line. Initial phase selection is based on the buffer and the vertical Initial phase select bit. Field mode uses the Active Field bit (bit 1 of this register) and the reverse Y bit to determine if the start address should be the value in the start address register or the start address register plus/minus stride. Field mode will increment/decrement the address by two times the stride as it increments from line to line. Initial phase selection is based on the field and the vertical Initial phase select. 0 = Frame Mode (or called Non-Interleaved Buffer Mode) 1 = Field Mode (or called Interleaved Buffer Mode)
4	0 = Normal mode 1 = Test mode
3:2	**Active Buffer Pointer (ACT_BUF)**: Selects which overlay buffer for display. This determines which buffer will be displayed when the overlay is enabled and the Ignore Buffer/Field bit (bit 4) of this register is cleared (to 0). It will otherwise be ignored and the internal buffer value (with the previously loaded value) will be used. The internal buffer value is readable through the status register (OC_BUF). 00 = Buffer 0 01 = Buffer 1 10 = Reserved 11 = Reserved

Bit	Description
1	**Active Field Select (ACT_F):** Selects which field in an interleaved overlay buffer for display. This determines which field will be displayed when the overlay is enabled and the Ignore Buffer/Field bit (bit 4) was cleared (to 0). It will otherwise be ignored and the internal field value (with the previously loaded value) will be used. The internal buffer value is readable through the status register (OC_FIELD). This bit is ignored if Buffer Display/Flip Type is in Frame Mode (non-Interleaved Buffer mode). 0 = Field 0 1 = Field 1
0	**Overlay Enable (OV_ENBL):** Changing this bit from a Zero to a One will cause the overlay to begin display after the next qualified flip event. This can only be done in conjunction with a flip packet that enables the reconfiguration of the cache using the flush flags. A disable (from 1 to 0) will cause the overlay to stop displaying on this current display/overlay VBLANK. When Overlay is flipped from one display pipe to another, it will be automatically temporarily disabled to allow for the switch. When disabled, it will cause the overlay to enter a low power state. Overlay when enabled, automatically prevents the special C3 display fetch mode from being activated. There is an override for the enable of this plane in the Pipe Configuration register. Overlay can also be disabled from the pipe control register of the pipe that overlay is assigned to. **Full on/off changes to the overlay require the shared render cache to be re-configured. This must be done through the command stream according to the packet definition. You can leave the cache intact while disabling the overlay (command register). You cannot enable the overlay without having previously configured the cache.** **Alviso/Grantsdale-G Erratum GRG04**: This bit must be manually disabled by clearing to '0' when toggling the Overlay Pipe Select bit **if BOTH display pipes are enabled**. This erratum will be fixed on future products. There are two cases: **1) Overlay is disabled prior to enabling and switching to other pipe:** a) Issue Overlay Flip ON request with Overlay Enable bit cleared to '0' and Overlay Pipe Select bit toggled b) Issue Overlay Flip CONTINUE request with Overlay Enable bit set to '1' **2) Overlay is enabled prior to switching to other pipe:** a) Issue Overlay Flip CONTINUE request with Overlay Enable bit cleared to '0' and Overlay Pipe Select bit toggled b) Issue Overlay Flip CONTINUE request with Overlay Enable bit set to '1' **Note**: A wait-on-flip event is always required between back-to-back overlay flip requests. 0 = Overlay Plane Disable (No display or memory fetches) (default) 1 = Overlay Plane Enable (Requires cache to be configured first)

3.2.15.6 OSTART_0Y— Overlay Surface Y 0 Base Address Register

Memory Address Offset:	70h (R/W)
On-chip Reg. Mem Addr Offset:	30170h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:0	**Overlay Surface Y 0 Base Address:** This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the OBUF_0Y register. When the surface is in linear memory, panning is specified using a linear offset in the OBUF_0Y register. This address must be 4K aligned. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. These registers do not have a readback path through On-chip Reg.

3.2.15.7 OSTART_1Y— Overlay Surface Y 1 Base Address Register

Memory Address Offset:	74h (R/W)
On-chip Reg. Mem Addr Offset:	30174h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:0	**Overlay Surface Y 1 Base Address:** This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the OBUF_0Y register. When the surface is in linear memory, panning is specified using a linear offset in the OBUF_0Y register. This address must be 4K aligned. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. These registers do not have a readback path through On-chip Reg.

3.2.15.8 OSTART_0U— Overlay Surface U 0 Base Address Register

Memory Address Offset:	78h (R/W)
On-chip Reg. Mem Addr Offset:	30178h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:0	**Overlay Surface U 0 Base Address:** This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the OBUF_0Y register. When the surface is in linear memory, panning is specified using a linear offset in the OBUF_0Y register. This address must be 4K aligned. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. These registers do not have a readback path through On-chip Reg.

3.2.15.9 OSTART_0V— Overlay Surface V 0 Base Address Register

Memory Address Offset:	7Ch (R/W)
On-chip Reg. Mem Addr Offset:	3017Ch (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:0	**Overlay Surface V 0 Base Address:** This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the OBUF_0Y register. When the surface is in linear memory, panning is specified using a linear offset in the OBUF_0Y register. This address must be 4K aligned. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. These registers do not have a readback path through On-chip Reg.

3.2.15.10 OSTART_1U— Overlay Surface U 1 Base Address Register

Memory Address Offset:	80h (R/W)
On-chip Reg. Mem Addr Offset:	30180h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:0	**Overlay Surface U 1 Base Address:** This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the OBUF_0Y register. When the surface is in linear memory, panning is specified using a linear offset in the OBUF_0Y register. This address must be 4K aligned. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. These registers do not have a readback path through On-chip Reg.

3.2.15.11 OSTART_1V— Overlay Surface V 1 Base Address Register

Memory Address Offset:	84h (R/W)
On-chip Reg. Mem Addr Offset:	30184h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:0	**Overlay Surface V 1 Base Address:** This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the OBUF_0Y register. When the surface is in linear memory, panning is specified using a linear offset in the OBUF_0Y register. This address must be 4K aligned. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. These registers do not have a readback path through On-chip Reg.

3.2.15.12 OTILEOFF_0Y— Overlay Surface Y 0 Tiled Offset Register

Memory Address Offset:	88h (R/W)
On-chip Reg. Mem Addr Offset:	30188h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

This register specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

Bit	Description
31:28	Reserved. Write as zero
27:16	**Source Start Y-Position.** These 12 bits specify the vertical position in lines of the beginning of the overlay source window relative to the overlay surface. When mirroring vertically (Y backward), this field specifies the vertical position of the lower right corner relative to the end of the overlay data in the unmirrored orientation.
15:12	Reserved. Write as zero
11:0	**Source Start X-Position.** These 12 bits specify the horizontal offset in pixels of the beginning of the overlay source window relative to the overlay surface. When mirroring horizontally (X backward), this offset specifies the last pixel of the line of the overlay data in its unmirrored orientation.

3.2.15.13 OTILEOFF_1Y— Overlay Surface Y 1 Tiled Offset Register

Memory Address Offset:	8Ch (R/W)
On-chip Reg. Mem Addr Offset:	3018Ch (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

This register specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

Bit	Description
31:28	Reserved. Write as zero
27:16	**Source Start Y-Position.** These 12 bits specify the vertical position in lines of the beginning of the overlay source window relative to the overlay surface. When mirroring vertically (Y backward), this field specifies the vertical position of the lower right corner relative to the end of the overlay data in the unmirrored orientation.
15:12	Reserved. Write as zero
11:0	**Source Start X-Position.** These 12 bits specify the horizontal offset in pixels of the beginning of the overlay source window relative to the overlay surface. When mirroring horizontally (X backward), this offset specifies the last pixel of the line of the overlay data in its unmirrored orientation.

3.2.15.14 OTILEOFF_0U— Overlay Surface U 0 Tiled Offset Register

Memory Address Offset:	90h (R/W)
On-chip Reg. Mem Addr Offset:	30190h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

This register specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

Bit	Description
31:28	Reserved. Write as zero
27:16	**Source Start Y-Position.** These 12 bits specify the vertical position in lines of the beginning of the overlay source window relative to the overlay surface. When mirroring vertically (Y backward), this field specifies the vertical position of the lower right corner relative to the end of the overlay data in the unmirrored orientation.
15:12	Reserved. Write as zero
11:0	**Source Start X-Position.** These 12 bits specify the horizontal offset in pixels of the beginning of the overlay source window relative to the overlay surface. When mirroring horizontally (X backward), this offset specifies the last pixel of the line of the overlay data in its unmirrored orientation.

3.2.15.15 OTILEOFF_0V— Overlay Surface V 0 Tiled Offset Register

Memory Address Offset:	94h (R/W)
On-chip Reg. Mem Addr Offset:	30194h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

This register specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

Bit	Description
31:28	Reserved. Write as zero
27:16	**Source Start Y-Position.** These 12 bits specify the vertical position in lines of the beginning of the overlay source window relative to the overlay surface. When mirroring vertically (Y backward), this field specifies the vertical position of the lower right corner relative to the end of the overlay data in the unmirrored orientation.
15:12	Reserved. Write as zero
11:0	**Source Start X-Position.** These 12 bits specify the horizontal offset in pixels of the beginning of the overlay source window relative to the overlay surface. When mirroring horizontally (X backward), this offset specifies the last pixel of the line of the overlay data in its unmirrored orientation.

3.2.15.16 OTILEOFF_1U— Overlay Surface U 1 Tiled Offset Register

Memory Address Offset:	98h (R/W)
On-chip Reg. Mem Addr Offset:	30198h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

This register specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

Bit	Description
31:28	Reserved. Write as zero
27:16	**Source Start Y-Position.** These 12 bits specify the vertical position in lines of the beginning of the overlay source window relative to the overlay surface. When mirroring vertically (Y backward), this field specifies the vertical position of the lower right corner relative to the end of the overlay data in the unmirrored orientation.
15:12	Reserved. Write as zero
11:0	**Source Start X-Position.** These 12 bits specify the horizontal offset in pixels of the beginning of the overlay source window relative to the overlay surface. When mirroring horizontally (X backward), this offset specifies the last pixel of the line of the overlay data in its unmirrored orientation.

3.2.15.17 OTILEOFF_1V— Overlay Surface V 1 Tiled Offset Register

Memory Address Offset:	9Ch (R/W)
On-chip Reg. Mem Addr Offset:	3019Ch (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

This register specifies the panning for the overlay surface. Bit 19 of OCOMD specifies whether the overlay surfaces are in linear or tiled memory. When the surface is in linear memory, the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

Bit	Description
31:28	Reserved. Write as zero
27:16	**Source Start Y-Position.** These 12 bits specify the vertical position in lines of the beginning of the overlay source window relative to the overlay surface. When mirroring vertically (Y backward), this field specifies the vertical position of the lower right corner relative to the end of the overlay data in the unmirrored orientation.
15:12	Reserved. Write as zero
11:0	**Source Start X-Position.** These 12 bits specify the horizontal offset in pixels of the beginning of the overlay source window relative to the overlay surface. When mirroring horizontally (X backward), this offset specifies the last pixel of the line of the overlay data in its unmirrored orientation.

3.2.16 Overlay Scaling Registers

3.2.16.1 Reserved FASTHSCALE—Fast Horizontal Downscale Register

Memory Address Offset:	A0h (R/W)
On-chip Reg. Mem Addr Offset:	301A0h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:21	Reserved: MBZ
20:16	**RESERVED (was YRGB_HSCALE):** The fast scale factors must be set so that the output of the fast scalar has a luminance to chrominance ratio of either 2:1 or 1:1. The original source format and the two fast scale factors determine that ratio. 0000 = No fast horizontal scaling (only precision) NNNN = Down scale by 2^N (N=1-5 for 3-line Vertical filter, N=1-6 for 2-line Vertical filter)
15:5	Reserved: MBZ
4:0	**RESERVED (was UV_HSCALE):** 0000 = No fast horizontal scaling (only precision) NNNN = Down scale by 2^N (N=1-5 for 3-line Vertical filter, N=1-6 for 2-line Vertical filter)

3.2.16.2 UVSCALEV—UV Vertical Downscale Integer Register

Memory Address Offset:	A4h (R/W)
On-chip Reg. Mem Addr Offset:	301A4h (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	32 bits

Bit	Description
31:27	Reserved: MBZ
26:16	**Y Vertical Scale Integer (Y_VINT):** This field is used for vertical downscale. It specifies the integer portion of the scale factor for packed data or the Y data in planar modes. A zero in this field indicates a vertical upscale operation.
15:11	Reserved: MBZ
10:0	**UV Vertical Scale Integer (UV_VINT):** Used only in YUV planar modes. This field is used only for vertical downscale. It specifies the integer portion of the scale factor for U/V data. The chrominance data may be subsampled. A zero value specifies a vertical upscale operation.

3.2.17 Overlay Polyphase Filter Coefficient Registers

The filter coefficient mantissa sizes are as follows:

Coefficient Type	Number x 17	Mantissa Size in Bits
Vertical Y Center Tap	1	8
Vertical Y Others	2	6
Vertical UV	3x2	6
Horizontal Y Center Tap	1	9
Horizontal Y Others	4	7
Horizontal UV Center Tap	1x2	9
Horizontal UV Others	2x2	7

Coefficients are stored in order of taps ordered right to left (for horizontal) and top to bottom (for vertical) followed by phases starting with phase zero. For products with three line mode, vertical coefficients for the 2-line mode should use zero for the center tap and phase and 1-phase for the top and bottom coefficients.

<table>
<tr><td>31</td><td>30</td><td></td><td>28</td><td>27</td><td>22/21/19</td><td>21/20/18</td><td>16</td></tr>
<tr><td>Sign</td><td colspan="3">Exponent</td><td colspan="2">Mantissa</td><td colspan="2">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>16</td><td>15</td><td>14</td><td>13</td><td>8/7/6</td><td>7/6/5</td><td>0</td></tr>
<tr><td>Sign</td><td colspan="3">Exponent</td><td colspan="2">Mantissa</td><td colspan="2">Reserved</td></tr>
</table>

3.2.17.1 Y_VCOEFS—Overlay Y Vertical Filter Coefficient Registers

Memory Address Offset:	200h–2FFh (R/W)
On-chip Reg. Mem Addr Offset:	30300h–303FFh (not readable)
Default Value:	00h
Access:	see address offset above
Size:	3 taps x 17 phases x 32/2 bits

3.2.17.2 Y_HCOEFS—Overlay Y Horizontal Filter Coefficient Registers

Memory Address Offset:	300h–4FFh (R/W)
On-chip Reg. Mem Addr Offset:	30400h–305FFh (not readable)
Default Value:	00h
Access:	see address offset above
Size:	5 taps x 17 phases x 32/2 bits

3.2.17.3 UV_VCOEFS—Overlay UV Vertical Filter Coefficient Registers

Memory Address Offset:	500h–5FFh (R/W)
On-chip Reg. Mem Addr Offset:	30600h–306FFh (not readable)
Default Value:	00h
Access:	see address offset above
Size:	3 taps x 17 phases x 32/2 bits

3.2.17.4 UV_HCOEFS—Overlay UV Horizontal Filter Coefficient Registers

Memory Address Offset:	600h–6FFh (R/W)
On-chip Reg. Mem Addr Offset:	30700h–307FFh (RO; debug path)
Default Value:	00h
Access:	see address offset above
Size:	3 taps x 17 phases x 32/2 bits

Bit	Description
31	**Sign:** 0 = Positive value 1 = Negative value
30:28	**Exponent:** 000 = b.bbbbbb (1.6) 001 = .bbbbbbb (0.7) 010 = .0bbbbbbb (0.8) 011 = .00bbbbbbb (0.9) 1xx = Reserved

Bit	Description
27:21	**Mantissa:** The mantissa varies in size based on which filter and which tap is being specified. Vertical filters use 6 bits except Y center tap is 8-bits Horizontal filters use 7-bits except Y/UV center taps are 9-bits
20:16	Reserved: MBZ
15	**Sign:** 0 = Positive value 1 = Negative value
14:12	**Exponent:** This field determines the placement of the binary point for each coefficient. Using the value of 000, the coefficient has a maximum value of just less than two. 000 = 1.6 001 = 0.7 010 = 0.8 011 = 0.9 1xx = Reserved
11:6/5	**Mantissa**:
5/4:0	Reserved: MBZ

§§

4 TV-Out [DevCL]

The TV-Out register block is offset from the MMADR by 0x68000. The registers are broken out into several sections based on function. There are many reserved testmode bits in this register block, in general software should preserve the value of those bits (unless otherwise specified) as the enabling of those functions may take place at boot time.

Address Offset	Register Name
68000	TV out Control [DevCL]
68004	TV DAC Control / Status
68008	reserved
6800C	reserved
68010	Color Space Convert Y
68014	Color Space Convert Y2
68018	Color Space Convert U
6801C	Color Space Convert U2
68020	Color Space Convert V
68024	Color Space Convert V2
68028	Color Knobs
6802C	Color Level Control
68030	H Control 1 - Hsync Htotal
68034	H Control 2 - H Burst Control
68038	H Control 3 - Blanking
6803C	V Control 1 - NBR & VI end
68040	V Control 2 - Vsync Control
68044	V Control 3 - Equalization Control
68048	V Control 4 - v burst field 1
6804C	V Control 5 - v burst field 2
68050	V Control 6 - v burst field 3
68054	V Control 7 - v burst field 4
68058	reserved
68060	SC Control 1 - enables, Burst level, SC DDA1
68064	SC Control 2 - SC DDA2
68068	SC Control 3 - SC DDA3
6806C	reserved

Address Offset	Register Name
68070	Window Position
68074	Window Size
68078-6807C	reserved
68080	Filter Control 1 - Mode and H frac
68084	Filter Control 2 - Vert scale
68088	Filter Control 3 - Vert Initial Phase
6808C	Cosine ROM
68090	CC Control
68094	CC Data Field 1
68098	CC Data Field 2
6809C-680AC	reserved
680B0	WSS Control
680B4	WSS Data
680B8-680FC	reserved
68100	H Filter Y Coefficients
68200	H Filter C Coefficients
68300	V Filter Y Coefficients
68400	V Filter C Coefficients

4.1 TV-Out Control Registers [DevCL]

4.1.1 TV_CTL—TV-Out Control

Address Offset:	68000–68003h
Default Value:	00000000h

This register contains controls for the basic functional modes of the TV-Out logic. When re-enabling the encoder after disabling, software should ensure that at least one vertical blank has occurred on the attached pipe before turning the encoder back on or turning pipe off. When operating in panel fitter mode the TV encoder is disabled and the filtered video is routed back to the ports assigned to the pipe selected in bit 30.

Bit	Description
31	**ENC_ENABLE:** Enables the encoder and associated logic 0 = encoder is off (default) 1 = encoder is on.
30	**ENC_PIPESEL:** Selects the pipe to which the encoder is attached 0 = pipe A 1 = pipe B
29:28	**ENC_OUTPUT_MODE[1:0]:** Selects analog output format, also configures the DAC output voltage levels 00 = Composite (default) uses DAC A only 01 = SVideo (Y data on DAC B, C data on DAC C) 10 = Component (Y : DAC B, Pb : DAC A, Pr : DAC C) 11 = Combo mode: [testrmode only, should not be used in production] Composite (DAC A) & SVideo (DAC B&C)
27	Reserved
26:24	Reserved. Read as zero
23	**RGB:** Enables RGB output. Only used in SMPTE253. ENC_OUTPUT_MODE must be set to 10 for this bit to be valid. 0 = Normal sync (default) 1 = RGB (R: DAC C, G: DAC B, B: DAC A)
22	**3CH_SYNC:** Enables sync generation on all channels. Used in SMPTE253, can also be used in YPbPr 0 = Normal sync (default) 1 = 3 channel sync
21	**TRILEVEL_SYNC:** Enables trilevel sync generation 0 = Normal sync (default) 1 = Trilevel sync
20	**SLOW_SYNC:** Enables slow sync generation for NTSC & PAL. 0 = Normal sync (default) 1 = Slow sync for NTSC & PAL

Bit	Description
19:18	**OVERSAMPLE[1:0]:** Selects between oversampling modes 00 = 4x oversample (for 480/576p) 01 = 2x oversample (for 720p & 1080i) 10 = no oversampling (for 1080p testmode) 11 = 8x oversample (for all Standard Def interlaced)
17	**PROG_INT:** Selects between progressive and interlaced scanning 0 = interlaced (default) 1 = progressive
16	**PAL_BURST:** Enables the PAL switch for colorburst phase reversal. This bit must be set in all PAL modes except PAL/M 0 = NTSC burst (default) 1 = PAL burst
15	**FIELD_SWAP: (Testmode)** 0 = Normal 1 = Inverted Fields
14:12	**YC_SKEW[2:0]:** An adjustment for Luminance timing relative to chroma. Actual skew amount depends on the output clock. Amounts shown indicate the amount of delay applied to the Y component. 000 = Normal (in phase) 001 = +0.5 pixel (advanced) 010 = +1.0 pixel 011 = +1.5 pixel 100 = +2.0 pixel 101 = +2.5 pixel others = reserved
11	**PF2_ENABLE:** Enable Panel Fitter 2 Mode* (PF2) [DevCL] 0 = PF2 is off (default) 1 = PF2 is on. ENC_ENABLE (bit 31) must also be set to 0.
10:9	**TP_SIZE:** Adjusts testpattern size for HD testing 00 = 640x480 01 = 1280x720 10 = 1920x1080 11 = reserved
8:6	**VGA_THRT[1:0]: (Testmode):** VGA data throttling override. Software must preserve the state of these bits: 000 = 50% Throttling (default) 001 = 33% (clk*2/3) 010 = 66% (clk/3) 011 = 75% (clk*1/4) 100 = 80% (clk*1/5) 111 = Throttling disabled

Bit	Description
5:4	**FUSE_STATE: (READ-ONLY).** These two bits indicate the state of the TV-Out function fuses. 00 = All features enabled (default) 01 = TV enabled but Macrovision disabled. For MV controlled countries. 1X = TV-Out function disabled. If TV is in the this mode writing to the TV-Out registers will have no effect on the TV-Out logic.
3:0	**Reserved.**

*Notes on using the filter as a panel fitter:

The filter will be attached to the pipe specified in bit 30 of this register. Only the filter control registers will have any affect, all other TV registers must be left in the default state. It is recommended to use the auto scaling mode (0x68080[31]=1). Scaling works the same as in the primary panel fitter, pipe source size is automatically scaled fit into the area defined by the TV_WIN_POS and TV_WIN_SIZE registers within the pipe's active area. The pipe must be off when enabling or disabling the scaler (enabling/disabling Panel Fitter 2 Mode), however while the scaler is enabled the pipe source size is allowed to be changed.

4.1.2 TV_DAC—TV DAC Control / Status [DevCL]

Address Offset: 68004–68007h
Default Value: U0000000h

This register contains controls for the basic functional modes of the TVDAC logic. It also contains status information for detecting the connection type. TV-Out must be enabled or set to monitor sense for the Sense logic to work.

Bit	Description
31	**TVDAC_STATE_CHG [R/O]:** Enables DAC state change detection logic. A sticky bit cleared by state change detection being disabled. 0 = unchanged since enabled 1 = changed since enabled
30	**TVDAC_A_SENSE [R/O] :** Indicates the voltage level of the TV DAC A is above threshold selected in TVDAC A SENSE CTL 0 = below threshold 1 = above threshold
29	**TVDAC_B_SENSE [R/O] :** Indicates the voltage level of the TV DAC B is above threshold selected in TVDAC B SENSE CTL 0 = below threshold 1 = above threshold
28	**TVDAC_C_SENSE [R/O] :** Indicates the voltage level of the TV DAC C is above threshold selected in TVDAC C SENSE CTL 0 = below threshold 1 = above threshold
27	**TVDAC_STATE_CHG_EN :** Enables DAC state change detection logic 0 = disabled 1 = enabled
26	**TVDAC_A_SENSE_CTL :** 0 = Low sense level (used for CVBS and Y signals) 1 = High sense level (used for C/Pr/Pb signals and monitor detect)
25	**TVDAC_B_SENSE_CTL :** 0 = Low sense level (used for CVBS and Y signals) 1 = High sense level (used for C/Pr/Pb signals and monitor detect)
24	**TVDAC_C_SENSE_CTL :** 0 = Low sense level (used for CVBS and Y signals) 1 = High sense level (used for C/Pr/Pb signals and monitor detect)
23:8	Reserved. Software must preserve the state of these bits

Bit	Description
7	**DAC_CTL_OVERRIDE :** DAC voltage level override values. This bit also overrides the state of ENC_ENABLE. 0 = use ENC_OUTPUT_MODE[1:0] to control the DACs 1 = use DAC_A_CTL, DAC_B_CTL, DAC_C_CTL, to control the voltage levels.
6	**ENC_TVDAC_SLEW : [Testmode].** Selects the DAC slew rate. Software must preserve the state of this bit. 0 = slow 1 = fast
5:4	**DAC_A_CTL[1:0] :** DAC voltage level override values. 00 = 1.3v 01 = 1.1v 10 = 0.7v 11 = DAC off
3:2	**DAC_B_CTL[1:0] :** DAC voltage level override values. 00 = 1.3v 01 = 1.1v 10 = 0.7v 11 = DAC off
1:0	**DAC_C_CTL[1:0] :** DAC voltage level override values. 00 = 1.3v 01 = 1.1v 10 = 0.7v 11 = DAC off

4.2 Color Conversion and Control Registers [DevCL]

These 5 registers contain the coefficients of the color space converter and the controls for the knob registers (Brightness, Contrast, Saturation, Hue).

The color space conversion registers are double buffered and are updated on each vertical sync following a write to CSC V2.

Attenuation registers are used to preserve dynamic range during filtering. These registers are 10 bits in b.bbbbbb... format and must not be programmed to values greaterthan 1. CSC coefficients must be normalized in order to use these registers.

Coefficients for the CSC are stored in exponent-mantissa format similar to the filter. Two CSC coefficients are stored in each dword, the tables below show the data packing in each of the registers. Three CSC coefficients use a special 12-bit format to have the ability to reach 1.0, these are GY, RV, and BU.

Bit	Description
10:9	**Exponent bits:** These bits are represented as 2^{-n} 00 = 1 or mantissa is 0.bbbbbbbbb 01 = 0.5 or mantissa is 0.0bbbbbbbbb 10 = 0.25 or mantissa is 0.00bbbbbbbbb 11 = 0.125 or mantissa is 0.000bbbbbbbbb
8:0	**Mantissa:**

Bit	Description
11:9	**Exponent bits:** These bits are represented as 2^{-n} 111 = 2 or mantissa is b.bbbbbbbb (can only use for 1.0) 000 = 1 or mantissa is 0.bbbbbbbbb 001 = 0.5 or mantissa is 0.0bbbbbbbbb 010 = 0.25 or mantissa is 0.00bbbbbbbbb 011 = 0.125 or mantissa is 0.000bbbbbbbbb others = reserved
8:0	**Mantissa:**

4.2.1 TV_CSC_Y—Color Space Convert Y

Address Offset: 68010–68013h
Default Value: 00000000h

Bit	Description
31:27	Reserved
26:16	**RY[10:0]:** CSC coefficient
15:12	Reserved
11:0	**GY[11:0]:** CSC coefficient

4.2.2 TV_CSC_Y2—Color Space Convert Y2

Address Offset: 68014–-68017h
Default Value: 00000000h

Bit	Description
31:27	Reserved
26:16	**BY[10:0]:** CSC coefficient
15:10	Reserved
9:0	**AY[9:0]:** Luma attenuation. Represents a number from 0 - 1.0 in 1.9 fixed point format

4.2.3 TV_CSC_U—Color Space Convert U

Address Offset: 68018–6801Bh
Default Value: 00000000h

Bit	Description
31:27	Reserved
26:16	**RU[10:0]:** CSC coefficient
15:11	Reserved
10:0	**GU[10:0]:** CSC coefficient

4.2.4 TV_CSC_U2—Color Space Convert U2

Address Offset: 6801C–6801Fh

Bit	Description
31:28	Reserved
27:16	**BU[11:0]:** CSC coefficient
15:10	Reserved
9:0	**AU[9:0]:** U attenuation (component video only). Represents a number from 0 - 1.0 in 1.9 fixed point format

4.2.5 TV_CSC_V—Color Space Convert V

Address Offset: 68020–68023h
Default Value: 00000000h

Bit	Description
31:28	Reserved
27:16	**RV[11:0]:** CSC coefficient
15:11	Reserved
10:0	**GV[10:0]:** CSC coefficient

4.2.6 TV_CSC_V2—Color Space Convert V2

Address Offset: 68024–68027h
Default Value: 00000000h

Bit	Description
31:27	Reserved
26:16	**BV[10:0]:** CSC coefficient
15:10	Reserved
9:0	**AV[9:0]:** V attenuation (component video only). Represents a number from 0 - 1.0 in 1.9 fixed point format

4.2.7 TV_CLR_KNOBS—Color Knobs

Address Offset: 68028–6802Bh
Default Value: 00000000h

If output is in RGB the CON & SAT fields must be programmed to the same value.

Bit	Description
31:24	**BRT[7:0]:** brightness modifier (2's comp value)
23:16	**CON[7:0]:** contrast modifier (2.6 fixed point value) [DevCL] Maximum programmed value is limited to 3.0.
15:8	**SAT[7:0]:** saturation modifier (2.6 fixed point value)
7:0	**HUE[7:0]:** hue modifier. 8-bit value represents a phase angle from 0 to 360 degrees.

4.2.8 TV_CLR_LEVEL—Color Level Control

Address Offset: 6802C–6802Fh
Default Value: 00000000h

Bit	Description
31:25	Reserved
24:16	**BLACK_LEVEL[8:0]:** The value to put out on the DAC for black
15:9	Reserved
8:0	**BLANK_LEVEL[8:0]:** The value to put out on the DAC for blanking

4.3 Timing Registers

The timing registers should only be modified while the encoder is disabled.

4.3.1 TV_H_CTL_1—H Control 1

Address Offset: 68030–68033h
Default Value: 00000000h

Bit	Description
31:29	Reserved
28:16	**HSYNC_END[12:0]:** The pixel number in which to stop the horizontal sync. Measured from the leading edge of H sync. Register is programmed with the actual pixel value.
15:13	Reserved
12:0	**HTOTAL[12:0]:** The total number of pixels in a line including sync and blanking. Register is programmed with one less than the actual number of pixels.

4.3.2 TV_H_CTL_2—H Control 2

Address Offset: 68034–68037h
Default Value: 00000000h

Bit	Description
31	**BURST_ENA:** Enable colorburst generation. 0 = color burst is disabled (for b/w or component enabled) 1 = colorburst is enabled
30:29	Reserved
28:16	**HBURST_START[12:0]:** The pixel number in which to start the colorburst. Measured from the leading edge of H sync. Register is programmed with one less than the actual pixel value.
15:8	Reserved
7:0	**HBURST_LEN[7:0]:** The length in pixels of the colorburst.

4.3.3 TV_H_CTL_3—H Control 3

Address Offset: 68038–6803Bh
Default Value: 00000000h

Bit	Description
31:29	Reserved
28:16	**HBLANK_END[12:0]:** The pixel number in which to end horizontal blanking. Measured from the leading edge of H sync. Register is programmed with one less than the actual pixel value.
15:13	Reserved
12:0	**HBLANK_START[12:0]:** The pixel number in which to start horizontal blanking. Measured from the leading edge of H sync. Register is programmed with one less than the actual pixel value.

4.3.4 TV_V_CTL_1—V Control 1

Address Offset: 6803C–6803Fh
Default Value: 00000000h

Bit	Description
31:27	Reserved
26:16	**NBR_END[10:0]:** Number of lines in in one field of the non-blanked region in interlace (or 1 frame in progressive mode). Register is programmed with one less than the actual number of lines.
15:14	Reserved
13:8	**VI_END_F1[5:0]:** The number of lines in the vertical interval of field 1. Register is programmed with one less than the actual number of lines.
7:6	Reserved
5:0	**VI_END_F2[5:0]:** The number of lines in the vertical interval of field 2. Register is programmed with one less than the actual number of lines.

4.3.5 TV_V_CTL_2—V Control 2

Address Offset: 68040–68043h
Default Value: 00000000h

Bit	Description
31:23	Reserved
22:16	**VSYNC_LEN[6:0]:** The duration of vertical sync in half lines
15	Reserved
14:8	**VSYNC_START_F1[6:0]:** The half line number of field 1 to start vertical sync. Register is programmed with one less than the actual number of half lines.
7	Reserved
6:0	**VSYNC_START_F2[6:0]:** The half line number of field 2 to start vertical sync. Register is programmed with one less than the actual number of half lines.

4.3.6 TV_V_CTL_3—V Control 3

Address Offset: 68044–68047h
Default Value: 00000000h

Bit	Description
31	**EQUAL_ENA:** Enable equalization generation. 0 = equalization disabled (default) 1 = equalization enabled
30:23	Reserved
22:16	**VEQ_LEN[6:0]:** The duration of the equalization interval in half lines
15	Reserved
14:8	**VEQ_START_F1[6:0]:** The half line number of field 1 to start the equalization interval. Register is programmed with one less than the actual number of half lines.
7	Reserved
6:0	**VEQ_START_F2[6:0]:** The half line number of field 2 to start the equalization interval. Register is programmed with one less than the actual number of half lines.

4.3.7 TV_V_CTL_4—V Control 4

Address Offset:	68048–6804Bh
Default Value:	00000000h

Bit	Description
31:22	Reserved
21:16	**VBURST_START_F1[5:0]:** The line number of field 1 to start inserting color burst. Measured from the start of the Vertical interval. Register is programmed with one less than the actual number of lines.
15:9	Reserved
8:0	**VBURST_END_F1[8:0]:** The line number of field 1 to stop inserting color burst. Measured from the start of the NBR. Register is programmed with one less than the actual number of lines.

4.3.8 TV_V_CTL_5—V Control 5

Address Offset:	6804C–6804Fh
Default Value:	00000000h

Bit	Description
31:22	Reserved
21:16	**VBURST_START_F2[5:0]:** The line number of field 2 to start inserting color burst. Measured from the start of the Vertical interval. Register is programmed with one less than the actual number of lines.
15:9	Reserved
8:0	**VBURST_END_F2[8:0]:** The line number of field 2 to stop inserting color burst. Measured from the start of the NBR. Register is programmed with one less than the actual number of lines.

4.3.9 TV_V_CTL_6—V Control 6

Address Offset: 68050–68053h
Default Value: 00000000h

Bit	Description
31:22	Reserved
21:16	**VBURST_START_F3[5:0]:** The line number of field 3 to start inserting color burst. Measured from the start of the Vertical interval. Register is programmed with one less than the actual number of lines. (should be programmed to the same value as VBURST_START_F1 for NTSC mode)
15:9	Reserved
8:0	**VBURST_END_F3[8:0]:** The line number of field 3 to stop inserting color burst. Measured from the start of the NBR. Register is programmed with one less than the actual number of lines. (should be programmed to the same value as VBURST_END_F1 for NTSC mode)

4.3.10 TV_V_CTL_7—V Control 7

Address Offset: 68054–68057h
Default Value: 00000000h

Bit	Description
31:22	Reserved
21:16	**VBURST_START_F4[5:0]:** The line number of field 4 to start inserting color burst. Measured from the start of the Vertical interval. Register is programmed with one less than the actual number of lines. (should be programmed to the same value as VBURST_START_F2 for NTSC mode)
15:9	Reserved
8:0	**VBURST_END_F4[8:0]:** The line number of field 4 to stop inserting color burst. Measured from the start of the NBR. Register is programmed with one less than the actual number of lines. (should be programmed to the same value as VBURST_END_F2 for NTSC mode)

4.4 Subcarrier Control Registers [DevCL]

Subcarrier generation should be disabled if encoder is set to component mode. The subcarrier registers should only be modified while the encoder is disabled.

4.4.1 TV_SC_CTL_1—SC Control 1

Address Offset: 68060–68063h
Default Value: 00000000h

Bit	Description
31	**SC_DDA1_EN:** Enables the primary subcarrier phase generation DDA
30	**SC_DDA2_EN:** Enables the secondary subcarrier phase generation DDA
29	**SC_DDA3_EN:** Enables the third subcarrier phase generation DDA This bit must be set for proper 625 line PAL operation
28:26	Reserved
25:24	**SC_RESET_MODE:** Determines the reset frequency of the subcarrier DDAs 00 = reset every other field 01 = reset every fourth field (NTSC M & J) 10 = reset every 8 fields (all PAL) 11 = never reset (NTSC 4.43)
23:16	**BURST_LEVEL[7:0]:** Value indicates one half of the peak amplitude of the colorburst at the DAC.
15:12	Reserved
11:0	**SCDDA1_INC[11:0]:** The increment value of the primary subcarrier phase generation DDA

4.4.2 TV_SC_CTL_2—SC Control 2

Address Offset: 68064–68067h
Default Value: 00000000h

Bit	Description
31	Reserved
30:16	**SCDDA2_SIZE[14:0]:** The rollover point of the of the secondary subcarrier phase generation DDA
15	Reserved
14:0	**SCDDA2_INC[14:0]:** The increment value of the secondary subcarrier phase generation DDA (this cannot be larger than SCDDA2_SIZE)

4.4.3 TV_SC_CTL_3—SC Control 3

Address Offset: 68068–6806Bh
Default Value: 00000000h

Bit	Description
31:27	Reserved
26:16	**SCDDA3_SIZE[10:0]:** The rollover point of the of the third subcarrier phase generation DDA
15:10	Reserved
9:0	**SCDDA3_INC[9:0]:** The increment value of the third subcarrier phase generation DDA

4.5 Window Control Registers [DevCL]

These registers are double buffered and are updated on every vertical sync following a write to the window size register. Coordinates are determined with a value of (0,0) being the upper left corner.

4.5.1 TV_WIN_POS—Window Position Register

Address Offset: 68070–68073h
Default Value: 00000000h

Bit	Description
31:29	Reserved
28:16	**XPOS[12:0]:** The x coordinate (in pixels) of the upper left most pixel of the display window. Measured from the end of horizontal blank.
15:12	Reserved
11:0	**YPOS[11:0]:** The y coordinate (in lines) of the upper left most pixel of the display window. Measured from the end of the start of the Non-Blanked Region (or end of the vertical interval, whatever).

4.5.2 TV_WIN_SIZE—Window Size Register

Address Offset: 68074–68077h
Default Value: 00000000h

Bit	Description
31:29	Reserved
28:16	**XSIZE[12:0]:** The horizontal size in pixels of the desired video window
15:12	Reserved
11:0	**YSIZE[11:0]:** The vertical size in pixels of the desired video window. lsb must be zero for interlaced modes.

4.6 Filter Control Registers [DevCL]

These registers are double buffered and are updated on every vertical sync following a write to the window size register.

4.6.1 TV_FILTER_CTL_1—Filter Control Register 1

Address Offset: 68080–68083h
Default Value: 00000000h

Bit	Description
31	**AUTO_SCALE_MODE:** 0 = the scaler will use the scaling factors in rest of this register 1 = the scaler will calculate the scale factors automatically, selected fractions can be read back in the registers.
30	**AUTO_SCALE_CALC: [TESTMODE]** Read-Only This read-only bit will be set while the auto scale function is in progress. It indicates that the values read back from the rest of the filter control registers should be ignored.
29	**V_FILTER_BYPASS**: Bypass the Vertical Filter Allows sources greater than 1024 pixels wide 0 = Vertical Filter Enabled (default) 1 = Vertical Filter Bypassed
28	**VADAPT : Adaptive Vertical Filter:** Puts the vertical filter into adaptive mode 0 = Adaptive filtering disabled (default) 1 = Adaptive filtering enabled

Bit	Description
27:26	**VADAPT_MODE : Adaptive Vertical Filter Mode:** Selects adaptive mode operation 00 = Least Adaptive (Recommended) 01 = Moderately Adaptive 10 = Reserved 11 = Most Adaptive
25:24	Reserved
23	**CHR_PREF: Chroma Prefilter enable** 0 = Prefilter disabled 1 = Prefilter enabled
22	**CHR_POSTF: Chroma Postfilter enable** (tentative)
21	**VERT3TAP: Force 3 tap vertical mode** (Testmode) 0 = Autodetection of 3 tap usage 1 = Force 3 tap vertical scaling
20:14	Reserved
13:0	**HSCALE_FRAC[13:0]:** The fractional part of the horizontal scaling factor divided by the oversampling rate. HSCALE_FRAC must be less than 1. HSCALE_FRAC = int((src width-1)/((oversample x dest width)-1)*2^14)

4.6.2 TV_FILTER_CTL_2—Filter Control Register 2

Address Offset: 68084-68087h
Default Value: 00000000h

This register can be considered a single 3.15 fixed point field or two fields as described below.

