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

Volume 10: High Efficiency Video Coding (HEVC)

For the 2014-2015 Intel Atom™ Processors, Celeron™ Processors and Pentium™ Processors based on the "Cherry Trail/Braswell" Platform
(Cherryview/Braswell graphics)

June 2015, Revision 1.0

Creative Commons License

Notices and Disclaimers

Table of Contents

High Efficiency Video Coding (HEVC) Introduction

The HEVC Codec Pipeline (HCP) is a fixed function hardware video codec responsible for decoding HEVC (High Efficiency Video Coding) video streams.

Scope

The primary scope of the HCP BSpec document is to provide a description of the HCP commands processed by the Video Command Streamer (VCS). The secondary scope is to provide a description of the status registers on the Message Channel Interface to support encoding and decoding of the HEVC video format.

The BSpec sections include:

- Summary of Features
- Architecture Overview
- Commands
- Register Definitions
- Acronyms and Applicable Standards

Summary of Features

The following sections define the HEVC Decoder and Encoder general features and the features specific to HEVC decoding and encoding, respectively.

HCP Hardware Pipeline Features

- Supports both decoder and encoder functions, setup on a per picture basis:
  - Hardware acceleration provides Ctb/CU level decode and encode.
  - No context switch is supported within a frame process.
- Supports Video Command Streamer (VCS):
  - Shared with MFX HW pipeline, and at any one time, only one pipeline (MFX or HCP) and one operation (decoding or encoding) can be active.
- Supports Message Channel Interface:

Feature
Supports Tile-Y Legacy.

- Supports NV12 video buffer plane:
  - Supports 4:2:0, 8-bit per pixel component (Y, Cb and Cr) video.
- 12KB motion compensation read cache.
- Support 8Kx8K frame size:

Feature
Platform real-time performance determines the actual maximum frame size and frame rate that can be achieved.
Minimum decode performance target is 4Kx2K@30fps.

HEVC Decoder Features

- Supports full-featured HEVC Main Profile standard, up to Level 6.2.
- Supports the long format HW decoding interface:
  - All headers (SPS, PPS, Slice Header) are parsed and decoded outside the HCP HW pipeline. They are then fed to the HW through a set of HCP state commands.
- Supports inner-loop decode with hardware entry points for Encoder.
- Error detection/resiliency down to the Ctb/CU level.

HCP Command Summary

The HCP is configured for decoding through a set of batch commands defined in the following sections. The software driver builds a frame level workload using these commands and stores these workloads in graphics memory where they are fetched by the Video Command Streamer (VCS) and presented to the HCP for processing. The commands are processed by the Workload Parser within the HCP and the hardware is configured by the Workload Parser prior to each frame level encode or decode. A workload is defined as a set of commands necessary to encode or decode one frame.

The software driver is required to read the HCP disable fuse to determine if the HCP is enabled. If it is disabled, then the software driver must not enable HCP batch commands to be sent to the HCP or a hang event may occur. Only when the HCP is enabled through the fuse, should the batch commands be sent to the HCP.

Workload Command Model

DWord0 of each command is defined in HCP DWord0 Command Definition. The HCP is selected with the **Media Instruction Opcode "7h**" for all HCP Commands.

HCP DWord0 Command Definition

DWord	Bits	Description
0	31:29	**Command Type** = PARALLEL_VIDEO_PIPE = 3h
	28:27	**Pipeline Type** = 2h
	26:23	**Media Instruction Opcode = Codec/Engine Name** = HCP = 7h
	22:16	**Media Instruction Command** = *<see Workload Command Model>*
	15:12	**Reserved: MBZ**
	11:0	**Dword Length** (Excludes Dwords 0, 1) = *<command length>*

Each HCP command has assigned a media instruction command as defined in HCP Media Instruction Commands (Opcode=7h).

HCP Media Instruction Commands (Opcode=7h)

Media Instruction Command	Command DWord0 [22:16]	Gen9 Mode	Scope
HCP_PIPE_MODE_SELECT	0h	Enc/Dec	Picture
HCP_SURFACE_STATE	1h	Enc/Dec	Picture
HCP_PIPE_BUF_ADDR_STATE	2h	Enc/Dec	Picture
HCP_IND_OBJ_BASE_ADDR_STATE	3h	Enc/Dec	Picture
HCP_QM_STATE	4h	Enc/Dec	Picture
HCP_FQM_STATE (encoder only)	5h	Enc	Picture
Reserved	8h-Fh		
HCP_PIC_STATE	10h	Enc/Dec	Picture
HCP_TILE_STATE	11h	Dec	Picture
HCP_REF_IDX_STATE	12h	Enc/Dec	Slice
HCP_WEIGHTOFFSET	13h	Enc/Dec	Slice
HCP_SLICE_STATE	14h	Enc/Dec	Slice
Reserved	15h-1Fh		
HCP_BSD_OBJECT_STATE (decoder only)	20h	Dec	Slice
HCP_PAK_OBJECT (encoder only)	21h	Enc	LCU
HCP_INSERT_PAK_OBJECT (encoder only)	22h	Enc	Bitstream
Reserved	23h-7Fh		

