Open Source Intel® HD Graphics Programmer's Reference Manual (PRM)

Observability Performance Counters for Intel® Core™ Processor Family

NOTICE:
This document contains information on products in the design phase of development, and Intel® reserves the right to add or remove product features at any time, with or without changes to this open source documentation.

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.

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.

Observability

Table of Contents

Observability Overview

As GFX-enabled systems and usage models have grown in complexity over time, a number of HW features have been added specifically for the purpose of providing more insight into HW behavior while running a commercially available operating system. This chapter documents these features with pointers to relevant sections in other chapters. Supported observability features include:

- Performance counters
- Various other debug registers

NOTE: This document is intended to be used as a companion document to describe the ability to monitor performance for the various Intel graphics open source programmer's reference manuals. Please review those documents to understand the terms, functionality and details for a specific Intel graphics device.

Device Tags and Definitions

The following table lists device "tags" (abbreviations) used in various parts of this document as aliases for the device names. Note that stepping information is sometimes appended to the device tag, e.g., DevSNB:E.

Information without any device tagging is applicable to all devices.

Device Tag	Program Name	Graphics Architecture	SKU	Product Name / Description
SNB	SandyBridge	Gen6	GT1	SandyBridge GT1
			GT2	SandyBridge GT2
IVB	IvyBridge	Gen7	GT1	IvyBridge GT1
			GT2	IvyBridge GT2
VLV	Valley View	Gen7LP	GT	Low-power variation of IVB. VLV, VLV1, VLVT and VLV:X0 are synonymous terms.
HSW	Haswell	Gen7.5	GT1	Haswell GT1
			GT2	Haswell GT2
			GT3	Haswell GT3

Trace

This section contains the following contents:

- Performance Visibility

Performance Visibility

Motivation For Hardware-Assisted Performance Visibility

As the focus on GFX performance and programmability has increased over time, the need for hardware (HW) support to rapidly identify bottlenecks in HW and efficiently tune the work sent to same has become correspondingly important. This part of the PRM describes the HW support for Performance Visibility.

Performance Event Counting

An earlier generation introduced dedicated GFX performance counters to address key issues associated with existing chipset CHAPs counters (lack of synchronization with GFX rendering work and low sampling frequency achievable when sampling via CPU MMIO read). The dedicated counter values are written to memory whenever an MI_REPORT_PERF_COUNT command is placed in the ring buffer.

While this approach eliminated much of the error associated with the previous approaches, it is still limited to sampling the counters only at the boundaries between ring commands. This inherently limited the ability of performance analysis tools to drill down into a primitive, which can contain thousands of triangles and require several hundreds of milliseconds to render.

DevIVB enhanced the aggregating counters to support the additional thread types generated by more advanced graphics APIs that support advanced features such as hull and domain shaders. The high rate at which interesting internal events can occur motivated adding an interrupt-generation capability so that HW could notify SW when the data buffer was approaching full.

DevHSW enhances support for high reporting frequencies by increasing the report buffer size in order to allow SW sufficient time between performance monitoring interrupts, enabling single run histogramming support for events like pixels per polygon. Issues with DevSNB support drove enhancements to enable performance monitoring with RC6 enabled, different report buffer ring wrap behavior, and MMIO visibility into performance counters.

HW Support

This section contains various reporting counters and registers for hardware support for Performance Visibility.

Performance Counter Reporting

When the MI_REPORT_PERF_COUNT command is received, a snapshot of the performance counter values is written to memory. The format used by HW for such reports is selected using the Counter Select field within the OACONTROL register. The organization and number of report formats vary per project and are detailed in the following section. In the following layouts, the RPT_ID is always stored in the lowest addressed DWORD.

[DevSNB]: Under conditions with high memory traffic, the values of the counters may not all be sampled at the same point in time.

[DevIVB]: In order to ensure coherent sampling of the counters, the counters are frozen and will not advance while sampling and reporting to memory is in progress. This may result in small counting errors in internally triggered reporting modes.

[DevSNB] When an over flow condition occurs and the buffers need to be reset, or when software wants to change the OABUFFER to point to a new area in memory, Programming of the performance ring must follow the sequence below.

- Clear OA enable bit by writing 0x2360[0] = 0
- Write OASTATUS2
- Write OABUFFER
- Write OASTATUS1
- Set OA enable bit by writing 0x2360[0] = 1

[DevSNB]: When software wants to reinitialize the OA buffer Tail pointer should follow the below sequence.

- Clear OA enable bit by writing 0x2360[0] = 0
- Ensure Render command streamer doesn't execute MI_REPORT_PERF_COUNT command until OA is enabled.
    - This can be achieved by ensuring render engine is IDLE and blocked from executing any new MI_REPORT_PERF_COUTN commands **OR**
    - Program the OA buffer programming through a series of MI_LOAD_REGISTER_IMM commands through render engine command buffer.
- Program OA Tail pointer, if required initialize OA head pointer.
- Set OA enable bit by writing 0x2360[0] = 1

[DevSNB] [Dev IVB] [DevVLV] [DevHSW] [all steppings] [all skus]: OA unit Is using render clock for its functionality. When trunk level clock gating takes place, OA clock would be gated, unable to count the events from non-render clock domain. Render clock gating must be disabled when OA is enabled to count the events from non-render domain. Unit level clock gating for RCS should also be disabled.

