Intel® Open Source HD Graphics and Intel Iris™ Plus Graphics

Programmer's Reference Manual

For the 2016 - 2017 Intel Core™ Processors, Celeron™ Processors, and Pentium™ Processors based on the "Kaby Lake" Platform

Volume 9: Media Video Enhancement (VEBOX) Engine

January 2017, Revision 1.0

Creative Commons License

You are free to Share - to copy, distribute, display, and perform the work under the following conditions:

- **Attribution.** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **No Derivative Works.** You may not alter, transform, or build upon this work.

Notices and Disclaimers

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.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

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 information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

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

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

Table of Contents

Media VEBOX Introduction

The VEBOX is an independent pipe with a variety of image enhancement functions.

The following sections are contained in Media VEBOX:

Feature
Denoise
Deinterlacer
VEBOX Output Statistics
VEBOX State

Denoise

This section discusses the Denoise feature in the chipset.

- **Denoise Filter** - detects noise in the input image and filters the image with either a temporal filter or a spatial filter. The temporal filter is applied when low motion is detected.
- **Chroma Denoise Filter** - detects noise in the U and V planes separately and applies a temporal filter.
- **Block Noise Estimate (BNE)** - as part of the Global Noise Estimate (GNE) algorithm, BNE estimates the noise over each block of pixels in the input picture.
- **Global Noise Estimate (GNE)** - GNE is calculated at the end of the frame by combining all the BNEs. The final GNE value is used to control the denoise filter for the next input frame. Noise estimates are kept between frames and blended together.

Motion Detection and Noise History Update

This logic detects motion of the current block for the denoise filter. The detected motion is then combined with motion detected in the co-located block of the past frame, and stored in the denoise history table.

Block Noise Estimate (Part of Global Noise Estimate)

The BNE estimates the amount of noise in each rectangular region of the input picture. The BNE estimate is computed separately for each color component in the input picture. The estimates from BNE are summed together to generate the Global Noise Estimate (GNE) for the entire input picture.

Temporal Filter

Temporal denoise is applied to each pixel based on the noise strength measured from the input pictures. Each pair of co-located pixels in the current and previous input pictures is blended together to generated the output pixel.

Context Adaptive Spatial Filter

For each pixel in the local neighborhood, its luma value is compared to the center pixel to be filtered.

Each pixel in the neighborhood for which the difference is less than good_neighbor_th is marked as a "good neighbor". The filtered output pixel is then equal to average of “good neighbor” pixels.

Chroma Noise Detection

The operation of chroma noise detection module is similar to the luma noise detection module. The U & V channels are processed individually to generate a noise estimate for each of the 2 channels.

Chroma Noise Reduction Filter

A simple and effective temporal-domain chroma noise reduction filter is applied. The Noise History is updated based on the motion detection result as before. The Noise History value is used to control the temporal denoise filter.

Denoise Blend

The denoise blend combines the temporal and spatial denoise outputs.

Deinterlacer

The deinterlacer (DI) takes the top and bottom fields of each input frame and converts them into two output frames. This block also gathers statistics from a film mode detector at the end of the frame. If the film mode detector determines that the input is progressive, then the input fields are put together directly to construct the progressive output frame.

Features:

- **Deinterlacer** - DI starts by estimating how much motion is present across the input fields. Low motion scenes are reconstructed by temporal deinterlacer, while high motion scenes are reconstructed by spatial deinterlacer.
- **Film Mode Detection** - This detector determines if the input fields are created by sampling film content and converting it to interlaced video. If so, the deinterlacer is turned off in favor of reconstructing the progressive output frame directly from adjacent input fields in the Progressive Cadence Reconstruction mode. Various sum-of-absolute differences are computed per block.
- **Chroma Upsampling** - If the input is 4:2:0, then chroma data is doubled vertically to 4:2:2. Chroma data is then processed by chroma deinterlacer or progressive cadence reconstruction.

VEBOX Output Statistics

Overall Surface Format

Statistics are gathered on both per-block (16x4) basis and per-frame basis. There are 16 bytes of encoder statistics data per 16x4 block, plus a variety of per frame data which are stored in a linear surface. The 16 bytes of encoder statistics per block are output if either DN or DI are enabled and are organized into a surface with a pitch equal to the output surface width rounded to the next higher 64 boundary (so that each line starts and ends on a cache line boundary). The height of the surface is ¼ of the height of the output surface. If both DN and DI are disabled then the encoder stats are not output and the per frame information is output at the base address. The encoder statistics can be disabled with a state variable to lower the output bandwidth. The offsets for the other statistics do not change.