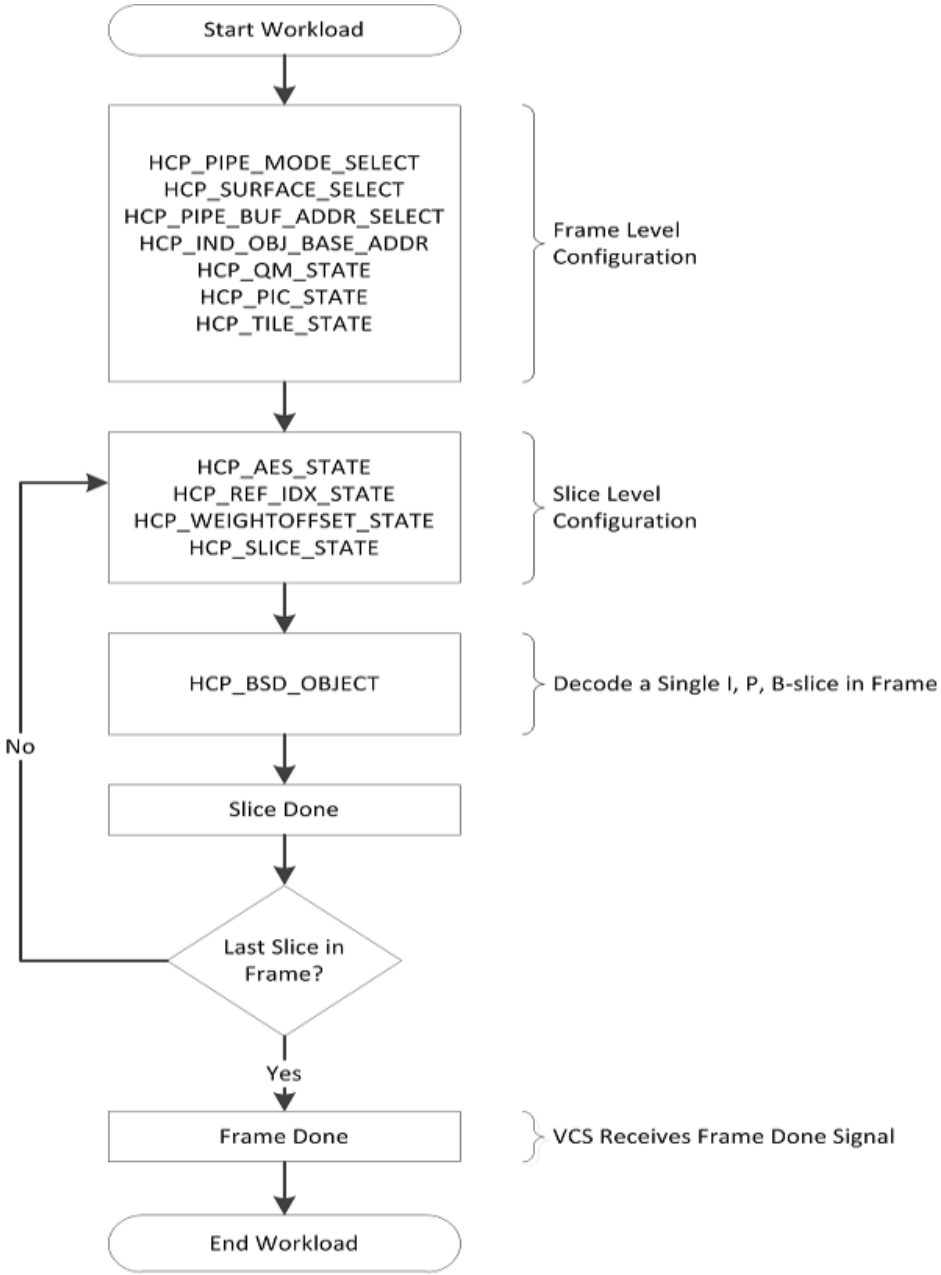
HCP Command Sequence Examples

This section is currently under development.

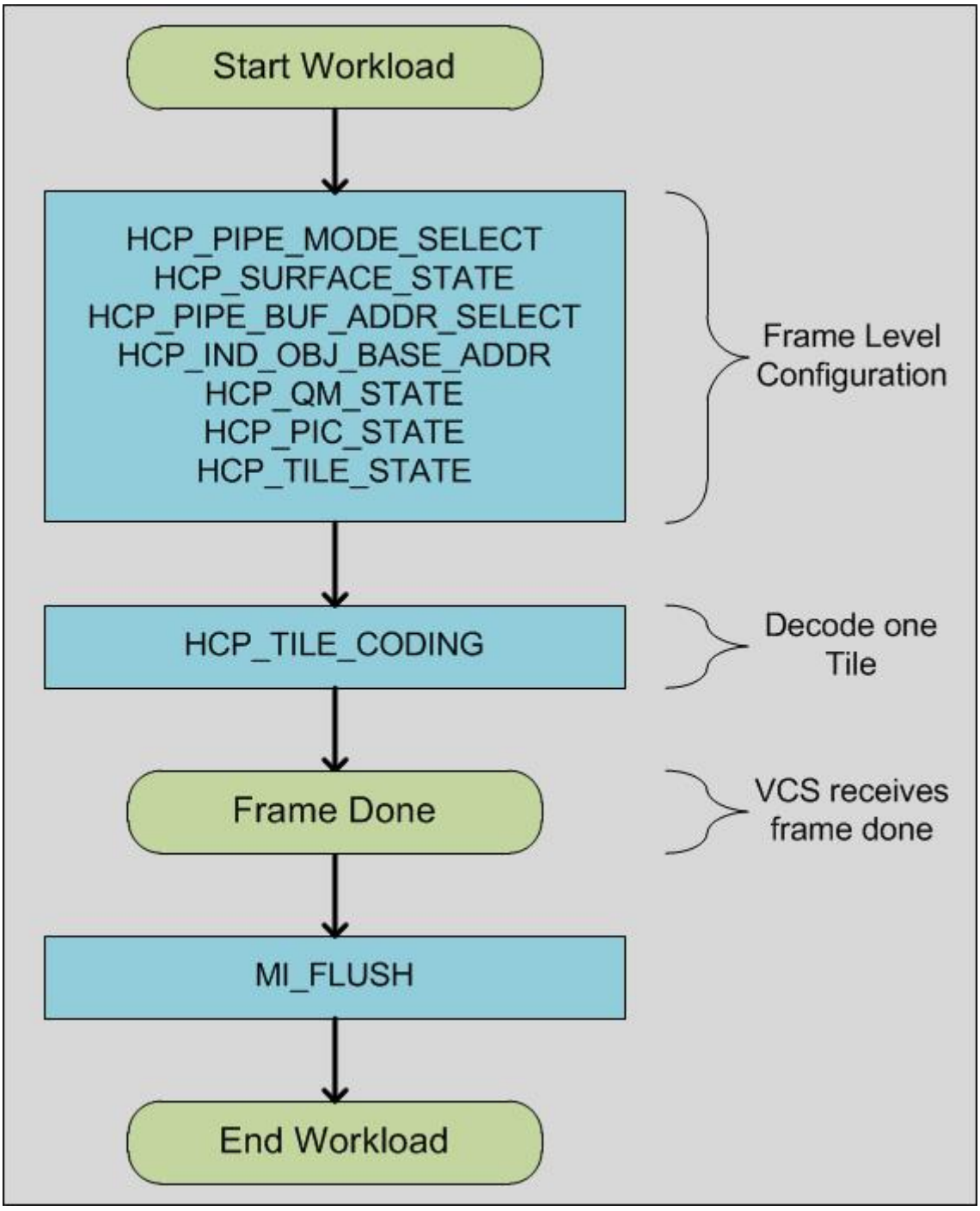
HCP Decoder Command Sequence

The long format workload for the HCP is based upon a single frame decode. There are no states saved between frame decodes in the HCP. Once the bit stream DMA is configured with the HCP_BSD_OBJECT command, and the bit stream is presented to the HCP, the frame decode will begin.

HCP Long Format Decode Workload Flowchart



The scalable decoder workload for the HCP backend pipe is based upon a single frame decode using multiple backend pipes. The frame is split into multiple "virtual" vertical (column) tiles and they are processed by multiple linked backend pipes. [NOTE: the above command sequence is still used for HCP CABAC decode and the decoded syntax elements are streamed to memory.]



HCP Encoder Command Sequence

For a single frame encoding process (w/o multiple slices per frame), the command sequence is listed below. There are no states saved between frame encoded in the HCP. There should be no other commands or context switch within a group of PAK OBJECT Commands, representing a complete slice. HCP and MFX share the same VCS, but there is no common encoding and decoding command that can be executed in both pipes, and mi_flush and MMIO commands.

------ Per Frame Level Commands

HCP_PIPE_MODE_SELECT

HCP_SURFACE_STATE

HCP_PIPE_BUF_ADDR_STATE

HCP_IND_OBJ_BASE_ADDR_STATE

HCP_FQM_STATE – issue n number of times

HCP_QM_STATE – issue n number of times

HCP_PIC_STATE

------- Per Slice Level Commands (2 cases)

------- A Frame with only 1 Slice:

HCP_REF_IDX_STATE – set to provide L0 list for a P or B-Slice

HCP_REF_IDX_STATE – set to provide L1 list for a B-Slice

HCP_WEIGHTOFFSET_STATE Command – set to provide for L0 of a P or B-Slice

HCP_WEIGHTOFFSET_STATE Command - set to provide for L1 of a B-Slice

HCP_SLICE_STATE

HCP_PAK_INSERT_OBJECT – if header present at 1st slice start

------- A group of LCUs Per Slice

HCP_PAK_OBJECT

…

HCP_PAK_INSERT_OBJECT – if tail present at frame end

MI_FLUSH – when the frame is done

------ A Frame with Multiple Slices:

HCP_REF_IDX_STATE – set to provide L0 list for a P or B-Slice

HCP_REF_IDX_STATE – set to provide L1 list for a B-Slice

HCP_WEIGHTOFFSET_STATE Command – set to provide for L0 of a P or B-Slice

HCP_WEIGHTOFFSET_STATE Command - set to provide for L1 of a B-Slice

HCP_SLICE_STATE

HCP_PAK_INSERT_OBJECT – if header present at 1st slice start of a frame

HCP_PAK_OBJECT - a group of LCUs for a slice or a frame

…

HCP_PAK_INSERT_OBJECT – if tail present at slice or frame end

HCP_REF_IDX_STATE – set to provide L0 list for a P or B-Slice

HCP_REF_IDX_STATE – set to provide L1 list for a B-Slice

HCP_WEIGHTOFFSET_STATE Command – set to provide for L0 of a P or B-Slice

HCP_WEIGHTOFFSET_STATE Command - set to provide for L1 of a B-Slice

HCP_SLICE_STATE

HCP_PAK_INSERT_OBJECT – if header present at slice start

HCP_PAK_OBJECT - a group of LCUs for a slice or a frame

…

HCP_PAK_INSERT_OBJECT – if tail present at last slice end (frame end)

MI_FLUSH – when the frame is done

MFX_STITCH_OBJECT – a generic bitstream stitching command from MFX pipe

MI_FLUSH

MI_FLUSH is not allowed between Slices. HEVC CABAC has simplified its operation from AVC. There is no longer a BSP_BUF_BASE_ADDR_STATE Command, as only a small local internal buffer is needed for BSP/BSE row store. THE HCP PAK_INSERT_OBJECT has been designed to support both inline and indirectly payload. Nevertheless, the MFX_STITCH_OBJECT command can still be used to stitch HEVC bitstreams together, and is run in the MFX pipe. No HEVC specific STITCH command is implemented. The SURFACE_STATE command for HEVC is redesigned and much simplified from that of MFX pipe.

Command Sequences with Tile Support (Gen11+)

Single Pipe Mode- Following flow chart shows the command sequence when encoding frame using a single pipe(VDbox).

Multiple Pipe Mode- When encoding a frame using multiple pipes, each pipe gets a single Tile Column or multiple Tile columns depending upon Number of tile columns to encode. Following flow chart shows commands sequence in a pipe (VDbox) when multiple pipes are used for encoding.

Memory Address Attributes

This section defines the memory address attributes for the third DWord of the HCP command buffer address.

NOTE: The first DWord defines the lower address range and the second Dword defines the upper address range in the HCP command buffer address.

MemoryAddressAttributes

HCP Pipe Common Commands

The HCP Pipe Common Commands specify the HEVC Decoder pipeline level configuration.

HCP_PIPE_MODE_SELECT (VideoCS)

HCP_SURFACE_STATE (VideoCS)

HCP_PIPE_BUF_ADDR_STATE (VideoCS)

Buffer Name	Minimum Size in CLs	Notes
Deblocking Filter Line Buffer	((picture_width_in_pixels + 31) & (-32)) » 3	Eq. ensures multiple of 4
Deblocking Filter Tile Line Buffer	((picture_width_in_pixels + 31) & (-32)) » 3	Eq. ensures multiple of 4
Deblocking Filter Tile Column Buffer	((picture_height_in_pixels + 6*pic_height_in_ctb + 31) & (-32)) » 3	Eq. ensures multiple of 4
Metadata Line Buffer (all intra slices)	(picture_width_in_pixels + pic_width_in_lcu*8 + 1023) » 9	Eq. ensures multiple of 2
CHV/A/B, /A/B/C/D: Metadata Tile Line Buffer (all intra slices)	(picture_width_in_pixels + pic_width_in_lcu *8 + 1023) » 9	Eq. ensures multiple of 2
CHV/C+, /E+: Metadata Tile Line Buffer (all intra slices)	(picture_width_in_pixels + pic_width_in_lcu *16 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Column Buffer (all intra slices)	(picture_height_in_pixels + picture_height_in_pixels *8 + 1023) » 9	Eq. ensures multiple of 2
Metadata Line Buffer (some inter slices)	(((picture_width_in_pixels+15)»4)*188 + pic_width_in_lcu*9 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Line Buffer (some inter slices)	(((picture_width_in_pixels +15)»4)*172 + pic_width_in_lcu *9 + 1023) » 9	Eq. ensures multiple of 2
CHV/A/B, /A/B/C/D: Metadata Tile Column Buffer (some inter slices)	(((picture_height_in_pixels +15)»4)*172 + picture_height_in_lcu *9 + 1023) » 9	Eq. ensures multiple of 2
CHV/C+, /E+: Metadata Tile Column Buffer (some inter slices)	(((picture_height_in_pixels +15)»4)*176 + picture_height_in_lcu *89 + 1023) » 9	Eq. ensures multiple of 2
SAO Line Buffer	(((picture_width_in_pixels » 1) + pic_width_in_ctb * 3)+15) & (-16)) » 3	Eq. ensures multiple of 2
SAO Tile Line Buffer	(((picture_width_in_pixels » 1) + picture_width_in_ctb * 6)+15) & (-16)) » 3	Eq. ensures multiple of 2
SAO Tile Column Buffer	(((picture_height_in_pixels » 1) + pic_height_in_ctb * 6)+15) & (-16)) » 3	Eq. ensures multiple of 2
Current and Collocated Motion Vector Temporal Buffer (lcu=16x16)	(((picture_width_in_pixels+63)»6)*((picture_height_in_pixels+15)»4)	Eq. ensures multiple of 2

Buffer Name	Minimum Size in CLs	Notes
Current and Collocated Motion Vector Temporal Buffer (lcu>16x16)	(((picture_width_in_pixels+31)»5)*((picture_height_in_pixels+31)»5)	Eq. ensures multiple of 2

HCP_IND_OBJ_BASE_ADDR_STATE (VideoCS)

HCP_QM_STATE (VideoCS)

Buffer Size Requirements

HEVC Buffer Size Requirements

Buffer Name	Minimum Size in CLs	Notes
Deblocking Filter Line Buffer	((picture_width_in_pixels + 31) & (-32)) » 3	Eq. ensures multiple of 4
Deblocking Filter Tile Line Buffer	((picture_width_in_pixels + 31) & (-32)) » 3	Eq. ensures multiple of 4
Deblocking Filter Tile Column Buffer	((picture_height_in_pixels + 6*pic_height_in_ctb + 31) & (-32)) » 3	Eq. ensures multiple of 4
Metadata Line Buffer (all intra slices)	(picture_width_in_pixels + pic_width_in_lcu*8 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Line Buffer (all intra slices)	(picture_width_in_pixels + pic_width_in_lcu *8 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Column Buffer (all intra slices)	(picture_height_in_pixels + picture_height_in_lcu * 16 + 1023) » 9	Eq. ensures multiple of 2
Metadata Line Buffer (some inter slices)	(((picture_width_in_pixels+15)»4)*188 + pic_width_in_lcu * 9 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Line Buffer (some inter slices)	(((picture_width_in_pixels +15)»4)*172 + pic_width_in_lcu * 9 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Column Buffer (some inter slices)	(((picture_height_in_pixels +15)»4)*256 + picture_height_in_lcu * 9 + 1023) » 9	Eq. ensures multiple of 2
SAO Line Buffer	(((picture_width_in_pixels » 1) + pic_width_in_ctb * 3)+15) & (-16)) » 3	Eq. ensures multiple of 2
SAO Tile Line Buffer	(((picture_width_in_pixels » 1) + picture_width_in_ctb * 6)+15) & (-16)) » 3	Eq. ensures multiple of 2
SAO Tile Column Buffer	(((picture_height_in_pixels » 1) + pic_height_in_ctb * 6)+15) & (-16)) » 3	Eq. ensures multiple of 2
Current and Collocated Motion Vector Temporal Buffer (lcu=16x16)	(((picture_width_in_pixels+63)»6)*((picture_height_in_pixels+15)»4)	Eq. ensures multiple of 2
Current and Collocated Motion Vector Temporal Buffer (lcu>16x16)	(((picture_width_in_pixels+31)»5)*((picture_height_in_pixels+31)»5)	Eq. ensures multiple of 2
SSE Line Buffer	(Picture_width_in_lcu + 2) « 4	