Performance Counter Report Formats

[DevSNB]

Counter Select = 000

A-Cntr 0	A-Cntr 1	A-Cntr 2	A-Cntr 3	A-Cntr 4	TIME_STAMP		RPT_ID
A-Cntr 5	A-Cntr 6	A-Cntr 7	A-Cntr 8	A-Cntr 9	A-Cntr 10	A-Cntr 11	A-Cntr 12

Counter Select = 001

A-Cntr 0	A-Cntr 1	A-Cntr 2	A-Cntr 3	A-Cntr 4	TIME_STAMP		RPT_ID
A-Cntr 5	A-Cntr 6	A-Cntr 7	A-Cntr 8	A-Cntr 9	A-Cntr 10	A-Cntr 11	A-Cntr 12
A-Cntr 13	A-Cntr 14	A-Cntr 15	A-Cntr 16	A-Cntr 17	A-Cntr 18	A-Cntr 19	A-Cntr 20
A-Cntr 21	A-Cntr 22	A-Cntr 23	A-Cntr 24	A-Cntr 25	A-Cntr 26	A-Cntr 27	A-Cntr 28

Counter Select = 010

A-Cntr 0	A-Cntr 1	A-Cntr 2	A-Cntr 3	A-Cntr 4	TIME_STAMP		RPT_ID
A-Cntr 5	A-Cntr 6	A-Cntr 7	A-Cntr 8	A-Cntr 9	A-Cntr 10	A-Cntr 11	A-Cntr 12
Reserved	Reserved	Reserved	Reserved	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Counter Select = 011

A-Cntr 0	A-Cntr 1	A-Cntr 2	A-Cntr 3	A-Cntr 4	TIME_STAMP		RPT_ID
A-Cntr 5	A-Cntr 6	A-Cntr 7	A-Cntr 8	A-Cntr 9	A-Cntr 10	A-Cntr 11	A-Cntr 12
A-Cntr 13	A-Cntr 14	A-Cntr 15	A-Cntr 16	A-Cntr 17	A-Cntr 18	A-Cntr 19	A-Cntr 20
A-Cntr 21	A-Cntr 22	A-Cntr 23	A-Cntr 24	A-Cntr 25	A-Cntr 26	A-Cntr 27	A-Cntr 28
Reserved	Reserved	Reserved	Reserved	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

[DevIVB] [DevVLV] [DevVLVT]

Counter Select = 000

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5

Counter Select = 001

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5
A-Cntr 20	A-Cntr 19	A-Cntr 18	A-Cntr 17	A-Cntr 16	A-Cntr 15	A-Cntr 14	A-Cntr 13
A-Cntr 28	A-Cntr 27	A-Cntr 26	A-Cntr 25	A-Cntr 24	A-Cntr 23	A-Cntr 22	A-Cntr 21

Counter Select = 010

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5
Reserved	Reserved	Reserved	Reserved	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Counter Select = 011

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5
A-Cntr 20	A-Cntr 19	A-Cntr 18	A-Cntr 17	A-Cntr 16	A-Cntr 15	A-Cntr 14	A-Cntr 13
A-Cntr 28	A-Cntr 27	A-Cntr 26	A-Cntr 25	A-Cntr 24	A-Cntr 23	A-Cntr 22	A-Cntr 21
Reserved	Reserved	Reserved	Reserved	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Counter Select = 101

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5
A-Cntr 20	A-Cntr 19	A-Cntr 18	A-Cntr 17	A-Cntr 16	A-Cntr 15	A-Cntr 14	A-Cntr 13
A-Cntr 28	A-Cntr 27	A-Cntr 26	A-Cntr 25	A-Cntr 24	A-Cntr 23	A-Cntr 22	A-Cntr 21
A-Cntr 36	A-Cntr 35	A-Cntr 34	A-Cntr 33	A-Cntr 32	A-Cntr 31	A-Cntr 30	A-Cntr 29
A-Cntr 44	A-Cntr 43	A-Cntr 42	A-Cntr 41	A-Cntr 40	A-Cntr 39	A-Cntr 38	A-Cntr 37
Reserved	Reserved	Reserved	Reserved	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Counter Select = 110

Reserved	Reserved	Reserved	Reserved	Reserved	TIME_STAMP		RPT_ID
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
A-Cntr 36	A-Cntr 35	A-Cntr 34	A-Cntr 33	A-Cntr 32	A-Cntr 31	A-Cntr 30	A-Cntr 29
A-Cntr 44	A-Cntr 43	A-Cntr 42	A-Cntr 41	A-Cntr 40	A-Cntr 39	A-Cntr 38	A-Cntr 37

[DevHSW] Counters layout for various values of select from the register:

Counter Select = 000

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5

Counter Select = 001

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5
A-Cntr 20	A-Cntr 19	A-Cntr 18	A-Cntr 17	A-Cntr 16	A-Cntr 15	A-Cntr 14	A-Cntr 13
A-Cntr 28	A-Cntr 27	A-Cntr 26	A-Cntr 25	A-Cntr 24	A-Cntr 23	A-Cntr 22	A-Cntr 21

Counter Select = 010

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5
B-Cntr 7	B-Cntr 6	B-Cntr 5	B-Cntr 4	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-cntr 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Counter Select = 011

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5
A-Cntr 20	A-Cntr 19	A-Cntr 18	A-Cntr 17	A-Cntr 16	A-Cntr 15	A-Cntr 14	A-Cntr 13
A-Cntr 28	A-Cntr 27	A-Cntr 26	A-Cntr 25	A-Cntr 24	A-Cntr 23	A-Cntr 22	A-Cntr 21
B-Cntr 7	B-Cntr 6	B-Cntr 5	B-Cntr 4	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Counter Select = 100

B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0	INST ADD	TIME_STAMP		RPT_ID
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Counter Select = 101

A-Cntr 4	A-Cntr 3	A-Cntr 2	A-Cntr 1	A-Cntr 0	TIME_STAMP		RPT_ID
A-Cntr 12	A-Cntr 11	A-Cntr 10	A-Cntr 9	A-Cntr 8	A-Cntr 7	A-Cntr 6	A-Cntr 5
A-Cntr 20	A-Cntr 19	A-Cntr 18	A-Cntr 17	A-Cntr 16	A-Cntr 15	A-Cntr 14	A-Cntr 13
A-Cntr 28	A-Cntr 27	A-Cntr 26	A-Cntr 25	A-Cntr 24	A-Cntr 23	A-Cntr 22	A-Cntr 21
A-Cntr 36	A-Cntr 35	A-Cntr 34	A-Cntr 33	A-Cntr 32	A-Cntr 31	A-Cntr 30	A-Cntr 29
A-Cntr 44	A-Cntr 43	A-Cntr 42	A-Cntr 41	A-Cntr 40	A-Cntr 39	A-Cntr 38	A-Cntr 37
B-Cntr 7	B-Cntr 6	B-Cntr 5	B-Cntr 4	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Counter Select = 110