Bit	Description
31:18	Reserved
17:15	**VSCALE_INT[2:0]:** The integer part of the vertical scale factor. VSCALE_INT = int((src height-1)/((interlace x dest height)-1))) where interlace = 1/2 in interlace modes, 1 in progressive modes
14:0	**VSCALE_FRAC[14:0]:** The fractional part of the vertical scale factor. VSCALE_FRAC = int((src height-1)/((interlace x dest height)-1)-VSCALE_INT)*2^^15))) where interlace = 1/2 in interlace modes, 1 in progressive modes

4.6.3 TV_FILTER_CTL_3—Filter Control Register 3

Address Offset:	68088–6808Bh
Default Value:	00000000h

This register can be considered a single 3.15 fixed point field or two fields as described below. For progressive scan modes this register should be programmed to all zeroes.

Bit	Description
31:18	Reserved
17:15	**VSCALE_INT_IP[2:0]:** The integer portion of the initial phase of the vertical scaler for the bottom field VSCALE_INT_IP = int((source height-1)/(1/4 x (destination height-1)))
14:0	**VSCALE_FRAC_IP[14:0]:** The fractional portion of the initial phase of the vertical scaler for the bottom field VSCALE_FRAC_IP = int((source height-1)/(1/4 x (destination height-1)–VSCALE_INT_IP)*2^^15)))

4.6.4 SIN_ROM—Sin ROM

Address Offset:	6808C–6808Fh
Default Value:	00000000h

This register can be used to calculate a sine (or cosine), one quarterwave is stored here. The intent is to enable calculation of the filter coefficients when the FPU is not available (VBIOS). Cosine angle is written as 18 bits but cosine read on [16:6] as a 11-bit fixed point 1.10 value.

Bit	Description
31:18	Reserved
17:0	**Cosine[17:0]:** Write the angle, read the cosine 00000 = 0 degrees 10000 = 90 degrees 20000 = 180 degrees 30000 = 270 degrees

4.7 Closed Captioning Registers [DevCL]

The Closed Captioning logic is programmed by the following two registers. The first register enables the feature and controls its position. The second register is used to transmit the data.

Note: TV Out hotplug will falsely detect a hotplug change when closed captioning data is being transmitted. Software should disable or ignore any TV Out hotplug indication when closed captioning is being transmitted.

4.7.1 TV_CC_CTL—CC Control Register [DevCL]

Address Offset: 68090–68093h
Default Value: 00000000h

Bit	Description
31	**CC_ENA:** Closed Captioning Enable 0 = disabled 1 = enabled
30:28	Reserved
27	**CC_FID: [alviso/calistoga only]** Indicates the field ID of the field to send CC data. Usually programmed to 0.
26	Reserved
25:16	**CC_HOFF[9:0]:** The horizontal offset of the CC run in. Usually programmed to 135.
15:6	Reserved
5:0	**CC_LINE[5:0]:** The line number to display the CC data minus 1. Usually programmed to 21.

4.7.2 TV_CC_DATA1—CC Data Register Field 1

Address Offset: 68094–68097h
Default Value: 00000000h

Bit	Description
31	**CC_F1_RDY:** [read-only] CC engine ready 1 = data fifo full 0 = CC engine can accept data
30:15	Reserved
14:8	**CC_F1_D2[6:0]:** CC data word 2
7	Reserved
6:0	**CC_F1_D1[6:0]:** CC data word 1 (transmitted first)

4.7.3 TV_CC_DATA2—CC Data Register Field 2

Address Offset: 68098–6809Bh
Default Value: 00000000h

Bit	Description
31	**CC_F2_RDY:** [read-only] CC engine ready 1 = data fifo full 0 = CC engine can accept data
30:15	Reserved
14:8	**CC_F2_D2[6:0]:** CC data word 2
7	Reserved
6:0	**CC_F2_D1[6:0]:** CC data word 1 (transmitted first)

4.8 PAL Wide Screen Signaling (WSS) Registers [DevCL]

The WSS logic is programmed by the following two registers. The first register enables the feature and controls its position. The second register is the data to transmit. IMPORTANT: Since WSS uses line 23 to transmit data, the active region has to be downsized by two lines and the vertical blanking has to be increased by two lines.

4.8.1 TV_WSS_CTL—WSS Control Register

Address Offset: 680B0–680B3h
Default Value: 00000000h

Bit	Description
31	**WSS_ENA:** WSS Enable 0 = disabled 1 = enabled
30	**WSS_FIELD:** Sets the field in which to display WSS data. For normal PAL this should be programmed to 1 (field 1) (Assumes that standard PAL programming provided in this document is used) 0 = Field 2 1 = Field 1
29:26	Reserved
25:16	**WSS_HOFF[9:0]:** The horizontal offset of the WSS run in. Usually programmed to 148.
15	Reserved
14:8	**WSS_BIT[6:0]:** The clock divider for WSS clock. Should be 27 for 5MHz bit clock.
7:6	Reserved
5:0	**WSS_LINE[5:0]:** The line number to display the WSS data minus 1. Usually programmed to 25 to display on line 23.

4.8.2 TV_WSS_DATA—WSS Data Register Field 1

Address Offset: 680B4–680B7h
Default Value: 00000000h

Bit	Description
31:14	Reserved
13:0	**WSS_DATA [13:0]:** WSS data word

4.9 Filter Coefficient Registers [DevCL]

Coefficients for the filters are stored in sign-exponent-mantissa format similar to the overlay. The number of mantissa bit varies base on the filter. There are three exponent bits but not all values are allowed, ranges are specified per filter. Two filter coefficients are stored in each DWord, the tables below show the data packing in each of the words. Unused bits are considered reserved and should be written zero. The default value of all coefficient registers is 00000000h.

15	14:12	11:0
sign bit	exponent bits	mantissa bits (left justified)

Bit	Description
15	**Sign bit:** 0 = positive 1 = negative
14:12	**Exponent bits:** These bits are represented as 2^{1-n} available range will vary depending on the requirements of the filter. 000 = 2 or mantissa is b.bbbbbb… 001 = 1 or mantissa is 0.bbbbbbb… 010 = 0.5 or mantissa is 0.0bbbbbbb… 011 = 0.25 or mantissa is 0.00bbbbbbb… 100 = 0.125 or mantissa is 0.000bbbbbbb… etc.
11:0	**Mantissa:** Size of the mantissa varies based on the filter, but the MSB of the mantissa is always bit 11.

Coefficients are packed in the horizontal coefficient registers as follows (with the letter representing the tap and the number representing the coefficient set):

Address	bits [31:16]	bits[15:0]
68x00	B0	A0
68x04	D0	C0
68x08	F0	E0
68x0C	A1	G0
68x10	C1	B1

etc.…

4.9.1 TV_H_LUMA—H Filter Luma Coeffs

Address Offset: 68100–681EFh

17 phases of 7 taps will require 60 DWords
center coeff is 1.2.9
other coeffs are 1.2.7

4.9.2 TV_H_CHROMA—H Filter Chroma Coeffs

Address Offset: 68200–682EFh

17 phases of 7 taps will require 60 DWords
center coeff is 1.2.9
other coeffs are 1.2.7

Coefficients are packed in the vertical coefficient registers as follows (with the letter representing the tap and the number representing the coefficient set). When the vertical filter is in 3 line mode the three taps used are A, C & E, B & C must be programmed to zero in three line mode.

Address	bits [31:16]	bits[15:0]
68x00	B0	A0
68x04	D0	C0
68x08	A1	E0
68x0C	C1	B1
68x10	E1	D1

etc….

4.9.3 TV_V_LUMA—V Filter Luma Coeffs

Address Offset: 68300–683ABh

17 phases of 5 taps will require 43 DWords
center coeff is 1.2.8
other coeffs are 1.2.6

4.9.4 TV_V_CHROMA—V Filter Chroma Coeffs

Address Offset: 68400–684ABh

17 phases of 5 taps will require 43 DWords
center coeff is 1.2.8
other coeffs are 1.2.6

5 TV-Out Programming Guide [DevCL]

5.1 TV-Out Register Descriptions

The TV-Out register block is offset from the MMADR by 0x68000. The registers are divided into several sections based on function. There are many reserved testmode bits in this register block; in general software should preserve the value of those bits (unless otherwise specified) as the enabling of those functions may take place at boot time.

Address Offset	Register Name
68000	TV out Control [DevCL]
68004	TV DAC Control / Status
68008	reserved
6800C	reserved
68010	Color Space Convert Y
68014	Color Space Convert Y2
68018	Color Space Convert U
6801C	Color Space Convert U2
68020	Color Space Convert V
68024	Color Space Convert V2
68028	Color Knobs
6802C	Color Level Control
68030	H Control 1 - Hsync Htotal
68034	H Control 2 - H Burst Control
68038	H Control 3 - Blanking
6803C	V Control 1 - NBR & VI end
68040	V Control 2 - Vsync Control
68044	V Control 3 - Equalization Control
68048	V Control 4 - v burst field 1
6804C	V Control 5 - v burst field 2
68050	V Control 6 - v burst field 3
68054	V Control 7 - v burst field 4
68058	reserved
68060	SC Control 1 - enables, Burst level, SC DDA1
68064	SC Control 2 - SC DDA2
68068	SC Control 3 - SC DDA3
6806C	reserved
68070	Window Position

Address Offset	Register Name
68074	Window Size
68078-6807C	reserved
68080	Filter Control 1 - Mode and H frac
68084	Filter Control 2 - Vert scale
68088	Filter Control 3 - Vert Initial Phase
6808C	Cosine ROM
68090	CC Control
68094	CC Data Field 1
68098	CC Data Field 2
68098-680FC	reserved
680B0	WSS Control
680B4	WSS Data
68098-680FC	reserved
68100-	H Filter Y Coefficients
68200-	H Filter C Coefficients
68300-	V Filter Y Coefficients
68400-	V Filter C Coefficients

5.2 TV-Out Programming

5.2.1 Television Standards

5.2.1.1 Timing tables

NTSC-M/J/4.43 and PAL-M use the 480i timing standard; all other PAL versions use 576i timing (except PAL-N as shown below). All DTV mode timings are only available with component video and as a result do no have burst programming. Values here represent the register programming (accounting for -1s). Note that 1x mode is tentatively supported, and does not support underscan.

Timing	480i/525 (Bt470)	576i/625 (Bt470)	576i/625 (PALN)	480p	576p
Vertical Refresh	**59.94**	**50**	**50**	**59.94**	**50**
root clock	13.5M	13.5M	13.5M	27M	27M
oversample clock	108M	108M	108M	108M	108M
oversampling	8x	8x	8x	4x	4x
h sync end	64	64	64	64	64
hblank end	124	142	128	122	139
hblank start	836	844	844	842	859
htotal	857	863	863	857†	863
prog/int	i	i	i	p	p
trilevel	0	0	0	0	0
vsync start f1	6	5	6	12	10
vsync start f2	7	6	7	12	10
vsync len	6	5	6	12	10
veq en	1	1	1	0	0
veq start f1	0	0	0	X	X
veq start f2	1	1	1	X	X
veq len	18	15	18	X	X
vi end f1	20	24	24	44	48
vi end f2	21	25	25	44	48
nbr end	240	286	286	479	575
burst ena	cvbs/sv	cvbs/sv	cvbs/sv	0	0
hburst start	72	73	73	X	X
hburst len	34	32	34	X	X
vburst start f1	9	8	8	X	X

Timing	480i/525 (Bt470)	576i/625 (Bt470)	576i/625 (PALN)	480p	576p
Vertical Refresh	**59.94**	**50**	**50**	**59.94**	**50**
vburst end f1	240	285	285	X	X
vburst start f2	10	8	8	X	X
vburst end f2	240	286	286	X	X
vburst start f3	9	9	9	X	X
vburst end f3	240	286	286	X	X
vburst start f4	10	9	9	X	X
vburst end f4	240	285	285	X	X

Timing	720p	720p	1080i	1080i
Vertical Refresh	**60**	**50**	**60**	**50**
root clock	74.25M	74.25M	74.25M	74.25M
oversample clock	148.5M	148.5M	148.5M	148.5M
oversampling	2x	2x	2x	2x
h sync end	80	80	88	88
hblank end	300	300	235	235
hblank start	1580	1580	2155	2155
htotal	1649*	1979	2199*	2639
prog/int	p	p	i	i
trilevel	1	1	1	1
vsync start f1	10	10	4	4
vsync start f2	10	10	5	5
vsync len	10	10	10	10
veq en	0	0	1	1
veq start f1	X	X	4	4
veq start f2	X	X	4	4
veq len	X	X	10	10
vi end f1	29	29	21	21
vi end f2	29	29	22	22
nbr end	719	719	539	539
burst ena	0	0	0	0

5.2.1.1.1 Notes about programming POS and SIZE registers (when preserving aspect ratio is NOT important)

The default POS values are 0,0.

The maximum SIZE register programming is dependant on the size of the nonblanked region. The following formulae should be useful.

if interlaced:

YSIZE = 2*(nbr_end+1)

if progressive:

YSIZE = nbr_end+1

in either case

XSIZE = (hblank_start - hblank_end)

5.2.1.1.2 Notes about programming POS and SIZE registers (when preserving aspect ratio is important)

The default POS values are 0,0.

The maximum SIZE register programming is dependant on the size of the nonblanked region. The following formulae should be useful.

if interlaced:

YSIZE = 2*(nbr_end+1)

if progressive:

YSIZE = nbr_end+1

in either case

XSIZE = (hblank_start - hblank_end)

In order to keep preserve aspect ratio it may be necessary to reduce one of the two values.

Example1: 4:3 source on NTSC(Bt470).

from above XSIZE = 712, YSIZE = 482.

if XSIZE is 712, NTSC pixel has 8:9 aspect ratio (at 13.5MHz) so in square pixels this is 633. so for 4:3 the YSIZE should be 475.

Active area is then 712x475.

to center this YPOS = (YSIZEmax - YSIZE)/2 = 3 (rounded)

***Note that the XSIZE of 633 square pixels represents the destination surface size and should be taken into account for purpose of calculating coefficients.

Example 2: 4:3 source on PAL B (Bt470)

from above XSIZE = 702, YSIZE = 576

PAL has 10:9 pixel aspect ratio (at 13.5 MHz) so in square pixels this is 780.

for 4:3 aspect this the YSIZE should be 576. so YPOS is zero.

5.2.1.2 Subcarrier Programming

For high definition modes the subcarrier generator is disabled.

	NTSC (M/J)	**PAL(most)**	**PAL Nc**	**PAL M**	**NTSC-4.43**
Oversample	8	8	8	8	8
DDA2 size	27456	27648	27648	27456	27456
DDA3 size	off	625	625	off	525
DDA1 inc	135	168	135	135	168
DDA2 inc	20800	4122	23578	16704	4093
DDA3 inc	off	67	134	off	310
SCreset	4	8	8	8	never
PAL-Burst	0	1	1	1	0

5.2.1.3 Color Space Conversion/Blank and Black Level Programming

The table below gives the register programming for the color space converter and the blank, blank and burst level registers based on output format and standard. PAL-M uses the NTSC-M programming, PAL-N is the same only with a larger burst level.

Topic	NTSC-M	NTSC-M	NTSC-J	NTSC-J
	composite	s-video	composite	s-video
blank level	225	266	225	266
black level	267	316	225	266
burst level	113	133	113	133
RY	0.2990	0.2990	0.2990	0.2990
GY	0.5870	0.5870	0.5870	0.5870
BY	0.1140	0.1140	0.1140	0.1140
AY	0.5082	0.6006	0.5495	0.6494
RU	-0.0749	-0.0885	-0.0810	-0.0957
GU	-0.1471	-0.1738	-0.1590	-0.1879
BU	0.2220	0.2624	0.2400	0.2836
AU	1.0000	1.0000	1.0000	1.0000
RV	0.3125	0.3693	0.3378	0.3992
GV	-0.2616	-0.3092	-0.2829	-0.3343
BV	-0.0508	-0.0601	-0.0549	-0.0649
AV	1.0000	1.0000	1.0000	1.0000

Topic	PAL	PAL	PAL-M	PAL-M	PAL-N	PAL-N
	composite	s-video	composite	s-video	composite	s-video
blank level	237	280	225	266	225	266
black level	237	280	267	316	267	316
burst level	118	139	113	133	118	139
RY	0.2990	0.2990	0.2990	0.2990	0.2990	0.2990
GY	0.5870	0.5870	0.5870	0.5870	0.5870	0.5870
BY	0.1140	0.1140	0.1140	0.1140	0.1140	0.1140
AY	0.5379	0.6357	0.5082	0.6006	0.5082	0.6006
RU	-0.0793	-0.0937	-0.0749	-0.0885	-0.0749	-0.0885
GU	-0.1557	-0.1840	-0.1471	-0.1738	-0.1471	-0.1738
BU	0.2350	0.2777	0.2220	0.2624	0.2220	0.2624
AU	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
RV	0.3307	0.3908	0.3125	0.3693	0.3125	0.3693
GV	-0.2769	-0.3273	-0.2616	-0.3092	-0.2616	-0.3092
BV	-0.0538	-0.0636	-0.0508	-0.0601	-0.0508	-0.0601
AV	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Topic	480(576) i/p	480(576) i/p	HDTV	HDTV
	YPrPb	RGB	YPrPb	RGB
blank level	279	279	279	279
black level	279	279	279	279
burst level	0	0	0	0
RY	0.2990	0	0.2126	0
GY	0.5870	1.000	0.7152	1.000
BY	0.1140	0	0.0722	0
AY	0.6364	0.7000	0.6364	0.7000
RU	-0.1687	0	-0.1146	0
GU	-0.3313	0	-0.3854	0
BU	0.5000	1.000	0.5000	1.000
AU	1.0000	0.7000	1.0000	0.7000
RV	0.5000	1.000	0.5000	1.000
GV	-0.4187	0	-0.4542	0
BV	-0.0813	0	-0.0458	0
AV	1.0000	0.7000	1.0000	0.7000

RGB modes are used for SMPTE 253 applications.

5.2.2 Underscanning and Pixel Aspect Ratios

Underscanning is used to fit the computer display inside the television's bezel. All televisions use an overscanned image to get picture to appear centered. Overscanning hides the corners of the transmitted image behind the bezel. The maximum picture area for NTSC is 720x480 pixels, the actual number of visible pixels depends on the size of the bezel. To account for this the TV-Out design in equipped with a variable scaler and underscan position control. Underscanning affects filter selection, and good quality underscanning is achievable up to 10% in all supported modes. Some modes can support further undescan if required.

Another issue in this area is the fact that the NTSC pixels aren't square, they have an aspect ratio of 9:8 (w:h) which means they are a little narrower than they are tall. This must be accounted for when calculating scale factors or when using the autoscaler. For example if a 640 pixel line is displayed on the TV the destination line size should be 640x9/8 = 720 pixels. If scaling is already being performed (from 10x7 or such) the 9/8 factor must be included or the image will look squished.

5.2.3 Programming Filter Coefficients

Programming the filter coefficients is very similar to the programming of the overlay coefficients. It uses a very similar data format, and coeffs are stored in the same ordering. However the filters are much sharper for the TV horizontal filter due to signal bandwidth limits. The intent of the vertical filter is to reduce the flicker caused by interlacing the computer output.

There are five filters implemented in the design requiring four unique filter coefficient sets, 2 sets for vertical filtering and 2 sets for horizontal filtering.

Different sets of coefficients are needed for each standard and output mode.

The following sections provide a formula (in six variables) that will generate the coefficient set for each and a table of various modes and what values to use for the sixvariables. The 6 variables will be selected from a table based on the standard, output type, scale factor

5.2.4 Setting the Mode

In general, the TV-Out logic should be programmed before it is enabled. The following steps should be followed.

1. disable the pipe if it is running.
2. set display PLL to the required clock frequency (from the table)
3. make sure that DPO programming is acceptable according to the rules outlined below, change them (and the planes) if needed
4. enable the pipe
5. program all TV-Out registers, then enable TV-Out

Exiting the mode is the reverse of these steps.

There are no special restrictions on planes used with TV-Out. It is okay to use double wide pixel mode if needed due to low core frequency.

5.2.4.1 Hi-Res Mode Table

TV-Out Mode Support

Source Aspect	Source Format	Destination Format						
		4:3			16:9			
		480i	576i	480p	480p	720p	1080i	1080p
4:3	640x480	Y	Y	Y	P	P	P	P
	800x600	Y	Y	Y	P	P	P	P
	1024x768	Y	Y	Y	P	P	P	P
	1280x1024	Y	Y	Y	P	P	P	P
	1600x1200	N	N	Y	P	P	P	P
16:9	848x480	L	L	L	Y	Y	Y	Y
	1280x720	L	L	L	Y	Y	Y	Y
	1600x900	N	N	L	Y	Y	Y	Y
	1920x1080	N*	N*	L	Y	Y	Y	Y
14:10	1400x1050	L	L	L	P	P	P	P

Y = yes the mode is supported
N = not supported.
P = the mode is supported in pillarbox
L = the mode is supported in letterbox
576p support is the same as 480p.
* Can be supported in letter box if source is interlaced. This results in a quality drop.

5.2.4.2 Hi-Res Modes

If TV-Out is to be used in a non-mirroring scenario Intel suggests using the native screen size for the display mode as this will present the best possible quality image. For SDTV & 480p(4:3) this is 640x480, 848x480 for 480p(16:9), 1280x720 for 720p and 1920x1080 for 1080i or 1080p. Bear in mind pixel aspect ratios when programming the horizontal scaler and window size.

DPO timing programming remains the same for all source modes, only the PLL clock frequency is changed. PLL frequency selection is described the Timing Standards section earlier in this document.

5.2.4.3 3 Tap Mode

3 tap mode enables when:

- Vertical filter bypass is disabled (not bypass)

  AND

- TV in 480p and above HD modes

  AND

- 3 tap mode has been set by ONE of the following scenarios:
  - HW sets 3 tap enable automatically when horizontal source size > 1024.
  - SW sets Filter Control Register 1 bit 21 (VERT3TAP) to 1 (in test mode only).

When 3 tap is enabled, 3 tap coefficients need to be loaded.

Interlaced mode settings:

Field Swap (FIELD_SWAP)	Pipe interlaced	TV interlaced (PROG_INT)	Vertical Filter Bypass (V_FILTER_BYPASS)	3 tap mode
Normal	off	on (set to 0)	Not bypass	On
Inverted	on	off (set to 1)	Bypass	Off

5.2.4.4 VGA Modes

If TV-Out is to be used with VGA, the VGA Mode Disable bit in the PLLA (or PLLB depending on which pipe VGA is assigned) control register must be set to bypass the VGA clocks. Centering is not required for these modes, the TV filter will properly scale the VGA image to the size indicated in the TV Window registers if auto scale is enabled.

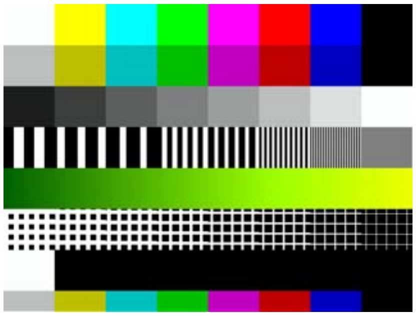
The VGA engine has tight restrictions on clocking in that the pixel clock is not allowed to exceed 45% of the core clock. The VGA throttling bits are used to ensure that the VGA does not exceed this limit.

5.2.4.5 Test Modes

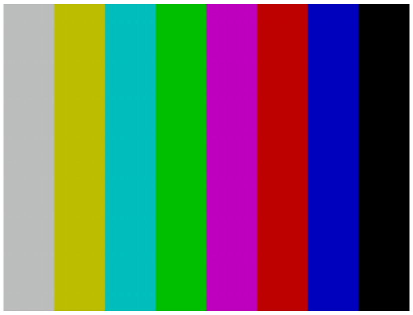
The TV-Out logic is equipped with several built in testing modes. They fall in to two categories: Video Quality Tests and DAC Quality Tests.

5.2.4.5.1 Video Quality Tests

The video quality tests provide several reference images intended for use in testing the quality of the video encoder itself. They do not require the pipe or plane logic to be enabled to function. The encoder is programmed normally for the television standard, including PLL frequency. Then the encoder testmode bits are set to one of the four patterns. The first 3 images are shown below, pattern 4 is random noise which varies from frame to frame. Pattern 5 is intended for use is testing linearity.

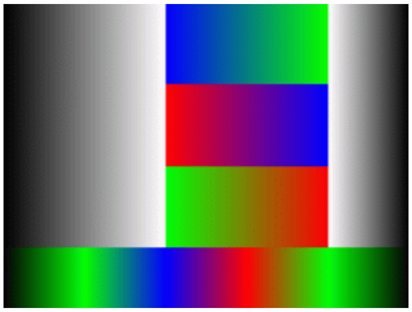


Test Image 1: Combo Pattern



Test Image 2: 75% color bars

Test Image 3: 75% horizontal bars



Test Image 5: Color Ramps

5.2.4.5.2 DAC Quality Tests

The DAC quality tests provided several reference patterns for measuring various quality standards of DAC performance. For these tests the subcarrier DDA is used as a to generate various patterns by using the sin lookup table or other function. To set the TV-Out logic for DAC testing DPLL A must be running, frequency is user selectable and is allowed to go as high as 150MHz. The subcarrier DDA1 increment is programmed to 1, DDA2 and DDA3 are disabled. The SC reset value should be set to 3 (never reset). The test modes are pretty self explanatory, check the register description for details of each test.

5.2.5 Detection and Determination of the Load

The monitor detection logic can work in two ways. A software driven polling method or and interrupt driven method. The interrupt method is the preferred method since it uses the least amount of power. The frequency of the polling in the software method will determine the power drain in that mode. In either detection mode determination of the load type is done in the same way. Alternatively, the lowest power option is to add a TV Detect button to the displays control panel (or system tray item) to force a polling mode detection loop.

5.2.5.1 Polling Based Load Connect Detection

In this mode the DACs are powered down completely, until software wishes to check for a connection change. The frequency at which this is performed is under SW control and directly affects the power consumption. The DACs are enabled in 0.7v mode. The pipe to which the TV-Out logic is assigned must have its PLL running. The encoder enable bit needs to be set to off. TESTMODE is set to monitor detect and the sense values are set to the high state. SW will need to wait for the PLL & DACs to come on, which may take some time. At this point the sense state bits are read, and if any of the bits are zero (indicating voltage level below threshold) then a load has been detected. After detection of a load the TESTMODE should immediately be set to normal. The PLL can be shut off.

5.2.5.2 Interrupt Based Load Connect Detection (with hotplug timer) (Not on [DevCL])

This feature is not on [DevCL].

The pipe PLL is turned on as in the polling mode. The software sets TEST MODE to 0. The threshold sense values are set to the high state. Enable state change detection. Finally, the TVDAC state change interrupt enable bit is set. The PLL can now be turned off. At this point any change in the sense state bits will indicate a change in the load and will generate an interrupt.

5.2.5.3 Interrupt Based Load Connect Detection (without hotplug timer)

This is not recommended since this leaves DACs enabled and drawing power, but is supplied here in case the detection mode can be changed when the power supply is connected.

In this mode two of the TVDACs must remain enabled in order to detect the change in the load (usually only DAC A&B). Using the TVDAC override controls the DACs are enabled in 0.7v mode. The pipe PLL is turned on as in the polling mode. Software places the DACs in monitor detect mode by entering TESTMODE 7. The threshold sense values are set to the high state. Enable state change detection. Finally the TVDAC state change interrupt enable bit is set. The PLL can now be turned off. At this point any change in the sense state bits will indicate a change in the load and will generate an interrupt.

5.2.5.4 Load Type Determination

The type of load can be determined after the Load Detection phase by the position of zeroes in the sense state bits. Combo mode is not detectable and also not recommended. This assumes the use of a standard pinout to the connector as recommended below.

A	B	C	Mode Detected
0	1	1	Composite
1	0	X	Svideo
0	0	0	Component

5.2.5.5 Disconnect Detection

This can be performed with the interrupt mode or the polling mode. The TV-Out logic is programmed to its normal mode. In either polling or interrupt case the channel(s) which contains either CVBS or Y data have their threshold sense levels set to low. Then the TVDAC state change enable bit can be set, and software can wait for an interrupt, or the software can periodically poll the sense state bits watching for a change. The bits are updated during vertical blank.

5.2.5.6 Alternate methods

It is also possible to remap the DAC outputs to different connector pins. In this way it would be possible to leave just one DAC on for interrupt mode detection. This is still a power drain, and would also lose connectivity with the dongle recommendation in the next section.

5.3 PCB Connector and Dongle Design

This dongle design is intended to give the best support for most of the users and conforms to the default configurations outlined in this specification. Two connector types are used, a 7 pin miniDIN connector and the RCA Phono jack. The miniDIN is expected to be used on the motherboard and is the source of all signals.

This connector is pin compatible with the standard Svideo connector (though 3 pins are not used) and so using svideo cables require no dongle at all. The two possible dongles are used to change the connector type to composite (1 connector) and component (3 connector), both of these connections rely on RCA type jacks for their connections.

Below is a diagram of the 7 pin miniDIN connector pinout, looking into the female connector which is a right angle connection mounted on the PCB. The shield is usually connected to chassis ground, (which is usually connected to PCB ground though a ferrite bead), but in either dongle the chassis ground connection is not used.

```
  _______
 / 4 7 3 \
| 2 6 5 1 |
 \  key  /
```

The dongles described below assume the following connections between the encoder and the connector. Optionally, regular GND can be used for the DAC RTN signals, if that level of signal degradation is acceptable.

miniDIN	PCB
1	DAC B RTN
2	DAC C RTN
3	DAC B OUT
4	DAC C OUT
5	DAC A OUT
6	DAC A RTN
7	nc
shield	CHAS

The RCA connection is a simple 1 wire shielded connection with the shield connected to PCB ground (not chassis GND) and the signal carried on the center connection. For our purposes, the center connection will be pin 1 and the shield will be pin 2.

The composite dongle is simple, it has one male 7 pin miniDIN and one male RCA (usually yellow). The wire used should be 75ohm coaxial cable, the thin kind is okay. The cable can also be built with a female RCA so that users can extend as needed, this way the dongle can be very short, as small as 3-6".

miniDIN	RCA
1	nc
2	nc
3	nc
4	nc

5	1
6	2
7	nc
shield	nc

The component dongle is a little more complicated. It has one male 7 pin miniDIN and three male RCAs (usually red, green and blue). Again the wire used should be 75ohm coaxial cable. The cable can also be built with female RCAs so that users can extend as needed, this way the dongle can be very short, as small as 3-6".

miniDIN	BLUE RCA	GREEN RCA	RED RCA
1	nc	2	nc
2	nc	nc	2
3	nc	1	nc
4	nc	nc	1
5	1	nc	nc
6	2	nc	nc
7	nc	nc	nc
shield	nc	nc	nc

It should be noted that the component video dongle can be used for composite connection by using the blue RCA only and letting the green and red dongles dangle.

Cables that were distributed with the Alviso A-0 Silicon had the Red and Blue RCA's swapped. If you are using one of these cables the Composite signal will be on the red RCA. For component the red RCA has Pb and the blue RCA has Pr.

6 VGA and Extended VGA Registers (00000h–00FFFh)

This section describes the registers and the functional operation notations for the observable registers in the VGA section. This functionality is provided as a means for support of legacy applications and operating systems. It is important to note that these registers will in general have the desired effects only when running VGA display modes.

The main exceptions to this are the palette interface which will allow real mode DOS applications and full screen VGA applications under an OS control running in high resolution (non-VGA) modes to access the palette through the VGA register mechanisms and the use of the ST01 status bits that determine when the VGA enters display enable and sync periods. Other exceptions include the register bits that control the memory accesses through the A000:0000 and B000:0000 memory segments which would get used during operating system emulation of VGA for "DOS box" applications. Some of the functions of the VGA are enabled or defeated through the programming of the VGA control register bits that which are located in the MMIO register space.

Given the legacy nature of this function, it has been adapted to the changing environment that it must operate within. The three most notable changes were the addition of high resolution display mode support, new operating system support, and the use of fixed resolution display devices (such as LCD panels). Additional control bits in the PCI Config space will affect the ability to access the registers and memory aperture associated with VGA.

Mode of Operation	VGA Disable	VGA Display	VGA Registers	Palette (VGA)	VGA Memory	VGA Banking
VGA DOS	No	Yes	Yes	Yes	Yes	No
HiRes DOS	Yes	No	Yes	Yes	No	Yes
Win9x FS DOS	Yes/No	No/Yes	Yes	Yes	Yes	Yes
Windows 9x DOS Emulation	Yes	No	Yes	Yes	Yes	Yes

VGA Display Mode	Dot Clock Select	Dot Clock Range	Can use DVO Stall	132 Column Text Support	9-Dot Disable Support	Main Use
Native	VGA Clock Select	25/28 MHz	No	No	No	Analog CRT (VGA connector)
Centered	Fixed at display Requirements	Product Specific	No	No	Yes	DVI connector
Upper left corner	Fixed at display Requirements	Product Specific	Yes	No	Yes	Internal Panel

Even in the native VGA display operational modes, not all combinations of bit settings result in functional operating modes. VGA display modes have the restriction that they can be used only when all other display planes are disabled except for the Cursor A when used as a popup over VGA (in devices that support that function).

In most cases, these registers can be accessed via either I/O space or memory space. The I/O space resides in the PCI compatibility hole and uses only the addresses that were part of the original VGA I/O space (which includes EGA and MDA emulation). Accesses to the VGA I/O addresses are steered to the proper bus and rely on proper setup of bridge registers. Extended VGA registers such as GR10 and GR11 use additional indexes for the already defined I/O addresses. VGA register accesses are allowed as 8 or 16-bit naturally aligned transactions only. Word transactions must have the lsb of the address set to zero. DWORD I/O operations should not be performed on these registers, **DWORD Writes are IGNORED. Reads always returns xxFFFFFFh.**

The memory space addresses listed are offsets from the base memory address programmed into the MMAPA register (PCI configuration offset 14h). For each register, the memory mapped address offset is the same address value as the I/O address. Several registers (GR10 and GR11) should never be accessed through the memory mapped aperture.

6.1 General Control and Status Registers

The setup, enable, and general registers are all directly accessible by the CPU. A sub indexing scheme is not used to read from and write to these registers.

Name	Function	Read		Write	
		I/O	Memory Offset	I/O	Memory Offset
ST00	VGA Input Status Register 0	3C2h	3C2h	—	—
ST01	VGA Input Status Register 1	3BAh/3DAh[1]	3BAh/3DAh[1]	—	—
FCR	VGA Feature Control Register	3CAh	3CAh	3BAh/3DAh[1]	3BAh/3DAh[1]
MSR	VGA Miscellaneous Output Register	3CCh	3CCh	3C2h	3C2h

Note:

1. The address selection for ST01 reads and FCR writes is dependent on CGA or MDA emulation mode as selected via the MSR register.

Various bits in these registers provide control over and the real-time status of the horizontal sync signal, the horizontal retrace interval, the vertical sync signal, and the vertical retrace interval. The horizontal retrace interval is the period during the drawing of each scan line containing active video data, when the active video data is not being displayed. This period includes the horizontal front and back porches, and the horizontal sync pulse. The horizontal retrace interval is always longer than the horizontal sync pulse. The vertical retrace interval is the period during which the scan lines not containing active video data are drawn. This includes the vertical front porch, back porch, and the vertical sync pulse. The vertical retrace interval is normally longer than the vertical sync pulse.

Display Enable is a status bit (bit 0) in VGA Input Status Register 1 that indicates when either a horizontal retrace interval or a vertical retrace interval is taking place. In the IBM* EGA graphics system (and the ones that preceded it, including MDA and CGA), it was important to check the status of this bit to ensure that one or the other retrace intervals was taking place before reading from or writing to the frame buffer. In these earlier systems, reading from or writing to frame buffer at times outside the retrace intervals meant that the CRT controller would be denied access to the frame buffer in while accessing pixel data needed to draw pixels on the display. This resulted in either "snow" or a flickering display.

6.1.1 ST00—Input Status 0

I/O (and Memory Offset) Address: 3C2h
Default: 00h
Attributes: Read-Only

7	6 5	4	3 0
CRT Interrupt	Reserved (00)	RGB Cmp / Sen	Reserved (0000)

Bit	Descriptions
7	**CRT Interrupt Pending.** This bit is here for EGA compatibility and **will always return zero**. Note that the generation of interrupts was originally enabled, through bits [4,5] of the Vertical Retrace End Register (CR11). This ability to generate interrupts at the start of the vertical retrace interval is a feature that is typically unused by DOS software and therefore is only supported through other means for use under a operating system support. 0 = CRT (vertical retrace interval) interrupt is not pending. 1 = CRT (vertical retrace interval) interrupt is pending
6:5	Reserved. Read as 0s.
4	**RGB Comparator / Sense.** This bit returns the state of the output of the RGB output comparator(s). Video BIOS uses this bit during POST to determine whether the display is connected and if it is a color or monochrome CRT. BIOS blanks the screen or clears the frame buffer to display only black. Next, BIOS outputs a ramp to the D-to-A converters to test for the presence of a color display by determining which code cause the comparator to switch. Finally, if the BIOS does not detect any termination resistors on Red or Blue, it tests for the presence of a display using the Green signal. The result of each such test is read via this bit. 0 = Below threshold 1 = Above threshold
3:0	Reserved. Read as 0s.

6.1.2 ST01—Input Status 1

I/O (and Memory Offset) Address: 3BAh/3DAh
Default: 00h
Attributes: Read-Only

The address selection is dependent on CGA or MDA emulation mode as selected via the MSR register.

7	6	5	4	3	2	1	0
Reserved (0)	Reserved (0)	Video Feedback		Vertical Retrace	Reserved (00)		Display Enable

Bit	Descriptions
7	Reserved (as per VGA specification). Read as 0s.
6	Reserved. Read as 0.
5:4	**Video Feedback 1, 0.** These are diagnostic video bits that are selected by the Color Plane Enable Register. These bits that are programmably connected to 2 of the 8 color bits sent to the palette. Bits 4 and 5 of the Color Plane Enable Register (AR12) selects which two of the 8 possible color bits become connected to these 2 bits of this register. The current software normally does not use these 2 bits. They exist for EGA compatibility.
3	**Vertical Retrace / Video.** 0 = VSYNC inactive (Indicates that a vertical retrace interval is not taking place). 1 = VSYNC active (Indicates that a vertical retrace interval is taking place). **Note:** VGA pixel generation is not locked to the display output but is loosely coupled. A VSYNC indication may not occur during the actual VSYNC going to the display but during the VSYNC that is generated as part of the VGA pixel generation. The exact relationship will vary with the VGA display operational mode. This status bit will remain active when the VGA is disabled and the device is running in high resolution modes (non-VGA) to allow for applications that (now it is incorrect) use these status registers bits. In this case, the status will come from the pipe that the VGA is assigned to. Bits 4 and 5 of the Vertical Retrace End Register (CR11) previously could program this bit to generate an interrupt at the start of the vertical retrace interval. This ability to generate interrupts at the start of the vertical retrace interval is a feature that is largely unused by legacy software. Interrupts are not supported through the VGA register bits.
2:1	Reserved. Read as 0s.