HEVC 10 bit Buffer Size Requirements

Buffer Name	Minimum Size in CLs	Notes
Deblocking Filter Line Buffer	((picture_width_in_pixels + 31) & (-32)) » **2**	Eq. ensures multiple of 4
Deblocking Filter Tile Line Buffer	((picture_width_in_pixels + 31) & (-32)) » **2**	Eq. ensures multiple of 4
Deblocking Filter Tile Column Buffer	((picture_height_in_pixels + 6*pic_height_in_ctb + 31) & (-32)) » **2**	Eq. ensures multiple of 4
Metadata Line Buffer (all intra slices)	(picture_width_in_pixels + pic_width_in_lcu*8 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Line Buffer (all intra slices)	(picture_width_in_pixels + pic_width_in_lcu *8 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Column Buffer (all intra slices)	(picture_height_in_pixels + picture_height_in_lcu * 16 + 1023) » 9	Eq. ensures multiple of 2
Metadata Line Buffer (some inter slices)	(((picture_width_in_pixels+15)»4)*188 + pic_width_in_lcu*9 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Line Buffer (some inter slices)	(((picture_width_in_pixels +15)»4)*172 + pic_width_in_lcu *9 + 1023) » 9	Eq. ensures multiple of 2
Metadata Tile Column Buffer (some inter slices)	(((picture_height_in_pixels +15)»4)*176 + picture_height_in_lcu * 89 + 1023) » 9	Eq. ensures multiple of 2
SAO Line Buffer	(((picture_width_in_pixels » 1) + pic_width_in_ctb * 3)+15) & (-16)) » **2**	Eq. ensures multiple of 2
SAO Tile Line Buffer	(((picture_width_in_pixels » 1) + picture_width_in_ctb * 6)+15) & (-16)) » **2**	Eq. ensures multiple of 2
SAO Tile Column Buffer	(((picture_height_in_pixels » 1) + pic_height_in_ctb * 6)+15) & (-16)) » **2**	Eq. ensures multiple of 2
Current and Collocated Motion Vector Temporal Buffer (lcu=16x16)	(((picture_width_in_pixels+63)»6)*((picture_height_in_pixels+15)»4)	Eq. ensures multiple of 2
Current and Collocated Motion Vector Temporal Buffer (lcu>16x16)	(((picture_width_in_pixels+31)»5)*((picture_height_in_pixels+31)»5)	Eq. ensures multiple of 2
SSE Line Buffer	(Picture_width_in_lcu + 2) « 4	

VP9 Common Commands

HCP_PIPE_MODE_SELECT

HCP_SURFACE_STATE

HCP_PIPE_BUF_ADDR_STATE

HCP_IND_OBJ_BASE_ADDR_STATE

HCP_VP9_SEGMENT_STATE

HCP_VP9_SEGMENT_STATE (optional)

HCP Common Commands

HCP_PIC_STATE (VideoCS)

HCP_TILE_STATE (VideoCS)

HCP_REF_IDX_STATE (VideoCS)

HCP_WEIGHTOFFSET_STATE (VideoCS)

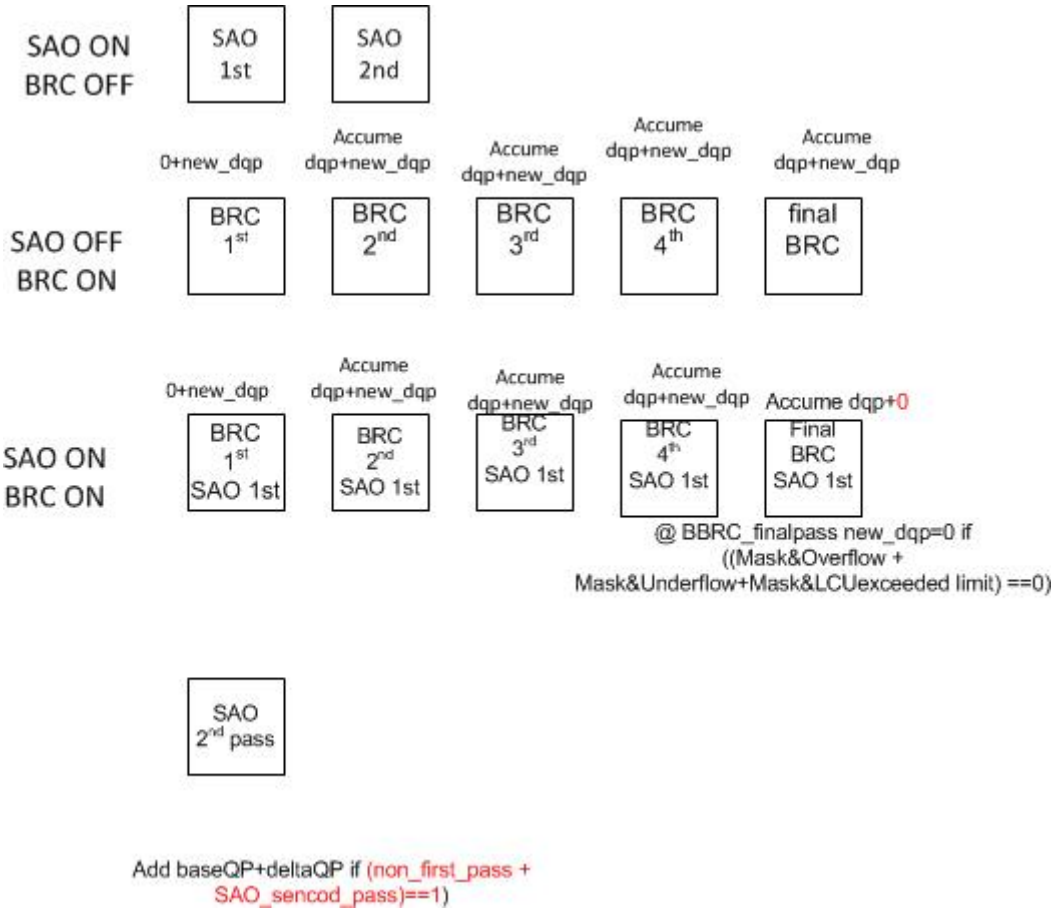
HCP_SLICE_STATE (VideoCS)

HCP Commands

The HCP Commands specify the HEVC BSD object and PAK object level configuration.

HCP_BSD_OBJECT (VideoCS)

Multipass flow during BRC and SAO



CU and Slice level stat streamOut

Along with final bitstream, HLC writes out two statistics related streamout cachelines to the memory. Streamout0 cacheline is composed of 4 quarter cachelines, each containing information on CU skip flag, coding block flag for the TUs in a CU, residual/coefficient bit count for a CU, total bit count for CU, LCU exceed limit flag. A typical streamout0 cacheline, therefore, has information on statistics for 4 CUs and lcu exceed limit flag.

Streamout1 cacheline is composed of quarter cachelines., each quarter cacheline consisting of bit count of current slice.

Pak pipeline streamout enable bit, set by HCP_PIPE_MODE_SELECT command, enables or disables the streamout.

