WAVE420L HEVC Encoder IP

Programmer's Guide

Version 1.8.14

Confidential
StarFive Inc.

The information contained in these documents is confidential, privileged and only for the information of the intended recipient and may not be used, published, or redistributed without the prior written consent of Chips&Media Corporation.

WAVE420L HEVC Encoder IP: Programmer's Guide

Version 1.8.14
Copyright © 2021 Chips&Media, Inc. All rights reserved

Revision History

Date	Revision	Change
2021/7/28	1.8.14	Release version

Proprietary Notice

Copyright for all documents, drawings and programs related with this specification are owned by Chips&Media Corporation. All or any part of the specification shall not be reproduced nor distributed without prior written approval by Chips&Media Corporation. Content and configuration of all or any part of the specification shall not be modified nor distributed without prior written approval by Chips&Media Corporation.

The information contained in these documents is confidential, privileged and only for the information of the intended recipient and may not be used, published, or redistributed without the prior written consent of Chips&Media Corporation.

Address and Phone Number

Chips&Media
NC Tower1, 7~8th FL, 509, Teheran-ro, Gangnam-gu, 125-880
Seoul, Korea
Tel: +82-2-568-3767
Fax: +82-2-568-3768

Homepage: *http://www.chipsnmedia.com*

Confidential
StarFive Inc.

Table of Contents

Confidential StarFive Inc.

Confidential
StarFive Inc.

Confidential StarFive Inc.

Confidential StarFive Inc.

Confidential
StarFive Only

Confidential
StarFive Inc.

Confidential
StarFive

Confidential
StarFive

Confidential
StarFive

Confidential
Starfive Inc.

Confidential
Starfive Inc.

List of Figures

Confidential
StarFive Inc.

List of Tables

Confidential
Chips&Media Inc.

Chips&Media Confidential

Confidential
Chips&Media inc.

Confidential

Confidential

Confidential
Chips&Media Inc.

Confidential
Chips&Media Inc.

Confidential

Confidential
Chips&Media

Confidential

Confidential
Chips&Media Inc.

Confidential
Chips&Media

Confidential
StarFive Inc.

List of Examples

Confidential
StarFive Inc.

Preface

This preface introduces the WAVE420L Programmer's Guide and its reference documentation. It contains the following sections:

1. About This Document

This document is the programmer's guide for WAVE420L HEVC Codec IP .

1.1. Intended audience

This document is for experienced hardware and software engineers who want to implement host applications by using the host interface registers.

1.2. Scope

This document mainly describes the host interface registers that are used for communication between a host and the VPU (Video Processing Unit) such as host commands, response, or temporal command arguments. It also covers the interrupt and video operating control flow, and sample application codes using API functions.

Note This is a standard document for CODA9 Video IP Core. So parts of the document may include standard information irrelevant to your IP.

Confidential
StarFive Inc.

1.3. Typographical conventions

The following typographical conventions are used in this document:

bold Highlights signal names within the text, and interface elements such as menu names. May also be used for emphasis in descriptive lists where appropriate.

Italic Highlights cross-references in blue, file names, and citations.

`Typewriter` Denotes example source codes and dumped characters or text, data types, function, register, and flag names.

2. Further reading

This section lists documents that are related to this product.

2.1. Other documents

- *WAVE420L Datasheet*
- *WAVE420L Verification Guide*
- *WAVE420L API Reference Manual*

Chapter 1
HOST INTERFACE

1.1. VPU Control Scheme

This section presents general description of host interfaces that are provided for host processor to control VPU.

1.1.1. Communication Models

VPU requires a dedicated path for exchanging messages and data with host processor. VPU uses a set of host interface registers for receiving a command from host processor or sending a response to host processor. The host interface registers can be accessed on AMBA APB (Advanced Peripheral Bus). Also VPU uses certain memory regions on SDRAM for storing data that is shared with host processor. This kind of dedicated memory can hold bitstream data and frame data which are accessible on ABMA AXI bus by both VPU and host processor.

A detailed diagram illustrating this scheme is presented in *Figure 1.1, "Exchanging data and messages between host and VPU"*.