Bit	Descriptions
0	**Display Enable Output.** Display Enable is a status bit (bit 0) in VGA Input Status Register 1 that indicates when either a horizontal retrace interval or a vertical retrace interval is taking place. This was used with the IBM* EGA graphics system (and the ones that preceded it, including MDA and CGA). In those cases, it was important to check the status of this bit to ensure that one or the other retrace intervals was taking place before reading from or writing to the frame buffer. In these earlier systems, reading from or writing to frame buffer at times outside the retrace intervals meant that the CRT controller would be denied access to the frame buffer. This resulted in either "snow" or a flickering display. This bit provides compatibility with software designed for those early graphics controllers. This bit is currently used in DOS applications that access the palette to prevent the sparkle associated with read and write accesses to the palette ram with the same address on the same clock cycle. **Note:** This status bit will remain active when the VGA display is disabled and the device is running in high resolution modes (non-VGA) to allow for applications that (now considered incorrect) use these status registers bits. In this case, the status will come from the pipe that the VGA is assigned to. When in panel fitting VGA or centered VGA operation, the meaning of these bits will not be consistent with native VGA timings. 0 = Active display data is being sent to the display. Neither a horizontal retrace interval nor a vertical retrace interval is currently taking place. 1 = Either a horizontal retrace interval (horizontal blanking) or a vertical retrace interval (vertical blanking) is currently taking place.

6.1.3 FCR—Feature Control

I/O (and Memory Offset) Address:	3BAh/3DAh— Write;	3CAh— Read
Default:	00h	
Attributes:	See Address above	

The I/O address used for writes is dependent on CGA or MDA emulation mode as selected via the MSR register. In the original EGA, bits 0 and 1 were used as part of the feature connector interface. Feature connector is not supported in these devices and those bits will always read as zero.

7 4	3	2 0
Reserved (0000)	VSYNC Control	Reserved (000)

Bit	Descriptions
7:4	Reserved. Read as 0.
3	**VSYNC Control.** This bit is provided for compatibility only and has no other function. Reads and writes to this bit have no effect other than to change the value of this bit. The previous definition of this bit selected the output on the VSYNC pin. 0 = Was used to set VSYNC output on the VSYNC pin (default). 1 = Was used to set the logical 'OR' of VSYNC and Display Enable output on the VSYNC pin. This capability was not typically very useful.
2:0	Reserved. Read as 0.

6.1.4 MSR—Miscellaneous Output

I/O (and Memory Offset) Address: 3C2h — Write; 3CCh— Read
Default: 00h
Attributes: See Address above

7	6	5	4	3	2	1	0
VSYNC Polarity	HSYNC Polarity	Page Select	Reserved (0)	Clock Select		Memory Enable	I/O Address

Bit	Descriptions
7	**CRT VSYNC Polarity.** This is a legacy function that is used in native VGA modes. For most cases, sync polarity will be controlled by the port control bits. The VGA settings can be optionally selected for compatibility with the original VGA when used in the VGA native mode. Sync polarity was used in VGA to signal the monitor how many lines of active display are being generated. 0 = Positive Polarity (default). 1 = Negative Polarity.
6	**CRT HSYNC Polarity.** This is a legacy function that is used in native VGA modes. For most cases, sync polarity will be controlled by the port control bits. The VGA settings can be optionally selected for compatibility with the original VGA when used in the VGA native mode. 0 = Positive Polarity (default). 1 = Negative Polarity
5	**Page Select.** In Odd/Even Memory Map Mode 1 (GR6), this bit selects the upper or lower 64 KB page in display memory for CPU access: 0 = Upper page (default) 1 = Lower page. Selects between two 64KB pages of frame buffer memory during standard VGA odd/even modes (modes 0h through 5h). Bit 1 of register GR06 can also program this bit in other modes. Note that this bit is would normally set to 1 by the software.
4	Reserved. Read as 0.
3:2	**Clock Select.** These bits can select the dot clock source for the CRT interface. The bits should be used to select the dot clock in standard native VGA modes only. When in the centering or upper left corner modes, these bits should be set to have no effect on the clock rate. The actual frequencies that these bits select, if they have any affect at all, is programmable through the DPLL registers that default to the standard values used for VGA. 00 = CLK0, 25.175 MHz (for standard VGA modes with 640 pixel (8-dot) horizontal resolution) (default) 01 = CLK1, 28.322 MHz. (for standard VGA modes with 720 pixel (9-dot) horizontal resolution) 10 = Was used to select an external clock (now unused) 11 = Reserved

Bit	Descriptions
1	**A0000–BFFFFh Memory Access Enable.** VGA Compatibility bit enables access to local video memory (frame buffer) at A0000–BFFFFh. When disabled, accesses to VGA memory are blocked in this region. This bit is independent of and does not block CPU access to the video linear frame buffer at other addresses. Note that it is typical for AGP chipsets to shadow this register to allow proper steering of memory accesses to the proper bus. 0 = Prevent CPU access to memory/registers/ROM through the A0000-BFFFF VGA memory aperture (default). 1 = Allow CPU access to memory/registers/ROM through the A0000-BFFFF VGA memory aperture. This memory must be mapped as UC by the CPU; see **VGA Host Access Memory Munging** in *Display and Overlay Functions*.
0	**I/O Address Select.** This bit selects 3Bxh or 3Dxh as the I/O address for the CRT Controller registers, the Feature Control Register (FCR), and Input Status Register 1 (ST01). Presently ignored (whole range is claimed), but will "ignore" 3Bx for color configuration or 3Dx for monochrome. Note that it is typical in AGP chipsets to shadow this bit and properly steer I/O cycles to the proper bus for operation where a MDA exists on another bus such as ISA. 0 = Select 3Bxh I/O address (MDA emulation) (default). 1 = Select 3Dxh I/O address (CGA emulation).

Note:

1. In standard VGA modes using the analog VGA connector, bits 7 and 6 indicate which of the three standard VGA vertical resolutions the standard VGA display should use. Extended modes, including those with a vertical resolution of 480 scan lines, may use a setting of 0 for both of these bits. Different connector standards and timing standards specify the proper use of sync polarity. This setting was "reserved" in the VGA standard.

Table 6-1. Analog CRT Display Sync Polarities

V	H	Display	Horizontal Frequency	Vertical Frequency
P	P	200 Line	15.7 KHz	60 Hz
N	P	350 Line	21.8 KHz	60 Hz
P	N	400 Line	31.5 KHz	70 Hz
N	N	480 Line	31.5 KHz	60 Hz

6.2 Sequencer Registers

The sequencer registers are accessed via either I/O space or Memory space. To access registers the VGA Sequencer Index register (SRX) at I/O address 3C4h (or memory address 3C4h) is written with the index of the desired register. Then the desired register is accessed through the data port for the sequencer registers at I/O address 3C5 (or memory address 3C5).

6.2.1 SRX—Sequencer Index

I/O (and Memory Offset) Address: 3C4h
Default: 00h
Attributes: Read/Write

7 3	2 0
Reserved (00000)	Sequencer Index

Bit	Descriptions
7:3	Reserved. Read as 0s.
2:0	**Sequencer Index.** This field contains a 3-bit Sequencer Index value used to access sequencer data registers at indices 0 through 7. **Notes:** SR02 is referred to in the VGA standard as the Map Mask Register. However, the word “map” is used with multiple meanings in the VGA standard and was, therefore, deemed too confusing; hence, the reason for calling it the Plane Mask Register. SR07 is a standard VGA register that was not documented by IBM. It is not a graphics controller extension.

6.2.2 SR00—Sequencer Reset

I/O (and Memory Offset) Address: 3C5h(Index=00h)
Default: 00h
Attributes: Read/Write

7 2	1	0
Reserved (000000)	Reserved	Reserved

Bit	Descriptions
7:2	Reserved. Read as 0.
1	Reserved. Reserved for VGA compatibility (was reset).
0	Reserved. Reserved for VGA compatibility. (was reset)

6.2.3 SR01—Clocking Mode

I/O (and Memory Offset) Address: 3C5h (Index=01h)
Default: 00h
Attributes: Read/Write

7	6	5	4	3	2	1	0
Reserved (00)		Screen Off	Shift 4	Dot Clock Divide	Shift Load	Reserved (0)	8/9 Dot Clocks

Bit	Descriptions
7:6	Reserved. Read as 0s.
5	**Screen Off.** 0 = Normal Operation (default). 1 = Disables video output (blanks the screen) and turns off display data fetches. Synchronization pulses to the display, however, are maintained. Setting this bit to 1 had been used as a way to more rapidly update and improve CPU access performance to the frame buffer during VGA modes. In non-VGA modes (VGA Disable=1), this bit has no effect. Before the VGA is disabled through the VGA control register, this bit should be set to stop the memory accesses from the display. **Programming Notes:** • In order to disable the VGA plane, SW must first write SR01, bit 5 = 1. It then must wait for 30us then disable the plan via Bit 31, Reg. 71400. Failure to do so will cause random VGA and CPU lockups.
4	**Shift 4.** 0 = Load video shift registers every 1 or 2 character clocks (depending on bit 2 of this register) (default). 1 = Load shift registers every 4th character clock.
3	**Dot Clock Divide.** Setting this bit to 1 stretches doubles all horizontal timing periods that are specified in the VGA horizontal CRTC registers. This bit is used in standard VGA 40-column text modes to stretch timings to create horizontal resolutions of either 320 or 360 pixels (as opposed to 640 or 720 pixels, normally used in standard VGA 80-column text modes). The effect of this is that there will actually be twice the number of pixels sent to the display per line. 0 = Pixel clock is left unaltered (used for 640 (720) pixel modes); (default). 1 = Pixel clock divided by 2 (used for 320 (360) pixel modes).
2	**Shift Load.** Bit 4 of this register must be 0 for this bit to be effective. 0 = Load video data shift registers every character clock (default). 1 = Load video data shift registers every other character clock.
1	Reserved. Read as 0.

Bit	Descriptions
0	**8/9 Dot Clocks.** This bit determines whether a character clock is 8 or 9 dot clocks long if clock doubling is disabled and 16 or 18 clocks if it is. This also changes the interpretation of the pixel panning values (see chart). An additional control bit determines if this bit is to be ignored and 8-dot characters are to be used always. The 9-dot disable would be used when doubling the horizontal pixels on a 1280 wide display or non-doubling on a 640 wide display. Panning however will occur according to the expected outcome. 0 = 9 dot clocks (9 horizontal pixels) per character in text modes with a horizontal resolution of 720 pixels. 1 = 8 dot clocks (8 horizontal pixels) per character in text or graphics modes with a horizontal resolution of 640 pixels.

6.2.4 SR02—Plane/Map Mask

I/O (and Memory Offset) Address: 3C5h (Index=02h)
Default: 00h
Attributes: Read/Write

7 4	3 0
Reserved	Memory Planes Processor Write Access Enable.

Bit	Descriptions
7:4	Reserved. Read as 0s.
3:0	**Memory Planes [3:0] Processor Write Access Enable.** In both the Odd/Even Mode and the Chain 4 Mode, these bits still control access to the corresponding color plane. 0 = Disable. 1 = Enable. **Note:** This register is referred to in the VGA standard as the Map Mask Register. However, the word “map” is used with multiple meanings in the VGA standard and was, therefore, considered too confusing; hence, the reason for calling it the Plane Mask Register.

6.2.5 SR03—Character Font

I/O (and Memory Offset) Address: 3C5h (index=03h)
Default: 00h
Attributes: Read/Write

7 6	5	4	3 2	1 0
Reserved (00)	Char Map A Select (bit 0)	Char Map B Select (bit 0)	Character Map A Select (bits 2 and 1)	Character Map B Select (bits 2 and 1)

<table>
<tr><th>Bit</th><th>Descriptions</th></tr>
<tr><td>7:6</td><td>Reserved. Read as 0s.</td></tr>
<tr><td>3:2,5</td><td><b>Character Map Select Bits for Character Map B.</b> These three bits are used to select the character map (character generator tables) to be used as the secondary character set (font). Note that the numbering of the maps is not sequential.
<table>
<tr><th>Bit [3:2, 5]</th><th>Map Number</th><th>Table Location</th></tr>
<tr><td>00,0</td><td>0</td><td>1st 8KB of plane 2 at offset 0 (default)</td></tr>
<tr><td>00,1</td><td>4</td><td>2nd 8KB of plane 2 at offset 8K</td></tr>
<tr><td>01,0</td><td>1</td><td>3rd 8KB of plane 2 at offset 16K</td></tr>
<tr><td>01,1</td><td>5</td><td>4th 8KB of plane 2 at offset 24K</td></tr>
<tr><td>10,0</td><td>2</td><td>5th 8KB of plane 2 at offset 32K</td></tr>
<tr><td>10,1</td><td>6</td><td>6th 8KB of plane 2 at offset 40K</td></tr>
<tr><td>11,0</td><td>3</td><td>7th 8KB of plane 2 at offset 48K</td></tr>
<tr><td>11,1</td><td>7</td><td>8th 8KB of plane 2 at offset 56K</td></tr>
</table></td></tr>
<tr><td>1:0,4</td><td><b>Character Map Select Bits for Character Map A.</b> These three bits are used to select the character map (character generator tables) to be used as the primary character set (font). Note that the numbering of the maps is not sequential.
<table>
<tr><th>Bit [1:0,4]</th><th>Map Number</th><th>Table Location</th></tr>
<tr><td>0,00</td><td>0</td><td>1st 8KB of plane 2 at offset 0 (default)</td></tr>
<tr><td>0,01</td><td>4</td><td>2nd 8KB of plane 2 at offset 8K</td></tr>
<tr><td>0,10</td><td>1</td><td>3rd 8KB of plane 2 at offset 16K</td></tr>
<tr><td>0,11</td><td>5</td><td>4th 8KB of plane 2 at offset 24K</td></tr>
<tr><td>1,00</td><td>2</td><td>5th 8KB of plane 2 at offset 32K</td></tr>
<tr><td>1,01</td><td>6</td><td>6th 8KB of plane 2 at offset 40K</td></tr>
<tr><td>1,10</td><td>3</td><td>7th 8KB of plane 2 at offset 48K</td></tr>
<tr><td>1,11</td><td>7</td><td>8th 8KB of plane 2 at offset 56K</td></tr>
</table></td></tr>
</table>

NOTES:

1. In text modes, bit 3 of the video data's attribute byte normally controls the foreground intensity. This bit may be redefined to control switching between character sets. This latter function is enabled whenever there is a difference in the values of the Character Font Select A and the Character Font Select B bits. If the two values are the same, the character select function is disabled and attribute bit 3 controls the foreground intensity.
2. Bit 1 of the Memory Mode Register (SR04) must be set to 1 for the character font select function of this register to be active. Otherwise, only character maps 0 and 4 are available.

6.2.6 SR04—Memory Mode Register

I/O (and Memory Offset) Address: 3C5h (index=04h)
Default: 00h
Attributes: Read/Write

7 4	3	2	1	0
Reserved (0000)	Chain 4	Odd/Even	Extended Memory	Reserved (0)

Bit	Description
7:4	Reserved. Read as 0.
3	**Chain 4 Mode.** The selections made by this bit affect both CPU Read and write accesses to the frame buffer. 0 = The manner in which the frame buffer memory is mapped is determined by the setting of bit 2 of this register (default). 1 = The frame buffer memory is mapped in such a way that the function of address bits 0 and 1 are altered so that they select planes 0 through 3. This setting is used in mode x13 to allow all four planes to be accessed via sequential addresses.
2	**Odd/Even Mode.** Bit 3 of this register must be set to 0 for this bit to be effective. The selections made by this bit affect only non-paged CPU accesses to the frame buffer through the VGA aperture. 0 = The frame buffer memory is mapped in such a way that the function of address bit 0 such that even addresses select planes 0 and 2 and odd addresses select planes 1 and 3 (default). 1 = Addresses sequentially access data within a bit map, and the choice of which map is accessed is made according to the value of the Plane Mask Register (SR02).
1	**Extended Memory Enable.** This bit must be set to 1 to enable the selection and use of character maps in plane 2 via the Character Map Select Register (SR03). 0 = Disable CPU accesses to more than the first 64KB of VGA standard memory (default). 1 = Enable CPU accesses to the rest of the 256KB total VGA memory beyond the first 64KB.
0	Reserved. Read as 0.

6.2.7 SR07—Horizontal Character Counter Reset

I/O (and Memory Offset) Address: 3C5h (index=07h)
Default: 00h
Attributes: Read/Write

For standard VGAs, writing this register (with any data) causes the horizontal character counter to be held in reset (the character counter output will remain 0). It remained in reset until a write occurred to any other sequencer register location with SRX set to an index of 0 through 6. In this implementation that sequence has no such special effect.

The vertical line counter is clocked by a signal derived from the horizontal display enable (which does not occur if the horizontal counter is held in reset). Therefore, if a write occurs to this register during the vertical retrace interval, both the horizontal and vertical counters will be set to 0. A write to any other sequencer register location (with SRX set to an index of 0 through 6) may then be used to start both counters with reasonable synchronization to an external event via software control. Although this was a standard VGA register, it was not documented by IBM.

Bit	Description
7:0	**Horizontal Character Counter.**

6.3 Graphics Controller Registers

The graphics controller registers are accessed via either I/O space or Memory space. Accesses to the registers of the VGA Graphics Controller are done through the use of address 3CEh (or memory address 3CEh) written with the index of the desired register. Then the desired register is accessed through the data port for the graphics controller registers at I/O address 3CFh (or memory address 3CFh). Indexes 10 and 11 should only be accessed through the I/O space only.

6.3.1 GRX—GRX Graphics Controller Index Register

I/O (and Memory Offset) Address: 3CEh
Default: 000UUUUUb (U=Undefined)
Attributes: Read/Write

7 4	3 0
Reserved (0000)	Graphics Controller Register Index

Bit	Description
7:4	Reserved. Read as 0.
3:0	**Sequencer Register Index.** This field selects any one of the graphics controller registers (GR00-GR11]) to be accessed via the data port at I/O (or memory offset) location 3CFh.

6.3.2 GR00—Set/Reset Register

I/O (and Memory Offset) Address: 3CFh (index=00h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7 4	3	2	1	0
Reserved (0000)	Plane 3	Plane 2	Plane 1	Plane 0

Bit	Description
7:4	Reserved. Read as 0.
3:0	**Set/Reset Plane [3:0].** When the Write Mode bits (bits 0 and 1) of the Graphics Mode Register (GR05) are set to select Write Mode 0, all 8 bits of each byte of each memory plane are set to either 1 or 0 as specified in the corresponding bit in this register, if the corresponding bit in the Enable Set/Reset Register (GR01) is set to 1. When the Write Mode bits (bits 0 and 1) of the Graphics Mode Register (GR05) are set to select Write Mode 3, all CPU data written to the frame buffer is rotated, then logically ANDed with the contents of the Bit Mask Register (GR08), and then treated as the addressed data's bit mask, while value of these four bits of this register are treated as the color value.

6.3.3 GR01—Enable Set/Reset Register

I/O (and Memory Offset) Address: 3CFh (Index=01h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7 4	3	2	1	0
Reserved (0000)	Enable Set/ Reset Plane 3	Enable Set/ Reset Plane 2	Enable Set/ Reset Plane 1	Enable Set/ Reset Plane 0

Bit	Description
7:4	Reserved. Read as 0.
3:0	**Enable Set/Reset Plane [3:0].** This register works in conjunction with the Set/Reset Register (GR00). The Write Mode bits (bits 0 and 1) must be set for Write Mode 0 for this register to have any effect. 0 = The corresponding memory plane can be read from or written to by the CPU without any special bitwise operations taking place. 1 = The corresponding memory plane is set to 0 or 1 as specified in the Set/Reset Register (GR00).

6.3.4 GR02—Color Compare Register

I/O (and Memory Offset) Address: 3CFh (Index=02h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7 4	3	2	1	0
Reserved (0000)	Color Compare			
	Plane 3	Plane 2	Plane 1	Plane 0

Bit	Description
7:4	Reserved. Read as 0.
3:0	**Color Compare Plane [3:0].** When the Read Mode bit (bit 3) of the Graphics Mode Register (GR05) is set to select Read Mode 1, all 8 bits of each byte of each of the 4 memory planes of the frame buffer corresponding to the address from which a CPU read access is being performed are compared to the corresponding bits in this register (if the corresponding bit in the Color Don't Care Register (GR07) is set to 1). The value that the CPU receives from the read access is an 8-bit value that shows the result of this comparison, wherein value of 1 in a given bit position indicates that all of the corresponding bits in the bytes across all of the memory planes that were included in the comparison had the same value as their memory plane's respective bits in this register.

6.3.5 GR03—Data Rotate Register

I/O (and Memory Offset) Address: 3CFh (Index=03h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7 5	4 3	2 0
Reserved	Function Select	Rotate Count

Bit	Description
7:5	Reserved. Read as 0.
4:3	**Function Select.** These bits specify the logical function (if any) to be performed on data that is meant to be written to the frame buffer (using the contents of the memory read latch) just before it is actually stored in the frame buffer at the intended address location. 00 = Data being written to the frame buffer remains unchanged, and is simply stored in the frame buffer. 01 = Data being written to the frame buffer is logically ANDed with the data in the memory read latch before it is actually stored in the frame buffer. 10 = Data being written to the frame buffer is logically ORed with the data in the memory read latch before it is actually stored in the frame buffer. 11 = Data being written to the frame buffer is logically XORed with the data in the memory read latch before it is actually stored in the frame buffer.
2:0	**Rotate Count.** These bits specify the number of bits to the right to rotate any data that is meant to be written to the frame buffer just before it is actually stored in the frame buffer at the intended address location.

6.3.6 GR04—Read Plane Select Register

I/O (and Memory Offset) Address: 3CFh (Index=04h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7 2	1 0
Reserved (000000)	Read Plane Select

Bit	Description
7:2	Reserved. Read as 0.
1:0	**Read Plane Select.** These two bits select the memory plane from which the CPU reads data in Read Mode 0. In Odd/Even Mode, bit 0 of this register is ignored. In Chain 4 Mode, both bits 1 and 0 of this register are ignored. The four memory planes are selected as follows: 00 = Plane 0 01 = Plane 1 10 = Plane 2 11 = Plane 3 These two bits also select which of the four memory read latches may be read via the Memory read Latch Data Register (CR22). The choice of memory read latch corresponds to the choice of plane specified in the table above. The Memory Read Latch Data register and this additional function served by 2 bits are features of the VGA standard that were never documented by IBM.

6.3.7 GR05—Graphics Mode Register

I/O (and Memory Offset) Address: 3CFh (Index=05h)
Default: 0UUU U0UUb (U=Undefined)
Attributes: Read/Write

7	6 5	4	3	2	1 0
Reserved (0)	Shift Register Control	Odd/Even	Read Mode	Reserved	Write Mode

Bit	Description
7	Reserved. Read as 0.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>6:5</td><td>
<p><b>Shift Register Control.</b> In standard VGA modes, pixel data is transferred from the 4 graphics memory planes to the palette via a set of 4 serial output bits. These 2 bits of this register control the format in which data in the 4 memory planes is serialized for these transfers to the palette.</p>
<p><b>Bits [6:5]=00</b></p>
<p>One bit of data at a time from parallel bytes in each of the 4 memory planes is transferred to the palette via the 4 serial output bits, with 1 of each of the serial output bits corresponding to a memory plane. This provides a 4-bit value on each transfer for 1 pixel, making possible a choice of 1 of 16 colors per pixel.</p>
<table>
<tr><th>Serial</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><th>Out</th><th>1st Xfer</th><th>2nd Xfer</th><th>3rd Xfer</th><th>4th Xfer</th><th>5th Xfer</th><th>6th Xfer</th><th>7th Xfer</th><th>8th Xfer</th></tr>
<tr><td>Bit 3</td><td>plane 3 bit 7</td><td>plane 3 bit 6</td><td>plane 3 bit 5</td><td>plane 3 bit 4</td><td>plane 3 bit 3</td><td>plane 3 bit 2</td><td>plane 3 bit 1</td><td>plane 3 bit 0</td></tr>
<tr><td>Bit 2</td><td>plane 2 bit 7</td><td>plane 2 bit 6</td><td>plane 2 bit 5</td><td>plane 2 bit 4</td><td>plane 2 bit 3</td><td>plane 2 bit 2</td><td>plane 2 bit 1</td><td>plane 2 bit 0</td></tr>
<tr><td>Bit 1</td><td>plane 1 bit 7</td><td>plane 1 bit 6</td><td>plane 1 bit 5</td><td>plane 1 bit 4</td><td>plane 1 bit 3</td><td>plane 1 bit 2</td><td>plane 1 bit 1</td><td>plane 1 bit 0</td></tr>
<tr><td>Bit 0</td><td>plane 0 bit 7</td><td>plane 0 bit 6</td><td>plane 0 bit 5</td><td>plane 0 bit 4</td><td>plane 0 bit 3</td><td>plane 0 bit 2</td><td>plane 0 bit 1</td><td>plane 0 bit 0</td></tr>
</table>
<p><b>Bits [6:5]=01</b></p>
<p>Two bits of data at a time from parallel bytes in each of the 4 memory planes are transferred to the palette in a pattern that alternates per byte between memory planes 0 and 2, and memory planes 1 and 3. First the even-numbered and odd-numbered bits of a byte in memory plane 0 are transferred via serial output bits 0 and 1, respectively, while the even-numbered and odd-numbered bits of a byte in memory plane 2 are transferred via serial output bits 2 and 3. Next, the even-numbered and odd-numbered bits of a byte in memory plane 1 are transferred via serial output bits 0 and 1, respectively, while the even-numbered and odd-numbered bits of memory plane 3 are transferred via serial out bits 1 and 3. This provides a pair of 2-bit values (one 2-bit value for each of 2 pixels) on each transfer, making possible a choice of 1 of 4 colors per pixel.</p>
<table>
<tr><th>Serial</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><th>Out</th><th>1st Xfer</th><th>2nd Xfer</th><th>3rd Xfer</th><th>4th Xfer</th><th>5th Xfer</th><th>6th Xfer</th><th>7th Xfer</th><th>8th Xfer</th></tr>
<tr><td>Bit 3</td><td>plane 2 bit 7</td><td>plane 2 bit 5</td><td>plane 2 bit 3</td><td>plane 2 bit 1</td><td>plane 3 bit 7</td><td>plane 3 bit 5</td><td>plane 3 bit 3</td><td>plane 3 bit 1</td></tr>
<tr><td>Bit 2</td><td>plane 2 bit 6</td><td>plane 2 bit 4</td><td>plane 2 bit 2</td><td>plane 2 bit 0</td><td>plane 3 bit 6</td><td>plane 3 bit 4</td><td>plane 3 bit 2</td><td>plane 3 bit 0</td></tr>
<tr><td>Bit 1</td><td>plane 0 bit 7</td><td>plane 0 bit 5</td><td>plane 0 bit 3</td><td>plane 0 bit 1</td><td>plane 1 bit 7</td><td>plane 1 bit 5</td><td>plane 1 bit 3</td><td>plane 1 bit 1</td></tr>
<tr><td>Bit 0</td><td>plane 0 bit 6</td><td>plane 0 bit 4</td><td>plane 0 bit 2</td><td>plane 0 bit 0</td><td>plane 1 bit 6</td><td>plane 1 bit 4</td><td>plane 1 bit 2</td><td>plane 1 bit 0</td></tr>
</table>
<p>This alternating pattern is meant to accommodate the use of the Odd/Even mode of organizing the 4 memory planes, which is used by standard VGA modes 2h and 3h.</p>
<p><b>Bits [6:5]=1x</b></p>
<p>Four bits of data at a time from parallel bytes in each of the 4 memory planes are transferred to the palette in a pattern that iterates per byte through memory planes 0 through 3. First the 4 most significant bits of a byte in memory plane 0 are transferred via the 4 serial output bits, followed by the 4 least significant bits of the same byte. Next, the same transfers occur from the parallel byte in memory planes 1, 2 and lastly, 3. Each transfer provides either the upper or lower half of an 8-bit value for the color for each pixel, making possible a choice of 1 of 256 colors per pixel. This is the setting used in mode x13.</p>
<table>
<tr><th>Serial</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><th>Out</th><th>1st Xfer</th><th>2nd Xfer</th><th>3rd Xfer</th><th>4th Xfer</th><th>5th Xfer</th><th>6th Xfer</th><th>7th Xfer</th><th>8th Xfer</th></tr>
<tr><td>Bit 3</td><td>plane 0 bit 7</td><td>plane 0 bit 3</td><td>plane 1 bit 7</td><td>plane 1 bit 3</td><td>plane 2 bit 7</td><td>plane 2 bit 3</td><td>plane 3 bit 7</td><td>plane 3 bit 3</td></tr>
<tr><td>Bit 2</td><td>plane 0 bit 6</td><td>plane 0 bit 2</td><td>plane 1 bit 6</td><td>plane 1 bit 2</td><td>plane 2 bit 6</td><td>plane 2 bit 2</td><td>plane 3 bit 6</td><td>plane 3 bit 2</td></tr>
<tr><td>Bit 1</td><td>plane 0 bit 5</td><td>plane 0 bit 1</td><td>plane 1 bit 5</td><td>plane 1 bit 1</td><td>plane 2 bit 5</td><td>plane 2 bit 1</td><td>plane 3 bit 5</td><td>plane 3 bit 1</td></tr>
<tr><td>Bit 0</td><td>plane 0 bit 4</td><td>plane 0 bit 0</td><td>plane 1 bit 4</td><td>plane 1 bit 0</td><td>plane 2 bit 4</td><td>plane 2 bit 0</td><td>plane 3 bit 4</td><td>plane 3 bit 0</td></tr>
</table>
<p>This pattern is meant to accommodate mode 13h, a standard VGA 256-color graphics mode.</p>
</td></tr>
</table>

Bit	Description
4	**Odd/Even Mode.** 0 = Addresses sequentially access data within a bit map, and the choice of which map is accessed is made according to the value of the Plane Mask Register (SR02). 1 = The frame buffer is mapped in such a way that the function of address bit 0 is such that even addresses select memory planes 0 and 2 and odd addresses select memory planes 1 and 3. **Note:** This works in a way that is the inverse of (and is normally set to be the opposite of) bit 2 of the Memory Mode Register (SR02).
3	**Read Mode.** 0 = During a CPU read from the frame buffer, the value returned to the CPU is data from the memory plane selected by bits 1 and 0 of the Read Plane Select Register (GR04). 1 = During a CPU read from the frame buffer, all 8 bits of the byte in each of the 4 memory planes corresponding to the address from which a CPU read access is being performed are compared to the corresponding bits in this register (if the corresponding bit in the Color Don't Care Register (GR07) is set to 1). The value that the CPU receives from the read access is an 8-bit value that shows the result of this comparison. A value of 1 in a given bit position indicates that all of the corresponding bits in the bytes across all 4 of the memory planes that were included in the comparison had the same value as their memory plane's respective bits in this register.
2	Reserved. Read as 0.
1:0	**Write Mode.** 00 = Write Mode 0 — During a CPU write to the frame buffer, the addressed byte in each of the 4 memory planes is written with the CPU write data after it has been rotated by the number of counts specified in the Data Rotate Register (GR03). If, however, the bit(s) in the Enable Set/Reset Register (GR01) corresponding to one or more of the memory planes is set to 1, then those memory planes will be written to with the data stored in the corresponding bits in the Set/Reset Register (GR00). 01 = Write Mode 1 — During a CPU write to the frame buffer, the addressed byte in each of the 4 memory planes is written to with the data stored in the memory read latches. (The memory read latches stores an unaltered copy of the data last read from any location in the frame buffer.) 10 = Write Mode 2 — During a CPU write to the frame buffer, the least significant 4 data bits of the CPU write data is treated as the color value for the pixels in the addressed byte in all 4 memory planes. The 8 bits of the Bit Mask Register (GR08) are used to selectively enable or disable the ability to write to the corresponding bit in each of the 4 memory planes that correspond to a given pixel. A setting of 0 in a bit in the Bit Mask Register at a given bit position causes the bits in the corresponding bit positions in the addressed byte in all 4 memory planes to be written with value of their counterparts in the memory read latches. A setting of 1 in a Bit Mask Register at a given bit position causes the bits in the corresponding bit positions in the addressed byte in all 4 memory planes to be written with the 4 bits taken from the CPU write data to thereby cause the pixel corresponding to these bits to be set to the color value. 11 = Write Mode 3 — During a CPU write to the frame buffer, the CPU write data is logically ANDed with the contents of the Bit Mask Register (GR08). The result of this ANDing is treated as the bit mask used in writing the contents of the Set/Reset Register (GR00) are written to addressed byte in all 4 memory planes.

6.3.8 GR06—Miscellaneous Register

I/O (and Memory Offset) Address: 3CFh (Index=06h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7 4	3 2	1	0
Reserved (0000)	Memory Map Mode	Chain Odd/Even	Graphics / Text Mode

Bit	Description
7:4	Reserved. Read as 0s.
3:2	**Memory Map Mode.** These 2 bits control the mapping of the VGA address range for frame buffer into the CPU address space as follows: **Bit [3:2] Frame Buffer Address Range** 00 A0000h – BFFFFh 01 A0000h – AFFFFh 10 B0000h – B7FFFh 11 B8000h – BFFFFh **Note:** This function is used in both in standard VGA modes, extended VGA modes (132 column text), and in non-VGA modes (hi-res). (132 column text modes are no longer supported). VGA aperture memory accesses are also controlled by the PCI configuration Memory Enable bit and the RAM enable bit in the Miscellaneous Output Register (3c2/3cc). For accesses using GR10 and GR11 to paged VGA RAM or to device MMIO registers, set these bits to 01to select the (A0000-AFFFF) range. The CPU must map this memory as uncacheable (UC); see **VGA Host Access Memory Munging** in *Display and Overlay Functions*.
1	**Chain Odd/Even.** This bit provides the ability to alter the interpretation of address bit A0, so that it may be used in selecting between the odd-numbered memory planes (planes 1 and 3) and the even-numbered memory planes (planes 0 and 2). 0 = A0 functions normally. 1 = A0 is switched with a high order address bit, in terms of how it is used in address decoding. The result is that A0 is used to determine which memory plane is being accessed (A0=0 for planes 0 and 2 and A0=1 for planes 1 and 3).
0	**Graphics/Text Mode.** This is one of two bits that are used to determine if the VGA is operating in text or graphics modes. The other bit is in AR10[0], these two bits need to be programmed in a consistent manner to achieve the proper results. 0 = Text mode. 1 = Graphics mode.

6.3.9 GR07—Color Don’t Care Register

I/O (and Memory Offset) Address: 3CFh (Index=07h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7	4	3	2	1	0
Reserved (0000)		Ignore Color Plane 3	Ignore Color Plane 2	Ignore Color Plane 1	Ignore Color Plane 0

Bit	Description
7:4	Reserved. Read as 0.
3:0	**Ignore Color Plane [3:0].** Note that these bits have effect only when bit 3 of the Graphics Mode Register (GR05) is set to 1 to select read mode 1. 0 = The corresponding bit in the Color Compare Register (GR02) will not be included in color comparisons. 1 = The corresponding bit in the Color Compare Register (GR02) is used in color comparisons.

6.3.10 GR08—Bit Mask Register

I/O (and Memory Offset) Address: 3CFh (Index=08h)
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Bit Mask.** 0 = The corresponding bit in each of the 4 memory planes is written to with the corresponding bit in the memory read latches. 1 = Manipulation of the corresponding bit in each of the 4 memory planes via other mechanisms is enabled. **Notes:** This bit mask applies to any writes to the addressed byte of any or all of the 4 memory planes, simultaneously. This bit mask is applicable to any data written into the frame buffer by the CPU, including data that is also subject to rotation, logical functions (AND, OR, XOR), and Set/Reset. To perform a proper read-modify-write cycle into frame buffer, each byte must first be read from the frame buffer by the CPU (and this will cause it to be stored in the memory read latches), this Bit Mask Register must be set, and the new data then written into the frame buffer by the CPU.

6.3.11 GR10—Address Mapping

I/O (avoid MMIO access) Address: 3CFh (Index=10h)
Default: 00h
Attributes: R/W

This register should only be accessed using I/O operations and never be accessed through the A/B segment addressing map, I/O space register map, or direct MMIO operations.

7 4	3	2 1	0
Page Select Extension	I/O Map Enable	Paging Target	Page Mapping

Bit	Description
7:4	**Page Select Extension** These bits form the upper bits of a 12-bit page selection value. They when combined with the GR11 <7:0> bits they define the offset into stolen memory to the 64KB page that is accessible via the VGA Memory paging mechanism. This register provides the upper Address[27:24] bits for the access allowing for a maximum of 256MB address range. The actual range that can be used is limited to the size of the stolen graphics memory region. Addresses specified through these bits and the bits in GR11 should be legal addresses that have been limited to the size of stolen memory, accesses outside this range produce unspecified results.
3	Reserved
2:1	**Paging Map Target.** When paging is enabled, these bits determine the target for data cycle accesses through the VGA memory aperture. VGA Graphics pre-allocated memory is used for cases where there is no local memory and in devices that do not support local memory. Local memory is used when it exists. Memory mapped register access is only available through this mechanism in a few devices. 00 = VGA Graphics Memory (Local/pre-allocated starting at a base of local address zero) 01 = Reserved (Was Memory Mapped Registers for previous devices) 10 = Reserved (Was Video BIOS ROM Memory only for discrete graphics devices) 11 = Reserved

Bit	Description
0	**Page Mapping Enable.** This mode allows the mapping of the VGA memory address space to either VGA memory (pre-allocated or local). In previous devices it was used to map MMIO registers, or Video BIOS ROM. This allowed video BIOS access to the entire memory mapped register space, allowed real mode DOS applications for BIOS flash operations, and extended video mode support for DOS applications in cases where the frame buffer is greater than the 64K bytes. **Some Notes on Paging.** Once this is enabled, no VGA memory address swizzel will be performed; addresses are directly mapped to memory. The same thing was true for ROM or register accesses for devices that support that operation. A single paging register is used to map the 64KB [A0000:AFFFF] window. An internal address is generated using GR11[7:0] as the address lines [23:16] extension to the lower address lines of the access A[15:2]. When mapping is enabled, the B0000:BFFFF area is always disabled using GR06<3:2>=01. The use of addresses in the A0000-BFFFF range requires that both the graphics device PCI configuration memory enable and MSR<1> be enabled. **Graphics Mode Select** (GMS). This field is used to select the amount of Main Memory that is “pre-allocated” to support the Internal Graphics device in VGA (non-linear) mode only. These 2 bits are valid only when Internal graphics is enabled. **(Paging must not step out of pre-allocated memory within main memory)** In discrete graphics devices, ROM pins are sometimes shared with other functions such as capture. Care must be taken to insure that ROM accesses do not conflict with other ongoing operations. In cases where SMM code is executing out of the A/B segment with both code and data cycles target the SMM memory, the programming of the target to memory mapped registers does not override SMM target selection. 0 = Disable (default) 1 = Enable

6.3.12 GR11—Page Selector

I/O Address (avoid MMIO access): 3CFh (Index=11h)
Default: 00h
Attributes: R/W

Bit	Description
7:0	**Page Select.** When concatenated with the GR10<7:4> bits, selects a 64KB window within target area when Page Mapping is enabled (GR10[0]=1). This requires that the graphics device PCI configuration space memory enable, the GR06<3:2> bits to be 01 (select A0000-AFFFF only), and the MSR<1:1> bit to be set. This register provides the Address[23:16] bits for the access allowing for a 256MB address range. VGA paging of frame buffer memory is for non-VGA packed modes only and should not be enabled when using basic VGA modes. This register should only be accessed using I/O operations and never be accessed through the A000 segment addressing map or direct MMIO operations. Addresses generated via this method should be restricted to within the size of the pre-allocated (stolen) memory that is currently available.