Streamout 0: Per CU Quarter Cacheline Format

Level	Field	Width		Comment
CU	CU Skip Flag	1	qcacheline[0]	Packed in Quarter Cacheline in CU format
LCU	LCU exceed limit	1	qcacheline[1]	Packed in Quarter Cacheline in CU format (valid in last CU of LCU)
	Reserved	14	qcacheline[15:2	Reserved
CU	TU CBF Y/U/V	48	qcacheline[63:16]	Packed in Quarter Cacheline in CU format
CU	CU Coefficient Bit Count (Only residual)	18	qcacheline[81:64]	Packed in Quarter Cacheline in CU format
CU	CU Bit Count (all CU Syntax)	18	qcacheline[113:96]	Packed in Quarter Cacheline in CU format
	Reserved	14	qcacheline[127:114]	Reserved
Streamout1				
Slice	Slice Bit Count (slice header + data + tail)	32	cacheline[31:0]	
		32	cacheline[63:32]	reserved
	SlicePositionX[15:0]	16	cacheline[79:64]	
	SlicePositionY[15:0]	16	cacheline[95:80]	
		32	cacheline[128:96]	reserved

HUC Command Summary

The HUC is configured through a set of batch commands defined in the following sections. The software driver builds a frame level workload using these commands and stores these workloads in graphics memory where they are fetched by the Video Command Streamer (VCS) and presented to the HUC for processing.

The software driver is required to read the HUC disable fuse to determine if the HUC is enabled and also the software driver is required to read the HUC authentication status bit to determine if the HUC firmware was successfully loaded. If the fuse is disabled or the firmware authentication is false, then the software driver must not enable HUC batch commands to be sent to the HUC or a hang event may occur. Only when the HUC is enabled through the fuse and the firmware authentication is true, should the batch commands be sent to the HUC.

Workload Command Model

DWord0 of each command is defined in HUC DWord0 Command Definition. The HUC is selected with the **Media Instruction Opcode "Bh"** for all HUC Commands.

HUC DWord0 Command Definition

DWord	Bits	Description
0	31:29	**Command Type** = PARALLEL_VIDEO_PIPE = 3h
	28:27	**Pipeline Type** = 2h
	26:23	**Media Instruction Opcode = Codec/Engine Name** = HUC = Bh
	22:16	**Media Instruction Command** = <*see Workload Command Model* >
	15:12	**Reserved: MBZ**
	11:0	**Dword Length** (Excludes Dwords 0, 1) = <*command length*>

HUC Media Instruction Commands (Opcode=Bh)

Media Instruction Command	Command DWord0 [22:16]
HUC_PIPE_MODE_SELECT	0h
HUC_IMEM_STATE	1h
HUC_DMEM_STATE	2h
HUC_CFG_STATE	3h
HUC_VIRTUAL_ADDR_STATE	4h
HUC_IND_OBJ_BASE_ADDR_STATE	5h
Reserved	8h-1Fh
HUC_STREAM_OBJECT	20h
HUC_START	21h
Reserved	22h-7Fh

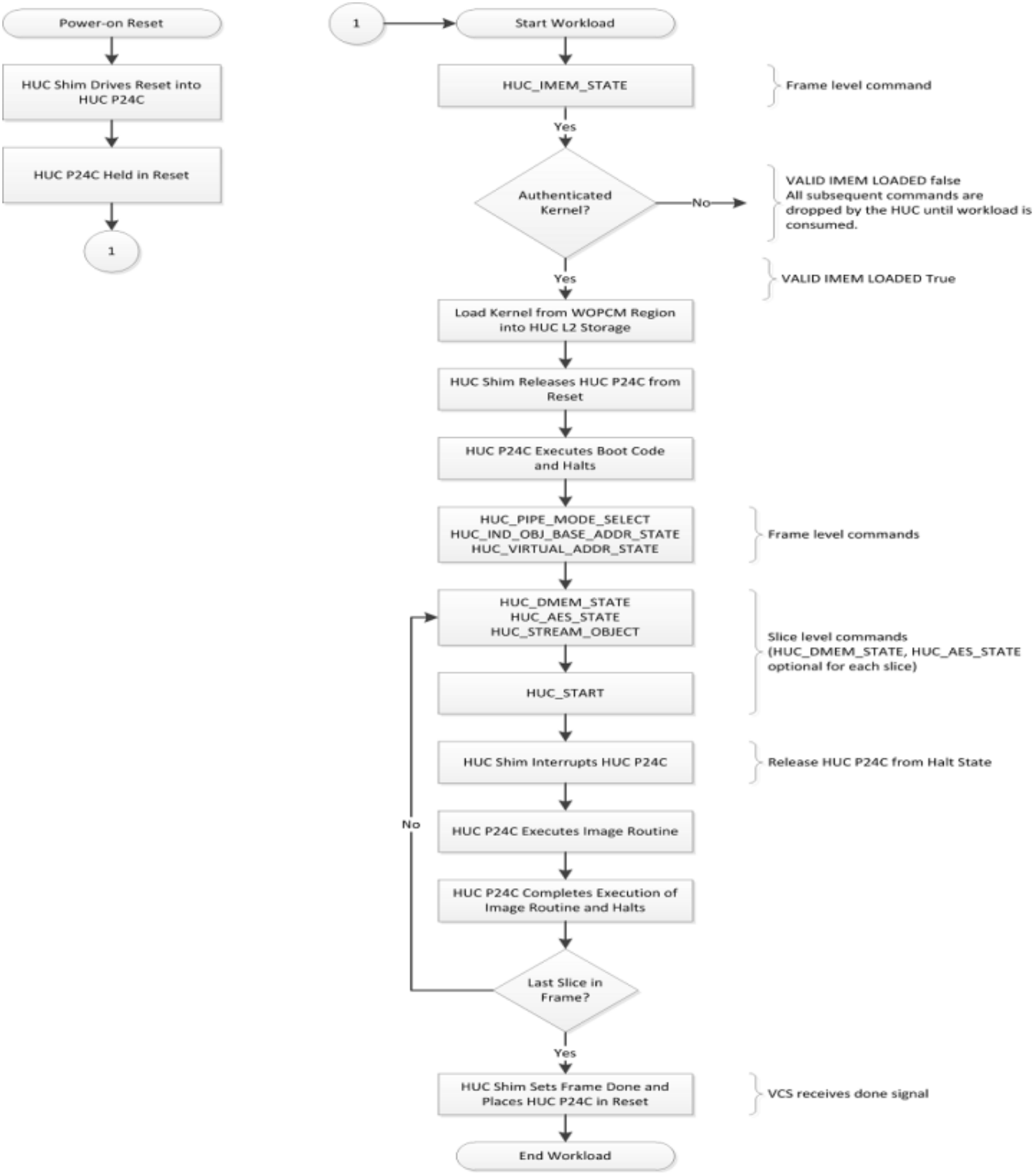
HUC Standalone Mode

The HUC operates in a standalone mode, where the HUC is active and the HCP is inactive for a workload. A typical application in this mode is short to long format conversion.

An MI_FLUSH_DW command must not be inserted in the workload between the HUC_IMEM_STATE DW command and the HUC_START DW command, as configuration of some units will be left incomplete after the HuC instruction memory (IMEM) starts execution.

The bitstream started with the HUC_STREAM_OBJECT_COMMAND may or may not reach the hucunit before the HuC FW is ready to access it. Therefore, when in Start Code detection mode, FW should first do a peekbits as a blocking command. This will pause the p24cunit until data is in the HuC's BSD. This allows the HuC's barrel shifter to get loaded with the first few valid bytes of data. In Start code detection mode, these first few bytes may be the start code prefix. So, after doing the peekbits command, FW should check the HUC_BSD_SCD_STATUS register to see if a start code was detected. When in Dashmode, the size of the length must be written to the HUC_BSD_CONTROL register first, then do a peekbits.

HUC Standalone Mode Workload Flowchart



HUC Commands

The HUC Commands specify the HUC level configuration.