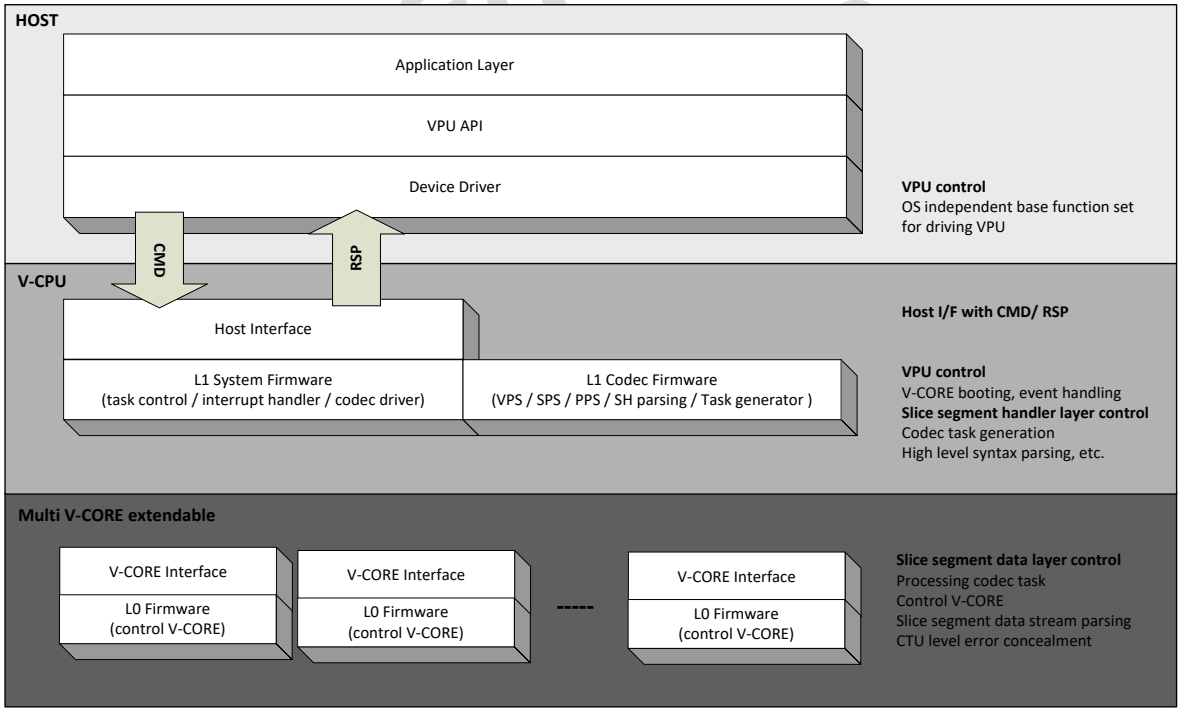


Figure 1.1. Exchanging data and messages between host and VPU

All bitstream data and picture data are directly accessed by host processor and VPU. The related information about all those data transfers are exchanged on host interface register via commands and responses. In order to do this, host interface of VPU provides a set of registers accessible from host processor.

Host interface registers are classified into two groups: control registers and command registers. The control registers are mostly used to give host processor the internal status or hardware information about VPU. The command registers are used to issue actual video commands and responses.

Basically, VPU provides a set of pre-defined commands and their corresponding responses. L1 firmware running on VPU is designed for these given set of commands and responses.

1.1.2. Data Handling

Host processor or VPU can directly handle all transactions regarding to output picture data or stream data on the external SDRAM. To assure safe transactions between host processor and VPU, VPU works with the command that is issued by host processor and after the job returns the result of command or other required information on the host interface register.

Generally, all of these transactions are one-directional transactions, which means one only writes data and the other only reads the written data on a single data buffer like stream buffer case. So by using a pair of read pointer and write pointer, all those transactions can be controlled very easily and safely. Frame buffers associated with picture decoding are managed by VPU only.

Besides the frame buffer and stream buffer, VPU requires secured memory regions for video processing, which are called a Code Buffer, Work Buffer and Temp Buffer. They are used for firmware code, static data and temporal data manipulation respectively. These buffers are only accessible by VPU.

Note There is a restriction that the base buffer addresses should be aligned in a 4KB unit, and 0xFFFF0000 to 0xFFFFFFFF (based on 32-bit) should not be assigned for the buffer addresses.