The per frame information is written twice per frame to allow for a 2 slice solution - in a single slice the second set of data will be all 0. The final per frame information is found by adding each individual Dword, clamping the data to prevent it from overflowing the Dword.

The Deinterlacer outputs two frames for each input frame. When IECP is applied to the Deinterlacer output, separate GCC and STD per frame statistics are created for each output frame. In this case, 4 copies of the per frame information are written - two copies for the two slice solution times two output frames. For the case of DN and no DI, only the first set of per frame statistics will be written.

Statistics Surface when DI Enabled and DN either On or Off

16 Bytes for X=0, Y=0	16 Bytes for X=16, Y=0	16 Bytes for X=32, Y=0	....	16 Bytes for X=Width-16, Y=0	
16 Bytes for X=0, Y=4					
16 Bytes for X=0, Y=8					
.....					
16 Bytes for X=0, Y=Height-4					
17 DWs Reserved		2DW of STD (Previous Slice 0)	2DW of GCC (Previous Slice 0)	11 DWs of Reserved	
11 DWs FMD (Slice 0)	6 DWs of GNE (Slice 0)	2DW of STD (Current Slice 0)	2DW of GCC (Current Slice 0)	2DWs of Auto focus (Current Slice 0)	9 DWs of Reserved
17 DWs Reserved		2DW of STD (Previous Slice 1)	2DW of GCC (Previous Slice 1)	11 DWs of Reserved	
11 DWs FMD (Slice 1)	6 DWs of GNE (Slice 1)	2DW of STD (Current Slice 1)	2DW of GCC (Current Slice 1)	2DWs of Auto focus (Current Slice 1)	9 DWs of Reserved

Statistics Surface when DN Enabled and DI Disabled

16 Bytes for X=0, Y=0	16 Bytes for X=16, Y=0	16 Bytes for X=32, Y=0	....	16 Bytes for X=Width-16, Y=0	
16 Bytes for X=0, Y=4					
16 Bytes for X=0, Y=8					
.....					
16 Bytes for X=0, Y=Height-4					
11 DWs White Balance Stats (Slice 0)	6 DWs of GNE (Slice 0)	2DW of STD (Slice 0)	2DW of GCC (Slice 0)	2DWs of Auto focus (Slice 0)	9 DWs of Reserved
11 DWs White Balance Stats (Slice 1)	6 DWs of GNE (Slice 1)	2DW of STD (Slice 1)	2DW of GCC (Slice 1)	2DWs of Auto focus (Slice 1)	9 DWs of Reserved

Statistics Surface when both DN and DI Disabled

11 DWs White Balance Stats (Slice 0)	6 DWs of GNE (Slice 0)	2DW of STD (Slice 0)	2DW of GCC (Slice 0)	2DWs of Auto focus (Slice 0)	9 DWs of Reserved
11 DWs White Balance Stats (Slice 1)	6 DWs of GNE (Slice 1)	2DW of STD (Slice 1)	2DW of GCC (Slice 1)	2DWs of Auto focus (Slice 1)	9 DWs of Reserved

When DN and DI are both disabled, only the per frame statistics are written to the output at the base address.

Statistics Offsets

The statistics have different offsets from the base address depending on what is enabled.

The encoder statistics size is based on the frame size:

```
Encoder_size = width * (height+3)/4
```

where

width is the width of the output surface rounded to the next higher 64 boundary and

height is the output surface height in pixels.

Offset	DI on	DI off + DN on	DI off + DN off
Per_Command_Previous_Slice0	Encoder_size	N/A	N/A
Per_Command_Current_Slice0	Encoder_size + 0x80	Encoder_size + 0x00	0x00
Per_Command_Previous_Slice1	Encoder_size + 0x100	N/A	N/A
Per_Command_Current_Slice1	Encoder_size + 0x180	Encoder_size + 0x80	0x80

LACE and ACE

The ACE Histograms are in a separate surface with the LACE Histograms and RGB Histograms.

For simplicity the offsets are the same in all modes - if a surface is disabled then it simply is not written, but the following surfaces offsets are not changed.