B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0	INST ADD	TIME_STAMP		RPT_ID
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
A-Cntr 36	A-Cntr 35	A-Cntr 34	A-Cntr 33	A-Cntr 32	A-Cntr 31	A-Cntr 30	A-Cntr 29
A-Cntr 44	A-Cntr 43	A-Cntr 42	A-Cntr 41	A-Cntr 40	A-Cntr 39	A-Cntr 38	A-Cntr 37

Counter Select = 111

Reserved	Reserved	Reserved	C-Cntr 0	INST ADD	TIME_STAMP		RPT_ID
B-Cntr 7	B-Cntr 6	B-Cntr 5	B-Cntr 4	B-Cntr 3	B-Cntr 2	B-Cntr 1	B-Cntr 0

Aggregating Counters

Cntr #	Event	Description
A0	Aggregated Core Array Active	The sum of all cycles on all cores spent actively executing instructions. This does not count the time taken to service Send instructions. This time is considered by shader active counters to give the result. [DevSNB]: A23 and A24 may be incorrect when post shader Z and/or stencil tests are required.
A1	Aggregated Core Array Stalled	The sum of all cycles on all cores spent stalled. (at least one thread loaded but the entire core is stalled for any reason)
A2	Vertex Shader Active Time	Total time in clocks the vertex shader spent active on all cores.
A3	Vertex Shader Stall Time (Event not supported in Gen 6 and so this counter should not be used for any performance analysis)	Total time in clocks the vertex shader spent stalled on all cores. This metric must be bucketed by stall type (“other” is ok – but must have buckets for things that are architecturally interesting)
A4	Vertex Shader Stall Time – Core Stall	Total time in clocks the vertex shader spent stalled on all cores – and the entire core was stalled as well. This metric must be bucketed by stall type (“other” is ok – but must have buckets for things that are architecturally interesting)
A5	# VS threads loaded	Number of VS threads loaded at any given time in the EUs.
A6	Vertex Shader ready but not running Time	Total time in clocks the vertex shader spent ready to run but not running on all cores.
A7	Geometry Shader/GPGPU Active Time	Total time in clocks the geometry shader or GPGPU spent active on all cores.
A8	Geometry Shader/GPGPU Stall Time (Event not supported in Gen 6 and so this counter should not be used for any performance analysis)	Total time in clocks the geometry shader or GPGPU spent stalled on all cores. This metric must be bucketed by stall type (“other” is ok – but must have buckets for things that are architecturally interesting)
A9	Geometry Shader/GPGPU Stall Time – Core Stall	Total time in clocks the geometry shader or GPGPU spent stalled on all cores – and the entire core was stalled as well. This metric must be bucketed by stall type (“other” is ok – but must have buckets for things that are architecturally interesting)
A10	# GS/GPGPU threads loaded	Number of GS or GPGPU threads loaded at any given time in the EUs.
A11	Geometry Shader/GPGPU ready but not running Time	Total time in clocks the geometry shader or GPGPU spent ready to run but not running on all cores.
A12	Pixel Shader Active Time	Total time in clocks the pixel shader spent active on all cores.
A13	Pixel Shader Stall Time (Event not supported in Gen 6 and so this counter should not be used for any performance analysis)	Total time in clocks the Pixel shader spent stalled on all cores. This metric must be bucketed by stall type (“other” is ok – but must have buckets for things that are architecturally interesting)
A14	Pixel Shader Stall Time – Core Stall	Total time in clocks the pixel shader spent stalled on all cores – and the entire core was stalled as well. This metric must be bucketed by stall type (“other” is ok – but must have buckets for things that are

Cntr #	Event	Description
		architecturally interesting)
A15	# PS threads loaded	Number of PS threads loaded at any given time in the EUs.
A16	Pixel Shader ready but not running Time	Total time in clocks the Pixel shader spent ready to run but not running on all cores.
A17	Reserved	Reserved
A18	Reserved	Reserved
A19	Reserved	Reserved
A20	Reserved	Reserved
A21	Pixel Kill Count	Number of pixels/samples killed in the pixel shader. (How about chroma key?) [DevSNB]: Count reported is 2X or 4X the actual count for non-dual source blending or dual source blending respectively.
A22	Alpha Test Pixels Failed	Number of pixels/samples that fail alpha-test. Alpha to coverage may have some challenges in per-pixel invocation.
A23	Post PS Stencil Pixels Failed	Number of pixels/samples fail stencil test in the backend.
A24	Post PS Z buffer Pixels Failed	Number of pixels/samples fail Z test in the backend.
A25	Pixels/samples Written in the frame buffer	MRT case will report multiple of those.
A26	*GPU Busy*	CSunit indicating that ring is idle.
A27	*CL active and not stalled*	Clipper Fixed Function is active but not stalled
A28	*SF active and stalled*	SF Fixed Function is active but not stalled

Aggregating Counters

Cntr #	Event	Description
A0	Aggregated Core Array Active	The sum of all cycles on all cores spent actively executing instructions. This does not count the time taken to service Send instructions. This time is considered by shader active counters to give the result.
A1	Aggregated Core Array Stalled	The sum of all cycles on all cores where the EU is not idle and is not actively executing ISA instructions. Generally this means that all loaded threads on the EU are stalled on some data dependency, but this also includes the time during which the TS is loading the thread dispatch header into the EU prior to thread execution and no other thread is fully loaded.
A2	Vertex Shader Active Time	Total time in clocks the vertex shader spent active on all cores.
A3	Reserved	Reserved
A4	Vertex Shader Stall Time – Core Stall	Total time in clocks the vertex shader spent stalled on all cores – and the entire core was stalled as well.
A5	# VS threads loaded	Number of VS threads loaded at any given time in the EUs.
A6	Reserved	Reserved
A7	Hull Shader Active Time	Total time in clocks the Hull shader spent active on all cores.