HUC_PIPE_MODE_SELECT (VideoCS)

HUC_IMEM_STATE (VideoCS)

HUC_DMEM_STATE (VideoCS)

HUC_CFG_STATE (VideoCS)

HUC_VIRTUAL_ADDR_STATE (VideoCS)

HUC_IND_OBJ_BASE_ADDR_STATE (VideoCS)

HUC_STREAM_OBJECT (VideoCS)

HUC_START (VideoCS)

HEVC Error Concealment

The HCP implements an error concealment policy, which is always enabled and cannot be disabled. The objective is that the HCP will always complete a frame/field workload by either decoding the bit stream normally until it finishes the workload or by concealing blocks until the slice or workload is completed. It should never be allowed to hang.

Error concealment, implemented by the HCP hardware, is configured for each slice in the HCP_BSD_OBJECT command. The following information in the HCP_BSD_OBJECT command is utilized for error concealment.

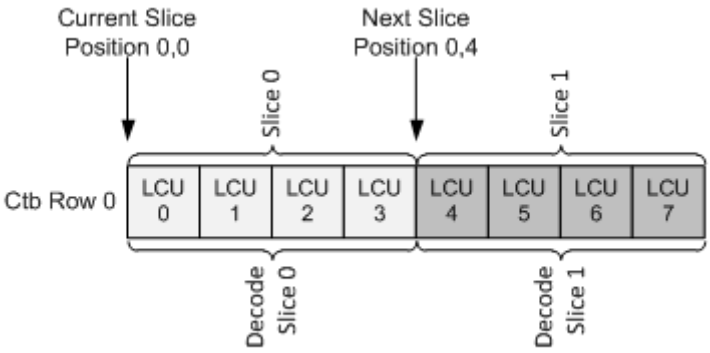
- **SliceStartCtbY, SliceStartCtbX:** The current slice position specified in Ctb coordinates.
- **NextSliceStartCtbY, NextSliceStartCtbX:** The next slice position specified in Ctb coordinates. If the current slice is the last slice in the picture, the next slice values are set to (0,0).
- **LastSliceofPic:** Indicates that the current slice is the last slice in the picture.
- **slice_type:** Indicates the picture type: I, P or B.

The host software will remove all extra slices in the picture. The HCP will not be given a workload that includes extra slices beyond the picture. The last slice in the picture will always be marked by the host software.

The host software will remove any overlapping slices in the picture. The HCP will not be given a workload that includes overlapping slices in the picture.

A HCP_BSD_OBJECT command will include the current slice position and the next slice position. For non-errored streams, it is guaranteed that the slice bit stream will be decoded by the HCP starting from the current slice position through to the Ctb (inclusive) adjacent to the Ctb indicated by the next slice position. *HEVC Error Concealment* illustrates the example of a non-errored stream decode starting with XXX.

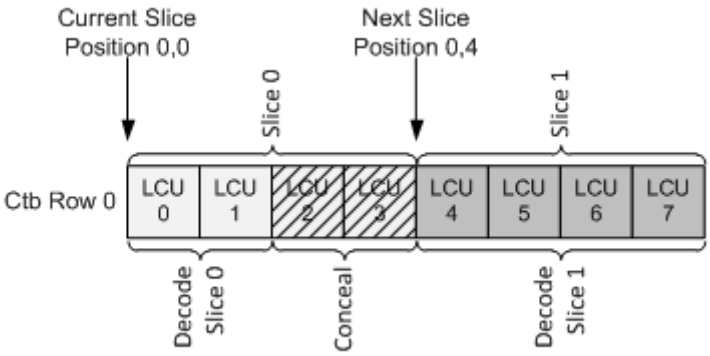
HEVC Slice Decode for Non-errored Stream Cases



For error stream cases where the next slice position does not align itself with the last successfully decoded Ctb in the current slice, the HCP will conceal Ctbs from the last decoded Ctb in the current slice through to the last Ctb prior to the Ctb indicated by the next slice position. If the error occurs such that the current decoded Ctb cannot be decoded, the HCP will ensure that the current Ctb is written out

by any means before writing out concealed Ctbs for the remaining Ctbs in the current slice. In the case of the last slice in a picture, the HCP will conceal Ctbs from the last decoded Ctb in the current slice through to the last Ctb position in the picture indicated by the resolution of the picture in the HCP_PICT_STATE command. *HEVC Error Concealment* illustrates the case described.

HEVC Slice Decode for Missing Blocks in a Slice



Since the host software removes overlapping slices, the next slice position will never be equal to or less than the current slice position.

A concealed Ctb for an I-slice is constructed by the HCP specifying the Intra_Planar prediction mode for the Ctb.

A concealed Ctb for a P-slice is constructed by the HCP specifying the skip_flag.

A concealed Ctb for a B-slice is constructed by the HCP specifying the skip_flag.

HEVC Register Definitions

The Message Channel Interface is a read-only bus used to access the HCP and HUC status registers. All registers are 32 bits where reserved bits return a value of zero and subtractive-decode is used to return 0x0000 for all register holes. The Unit ID is 28h. For HCP, the address range is 0x0001E900h to 0001E9FFh and for HUC, the address range is 0x0000D000h to 0000D7FFh.

Register Attributes Description

Host Register Attributes gives the defined register tags and their description.

Host Register Attributes

Tag	Name	Description
R/W	Read/Write	Bit is read and writeable.
R/SW	Read/Special Write	Bit is readable. Write is only allowed once after a reset.
RO	Read Only	Bit is only readable, but writes have no effects.
WO	Write Only	Bit is only writeable, reads return zeros.
RV	Reserved	Bit is reserved and not visible. Reads will return 0, and writes have no effect.
NA	Not Accessible	This bit is not accessible.

HCP Decoder Register Map

Register	Address	Default	Description
HCP_DEC_STATUS	1E900h	00000000H	HCP decode status
HCP_CABAC_STATUS	1E904h	00000000H	HCP CABAC status
HCP_LAST_POSITION	1E908h	00000000H	Last row & column position of the decoder
HCP_PMU_STATUS	1E90Ch	00000000H	PMU counter overflow status
HCP_PMU_LUMA_CNTR	1E910h	00000000H	Luma cache miss event counter
HCP_PMU_CHROMA_CNTR	1E914h	00000000H	Chroma cache miss event counter
HCP_PMU_FRAME_DEC_ACT_CNTR	1E918h	00000000H	Frame decode active event counter
HCP_PICTURE_CHECKSUM_CIDX0	1E91Ch	00000000h	Picture checksum cIdx0
HCP_PICTURE_CHECKSUM_CIDX1	1E920h	00000000h	Picture checksum cIdx1
HCP_PICTURE_CHECKSUM_CIDX2	1E924h	00000000h	Picture checksum cIdx2

HCP Decoder Register Descriptions

The HCP implements the following MMIO registers. A description of the register including its address and DWord descriptions are provided.

HCPDecodeStatus (VideoCS)

HCPCABACStatus (VideoCS)

HCPLastPosition (VideoCS)

HCPPMUStatus (VideoCS)

HCPPMULumaCacheMissCounter (VideoCS)

HCPPMUChromaCacheMissCounter (VideoCS)

HCPPMUFrameDecodeActiveCounter (VideoCS)

HCPPictureChecksumcIdx0 (VideoCS)

HCPPictureChecksumcIdx1 (VideoCS)

HCPPictureChecksumcIdx2 (VideoCS)

HCP Encoder Register Map

These are MMIO register definitions for encoder.