1.2. Host Interface Registers

1.2.1. Overview of host interface registers

VPU provides a set of commands for controlling video operations as well as for corresponding responses on a frame-by-frame basis. Host and VPU can exchange data on Host Interface Memory that is memory-mapped through APB. The following table shows the range of APB offsets for access to VPU.

Table 1.1. APB Offsets for access to VPU

APB start	APB end	Region	Note
0x0000	0x01FF	Host Interface Register	Where commands and responses are given from both sides VPU and host processor
0x0200	0x3FFF	Product Information	0x1040 and 0x1044 are connected to inform product name and code. Others remain to be reserved. [a]
0x4000	0x7FFF	Internal Peri	(Optional) Debugging purpose only
0x8000	0xFFFF	vCORE Access	(Optional) Debugging purpose only

[a]The register 0x1040 returns the product name, WAVE. The register 0x1044 presents its product code, i.e 4102 or 4120.

It is recommended to only access 0x0000 to 0x1FF for interfacing with VPU. Other ranges are used for internal access. Host processor can give any command to VPU by setting COMMAND register(0x100) and can also send an interrupt request to VPU for the command issued through VPU_HOST_INT_REQ register(0x38).

But before giving a command, host processor needs to make sure whether VPU is currently working through VPU_BUSY_STATUS(0x070) register. Most commands cannot be valid in VPU busy status (VPU_BUSY_STATUS=1) and if a command is given in an invalid state, VPU immediately returns an error. After confirming that VPU_BUSY_STATUS is 0, host processor must set VPU_BUSY_STATUS to 1 and then issue a command for secure operation.

Host processor can receive any response to the command through the host interface register and get informed with an interrupt whether the command has been done when VPU_VINT_ENABLE register (0x48) is set.

Confidential StarFive Inc.

Figure 1.2, "Host Interface Memory Map" represents the APB connected memory map for host's access to VPU.