6.3.13 GR18—Software Flags

I/O (and Memory Offset) Address: 3CFh (Index=18h)
Default: 00
Attribute: R/W

Bit	Description
7:0	**Software Flags.** Used as scratch pad space in video BIOS. These bits are separate from the bits which appear in the memory mapped IO space. They are used specifically by the SMI BIOS which does not have access to memory mapped IO at the time they are required. These register bits have no effect on H/W operation.

6.4 Attribute Controller Registers

Unlike the other sets of indexed registers, the attribute controller registers are not accessed through a scheme employing entirely separate index and data ports. I/O address 3C0h (or memory address 3C0h) is used both as the read and write for the index register, and as the write address for the data port. I/O address 3C1h (or memory address 3C1h) is the read address for the data port.

To write to the attribute controller registers, the index of the desired register must be written to I/O address 3C0h (or memory address 3C0h), and then the data is written to the very same I/O (memory) address. A flip-flop alternates with each write to I/O address 3C0h (or memory address 3C0h) to change its function from writing the index to writing the actual data, and back again. This flip-flop may be deliberately set so that I/O address 3C0h (or memory address 3C0h) is set to write to the index (which provides a way to set it to a known state) by performing a read operation from Input Status Register 1 (ST01) at I/O address 3BAh (or memory address 3BAh) or 3DAh (or memory address 3DAh), depending on whether the graphics system has been set to emulate an MDA or a CGA as per MSR[0].

To read from the attribute controller registers, the index of the desired register must be written to I/O address 3C0h (or memory address 3C0h), and then the data is read from I/O address 3C1h (or memory address 3C1h). A read operation from I/O address 3C1h (or memory address 3C1h) does not reset the flip-flop to writing to the index. Only a write to 3C0h (or memory address 3C0h) or a read from 3BAh or 3DAh (or memory address 3BAh or 3DAh), as described above, will toggle the flip-flop back to writing to the index.

6.4.1 ARX—Attribute Controller Index Register

I/O (and Memory Offset) Address: 3C0h
Default: 00UU UUUUb (U=Undefined)
Attributes: Read/Write

7 6	5	4 0
Reserved (00)	Video Enable	Attribute Controller Register Index

Bit	Description
7:6	Reserved. Read as 0s.
5	**Video Enable.** Note that In the VGA standard, this is called the "Palette Address Source" bit. Clearing this bit will cause the VGA display data to become all 00 index values. For the default palette, this will cause a black screen. The video timing signals continue. Another control bit will turn video off and stop the data fetches. 0 = Disable. Attribute controller color registers (AR[00:0F]) can be accessed by the CPU. 1 = Enable. Attribute controller color registers (AR[00:0F]) are inaccessible by the CPU.
4:0	**Attribute Controller Register Index.** These five bits are used to select any one of the attribute controller registers (AR[00:14]), to be accessed. **Note:** AR12 is referred to in the VGA standard as the Color Plane Enable Register. The words "plane," "color plane," "display memory plane," and "memory map" have been all been used in IBM* literature on the VGA standard to describe the four separate regions in the frame buffer where the pixel color or attribute information is split up and stored in standard VGA planar modes. This use of multiple terms for the same subject was deemed to be confusing; therefore, AR12 is called the Memory Plane Enable Register. Attribute Controller Register Index.

6.4.2 AR[00:0F]—Palette Registers [0:F]

I/O (and Memory Offset) Address: Read at 3C1h and Write at 3C0h; (index=00h-0Fh)
Default: 00UU UUUUb (U=Undefined)
Attributes: Read/Write

7 6	5 0
Reserved	Palette Bits P[5:0]

Bit	Description
7:6	Reserved. Read as 0.
5:0	**Palette Bits P[5:0].** In each of these 16 registers, these are the lower 6 of 8 bits that are used to map either text attributes or pixel color input values (for modes that use 16 colors) to the 256 possible colors available to be selected in the palette. **Note:** Bits 3 and 2 of the Color Select Register (AR14) supply bits P7 and P6 for the values contained in all 16 of these registers. Bits 1 and 0 of the Color Select Register (AR14) can also replace bits P5 and P4 for the values contained in all 16 of these registers, if bit 7 of the Mode Control Register (AR10) is set to 1.

6.4.3 AR10—Mode Control Register

I/O (and Memory Offset) Address: Read at 3C1h and Write at 3C0h; (index=10h)
Default: UUh (U=Undefined)
Attributes: Read/Write

7	6	5	4	3	2	1	0
Palette Bits P5, P4 Select	Pixel Width/ Clock Select	Pixel Panning Compat	Reserved (0)	Enable Blink/ Select Bkgnd Int	Enable Line Graphics Char Code	Select Display Type	Graphics/ Alpha Mode

Bit	Description
7	**Palette Bits P5, P4 Select.** 0 = P5 and P4 for each of the 16 selected colors (for modes that use 16 colors) are individually provided by bits 5 and 4 of their corresponding Palette Registers (AR[00:0F]). 1 = P5 and P4 for all 16 of the selected colors (for modes that use 16 colors) are provided by bits 1 and 0 of Color Select Register (AR14).

Bit	Description
6	**Pixel Width/Clock Select.** 0 = Six bits of video data (translated from 4 bits via the palette) are output every dot clock. 1 = Two sets of 4 bits of data are assembled to generate 8 bits of video data which is output every other dot clock, and the Palette Registers (AR[00:0F]) are bypassed. **Note:** This bit is set to 0 for all of the standard VGA modes, except mode 13h.
5	**Pixel Panning Compatibility.** 0 = Scroll both the upper and lower screen regions horizontally as specified in the Pixel Panning Register (AR13). 1 = Scroll only the upper screen region horizontally as specified in the Pixel Panning Register (AR13). **Note:** This bit has application only when split-screen mode is being used, where the display area is divided into distinct upper and lower regions which function somewhat like separate displays.
4	Reserved. Read as 0.
3	**Enable Blinking/Select Background Intensity.** 0 = Disables blinking in graphics modes, and for text modes, sets bit 7 of the character attribute bytes to control background intensity, instead of blinking. 1 = Enables blinking in graphics modes and for text modes, sets bit 7 of the character attribute bytes to control blinking, instead of background intensity. **Note:** The blinking rate is derived by dividing the VSYNC signal. The Blink Rate Control field of the VGA control register defines the blinking rate.
2	**Enable Line Graphics Character Code.** 0 = Every 9th pixel of a horizontal line (i.e., the last pixel of each horizontal line of each 9-pixel wide character box) is assigned the same attributes as the background of the character of which the given pixel is a part. 1 = Every 9th pixel of a horizontal line (i.e., the last pixel of each horizontal line of each 9-pixel wide character box) is assigned the same attributes as the 8th pixel if the character of which the given pixel is a part. This setting is intended to accommodate the line-drawing characters of the PC's extended ASCII character set -- characters with an extended ASCII code in the range of B0h to DFh. **Note:** In IBM* literature describing the VGA standard, the range of extended ASCII codes that are said to include the line-drawing characters is mistakenly specified as C0h to DFh, rather than the correct range of B0h to DFh.
1	**Select Display Type.** 0 = Attribute bytes in text modes are interpreted as they would be for a color display. 1 = Attribute bytes in text modes are interpreted as they would be for a monochrome display.
0	**Graphics/Alphanumeric Mode.** This bit (along with GR06[0]) select either graphics mode or text mode. These two bits must be programmed in a consistent manner to achieve the desired results. 0 = Alphanumeric (text) mode. 1 = Graphics mode.

6.4.4 AR11—Overscan Color Register

I/O (and Memory Offset) Address:	Read at 3C1h and Write at 3C0h; (index=11h)
Default:	UUh (U=Undefined)
Attributes:	Read/Write

Bit	Description
7:0	**Overscan.** These 8 bits select the overscan (border) color index value. The actual border color will be determined by the contents of the palette at the selected index. The border color is displayed between the end of active and the beginning of blank or the end of blank and the beginning of active on CRT type devices driven from the DAC output port. For native VGA modes on digital display ports there is the option of including the border in the active region or not depending on a control bit in the port control register. For centered VGA modes, the VGA control register determines if the border is included in the centered region or not. For monochrome displays, this value should be set to 00h.

6.4.5 AR12—Memory Plane Enable Register

I/O (and Memory Offset) Address:	Read at 3C1h and Write at 3C0h; (index=12h)
Default:	00UU UUUUb (U=Undefined)
Attributes:	Read/Write

7 6	5 4	3	2	1	0
Reserved (00)	Video Status Mux	Enable Plane 3	Enable Plane 2	Enable Plane 1	Enable Plane 0

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7:6</td><td>Reserved. Read as 0.</td></tr>
<tr><td>5:4</td><td><b>Video Status Mux.</b> These 2 bits are used to select 2 of the 8 possible palette bits (P7-P0) to be made available to be read via bits 5 and 4 of the Input Status Register 1 (ST01). The table below shows the possible choices.
<table>
<tr><th>Bit [5:4]</th><th>ST01 Bit 5</th><th>ST01 Bit 4</th></tr>
<tr><td>00</td><td>P2 (default)</td><td>P0 (default)</td></tr>
<tr><td>01</td><td>P5</td><td>P4</td></tr>
<tr><td>10</td><td>P3</td><td>P1</td></tr>
<tr><td>11</td><td>P7</td><td>P6</td></tr>
</table>
These bits are typically unused by current software; they are provided for EGA compatibility.</td></tr>
<tr><td>3:0</td><td><b>Enable Plane [3:0].</b> These 4 bits individually enable the use of each of the 4 memory planes in providing 1 of the 4 bits used in video output to select 1 of 16 possible colors from the palette to be displayed.
0 = Disable the use of the corresponding memory plane in video output to select colors, forcing the bit that the corresponding memory plane would have provided to a value of 0.
1 = Enable the use of the corresponding memory plane in video output to select colors.
<b>Note:</b> AR12 is referred to in the VGA standard as the Color Plane Enable Register. The words “plane,” “color plane,” “display memory plane,” and “memory map” have been all been used in IBM™ literature on the VGA standard to describe the 4 separate regions in the frame buffer that are amongst which pixel color or attributes information is split up and stored in standard VGA planar modes. This use of multiple terms for the same subject was considered confusing; therefore, AR12 is called the Memory Plane Enable Register.</td></tr>
</table>

6.4.6 AR13—Horizontal Pixel Panning Register

I/O (and Memory Offset) Address: Read at 3C1h and Write at 3C0h; (index=13h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7 4	3 0
Reserved (0000)	Horizontal Pixel Shift

<table>
<tr><th>Bit</th><th colspan="5">Description</th></tr>
<tr><td>7:4</td><td colspan="5">Reserved</td></tr>
<tr><td rowspan="12">3:0</td><td colspan="5"><b>Horizontal Pixel Shift 3-0.</b> This field holds a 4-bit value that selects the number of pixels by which the image is shifted horizontally to the left. This function is available in both text and graphics modes and allows for pixel panning.
In text modes with a 9-pixel wide character box, the image can be shifted up to 9 pixels to the left. In text modes with an 8-pixel wide character box, and in graphics modes other than those with 256 colors, the image can be shifted up to 8 pixels to the left. A pseudo 9-bit mode is when the 9-dot character is selected but overridden by the VGA control bit.
In standard VGA mode 13h (where bit 6 of the Mode Control Register, AR10, is set to 1 to support 256 colors), bit 0 of this register must remain set to 0, and the image may be shifted up to only 4 pixels to the left. In this mode, the number of pixels by which the image is shifted can be further controlled using bits 6 and 5 of the Preset Row Scan Register (CR08).</td></tr>
<tr><th colspan="5">Number of Pixels Shifted</th></tr>
<tr><th>Bits [3:0]</th><th>9-dot</th><th>Pseudo 9-dot</th><th>8-dot</th><th>256-Color</th></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>2</td><td>2</td><td>1</td><td>Undefined</td></tr>
<tr><td>2</td><td>3</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>3</td><td>4</td><td>4</td><td>3</td><td>Undefined</td></tr>
<tr><td>4</td><td>5</td><td>5</td><td>4</td><td>2</td></tr>
<tr><td>5</td><td>6</td><td>6</td><td>5</td><td>Undefined</td></tr>
<tr><td>6</td><td>7</td><td>7</td><td>6</td><td>3</td></tr>
<tr><td>7</td><td>8</td><td>7</td><td>7</td><td>Undefined</td></tr>
<tr><td>8</td><td>0</td><td>0</td><td>Undefined</td><td>Undefined</td></tr>
</table>

6.4.7 AR14—Color Select Register

I/O (and Memory Offset) Address: Read at 3C1h and Write at 3C0h; (index=14h)
Default: 0Uh (U=Undefined)
Attributes: Read/Write

7 4	3	2	1	0
Reserved (0000)	P7	P6	Alt P5	Alt P4

Bit	Description
7:4	Reserved
3:2	**Palette Bits P[7:6].** These are the 2 upper-most of the 8 bits that are used to map either text attributes or pixel color input values (for modes that use 16 colors) to the 256 possible colors contained in the palette. These 2 bits are common to all 16 sets of bits P5 through P0 that are individually supplied by Palette Registers 0-F (AR[00:0F]).
1:0	**Alternate Palette Bits P[5:4].** These 2 bits can be used as an alternate version of palette bits P5 and P4. Unlike the P5 and P4 bits that are individually supplied by Palette Registers 0-F (AR[00:0F]), these 2 alternate palette bits are common to all 16 of Palette Registers. Bit 7 of the Mode Control Register (AR10) is used to select between the use of either the P5 and P4 bits that are individually supplied by the 16 Palette Registers or these 2 alternate palette bits.

6.5 VGA Color Palette Registers

In devices that have two display pipes, there are two palettes, one for each display pipe. These palettes are the same for VGA modes and non-VGA modes. Accesses through VGA register methods can optionally read or write from either one.

For each palette in the device, for each palette, the color data stored in these 256 color data positions can be accessed only through a complex sub-addressing scheme, using a data register and two index registers. The Palette Data Register at I/O address 3C9h (or memory address offset 3C1h) is the data port. The Palette Read Index Register at I/O address 3C7h (or memory address offset 3C7h) and the Palette Write Index Register at I/O address 3C8h (or memory address offset 3C8h) are the two index registers. The Palette Read Index Register is the index register that is used to choose the color data position that is to be read from via the data port, while the Palette Write Index Register is the index register that is used to choose the color data position that is to be written to through the same data port. This arrangement allows the same data port to be used for reading from and writing to two different color data positions. Reading and writing the color data at a color data position involves three successive reads or writes since the color data stored at each color data position consists of three bytes.

To read a palette color data position, the index of the desired color data position must first be written to the Palette Read Index Register. Then all three bytes of data in a given color data position may be read at the Palette Data Register. The first byte read from the Palette Data Register retrieves the 8-bit value specifying the intensity of the red color component. The second and third bytes read are the corresponding 8-bit values for the green and blue color components respectively. After completing the third read operation, the Palette Read Index Register is automatically incremented so that the data of the next color data position becomes accessible for being read. This allows the contents of all of the 256 color data positions of the palette to be read in sequence. This is done by specifying only the index of the 0th color data position in the Palette Read Index Register, and then simply performing 768 successive reads from the Palette Data Register.

Writing a color data position, entails a very similar procedure. The index of the desired color data position must first be written to the Palette Write Index Register. Then all three bytes of data to specify a given color may be written to the Palette Data Register. The first byte written to the Palette Data Register specifies the intensity of the red color component, the second byte specifies the intensity for the green color component, and the third byte specifies the same for the blue color component. One important detail is that all three of these bytes must be written before the hardware will actually update these three values in the given color data position. When all three bytes have been written, the Palette Write Index Register is automatically incremented so that the data of the next color data position becomes accessible for being written. This allows the contents of all of the 256 color data positions of the palette to be written in sequence. This is done by specifying only the index of the 0th color data position in the Palette Write Index Register, and then simply performing 768 successive writes to the Palette Data Register.

6.5.1 DACMASK—Pixel Data Mask Register

I/O (and Memory Offset) Address:	3C6h
Default:	Undefined
Attributes:	Read/Write

Bit	Description
7:0	**Pixel Data Mask.** In indexed-color mode, the 8 bits of this register are logically ANDed with the 8 bits of pixel data received from the frame buffer for each pixel. The result of this ANDing process becomes the actual index used to select color data positions within the palette. This has the effect of limiting the choice of color data positions that may be specified by the incoming 8-bit data. 0 = Corresponding bit in the resulting 8-bit index being forced to 0. 1 = Allows the corresponding bit in the resulting index to reflect the actual value of the corresponding bit in the incoming 8-bit pixel data.

6.5.2 DACSTATE—DAC State Register

I/O (and Memory Offset) Address: 3C7h
Default: 00h
Attributes: Read-Only

7	2	1	0
Reserved (000000)		DAC State	

Bit	Description
7:2	Reserved. Read as 0.
1:0	**DAC State.** This field indicates which of the two index registers was most recently written. **Bits [1:0] Index Register Indicated** 00 = Palette Write Index Register at I/O Address 3C7h (default) 01 = Reserved 10 = Reserved 11 = Palette Read Index Register at I/O Address 3C8h

6.5.3 DACRX—Palette Read Index Register

I/O (and Memory Offset) Address: 3C7h
Default: 00h
Attributes: Write-Only

Bit	Description
7:0	**Palette Read Index.** The 8-bit index value programmed into this register chooses which of 256 standard color data positions within the palette are to be made accessible for being read from via the Palette Data Register (DACDATA). The index value held in this register is automatically incremented when all three bytes of the color data position selected by the current index have been read. A write to this register will abort a uncompleted palette write sequence. This register allows access to the palette even when running non-VGA display modes.

6.5.4 DACWX—Palette Write Index Register

I/O (and Memory Offset) Address: 3C8h
Default: 00h
Attributes: Write-Only

Bit	Description
7:0	**Palette Write Index.** The 8-bit index value programmed into this register chooses which of 256 standard color data positions within the palette are to be made accessible for being written via the Palette Data Register (DACDATA). The index value held in this register is automatically incremented when all three bytes of the color data position selected by the current index have been written. This register allows access to the palette even when running non-VGA display modes.

6.5.5 DACDATA—Palette Data Register

I/O (and Memory Offset) Address: 3C9h
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Palette Data.** This byte-wide data port provides read or write access to the three bytes of data of each color data position selected using the Palette Read Index Register (DACRX) or the Palette Write Index Register (DACWX). The three bytes in each color data position are read or written in three successive read or write operations. The first byte read or written specifies the intensity of the red component of the color specified in the selected color data position. The second byte is for the green component, and the third byte is for the blue component. When writing data to a color data position, all three bytes must be written before the hardware will actually update the three bytes of the selected color data position. When reading or writing to a color data position, ensure that neither the Palette Read Index Register (DACRX) nor the Palette Write Index Register (DACWX) are written to before all three bytes are read or written. A write to either of these two registers causes the circuitry that automatically cycles through providing access to the bytes for red, green and blue components to be reset such that the byte for the red component is the one that will be accessed by the next read or write operation via this register. This register allows access to the palette even when running non-VGA display modes. Writes to the palette can cause sparkle if not done during inactive video periods. This sparkle is caused by an attempt to write and read the same address on the same cycle. Anti-sparkle circuits will substitute the previous pixel value for the read output.

6.6 CRT Controller Register

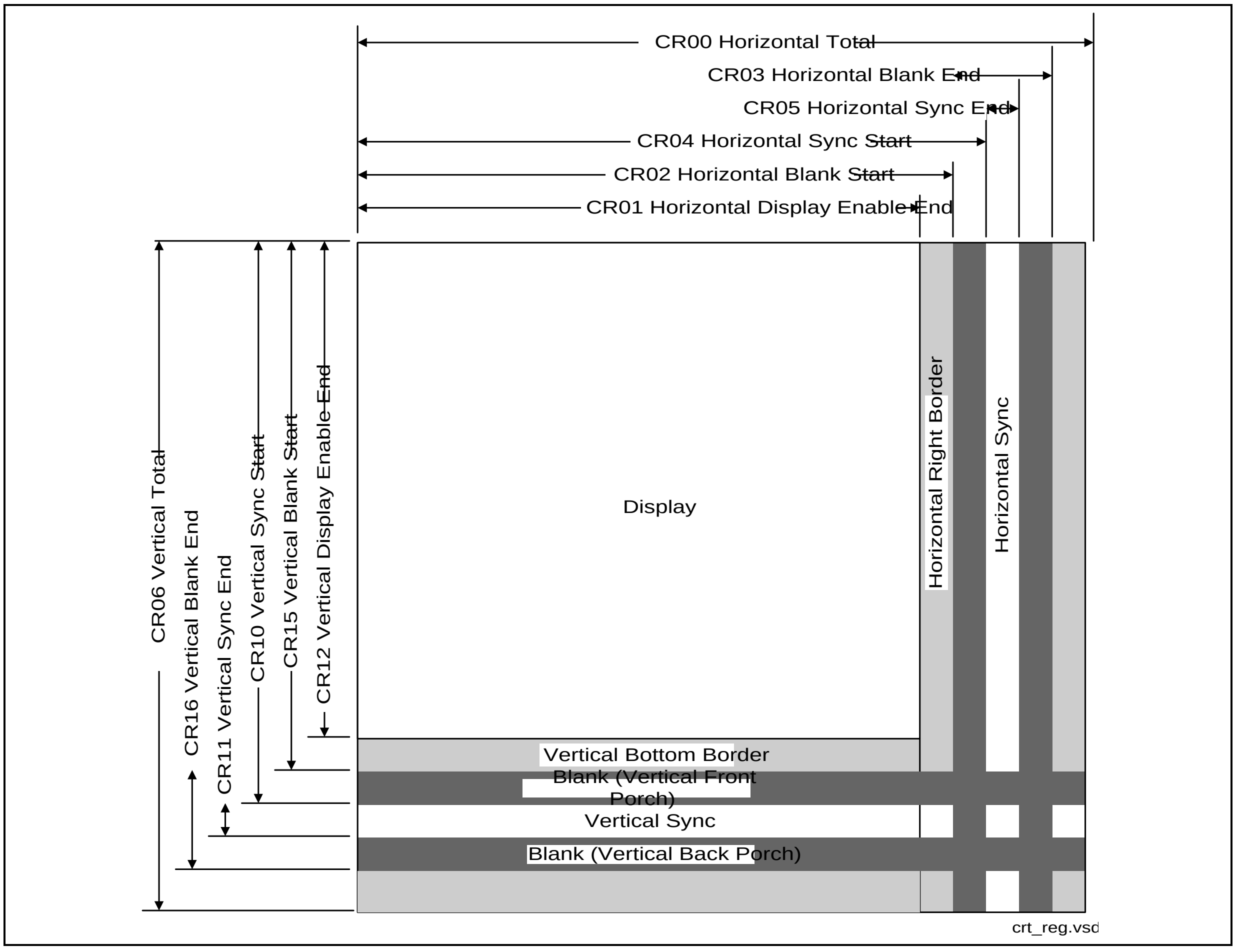
For native VGA modes, the CRT controller registers determine the display timing that is to be used. In centered VGA modes, these registers determine the size of the VGA image that is to be centered in the larger timing generator defined rectangle.

The CRT controller registers are accessed by writing the index of the desired register into the CRT Controller Index Register at I/O address 3B4h or 3D4h, depending on whether the graphics system is configured for MDA or CGA emulation. The desired register is then accessed through the data port for the CRT controller registers located at I/O address 3B5h or 3D5h, again depending upon the choice of MDA or CGA emulation as per MSR[0]. For memory mapped accesses, the Index register is at 3B4h (MDA mode) or 3D3h (CGA mode) and the data port is accessed at 3B5h (MDA mode) or 3D5h (CGA mode).

Notes:

1. **Group 0 Protection:** In the original IBM VGA, CR[0:7] could be made write-protected by CR11[7]. In BIOS code, this write protection is set following each mode change. Other protection groups have no current use, and would not be used going forward by the BIOS or by drivers. They are the result of an industry fad some years ago to attempt to write protect other groups of registers; however, all such schemes were chip specific. Only the IBM compatible write protection (Group 0 Protection) is supported.

The following figure shows display fields and dimensions and the particular CRxx register that provides the control.


CR00 Horizontal Total
CR03 Horizontal Blank End
CR05 Horizontal Sync End
CR04 Horizontal Sync Start
CR02 Horizontal Blank Start
CR01 Horizontal Display Enable End
CR06 Vertical Total
CR16 Vertical Blank End
CR11 Vertical Sync End
CR10 Vertical Sync Start
CR15 Vertical Blank Start
CR12 Vertical Display Enable End
Display
Horizontal Right Border
Horizontal Sync
Vertical Bottom Border
Blank (Vertical Front Porch)
Vertical Sync
Blank (Vertical Back Porch)
crt_reg.vsd

6.6.1 CRX—CRT Controller Index Register

I/O (and Memory Offset) Address: 3B4h/3D4h
Default: 0Uh (U=Undefined)
Attributes: Read/Write

Bit	Description
7	Reserved. Read as 0.
6:0	**CRT Controller Index.** These 7 bits are used to select any one of the CRT controller registers to be accessed via the data port at I/O location 3B5h or 3D5h, depending upon whether the graphics system is configured for MDA or CGA emulation. The data port memory address offsets are 3B5h/3D5h.

6.6.2 CR00—Horizontal Total Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=00h)
Default: 00h
Attributes: Read/Write (Group 0 Protection)

Bit	Description
7:0	**Horizontal Total.** This register is used to specify the total length of each scan line. This encompasses both the part of the scan line that is within the active display area and the part that is outside of it. Programming this register to a zero has the effect of stopping the fetching of display data. This field should be programmed with a value equal to the total number of character clocks within the entire length of a scan line, minus 5.

6.6.3 CR01—Horizontal Display Enable End Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=01h)
Default: Undefined
Attributes: Read/Write (Group 0 Protection)

Bit	Description
7:0	**Horizontal Display Enable End.** This register is used to specify the end of the part of the scan line that is within the active display area relative to its beginning. In other words, this is the horizontal width of the active display area. This field should be programmed with a value equal to the number of character clocks that occur within the horizontal active display area, minus 1. Horizontal display enable will go active at the beginning of each line during vertical active area, it will go inactive based on the programming of this register or the programming of the horizontal total (CR00) register. When this register value is programmed to a number that is larger than the total number of characters on a line, display enable will be active for all but the last character of the horizontal display line.

6.6.4 CR02—Horizontal Blanking Start Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=02h)
Default: Undefined
Attributes: Read/Write (Group 0 Protection)

Bit	Description
7:0	**Horizontal Blanking Start.** This register is used to specify the beginning of the horizontal blanking period relative to the beginning of the active display area of a scan line. Horizontal blanking should always be set to start no sooner than after the end of horizontal active. This field should be programmed with a value equal to the number of character clocks that occur on a scan line from the beginning of the active display area to the beginning of the horizontal blanking.

6.6.5 CR03—Horizontal Blanking End Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=03h)
Default: 1UUU UUUUb (U=Undefined)
Attributes: Read/Write (Group 0 Protection)

7	6 5	4 0
Reserved	Display Enable Skew Control	Horizontal Blanking End Bits 4:0

Bit	Description
7	**Reserved.** Values written to this bit are ignored, and to maintain consistency with the VGA standard, a value of 1 is returned when this bit is read. At one time, this bit was used to enable access to certain light pen registers. At that time, setting this bit to 0 provided this access, but setting this bit to 1 was necessary for normal operation.
6:5	**Display Enable Skew Control.** Defines the degree to which the start and end of the active display area are delayed along the length of a scan line to compensate for internal pipeline delays. These 2 bits describe the delay in terms of a number character clocks. **Bit [6:5]** **Amount of Delay** 00 no delay 01 delayed by 1 character clock 10 delayed by 2 character clocks 11 delayed by 3 character clocks
4:0	**Horizontal Blanking End Bits [4:0].** This field provides the 5 least significant bits of a 6-bit value that specifies the end of the blanking period relative to its beginning on a single scan line. Bit 7 of the Horizontal Sync End Register (CR05) supplies the most significant bit. This 6-bit value should be programmed to be equal to the least significant 6 bits of the result of adding the length of the blanking period in terms of character clocks to the value specified in the Horizontal Blanking Start Register (CR02). End of blanking should occur before horizontal total.

6.6.6 CR04—Horizontal Sync Start Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=04h)
Default: Undefined
Attributes: Read/Write (Group 0 Protection)

Bit	Description
7:0	**Horizontal Sync Start.** This register is used to specify the position of the beginning of the horizontal sync pulse relative to the start of the active display area on a scan line. This field should be set equal to the number of character clocks that occur from beginning of the active display area to the beginning of the horizontal sync pulse on a single scan line. Horizontal sync should always occur between the start and end of horizontal blank. The actual start of sync will also be affected by both the horizontal sync skew register field and whether it is a text or graphics mode.

6.6.7 CR05—Horizontal Sync End Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=05h)
Default: 00h
Attributes: Read/Write (Group 0 Protection)

7	6 5	4 0
Hor Blank End <5>	Horizontal Sync Delay	Horizontal Sync End

Bit	Description
7	**Horizontal Blanking End Bit 5.** This bit provides the most significant bit of a 6-bit value that specifies the end of the horizontal blanking period relative to its beginning. Bits [4:0] of Horizontal Blanking End Register (CR03) supply the 5 least significant bits. See CR03[4:0] for further details. This 6-bit value should be set to the least significant 6 bits of the result of adding the length of the blanking period in terms of character clocks to the value specified in the Horizontal Blanking Start Register (CR02).
6:5	**Horizontal Sync Delay.** This field defines the degree to which the start and end of the horizontal sync pulse are delayed to compensate for internal pipeline delays. This capability is supplied to implement VGA compatibility. This field describes the delay in terms of a number character clocks. **Bit [6:5]** **Amount of Delay** 00 no delay 01 delayed by 1 character clock 10 delayed by 2 character clocks 11 delayed by 3 character clocks
4:0	**Horizontal Sync End.** This field provides the 5 least significant bits of a 5-bit value that specifies the end of the horizontal sync pulse relative to its beginning. A value equal to the 5 least significant bits of the horizontal character counter value at which time the horizontal retrace signal becomes inactive (logical 0). Thus, this 5-bit value specifies the width of the horizontal sync pulse. To obtain a retrace signal of W, the following algorithm is used: Value of Horizontal Sync start Register (CR04) + width of horizontal retrace signal in character clock units = 5 bit result to be programmed in this field

6.6.8 CR06—Vertical Total Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=06h)
Default: 00h
Attributes: Read/Write (Group 0 Protection)

Bit	Description
7:0	**Vertical Total Bits [7:0].** This field provides the 8 least significant bits of either a 10-bit or 12-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area. In standard VGA modes, the vertical total is specified with a 10-bit value. The 8 least significant bits of this value are supplied by these 8 bits of this register, and the 2 most significant bits are supplied by bits 5 and 0 of the Overflow Register (CR07).

6.6.9 CR07—Overflow Register (Vertical)

I/O (and Memory Offset) Address:	3B5h/3D5h (index=07h)
Default:	UU0U UUU0b (U=Undefined)
Attributes:	Read/Write (Group 0 Protection on bits [7:5, 3:0])

7	6	5	4	3	2	1	0
Vert Sync Start <9>	Vert Disp Enable <9>	Vert Total <9>	Line Cmp<8>	Vert Blank Start<8>	Vert Sync Start<8>	Vert Display Enable <8>	Vert Total <8>

Bit	Description
7	**Vertical Sync Start Bit 9.** The vertical sync start is a 10-bit that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Sync Start Register (CR10), and the most and second-most significant bits are supplied by this bit and bit 2, respectively, of this register. This 10-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the start of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.
6	**Vertical Display Enable End Bit 9.** The vertical display enable end is a 10-bit that specifies the number of the last scan line within the active display area. In standard VGA modes, the vertical display enable end is specified with a 10-bit value. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Display Enable End Register (CR12), and the most and second-most significant bits are supplied by this bit and bit 1, respectively, of this register. This 10-bit value should be programmed to be equal to the number of the last scan line within in the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, minus 1.
5	**Vertical Total Bit 9.** The vertical total is a 10-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Total Register (CR06), and the most and second-most significant bits are supplied by this bit and bit 0, respectively, of this register. This 10-bit value should be programmed equal to the total number of scan lines, minus 2.
4	**Line Compare Bit 8.** This bit provides the second most significant bit of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 6 of the Maximum Scan Line Register (CR09) supplies the most significant bit, and bits 7-0 of the Line Compare Register (CR18) supply the 8 least significant bits. Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part. When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display what data exists in the frame buffer starting at the address specified in the two aforementioned start address registers, while the bottom part will display what data exists in the frame buffer starting at the first byte of the frame buffer.

Bit	Description
3	**Vertical Blanking Start Bit 8.** The vertical blanking start is a 10-bit that specifies the beginning of the vertical blanking period relative to the beginning of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Blanking Start Register (CR15), and the most and second-most significant bits are supplied by bit 5 of the Maximum Scan Line Register (CR09) and this bit of this register, respectively. This 10-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the beginning of the blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical blanking period begins.
2	**Vertical Sync Start Bit 8.** The vertical sync start is a 10-bit value that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Sync Start Register (CR10), and the most and second-most significant bits are supplied by bit 7 and this bit, respectively, of this register. This 10-bit value should be programmed to be equal to the number of scan lines from the beginning of the active display area to the start of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.
1	**Vertical Display Enable End Bit 8.** The vertical display enable end is a 10-bit value that specifies the number of the last scan line within the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Display Enable End Register (CR12), and the two most significant bits are supplied by bit 6 and this bit, respectively, of this register. This 10-bit or value should be programmed to be equal to the number of the last scan line within in the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, minus 1.
0	**Vertical Total Bit 8.** The vertical total is a 10-bit value that specifies the total number of scan lines. This includes the scan lines both inside and outside of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Total Register (CR06), and the most and second-most significant bits are supplied by bit 5 and this bit, respectively, of this register. This 10-bit value should be programmed to be equal to the total number of scan lines, minus 2.

6.6.10 CR08—Preset Row Scan Register

I/O (and Memory Offset) Address:	3B5h/3D5h (index=08h)
Default:	0UUU UUUUb (U=Undefined)
Attributes:	Read/Write

7	6 5	4 0
Reserved (0)	Byte Panning	Starting Row Scan Count

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>Reserved. Read as 0s.</td></tr>
<tr><td>6:5</td><td><b>Byte Panning.</b> This field holds a 2-bit value that selects number of bytes (up to 3) by which the image is shifted horizontally to the left on the screen. This function is available in both text and graphics modes.
In text modes with a 9-pixel wide character box, the image can be shifted up to 27 pixels to the left, in increments of 9 pixels. In text modes with an 8-pixel wide character box, and in all standard VGA graphics modes, the image can be shifted up to 24 pixels to the left, in increments of 8 pixels. When the Nine dot disable bit of the VGA control register is set, the pixel shift will be equivalent to the 8-dot mode.
The image can be shifted still further, in increments of individual pixels, through the use of bits [3:0] of the Horizontal Pixel Panning Register (AR13).
<b>Number of Pixels Shifted</b>
<table>
<tr><th>Bit [6:5]</th><th>9-Pixel Text</th><th>8-Pixel Text & Graphics</th></tr>
<tr><td>00</td><td>0</td><td>0</td></tr>
<tr><td>01</td><td>9</td><td>8</td></tr>
<tr><td>10</td><td>18</td><td>16</td></tr>
<tr><td>11</td><td>27</td><td>24</td></tr>
</table></td></tr>
<tr><td>4:0</td><td><b>Starting Row Scan Count.</b> This field specifies which horizontal line of pixels within the character boxes of the characters used on the top-most row of text on the display will be used as the top-most scan line. The horizontal lines of pixels of a character box are numbered from top to bottom, with the top-most line of pixels being number 0. If a horizontal line of the character boxes other than the top-most line is specified, then the horizontal lines of the character box above the specified line of the character box will not be displayed as part of the top-most row of text characters on the display. Normally, the value specified by these 5 bits should be 0, so that all of the horizontal lines of pixels within these character boxes will be displayed in the top-most row of text, ensuring that the characters in the top-most row of text do not look as though they have been cut off at the top.</td></tr>
</table>

6.6.11 CR09—Maximum Scan Line Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=09h)
Default: 00h
Attributes: Read/Write

7	6	5	4	0
Double Scanning	Line Cmp <9>	Vert Blank Start <9>	Starting Row Scan Count	

Bit	Description
7	**Double Scanning Enable.** 0 = Disable. When disabled, the clock to the row scan counter is equal to the horizontal scan rate. This is the normal setting for many of the standard VGA modes. 1 = Enable. When enabled, the clock to the row scan counter is divided by 2. This is normally used to allow CGA-compatible modes that have only 200 scan lines of active video data to be displayed as 400 scan lines (each scan line is displayed twice).
6	**Line Compare Bit 9.** This bit provides the most significant bit of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 4 of the Overflow Register (CR07) supplies the second most significant bit, and bits 7-0 of the Line Compare Register (CR18) supply the 8 least significant bits. Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part. When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display whatever data exists in the frame buffer starting at the address specified in the two aforementioned start address registers, while the bottom part will display whatever data exists in the frame buffer starting at the first byte of the frame buffer.
5	**Vertical Blanking Start Bit 9.** The vertical blanking start is a 10-bit value that specifies the beginning of the vertical blanking period relative to the beginning of the active display area. The 8 least significant bits of this value are supplied by bits [7:0] of the Vertical Blanking Start Register (CR15), and the most and second-most significant bits are supplied by this bit and bit 3 of the Overflow Register (CR07), respectively. This 10-bit value should be programmed to be equal to the number of scan line from the beginning of the active display area to the beginning of the blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical blanking period begins.
4:0	**Starting Row Scan Count.** This field provides all 5 bits of a 5-bit value that specifies the number of scan lines in a horizontal row of text. This value should be programmed to be equal to the number of scan lines in a horizontal row of text, minus 1.

6.6.12 CR0A—Text Cursor Start Register

I/O (and Memory Offset) Address:	3B5h/3D5h (index=0Ah)
Default:	00UU UUUUb (U=Undefined)
Attributes:	Read/Write

7 6	5	4 0
Reserved (00)	Text Cursor Off	Text Cursor Start

Bit	Description
7:6	Reserved. Read as 0.
5	**Text Cursor Off.** This cursor is the text cursor that is part of the VGA standard, and should not be confused with the hardware cursor and popup which are intended to be used in extended graphics modes only. This text cursor exists only in text modes, and thus, this register is entirely ignored in graphics modes. 0 = Enables the text cursor. 1 = Disables the text cursor.
4:0	**Text Cursor Start.** This field specifies which horizontal line of pixels in a character box is to be used to display the first horizontal line of the cursor in text mode. The horizontal lines of pixels in a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the first horizontal line of pixels on which the cursor is to be shown.

6.6.13 CR0B—Text Cursor End Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=0Bh)
Default: 0UUU UUUUb (U=Undefined)
Attributes: Read/Write

7	6 5	4 0
Reserved (0)	Text Cursor Skew	Text Cursor End

Bit	Description
7	Reserved. Read as 0.
6:5	**Text Cursor Skew.** This field specifies the degree to which the start and end of each horizontal line of pixels making up the cursor is delayed to compensate for internal pipeline delays. These 2 bits describe the delay in terms of a number character clocks. **Bit [6:5]** **Amount of Delay** 00 No delay 01 Delayed by 1 character clock 10 Delayed by 2 character clocks 11 Delayed by 3 character clocks
4:0	**Text Cursor End.** This field specifies which horizontal line of pixels in a character box is to be used to display the last horizontal line of the cursor in text mode. The horizontal lines of pixels in a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the last horizontal line of pixels on which the cursor is to be shown.