MMIO Register Name	Description	Read/Write	Address	Mode
HCP Frame Performance count	Number of clocks spent decoding/encoding a frame	RO	1E960h	Dec/Enc
HCP Memory Latency count1	Reference picture read latency - min and max	RO	1E968h	Dec/Enc
HCP Memory Latency count2	Reference picture read latency - Accumulative (used for compute AVE latency)	RO	1E96Ch	Dec/Enc
HCP Memory Latency count3	Row-store memory read latency - min and max	RO	1E970h	Dec/Enc
HCP Memory Latency count4	Row-store memory read latency - accumulative (used to compute AVE latency)	RO	1E974h	Dec/End
HCP Memory Latency count5	PAK Object read latency min/max	RO	1E978h	Enc
HCP Memory Latency count6	Source Pixel read latency min/max.	RO	1E97Ch	Enc
HCP_DEC_BIN_CT	HCP Frame Number of BINs total number of Bins decoded	RO	1E980h	Dec
HCP Motion Comp read Count	Total number of CL memory accesses per frame	RO	1E984h	Dec/Enc
HCP Motion Comp MISS Count	Total number of CL HITs per frame	RO	1E988h	Dec/Enc
Reserved	Reserved	-	1E98Ch – 1E99Ch	Enc
HCP_BITSTREAM_BYTECOUNT_FRAME	Total Bitstream Output Byte Count register per Frame	RO	1E9A0h	Enc
HCP_BITSTREAM_BYTECOUNT_FRAME_NO_HEADER	Reported Bitstream Output Byte Count without header per Frame Register	RO	1E9A4h	ENC
HCP_BITSTREAM_SE_BITCOUNT_FRAME	Bitstream Output total Byte Count for syntax elements (total byes of MB	RO	1E9A8h	Enc

MMIO Register Name	Description	Read/Write	Address	Mode
	data from SEC per frame)			
HCP_CABAC_BIN_COUNT_FRAME	Bitstream Output total bin count per frame	RO	1E9ACh	Enc
HCP_CABAC_INSERTION_COUNT	Bitstream Output CABAC Insertion Count Register	RO	1E9B0h	Enc
HCP_MINSIZE_PADDING_COUNT	Bitstream Output Minimal Size Padding Count Register	RO	1E9B4h	Enc
HCP_IMAGE_STATUS_MASK	Image status(flags)	RO	1E9B8h	Enc
HCP_IMAGE_STATUS_CONTROL	Suggested data for next frame in multi-pass	R/W	1E9BCh	Enc
HCP_QP_STATUS_COUNT	Overall adjusted delta QP via multi-pass, Sum of QPY for all LCU of the frame	RO	1E9C0h – 1E9C4h	Enc
HCP_SLICE_COUNT	Number of Slices in a frame	RO	1E9C8h	Enc
Reserved	Reserved	-	1E9CCh	Enc
HCP_UNIT_DONE	Unit Done	RO	1E9D8h	Enc
HCP_VDEncmode_Timer	Reports the time budget overflow along with the start LCU position of the timeout.	RO	1E9DCh	Enc
Reserved	Reserved	-	1E9E0h – 1E9FFh	Enc
Reserved	Reserved	-	1EA00h - 1EAFFh	Enc

HUC Status Register Descriptions

Following are HUC Status Register Descriptions:

HUC_STATUS - HEVC Microcontroller Status

HUC STATUS 2 - HEVC Microcontroller Status

UKERNEL_HDR_INFO - uKernel Header Info

JMP_DEST - Jump Location

MIA_IDLE - Minute IA Idle Status

SHIM_CTRL_1 - SHIM Control 1

SHIM_ERR_TRAP - SHIM Error Record

P24C_RESET_CTRL - PC24C Reset Control

PERF_CNTR_0 - PERFORMANCE COUNTER 0

PERF_CNTR_1 - PERFORMANCE COUNTER 1

PERF_CNTR_2 - PERFORMANCE COUNTER 2

PERF_CNTR_3 - PERFORMANCE COUNTER 3

P24C_INSTRUCTION_POINTER - Shadow of the Minute IA EIP

LAST_GFXMEM_FETCH_ADDR - Address of the last graphics memory fetch

LAST_WOPCM_FETCH_ADDR - Address of the last WOPCM memory fetch

LAST_SRAMCODE_FETCH_ADDR - Address of the last SRAM code fetch

LAST_SRAMDATA_FETCH_ADDR - Address of the last SRAM data fetch

HUC DMA Register Descriptions

The HUC DMA uses two 64-bit registers (4 DWord Registers) to indicate the 48-bit Source and Destination addresses along with fetch type indication, etc. Bit_1 of the DMA Control Register pins DMA Address Register 0 to Source addressing or Destination, and vice versa for the DMA Address Register 1. By default, DMA Address Register 0 is assigned to Source addressing and DMA Address Register 1 to Destination addressing.

Programming Notes:

- The lower 6 bits of addressing of both Source and Destination addresses must be same. There is no provision for barrel rotation across byte locations, during a DMA Transfer.
- The following restrictions shall be followed for placement of uOS and uApps.
  - uOS and uApps are always located on a 64-byte aligned address.
  - uOS and uApps hash data is always on a 64-byte aligned address.
- uApps are not automatically loaded into SRAM by HW. uKernel must explicitly copy a uApp into SRAM from WOPCM when using it.
- The hardware shall not allow the DMA to begin if the SW programs the DMA WOPCM destination address and byte count to values such that a copy will exceed the upper bounds of the WOPCM area. The hardware simply ignores the START and triggers an error.
- Once the ukernel and the uApps have been loaded successfully into the WOPCM area, the hardware shall not allow DMA operations to overwrite them.
- Before programming the DMA engine to access memory in the Per Process GTT address space, the HUC SW must setup the PPGTT by programming registers: These registers are located in the HUC_PM unit (offsets: 0xC3B8 - 0xC3F0):
  - CTXT_INFO
  - PDP0, PDP1, PDP2, PDP3
  - PPGTT_ENABLE

Following are HUC DMA Register Descriptions:

DMA_EXT_LOW_ADDRESS - DMA Source Address Low Bits

DMA_EXT_HIGH_ADDRESS - DMA Source Address High Bits

DMA_INT_ADDRESS - DMA Destination Address

DMA_COPY_SIZE - DMA Copy Size

DMA_CTRL - DMA Control

HUC Control Register Descriptions

The HUC Control registers are not accessible by the host and are thus not mapped to MMIO address space.

Acronyms and Abbreviations

The table below defines acronyms and abbreviations used in this document.

Acronyms

Acronym	Meaning
AAC	Advanced Audio Coding — part of the MPEG specification, AAC is the latest development in audio compression. It provides higher-quality audio reproduction than MPEG-1 Layer 3 (MP3), while requiring nearly 50% less data. It is defined in ISO/IEC 13818-7.
ADSL	Asymmetrical Digital Subscriber Line — an asymmetrical DSL technology that takes advantage of the one-way nature of most multimedia communication, and provides much faster data rates for downstream (to the subscriber) then the upstream.
API	Application Programming Interface — a set of routines used by an application program to request and carry out low-level services performed by the operating system.
ARGB	Alpha Red Green Blue — color channel components.
ARIB	Association of Radio Industries and Business — designated by the Ministry of Public Management, Home Affairs, Posts and Telecommunications (MPHPT) in Japan. ARIB members include broadcasters, radio equipment manufacturers, telecommunication operators, and related organizations.
ASP	Advanced Simple Profile – MPEG4-2
ATSC	ATSC Advanced Television Systems Committee – an organization in US that establishes and promotes technical standards for advanced television systems, such as digital television (DTV).
BDU	Bit-stream Data Unit
BIST	Built In Self Test
BPP	Bits Per Pixel
BSD	Byte Stream Decoder
CA, CAM	Conditional Access, Conditional Access Module – the removable descrambling module implemented in digital cable or satellite television system. The data flows through the module, which can have any proprietary scrambling algorithm implemented, yet maintaining system interface compatibility. The CAMs are usually provided by the operators in the TV network.
CPU	Central Processing Unit
DAA	Direct Access Arrangement
DAC	Digital-to-Analog Converter
DDA	Digital Difference Analyzer
DDS	Direct Digital Synthesizer
DPB	Decoded Picture Buffer. This buffer holds the decoded pictures for reference and for output along with the currently decoding picture. This differs from the DPB in the standard, which only holds the decoded pictures for reference.
DVB	Digital Video Broadcasting — a set of open worldwide standards that define digital broadcasting using existing satellite, cable, and terrestrial infrastructures. It uses MPEG-2 specification as a universal foundation and expands it with DVB data structures and processes DVB-compliant digital broadcasting and equipment is widely available to consumers and is indicated with the DVB