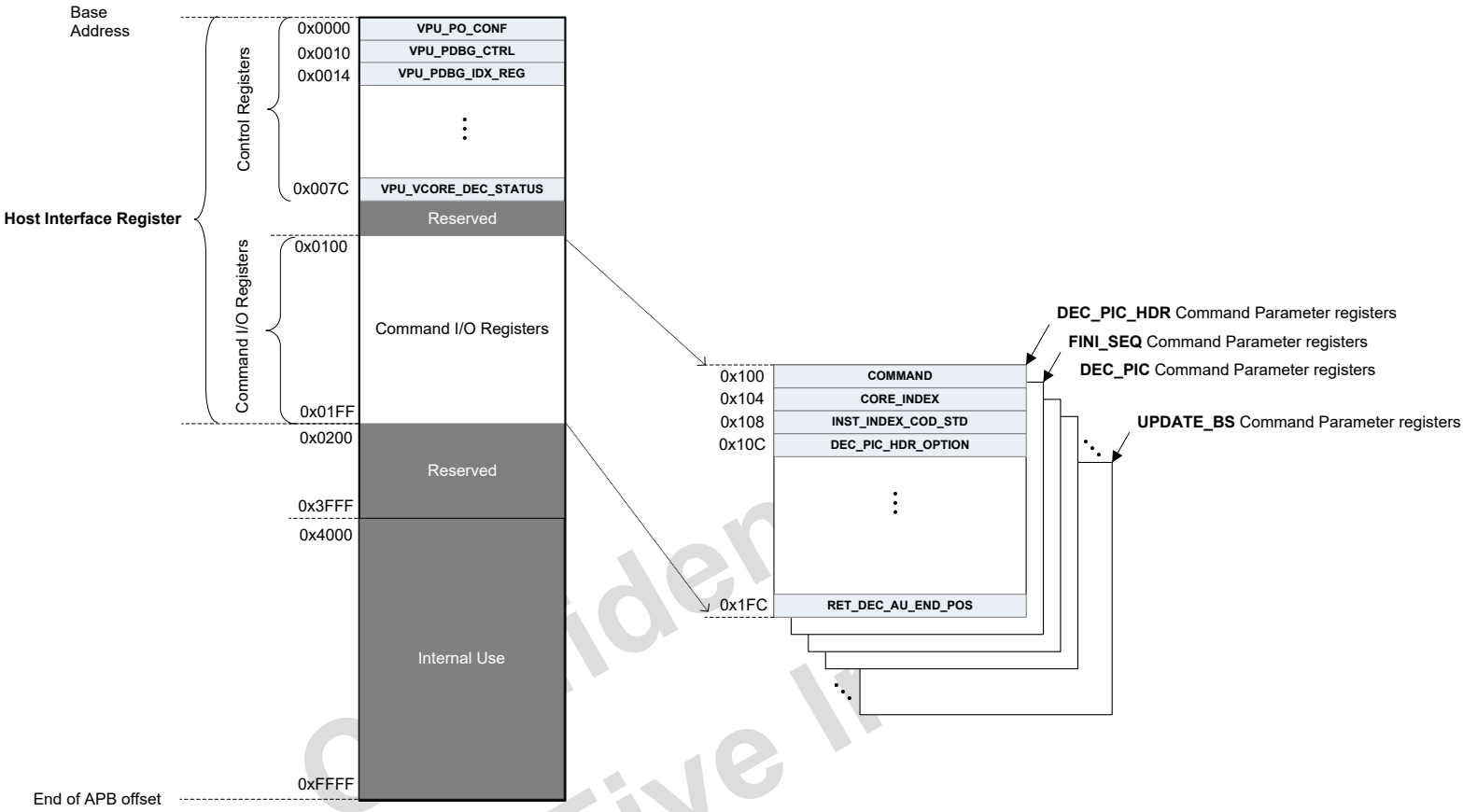


Figure 1.2. Host Interface Memory Map

Among the offset ranges, 0x0000 to 0x01FF are host interface registers, and these are partitioned into two categories as follows:

Control Registers

Host interface registers in this category (0x0000 to 0x07C) are used to update or show the VPU status. Host processor can use most of these registers for initializing VPU during boot-up.

Command I/O Registers

Host interface registers in this category (0x0100 to 0x01FF) are overwritten or updated whenever a new command is given to VPU from Host processor. All the commands with input arguments and all the corresponding responses with return values are delivered through these registers.

VPU shares this register region for parameters of the VPU command. It can be used for any of the commands, and certain registers might have different meaning according to the command that has been issued by host processor. For instance, the base address + 0x010C means initialization options for VPU when host processor sets INIT_VPU command, while this address is used to specify decode header options for DEC_PIC_HDR command.

Note Detailed information about each of the command I/O registers, please see the *Section 1.2.3, "Register Descriptions"*.

1.2.2. Summary of Host Interface Registers

1.2.2.1. Summary of Control Registers

Table 1.2. Summary of Control Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000000	RW	0x0	*VPU_PO_CONF*	Power On Configurations
0x0000000C	RW	0x0	*VPU_PDBG_STEP_MASK*	V-CPU Debugger Step Mask
0x00000010	RW	0x0	*VPU_PDBG_CTRL*	V-CPU Debugger Control
0x00000014	RW	0x0	*VPU_PDBG_IDX_REG*	V-CPU Debugger Index
0x00000018	RW	0x0	*VPU_PDBG_WDATA_REG*	V-CPU Debugger Write Data
0x00000020	RW	0x0	*VPU_FIO_CTRL_ADDR*	FastIO Control/Address
0x00000024	RW	0x0	*VPU_FIO_DATA*	FastIO Data
0x00000034	WO	0x0	*VPU_VINT_REASON_CLR*	Interrupt Reason Clear
0x00000038	WO	0x0	*VPU_HOST_INT_REQ*	Host Interrupt Request
0x0000003C	WO	0x0	*VPU_VINT_CLEAR*	VPU Interrupt Clear
0x00000048	RW	0x0	*VPU_VINT_ENABLE*	VPU Interrupt Enable
0x0000004C	RW	0x0	*VPU_VINT_REASON*	VPU Interrupt Reason
0x00000050	RW	0x0	*VPU_RESET_REQ*	VPU Reset Request
0x00000058	RW	0x0	*VCPU_RESTART*	VCPU Restart Request
0x0000005C	RW	0x0	*VPU_CLK_MASK*	VPU Clock Control
0x00000060	RW	0x0	*VPU_REMAP_CTRL*	Remap Control
0x00000064	RW	0x0	*VPU_REMAP_VADDR*	Remap Virutal (internal) Address
0x00000068	RW	0x0	*VPU_REMAP_PADDR*	Remap Physical (external) Address
0x0000006C	RW	0x0	*VPU_REMAP_CORE_START*	VPU Start Request
0x00000070	RW	0x0	*VPU_BUSY_STATUS*	VPU Busy Status
0x000000F8	RW	0x0	*VCPU_DDR_CH_SELECT*	DDR channel selector
OUTPUT RETURN				
0x00000004	RW	0x0	*VCPU_CUR_PC*	Current PC
0x0000001C	RW	0x0	*VPU_PDBG_RDATA_REG*	V-CPU Debugger Read Data
0x00000030	RW	0x0	*VPU_VINT_REASON_USR*	Interrupt Reason User
0x00000040	RO	0x0	*VPU_HINT_CLEAR*	Host Interrupt Clear
0x00000044	RO	0x0	*VPU_VPU_INT_STS*	VPU Interrupt Status
0x00000054	RW	0x0	*VPU_RESET_STATUS*	VPU Reset Status
0x00000074	RW	0x0	*VPU_HALT_STATUS*	VPU Halt Status
0x00000078	RW	0x0	*VPU_VCORE_PARSE_STATUS*	N/A
0x0000007C	RW	0x0	*VPU_VCORE_DEC_STATUS*	N/A