Cntr #	Event	Description
A8	Reserved	Reserved
A9	Hull Shader Stall Time – Core Stall	Total time in clocks the Hull shader spent stalled on all cores – and the entire core was stalled as well.
A10	# HS threads loaded	Number of HS threads loaded at any given time in the EUs.
A11	Reserved	Reserved
A12	Domain Shader Active Time	Total time in clocks the Domain shader spent active on all cores.
A13	Reserved	Reserved
A14	Domain Shader Stall Time – Core Stall	Total time in clocks the domain shader spent stalled on all cores – and the entire core was stalled as well.
A15	# DS threads loaded	Number of DS threads loaded at any given time in the EUs.
A16	Reserved	Reserved
A17	Compute Shader Active Time	Total time in clocks the compute shader spent active on all cores.
A18	Reserved	Reserved
A19	Compute Shader Stall Time – Core Stall	Total time in clocks the compute shader spent stalled on all cores – and the entire core was stalled as well.
A20	# CS threads loaded	Number of CS threads loaded at any given time in the EUs.
A21	Reserved	Reserved
A22	Geometry Shader Active Time	Total time in clocks the geometry shader spent active on all cores.
A23	Reserved	Reserved
A24	Geometry Shader Stall Time – Core Stall	Total time in clocks the geometry shader spent stalled on all cores – and the entire core was stalled as well.
A25	# GS threads loaded	Number of GS threads loaded at any given time in the EUs.
A26	Reserved	Reserved
A27	Pixel Shader Active Time	Total time in clocks the pixel shader spent active on all cores.
A28	Reserved	Reserved
A29	Pixel Shader Stall Time – Core Stall	Total time in clocks the pixel shader spent stalled on all cores – and the entire core was stalled as well.
A30	# PS threads loaded	Number of PS threads loaded at any given time in the EUs.

Cntr #	Event	Description
A31	Reserved	Reserved
A32	HIZ Fast Z Test Pixels Passing	[DevIVB] Count of pixels that pass the fast check (8x8). This counter under-counts slightly; a B-counter can be defined to correct the count. [DevVLV, DevVLVT] Count of pixels that pass the fast check (8x8). [DevHSW] Count of pixels that pass HiZ (8x8).
A33	HIZ Fast ZTest Pixels Failing	[DevIVB:GT1, DevVLV, DevVLVT, DevHSW] Count of pixels that fail the fast check (8x8). [DevIVB:GT2] Reserved.
A34	[DevIVB] Slow Ztest Pixels Passing [DevHSW] Reserved	[DevIVB] Count of pixels passing the slow check (2x2) [DevHSW] Reserved
A35	Slow ZTest Pixels Failing	Count of pixels that fail the slow check (2x2)
A36	Pixel Kill Count	Number of pixels/samples killed in the pixel shader. Erratum: [DevIVB, DevVLV, DevVLVT]: Count reported is 2X the actual count for or dual source render target messages i.e. when PS has two output colors.
A37	Alpha Test Pixels Failed	Number of pixels/samples that fail alpha-test. Alpha to coverage may have some challenges in per-pixel invocation.
A38	Post PS Stencil Pixels Failed	Number of pixels/samples failing stencil test after the pixel shader has executed.
A39	Post PS Z buffer Pixels Failed	Number of pixels/samples fail Z test after the pixel shader has executed.
A40	3D/GPGPU Render Target Writes	MRT scenarios will cause this counter to increment multiple times.
A41	Render Engine Busy	Render engine is not idle. GPU Busy aggregate counter doesn't increment under the following conditions: 1. Context Switch in Progress. 2. GPU stalled on executing MI_WAIT_FOR_EVENT. 3. GPU stalled on execution MI_SEMAPHORE_MBOX. 4. RCS idle but other parts of GPU active (e.g. only media engines active)
A42	VS bottleneck	VSunit is stalling VF (upstream unit) and starving HS (downstream unit)
A43	GS bottleneck	GSunit is stalling DS (upstream unit) and starving SOL(downstream unit)
A44	Reserved	Reserved

MI_REPORT_PERF_COUNT

MI_REPORT_PERF_COUNT

Project: DevIVB+

Length Bias: 2

The MI_REPORT_PERF_COUNT command causes the GFX hardware to write out a snap-shot of performance counters to the address specified in this command along with constant ID field supplied and the time-stamp counter. This write is required to be treated as a cacheable write irrespective of GTT entry memory type. This command is specific to the render engine.

Programming Notes	Project
This command is to be used for performance debug mode and can be inserted after events of interest (frequently before and after a 3DPRIMITIVE command). SW is entirely responsible for managing the ID field and addresses used by such a series of commands.	
GTT_SELECT must not be set to 1 (i.e. GGTT) when MI_REPORT_PERF_COUNT command is programmed in a non-privileged batch buffer. Refer to the "User Mode Privileged commands" Table in MI_BATCH_BUFFER_START command section for more details. Each batch buffer is explicitly tagged as privileged or non-privileged.	DevHSW

DWord	Bit	Description
0	31:29	**Command Type** Default Value: 0h MI_COMMAND Format: OpCode
	28:23	**MI Command Opcode** Default Value: 28h MI_ Format: OpCode
	22:6	**Reserved** Format: MBZ
	5:0	**DWord Length** Format: =n Total Length - 2 Value: 1h — Name: Excludes DWord (0,1) **[Default]** — Project: IVB,HSW
1	31:6	**Memory Address** Format: GraphicsAddress[31:6] This field specifies 64B aligned GFX MEM address where the chap counter values are reported.
	5	**Reserved** Format: MBZ
	4	**Core Mode Enable** Project: DevHSW+ Format: U1 This bit is set then the address will be offset by the Core ID:If Core ID 0, then there is no offsetIf Core ID 1, then the Memory is offset by the size of the data(64b).
	3:1	**Reserved** Format: MBZ
	0	**GTT Select**

		Format: U1 This field when set (i.e. bit = 1) selects the GGTT for address translation. When this bit is 0 (default value), HW should use PGTT for address translation.
2 **Project:** DevIVB+	31:0	**Report ID** Project: DevIVB+ Format: U32 This field specifies the ID provided by SW for a given report command. It can be tracked to use different flavors of these reports based on where in command-stream they are inserted.This field is reported only when Counter Select Field is 0.