The starting offsets from the beginning of LACE/ACE/RGB Histogram Output are:

Offset	DI on	RGB on	DI on + RGB on
Red Histogram_Slice0	N/A	0x0000	0x0000
Green_Histogram_Slice0	N/A	0x0400	0x0400
Blue_Histogram_Slice0	N/A	0x0800	0x0800
Red Histogram_Slice1	N/A	0x0C00	0x0C00
Green_Histogram_Slice1	N/A	0x1000	0x1000
Blue_Histogram_Slice1	N/A	0x1400	0x1400
ACE_Histo_Previous_Slice0	0x1800	N/A	0x1800
ACE _Histo_Current_Slice0	0x1C00	0x1C00	0x1C00

Offset	DI on	RGB on	DI on + RGB on
ACE_Histo_Previous_Slice1	0x2000	N/A	0x2000
ACE_Histo_Current_Slice1	0x2400	0x2400	0x2400
LACE Histogram Current	0x1800	0x1800	0x1800
LACE Histogram Previous	0x1800+LACE_hist_size	N/A	0x1800+LACE_hist_size

Per Command Statistics

The Per Command Statistics are placed after the encoder statistics if either DN or DI is enabled. If the frame is split into multiple calls to the VEBOX, each call outputs only the statistics gathered during that call and software provides different base address per call and sums the resulting output to compute the per-frame data.

The final address of each statistic is:

> **Statistics Output Address** + **Per_Command_Offset** (pick the one for the slice desired and the current/previous frame for Deinterlacer) + **PerStatOffset**

FMD Variances and GNE Statistics

These are the 11 FMD variances (Variance 0 ~ 10) and Global Noise Estimate Statistics (Sums and Counts) collected in each VEBOX call.

Note that pixel values in blocks that are close to the edge of the frame (within a 16x4 block that intersects or touches the frame edge) are not used in the variance computation.

FMD variances are 0 when the Deinterlacer is disabled.

GNE estimates are 0 when the Denoise is disabled.

Counter Id	PerStatOffset	Associated Counter
0	0x00	FMD Variance 0
1	0x04	FMD Variance 1
2	0x08	FMD Variance 2
3	0x0C	FMD Variance 3
4	0x10	FMD Variance 4
5	0x14	FMD Variance 5
6	0x18	FMD Variance 6
7	0x1C	FMD Variance 7
8	0x20	FMD Variance 8
9	0x24	FMD Variance 9
10	0x28	FMD Variance 10
11	0x2C	GNE Sum Luma (Sum of BNEs for all passing blocks)
12	0x30	GNE Sum Chroma U

Counter Id	PerStatOffset	Associated Counter
13	0x34	GNE Sum Chroma V
14	0x38	GNE Count Luma (Count of number of block in GNE sum)
15	0x3C	GNE Count Chroma U
16	0x40	GNE Count Chroma V

Skin-Tone Data

The register Ymax stores the largest luma value. It is reset at the start of a command to zero.

The register Ymin stores the smallest luma value. It is reset at the start of a command to:

Reset Value
0xFFFF (65535 in 16 bits)

When comparing the luma values to Ymax or Ymin,

Feature
16-bit values are used: For 16bpc luma pixels: the entire 16 bits are used. For 8bpc luma pixels: 8 zeros are appended to the LSB.

There is also a 29-bit counter of all the skin pixels (Number of Skin Pixles).

Register values are 0 if the STD/STE function is disabled.

The registers are stored with the offsets as shown below.

PerStatOffset	Associated Register
0x044	Ymax (bits 31:16), Ymin (bits 15:0)
0x048	Number of Skin Pixels (bits [28:0], other bits zero)

Gamut Compression: Out of Range Pixels

The statistics gathered for Gamut Compression are:

1. Count of out-of-range pixels (29 bits) and
2. Sum of the distances from out-of-range pixels to the closest range boundaries (32 bits).

If the sum is greater than the maximum 32-bit value, then it is clamped to the maximum 0xFFFFFFFF.

Both values are reset to zero at the start of each command.

Both values are zero if the GCC function is disabled.

PerStatOffset	Associated Register
0x04C	Sum of distances of out-of-range pixels (clamped to 0xFFFFFFFF)
0x050	Number of out-of-range pixels (bits [28:0], other bits are zero)

Histograms

The histograms are in separate surfaces called the ACE/LACE and RGB Histogram.

Ace Histogram

The Ace Histogram counts the number of pixels at different luma values.

It has 256 bins, each of which is 24 bits.

Any count that exceeds 24 bits is clamped to the maximum value.

The data is stored on DWord boundaries with the upper 8 bits equal to zero.