1.2.2.2. Summary of Command I/O registers

Among host interface registers, Command I/O Registers are used in a pre-defined way for each command controlling VPU. Sample usage of these Command I/O registers can be summarized as the following sub-sections.

Confidential Starfive Inc.

1.2.2.2.1. INIT_VPU Parameter Registers

Table 1.3. INIT_VPU Parameter Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x0000010C	RW	0x0	*INIT_OPTION*	Initialization options
0x00000118	RW	0x0	*ADDR_CODE_BASE*	Code buffer base address
0x0000011C	RW	0x0	*CODE_SIZE*	Code buffer size
0x00000120	RW	0x0	*CODE_PARAM*	Code buffer parameters
0x00000124	RW	0x0	*HW_OPTION*	VPU hardware options
0x00000134	RW	0x0	*TIMEOUT_CNT*	Time out count
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons

1.2.2.2.2. SET_PARAM Parameter Registers

1.2.2.2.2.1. COMMON

Table 1.4. Register summary

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x00000104	RW	0x0	*CORE_INDEX*	V-CORE index
0x00000108	RW	0x0	*INST_INDEX_COD_STD*	Codec standard/ Instance index
0x0000010C	RW	0x0	*CMD_ENC_SET_PARAM_OPTION*	SET_PARAM command options
0x00000120	RW	0x0	*BS_START_ADDR*	Bitstream Buffer Start Address
0x00000124	RW	0x0	*BS_SIZE*	Bitstream Buffer Size
0x00000128	RW	0x0	*BS_PARAM*	Bitstream buffer parameters
0x0000012C	RW	0x0	*BS_OPTIONS*	Bistream buffer options
0x00000130	RW	0x0	*BS_RD_PTR*	Bistream buffer read pointer
0x00000134	RW	0x0	*BS_WR_PTR*	Bistream buffer write pointer
0x00000138	RW	0x0	*ADDR_WORK_BASE*	Work buffer base address
0x0000013C	RW	0x0	*WORK_SIZE*	Work buffer size
0x00000140	RW	0x0	*WORK_PARAM*	Work buffer parameters
0x00000144	RW	0x0	*ADDR_TEMP_BASE*	Temporal buffer base address
0x00000148	RW	0x0	*TEMP_SIZE*	Temporal buffer size
0x0000014C	RW	0x0	*TEMP_PARAM*	Temporal buffer parameter
0x00000150	RW	0x0	*ADDR_SEC_AXI*	Secondary AXI base address
0x00000154	RW	0x0	*SEC_AXI_SIZE*	Seconary AXI memory size
0x00000158	RW	0x0	*USE_SEC_AXI*	Secondary AXI usage
0x0000015C	RW	0x0	*CMD_ENC_SET_PARAM_ENABLE*	Encoder parameter change setting
0x00000160	RW	0x0	*CMD_ENC_SEQ_SRC_SIZE*	A size of source picture
0x0000016C	RW	0x0	*CMD_ENC_SEQ_PARAM*	HEVC encoder sequence parameters
0x00000170	RW	0x0	*CMD_ENC_SEQ_GOP_PARAM*	GOP parameters
0x00000174	RW	0x0	*CMD_ENC_SEQ_INTRA_PARAM*	Intra picture coding parameters