Performance Statistics Registers

OACONTROL - Observation Architecture Control	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02360h
Valid Projects:	[DevSNB, DevIVB, DevVLV, DevVLVT, DevHSW]

This register controls global OA functionality, report format, interrupt steering and context filtering.

Workaround	Project
Workaround : If software intends to reset the OA buffer to start a new one, after clearing the Timer Enable bit, software must check to see if the head pointer in OASTATUS2 is greater than the tail pointer in OASTATUS1. If so software must program the head pointer to a value less than the current head pointer value. This must be done before the buffer becomes active again	DevSNB:GT2

DWord	Bit	Description
0	31:12	**Select Context ID** Project: DevSNB, DevIVB, DevVLV, DevHSW Specifies the context ID of the one context that affects the performance counters. All other contexts are ignored.
	11:6	**Timer Period** Project: DevIVB, DevVLV, DevVLVT, DevHSW Format: Select Specifies the period of the timer strobe as a function of the minimum TIME_STAMP resolution. The period is determined by selecting a specified bit from the TIME_STAMP register as follows: StrobePeriod = MinimumTimeStampPeriod * $2^{(\text{TimerPeriod} + 1)}$ The exponent is defined by this field. Note: The TIME_STAMP is not reset at start time so the phase of the strobe is not synchronized with the enable of the OA unit. This could result in approximately a full StrobePeriod elapsing prior to the first trigger. Usage for this mechanism should be time based periodic triggering, typically.
	11:6	**Timer Period** Project: DevSNB Format: Select Specifies the period of the timer strobe as a function of the minimum TIME_STAMP resolution. The period is determined by selecting a specified bit from the TIME_STAMP register as follows: StrobePeriod = MinimumTimeStampPeriod * $2^{(\text{TimerPeriod} + 4)}$ The exponent is defined by this field. Note: The TIME_STAMP is not reset at start time so the phase of the strobe is not synchronized with the enable of the OA unit. This could result in approximately a full StrobePeriod elapsing prior to the first trigger. Usage for this mechanism should be time based periodic triggering, typically.

OACONTROL - Observation Architecture Control

5 — Timer Enable

Project:	DevSNB, DevIVB, DevVLV, DevVLVT, DevHSW
Format:	Enable

Description	Project
This field enables the timer logic to output a periodic strobe, as defined by the Timer Period. When disabled the timer output is not asserted.	

Value	Name	Description
0h	Disable **[Default]**	Counter does not get written out on regular interval
1h	Enable	Counter gets written out on regular intervals, defined by the Timer Period

4:2 — Counter Select

Project:	Pre-DevHSW
Format:	Performance Counter Report Format

This field selects which performance counter report format to use, please refer to Performance Counter Report Formats section for more details on the structure of the format.

4:2 — Reserved

Project:	DevHSW
Format:	MBZ

1 — Specific Context Enable

Format:	Enable

Description	Project
Enables counters to work on a context specific workload. The context is given by bits 31:12. OA unit level clock gating must be ENABLED when using specific ContextID feature.	

Value	Name	Description	Project
0h	Disable **[Default]**	All contexts are considered	
1h	Enable	Only the contexts with the Select Context ID are considered	DevHSW

Programming Notes	Project
Must be set to '1' (context aware)	DevSNB:GT2:A

0 — Performance Counter Enable

Project:	All
Format:	Enable

Global performance counter enable. If clear, no counting will occur. MI_REPORT_PERF_COUNT is undefined when clear.

Programming Notes	Project
When this bit is set, in order to have coherent counts, RC6 power state and render trunk clock gating must be disabled. This can be achieved by programming MMIO registers as 0xA094=0x0 and 0xA090[31]=1.	DevSNB, DevIVB, DevVLV, DevVLVT, DevHSW

Workaround	Project
Workaround : EU clock gating must be disabled when this bit is set.	DevSNB

OASTATUS1 - Observation Architecture Status Register 1

Register Space:	MMIO: 0/2/0
Project:	DevSNB,DevIVB,DevHSW,DevVLV
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02364h
Valid Projects:	DevSNB,DevIVB,DevHSW,DevVLV

This register is used to program the OA unit.

DWord	Bit	Description
0	31:6	**Tail Pointer** Project: DevSNB,DevIVB,DevHSW,DevVLV Virtual address of the internal trigger based buffer and it is updated for every 64B cacheline write to memory when reporting via internal trigger. This pointer will not be updated for MI_REPORT_PERF_COUNT command based writes. **Programming Notes** When OA is enabled, this address must be programmed by SW to the base address of the internal trigger base mechanism. SW must ensure that Tail pointer and the Head Pointer (in OASTATUS2) do not have different values while programming.
	5:3	**Inter Trigger Report Buffer Size** Project: DevSNB,DevIVB,DevHSW,DevVLV This field indicates the size of buffer for internal trigger mechanism. This field is programmed in terms of multiple of 128KB.