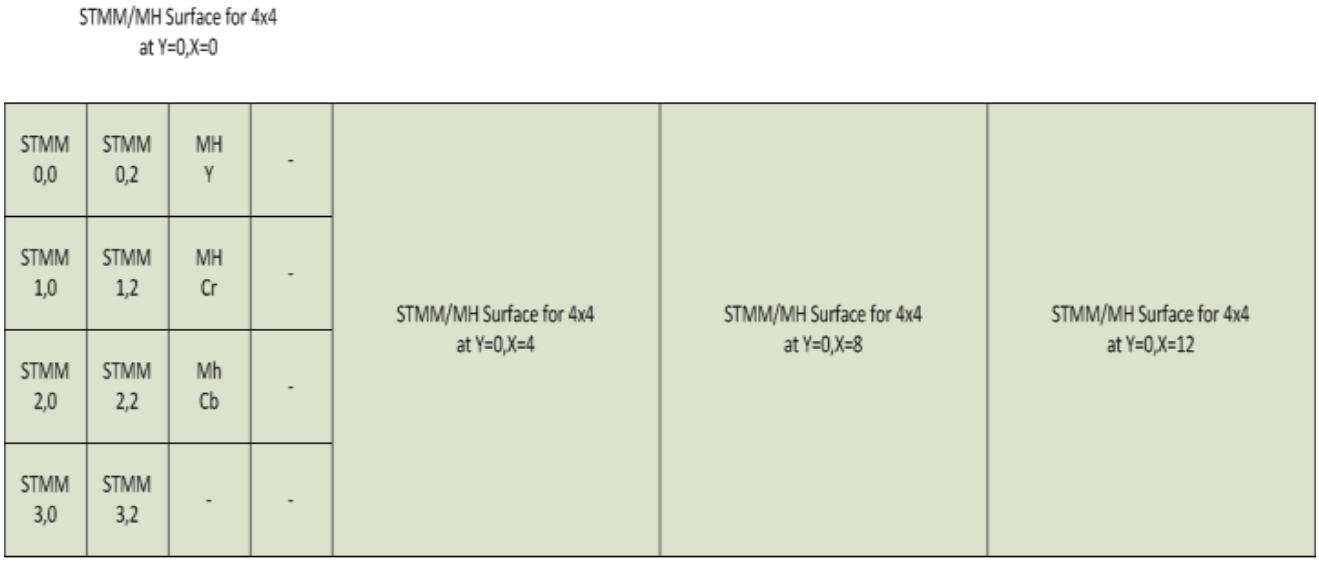
The Ace Histogram in non-capture mode is created by summing two independent 24-bit histograms, each of which is clamped to 24 bits. This means that the sum can reach 25 bits when both independent histograms are clamped or near clamped.

Y[9:2]	PerStatOffset	Associated Counter
0	0x000	ACE histogram, bin 0
1	0x004	ACE histogram, bin 1
2	0x008	ACE histogram, bin 2
...	...	...
255	0x3fc	ACE histogram, bin 255

STMM / Denoise

The STMM/Denoise history is a custom surface used for both input and output. The previous frame information is read in for the DN (Denoise History) and DI (STMM) algorithms; while the current frame information is written out for the next frame.

STMM / Denoise Motion History Cache Line



Byte	Data
0	STMM for 2 luma values at luma Y=0, X=0 to 1
1	STMM for 2 luma values at luma Y=0, X=2 to 3
2	Luma Denoise History for 4x4 at 0,0
3	Not Used
4-5	STMM for luma from X=4 to 7
6	Luma Denoise History for 4x4 at 0,4
7	Not Used
8-15	Repeat for 4x4s at 0,8 and 0,12
16	STMM for 2 luma values at luma Y=1,X=0 to 1
17	STMM for 2 luma values at luma Y=1, X=2 to 3
18	U Chroma Denoise History
19	Not Used
20-31	Repeat for 3 4x4s at 1,4, 1,8 and 1,12
32	STMM for 2 luma values at luma Y=2,X=0 to 1
33	STMM for 2 luma values at luma Y=2, X=2 to 3
34	V Chroma Denoise History
35	Not Used

Byte	Data
36-47	Repeat for 3 4x4s at 2,4, 2,8 and 2,12
48	STMM for 2 luma values at luma Y=3,X=0 to 1
49	STMM for 2 luma values at luma Y=3, X=2 to 3
50-51	Not Used
36-47	Repeat for 3 4x4s at 3,4, 3,8 and 3,12

VEBOX State and Primitive Commands

Every engine can have internal state that can be common and reused across the data entities it processes instead of reloading for every data entity.