6.6.14 CR0C—Start Address High Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=0Ch)
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Start Address Bits [15:8].** This register provides either bits 15 through 8 of a 16-bit value that specifies the memory address offset from the beginning of the frame buffer at which the data to be shown in the active display area begins. (default is 0) In standard VGA modes, the start address is specified with a 16-bit value. The eight bits of this register provide the eight most significant bits of this value, while the eight bits of the Start Address Low Register (CR0D) provide the eight least significant bits.

6.6.15 CR0D—Start Address Low Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=0Dh)
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Start Address Bits [7:0].** This register provides either bits 7 through 0 of a 16-bit value that specifies the memory address offset from the beginning of the frame buffer at which the data to be shown in the active display area begins. (default is 0) In standard VGA modes the start address is specified with a 16-bit value. The eight bits of the Start Address High Register (CR0C) provide the eight most significant bits of this value, while the eight bits of this register provide the eight least significant bits.

6.6.16 CR0E—Text Cursor Location High Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=0Eh)
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Text Cursor Location Bits [15:8].** This field provides the 8 most significant bits of a 16-bit value that specifies the address offset from the beginning of the frame buffer at which the text cursor is located. Bit 7:0 of the Text Cursor Location Low Register (CR0F) provide the 8 least significant bits.

6.6.17 CR0F—Text Cursor Location Low Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=0Fh)
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Text Cursor Location Bits [7:0].** This field provides the 8 least significant bits of a 16-bit value that specifies the address offset from the beginning of the frame buffer at which the text cursor is located. Bits 7:0 of the Text Cursor Location High Register (CR0D) provide the 8 most significant bits.

6.6.18 CR10—Vertical Sync Start Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=10h)
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Vertical Sync Start Bits [7:0].** This register provides the 8 least significant bits of a 10-bit that specifies the beginning of the vertical sync pulse relative to the beginning of the active display area of a screen. In standard VGA modes, this value is described in 10 bits with bits [7,2] of the Overflow Register (CR07) supplying the 2 most significant bits. This 10-bit value should equal the vertical sync start in terms of the number of scan lines from the beginning of the active display area to the beginning of the vertical sync pulse. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which the vertical sync pulse begins.

6.6.19 CR11—Vertical Sync End Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=11h)
Default: 0U00 UUUUb (U=Undefined)
Attributes: Read/Write

7	6	5	4	3 0
Protect Regs 0:7	Reserved	Vert Int Enable	Vert Int Clear	Vertical Sync End

Bit	Description
7	**Protect Registers [0:7].** Note that the ability to write to Bit 4 of the Overflow Register (CR07) is not affected by this bit (i.e., bit 4 of the Overflow Register is always writeable). 0 = Enable writes to registers CR[00:07]. (default) 1 = Disable writes to registers CR[00:07].
6	Reserved. In the VGA standard, this bit was used to switch between 3 and 5 frame buffer refresh cycles during the time required to draw each horizontal line.
5	**Vertical Interrupt Enable.** This bit is reserved for compatibility only. While this bit may be written or read, its value will have no effect. Note that the VGA does not provide an interrupt signal which would be connected to an input of the system's interrupt controller. Bit 7 of Input Status Register 0 (ST00) originally indicated the status of the vertical retrace interrupt. 0 = Enable the generation of an interrupt at the beginning of each vertical retrace period. 1 = Disable the generation of an interrupt at the beginning of each vertical retrace period.
4	**Vertical Interrupt Clear.** This is reserved for compatibility only. Note that the VGA does not provide an interrupt signal which would be connected to an input of the system's interrupt controller. 0 = Setting this bit to 0 clears a pending vertical retrace interrupt. This bit must be set back to 1 to enable the generation of another vertical retrace interrupt.
3:0	**Vertical Sync End.** This 4-bit field provides a 4-bit value that specifies the end of the vertical sync pulse relative to its beginning. This 4-bit value should be set to the least significant 4 bits of the result of adding the length of the vertical sync pulse in terms of the number of scan lines that occur within the length of the vertical sync pulse to the value that specifies the beginning of the vertical sync pulse (see the description of the Vertical Sync Start Register for more details).

6.6.20 CR12—Vertical Display Enable End Register

I/O (and Memory Offset) Address:	3B5h/3D5h (index=12h)
Default:	Undefined
Attributes:	Read/Write

Bit	Description
7:0	**Vertical Display Enable End Bits [7:0].** This register provides the 8 least significant bits of a 10-bit value that specifies the number of the last scan line within the active display area. In standard VGA modes, this value is described in 10 bits with bits [6,1] of the Overflow Register (CR07) supplying the 2 most significant bits. This 10-bit value should be programmed to be equal to the number of the last scan line within in the active display area. Since the active display area always starts on the 0th scan line, this number should be equal to the total number of scan lines within the active display area, minus 1.

6.6.21 CR13—Offset Register

I/O (and Memory Offset) Address:	3B5h/3D5h (index=13h)
Default:	Undefined
Attributes:	Read/Write

Bit	Description
7:0	**Offset Bits [7:0].** This register provides either all 8 bits of an 8-bit value that specifies the number of words or DWords of frame buffer memory occupied by each horizontal row of characters. Whether this value is interpreted as the number of words or DWords is determined by the settings of the bits in the Clocking Mode Register (SR01). In standard VGA modes, the offset is described with an 8-bit value, all the bits of which are provided by this register. This 8-bit value should be programmed to be equal to either the number of words or DWords (depending on the setting of the bits in the Clocking Mode Register, SR01) of frame buffer memory that is occupied by each horizontal row of characters.

6.6.22 CR14—Underline Location Register

I/O (and Memory Offset) Address:	3B5h/3D5h (index=14h)
Default:	0UUU UUUUb (U=Undefined)
Attributes:	Read/Write

7	6	5	4 0
Reserved (0)	Dword Mode	Count By 4	Underline Location

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>Reserved. Read as 0.</td></tr>
<tr><td>6</td><td><b>DWord Mode.</b>
0 = Frame buffer addresses are interpreted by the frame buffer address decoder as being either byte addresses or word addresses, depending on the setting of bit 6 of the CRT Mode Control Register (CR17).
1 = Frame buffer addresses are interpreted by the frame buffer address decoder as being DWord addresses, regardless of the setting of bit 6 of the CRT Mode Control Register (CR17).
Note that this bit is used in conjunction with bits 6 and 5 of the CRT Mode Control Register (CR17) to select how frame buffer addresses from the CPU are interpreted by the frame buffer address decoder as shown below:
<table>
<tr><th>CR14[6]</th><th>CR17[6]</th><th>Addressing Mode</th></tr>
<tr><td>0</td><td>0</td><td>Word Mode</td></tr>
<tr><td>0</td><td>1</td><td>Byte Mode</td></tr>
<tr><td>1</td><td>0</td><td>DWord Mode</td></tr>
<tr><td>1</td><td>1</td><td>DWord Mode</td></tr>
</table></td></tr>
<tr><td>5</td><td><b>Count By 4.</b>
0 = The memory address counter is incremented either every character clock or every other character clock, depending upon the setting of bit 3 of the CRT Mode Control Register.
1 = The memory address counter is incremented either every 4 character clocks or every 2 character clocks, depending upon the setting of bit 3 of the CRT Mode Control Register. . This is used in mode x13 to allow for using all four planes.
Note that this bit is used in conjunction with bit 3 of the CRT Mode Control Register (CR17) to select the number of character clocks are required to cause the memory address counter to be incremented as shown, below:
<table>
<tr><th>CR14[5]</th><th>CR17[3]</th><th>Address Incrementing Interval</th></tr>
<tr><td>0</td><td>0</td><td>every character clock</td></tr>
<tr><td>0</td><td>1</td><td>every 2 character clocks</td></tr>
<tr><td>1</td><td>0</td><td>every 4 character clocks</td></tr>
<tr><td>1</td><td>1</td><td>every 2 character clocks</td></tr>
</table></td></tr>
<tr><td>4:0</td><td><b>Underline Location.</b> This field specifies which horizontal line of pixels in a character box is to be used to display a character underline in text mode. The horizontal lines of pixels in a character box are numbered from top to bottom, with the top-most line being number 0. The value specified by these 5 bits should be the number of the horizontal line on which the character underline mark is to be shown.</td></tr>
</table>

6.6.23 CR15—Vertical Blanking Start Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=15h)
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Vertical Blanking Start Bits [7:0].** This register provides the 8 least significant bits of a 10-bit value that specifies the beginning of the vertical blanking period relative to the beginning of the active display area of the screen. In standard VGA modes, the vertical blanking start is specified with a 10-bit value. The most and second-most significant bits of this value are supplied by bit 5 of the Maximum Scan Line Register (CR09) and bit 3 of the Overflow Register (CR07), respectively. This 10-bit value should be programmed to be equal the number of scan lines from the beginning of the active display area to the beginning of the vertical blanking period. Since the active display area always starts on the 0th scan line, this number should be equal to the number of the scan line on which vertical blanking begins.

6.6.24 CR16—Vertical Blanking End Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=16h)
Default: Undefined
Attributes: Read/Write

This register provides a 8-bit value that specifies the end of the vertical blanking period relative to its beginning.

Bit	Description
7:0	**Vertical Blanking End Bits [7:0].** This 8-bit value should be set equal to the least significant 8 bits of the result of adding the length of the vertical blanking period in terms of the number of scan lines that occur within the length of the vertical blanking period to the value that specifies the beginning of the vertical blanking period (see the description of the Vertical Blanking Start Register for details).

6.6.25 CR17—CRT Mode Control

I/O (and Memory Offset) Address: 3B5h/3D5h (index=17h)
Default: 0UU0 UUUUb (U=Undefined)
Attributes: Read/Write

7	6	5	4	3	2	1	0
CRT Ctrl Reset	Word or Byte Mode	Address Wrap	Reserved (0)	Count By 2	Horizontal Retrace Select	Select Row Scan Cntr	Compat Mode Support

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td><b>CRT Controller Reset.</b> This bit has no effect except in native VGA modes (non-centered).
0 = Forces horizontal and vertical sync signals to be inactive. No other registers or outputs are affected.
1 = Permits normal operation.</td></tr>
<tr><td>6</td><td><b>Word Mode or Byte Mode.</b>
0 = The memory address counter’s output bits are shifted by 1 bit position before being passed on to the frame buffer address decoder such that they are made into word-aligned addresses when bit 6 of the Underline Location Register (CR17) is set to 0.
1 = The memory address counter’s output bits remain unshifted before being passed on to the frame buffer address decoder such that they remain byte-aligned addresses when bit 6 of the Underline Location Register (CR17) is set to 0.
Note that this bit is used in conjunction with bits 6 and 5 of the CRT Mode Control Register (CR17) to control how frame buffer addresses from the memory address counter are interpreted by the frame buffer address decoder as shown below:
<table>
<tr><th>CR14[6]</th><th>CR17[6]</th><th>Addressing Mode</th></tr>
<tr><td>0</td><td>0</td><td>Word Mode—Addresses from the memory address counter are shifted once to become word-aligned</td></tr>
<tr><td>0</td><td>1</td><td>Byte Mode—Addresses from the memory address counter are not shifted</td></tr>
<tr><td>1</td><td>0</td><td>DWord Mode—Addresses from the memory address counter are shifted twice to become DWord-aligned</td></tr>
<tr><td>1</td><td>1</td><td>DWord Mode—Addresses from the memory address counter are shifted twice to become DWord-aligned</td></tr>
</table></td></tr>
<tr><td>5</td><td><b>Address Wrap.</b> Note that this bit is only effective when word mode is made active by setting bit 6 in both the Underline Location Register and this register to 0.
0 = Wrap frame buffer address at 16 KB. This is used in CGA-compatible modes.
1 = No wrapping of frame buffer addresses.</td></tr>
<tr><td>4</td><td>Reserved. Read as 0.</td></tr>
</table>

Bit	Description
3	**Count By 2.** This bit is used in conjunction with bit 5 of the Underline Location Register (CR14) to select the number of character clocks are required to cause the memory address counter to be incremented. 0 = The memory address counter is incremented either every character clock or every 4 character clocks, depending upon the setting of bit 5 of the Underline Location Register. 1 = The memory address counter is incremented either every other clock. **CR14[5] CR17[3] Address Incrementing Interval** 0 0 every character clock 0 1 every 2 character clocks 1 0 every 4 character clocks 1 1 every 2 character clocks
2	**Horizontal Retrace Select.** This bit provides a way of effectively doubling the vertical resolution by allowing the vertical timing counter to be clocked by the horizontal retrace clock divided by 2 (usually, it would be undivided). 0 = The vertical timing counter is clocked by the horizontal retrace clock. 1 = The vertical timing counter is clocked by the horizontal retrace clock divided by 2.
1	**Select Row Scan Counter.** 0 = A substitution takes place, where bit 14 of the 16-bit memory address generated of the memory address counter (after the stage at which these 16 bits may have already been shifted to accommodate word or DWord addressing) is replaced with bit 1 of the row scan counter at a stage just before this address is presented to the frame buffer address decoder. 1 = No substitution takes place. See following tables.
0	**Compatibility Mode Support.** 0 = A substitution takes place, where bit 13 of the 16-bit memory address generated of the memory address counter (after the stage at which these 16 bits may have already been shifted to accommodate word or DWord addressing) is replaced with bit 0 of the row scan counter at a stage just before this address is presented to the frame buffer address decoder. 1 = No substitution takes place. See following tables.

The following tables show the possible ways in which the address bits from the memory address counter can be shifted and/or reorganized before being presented to the frame buffer address decoder. First, the address bits generated by the memory address counter are reorganized, if need be, to accommodate byte, word or DWord modes. The resulting reorganized outputs (MAOut15-MAOut0) from the memory address counter may also be further manipulated with the substitution of bits from the row scan counter (RSOut1 and RSOut0) before finally being presented to the input bits of the frame buffer address decoder (FBIn15-FBIn0).

Table 6-2. Memory Address Counter Address Bits [15:0]

	Byte Mode CR14 bit 6=0 CR17 bit 6=1 CR17 bit 5=X	Word Mode CR14 bit 6=0 CR17 bit 6=0 CR17 bit 5=1	Word Mode CR14 bit 6=0 CR17 bit 6=0 CR17 bit 5=0	DWord Mode CR14 bit 6=1 CR17 bit 6=X CR17 bit 5=X
MAOut0	0	15	13	12
MAOut1	1	0	0	13
MAOut2	2	1	1	0
MAOut3	3	2	2	1
MAOut4	4	3	3	2
MAOut5	5	4	4	3
MAOut6	6	5	5	4
MAOut7	7	6	6	5
MAOut8	8	7	7	6
MAOut9	9	8	8	7
MAOut10	10	9	9	8
MAOut11	11	10	10	9
MAOut12	12	11	11	10
MAOut13	13	12	12	11
MAOut14	14	13	13	12
MAOut15	15	14	14	13

X = Don’t Care

Table 6-3. Frame Buffer Address Decoder

	CR17 bit 1=1	CR17 bit 1=1	CR17 bit 1=0	CR17 bit 1=0
	CR17 bit 0=1	CR17 bit 0=0	CR17 bit 0=1	CR17 bit 0=0
FBIn0	MAOut0	MAOut0	MAOut0	MAOut0
FBIn1	MAOut1	MAOut1	MAOut1	MAOut1
FBIn2	MAOut2	MAOut2	MAOut2	MAOut2
FBIn3	MAOut3	MAOut3	MAOut3	MAOut3
FBIn4	MAOut4	MAOut4	MAOut4	MAOut4
FBIn5	MAOut5	MAOut5	MAOut5	MAOut5
FBIn6	MAOut6	MAOut6	MAOut6	MAOut6
FBIn7	MAOut7	MAOut7	MAOut7	MAOut7
FBIn8	MAOut8	MAOut8	MAOut8	MAOut8
FBIn9	MAOut9	MAOut9	MAOut9	MAOut9
FBIn10	MAOut10	MAOut10	MAOut10	MAOut10
FBIn11	MAOut11	MAOut11	MAOut11	MAOut11
FBIn12	MAOut12	MAOut12	MAOut12	MAOut12
FBIn13	MAOut13	MAOut13	RSOut0	RSOut0
FBIn14	MAOut14	RSOut1	MAOut14	RSOut1
FBIn15	MAOut15	MAOut15	MAOut15	MAOut15

6.6.26 CR18—Line Compare Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=18h)
Default: Undefined
Attributes: Read/Write

Bit	Description
7:0	**Line Compare Bits [7:0].** This register provides the 8 least significant bits of a 10-bit value that specifies the scan line at which the memory address counter restarts at the value of 0. Bit 6 of the Maximum Scan Line Register (CR09) supplies the most significant bit, and bit 4 of the Overflow Register (CR07) supplies the second most significant bit. Normally, this 10-bit value is set to specify a scan line after the last scan line of the active display area. When this 10-bit value is set to specify a scan line within the active display area, it causes that scan line and all subsequent scan lines in the active display area to display video data starting at the very first byte of the frame buffer. The result is what appears to be a screen split into a top and bottom part, with the image in the top part being repeated in the bottom part. (This register is only used in split screening modes, and this is not a problem because split screening is not actually used for extended modes. As a result, there is no benefit to extending the existing overflow bits for higher resolutions.) When used in cooperation with the Start Address High Register (CR0C) and the Start Address Low Register (CR0D), it is possible to create a split display, as described earlier, but with the top and bottom parts displaying different data. The top part will display whatever data exists in the frame buffer starting at the address specified in the two aforementioned start address registers, while the bottom part will display whatever data exists in the frame buffer starting at the first byte of the frame buffer.

6.6.27 CR22—Memory Read Latch Data Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=22h)
Default: 00h
Attributes: Read-Only

Bit	Description
7:0	**Memory Read Latch Data.** This field provides the value currently stored in 1 of the four memory read latches. Bits 1 and 0 of the Read Map Select Register (GR04) select which of the four memory read latches may be read via this register.

6.6.28 CR24— Test Register for Toggle State of Attribute Controller Register

I/O (and Memory Offset) Address: 3B5h/3D5h (index=24h)
Default: 00h
Attributes: Read-Only

7	6	0
Toggle Status	Reserved (0000000)	

Bit	Description
7	**Toggle Status.** Indicates where the last write to attribute register was to: 0 = index port 1 = data port
6:0	Reserved. Read as 0.

7 AVC Bit-Serial Decoder

7.1 Introduction

A standalone Bit-Serial Decoding (BSD) module designed specifically for supporting real-time dual streams CABAC/CAVLD decoding, according to AVC standard, with the expected aggregated input data rate of up to 50 mbps..

Support an AVC Main and High Profiles, up to 4.1 Level

Support a second AVC Standard Definition stream of up to 10 mbps

7.1.1 Terminologies

AVC	Advanced Video Codec, namely H.264 or MPEG4 Part 10
AVC_FE	AVC Front End Unit (AI + AC + AM)
AI	AVC Input Interface Unit
AM	AVC Memory Interface Unit
AC	AVC Bitstream Parser Core Unit
BCS	AVC_FE Command Streamer
BSD	Bit-Serial Decoding Unit
CABAC	Context Adaptive Binary Arithmetic Coder
CAVLD	Context Adaptive Variable Length Decoder
DPB	Display Buffer Store (holding all the reference frames and current decoded picture)
I_PCM	Intra-Frame Pulse Code Modulation, Coding Mode
IT	Inverse Transform Module
MBAFF	Macro-block Adaptive Frame/Field Coding Mode
MPR	Motion Vector Prediction Module
PPS	Picture Parameter Set (AVC Syntax)
SPS	Sequence Parameter Set (AVC Syntax)

7.1.2 Design Assumptions

MbaffFrameFlag can't have different values in different slices of the same picture

512 bits memory access interface, cache line size

No interruptability within a slice decoding. In addition, according to the AVC specification, there is no dependency across Slice boundary with respect to bit stream decoding. Hence, context switching mechanism is much simplified. Currently, a video stream context is consisted of mainly the starting MB number of a Slice in that stream. Context switching mechanism is needed for handling concurrent multiple stream decoding and error handling.

There is no pre-emption for the BCS pipeline (CPU is not able to interrupt the BCS-BSD pipe), hence every command buffer is required to contain all the states setup (preamble), and therefore no need to save and restore context.

If different H.264 applications are running, each will have its own command buffer (responsible for decoding at least one Slice). The driver is required to collect all the command buffers of a picture of a video sequence. Each command buffer will run to completion serially.

If an H.264 application is playbacking multiple video sequences, each sequence has its own command buffer list.

The switching across different video streams of different applications can only be done on a picture boundary.

In BLC, we will continue to use the batch buffer. But for Cantiga, we will implement the run list mechanism (limited to a single element run list) for multi-context scheduling, including ring context and PPGTT.

A command buffer can contain multiple Slices to be decoded, but will not cross the picture boundary. But there is no driver interruption in between Slices. A command buffer must run to its completion. The worst case latency is one picture time, since a Slice can be as big as a picture.

7.1.3 Bit-Serial Decoding (BSD) Unit

A standalone H/W unit uses for performing the AVC CABAC/CAVLD decoding function. It is operating concurrently with the GENX GPE, with its own Command Streamer. The communications and synchronization between BSD and GENX GPE is primarily through memory. The Host will parse all header information from an AVC bit-stream, and only send down the raw Slice Data in memory to the BSD. The decoded data are written back to memory for the GENX GPE to consume.

Starting code detection is done in the Host driver, only the Slice Data layer is passed down to the HW for processing. The H/W is also responsible for

The BSD gets its commands, states and parameters from the CS unit Priori to BLC, CS would get the indirect data for the media front-end engine. But for BLC, AI unit now fetches/pre-fetches the indirect slice payload through AM/CS/CI units and takes care of start/end of slice before sending the indirect data (slice payload) to AC unit. AI unit also needs to take care of emulation prevention byte for AC unit.

7.1.4 Intra Prediction for I Slices

7.1.4.1 4x4 Spatial Luma, 16x16 Spatial Luma, 8x8 Spatial Chroma

Intra_16x16 uses one intra prediction scheme for the whole macroblock. Pixels may be filled from surrounding macroblocks at the left and the upper edge using one of four possible prediction modes. Intra prediction is also performed for the chroma planes using the same range of prediction modes. However, different modes may be selected for luma and chroma.

Intra_4x4 subdivides the macroblock into 16 subblocks and assigns one of nine prediction modes to each of these 4x4 blocks. The prediction modes offered in Intra_4x4 blocks support gradients or other smooth structures that run in one of eight distinct directions. One additional mode fills a whole subblock with a single value and is used if no other mode fits the actual pixel data inside a block. Intra chroma prediction is performed in an identical way to the one used in Intra_16x16 prediction. Thus, the chroma predictions are not split into subblocks as it is done for luma.

7.1.4.2 I_PCM

Transmit picture samples (Luma and Chroma) directly without any prediction, transformation and compression. It sets the upper limit (maximum number of bits) for each coded MB. The I_PCM mode is a type of "fail safe" mode that bounds the size of a macroblock in the compressed video bitstream. In I_PCM, coefficient data is replaced by actual 8-bit pixel sample values. Hence, instead of 16-bit per coefficient data, a MB in I_PCM mode will contain packed 8-bit values of Luma and Chroma, and each MB will be exactly 384 bytes (luma + chroma) in size.

According to the AVC Spec., I_PCM mode is enlisted amongst Intra-Prediction modes. And internal parameter settings (listed below) in I_PCM mode are similar to those of Intra16x16 mode.

For a macroblock coded with mb_type equal to I_PCM, the Intra macroblock prediction mode shall be inferred.

Hence, there will be no separation of DC terms from the AC terms as in inter-prediction coding

So, all the Dcblock flags should be either ignored by the H/W in the I_PCM mode (as in intra-prediction modes) or set to 0 (to indicate their absence)

For macroblocks in I_PCM mode, there is no coded_block_pattern syntax element present in the bitstream to derive the variables CodedBlockPatternLuma and CodedBlockPatternChroma.

When in I_PCM mode, the parameters - Intra16x16PredMode, CodedBlockPatternChroma and CodeBlockPattern Luma are don’t care.

Hence, in I_PCM mode, we treat each MB as non-skipped and all 16x16 Luma coefficients and 8x8 Chroma (Cr and Cb) coefficients are present and no separation of DC terms. So, the cbp bits for all Luma and Cr/Cb 8x8 blocks should be set to indicate their presence. In monochrome mode, Cr/Cb should be set to absent. And the separated DC block flags should be 0 to indicate their absence.

When chroma_format_idc is equal to 0 (i.e. monochrome), CodedBlockPatternChroma shall be equal to 0.

7.2 H/W Asynchronous Reset

Global H/W Reset is signalled during system power-up to reset the entire BSD pipeline and to initialize it to a known state. It is also used in situation of errors to clear the pipeline. When the Command Stream Parser is stalled, pipeline reset implemented as CS command would not get through. Host S/W WatchDog timers may be implemented to detect stalled conditions. After H/W reset, the ring buffer will be lost, and driver re-programming is required to continue. The reset immmediately takes effect regardless the current operation.

7.3 H/W Pipeline Flush

A flush is implicitly carried out at the end of decoding a Slice. An explicit flush can also be issued to the BCS at the picture boundary (at the end of processing a picture). BCS unit will then perform the flush by monitoring the "done" signals from AI, AC and AM units, and issue a subsequent "store Dword".

7.4 MMIO Interface

MMIO is involved in interrupt generation during error handling. MMIO address 4400h is defined for the BSD registers, and 4000h for the BCS MMIO. These are read and write registers.

Currently, there are two set of MMIO registers

AVC BSD Error Register (16-bit)

AVC BSD Error Count Register (12-bit each) – do not wrap around when maximum count has reached.

AVC BSD Error Register : MMIO Address : 4400h

The only MMIO write operation is to reset each individual bit of this register to a 0 value, after the error information has been read and/or processed. The driver may choose to read this register in between pictures and video sequence and upon video stream switching. This register is set to 0 at powerup.

16-bit Read/Write Register (32-bit aligned)

Bit	Description
15:4	Reserved. MBZ
3	BSD Premature Completion Error Status Flag When a BSD Premature Completion error has occurred and the BSDPrematureComplete Error Handling bit in the inline data of the AVC_BSD_OBJECT command is set, this error status flag is set until being cleared by a subsequent MMIO write to this register.
2	MPR Error Status Flag When a MPR error has occurred and the MPR Error Handling bit in the inline data of the AVC_BSD_OBJECT command is set, this error status flag is set until being cleared by a subsequent MMIO write to this register.

Bit	Description
1	VLD Error Status Flag When a VLD error has occurred and the VLD Error Handling bit in the inline data of the AVC_BSD_OBJECT command is set, this error status flag is set until being cleared by a subsequent MMIO write to this register.
0	BSD Error Status Flag When a BSD error has occurred and the BSD Error Handling bit in the inline data of the AVC_BSD_OBJECT command is set, this error status flag is set until being cleared by a subsequent MMIO write to this register.

AVC BSD Error Count Register : MMIO Address : 4404h

The only MMIO write operation is to reset this register to a 0 value. The driver may choose to read this register in between pictures and video sequence and upon video stream switching. This register is set to 0 at powerup.

16-bit Read/Write Register (32-bit aligned)

Bit	Description
15:12	Reserved. MBZ
11:0	BSD Error Count Increment by 1 when any of the recognized errors (BSD, VLD, MPR and PrematureCompletion) has occurred. Do not wrap around when the maximum count has reached.

4408 for Sync Reset (check with Victor)

7.5 Programming the BSD Unit

7.5.1 Example Command Sequence for Decoding a Single Video Stream

BSD Unit uses a simple programming model that does not support pipelined state changes. All state information required to decode a Slice are always sent along (and prior to) with the compressed video data. This eliminates the need to reset the internal state registers for decoding a new Slice; they will be over-written anyway.

The basic steps in programming the BSD Unit are listed below. There is no explicit initialization step for the BSD engine, the H/W unit will be properly initialized itself at power-up and at reset. Some of the steps are optional and some are repetitive. The recommended order should be followed strictly. Some usage restrictions are highlighted for illustration purpose. For details, reader should refer to the respective chapters for these commands.

Step 1: Issue the AVC_IND_OBJ_BASE_ADDR command for system configuration

This command is mandatory for this step (i.e. at least one).

Multiple such commands in this step are allowed. The last one overwrites previous ones.

This command must precede any other state commands below.

The fields AVC Indirect Object Base Address and AVC Indirect Object Access Upper Bound are used to control indirect object load.

Step 2: Issue AVC_BSD_State commands to program the AVC decoding pipeline states

These commands are mandatory, since Slice Data decoding cannot rely on states set by the previous Slice.

Multiple such commands in this step are allowed. When the same command is issued multiple times, the last one overwrites previous ones.

There are 4 state commands to be sent in a strict order, Except the IMG_STATE that should only be changed on a per picture basis, the others are re-sent for decoding a new Slice whenever there is a change :

AVC_BSD_IMG_STATE – sent in for every new picture (first slice of each picture)

AVC_BSD_QM_STATE – sent in for every Slice to be decoded

AVC_BSD_SLICE_STATE – sent in for every Slice to be decoded

AVC_BSD_BUF_BASE_STATE – sent in every Slice to be decoded

Step 3: Issue the media/primitive AVC_BSD_OBJECT command

Multiple AVC_BSD_OBJECT commands can be issued to continue processing BSD media primitives, only if all the state information is unchanged.

An implicit pipeline flush is executed at the end of decoding a Slice. The BSD Command Stream Parser will be halted until it has received a complete signal from the BSD unit.

Step 4: Repeat Step 2 and 3 as many times as needed for processing an entire picture

Step 5: Send a phantom slice at the end of each picture

To make sure that the automatic error concealment in the H/W is triggered to the end of the picture when needed.

Step 6 : BCS flush sent in at the end of each picture, and perform a subsequent store Dword command

Step 7: Repeat Step 2 through 5 as many times as needed for processing an entire video sequence

7.6 AVC_BSD Commands

7.6.1 BSD_IND_OBJ_BASE_ADDR Command

The BSD_IND_OBJ_BASE_ADDR command sets the memory base address pointers for the subsequent Indirect Data Start Address specified in the AVC_BSD_OBJECT commands. This command is shared by VC1 BSD pipe.

While the use of this base addresses is unconditional, the indirection can be effectively disabled by setting the base addresses to zero.

The Command Streamer (BCS) will perform the memory access bound check automatically using the AVC Indirect Object Access Upper Bound specification. If any access is at or beyond this bound, zero value is returned. The request to memory still being sent, but the BSD H/W will detect and perform the zeroing. If the Upper Bound is turned off, the beyond bound request will return whatever on the bus (invalid data).

Notation	Definition
PhysicalAddress[n:m]	Corresponding bits of a physical graphics memory byte address (not mapped by a GTT)
GraphicsAddress[n:m]	Corresponding bits of an absolute, virtual graphics memory byte address (mapped by a GTT)

Dword	Bit	Description
0	31:29	Command Type = PARALLEL_VIDEO_PIPE = 3h
	28:16	AVC Command Opcode = BSD_IND_OBJ_BASE_ADDR Pipeline[28:27] = BSD = 2h; Opcode[26:24] = AVC = 4h; Sub Opcode[23:16] = 4h
	15:0	DWord Length (Excludes DWords 0, 1) = 0001h
1	31:12	AVC Indirect Object Base Address. Specifies the 4K-byte aligned memory base address for indirect object load in AVC_BSD_OBJECT command. Format = GraphicsAddress[31:12]
	11:0	Reserved : MBZ
2	31:12	AVC Indirect Object Access Upper Bound. This field specifies the 4K-byte aligned (exclusive) maximum Graphics Memory address access by an indirect object load in a AVC_BSD_OBJECT command. Indirect data accessed at this address and beyond will appear to be 0. Setting this field to 0 will cause this range check to be ignored. If non-zero, this address must be greater than the AVC Indirect Object Base Address. Hardware ignores this field if indirect data is not present. Format = GraphicsAddress[31:12]
	11:0	Reserved: MBZ

7.6.2 AVC_BSD_STATE Commands

The AVC_BSD_STATE commands are used to load various state information and parameters into the BSD unit to govern its operations. They include the decoding parameters extracted from different AVC syntax layers (e.g. NAL, PPS, SPS, Slice Header, SEI, etc.) that are to be consumed within the BSD (and MPR) unit, as well as decoding information that are used to drive and to be passed along to other processing stages (IT, MC and ILDB) that follow the BSD unit. Some of these state parameters are directly set to the corresponding AVC syntax elements' values; and others (with similar name as the corresponding AVC syntax element) are derived from them as indicated. There are also additional decoding parameters that are not defined as part of the AVC specification, but are needed for proper functioning (e.g. error handling state parameters).

Each state command is of different size, and is designed to be Dword (32-bit) aligned. Some of these parameters are defined in according to the AVC specification and some are only related to the current implementation of the BSD unit and Intel architecture.

These commands are issued prior to a set of BSD Object commands, so that all the required state variables are set appropriately prior to the actual decoding of the corresponding Slice Data portion of the bitstream. The BSD unit is a stateless machine, all state information and parameters must be set prior to the decoding. However, unless a reset is triggered, the current parameter values (programmed from a previous state command) remain until another corresponding AVC_BSD_STATE command comes in to re-program them.

There is an inherent order of setting the state variables, as dedicated by the H/W interface, and therefore the order of issuing the corresponding AVC_BSD_STATE commands. In general, we will follow the order listed below :

AVC_BSD_IMG_STATE (fixed size)

AVC_BSD_QM_STATE (variable size)

AVC_BSD_SLICE_STATE (variable size with fixed sized data structures)

AVC_BSD_BUF_BASE_STATE (fixed size)

All state information are sent to the BSD unit through AVC_BSD_STATE commands as inline data (some are fixed size, and some are variable in length but are predetermined by the driver before issuing the corresponding STATE Command).

The current active SPS can be chosen from an array of SPS with an index in the driver. Likewise, the current active PPS can be chosen from an array of PPS with an index in the driver.

7.6.2.1 AVC_BSD_IMG_STATE Command

AVC_BSD_IMG_STATE command encapsulates the decoding parameters that are read or derived from bitstream syntax elements of a NAL unit, a PPS unit, a SPS unit and a Slice Header (only includes those that are invariant within a picture). It includes all the relevant settings for the Frame Parameter and the Image Parameter Interface of the BSD Unit. These parameters are not changes from slice to slice of the same picture, but may change from picture to picture. Hence, this command is only issued at the beginning of processing a new picture and prior to the AVC_BSD_OBJECT command. It has a fixed size of 8 DWs (including the Command Opcode DW0).

AVC_BSD_IMG_STATE command must be sent in order for the H/W to sync to the beginning of a picture.

The values set for these state variables are retained internally across Slices, until they are reset by H/W Asynchronous Reset or changed by the next AVC_BSD_IMG_STATE command.

Dword	Bit	Description
0	31:29	Command Type = PARALLEL_VIDEO_PIPE = 3h
	28:16	AVC Command Opcode = AVC_BSD_IMG_STATE Pipeline[28:27] = BSD = 2h; Opcode[26:24] = AVC = 4h; Sub Opcode[23:16] = 0h
	15:0	DWord Length (Excludes DWords 0,1) = 0004h
1	31:16	Reserved. MBZ.
	15:0	Frame size in MB unit, FrameSizeInMBs[15:0] Note: bit 15 must be 0 and is ignored by the current H/W implementation, i.e. only bits[14:0] are significant. Unsigned integer value. The value for FrameSizeInMBs must match the product of FrameWidthInMBs and FrameHeightInMBs. Max. screen resolution is therefore limited to 2K x 2K in MB unit e.g for 1920x1080, FrameSizeInMBs[15:0] = 8160 (1920/16 * 1088/16; rounded up 1080) This parameter is specified for Intel interface only.
2	31:24	Reserved. MBZ.
	23:16	Frame height in MB unit, FrameHeightInMBs[7:0] Unsigned integer value. It is set to the value of (FrameHeightInMBsMinus1+ 1). Since the max value for FrameHeightInMBs is 255, the max allowed value for FrameHeightInMBsMinus1 is only 254. The min value for FrameHeightInMBs is 1. Although the max. value that can be specified for FrameHeightInMBs is 255 (in the current implementation), FrameWidthInMBs * FrameHeightInMBs must not exceed the max value of FrameSizeInMBs[14:0]. e.g. for 1920x1080, FrameHeightInMBs[7:0] is equal to 68 (1080 divided by 16, and rounded up, i.e. effectively specified as 1088 instead). It is derived from FrameHeightInMbs = (2 – frame_mbs_only_flag) * PicHeightInMapUnits (firmware ???), or PicHeightInMbs = FrameHeightInMbs / (1 + field_pic_flag) internally done (???)
	15:8	Reserved. MBZ. (bit [15:8] must be zero)

Dword	Bit	Description
	7:0	Frame width in MB unit, FrameWidthInMBs[7:0] Unsigned integer value. It is set to the value of (FrameWidthInMBsMinus1 + 1). FrameWidthInMBsMinus1 is equal to the value of the syntax element in the current active SPS. Since the max value for FrameWidthInMBs is 255, the max value allowed for FrameWidthInMBsMinus1 is only 254. The min value for FrameWidthInMBs is 1. Although the max. value that can be specified for FrameWidthInMBs is 255 (in the current implementation), FrameWidthInMBs * FrameHeightInMBs must not exceed the max value of FrameSizeInMBs[14:0]. e.g. for 1920x1080, FrameWidthInMBs[7:0] is equal to 120 (1920 divided by 16).
3	31:29	Reserved. MBZ (bit [31:29] must be zero)
	28:24	Second Chroma QP Offset, second_chroma_qp_offset[4:0] Signed integer value. It should be in the range of -12 to +12 (according to AVC spec). It specifies the offset for determining QP Cr from QP Y. It is set to the upper 5 bits of the value of the syntax element (Chroma_qp_offset[9:0]) read from the current active PPS. Chroma_qp_offset [4:0] – chroma_qp_offset_bits (from the current active PPS) Chroma_qp_offset [9:5] – second_chroma_qp_offset_bits
	32:21	Reserved. MBZ (bit[32:21] must be zero)
	20:16	Chroma QP Offset, chroma_qp_offset[4:0] Signed integer value. It should be in the range of -12 to +12 (according to AVC spec). It specifies the offset for determining QP Cb from QP Y. It is set to the lower 5 bits of the value of the syntax element (Chroma_qp_offset[9:0]) read from the current active PPS. Chroma_qp_offset [4:0] – chroma_qp_offset_bits (from the current active PPS) Chroma_qp_offset [9:5] – second_chroma_qp_offset_bits
	15	Special Kernel MB Scan Order for AVC ILDB Data Writes It controls the way of writing the generated ILDB Data package to memory. 0 – MB are arranged in Raster Scan Order 1 – MB are arranged in the Special Scan Order This parameter is specified for Intel interface only.