Value	Name	Description	Project
0h	All context considered **[Default]**		
0b		128KB	DevHSW
1b		256KB	DevHSW
2		512KB	DevHSW
3		1MB	DevHSW
4		2MB	DevHSW
5		4MB	DevHSW
6		8MB	DevHSW
7		16MB	DevHSW
0b		16KB	DevSNB, DevIVB
1b		32KB	DevSNB, DevIVB
2		48KB	DevSNB, DevIVB
3		64KB	DevSNB, DevIVB
4		80KB	DevSNB, DevIVB
5		96KB	DevSNB, DevIVB
6		112KB	DevSNB, DevIVB
7		128KB	DevSNB, DevIVB

OASTATUS1 - Observation Architecture Status Register 1

	2	**Counter OverFlow Error** Format: Select **Description** / **Project** This bit is set if any of the counters overflows. This bit can be reset by SW in B0. / Counter Overflow generation for B counters is incorrect. Counter overflow generation is getting generated on counter bit[31] transitioning form 0->1 OR 1->0. Counter overflow should be generated only when counter bit[31] transitions from 1->0, i.e when the counter wraps around max value. SW Should consider counter overflow as valid only when counter bit[31] is '0'. A and C counter overflow generation is correct. / DevSNB, DevIVB **Programming Notes** / **Project** This bit must be cleared after the ring is enabled and before OA is enabled. / DevSNB
	1	**Buffer Overflow** Default Value: 0h This bit is set when the Tail-pointer - Head pointer > max internal trigger buffer size
	0	**Report Lost Error** Format: Enable This bit is set if the Report Logic is requested to write out the counter values before the previous report request was completed. The report request is ignored and the counter continue to count.This bit can be reset by SW in B0. **Programming Notes** / **Project** Report Lost Error status is not functional and must not be looked at for any purposes. / VLV,VLVT

OASTATUS2 - Observation Architecture Status Register 2

Register Space:	MMIO: 0/2/0
Project:	DevSNB, DevIVB, DevHSW
Default Value:	0x00000000 [IVB,VLV,VLVT,SNB] 0x00000001 [HSW]
Access:	R/W
Size (in bits):	32
Address:	02368h

This register is used to program the OA unit.

DWord	Bit	Description
0	31:6	**Head Pointer** Virtual address of the internal trigger based buffer that is updated by software after consuming from the report buffer. This pointer must be updated by SW for internal trigger base buffer only.
	5	**Reserved** Format: MBZ
	4	**Tail Pointer Wrap Mask** Project: DevHSW **Programming Notes** This bit should be set in order to program Tail Pointer Wrap Flag. This bit is for HW internal use. SW should always set this bit to 0.
	3	**Tail Pointer Wrap Flag** Project: DevHSW Format: U1 **Programming Notes** This bit is for HW internal use to context save /restore Tail Pointer Wrap Flag. SW should not program this bit. This bit gets programmed only when Tail Pointer Wrap Mask bit is set.
	2	**Head Pointer Wrap Mask** Project: DevHSW **Programming Notes** This bit should be set in order to program Head Pointer Wrap Flag. This bit is for HW internal use. SW should always set this bit to 0.
	4:1	**Reserved** Project: Pre-DevHSW Format: MBZ
	1	**Head Pointer Wrap Flag** Project: DevHSW Format: U1 **Programming Notes**

OASTATUS2 - Observation Architecture Status Register 2		
		This bit is for HW internal use to context save /restore Head Pointer Wrap Flag. SW should not program this bit. This bit gets programmed only when Head Pointer Wrap Mask bit is set.
	0	**Memory select PPGTT/GGTT access** Project: DevHSW Access: RO Value: 0 — PPGTT Value: 1 — GGTT **[Default]** OABUFFER must always reside in GGTT memory. This bit must be set to '1'. DevHSW
	0	**Memory select PPGTT/GGTT access** Project: DevSNB, DevIVB Access: R/W Value: 0 — PPGTT **[Default]** Value: 1 — GGTT

OABUFFER - Observation Architecture Buffer	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000 [SNB,IVB,VLV,VLVT,HSW]
Size (in bits):	32
Address:	023B0h
Valid Projects:	[DevSNB]
Access:	WO
Address:	023B0h
Valid Projects:	[DevIVB, DevVLV, DevVLVT, DevHSW]
Access:	R/W

This register is used to program the OA unit.

Programming Notes	Project
This MMIO must be set before the OASTATUS1 register and set after the OASTATUS2 register. This is to enable proper functionality of the overflow bit.	DevSNB+
Report Buffer Offset Must be 512KB aligned.	DevSNB, DevIVB, DevVLV, DevVLVT
Report Buffer Offset Must be 16MB aligned.	DevHSW

DWord	Bit	Description
0	31:6	**Report Buffer Offset**

	Format:	GraphicsAddress[31:6]

This field specifies 64B aligned GFX MEM address where the chap counter values are reported.

5 Reserved

Project:	DevSNB+
Format:	MBZ

4 OVERRUN STATUS

Default Value:	0h Enabled
Project:	DevHSW
Format:	Enable

This field indicates the status of overrun for debug purpose. This bit is read only and writing to this bit will have no effect. This bit will reflect the status of overrun irrespective of Overrun Mode enabled or disabled.

4:3 Reserved

Project:	Pre-DevHSW
Format:	MBZ

3 Disable Overrun Mode

Project:	DevHSW
Format:	Enable

This field defines the mode of reporting for internal trigger/timer based reporting. When this bit is set, overrun does not lose reports but stops reporting. Based on the head and tail pointer, when HW detects room for the report, it would resume reporting to the buffer. This mode would not set the over-run bit in the register. When this mode bit is reset, buffer overrun can happen and lose the reports while setting the buffer over-run bit.

Value	Name	Description	Project
0h	Disable **[Default]**	Counter gets written out on regular intervals, defined by the Timer Period	DevHSW
1h	Enable	Counter does not get written out on regular interval	DevHSW

2 Reserved

Project:	DevSNB
Format:	MBZ

1 Counter Stop Resume Mechanism Enable

Project:	DevIVB, DevHSW

1:0 Reserved

Project:	DevSNB
Format:	MBZ

0 Counter Stop-Resume Mechanism

Project:	DevIVB, DevHSW
Format:	Enable

Value	Name	Description
0	**[Default]**	Reserved
1		Resume counting for all counters

CEC0-0 - Customizable Event Creation 0-0	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02390h
Valid Projects:	[DevSNB, DevIVB]
Address:	02770h
Valid Projects:	[DevHSW+]
This register is used to define custom counter event 0, set this register to 0x00000003 and configure CEC0-1 properly in order to have counter B0 increment every GPU clock.	