There are two kinds of state information:

1. Surface state or state of the input and output data containers.
2. Engine state or the architectural state of the processing unit.

For example in the case of DN/DI, architectural state information such as denoise filter strength can be the same across frames. This section gives the details of both the surface state and engine state.

Each frame should have these commands, in this order:

1. VEBOX_State
2. VEBOX_Surface_state for input & output
3. VEB_DI_IECP

Alternatively VEBOX_Tiling_Convert can be used instead of VEB_DI_IECP.

VEBOX State

This chapter discusses various commands that control the internal functions of the VEBOX. The following commands are covered:

Feature
DN/DI State Table Contents
VEBOX_IECP_STATE
VEBOX_FORWARD_GAMMA_CORRECTION_STATE

Command
VEBOX_STATE
VEBOX_Ch_Dir_Filter_Coefficient

DN-DI State Table Contents

This section contains tables that describe the state commands that are used by the Denoise and Deinterlacer functions.

Command
VEBOX_DNDI_STATE

VEBOX_IECP_STATE

For all piecewise linear functions in the following table, the control points must be monotonically increasing (increasing continuously) from the lowest control point to the highest. Functions which have bias values associated with each control point have the additional restriction that any control points which have the same value must also have the same bias value. The piecewise linear functions include:

- For Skin Tone Detection:
    - Y_point_4 to Y_point_0
    - P3L to P0L
    - P3U to P0U
    - SATP3 to SATP1
    - HUEP3 to HUEP1
    - SATP3_DARK to SATP1_DARK
    - HUEP3_DARK to HUEP1_DARK
- For ACE:
    - Ymax, Y10 to Y1 and Ymin
    - There is no state variable to set the bias for Ymin and Ymax. The biases for these two points are equal to the control point values: B0 = Ymin and B11 = Ymax. That means that if control points adjacent to Ymin and Ymax have the same value as Ymin/Ymax then the biases must also be equal to the Ymin/Ymax control points based on the restriction mentioned above.
- Forward Gamma correction
- Gamut Expansion:
    - Gamma Correction

Inverse Gamma Correction

Command
VEBOX_IECP_STATE
VEBOX_STD_STE_STATE
VEBOX_ACE_LACE_STATE

Command
VEBOX_TCC_STATE
VEBOX_PROCAMP_STATE
VEBOX_CSC_STATE
VEBOX_ALPHA_AOI_STATE
VEBOX_FRONT_END_CSC_STATE

Command
VEBOX_VERTEX_TABLE
VEBOX_CAPTURE_PIPE_STATE
VEBOX_FORWARD_GAMMA_CORRECTION_STATE
VEBOX_RGB_TO_GAMMA_CORRECTION

VEBOX Surface State

Command
VEBOX_SURFACE_STATE

Surface Format Restrictions

The surface formats and tiling allowed are restricted, depending on which function is consuming or producing the surface.

Surface Formats - Feature Notes

Feature
Surfaces are 4 kb aligned, chroma X offset is cache line aligned (16 byte).
If Y8/Y16 is used as the input format, it must also be used for the output format (chroma is not created by VEBOX).
If IECP and either DN or DI are enabled at the same time, it is possible to select any input that is legal for DN/DI and any output which is legal for IECP. The only exception is that if DN or DI are enabled, the IECP is not able to output P216 and P016.
16-bit data from IECP or DN is rounded when converting to 8-bit output formats.
High Speed Bypass has the same format limitations as IECP Input/Output, but the surface formats for the input and output must be the same.
Capture Input is only linear Bayer Surface Format.
Input formats for Demosaic, White Balance, Vignette and Black Level Correction must be linear Bayer.
For capture pipe, we **can** support the combination of DN and P216 and P016. For capture pipe with a P016 output the U/V output is not an average of the 4 component pixels, but the U/V for pixel 4 (the lower right pixel of the 4).

VEB DI IECP Commands

The VEB_DI_IECP command causes the VEBOX to start processing the frames specified by VEB_SURFACE_STATE using the parameters specified by VEB_DI_STATE and VEB_IECP_STATE.

Command
VEB_DI_IECP Command
VEB_DI_IECP Command
VEB_TilingConvert Command

The Surface Control bits for each surface:

Command
VEB_DI_IECP
VEB_DI_IECP Command Surface Control Bits
VEBOX_TILING_CONVERT