Dword	Bit	Description
	14	Special Kernel MB Scan Order for AVC-IT Command Writes It controls the way of writing the generated AVC-IT Command package to memory. 0 – MB are arranged in Raster Scan Order 1 – MB are arranged in the Special Scan Order This parameter is specified for Intel interface only.
	13	Special Kernel MB Scan Order for AVC-IT Data Writes It controls the way of writing the generated AVC-IT Data package to memory. 0 – MB are arranged in Raster Scan Order 1 – MB are arranged in the Special Scan Order This parameter is specified for Intel interface only.
	12	Monochrome Prediction Weighting Table Decoding Mode Flag, monochrome_pwt_flag Added to cover unapproved H.264 Spec. modification. Should always be written as “1”. This parameter is specified for Intel interface only.
	11	Reserved. MBZ
	10	QM Present Flag, qm_present_flag 0 – no Scaling Matrix is pressent nor sent to the BSD unit; use the flat4x4 (when transform_8x8_mode_flag == 0) or flat8x8 (when transform_8x8_mode_flag == 1) built-in QM matrices. That is, no QM_State command should follow, or the BSD unit should ignore it if it is issued (the BCS will continue to parse QM_State command, but the BSD unit will ignore and not to receive the incoming QM matrices data). 1 – Scaling Matrix may be present,depending on the use_default and list_present flags present in the inline data of a subsequent QM_STATE command. A QM_STATE command must follow to complete the specification of the QM matrices for the current picture decoding. It is derived from the pic_scaling_matrix_present_flag in the current active PPS and the seq_scaling_matrix_present_flag in the current active SPS. This parameter is specified for Intel interface only.

Dword	Bit	Description
	9:8	Image Structure, img_structure[1:0] The current decoding picture structure can only takes on 3 possible values : 00 – Frame Picture 01 – Top Field Picture 11 – Bottom Field Picture 10 – Invalid, not allowed. img_structure[0] can be used as a flag to distinguish between frame and field structure. It must be consistent with the field_pic_flag setting in the Slice Header. This parameter is specified for Intel interface only as a separate state (instead the img_structure[1]).
	7:0	Current Decoded Image Frame Store ID, img_dec_fs_idc Unsigned integer value. It specifies the destination (currently decoded) picture by its binding table index (intel implementation) of the current DPB. This parameter is programmed in frame. img_dec_fs_idc[4:0] can only be equal to 0 to 16. Bit[7:5] must be 0. When it is in the range of 0 to 15, hardware uses the selected address from direct_mv_rd0 to direct_mv_rd31 (based on img_structure and img_dec_fs_idc) as the Direct MV Write Base Address for the Current Picture. When it is 16, hardware uses the direct_mv_wr0_top or direct_mv_wr0_bottom (depending on img_structure) as the Direct MV Write Base Address for the Current Picture. direct_mv_wr0_top is selected for a frame picture or a top field picturem, and direct_mv_wr0_bottom for a bottom field picture. The typical usage is to program this field to 16. And specifically, in order to support 16 reference frames, this field must be programmed as 16. For picture referencing/indexing, Frame Store ID Bit[6:0] = 7F is used to designate a non-existing picture; otherwise, Bit[7:5] should always be 0, For img_dec_fs_idc[7:0] (current decoding picture), it should never be a non-existing picture. (It was used to write the POC of the current frame to the POC list. Since we are now writing the entire POC list at a time this is no longer used for this purpose) It is used to read the POC of the current frame for temporal direct motion vector weighting calculations. We could get the current POC with POCList[RefList[0]], but instead use the existing img_dec_fs_idc to determine the POC from the POCList[img_dec_fs_idc].
4	31:24	Residual Data Offset This gives the offset of the residual data in dwords. It should be 8 dword aligned. This should be programmed to the same value as the Residual Data Source Offset field in Vfe State Ex.
	23:18	Reserved: MBZ

Dword	Bit	Description
	17	Thread Synchronization Overwrite. This field indicates whether hardware overwrites the Thread Synchronization field in the MEDIA_OBJECT_EX command of each output macroblock. When this field is set, the Thread Synchronization field is set only for intra macroblocks (and at least one of the neighbors A, B, C, D, Left-Bottom is available). Otherwise (if a macroblock is a P, B or I_PCM), the field is not set. ***0 = Not Overwrite*** ***1 = Overwrite***
	16	Thread Synchronization. This field indicates whether the thread is a Synchronized Root Thread (SRT). When Thread Synchronization Overwrite is not set, this field is forwarded to the corresponding field in the MEDIA_OBJECT_EX command of each output macroblock. When Thread Synchronization Overwrite is set, hardware ignores this field. 0 = Not a SRT 1 = SRT
	15:13	Reserved. MBZ.
	12	Enable16MV This field, when set, enables the 16MV motion vector format in the AVC-IT output data buffer. If this field is set to 0, all MvSize = 16MV are mapped to 32MV. 0 – no MvSize of 16MV 1 – MvSize of 16MV enabled
	11:10	Chroma Format IDC, ChromaFormatIdc[1:0] It specifies the sampling of chroma component (Cb, Cr) in the current picture as follows : 00 – monochrome picture 01 – 4:2:0 picture 10 – 4:2:2 picture (not supported) 11 – 4:4:4 picture (not supported) It is set to the value of the syntax element read from the current active SPS. The corresponding Monochrome Flag (monochrome_flag) can be derived from this field as follows : 0 – Color Picture 1 – Monochrome picture (i.e. no Chroma components)
	9	Reserved. MBZ.

Dword	Bit	Description
	8	Enable the In-Loop Deblocking Filter information write, EnableILDBWrite 1 – enable ILDB writing output 0 – disable the ILDB writing output This bit controls ILDB information write to memory regardless of whether In-Loop Deblocking Filter is enabled or not for a given slice. If this bit is zero, even if a slice has in-loop deblocking enabled, no ILDB data is written to memory. This may be used for debugging purpose. On the other hand, if this bit is set, and if ILDB is disabled for a slice, ILDB data (default to no filtering) are written to memory for each macroblock in the slice. This allows ILDB to be performed at picture level regardless of the mixer of different slices (some has deblocking enabled and some disabled) within the picture. Default behavior when some slices of a picture has deblocking disabled. So, this will control the filling of those slices to the final output. This parameter is specified for Intel interface only.
	7	Entropy Coding Flag, entropy_coding_flag 0 – CAVLD bit-serial decoding mode. 1 – CABAC bit-serial decoding mdoe. It specifies one of the two possible bit stream decoding modes in the AVC. It is set to the value of the syntax element read from the current active PPS.
	6	Current Img Disposable Flag or Non-Reference Picture Flag, ImgDisposableFlag 0 – the current decoding picture may be used as a reference picture for others. 1 – the current decoding picture is not used as a reference picture (e.g. a B-picture cannot be a reference picture for any subsequent decoding) It is derived from ImgDisposableFlag = (nal_ref_idc == 0). nal_ref_idc is a syntax element from a NAL unit.
	5	Constrained Intra Prediction Flag, constrained_ipred_flag 0 – allows both intra and inter neighbouring MB to be used in the intra-prediction decoding of the current MB. 1 – allows only to use neighboring Intra MBs in the intra-prediction decoding of the current MB. If the neighbor is an inter MB, it is considered as not available. It is set to the value of the syntax element in the current active PPS.
	4	Direct 8x8 Inference Flag, direct_8x8_infer_flag 0 – allows subpartitioning to go below 8x8 block size (i.e. 4x4, 8x4 or 4x8) 1 – allows processing only at 8x8 block size. MB Info is stored for 8x8 block size. It is set to the value of the syntax element in the current active SPS. It specifies the derivation process for luma motion vectors in the Direct MV coding modes (B_Skip, B_Direct_16x16 and B_Direct_8x8). When frame_mbs_only_flag is equal to 0, direct_8x8_inference_flag shall be equal to 1. It must be consistent with the frame_mbs_only_flag and transform_8x8_mode_flag settings.

Dword	Bit	Description
	3	8x8 IDCT Transform Mode Flag, trans8x8_mode_flag Specifies 8x8 IDCT transform may be used in this picture 0 – no 8x8 IDCT Transform, only 4x4 IDCT transform blocks are present 1 – 8x8 Transform is allowed It is set to the value of the syntax element in the current active PPS.
	2	Frame MB only flag, frame_mbs_only_flag 0 – not true ; effectively enables the possibility of MBAFF mode. 1 – true, only frame MBs can occur in this sequence, hence disallows the MBAFF mode and field picture. It is set to the value of the syntax element in the current active SPS.
	1	MBAFF mode is active, mbaff_frame_flag. This bit is valid only when the img_structure[1:0] indicates the current picture is a frame. 0 – not in MBAFF mode 1 – in MBAFF mode It is derived from MbaffFrameFlag = (mb_adaptive_frame_field_flag && ! field_pic_flag). mb_adaptive_frame_field_flag is a syntax element in the current active SPS and field_pic_flag is a syntax element in the current Slice Header. They both are present only if frame_mbs_only_flag is 0. Although mbaff_frame_flag is a Slice Header parameter, its value is expected to be the same for all the slices of a picture. It must be consistent with the mb_adaptive_frame_field_flag, the field_pic_flag and the frame_mbs_only_flag settings.
	0	Field picture flag, field_pic_flag, specifies the current slice is a coded field or not. 0 – a slice of a coded frame 1 – a slice of a coded field It is set to the same value as the syntax element in the Slice Header. It must be consistent (??? Expand out) with the img_structure[1:0] and the frame_mbs_only_flag settings. Although field_pic_flag is a Slice Header parameter, its value is expected to be the same for all the slices of a picture.
5	31:0	AVC-IT Command Header, AvcItCmdHeader It allows the driver to directly specify the DW0 of the AVC-IT Command to be generated by the BSD unit that will pass down to drive the VFE unit. This parameter is specified for Intel interface only,.

7.6.2.2 AVC_BSD_QM_STATE Command

This command is only issued when qm_present_flag is set in the IMG_STATE command. If qm_present_flag is not set in the IMG_State command, and AVC_BSD_QM_STATE command is issued, the BCS will still process the command, but the BSD unit will ignore the data being forwarded.

Quantization matrices are loaded only if

- The current Slice is the first of a picture,
- Active SPS or active PPS has read in,
- Active PPS has changed between pictures

Only the latest scaling matrices read from the bitstream in processing the current active SPS and PPS are sent down to the BSD unit, as needed. Maximum there can be 8 matrices. If transform_8x8_mode_flag is set, maximum two 8x8 scaling matrices and six 4x4 scaling matrices may be present; otherwise only maximum six 4x4 scaling matrices may be present. Hence, this state descriptor has a variable length; the Pressent Flag, the List_Present flags and the Use_Default flags together can determine which and how many scaling matrices to follow. If the qm_present_flag is set in the IMG_State command, but a use_default flag of QM_State command is set, the corresponding scaling matrix is not sent. That is, all default scaling matrices are generated inside the BSD unit.

Each entry of a scaling matrix occupies 1 byte. Hence, the minimum size of the state descriptor is 1 Dword, and the maximum is 57 Dwords.

The Scaling Matrices, if present, must follow a strict ordering in the state descriptor. They are packed in the ascending order of the i index. Hence, the Dword 1 is refering to the Scaling Matrix with the lowest i. This state descriptor may be sent prior to decoding each Slice. The values set for these state variables are retained internally across Slices (QM_State is actually separated out from the IMG_State), until they are reset by H/W Asychronous Reset or changed by the next AVC_BSD_QM_STATE Command.

Dword	Bit	Description
0	31:29	Command Type = PARALLEL_VIDEO_PIPE = 3h
	28:16	AVC Command Opcode = AVC_BSD_QM_STATE Pipeline[28:27] = BSD = 2h; Opcode[26:24] = AVC = 4h; Sub Opcode[23:16] = 1h
	15:0	DWord Length (Excludes DWords 0,1) = 0000h to max 0038h
1	31:16	Reserved. MBZ

Dword	Bit	Description

	15:8	Use built-in Default QM Flags for the current Slice, use_default_flags[i], i = 0 to 7
		i=0 – 4x4 Intra Y
		i=1 – 4x4 Intra Cr
		i=2 – 4x4 Intra Cb
		i=3 – 4x4 Inter Y
		i=4 – 4x4 Inter Cr
		i=5 – 4x4 Inter Cb
		i=6 – 8x8 Intra Y
		i=7 – 8x8 Inter Y
		use_default_flags[i] = 1, and if the corresponding qm_list_flags[i] and qm_present_flag are also set, then use the built-in default scaling matrix
		use_default_flags[i] = 0, indicates the built-in default QM matrix is not used, and the scaling matrix read from the bitstream is used.
		qm_present_flag (in IMG_STATE), qm_list_flags[i], use_default_flags[i] and LevelScale matrices must be set in consistent and coherent.
		This parameter is specified for Intel interface only.

	7:0	QM List Present Flags for the current Slice, qm_list_flags[i], i = 0 to 7
		i=0 – 4x4 Intra Y
		i=1 – 4x4 Intra Cb
		i=2 – 4x4 Intra Cr
		i=3 – 4x4 Inter Y
		i=4 – 4x4 Inter Cb
		i=5 – 4x4 Inter Cr
		i=6 – 8x8 Intra Y
		i=7 – 8x8 Inter Y
		qm_list_flags[i] = 1 and qm_present_flag is set, indicates either using the scaling matrix read from the bit stream, or use the built-in default matrix if the Use_Default is also set. The following LevelScale field is present if and only if the corresponding bit in this field is set.
		qm_list_flag[i] = 0, use the fallback rule specified in the AVC spec.
		qm_present_flag (in IMG_STATE), qm_list_flags[i], use_default_flags[i] and LevelScale matrices must be set in consistent and coherent.
		This parameter is specified for Intel interface only.

Dword	Bit	Description
0 or 4 Dwords	16 bytes	Luma4x4 Intra WeightScale (i=0) Unsigned integer value, ranging from 0 to 255. An array of 4x4 scaling values (one-to-one correspondence with the coefficient position) for an Intra Luma block that is stored in raster order. It is set to the values derived from the syntax elements in the current active PPS and active SPS.
0 or 4 Dwords	16 bytes	Cb4x4 Intra WeightScale (i=1) Unsigned integer value, ranging from 0 to 255. An array of 4x4 scaling values (one-to-one correspondence with the coefficient position) for an Intra Chroma Cb block that is stored in row major order (i.e. raster scan order). It is set to the values derived from the syntax elements in the current active PPS and active SPS.
0 or 4 Dwords	16 bytes	Cr4x4 Intra WeightScale (i=2) Unsigned integer value, ranging from 0 to 255. An array of 4x4 scaling values (one-to-one correspondence with the coefficient position) for an Intra Chroma Cr block that is stored in row major order (i.e. raster scan order). It is set to the values derived from the syntax elements in the current active PPS and active SPS.
0 or 4 Dwords	16 bytes	Luma4x4 Inter WeightScale (i=3) Unsigned integer value, ranging from 0 to 255. An array of 4x4 scaling values (one-to-one correspondence with the coefficient position) for an Inter Luma block that is stored in row major order (i.e. raster scan order). It is set to the values derived from the syntax elements in the current active PPS and active SPS.
0 or 4 Dwords	16 bytes	Cb4x4 Inter WeightScale (i=4) Unsigned integer value, ranging from 0 to 255. An array of 4x4 scaling values (one-to-one correspondence with the coefficient position) for an Inter Chroma Cb block that is stored in row major order (i.e. raster scan order). It is set to the values derived from the syntax elements in the current active PPS and active SPS.
0 or 4 Dwords	16 bytes	Cr4x4 Inter WeightScale (i=5) Unsigned integer value, ranging from 0 to 255. An array of 4x4 scaling values (one-to-one correspondence with the coefficient position) for an Inter Chroma Cr block that is stored in row major order (i.e. raster scan order). It is set to the values derived from the syntax elements in the current active PPS and active SPS.

Dword	Bit	Description
0 or 16 Dwords	64 bytes	Luma8x8 Intra WeightScale (i=6) Unsigned integer value, ranging from 0 to 255. An array of 8x8 scaling values (one-to-one correspondence with the coefficient position) for an Intra Luma block that is stored in row major order (i.e. raster scan order). It is set to the values derived from the syntax elements in the current active PPS and active SPS.
0 or 16 Dwords	64 bytes	Luma8x8 Inter WeightScale (i=7) Unsigned integer value, ranging from 0 to 255. An array of 8x8 scaling values (one-to-one correspondence with the coefficient position) for an Inter Luma block that is stored in row major order (i.e. raster scan order). It is set to the values derived from the syntax elements in the current active PPS and active SPS.

7.6.2.3 AVC_BSD_SLICE_STATE Command

This state descriptor includes all the information for

Inter-Prediction Reference Lists L0 and L1

Weights and Offset for Inter-Prediction

They are all syntax elements read from or derived from the Slice Header and may, therefore, be changed on a Slice boundary.

The state descriptor of the Slice_State command is of variable size. However, all the constitute components, when present, are of fixed size structure. The Slice_State command should not be issued at all, if there is no inter-prediction decoding or none of its components are present. If the present flag is set to 0, and the corresponding data is still sent, unpredicted result. If

The state information for motion vector prediction specifies the frame indices of the reference list list_0 and list_1. It is sent to the MPR unit for determining which frames to fetch for Temporal Direct Motion Vectors. These MPR states may be changed on a Slice boundary, the entire reference lists are sent, and are packed in a consecutive order (and there is no bubble – i.e. no empty/invalid/unspecified entry in between).

The very first entry of each reference list is always started with index 0, and is named as the top of the reference list.

For the weights and offsets specification in the SLICE_STATE state descriptor, it specifies the weights and offsets for inter-prediction using the reference list list_0 and list_1. It is outputted from the BSD unit to the AVC-IT interface. These Weights and Offsets states are set to the values of syntax elements read from the Slice Header, which may be changed on a Slice boundary. They are packed in a consecutive order (and there is no bubble – i.e. no empty/invalid/unspecified entry in between). As a result, the weight and offset state descriptor has a fixed length, if it is present.

Only when WeightedPredFlag = 1 (enable weighted prediction), will there be a L0 weight+ offset table (and there is no L1 stuff in P slice) being sent to the BSD unit through the Slice_State command. If WeightedBiPredIdc[1:0] != 1 and WeightedPredFlag = 0, no Slice_State command should be issued for this table.

Only when WeightedBiPredIdc[1:0] = 1 (explicit weighted prediction), will there be a L1 and/or a L0 weight+offset tables being sent to the BSD unit through the Slice_State command. If WeightedBiPredIdc[1:0] != 1 and WeightedPredFlag = 0, no Slice_State command should be issued for this table.

As a result, this state descriptor has a variable length, and is predetermined by the driver. The minimum size is 2 Dwords if this command is ever issued, since at minimum even there is no L1 and L0 and no weight and offset, there is still DW0 and DW1. The maximum size is 113 Dwords.

Weight-Offset programming model[Errata – Dev Ctg/DevEL] Driver needs to look ahead all slices for the frame and check if any weight value is 128. This can only happen when luma_weight_l0/l1/chroma_weight_l0/l1 flags are 0. If it finds a 128 then it needs to remap this value to a non used value within a 16 bit twos compliment range and then give the remapped number to the kernel for identifying this case.

Dword	Bit	Description
0	31:29	Command Type = PARALLEL_VIDEO_PIPE = 3h
	28:16	AVC Command Opcode = AVC_BSD_SLICE_STATE Pipeline[28:27] = BSD = 2h; Opcode[26:24] = AVC = 4h; Sub Opcode[23:16] = 2h
	15:0	DWord Length (Excludes DWords 0,1) = 0000h (min) to 00D0h (max)
1	31:4	Reserved. MBZ.
	3	Weight+Offset L1Table Present Flag If Weight+OffsetL1 Table Present Flag = 1, indicates the full 32-entry L1 Table for Y, Cb and Cr is followed; otherwise it is not present at all (no further inline data). It must be set in consistent with the WeightedPredFlag and WeightedBiPredIdc in the Img_State command. This parameter is specified for Intel interface only.
	2	Weight+OffsetL0 Table Present Flag If Weight+OffsetL0 Table Present Flag = 1, indicates the full 32-entry L0 Table for Y, Cb and Cr is followed; otherwise it is not present at all (no further inline data). It must be set in consistent with the WeightedPredFlag and WeightedBiPredIdc in the Img_State command. This parameter is specified for Intel interface only.

Dword	Bit	Description
	1	RefPicListL1 Present Flag If RefPicListL1 Present Flag = 1, indicates the full 32-entry L1 list is followed; otherwise it is not present at all (no further inline data). It must be set in consistent with the Num_Ref_Idx_L1 derived from the syntax element in the Slice Header, and if it is a B Slice. This parameter is specified for Intel interface only.
	0	RefPicListL0 Present Flag If RefPicListL0 Present Flag = 1, indicates the full 32-entry L0 list is followed; otherwise, it is not present at all (no further inline data). It must be set in consistent with the Num_Ref_Idx_l0 derived from the syntax element in the Slice Header, and if it is a P or B Slice. This parameter is specified for Intel interface only.

Dword	Bit	Description
2-9 or none	31:0 each	Reference List0 Write This field is present only if the RefPicListL0 Present Flag = 1; otherwise, it should not be present. If present, it always specifies 32 reference pictures in list0, regardless they are "existing picture" or not. If a picture is non-existing, the corresponding entry should be set to all ones. Each list entry is 1 byte. A 32-bit DW can hold 4 list entries in the following format 31:24 entry X+3 (e.g. list0_3) 23:16 entry X+2 (e.g. list0_2) 15:8 entry X+1 (e.g. list0_1) 7:0 entry X (e.g. list0_0) X is replaced by the paddr[2:0] * 4 ; paddr[5:0] with 0x30 and 0x37 The byte definition for a reference picture : Bit 7 : Non-Existing – indicates that frame store index that should have been at this entry did not exist and was replaced by an index for error concealment Bit 6 : Long term bit – set this reference picture to be used as long term reference Bit 5 : Field picture flag – indicates frame/field Bit 4:0 : Frame store index or Frame Store ID (Bit 4:1 is used to form the binding table index in intel implementation) This is the final Reference List L0 after any reordering specified in the Slice Header as well as modified by the driver, and its indices values are all translated to the intel specification. If the reference picture is a frame (Bit5 = 1), frame store ID is always an even number. The frame store ID = 0 and 1 are assigned to the current decoding picture, hence reference picture ID will start from 2 onwards. This list is used in MV information output of the BSD unit (but non-existing pictures are mapped to 00 to simplify the kernel). DMV also read and write Mvlist0 using this frame store ID.

Dword	Bit	Description
10-17 or none	31:0 each	Reference List1 Write This field is present only if the RefPicListL1 Present Flag = 1; otherwise, it should not be present. If present, it always specifies 32 reference pictures in list1, regardless they are existing or not. If a picture is non-existing, the corresponding entry should be set to all ones. Each list entry is 1 byte. A 32-bit DW can hold 4 list entries in the following format 31:24 entry X+3 (e.g. list1_7) 23:16 entry X+2 (e.g. list1_6) 15:8 entry X+1 (e.g. list1_5) 7:0 entry X (e.g. list1_4) X is replaced by the paddr[2:0] * 4 ; paddr[5:0] with 0x38 and 0x3F The byte definition for a reference picture: Bit 7 : Non-Existing – indicates that frame store index that should have been at this entry did not exist and was replaced by an index to for error concealment Bit 6 : Long term bit – set this reference picture to be used as long term reference Bit 5 : Field picture flag – indicates frame/field Bit 4:0 : Frame store index (or also known as binding table index in intel implementation) This is the final Reference List L1 after any reordering specified in the Slice Header as well as modified by the driver, and its indices values are all translated to the intel specification. This list is used in MV information output of the BSD unit. DMV also read and write Mvlist1 using this frame store ID.

Dword	Bit	Description
18-65 or none	31:0 each	WeightOffset[L0=0][i=0 to 31][Y/Cb/Cr][weight/offset] WeightOffset[0][i=0][Y=0][Offset=0] WeightOffset[0][i=0][Y=0][Weight=1] WeightOffset[0][i=0][Cb=1][Offset=0] WeightOffset[0][i=0][Cb=1][Weight=1] WeightOffset[0][i=0][Cr=2][Offset=0] WeightOffset[0][i=0][Cr=2][Weight=1] : WeightOffset[0][i=31][Y=0][Offset=0] WeightOffset[0][i=31][Y=0][Weight=1] WeightOffset[0][i=31][Cb=1][Offset=0] WeightOffset[0][i=31][Cb=1][Weight=1] WeightOffset[0][i=31][Cr=2][Offset=0] WeightOffset[0][i=31][Cr=2][Weight=1] Signed integer values, ranging from -128 to 127. This field is present only if the Weight+OffsetL0 Table Present Flag= 1; otherwise, it should not be present. If present, the full table is always specified. Any reference list L0[i] that does not exist, the corresponding weight and offset are set to 0. Weight and Offset are 1 byte each. Weight is the high byte and Offset is the lower byte. Hence, we are packing 2 sets of weight and offset into 1 DW. Bits[31:16] will store the set of weight and offset will higher values of i or Y/Cr/Cb (represented as 0/1/2 in the indexing). WeightOffset[L0=0][i=0 to 31][Y=0] (i.e. luma_weight_l0[i]) are specified for the weighting and offset factors applied to the luma prediction value for list 0 prediction using RefPicList0[i] (one-to-one correspondence in i). When luma_weight_l0_flag (Slice Header syntax element) is equal to 1, the value of luma_weight_l0[i] shall be in the range of –128 to 127. When luma_weight_l0_flag is equal to 0, luma_weight_l0[i] shall be inferred to be equal to 2luma_log2_weight_denom for RefPicList0[i]. luma_log2_weight_denom is a Slice Header syntax element. WeightOffset[L0=0][i=0 to 31][Cb=1] (i.e. chromaCb_weight_l0[i]) are specified for the weighting and offset factors applied to the chroma Cb prediction values for list 0 prediction using RefPicList0[i] (one-to-one correspondence in i). When chroma_weight_l0_flag (Slice Header syntax element) is equal to 1, the value of chromaCb_weight_l0[i] shall be in the range of –128 to 127. When chroma_weight_l0_flag is equal to 0, chromaCb_weight_l0[i] shall be inferred to be equal to 2chroma_log2_weight_denom for RefPicList0[i]. chroma_log2_weight_denom is a Slice Header syntax element. WeightOffset[L0=0][i=0 to 31][Cr=2] (i.e. chromaCr_weight_l0[i]) are specified for the weighting and offset factors applied to the chroma Cr prediction values for list 0 prediction using RefPicList0[i] (one-to-one correspondence in i). When chroma_weight_l0_flag (Slice Header syntax element) is equal to 1, the value of chromaCr_weight_l0[i] shall be in the range of –128 to 127. When chroma_weight_l0_flag is equal to 0, chromaCr_weight_l0[i] shall be inferred to be equal to 2chroma_log2_weight_denom for RefPicList0[i]. The table is set to the values derived from the syntax elements read in the Slice Header.

Dword	Bit	Description

66-113
or none
	31:0
	each
		WeightOffset[L1=1][32][Y/Cb/Cr][weight/offset]
		WeightOffset[1][0][Y=0][Offset=0]
		WeightOffset[1][0][Y=0][Weight=1]
		WeightOffset[1][0][Cb=1][Offset=0]
		WeightOffset[1][0][Cb=1][Weight=1]
		WeightOffset[1][0][Cr=2][Offset=0]
		WeightOffset[1][0][Cr=2][Weight=1]
		:
		WeightOffset[1][31][Y=0][Offset=0]
		WeightOffset[1][31][Y=0][Weight=1]
		WeightOffset[1][31][Cb=1][Offset=0]
		WeightOffset[1][31][Cb=1][Weight=1]
		WeightOffset[1][31][Cr=2][Offset=0]
		WeightOffset[1][31][Cr=2][Weight=1]
		Signed integer values, ranging from -128 to 127.
		This field is present only if the Weight+OffsetL1 Table Present Flag= 1; otherwise, it should not be present. If present, the full table is always specified. Any reference list L1[i] that does not exist, the corresponding weight and offset are set to 0.
		Weight and Offset are 1 byte each. Weight is the high byte and Offset is the lower byte. Hence, we are packing 2 sets of weight and offset into 1 DW. Bits[31:16] will store the set of weight and offset will higher values of i or Y/Cr/Cb (represented as 0/1/2 in the indexing).
		WeightOffset[L1=1][i=0 to 31][Y=0] (i.e. luma_weight_l1[i]) are specified for the weighting and offset factors applied to the luma prediction value for list 1 prediction using RefPicList1[i] (one-to-one correspondence in i). When luma_weight_l1_flag (Slice Header syntax element) is equal to 1, the value of luma_weight_l1[i] shall be in the range of –128 to 127. When luma_weight_l1_flag is equal to 0, luma_weight_l1[i] shall be inferred to be equal to 2luma_log2_weight_denom for RefPicList1[i]. luma_log2_weight_denom is a Slice Header syntax element.
		WeightOffset[L1=0][i=0 to 31][Cb=1] (i.e. chromaCb_weight_l1[i]) are specified for the weighting and offset factors applied to the chroma Cb prediction values for list 1 prediction using RefPicList1[i] (one-to-one correspondence in i). When chroma_weight_l1_flag (Slice Header syntax element) is equal to 1, the value of chromaCb_weight_l1[i] shall be in the range of –128 to 127. When chroma_weight_l1_flag is equal to 0, chromaCb_weight_l1[i] shall be inferred to be equal to 2chroma_log2_weight_denom for RefPicList1[i]. chroma_log2_weight_denom is a Slice Header syntax element.
		WeightOffset[L1=0][i=0 to 31][Cr=2] (i.e. chromaCr_weight_l1[i]) are specified for the weighting and offset factors applied to the chroma Cr prediction values for list 1 prediction using RefPicList1[i] (one-to-one correspondence in i). When chroma_weight_l1_flag (Slice Header syntax element) is equal to 1, the value of chromaCr_weight_l1[i] shall be in the range of –128 to 127. When chroma_weight_l1_flag is equal to 0, chromaCr_weight_l1[i] shall be inferred to be equal to 2chroma_log2_weight_denom for RefPicList1[i].
		The table is set to the values derived from the syntax elements read in the Slice Header.

7.6.2.4 AVC_BSD_BUF_BASE_STATE Command

This command contains the base addresses for memory buffers allocated for BSD operations. It does not include the byte stream read address, which is to be specified in the AVC_BSD_OBJECT Command.

There are six base addresses being defined:

BSD Row Store memory base address for R/W,

MPR Row Store memory base address for R/W,

MB Info memory base address for Write-only,

Decoded Uncompressed Coefficients, IPCM Coefficients or MV array memory base address for Write-only,

Direct MV memory base address for Write-only, and

31 Direct MV memory base addresses for Read-only.

Not including the current frame, which is represented by the Write Base address

Although the number of Direct MV base addresses should be the same as the number of reference frames in the reference list (specified in the Slice_STATE), since it is possible to have the same frame presents in both list0 and list1 and there is no logics to cross checking, it is decided to send the complete addressess list always.

The state descriptor provides the state and information required to decode a Slice, and their values may subject to change on a Slice boundary. It should be sent prior to decoding each Slice. The values set for these state variables are retained internally across Slices, until they are reset by H/W Asychronous Reset or changed by the next AVC_BSD_BUF_BASE_STATE Command.

When switching video stream, these base address must be reprogrammed.

Dword	Bit	Description
0	31:29	Command Type = PARALLEL_VIDEO_PIPE = 3h
	28:16	AVC Command Opcode = AVC_BSD_BUF_BASE_STATE Pipeline[28:27] = BSD = 2h; Opcode[26:24] = AVC = 4h; Sub Opcode[23:16] = 3h
	15:0	DWord Length (Excludes DWords 0,1) = 0048h

Dword	Bit	Description
1	31:6	BSD Row Store Base Address This field provides the base address of the scratch buffer used by BSD unit to store BSD information for the previous row to assist BSD decoding of the macroblocks in the current row. The BSD Row Store buffer must be 64-byte cacheline aligned. 1.5 cacheline (CL) per MB when in MBAFF mode (row of MB pair); 0.75 CL per MB for non-MBAFF. So, to support 120 MBs per row (1920 screen resolution), 1.5 * 120 * 64 bytes = 11,520 bytes are required. Half cacheline alignment should be followed. Hardware uses the horizontal address of each macroblock to address the BSD Row Store.
	5:0	Reserved : MBZ
2	31:6	MPR Row Store Base Address This field provides the base address of the scratch buffer used by BSD unit to store MPR information for the previous row to assist MPR decoding of the macroblocks in the current row. 1 cacheline (CL) per MB when in MBAFF mode (row of MB pair); 0.5 CL per MB for non-MBAFF, So, ot support 120 MBs per row (1920 screen resolution), 1 * 120 * 64 bytes = 7,680 bytes are required. Half cacheline alignment should be followed. Hardware uses the horizontal address of each macroblock to address the MPR Row Store.
	5:0	Reserved : MBZ
3	31:6	AVC-IT Command MB Info Write Base Address (defined for streamout mode) This field provides the base address of the AVC-IT command output buffer that contains the MB Info needed for inverse transform and motion compensation. This buffer must be 64-byte cacheline aligned. Hardware uses the MB number within the picture to address this buffer.. [Also provide the data structure for each macroblock. This is important as the interface is shared between this pipe and the 3DPIPE.] [Addressing in raster order and also the wave-front order. Providing a reference to the description of these two orders. Also discuss how to handle MBAFF.] 1 cacheline (CL) for each MB, 8160 MBs (1920x1088 screen resolution) per frame, hence 1*8160*64 bytes per frame = 522,240 bytes are required. Cacheline alignment should be followed.
	5:0	Reserved : MBZ

Dword	Bit	Description
4	31:12	AVC-IT Data Write Base Address This is the 4KB aligned portion of the base address for the AVC-IT data write buffer. Together with the AVC-IT Data Write Offset, it forms the base pointer where the AVC-IT data (including for example motion vectors, weight-offsets, non-zero residual data), where data for the first macroblock will be written to. This field affect the final memory address computed for the AVC-IT Data output. However, it doesn't affect the Indirect Data Start Address computed by BSD hardware that is in the AVC-IT Command output (DW3 of the output AVC-IT Command). Each macroblock within the AVC-IT data buffer has a fixed size even though only valid data are written to memory. The remaining bytes for each macroblock are undefined. Assuming 32 cachelines are needed per macroblock, a HD buffer (with a resolution of 1920x1088) is 16,711,680 bytes. Format = GraphicsAddress[31:12] Programming Note: As the AVC-IT data buffer will be used by GPE as the indirect object buffer for MEDIA_OBJECT_EX command, this field should be programmed the same as the Indirect Object Base Address field of the STATE_BASE_ADDRESS command.
	11:6	AVC-IT Data Write Offset This offset is relative to the AVC-IT Data Write Buffer Base Address. This field, together with the AVC-IT Data Write Base Address, forms the 64-byte cacheline aligned address of the AVC-IT data write buffer. ***This field is used to compute the Indirect Data Start Address in the AVC-IT Command output (DW3 of the output AVC-IT Command).***
	5:0	Reserved : MBZ
5	31:6	ILDB Data Write Base Address This field provides the base address of the ILDB output data buffer. This buffer must be 64-byte cacheline aligned. 2 cachelines (CL) for each MB, 8160 MBs (1920x1088 screen resolution) per frame, hence 2*8160*64 bytes per frame = 1,044,480 bytes are required. Real data only in 1 CL, extra CL added due to driver/kernel requirement for data expansion. Cacheline alignment should be followed.
	5:0	Reserved : MBZ
6	31:6	Direct MV Read Base Address for Reference Frame 0, 64-bytes cache-line aligned – direct_mv_rd0[31:6] This is a ROW Store private buffer space. The read buffer size is 557,056 bytes for 1 frame (the selected colPic). It is scalable with frame height, but does not scale with frame width as H/W uses a fixed with of 128 MBs for row store buffers which is the smallest power of 2 value larger than 120 – 1920x1088 screen resolution. This field may also be used as the Direct MV Write Base Address for the Current Picture if img_dec_fs_idc is set to 0 to 15. See img_dec_fs_idc for more details.
	5:0	Reserved : MBZ
	5:0	Reserved : MBZ

Dword	Bit	Description
7-37	31:6	Direct MV Read Base Address for Reference Frame 1 to Reference Frame 31 Definition follows that of direct_mv_rd0[31:6]
	5:0	Reserved : MBZ
38	31:6	Direct MV Write Base Address for the Current Picture (default to be Reference Frame 0 TOP field), 64-bytes cache-line aligned –direct_mv_wr0_top[31:6] This is a ROW Store private buffer space The write buffer size is 557,056 bytes for 1 frame (the selected colPic). It is scalable with frame height, but does not scale with frame width as H/W uses a fixed with of 128 MBs for row store buffers which is the smallest power of 2 value larger than 120 – 1920x1088 screen resolution. There is only one write buffer for capturing the DMVs for the current picture. This field is used if img_dec_fs_idc is 16 and the current picture is a frame picture or a top field picture.
	5:0	Reserved : MBZ
39	31:6	Direct MV Write Base Address for the Current Picture (default to be Reference Frame 0 BOTTOM field), 64-bytes cache-line aligned – direct_mv_wr0_bottom[31:6] This is a ROW Store private buffer space The write buffer size is 557,056 bytes for 1 frame (the selected colPic). It is scalable with frame height, but does not scale with frame width as H/W uses a fixed with of 128 MBs for row store buffers which is the smallest power of 2 value larger than 120 – 1920x1088 screen resolution. There is only one write buffer for capturing the DMVs for the current picture. This field is used if img_dec_fs_idc is 16 and the current picture is a bottom field picture.
	5:0	Reserved : MBZ
40-73	31:0	POC List, POCList[0…33][31:0] Each POC value is a signed 32-bit number. One-to-one correspondence with the 34 Direct MV Read/Write Address for Reference Frames. There are 34 POC entries in the list, indexed by the frame_store_ID. POCList[0] is the CurrPic POC. If it is in a field mode, POCList[0] is the CurrFOC for the top field and the POCLIst[1] is the CurrFOC for the bottom field. For frame mode, every other entry is skipped. Frame_Store_ID[5:1]+T/B bit[0] is used to index into the POCList.

7.6.3 AVC_BSD_OBJECT Command

The AVC_BSD_OBJECT command is the only primitive command for the AVC_BSD Unit. The same command is used for both CABAC and CAVLD modes. The Slice Data portion of the bitstream is loaded as indirect data object.

Before issuing an AVC_BSD_OBJECT command, all BSD states need to be valid. Therefore the commands used to set these states need to have been issued at some point prior to the issue of an AVC_BSD_OBJECT command.

The Encryption of bit stream is indicated in bit 31 of DW 1. For such encrypted bit streams, this command has one extra dword (DW 8), to load in the initial counter value for AES-128 Counter mode decryption of the bit stream.

To handle encrypted bitstream decoding, host software is required to align the 16byte chunk containing this first byte of actual bit-stream slice, to a naturally aligned 16byte boundary, i.e. that 16byte chunk should start on an aligned 16byte address boundary. Host software should ensure that after the last valid byte of bitstream data and before encryption, enough padding is added up to the end of a 16byte chunk. In addition, for a protected context, all kernel source and destination data surface addresses must be in PCM space.

Dword	Bits	Description
0	31:29	Command Type = PARALLEL_VIDEO_PIPE = 3h
	28:16	AVC Command Opcode = AVC_BSD_OBJECT Pipeline[28:27] = BSD = 2h; Opcode[26:24] = AVC = 4h; Sub Opcode[23:16] = 8h
	15:0	DWord Length (Excludes DWords 0,1) [DevCTG, DevELK] = 0006h without encryption (bit 31 of DW1 is set to 0), where inline data are in dwords 3-7 = 0007h with AES decryption on (bit 31 of DW1 is set to 1), where inline data are in dwords 3-8 = 000Ah for all cases where inline data are in dwords 3-15
1	31	Bit stream is AES-128 Counter mode Encrypted ‘0’ – Not encrypted ‘1’ - Encrypted
	30	
	29	
	28:22	
	21:0	
2	31:29	
	28:0	

Dword	Bits	Description
3-variable	31:0 Each	

Note that scattered raw slice data format is not supported by hardware. The size of a compression buffer for a slice needs to be allocated, which can be big, according to the minimal compression ratio required by a given profile/level. Hardware doesn't support the option for the driver to allocate 'nominally sized raw slice buffers' and use multiple such buffers for the worst case slice data.