CEC0-1 - Customizable Event Creation 0-1	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02394h
Valid Projects:	[DevSNB, DevIVB, DevVLV, DevVLVT]
Address:	02774h
Valid Projects:	[DevHSW+]
This register is used to define custom counter event 0, set it to 0x0000FFFE and configure CEC0-0 properly in order to have counter B0 increment every GPU clock.	

CEC1-0 - Customizable Event Creation 1-0	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02398h
Valid Projects:	[DevSNB, DevIVB, DevVLV, DevVLVT]
Address:	02778h
Valid Projects:	[DevHSW+]
This register is used to define custom counter event 1, set this register to 0x00000003 and configure CEC1-1 properly in order to have counter B1 increment every GPU clock.	

CEC1-1 - Customizable Event Creation 1-1	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	0239Ch
Valid Projects:	[DevSNB, DevIVB]
Address:	0277Ch
Valid Projects:	[DevHSW+]
This register is used to define custom counter event 1, set it to 0x0000FFFE and configure CEC1-0 properly in order to have counter B1 increment every GPU clock.	

CEC2-0 - Customizable Event Creation 2-0	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	023A0h
Valid Projects:	[DevSNB, DevIVB]
Address:	02780h
Valid Projects:	[DevHSW+]
This register is used to define custom counter event 2, set this register to 0x00000003 and configure CEC2-1 properly in order to have counter B2 increment every GPU clock.	

CEC2-1 - Customizable Event Creation 2-1	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	023A4h
Valid Projects:	[DevSNB, DevIVB]
Address:	02784h
Valid Projects:	[DevHSW+]
This register is used to define custom counter event 2, set it to 0x0000FFFE and configure CEC2-0 properly in order to have counter B2 increment every GPU clock.	

CEC3-0 - Customizable Event Creation 3-0	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	023A8h
Valid Projects:	[DevSNB, DevIVB]
Address:	02788h
Valid Projects:	[DevHSW+]
This register is used to define custom counter event 3, set this register to 0x00000003 and configure CEC3-1 properly in order to have counter B3 increment every GPU clock.	

CEC3-1 - Customizable Event Creation 3-1	
Register Space:	MMIO: 0/2/0
Project:	DevSNB+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	023ACh
Valid Projects:	[DevSNB, DevIVB]
Address:	0278Ch
Valid Projects:	[DevHSW+]
This register is used to define custom counter event 3, set it to 0x0000FFFE and configure CEC3-0 properly in order to have counter B3 increment every GPU clock.	

CEC4-0 - Customizable Event Creation 4-0	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02790h
This register is used to define custom counter event 4, set this register to 0x00000003 and configure CEC4-1 properly in order to have counter B4 increment every GPU clock.	

CEC4-1 - Customizable Event Creation 4-1	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02794h
This register is used to define custom counter event 4, set it to 0x0000FFFE and configure CEC4-0 properly in order to have counter B4 increment every GPU clock.	

CEC5-0 - Customizable Event Creation 5-0	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02798h
This register is used to define custom counter event 5, set this register to 0x00000003 and configure CEC5-1 properly in order to have counter B0 increment every GPU clock.	

CEC5-1 - Customizable Event Creation 5-1	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	0279Ch
This register is used to define custom counter event 5, set it to 0x0000FFFE and configure CEC5-0 properly in order to have counter B5 increment every GPU clock.	

CEC6-0 - Customizable Event Creation 6-0	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	027A0h
This register is used to define custom counter event 6, set this register to 0x00000003 and configure CEC6-1 properly in order to have counter B6 increment every GPU clock.	

CEC6-1 - Customizable Event Creation 6-1	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	027A4h
This register is used to define custom counter event 6, set it to 0x0000FFFE and configure CEC6-0 properly in order to have counter B6 increment every GPU clock.	

CEC7-0 - Customizable Event Creation 7-0	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	027A8h
This register is used to define custom counter event 7, set this register to 0x00000003 and configure CEC7-1 properly in order to have counter B7 increment every GPU clock.	

CEC7-1 - Customizable Event Creation 7-1	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	027ACh
This register is used to define custom counter event 7, set it to 0x0000FFFE and configure CEC7-0 properly in order to have counter B7 increment every GPU clock.	

The following Performance Statistics registers must be part of the power context:

OAPERF_A0 - Aggregate Perf Counter A0		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02800h	
This register reflects the count value of the OA Performance counter A0. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A1 - Aggregate Perf Counter A1		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02804h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A1. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A2 - Aggregate Perf Counter A2		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02808h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A2. DefaultValue="00000000h"		
DWord	**Bit**	**Description**

OAPERF_A2 - Aggregate Perf Counter A2		
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A3 - Aggregate Perf Counter A3	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	0280Ch
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A3. DefaultValue="00000000h"

DWord	Bit	Description
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A4 - Aggregate Perf Counter A4	
Register Space:	MMIO: 0/2/0
Project:	HSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02810h
Valid Projects:	[HSW]

This register reflects the count value of the OA Performance counter A4. DefaultValue="00000000h"

DWord	Bit	Description
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A5 - Aggregate Perf Counter A5	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02814h
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A5. DefaultValue="00000000h"

DWord	Bit	Description
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A6 - Aggregate Perf Counter A6	
Register Space:	MMIO: 0/2/0
Project:	HSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02818h
Valid Projects:	[HSW]

This register reflects the count value of the OA Performance counter A6. DefaultValue="00000000h"

DWord	Bit	Description
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A7 - Aggregate Perf Counter A7	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	0281Ch
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A7. DefaultValue="00000000h"

DWord	Bit	Description
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A8 - Aggregate Perf Counter A8	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02820h
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A8. DefaultValue="00000000h"