Confidential Start Year

Offset	Type	Reset Value	Name	Description
0x00000178	RW	0x0	*CMD_ENC_SEQ_CONF_WIN_TOP_BOT*	Top and bottom size for conformance window
0x0000017C	RW	0x0	*CMD_ENC_SEQ_CONF_WIN_LEFT_RIGHT*	Left and right size for conformance window
0x00000180	RW	0x0	*CMD_ENC_SEQ_FRAME_RATE*	Frame rate
0x00000184	RW	0x0	*CMD_ENC_SEQ_INDEPENDENT_SLICE*	Slice parameters for an independent slice segment
0x00000188	RW	0x0	*CMD_ENC_SEQ_DEPENDENT_SLICE*	Slice parameters for a dependent slice segment
0x0000018C	RW	0x0	*CMD_ENC_SEQ_INTRA_REFRESH*	Intra refresh mode
0x00000190	RW	0x0	*CMD_ENC_PARAM*	HEVC encoder parameters related to coding tools
0x00000194	RW	0x0	*CMD_ENC_RC_MIN_MAX_QP*	Min/Max QP for rate control
0x00000198	RW	0x0	*CMD_ENC_RC_PARAM*	Rate control parameters
0x0000019C	RW	0x0	*CMD_ENC_RC_INTRA_MIN_MAX_QP*	Min/Max Intra QP for rate control
0x000001A0	RW	0x0	*CMD_ENC_RC_BIT_RATIO_LAYER_0_3*	RC_BIT_RATIO_LAYER_0_3
0x000001A4	RW	0x0	*CMD_ENC_RC_BIT_RATIO_LAYER_4_7*	RC_BIT_RATIO_LAYER_4_7
0x000001A8	RW	0x0	*CMD_ENC_NR_PARAM*	Noise reduction parameter
0x000001AC	RW	0x0	*CMD_ENC_NR_WEIGHT*	Noise reduction weight parameter
0x000001B0	RW	0x0	*CMD_ENC_NUM_UNITS_IN_TICK*	NUM_UNITS_IN_TICK
0x000001B4	RW	0x0	*CMD_ENC_TIME_SCALE*	TIME_SCALE
0x000001B8	RW	0x0	*CMD_ENC_NUM_TICKS_POC_DIFF_ONE*	NUM_TICKS_POC_DIFF_ONE
0x000001BC	RW	0x0	*CMD_ENC_RC_TRANS_RATE*	Peak transmission bitrate
0x000001C0	RW	0x0	*CMD_ENC_RC_TARGET_RATE*	Target bitrate for rate control
0x000001C4	RW	0x0	*CMD_ENC_RESERVED*	Reserved
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons
0x000001CC	RW	0x0	*RET_ENC_MIN_FB_NUM*	Minimum number of frame buffer required for encoding
0x000001D0	RW	0x0	*RET_ENC_NAL_INFO_TO_BE_ENCODED*	NAL unit type for next encode command
0x000001D4	RW	0x0	*RET_FRAME_CYCLE*	Frame processing cycle
0x000001D8	RW	0x0	*RET_MIN_SRC_BUF_NUM*	Minimum number of source frame buffer required for encoding

1.2.2.2.2.2. GOP

Table 1.5. Register summary

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x00000104	RW	0x0	*CORE_INDEX*	V-CORE index
0x00000108	RW	0x0	*INST_INDEX_COD_STD*	Codec standard/ Instance index
0x0000010C	RW	0x0	*CMD_ENC_SET_PARAM_OPTION*	SET_PARAM command options
0x00000120	RW	0x0	*BS_START_ADDR*	Bitstream Buffer Start Address
0x00000124	RW	0x0	*BS_SIZE*	Bitstream Buffer Size
0x00000128	RW	0x0	*BS_PARAM*	Bitstream buffer parameters
0x0000012C	RW	0x0	*BS_OPTIONS*	Bistream buffer options
0x00000130	RW	0x0	*BS_RD_PTR*	Bistream buffer read pointer

Confidential Start Inc.