7.6.3.1 Inline Data Description

The Inline Data includes all the required slice decoding states, extracted primarily from the Slice Header and its derivatives. It provides information for the following operations:

CABAC/CAVLD decoding

Internal error handling at the Slice boundary

H/W Automatic Error Concealment

Motion vector prediction decoding (MPR)

BSD output compositing (feeding the subsequent IT/MC/ILDB operations)

The only H/W error detection and concealment mechanism is comparing the Slice_Start_MB_Num of decoding the new current slice with the Current_MB_Num resulted from decoding the previous slice. If they do not match, an error is assumed. There are two possible cases, either "greater than" or "less than". If the Slice_Start_MB_Num is less than the Current_MB_Num, the Current_MB_Num is adjusted internally to be the same as the Slice_Start_MB_Num. If the Slice_Start_MB_Num is greater than, there is a gap between the Current_MB_Num and the Slice_Start_MB_Num that needs to be filled. The filling is started from [Current_MB_Num – Rewind_Num], and the method of filling can be either intra or inter as specified in the state descriptor.

These state/parameter values may subject to change on a Slice boundary, and must be provided in each AVC_BSD_OBJECT command. The values set for these variables are retained internally, until they are reset by H/W Asynchronous Reset or changed by the next AVC_BSD_OBJECT command.

Dwords 3-7 are the common in-line data for [DevCTG, DevELK]

Dword	Bit	Description
3	31	Concealment Method This field specifies the method used for concealment when error is detected. If set, a copy from collocated macroblock location is performed from the concealment reference indicated by the ConCeal_Pic_Id field. If it is not set, a copy from the current picture is performed using Intra 16x16 Prediction method. 0 – Intra 16x16 Prediction 1 – Inter P Copy

Dword	Bit	Description
	30	Init Current_MB_Number When set, the current Slice_Start_MB_Num, Slice_MB_Start_Hor_Pos and Slice_MB_Start_Vert_Pos fields will be used to initialize the Current_MB_Number register. This effectively disables the concealment capability.
	29:28	Reserved : MBZ
	27:24	Rewind_Num This field provides the number of MBs or MBAFF pairs to rewind when performing concealment. This is ignored if Slice_Start_MB_Num is smaller than or equal to the Current_MB_Num. Error Concealment MB start position = Current_MB_NUMBER – MIN { Decoded_MB_NUMBER, (MBAFF ? 2 : 1)*Force_Skip_Rewind } Decoded_MB_NUMBER = Number of MBs decoded in the previous slice.
	23:22	Reserved : MBZ
	21:16	Conceal_Pic_Id (Concealment Picture ID) This field identifies the picture in the reference list to be used for concealment. This field is only valid if Concealment Method is Inter P Copy. Bit 21 = 0 – Frame Picture = 1 – Field Picture Bit 20:16 – Frame Store Index[4:0]
	15	Reserved : MBZ
	14	BSDPrematureComplete Error Handling 1 – Set the interrupt to the driver (provide MMIO registers for MB address R/W) (???) 0 – Ignore the error and continue (masked the interrupt), assume the H/W automatically perform the error handling It occurs in situation where the Slice decode is completed but there are still data in the bitstream.
	13	Reserved : MBZ
	12	MPR Error (MV out of range) Handling– what to do when the specific error has occurred 1 – Set the interrupt to the driver (provide MMIO registers for MB address R/W) (???) 0 – Ignore the error and continue (masked the interrupt), assume the H/W automatically perform the error handling
	11	Reserved : MBZ
	10	Entropy Error Handling – what to do when the specific error has occurred 1 – Set the interrupt to the driver (provide MMIO registers for MB address R/W) (???) 0 – Ignore the error and continue (masked the interrupt), assume the H/W automatically perform the error handling

Dword	Bit	Description
	9	Reserved : MBZ
	8	MB Header Error Handling – what to do when the specific error has occurred 1 – Set the interrupt to the driver (provide MMIO registers for MB address R/W) (???) 0 – Ignore the error and continue (masked the interrupt), assume the H/W automatically perform the error concealment.
	7:4	Reserved : MBZ
	3:0	Slice_Type (Slice Type) 0000 – P Slice 0001 – B Slice 0010 – I Slice Other encodings are reserved (note that bits[3:2] must be 0) It is set to the value of the syntax element read from the Slice Header.
4	31:30	Reserved : MBZ
	29:24	Num_Ref_Idx_L1 (Number of Reference Pictures in Inter-prediction List 1) This field is valid only for decoding a B Slice, for which it is expected to have at least one entry in the reference list L1; otherwise (if Slice Type is not a B Slice), this field must be set to 0. This field can be derived for a B Slice from the Slice Header syntax element NumRefIdxActiveMinus1 as, Num_Ref_Idx_L1 = NumRefIdxActiveMinus1[1] + 1. Format U6 with valid range of [0, 32].
	23:22	Reserved : MBZ
	21:16	Num_Ref_Idx_L0 (Number of Reference Pictures in Inter-prediction List 0) This field is valid for decoding a P or B Slice, for which it is expected to have at least one entry in the reference list L0; otherwise (if Slice Type is not a P or B Slice), this field must be set to 0. This field can be derived for a P or B Slice from the Slice Header syntax element NumRefIdxActiveMinus1 as, Num_Ref_Idx_L0 = NumRefIdxActiveMinus1[0] + 1. Format: U6 with valid range of [0, 32]
	15:11	Reserved : MBZ
	10:8	Log2WeightDenomChroma It is the base 2 logarithm of the denominator for all Chroma (Cb and Cr) weighting factors. The value of chroma_log2_weight_denom should be in the range of 0 to 7. It is set to the value of the syntax element read from the Slice Header Pred_Weight_Table(). Format: U3 with valid range of [0, 7].
	7:3	Reserved : MBZ

Dword	Bit	Description
	2:0	Log2WeightDenomLuma It is the base 2 logarithm of the denominator for all Luma weighting factors. It is set to the value of the syntax element read from the Slice Header Pred_Weight_Table(). Format: U3 with valid range of [0, 7].
5	31:30	Weighted_Pred_Idc (Weighted Prediction Indicator) This field indicates the Weighted Prediction mode for a P or B Slice. It is a combined field corresponding to the syntax element WeightedBiPredIdc or WeightedPredFlag read from the current active PPS. If it is a B-Slice, these 2 bits is interpreted as : 00 – specifies the default weighted inter-prediction to be applied 01 – specifies the explicit weighted inter-prediction to be applied 10 – specifies the implicit weighted inter-prediction to be applied 11 – Reserved (not allowed) If it is a P Slice, Weighted_Pred_Idc is interpreted as : 00 – disables weighted inter-prediction (default weighted) 01 – enables weighted inter-prediction (explicit weighted) 10-11 – Reserved Only when in B Slice with Weighted_Pred_Idc = 1 (explicit weighted prediction), will there be a L1 and/or a L0 weight+offset tables being sent to the BSD unit through the Slice_State command. Only when in P Slice with Weighted_Pred_Idc = 1, will there be a L0 weight+offset table being sent to the BSD. If Weighted_Pred_Idc != 1 for B Slice or Weighted_Pred_Idc =0 for P Slice, no Slice_State command should be issued to send these tables. If still being issued, the data is read but ignored.
	29	Direct_Pred_Type (Direct Prediction Type) Type of direct prediction used for B Slices. This field is valid only for Slice_Type = B Slice; otherwise, it must be set to 0. 0 – Temporal 1 – Spatial
	28:27	Disable_Deblocking_Filter_Idc (Disable Deblocking Filter Indicator) 00 - filterInternalEdgesFlag is set equal to 1 01 – disable all deblocking operation, no deblocking parameter syntax element is read; filterInternalEdgesFlag is set equal to 0 10 - macroblocks in different slices are considered not available; filterInternalEdgesFlag is set equal to 1 11 – Reserved (not defined in AVC)
	26	Reserved : MBZ

Dword	Bit	Description
	25:24	Cabac_Init_Idc Specifies the index for determining the initialization table used in the context variable initialization process.
	23:22	Reserved : MBZ
	21:16	Slice_Qp (Slice Quantization Parameter) Quantization Parameter for current slice. Derived from PPS and slice_delta_qp syntax element in Slice Header.
	15:12	Reserved : MBZ
	11:8	Slice_Beta_Offset_Div2 Specifies the offset used in accessing the deblocking filter strength tables. It is a sign 2's complement number, ranging from -6 to +6 inclusive. Format: S4 with valid range [-6, 6]
	7:4	Reserved. MBZ.
	3:0	Slice_Alpha_C0_Offset_Div2 Specifies the offset used in accessing the deblocking filter strength tables. It is a sign 2's complement number, range from -6 to +6 inclusive. Format: S4 with valid range [-6, 6]
6	31:24	Slice_MB_Start_Vert_Pos (Slice Vertical Position) This field specifies the position in y-direction of the first macroblock in the Slice in unit of macroblocks.
	23:16	Slice_MB_Start_Hor_Pos (Slice Horizontal Position) This field specifies the position in x-direction of the first macroblock in the Slice in unit of macroblocks.
	15	Reserved : MBZ
	14:0	Slice_Start_Mb_Num The MB number (linear MB address in a picture) at the start of a Slice, it must match with the Slice Horizontal Position (Slice_MB_Start_Hor_Pos) and Vertical Position (Slice_MB_Start_Vert_Pos) in the picture. In creating the Phantom Slice for error concealment, this field should set to the total number of MB in the current picture + 1.
7	31:8	Reserved : MBZ
	7	Fix_Prev_Mb_Skipped Enables an alternative method for decoding mb_skipped, to cope with an encoder that codes a skipped MB as a direct MB with no coefficient.
	6:3	Reserved : MBZ

Dword	Bit	Description
	2:0	First_MB_Bit_Offset (First Macroblock Bit Offset) This field provides the bit offset of the first macroblock of the Slice in the first byte of the input compressed bitstream. This field is the number of valid bits minus 1 of the first byte. For example when this field is set to 7, it indicates that the whole first byte (all 8 bits) are valid; If this field is set to 0, it indicates that only the 1 LSB of the first byte is valid data for the first macroblock. Programming notes: Byte alignment of CABAC. According to AVC Specification, if a slice is coded as CABAC (entropy_coding_flag = CABAC), the first macroblock of a slice is always byte aligned, i.e., the slice header is bit padded to align on byte boundary. So for CABAC case, this field must be set to 7 indicating that the first macroblock is byte aligned. Host software must take care of slice header bit padding to find the first bit of the first macroblock. It should be noted that hardware can take care of the slice header bit padding for most cases, but it falls to deal with one corner condition – when the first byte where the slice header ends is the last byte of a cache line. Format: U3

Dwords 8 is only present when AES decryption is enabled (Bit31 of DW1 is set to 1) for [DevCTG, DevELK].

Dword	Bit	Description
8	31:0	Initial counter value for AES-128 Counter mode decryption. 32-bit initial counter value (which does not need to be secure) for the AES counter mode decryption. This counter value should be applied in the AES Counter mode decryption of the first 16byte chunk of aligned encrypted slice data which contains the 1st starting byte of valid slice bitstream. If Bit stream is AES-128 Counter mode Encrypted is 0, i.e. not encypted bit stream, this field is ignored and should be MBZ.

Dwords 9-15

Dword	Bit	Description
9	31:23	Reserved. MBZ

Dword	Bit	Description
	22:0	Indirect Data's Raw Data Length of Bitstream and Headers. This field provides the length in bytes of the indirect data. This DWord field is valid only for bitstream from integrated hw decryption. It includes the byte length from the beginning of the first byte of the bitstream till the last byte of the bitstream, including all the intermediate header bytes. In the case of encrypted bit stream, the last byte of valid bit stream data, can line up on any byte aligned location within a 16byte encrypted chunk. This means that the app. will do the rounding or padding up to the end of the 16bytes chunk before encryption. The length programmed has one extra bit than the actual byte length of the valid bit-stream data. Format = U23 in bytes If the bitstream is not packet-based, this field is ignored and MBZ
10	31:29	Packet-based function. Starting bit position of the Bit Vector in the 1st byte of the fetched Bit Vector surface.
	28:16	Reserved : MBZ
	15:0	Total Packet Count. This is a total count of all the packets in the packet-based bitstream, including those that are fetched for processing and, also those packets which are skipped (because they belonged to a different stream). Format = U23 in bytes If the bitstream is not packet-based, all the fields in this DWord are ignored and MBZ.
11	31:29	Reserved : MBZ
	28:0	Packet-based Bitsream Bit Vector Surface starting Byte address. This field specifies the Graphics Memory starting address of the Bit Vector to be used for skipping/fetching bitstream slice packets. This virtual address points to a 4K page and is therefore 4KByte aligned. Each data bit in this surface refers to a bitstream packet and indicates: '0' – The packet is used in the bitstream decode for the current slice, and therefore needs to be fetched. '1' – The packet is not used in the bitstream decode for the current slice, and therefore need not be fetched. If the bitstream is not packet-based, this field is ignored and MBZ.

Dword	Bit	Description
12	31:0	Weight128LumaL0 – One bit for each of the l0/l1 entries for luma and chroma components. This bit should be set by driver whenever the 16 bit weight value from the application is equal to 128. this will be passed on as inline data in the media object command. Each of the bits is for one entry in the L0 list with the LSB representing if index 0 is 128 or not for the luma component
13	31:0	Weight128LumaL1 – One bit for each of the l0/l1 entries for luma and chroma components. This bit should be set by driver whenever the 16 bit weight value from the application is equal to 128. this will be passed on as inline data in the media object command. Each of the bits is for one entry in the L1 list with the LSB representing if index 0 is 128 or not for the luma component.
14	31:0	Weight128ChromaL0 – One bit for each of the l0/l1 entries for luma and chroma components. This bit should be set by driver whenever the 16 bit weight value from the application is equal to 128. this will be passed on as inline data in the media object command. Each of the bits is for one entry in the L0 list with the LSB representing if index 0 is 128 or not for the chroma component
15	31:0	Weight128ChromaL1– One bit for each of the l0/l1 entries for luma and chroma components. This bit should be set by driver whenever the 16 bit weight value from the application is equal to 128. this will be passed on as inline data in the media object command. Each of the bits is for one entry in the L1 list with the LSB representing if index 0 is 128 or not for the chroma component.

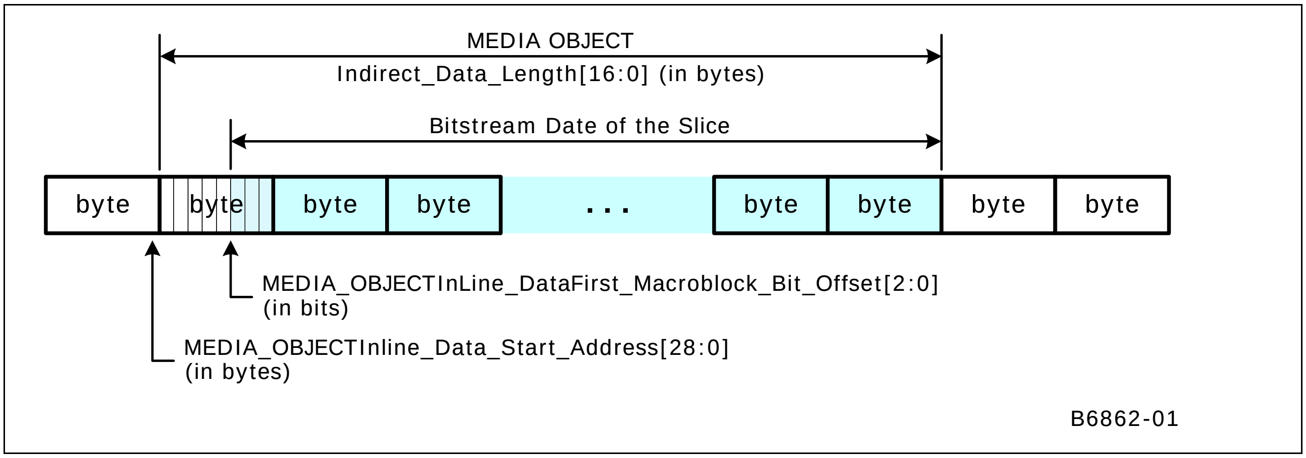
7.6.3.2 Indirect Data Format

Each AVC_BSD_OBJECT command corresponds to the processing of one slice of a picture (a picture can have only one slice). All syntax and derived parameters above Slice Data Layer are passed in either using State commands or as Inline data of the AVC_BSD_OBJECT command, and the Slice Data Layer and below are passed in as indirect data.

The indirect data start address in AVC_BSD_OBJECT specifies the starting Graphics Memory address of the bitsream data that follows the slice header. It provides the byte address for the first macroblock of the slice. Together with the First MB Bit Offset field, it provides the starting bit location of the MB within the compressed bitstream.

The indirect data length in AVC_BSD_OBJECT provides the length in bytes of the bitstream data for this slice. It includes the first byte of the first MB (includes the one that contains the FirstMBBiteOffset) and the last non-zero byte of the last MB in the slice. H/W ignores the contents after the last non-zero byte.

Figure 7-1. Indirect data buffer for a slice


MEDIA OBJECT
Indirect_Data_Length[16:0] (in bytes)
Bitstream Date of the Slice
byte
byte
byte
byte
. . .
byte
byte
byte
byte
MEDIA_OBJECTInLine_DataFirst_Macroblock_Bit_Offset[2:0]
(in bits)
MEDIA_OBJECTInline_Data_Start_Address[28:0]
(in bytes)
B6862-01

7.7 AVC_BSD Output Definitions

Three data buffers are output from the BSD engine:

AVC-IT Command Buffer – MB info signals from AC to AM (mb_avail, cbp) + AM (avail, cbp)

AVC-IT Data Buffer – IPCM or Coeff + MV/Weight+Offset

AVC-ILDB Data Buffer

The format of the AVC-IT Command Buffer can be found in the Media Chapter, under section MEDIA_OBJECT_EX Command as well as its subsection In-line Data Format in AVC-IT Mode.

The format of the AVC-IT Data Buffer can be found in the Media Chapter, under section MEDIA_OBJECT_EX Command and subsection Indirect Data Format in AVC-IT Mode.

The format of the AVC-LF Data Buffer can be found in the Data Port Chapter, under section AVC Loop Filter Read.

7.7.1 AVC-IT Data Buffer Description

7.7.1.1 Motion Vectors

Motion vectors (x and y components) are sent to the MC as indirect data in memory together with residual data. In non-I_PCM mode, they are the packed computed MVs (Mvpredicted + MVDelta) of a MB; in I_PCM mode, there is no MV information need to be sent. MV can be generated for DMV-spatial, DMV-temporal or motion-search MV. The two components of MV (MVx, Mvy) are signed extended into a 16-bit signed number each and are packed into a single Dword. MVs of a MB are packed in 512-bit cache lines (or in half-cacheline). Based on the MB_type, we can determine the number of MVs being present in the corresponding MB. There can be up to 2, 4, 8, 16 and 32 MVs for a MB in both directions. A MV can associate with a 4x4,4x8,8x4 sub-block, a 8x8, 8x16, 16x8 block or 16x16 MB, and for a direction (L0 and L1). The MV for sub-blocks are sent in a raster order within a block, and in raster order in respect to the blocks within a MB. Zero-value MVs are sent as is. When L1 is not in use, no MV are sent for L1.

DWord	Bit	Description
For each Dword	31:16	MV_y Signed extended 2's complement
	15:0	MV_x Signed extended 2's complement

7.7.1.1.1 MV Organization for 16x16 Block Size

The MVs are specified in a fixed data structure, which is organized as if there are always four 8x8 blocks. Hence, the real MV(s) is (are) replicated across the remaining entries.

7.7.1.1.2 MV Organization for 16x8 Block

A MB that is 16x8 partitioned consists of 2 parts – top and bottom partitions. However, its MVs are specified as if the MB is broken down into four 8x8 blocks. That is, each 16x8 partition is treated as two identical 8x8 blocks.

7.7.1.1.3 MV Organization for 8x16 Block

A MB that is 8x16 partitioned consists of 2 parts – left and right partitions. However, its MVs are specified as if the MB is broken down into four 8x8 blocks. That is, each 8x16 partition is treated as two identical 8x8 blocks.The MV are specified as if the MB is broken down into four 8x8 blocks.

7.7.1.1.4 MV Organization for 8x8 Block Without Subpartition

A MB that is 8x8 partitioned consists of 4 parts – top-left (0), top-right(1), bottom-left(2) and bottom-right(3) partitions. The MVs are specified for each 8x8 blocks.

7.7.1.1.5 MV Organization for 8x8 Block with Subpartition

A MB that has at least one sub-partitioning (4x4, 8x4 or 4x8), the entire MB is broken down into 16 4x4 units for the specification of MV. That is, the MVs are specified for each 4x4 blocks.

The 4x4 block are scanned in the following order :

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

Replication Rules :

For an 8x8 partition, the 8x8 MVs are replicated into all the four 4x4 blocks.

For an 4x8 subpartition within an 8x8 partition, each 4x8 is broken down into 2 4x4 stacking vertically. The 4x8 MVs are replicated into both 4x4 blocks.

For an 8x4 subpartition within an 8x8 partition, each 8x4 is broken down into 2 4x4 stacking horizontally. The 8x4 MVs are replicated into both 4x4 blocks.

For an 4x4 subpartition within an 8x8 partition, each 4x4 has its own MVs.

For L0 MV

If L0 interprediction exists, the corresponding L0 MV is used

Else if L1 interprediction exits (of the same block), set to the same as L1 MV

Else set to 0

For L1 MV

If L1 interprediction exists, the corresponding L1 MV is used

Else if L0 interprediction exits (of the same block), set to the same as L0 MV

Else set to 0

MBLK data (containing the packed MV/Weight,Offset/Coeff) offset in the indirect buffer can be dword aligned. MV/Weight,offset/Coeff are packed in memory as dword aligned data structure (MS constraint to Dword align for a buffer), VFE has to derive the size for each based on bMSize. BSD output is half_cacheline aligned.

Block address (right to left)

MV write out first, then coeff

Weight offset does not go down to 4x4.

Replicate Weight/Offset, decomp 16x8 to 8x8, with the same Weight/Offset.

7.7.1.2 Inter-prediction Weight and Offset Block

Weight and offset are packed together next to each other. A weight and an offset are each 1 byte. The high byte is always the weight and the lower byte is always the offset. The smallest block size for Weight and Offset specifications is 8x8 (no subpartitioning, since interprediction mode does not go below 8x8).

7.7.1.2.1 Weight+Offset Organization for 16x16 Block Size

The Weight+Offset are specified in a fixed data structure. Hence, the real Weight+Offsets are replicated across the remaining entries.

	DWord	Description
	DW0 [15:0]	L0 Weight+Offset_Y0 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Y0 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Y0 Else set to 0
	DW0 [31:16]	L1 Weight+Offset_Y0 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Y0 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Y0 Else set to 0
	DW1 [15:0]	L0 Weight+Offset_Cb0 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cb0 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cb0 Else set to 0
	DW1 [31:16]	L1 Weight+Offset_Cb0 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cb0 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cb0 Else set to 0
	DW2 [15:0]	L0 Weight+Offset_Cr0 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cr0 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cr0 Else set to 0
	DW2 [31:16]	L1 Weight+Offset_Cr0 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cr0 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cr0 Else set to 0
	DW3	Reserved.

	DWord	Description
	DW4	Replicated DW0 (L0 Weight+Offset_Y0 and L1 Weight+Offset_Y0)
	DW5	Replicated DW1 (L0 Weight+Offset_Cb0 and L1 Weight+Offset_Cb0)
	DW6	Replicated DW2 (L0 Weight+Offset_Cr0 and L1 Weight+Offset_Cr0)
	DW7	Reserved.

7.7.1.2.2 Weight+Offset Organization for non-16x16 Block Size

The Weight+Offset are specified in a fixed data structure. It is organized as if the MB is broken into four 8x8 blocks. Hence, the real Weight+Offsets are replicated across the remaining entries.

Replication Rules :

For 8x16 partition, each 8x16 is broken down into 2 8x8 stacking vertically. The 8x16 Weight+Offset are replicated into the two 8x8 blocks.

For 16x8 partition, each 16x8 is broken down into 2 8x8 stacking horizontally. The 16x8 Weight+Offset are replicated into the two 8x8 blocks.

For 8x8 partition, the MB is broken down into 4 8x8 blocks. Each has its own set of Weight and Offset.

	DWord	Description
	DW0 [15:0]	L0 Weight+Offset_Y0 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Y0 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Y0 Else set to 0
	DW0 [31:16]	L1 Weight+Offset_Y0 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Y0 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Y0 Else set to 0
	DW1 [15:0]	L0 Weight+Offset_Cb0 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cb0 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cb0 Else set to 0

	DWord	Description
	DW1 [31:16]	L1 Weight+Offset_Cb0 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cb0 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cb0 Else set to 0
	DW2 [15:0]	L0 Weight+Offset_Cr0 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cr0 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cr0 Else set to 0
	DW2 [31:16]	L1 Weight+Offset_Cr0 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cr0 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cr0 Else set to 0
	DW3	Reserved.
	DW4 [15:0]	L0 Weight+Offset_Y1 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Y1 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Y1 Else set to 0
	DW4 [31:16]	L1 Weight+Offset_Y1 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Y1 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Y1 Else set to 0
	DW5 [15:0]	L0 Weight+Offset_Cb1 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cb1 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cb1 Else set to 0

	DWord	Description
	DW5 [31:16]	L1 Weight+Offset_Cb1 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cb1 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cb1 Else set to 0
	DW6 [15:0]	L0 Weight+Offset_Cr1 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cr1 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cr1 Else set to 0
	DW6 [31:16]	L1 Weight+Offset_Cr1 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cr1 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cr1 Else set to 0
	DW7	Reserved.
	DW8 [15:0]	L0 Weight+Offset_Y2 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Y2 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Y2 Else set to 0
	DW8 [31:16]	L1 Weight+Offset_Y2 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Y2 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Y2 Else set to 0
	DW9 [15:0]	L0 Weight+Offset_Cb2 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cb2 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cb2 Else set to 0

	DWord	Description
	DW9 [31:16]	L1 Weight+Offset_Cb2 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cb2 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cb2 Else set to 0
	DW10 [15:0]	L0 Weight+Offset_Cr2 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cr2 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cr2 Else set to 0
	DW10 [31:16]	L1 Weight+Offset_Cr2 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cr2 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cr2 Else set to 0
	DW11	Reserved.
	DW12 [15:0]	L0 Weight+Offset_Y3 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Y3 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Y3 Else set to 0
	DW12 [31:16]	L1 Weight+Offset_Y3 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Y3 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Y3 Else set to 0
	DW13 [15:0]	L0 Weight+Offset_Cb3 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cb3 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cb3 Else set to 0

	DWord	Description
	DW13 [31:16]	L1 Weight+Offset_Cb3 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cb3 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cb3 Else set to 0
	DW14 [15:0]	L0 Weight+Offset_Cr3 The high byte is weight and the low byte is offset If L0 interprediction exists, it is equal to L0 Weight+Offset_Cr3 Else if L1 interprediction exists, it is a replication of L1 Weight+Offset_Cr3 Else set to 0
	DW14 [31:16]	L1 Weight+Offset_Cr3 The high byte is weight and the low byte is offset If L1 interprediction exists, it is equal to L1 Weight+Offset_Cr3 Else if L0 interprediction exists, it is a replication of L0 Weight+Offset_Cr3 Else set to 0
	DW15	Reserved.

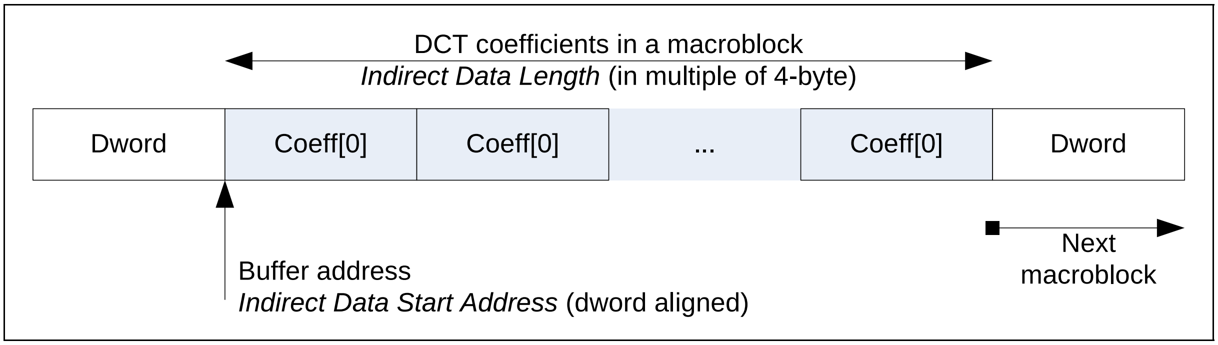
7.7.1.3 Quantized DCT Coefficient Block

In non-I_PCM mode, quantized DCT coefficients are sent to AVC-IT as indirect data in memory (AVC-IT Data Buffer). They are organized on a per MB basis (does it include skip MB and are they zero filled), i.e. all coefficients of a MB are packed together; in I_PCM mode, the AVC-IT buffer is packed with actual 8-bit pixel samples on a per MB basis.

Coefficients are packed in 512-bit cache lines (Note: could be dword aligned, and not cache line aligned).

Only the non-zero quantized DCT coefficients within a MB are sent and they are packed in the order of 4x4DCY (if any), 8x8Y0, 8x8Y1, 8x8Y2, 8x8Y3, 2x2DCCb (if any), 2x2DCCr (if any), 8x8Cb4 and 8x8Cr5, as shown in Figure 7-2. For I16x16Intra mode, the DCY is the very first block; in all other prediction modes, 8x8Y0 is the first block.

Figure 7-2. Structure of the Quantized DCT Coefficients Data Buffer


DCT coefficients in a macroblock
Indirect Data Length (in multiple of 4-byte)
Dword
Coeff[0]
Coeff[0]
...
Coeff[0]
Dword
Buffer address
Indirect Data Start Address (dword aligned)
Next
macroblock

Each non-zero coefficient in the indirect data buffer is contained in a doubleword-size (32-bit) data structure containing the coefficient index, end of block (EOB) flag and the fixed-point coefficient value in 2's compliment form.

As shown in Table 7-1, index is the row major 'raster' index of the coefficient within an 8x8 block. DCT coefficient is a 16-bit value in 2's complement, which is clamped to a 12-bit signed value by the host.

Table 7-1. Structure of a quantized DCT coefficient block

DWord	Bit	Description
For each Dword	31:16	Quantized DCT Coefficient Value or Pixel sample value This field contains the value of the non-zero quantized DCT coefficient in 2's compliment.
	15:7	Reserved: MBZ

	6:1	Index
		This field specifies the raster-scan address (raw address) of the quantized DCT coefficient (or pixel sample in I_PCM) within their corresponding block (some of size 8x8, and some are 4x4 or 2x2) . For example, coefficient in an 8x8Y0 at location (row, column) = (0, 0) has an index of 0; that at (2, 3) has an index of 2*8 + 3 = 19. For 4x4 YDC block, index ranges from 0 to 15; for 2x2 CrDC/CbDC, ranges from 0 to 3.
		Format = U6
		Range = [0, 63]

	0	EOB (End of Block) This field indicates whether the quantized DCT coefficient is the last one of the current block (8x8, 4x4 or 2x2).

7.7.1.4 I_PCM Pixel Sample Block

DWord	Bit	Description
For each Dword	31:0	8-bit Packed Pixel sample value Each Dword contains 4 pixel sample values (zero or not) packed together.

7.7.1.5 Motion Prediction Reference Indices

Motion prediction reference indices for L0 and L1 are sent separately to the MC as inline data in memory. In non-I_PCM mode, they are the packed for a MB and are aligned with the associated MVs; in I_PCM mode, there is no MV information need to be sent. MV can be generated for DMV-spatial, DMV-temporal or motion-search MV. Each reference index is a 4bit unsigned number, except that 1111 (-1) is used to mark for a non-existing reference picture. 8 reference indices can be packed into a single Dword. Reference indices of a MB are packed in 512-bit cache lines (or in half-cacheline ???). Based on the MB_type, we can determine the number of reference indices being present in the corresponding MB. A Reference indices can associate with a a 8x8, 8x16, 16x8 block or 16x16 MB, and for a direction (L0 and L1). The Reference indices for sub-blocks are sent in a raster order within a block, and in raster order in respect to the blocks within a MB. When L1 is not in use, no reference indices are sent for L1.

7.7.1.6 MB Neighbor Availability in Intra-Prediction Modes

mb_nb_avail_intra[5:0] provides the individual MB availability information as follows. [4:3] bit ordering is under discussion. Current MB is labelled as X. For a MB pair, both the top and the bottom have the same picture type.

mb_nb_avail_intra[0] : MB D (top left neighbor of current MB X) availability

mb_nb_avail_intra[1] : MB C (top right neighbor of current MB X) availability

mb_nb_avail_intra[2] : MB B (top neighbor of current MB X) availability

mb_nb_avail_intra[3] : MB A availability. A is the left neighbor of the current MB X, or A is the top MB of the left neighbor of the current MB pair X.

mb_nb_avail_intra[4] : MB E availability. E is the bottom MB of the left neighbor of the current MB pair X.

mb_nb_avail_intra[5] : MB F availability. F is the upper-left corner neighbor pixel ([-1,7]) for the lower-left 8x8 sub macroblock.

mb_nb_avail_intra[5 :0] is derived from the internal intermediate variable mb_avail_i [0..7] (internally generated by AM unit based on mb_nb_avail[0..3]), which in turns is derived from the initial variable mb_nb_avail[3:0].

mb_nb_avail[3:0] is derived for a MB (a MB unit) if in non-MBAFF mode or for a MB pair (a MB unit) if in MBAFF mode, from the following conditions :

A MB is marked as “Not Available” if anyone of the following conditions is true :

(MbAddr – Start_MB_addr) < 0 (error condition), for each Slice

MbAddr > CurrMBAddr (go beyond the current MB location, i.e. to the right or below)

The MB with address MbAddr belongs to a different slice than the MB with address CurrMbAddr

Set the mb_nb_avail[i] accordingly for the MB; mb_nb_avail[i] = 0 if “not available”, i = 0 to 3

For each non-pair MB with address CurrMbAddr, check the following availability, and set the mb_nb_avail[i] accordingly for the corresponding MB

mbAddrA = CurrMbAddr – 1,

mbAddrA is marked as not available when CurrMbAddr % PicWidthInMbs is equal to 0 (i.e. the MB with address CurrMbAddr is located along the left edge of the picture)

mbAddrB = CurrMbAddr – PicWidthInMbs

mbAddrC = CurrMbAddr – PicWidthInMbs + 1

mbAddrC is marked as not available when (CurrMbAddr + 1) % PicWidthInMbs is equal to 0 (i.e. the MB with address CurrMbAddr is located along the right edge of the picture)

mbAddrD = CurrMbAddr – PicWidthInMbs – 1

mbAddrD is marked as not available when CurrMbAddr % PicWidthInMbs is equal to 0 (i.e. the MB with address CurrMbAddr is located along the left edge of the picture)

mbAddrD D (top-left) mb_nb_avail[0] mb_avail_i[0,1]	mbAddrB B (top) mb_nb_avail[2] mb_avail_i[4,5]	mbAddrC C (top-right) mb_nb_avail[1] mb_avail_i[2,3]
mbAddrA A (left) mb_nb_avail[3] mb_avail_i[6,7]	X CurrMbAddrX	N/A
N/A	N/A	N/A

For each MB pair (in MBAFF mode) with address CurrMbAddr, check the following availability, and set the mb_nb_avail[i] accordingly for the corresponding neighbor MB pair (Note that in MBAFF mode, the top MB always has an even address and the bottom MB has an odd address; also in MBAFF, both MBs must have the same availability value, hence no need to specify top or bottom

MB in the mb_nb_avail[i]. However, the top and the bottom MB may code with different modes - intra or inter)

mbAddrA =2 * (CurrMbAddr / 2 – 1)

mbAddrA is marked as not available when (CurrMbAddr / 2) % PicWidthInMbs is equal to 0 (i.e. the MB pair with address CurrMbAddr is located along the left edge of the picture)

mbAddrB =2 * (CurrMbAddr / 2 – PicWidthInMbs)

mbAddrC = 2 * (CurrMbAddr / 2 – PicWidthInMbs + 1)

mbAddrC is marked as not available when (CurrMbAddr / 2 + 1) % PicWidthInMbs is equal to 0 (i.e. the MB with address CurrMbAddr is located along the right edge of the picture)

mbAddrD = 2 * (CurrMbAddr / 2 – PicWidthInMbs - 1)

mbAddrD is marked as not available when (CurrMbAddr / 2) % PicWidthInMbs is equal to 0 (i.e. the MB with address CurrMbAddr is located along the left edge of the picture)

mbAddrE = mbAddrA + 1

its MB availability should be the same as that of mbAddrA

mbAddrD D0 mb_nb_avail[0] mb_avail_i[0]	mbAddrB B0 mb_nb_avail[2] mb_avail_i[4]	mbAddrC C0 mb_nb_avail[1] mb_avail_i[2]
mbAddrD+1 D1 mb_nb_avail[0] mb_avail_i[1]	mbAddrB+1 B1 mb_nb_avail[2] mb_avail_i[5]	mbAddrC+1 C1 mb_nb_avail[1] mb_avail_i[3]
mbAddrA A0 mb_nb_avail[3] mb_avail_i[6]	CurrMbAddrX X0 or	N/A
mbAddrA+1 A1/E (mbAddrE) mb_nb_avail[3] mb_avail_i[7]	CurrMbAddrX X1	N/A

Additional conditions for each Intra_prediction Modes can further refine the MB neighbor's (N = A, B, C, D, or E) availability :

Intra_4x4 prediction

If any of the following conditions is true, the neighbor is marked as "not available for Intra_4x4 prediction"

mbAddrN is not available,

the macroblock mbAddrN is coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1.

the top right 4x4 block (luma4x4BlkIdx = 4 or 12) to the block with luma4x4BlkIdx equal to 3 or 11 (because not decoded yet) – Kernel will take care of it for internal blocks.

Intra_8x8 prediction

If any of the following conditions is true, the neighbor is marked as "not available for Intra_8x8 prediction"

mbAddrN is not available,

the macroblock mbAddrN is coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1,

Intra_16x16 prediction

If any of the following conditions is true, the neighbor is marked as "not available for Intra_16x16 prediction"

mbAddrN is not available,

the macroblock mbAddrN is coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1.