DWord	Bit	Description
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A9 - Aggregate Perf Counter A9	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02824h
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A9. DefaultValue="00000000h"

DWord	Bit	Description

OAPERF_A9 - Aggregate Perf Counter A9		
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A10 - Aggregate Perf Counter A10		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02828h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A10. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A11 - Aggregate Perf Counter A11		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	0282Ch	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A11. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A12 - Aggregate Perf Counter A12		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02830h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A12. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A13 - Aggregate Perf Counter A13		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02834h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A13. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A14 - Aggregate Perf Counter A14		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02838h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A14. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations**

OAPERF_A14 - Aggregate Perf Counter A14		
		This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A15 - Aggregate Perf Counter A15		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	0283Ch	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A15. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A16 - Aggregate Perf Counter A16		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02840h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A16. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A17 - Aggregate Perf Counter A17		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02844h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A17. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A18 - Aggregate Perf Counter A18		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02848h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A9. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A19 - Aggregate Perf Counter A19		
Register Space:	MMIO: 0/2/0	
Project:	HSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	0284Ch	
Valid Projects:	[HSW]	
This register reflects the count value of the OA Performance counter A19. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A20 - Aggregate Perf Counter A20		
Register Space:	MMIO: 0/2/0	
Project:	HSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02850h	
Valid Projects:	[HSW]	
This register reflects the count value of the OA Performance counter A20. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A21 - Aggregate Perf Counter A21		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02854h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A21. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A22 - Aggregate Perf Counter A22		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02858h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A22. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A23 - Aggregate Perf Counter A23		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	0285Ch	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A23. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A24 - Aggregate Perf Counter A24		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02860h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A24. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A25 - Aggregate Perf Counter A25		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02864h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A25. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations**

OAPERF_A25 - Aggregate Perf Counter A25		
		This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A26 - Aggregate Perf Counter A26		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02868h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A26. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A27 - Aggregate Perf Counter A27		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	0286Ch	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A27. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A28 - Aggregate Perf Counter A28	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02870h

OAPERF_A28 - Aggregate Perf Counter A28		
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A28. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A29 - Aggregate Perf Counter A29		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02874h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A29. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A30 - Aggregate Perf Counter A30		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02878h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A30. DefaultValue="00000000h"		
DWord	**Bit**	**Description**
0	31:0	**Considerations** This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A31 - Aggregate_Perf_Counter_A31		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	0278Ch	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A31		
DWord	**Bit**	**Description**
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A32 - Aggregate_Perf_Counter_A32		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02880h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A32		
DWord	**Bit**	**Description**
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A33 - Aggregate_Perf_Counter_A33	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	02884h

OAPERF_A33 - Aggregate_Perf_Counter_A33		
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A33		
DWord	**Bit**	**Description**
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A34 - Aggregate_Perf_Counter_A34		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW+	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02888h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A34		
DWord	**Bit**	**Description**
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A35 - Aggregate_Perf_Counter_A35		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW+	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	0288Ch	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A35		
DWord	**Bit**	**Description**
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A36 - Aggregate_Perf_Counter_A36		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02890h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A36		

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A37 - Aggregate_Perf_Counter_A37		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02894h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A37		

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A38 - Aggregate_Perf_Counter_A38		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	02898h	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A38		

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A39 - Aggregate_Perf_Counter_A39		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	0289Ch	
Valid Projects:	[DevHSW]	
This register reflects the count value of the OA Performance counter A39		

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A40 - Aggregate_Perf_Counter_A40	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028A0h
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A40

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A41 - Aggregate_Perf_Counter_A41	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028A4h
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A41

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A42 - Aggregate_Perf_Counter_A42	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028A8h
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A42

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A43 - Aggregate_Perf_Counter_A43	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028ACh
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A43

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_A44 - Aggregate_Perf_Counter_A44	
Register Space:	MMIO: 0/2/0
Project:	DevHSW
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028B0h
Valid Projects:	[DevHSW]

This register reflects the count value of the OA Performance counter A44

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_B0 - Boolean_Counter_B0		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW+	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	028B4h	
Valid Projects:	[DevHSW]	

This register enables the current live value of performance counter B0 to be read. Since what conditions cause B0 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_B1 - Boolean_Counter_B1		
Register Space:	MMIO: 0/2/0	
Project:	DevHSW+	
Default Value:	0x00000000	
Access:	R/W	
Size (in bits):	32	
Address:	028B8h	
Valid Projects:	[DevHSW]	

This register enables the current live value of performance counter B1 to be read. Since what conditions cause B1 to increment are defined by the programming of CEC1-0/CEC1-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_B2 - Boolean_Counter_B2	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028BCh
Valid Projects:	[DevHSW]

This register enables the current live value of performance counter B2 to be read. Since what conditions cause B2 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_B3 - Boolean_Counter_B3	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028C0h
Valid Projects:	[DevHSW]

This register enables the current live value of performance counter B3 to be read. Since what conditions cause B3 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_B4 - Boolean_Counter_B4	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028C4h
Valid Projects:	[DevHSW]

This register enables the current live value of performance counter B4 to be read. Since what conditions cause B4 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_B5 - Boolean_Counter_B5	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028C8h
Valid Projects:	[DevHSW]

This register enables the current live value of performance counter B5 to be read. Since what conditions cause B5 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_B6 - Boolean_Counter_B6	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028CCh
Valid Projects:	[DevHSW]

This register enables the current live value of performance counter B6 to be read. Since what conditions cause B6 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

OAPERF_B7 - Boolean_Counter_B7	
Register Space:	MMIO: 0/2/0
Project:	DevHSW+
Default Value:	0x00000000
Access:	R/W
Size (in bits):	32
Address:	028D0h
Valid Projects:	[DevHSW]

This register enables the current live value of performance counter B7 to be read. Since what conditions cause B7 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

DWord	Bit	Description
0	31:0	**Considerations** Format: U32 This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.