Acronym	Meaning
	logo.
DVB-S	Satellite television DVB standards, based on QPSK and 8-DPSK modulation.
DVB-T	Terrestrial television DVB standards, based on 2k and 8k OFDM modulation.
DVD	Digital Versatile Disc
DVD-R	Recordable DVD. Since different disk formats are currently in use, including DVD-R,DVD+R, they are collectively mentioned as DVD-R in this document
DVI	Digital Visual Interface standard (EIA/CEA-861A). The standard defines a method for sending digital video signals over DVI and OpenLDI interface specifications. The standard is fully backward compatible with earlier DVI standards. New features include carrying auxiliary video information, such as aspect ratio and native video format information.
DSL xDSL	Digital Subscriber Line – transmission of data over copper telephone lines capable of bringing high-bandwidth to subscribers. Many flavors of DSL are currently in use, which are collectively called xDSL throughout the document.
DSP	Digital Signal Processor
DST	Destination
DWord	A 32-bit word
ES	Elementary Streams — the raw output of an encoder, containing only what is necessary for a decoder to approximate the original picture or audio.
FIFO	First in First Out
FIR	Finite Impulse Response
FPU	Floating Point Unit
FW	Firmware running on the decoder controller, as used in Volume 4 of the *Olo River Plus Silicon EAS*
IDR	Instantaneous Decoding Refresh
IEEE 1394 1394	IEEE 1394 or iLink* or FireWire* An IEEE electronics industry standard for connecting multimedia and computing Up to 63 devices can be attached to your PC via a single plug-and-socket connection.
IEEE 802.11 802.11	The Institute for Electronics and Electrical Engineers (IEEE) wireless network specification. 802.11g and 802.11a networks can transmit payload at the rates in excess 34Mbits/s and allow for the wireless transmission at distances from several dozen to several hundred feet indoors.
IF	Intermediate Frequency — the fixed, relatively low-frequency carrier to which current programs are ported by the tuner.
GMCH	Graphics and Memory Control Hub — a chip that connects the IA processor to memory and other components in PC.
HDD	Hard Disk Drive — magnetic mass storage device used in media centers for audiovisual program recording.
HDMI	High Definition Multimedia Interface (HDMI). This interface is used between any audio/video source, such as a set-top box, DVD player, or A/V receiver, and an audio or video monitor, such as a DTV. HDMI supports standard, enhanced or high-definition video, plus multi-channel digital audio on a single cable. The format transmits all ATSC HDTV standards and supports eight-

Acronym	Meaning
	channel digital audio (at up to a 192kHz sampling rate), with bandwidth to spare for future enhancements.
HDTV	High-Definition Television — HDTV specifically refers to the highest-resolution formats of the 18 total DTV formats, true HDTV is generally considered to be 1,080-line interlaced (1080i) or 720-line progressive (720p).
HSR	Hidden Surface Removal
HW	Hardware
I/F	Interface
IEEE	IEEE 32-bit Floating Point number format representation
ISP	Image Synthesis Processor — A collective term to describe all components of the hidden surface removal operation within the PowerVR architecture.
LOD	Level Of Detail — used in texturing calculations.
LSB	Least Significant Bit
LUT	Look-up table
MBAFF	Block Adaptive Field Frame mode
MFD	Multi-Format Decoder
MMU	Memory Management Unit
MMMC	Multi-port, Multi-channel Memory Controller
MSA	Intel Micro Signal Architecture — microprocessor architecture combining the features of microcontroller and digital signal processor. MSA is used here as a synonym of the processor core used in Olo River Plus
MSB	Most Significant Bit
MPEG	Motion Picture Experts Group – Organization that develops standards for digital video and digital audio compression.
MPR	Inter Prediction Module
NAL	Network Abstraction Layer
NAL unit	Syntax structure in a H.264 stream
NTSC	National Television System Committee, North American 525-line analog broadcast TV standard.
NIM	Network Interface Module – the integrated tuner and digital demodulator in the (satellite) TV systems. The DVB NIMs output MPEG transport stream.
NOP	No operation
OEM	Original Equipment Manufacturer
OGL/OpenGL	Open GL application programming interface
PAL	Phase Alternation Line - TV standard used in Europe. PAL uses 625 lines per frame, a 25 frames per second update rate and YUV color encoding. The number of visible pixels for PAL video is 768 x 576.
PCI	Peripheral Component Interconnect bus, a bi-directional bus defined in PCI 2.x specification
PES	Packetized Elementary Streams — packetized streams are the ES streams arranged in data packets with PES header starting every packet. The syntax of the ES and PES is defined in MPEG. See

Acronym	Meaning
	definition for ES.
PIP	Picture In Picture display mode
POD	Point of Deployment conditional access module — the removable conditional access module defined in the OpenCable* specification in US.
PPS	Picture Parameter set
PTS	Presentation time stamp
PVR	Personal Video Recorder, also PDR or personal digital recorder — an interactive TV-recording device that records programs in digital format and allows users to search for/record shows based on type (for instance all basketball games or all episodes of a particular program). Users can also pause, rewind, stop, or fast-forward live programs with only a small time lag.
PWL	Piece-wise Linear
PXD	Pixel Decoder Module
RF	Radio Frequency – usually, modulated carriers which can be directly received by the tuners of TVs or radio receivers
RISC	Reduced Instruction Set Computer
RHW	Reciprocal Homogenous W — W is a 3-D coordinate representation like X Y Z
RSB	Row Store Buffer
RTL	Register Transfer Language/Level
SEI	Supplementary Enhancement Information
SIF	Semaphore Interface Module
SIMD	Single Instruction Multiple Data
SMPTE	Society of Motion Picture and Television Engineers
SOC	System on chip
SP	Simple Profile – MPEG4-2
SPS	Sequence Parameter set
SRC	Source
SDTV	Standard-Definition Television — a digital television system that is similar to current analog TV standards in picture resolution and aspect ratio. Typical SDTV resolution is 480i or 480p.
STB	Set Top Box — a device that effectively turns a television set into an interactive Internet device and/or allows the television to receive and decode digital television (DTV) broadcasts.
TA	Tile Accelerator
TS	MPEG-2 Transport Stream — a sequence of 188-byte packets carrying the multi-program audiovisual data
TSP	Texture Shading Processor — a collective term to describe all components of the texture, shading and pixel blending operations within the PowerVR architecture.
VCL	Video Coded Layer
VCXO	Voltage Controlled Crystal Oscillator
VGP/ VGP Lite	Vertex Geometry Processor

Acronym	Meaning
VLC	Variable length coded. This refers to the collection of coding techniques that are used in VC1, and include CABAC, CAVLC and Exp-Golomb.
VOL	Video Object Layer
VOP	Video Object Plane
WAN	Wide Area Network
WSS	Wide Screen Signaling
XDS	Extended Data Services — data services sending data in line 21/283 of the analog NTSC TV signal
XSI	Intel® XScale® System Interconnect
X, Y, Z, W	3-D coordinate representations
YUV	YUV texture format, primarily for video formats