The above derivation of MB availability flag can be summarized into one equation:

mb_avail_i[i] = mb_nb_avail[i/2] & (is_intra_MB[i] | ! constrained_intra_pred_flag), i = 0 to 7.

mb_avail_i[0] – for MB D0

mb_avail_i[1] – for MB D1

mb_avail_i[2] – for MB C0

mb_avail_i[3] – for MB C1

mb_avail_i[4] – for MB B0

mb_avail_i[5] – for MB B1

mb_avail_i[6] – for MB A0

mb_avail_i[7] – for MB A1

is_intra_MB[1:0] - Upleft_mb_iblocks[0:1], i.e. D1 and D0

Specifies if the upper left MB or the upper left MB pair to the current MB is an intra MB or not

For the upper left neigbor being a MB pair, the Upleft_mb_iblocks[1] is for the Top MB and the Upleft_mb_iblocks[0] is for the Bottom MB

is_intra_MB[3:2] - Upright_mb_iblocks[0:1] , i.e. C1 and C0

Specifies if the upper right MB or the upper right MB pair to the current MB is an intra MB or not

For the upper right neigbor being a MB pair, the Upright_mb_iblocks[1] is for the Top MB and the Upright_mb_iblocks[0] is for the Bottom MB

is_intra_MB[5:4] - Up_mb_iblocks[0:1] , i.e. B1 and B0

Specifies if the upper MB or the upper MB pair to the current MB is an intra MB or not

For the upper neigbor being a MB pair, the Up_mb_iblocks[1] is for the Top MB and the Up_mb_iblocks[0] is for the Bottom MB

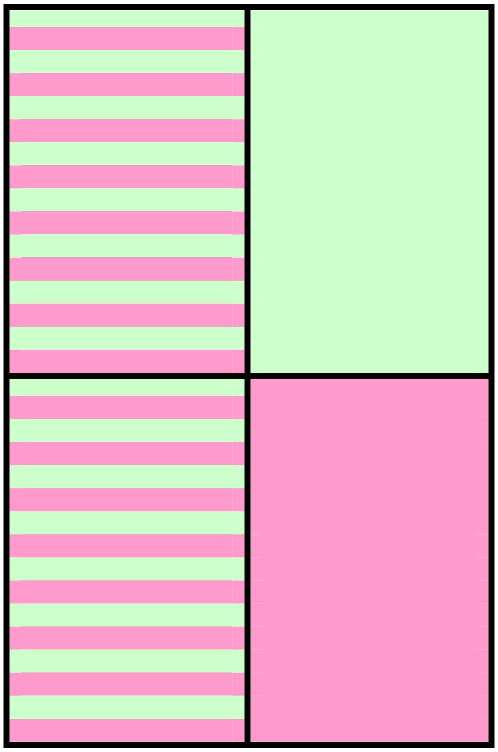
is_intra_MB[7:6] - left_mb_iblocks[0:1] , i.e. A1 and A0

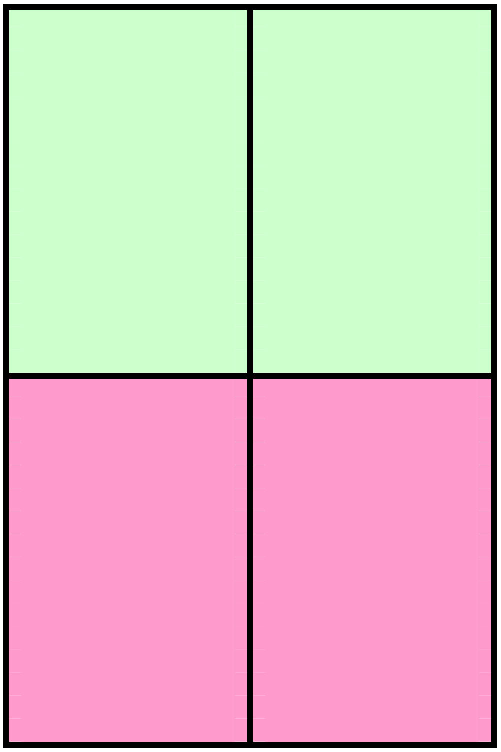
Specifies if the left MB or the left MB pair to the current MB is an intra MB or not

For the left neigbor being a MB pair, the Left_mb_iblocks[1] is for the Top MB and the Left_mb_iblocks[0] is for the Bottom MB

To speed up the MC kernels performance for Intra-Prediction modes, the 8-bit MB availability flags Mb_avail_i[7:0] are refined to reduce the number of kernel instructions. The modified availability information has 6 bits, mb_nb_avail_intra[5:0]. Bits [4] and [5] are used only in MBAFF mode. The only situation when mb_nb_avail_intra[4] and mb_nb_avail_intra[3] may be different is when the current MB pair is a field MB and the left neighbor MB pair is a frame MB.

The diagram shows the physical storage location for a MB pair (16 pixel x 32 rows, numbered from 0 to 31). Logically, the top MB (16x16) of a MB pair contains all the even lines (labelled with index 0, e.g. A0), and the bottom MB (16x16) of a MB pair contains all the odd lines (labelled with index 1, e.g. A1).



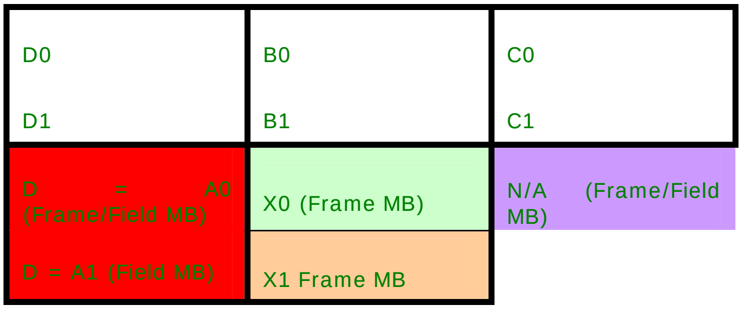


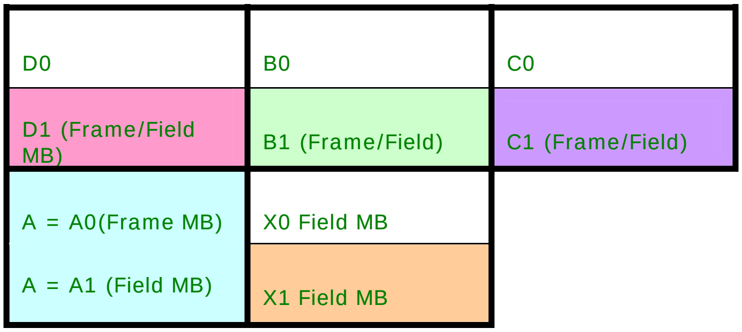
D1(Frame/Field MB)	B1 (Frame/Field MB)	C1 (Frame/Field)

A = E = A0 (Frame/Field MB)
A = E = A1 (Field MB)
	X0 Frame MB
	X1 Frame MB	

D1 (Frame MB)	B1 (Frame MB)	C1 (Frame MB)

A = A0 (Frame/Field MB)
E = A1 (Frame MB)
	X0 Field MB
	X1 Field MB	





The Mb_nb_avail_intra[5:0] is derived from the following information :

The current MB decoding is in MBAFF or not

current_mb_y[0] (=0 for even address, i.e. the top MB; =1 for odd address, i.e. bottom MB of a pair)

It is equivalent to current_mb_is_bottom flag

mb_field_flag of the current MB (0 is a frame MB, 1 is a field MB)

current_mbu_top_avail

The availability of the top MB of the current MB unit (a single MB if non-MBAFF, a MB pair if MBAFF). It is relevant only when the current MB is the bottom MB

It is equal to (last_mb_i | ! constrained_intra_pred_flag)

mb_nb_field_flags[3:0] (the field flags for the 4 neighbor MBs)

mb_nb_field_flags [0] ; left neighbor MB; labeled as A

mb_nb_field_flags [1] : top neighbor MB; labeled as B

mb_nb_field_flags [2] : top right neighbor MB; labeled as C

mb_nb_field_flags [3] : top left neighbor MB; labeled as D

The mb_avail_i[7:0] for the current MB

AVC Spec from Portion of the Table - Specification of mbAddrN and yM

< 0	0 .. maxH - 1	1	1	mbAddrA	1		mbAddrA	yN
					0	yN % 2 = = 0	mbAddrA	yN >> 1
						yN % 2 != 0	mbAddrA + 1	yN >> 1
			0	mbAddrA	1		mbAddrA + 1	yN
					0	yN % 2 = = 0	mbAddrA	(yN + maxH) >> 1
						yN % 2 != 0	mbAddrA + 1	(yN + maxH) >> 1
		0	1	mbAddrA	1	yN < (maxH / 2)	mbAddrA	yN <<1
						yN >= (maxH / 2)	mbAddrA + 1	(yN <<1) - maxH
					0		mbAddrA	yN
			0	mbAddrA	1	yN < (maxH / 2)	mbAddrA	(yN <<1) + 1
						yN >= (maxH / 2)	mbAddrA + 1	(yN <<1) + 1 – maxH
					0		mbAddrA + 1	yN

The following table depicts the generation of mb_nb_avail_intra[5:0] signals in a condensed form. It should note that for most cases only one mb_avail_i[] signal is assigned for each condition, except four places for deriving left neighbor A and E where the neighbor is only available if left neighbor pair (A0 and A1) are both available (A0&A1). Also note that F takes output very similar to that for A except the two "AND" conditions, where F is assigned to A1 instead of (A0&A1).

Table 7-2. Definition of intra-prediction neighbor availability calculation in MBAFF mode

Output ➔		D		B		C		A		E		F**	
Current X \ Neighbor Y		Y-Frame	Y-Field	Y-Frame	Y-Field	Y-Frame	Y-Field	Y-Frame	Y-Field	Y-Frame	Y-Field	Y-Frame	Y-Field
X0 (Top)	X-Frame	D1*	D1*	B1	B1	C1	C1	A0	A0 & A1	A0	A0 & A1	A0	A1
	X-Field	D1*	D0	B1	B0	C1	C0	A0	A0	A1	A0	A0	A0
X1 (Bottom)	X-Frame	A0	A1	X0	N/A	0	0	A1	A0 & A1	A1	A0 & A1	A1	A1
	X-Field	D1	D1	B1	B1	C1	C1	A0	A1	A1	A1	A0	A1

In Table 7-2. Definition of intra-prediction neighbor availability calculation in MBAFF mode, X-Frame or X-Field indicates the frame/field mode of the current MB; and Y-Frame or Y-Field indicates the corresponding neighbor MB for the given neighbor location, being upper left (D) or left

(A) for example. Therefore, “Y-” takes the selected neighbor MB name as in the output cell in the same column. For example, for output D, if X1 is a frame MB, Y = A, if X1 is a field MB, Y = D.

For non-MBAFF mode, as A0=A1, B0=B1, C0=C1 and D0=D1, the neighbor assignment is degenerated into the following simple table. Here, E is assigned to the same as A and F is forced to 0.

Table 7-3. Definition of intra-prediction neighbor availability calculation in non-MBAFF mode

Output ➔	D	B	C	A	E	F
X	D0	B0	C0	A0	A0	0

To further explain the neighbor assignment rules in Table 7-2, the following table provides description for each condition. Please note that this table is informative as it provides redundant information as in the table below.

Table 7-4. Detailed explanation of intra-prediction neighbor availability calculation in MBAFF mode

Current MB	Current MB Field	Neighbor MB Field	Neighbor MB Select (Y=?)	Neighbor Avail Result (OUTPUT)	Description
D					
X0 (Top)	X-Frame	Y-Frame	D	D1*	Top Frame MB uses [-1,-1] = D_31, thus D1 only, regardless D frame or field pair
	X-Frame	Y-Field	D	D1*	
	X-Field	Y-Frame	D	D1*	Top Field MB uses [-1,-2] = D_30, thus if D is frame pair, takes D1 (D1_14 pixel), and if D is field pair, takes D0 (D0_15 pixel)
	X-Field	Y-Field	D	D0	
X1 (Bottom)	X-Frame	Y-Frame	A	A0	Bottom Frame MB uses [-1,15] = A_15, thus A0 (A0_15 pixel) if A is a frame pair, or A1 (A1_7 pixel), if A is a field pair
	X-Frame	Y-Field	A	A1	
	X-Field	Y-Frame	D	D1	Bottom Field MB uses [-1,-1] = D_31, thus D1 only, regardless D frame or field pair
	X-Field	Y-Field	D	D1	
B					
X0 (Top)	X-Frame	Y-Frame	B	B1	Top Frame MB uses [0…15,-1] = B_31, thus B1 only, regardless B frame or field pair
	X-Frame	Y-Field	B	B1	
	X-Field	Y-Frame	B	B1	Top Field MB uses [0…15,-2] = B_30, thus if B is frame pair, takes B1 (B1_14 row), and if B is field pair, takes B0 (B0_15 row)
	X-Field	Y-Field	B	B0	

Current MB	Current MB Field	Neighbor MB Field	Neighbor MB Select (Y=?)	Neighbor Avail Result (OUTPUT)	Description
X1 (Bottom)	X-Frame	Y-Frame	X	X0	Bottom Frame MB uses [0…15,15], thus X0 (X0_15 row)
	X-Frame	Y-Field	X	n/a	Note: X0 and X1 must have the same field type, this row is n/a.
	X-Field	Y-Frame	B	B1	Bottom Field MB uses [0…15,-1] = B_31, thus B1 only, regardless B frame or field pair
	X-Field	Y-Field	B	B1	
				C	
X0 (Top)	X-Frame	Y-Frame	C	C1	Top Frame MB uses [16…23,-1] = C_31, thus C1 only, regardless C frame or field pair
	X-Frame	Y-Field	C	C1	
	X-Field	Y-Frame	C	C1	Top Field MB uses [16…23,-2] = C_30, thus if C is frame pair, takes C1 (C1_14 row), and if C is field pair, takes C0 (C0_15 row)
	X-Field	Y-Field	C	C0	
X1 (Bottom)	X-Frame	Y-Frame	n/a	0	Bottom Frame MB doesn't have left-top neighbor by definition, thus forced to 0
	X-Frame	Y-Field	n/a	0	
	X-Field	Y-Frame	C	C1	Bottom Field MB uses [16…23,-1] = C_31, thus C1 only, regardless C frame or field pair
	X-Field	Y-Field	C	C1	
				A	
X0 (Top)	X-Frame	Y-Frame	A	A0	First Half of Top Frame MB uses [-1,0…7], thus A0 if A is a frame pair; but is only avail if both A0 and A1 are avail if A is a field pair due to the mix
	X-Frame	Y-Field	A	A0&A1	
	X-Field	Y-Frame	A	A0	First Half of Top Field MB uses [-1,0..2..4..14], thus take A0 (if A is frame pair, takes A0 even lines, and if A is field pair, takes A0 first half)
	X-Field	Y-Field	A	A0	
X1 (Bottom)	X-Frame	Y-Frame	A	A1	First Half of Bottom Frame MB uses [-1,16…23], thus A1 if A is a frame pair; but is only avail if both A0 and A1 are avail if A is a field pair due to the mix
	X-Frame	Y-Field	A	A0&A1	
	X-Field	Y-Frame	A	A0	First Half of Bottom Field MB uses [-1,1..3..15], thus take A0 (if A is frame pair, takes A0 odd lines, and if A is field pair, takes A1 first half)
	X-Field	Y-Field	A	A1	

Current MB	Current MB Field	Neighbor MB Field	Neighbor MB Select (Y=?)	Neighbor Avail Result (OUTPUT)	Description
				E	
X0 (Top)	X-Frame	Y-Frame	A	A0	Second Half of Top Frame MB uses [-1,8…15], thus A0 if A is a frame pair; but is only avail if both A0 and A1 are avail if A is a field pair due to the mix
	X-Frame	Y-Field	A	A0&A1	
	X-Field	Y-Frame	A	A1	Second Half of Top Field MB uses [-1,16..18..30], thus take A1 (if A is frame pair, takes A1 even lines, and if A is field pair, takes A0 second half)
	X-Field	Y-Field	A	A0	
X1 (Bottom)	X-Frame	Y-Frame	A	A1	Second Half of Bottom Frame MB uses [-1,24…31], thus A1 if A is a frame pair; but is only avail if both A0 and A1 are avail if A is a field pair due to the mix
	X-Frame	Y-Field	A	A0&A1	
	X-Field	Y-Frame	A	A1	Second Half of Bottom Field MB uses [-1,17..19..31], thus takes A1 (if A is frame pair, takes A1 odd lines, and if A is field pair, takes A1 second half)
	X-Field	Y-Field	A	A1	
				F**	
X0 (Top)	X-Frame	Y-Frame	A	A0	Top Frame MB uses [-1,7] = A_7 (odd location), thus A0 if A is frame pair and A1 if field pair
	X-Frame	Y-Field	A	A1	
	X-Field	Y-Frame	A	A0	Top Field MB uses [-1,14] = A_14 (even location), thus A0 regardless A frame or field pair
	X-Field	Y-Field	A	A0	
X1 (Bottom)	X-Frame	Y-Frame	A	A1	Bottom Frame MB uses [-1,23] = A_23 (odd location), thus A1 regardless A frame or field pair
	X-Frame	Y-Field	A	A1	
	X-Field	Y-Frame	A	A0	Bottom Field MB uses [-1,15] = A_15 (odd location), thus A0 if A is frame pair and A1 if A is field pair
	X-Field	Y-Field	A	A1	

* [DevCTG, DevEL] Erratum: For the cells marked by (*) in both Table 7-2 and Table 7-3, the output from hardware is, incorrectly, as D = A1&D1. So when A1 = 0 and D1 = 1, the result is, again incorrectly, 0 instead of the correct 1.

** [DevCTG, DevEL] Erratum and Implementation Note: For the cells marked by (**) in both Table 7-2 and Table 7-3, instead of following the rule in the above tables, hardware actually assigns F = Is_Left_MB_Field & Is_Left_Bottom_MB_Intra. Kernels will need to check this bit (if it is 1) only when all following conditions are met:

1. The picture is an MBAFF Frame picture;

2. Current MB is frame MB with intra_8x8 prediction mode;

3. Both bits A and E are “0”.

It is only used to determine the reference pixel pre-filter operation for the left bottom sub-macroblock for intra_8x8 prediction as explained by the following.

When current MB is a frame MB in an MBAFF frame picture, both bits A and E in MbIntraStruct field in in-line data DWORD 3 are set to be the same. If the left MB pair is a field MB pair and the top field is an inter MB and the bottom field is an intra MB and the constrained_intra_pred_flag is true, both A and E bits will be set to “0” meaning the left MB pair is not available for intra prediction.

The issue occurs when current MB has intra_8x8 prediction mode. The reference pixel pre-filtering process of the sub-macroblock #2 will access the reference pixel at [-1, 7] relative location which is in its left bottom MB. Since the left bottom MB is an intra MB, the pixel [-1, 7] is available and therefore should be used in the pre-filtering. However, this information is currently missing from A and E. Therefore, F bit as provided by hardware can be used to indicate if pixel [-1,7] is available.

7.7.1.7 MC Kernel Interface Descriptor Table

A set of Kernels are developed to cover all the AVC motion compensated interpolation for reconstructing the display picture. These kernels are loaded into the main memory (by driver) and are referenced by the corresponding interface descriptor. The interface descriptors collectively form an array/table of entries. To decouple the direct referencing of kernels inside the BSD unit, a remapping table is used. The remapping table is indexed by the modified Mb_type. Its content are the pointers to the interface descriptor table.

Two additional flags are derived as follows :

MB Subpartition Exist

If at least one 8x8 block within a MB has a subpartition (4x4, 4x8 or 8x4, as indicated by the Sub_mb_type SE), this flag is set to 1; otherwise it is set to 0

MB BiPred Exist

If at least one 8x8 block within a MB is marked with BiPred (as indicated by the Sub_Pred_mode SE), this flag is set to 1; otherwise it si set to 0.

Modified MbType	AVC MbType						Remapping Table Entry Index
	I Slice	P Slice		B Slice			
	Mb_type SE	Mb_type SE	Sub_mb_type SE	Mb_type SE	MB Subpartition Exist ? (Sub_mb_type SE)	MB BiPred Exist ? (Sub_Pred_modeSE)	
intra_16x16	1-24	6-29	n/a	24-47	n/a	n/a	0
intra_8x8	0 / txfomsize8x8=1	5 / txfomsize8x8=	n/a	23 / txfomsize8	n/a	n/a	1

		1		x8=1			
intra_4x4	0 / txformsize8x8=0	5 / txformsize8x8 =0	n/a	23 / txformsize8 x8=0	n/a	n/a	2
intra_pcm	25	30	n/a	48	n/a	n/a	3
L0_L1_16x16	n/a	0, P_skip	n/a	1, 2	n/a	n/a	4
L0_L1_16x8	n/a	1	n/a	4, 6, 8, 10	n/a	n/a	5
L0_L1_8x16	n/a	2	n/a	5, 7, 9, 11	n/a	n/a	6
L1_L0 8x8_nosub	n/a	3,4	0	0, 22, B_skip	0 (0*,inferred*, 1, 2)	0 (Direct*, Pred_L0, Pred_L1)	7
L0_L1_8x8_sub	n/a	3, 4	1, 2, 3	0, 22, B_skip	1 (0*,inferred*, 4, 5, 6, 7, 10, 11)	0 (Direct*, Pred_L0, Pred_L1)	8
Undefined	n/a	n/a	n/a	n/a	n/a	n/a	9
BI_16x16	n/a	n/a	n/a	3	n/a	n/a	10
BI_16x8	n/a	n/a	n/a	12, 14, 16, 18, 20	n/a	n/a	11
BI_8x16	n/a	n/a	n/a	13, 15, 17, 19, 21	n/a	n/a	12
BI_8x8_nosub	n/a	n/a	n/a	0, 22, B_skip	0 (0*, inferred*, 3)	1 (Direct*, BiPred)	13
BI_8x8_sub	n/a	n/a	n/a	0, 22, B_skip	1 (0*, inferred*, 8, 9, 12)	1 (Direct*, BiPred)	14
Undefined	n/a	n/a	n/a	n/a	n/a	n/a	15

For B MB, in the case of the L1_L0_8x8_nosub (i.e. either L1 or L0 Pred mode for an 8x8 block with no sub-partitioning (i.e. no 4x8, 8x4, or 4x4)), there are 3 possible Mb_type syntax elements (SE), but each may support a different set of sub_mb_type SE and sub_Pred_mode, depending on the direct_8x8_inference_flag and direct mode :

Mb_type SE = 0 (B_Direct_16x16) and B MB,

Internal mapped into Mb_type SE = 22 and sub_mb_type[mbPartIdx]=0 (B_Direct_8x8) with direct_8x8_inference_flag == 1, spatial direct mode, and only predFlagL0 or predFLagL1 (described in the AVC spec) is set but not both

PredL0 or PredL1 8x8 only

Mb_type SE = 22 (B_8x8) and B MB,

sub_mb_type[mbPartIdx]=0 (B_Direct_8x8) with

direct_8x8_inference_flag == 1, spatial direct mode, and only predFlagL0 or predFLagL1 (described in the AVC spec) is set but not both

PredL0 or PredL1 8x8 only

sub_mb_type[mbPartIdx]=1 (B_L0_8x8), =2 (B_L1_8x8)

Mb_type = inferred B_skip

Always mapped internally to Mb_type SE = 0 (B_Direct_16x16) , and treated with sub_mb_type[mbPartIdx]=0 (B_Direct_8x8)

For B MB, in the case of the L1_L0_8x8_sub (i.e. either L1 or L0 Pred mode for an 8x8 block with sub-partitioning (i.e. 4x8, 8x4, or 4x4)), there are 3 possible Mb_type syntax elements (SE), but each may support a different set of sub_mb_type SE and sub_Pred_mode, depending on the direct_8x8_inference_flag and direct mode :

Mb_type SE = 0 (B_Direct_16x16) and B MB,

Internal mapped into Mb_type SE = 22 and sub_mb_type[mbPartIdx]=0 (B_Direct_8x8) with

direct_8x8_inference_flag == 0, spatial direct mode, and only predFlagL0 or predFLagL1 (described in the AVC spec) is set but not both

PredL0 or PredL1 4x4 only

Mb_type SE = 22 (B_8x8) and B MB,

sub_mb_type[mbPartIdx]=0 (B_Direct_8x8) with

direct_8x8_inference_flag == 0, spatial direct mode, and only predFlagL0 or predFLagL1 (described in the AVC spec) is set but not both

PredL0 or PredL1 4x4 only

sub_mb_type[mbPartIdx]=4,5,6,7,10, and 11 (4x8, 8x4, 4x4 with PredL0 or PredL1)

Mb_type = inferred B_skip

Always mapped internally to Mb_type SE = 0 (B_Direct_16x16) , and treated with sub_mb_type[mbPartIdx]=0 (B_Direct_8x8)

For B MB, in the case of the BI_8x8_nosub (i.e. both L1 and L0 Pred modes are in use for an 8x8 block with no sub-partitioning (i.e. no 4x8, 8x4, or 4x4)), there are 3 possible Mb_type syntax elements (SE), but each may support a different set of sub_mb_type SE and sub_Pred_mode, depending on the direct_8x8_inference_flag and direct mode :

Mb_type SE = 0 (B_Direct_16x16) and B MB, Internal mapped into Mb_type SE = 22 and sub_mb_type[mbPartIdx]=0 (B_Direct_8x8) with

direct_8x8_inference_flag == 1, both direct modes (spatial and temporal) with both predFlagL0 and predFLagL1 (described in the AVC spec) are set

BiPred 8x8 only

Mb_type SE = 22 (B_8x8) and B MB,

sub_mb_type[mbPartIdx]=0 (B_Direct_8x8) with

direct_8x8_inference_flag == 1, both direct modes (spatial and temporal) with both predFlagL0 and predFLagL1 (described in the AVC spec) are set

BiPred 8x8 only

sub_mb_type[mbPartIdx]=3 (B_Bi_8x8)

Mb_type = inferred B_skip

Always mapped internally to Mb_type SE = 0 (B_Direct_16x16) , and treated with sub_mb_type[mbPartIdx]=0 (B_Direct_8x8)

For B MB, in the case of the BI_8x8_sub (i.e. both L1 or L0 Pred modes are in use for an 8x8 block with sub-partitioning (i.e. 4x8, 8x4, or 4x4)), there are 3 possible Mb_type syntax elements (SE), but each may support a different set of sub_mb_type SE and sub_Pred_mode, depending on the direct_8x8_inference_flag and direct mode :

Mb_type SE = 0 (B_Direct_16x16) and B MB,

Internal mapped into Mb_type SE = 22 and sub_mb_type[mbPartIdx]=0 (B_Direct_8x8) with

direct_8x8_inference_flag == 0, both direct modes (spatial and temporal) with both predFlagL0 and predFLagL1 (described in the AVC spec) are set

BiPred 4x4 only

Mb_type SE = 22 (B_8x8) and B MB,

sub_mb_type[mbPartIdx]=0 (B_Direct_8x8) with

direct_8x8_inference_flag == 0, both direct modes (spatial and temporal) with both predFlagL0 and predFLagL1 (described in the AVC spec) are set

BiPred 4x4 only

sub_mb_type[mbPartIdx]=8, 9, 12 (4x8, 8x4, 4x4 with BiPred)

Mb_type = inferred B_skip

Always mapped internally to Mb_type SE = 0 (B_Direct_16x16) , and treated with sub_mb_type[mbPartIdx]=0 (B_Direct_8x8)

Internally, after MV and Ref_Idx are generated for the direct modes (B MBs), the remaining processing and MB reconstruction are identical to the regular MB with MVinfo obtained from the bitstream. That is, all “Direct” predmode are internally remapped into PredL0, PredL1 or BiPred.

Although the “Remapping Table Entry Index” ranges from 0 to 15, the number of actual kernels may be lesser. That is, mulitple Remapping Indices may refer to the same kernel.

7.7.2 Special Scan order for MB / MB Pair Kernel Processing

MBAFF mode with special_scan_order = 1, we will be spawning MBs in this order: (for IT/MC – top of the pairs in the same wave front first and the bottom of the pairs)

	0	1	2	3	4
0	0	2	4	8	12
1	1	3	6	10	15
2	5	9	13	18	22
3	7	11	16	20	24
4	14	19	23	26	28
5	17	21	25	27	29

(Performance Based Spawning)

And ILDB, the special scan order, do a pair at a time. The kernel is simpler when the edge control data are layout in this order. The pair can be split in processing if they are not depending on each other.

	0	1	2	3	4
0	0	2	4	8	12
1	1	3	5	9	13
2	6	10	14	18	22
3	7	11	15	19	23
4	16	20	24	26	28
5	17	21	25	27	29

(MBs spawned in pairs)

7.8 Error Handling and Error Concealment

Design objective is to keep the playback continue going forward in time even when errors are encountered (detected or not). Hence, the system will reconstruct and display the new picture as much as possible, without dropping the current picture nor repeating the previous decoded picture.

With the limitation of the current architecture and H/W design, error handling is performed only at the Slice boundary, and in between pictures.

To catch the case when the AVC_BSD has finished decoding the current picture but the last MB is not reached, a Phantom Slice is required to conceal such error condition. Therefore, it is required to submit an additional AVC_BSD_OBJECT command for the Phantom Slice at the end of each picture. A Phantom Slice is one that has the indirect_data_length set to 0 and Slice_Start_Mb_Num set to the total number of MB in the current picture + 1.

7.8.1 Design Assumptions

Operating Assumptions :

The BSD unit is either actively decoding a picture, or in between pictures when errors occur.

At the beginning of decoding a new picture (the first slice of a picture), the current MB location registers are reset to 0 by the H/W.

The end of a picture is tracked by comparing the current MB location (just being decoded) to the the picture size parameters (in MB unit) in the BSD unit. The BSD unit will never attempt to decode beyond the picture size programmed, even when there are data remained at the BSD input.

The end of a picture sequence is only known in the driver. BSD unit will continue operating until no more decoding request commands are injected or when there is an optional time-out by the driver.

BSD unit has no notion of picture structure, it only deals with Slice as instructed and directed by the State Commands. It does not know it is decoding the last slice of a picture (until it starts decoding the next picture).

There is no H/W Start_Code_Detector at the front end of the BSD unit, so the end of Slice signal can only be detected and generated internally.

7.8.2 Error Concealment and Recovery Strategies

Depending on the category and position of error being detected, a concealment strategy will be adpoted.

Picture Repeat

Handled by the display controller – when it does not receive a new picture to display at the appropriate time period

H/W MB Fillings (as described below)

S/W Re-sync at the next Slice or picture boundary

Dropping a Slice, or

The driver can ignore the rest of the bitstream until the start of the next NAL is detected. That is to start decoding again at slice boundary

dropping a picture entirely.

The driver can decide to ignore all subsequent slices until a new IDR has arrived, and to start decoding again from there on.

7.8.3 Error Handling

7.8.3.1 S/W Driver

Provide high level error handling capabilities – interrupts and/or polling. Perform error handling and recovery in between Slices and Pictures boundaries.

It also checks for the bitstream syntax errors at and above Slice Header Layers, before proceeding with the Slice decoding.

Currently there is no communication of error handling at the application level, which will require API definitions to support and to enable it.

There are also two error status bits generated in the H/W that the S/W can poll regularly :

Force_In_Progress

A H/W Status Bit (read-only) to indicate filling missing MBs is in progress

Error_In_Slice

A H/W Status Bit (read-only) to indicate an error has detected in the HW, including VLC Decoding Error, BSD Decoded Data Error and MPR Error

7.8.3.2 H/W MB Filling

This is the only errror handling mechanism built-in the BSD unit. It will automatically perform error detection and error concealment in the Slice Data Layer at the Slice boundary. There is no direct S/W control.

When decoding a Slice (contains multiple MBs), internally the BSD unit will keep track of the number of MBs being decoded so far. This number also represents the position of the corresponding MB in a picture, assuming a raster-scan MB order (top to bottom, left to right). Hence, the Current_MB_Number register is maintained and updated in the H/W automatically.

For each Slice Data, there is also a Start_MB_Number presented at the corresponding Slice Header. This number specifies the picture location of the very first MB decoded from this Slice Data.

Just before start decoding the current new Slice Data, the last value of the Current_MB_Number register (holding the last MB position of the previously decoded Slice) is compared against the Start_MB_Number. If there is a match, the decoding of current new Slice continues. Otherwise, error concealment is immediately triggered.

There are 2 errorneous cases :

Current_MB_Number > Start_MB_Number

Current_MB_Number will force to equal to Start_MB_Number, and decoding continues at the newly-adjusted Current_MB_Number location.

If Current_MB_Number has reached the end of the picture, any data remains in the input must be discarded.

Current_MB_Number < Start_MB_Number

The gap is automatically filled, so that the decoding can continue at the after-filled Current_MB_Number location

The starting position for MB filling is adjustable, and is equal to Current_MB_Number – MIN{ Decoded_MB_Count_Prev, [(MBAFF 2:1) * Force_Skip_Rewind] }

The MBs used to fill the gap are derived as follows :

Intra Concealment Mode

If there are no pictures in the DPB, the missing MBs are concealed as if they had been Intra_16x16_2_0_0 and Intra_Chroma_DC MBs

Inter Concealment Mode

The missing MBs are concealed by copying co-located MBs from the frame store specified as the error concealment frame store during the slice start command. By default the top of the list_0 is used.

If the current premature completed slice is the last slice of a picture, SW will need to inject a Phantom Slice to cause the HW to fill the missing MBs to the end of the picture

A Phantom Slice with Start_MB_NUMBER set beyond or equal to the picture size

Decoding stops when the picture size has reached.

For example : a picture contains 3 Slices

Slice 1 – contains 100 MBs

Slice 2 – contains 200 MBs

Slice 3 – contains 50 MBs

After decoded the Slice Data of Slice 1, the Current_MB_Number is rest at 80, When start to decode Slice 2, its Start_MB_Number is set at 100. Hence, the H/W will automatically fill-in the missing MBs from 81 to 99. Then Slice 2 will continue its decoding from location 100.

Errorneous MB decoding will never pass to the next picture. Since, internally the BSD unit has stored the size of the picture (in MB unit), and it will stop further processing (error fillings, or MB decoding) once this boundary value has reached.

Refer to later section on supporting multiple stream decoding, in which the Current_MB_Number register can be programmed from the driver to allow additional flexibility in error handling.

7.8.3.3 Error Statistics

Another high level error handling mechanism is to keep track of

the error count in decoding each picture of a video sequence, and/or

the number of MBs being concealed in a picture

Depending on the allowable thresholds, if the error counts are within tolerances, nothing will be done. Otherwise, actions like – re-establishing the video source linkage, can be taken, or choosing a different error handling processing.

7.8.3.4 WatchDog Timers

At the driver level an additional error detection mechanism is needed. In the situation where errors have stalled the BSD unit from progressing, there must be a way to tell that the H/W is hung. Watch-dog timers are preset with some known timing constraints that are related to the decoding, e.g. an expected decoding time of a slice or a picture. Once the threshold has reached, a stall condition is assumed, and action will take place to reset the BSD unit, and up to the driver to decide where to start decoding again or simply abort the decoding processing.

7.8.4 Error Handling for Non-Existing Reference Picture

There are 3 options :

For B-MB/Partition Direct Prediction Modes :

Spatial Direct Mode

If co-located MB reference picture does not exist, zero-motion vectors are assumed.

Temporal Direct Mode

If either co-located MB reference picture does not exist, zero-motion vectors are assumed.

Let the S/W to provide an appropriate replacement/alternative frame store ID

Each Reference Picture is addressed by a byte in the MPR List

If a list entry reference a picture that does not exist, the value written for that entry should flag it as non-existing

Bit 7 is the Non-Existing Flag, Bit4:0 will provide the alternate Frame Store Index

Follow the SEI Recovery Point Conformance (not implement in this version, since we have taken out the Greylevel Non-Existing Reference flag)

The recovery point SEI message assists a decoder

After decoder initiates random access, or after the encoder indicates a broken link in the sequence

Decoder has the option for Inter-Prediction to use mid-level grey for all references to “not available” reference pictures (instead of selecting an alternative frame store). That is references to pictures containing only intra macroblocks and having sample values given by Y equal to 128, Cb equal to 128, and Cr equal to 128 (mid-level grey)

7.9 Concurrent, Multiple Video Stream Decoding Support

The natural place for switching across multiple streams is at the Slice boundary. Each Slice is a self-sustained unit of compressed video data and has no dependency with its neighbors (except for the Deblocking process). In addition, there is no interruptability within a Slice. However, when ILDB is invoked, the processing of some MBs will require neighbour MB information that cross the Slice boundary. Hence, to limit the buffering requirement, in this version of H/W design, the stream switching can only be performed at the picture boundary instead.

7.10 Performance and Latency Estimation

7.11 Media Messages

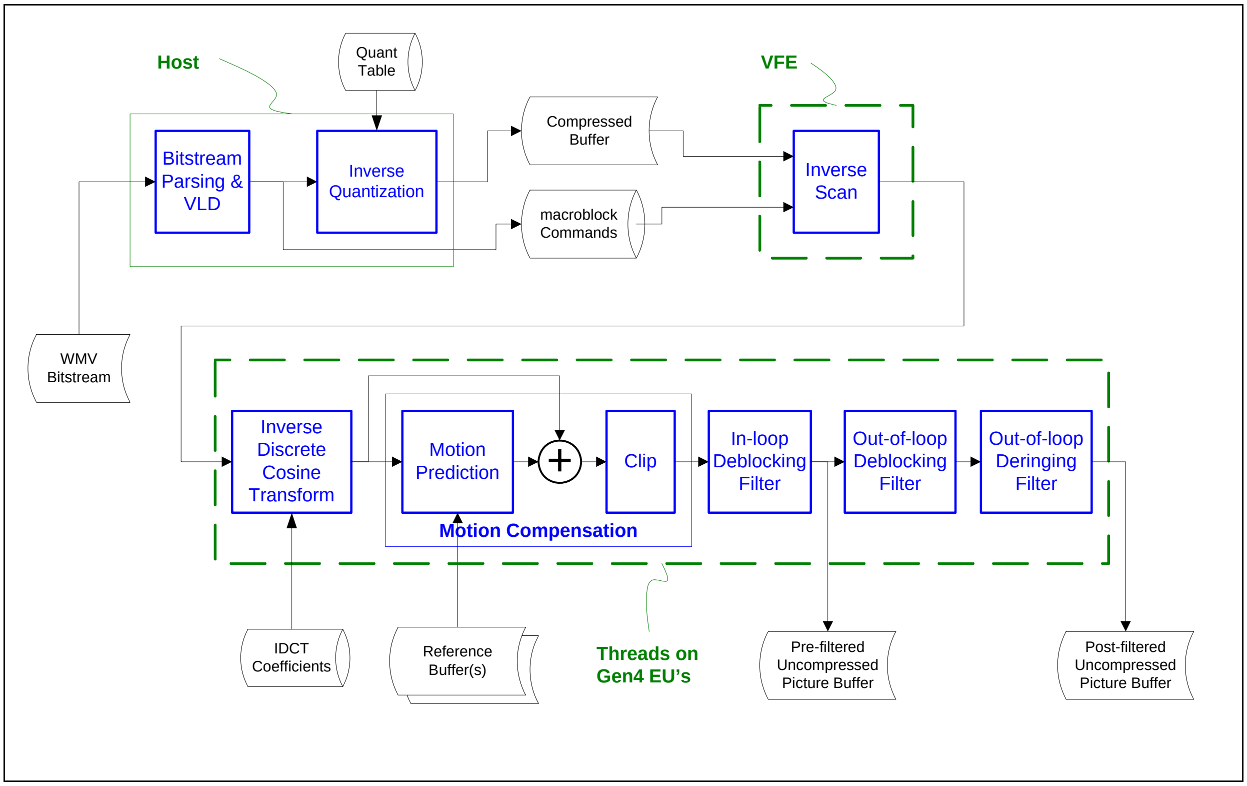
Since this is a standalone bit stream co-processor, no Media Messaging is implemented. It does not have threading capability to the EU Subsystem.

7.12 Media Applications with Specific Hardware Support

7.12.1 AVC – Off-host CABAC/CAVLD Acceleration

Figure 7-3 shows the complete AVC video decode pipeline. This partition can be referred to as the off-host CABAC/CAVLD partition as the graphics device is responsible of performance inverse-scan, IDCT, motion compensation and the rest of the post filter operations.

Figure 7-3. AVC decoding pipeline with off-host CABAC/CAVLD acceleration


Host
Quant Table
VFE
Bitstream Parsing & VLD
Inverse Quantization
Compressed Buffer
macroblock Commands
Inverse Scan
WMV Bitstream
Inverse Discrete Cosine Transform
Motion Prediction
+
Clip
In-loop Deblocking Filter
Out-of-loop Deblocking Filter
Out-of-loop Deringing Filter
Motion Compensation
IDCT Coefficients
Reference Buffer(s)
Threads on Gen4 EU's
Pre-filtered Uncompressed Picture Buffer
Post-filtered Uncompressed Picture Buffer