Offset	Type	Reset Value	Name	Description
0x00000134	RW	0x0	*BS_WR_PTR*	Bistream buffer write pointer
0x00000138	RW	0x0	*ADDR_WORK_BASE*	Work buffer base address
0x0000013C	RW	0x0	*WORK_SIZE*	Work buffer size
0x00000140	RW	0x0	*WORK_PARAM*	Work buffer parameters
0x00000144	RW	0x0	*ADDR_TEMP_BASE*	Temporal buffer base address
0x00000148	RW	0x0	*TEMP_SIZE*	Temporal buffer size
0x0000014C	RW	0x0	*TEMP_PARAM*	Temporal buffer parameter
0x00000150	RW	0x0	*ADDR_SEC_AXI*	Secondary AXI base address
0x00000154	RW	0x0	*SEC_AXI_SIZE*	Seconary AXI memory size
0x00000158	RW	0x0	*USE_SEC_AXI*	Secondary AXI usage
0x0000015C	RW	0x0	*CMD_ENC_SET_CUSTOM_GOP_ENABLE*	Custom GOP setting
0x00000160	RW	0x0	*CMD_ENC_CUSTOM_GOP_PARAM*	Size of source pictures of custom GOP
0x00000164	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_PARAM_0*	Parameters for the 0th picture of custom GOP
0x00000168	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_PARAM_1*	Parameters for the 1st picture of custom GOP
0x0000016C	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_PARAM_2*	Parameters for the 2nd picture of custom GOP
0x00000170	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_PARAM_3*	Parameters for the 3rd picture of custom GOP
0x00000174	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_PARAM_4*	Parameters for the 4th picture of custom GOP
0x00000178	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_PARAM_5*	GOP parameter set 5 of the given picture
0x0000017C	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_PARAM_6*	GOP parameter set 6 of the given picture
0x00000180	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_PARAM_7*	GOP parameter set 7 of the given picture
0x00000184	RW	0x0	*CMD_ENC_CUSTOM_GOP_RESERVED*	Reserved
0x00000188	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_0*	Lamda weight of picture 0
0x0000018C	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_1*	Lamda weight of picture 1
0x00000190	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_2*	Lamda weight of picture 2
0x00000194	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_3*	Lamda weight of picture 3
0x00000198	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_4*	Lamda weight of picture 4
0x0000019C	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_5*	Lamda weight of picture 5
0x000001A0	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_6*	Lamda weight of picture 6
0x000001A4	RW	0x0	*CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_7*	Lamda weight of picture 7
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons
0x000001CC	RW	0x0	*RET_ENC_MIN_FB_NUM*	Minimum number of frame buffer required for encoding
0x000001D4	RW	0x0	*RET_FRAME_CYCLE*	Frame processing cycle
0x000001D8	RW	0x0	*RET_MIN_SRC_BUF_NUM*	Minimum number of source frame buffer required for encoding

1.2.2.2.2.3. VUI

Table 1.6. Register summary

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command

Confidential Start Fabless

Offset	Type	Reset Value	Name	Description
0x00000104	RW	0x0	*CORE_INDEX*	V-CORE index
0x00000108	RW	0x0	*INST_INDEX_COD_STD*	Codec standard/ Instance index
0x0000010C	RW	0x0	*CMD_ENC_SET_PARAM_OPTION*	SET_PARAM command options
0x00000120	RW	0x0	*BS_START_ADDR*	Bitstream Buffer Start Address
0x00000124	RW	0x0	*BS_SIZE*	Bitstream Buffer Size
0x00000128	RW	0x0	*BS_PARAM*	Bitstream buffer parameters
0x0000012C	RW	0x0	*BS_OPTIONS*	Bistream buffer options
0x00000130	RW	0x0	*BS_RD_PTR*	Bistream buffer read pointer
0x00000134	RW	0x0	*BS_WR_PTR*	Bistream buffer write pointer
0x00000138	RW	0x0	*ADDR_WORK_BASE*	Work buffer base address
0x0000013C	RW	0x0	*WORK_SIZE*	Work buffer size
0x00000140	RW	0x0	*WORK_PARAM*	Work buffer parameters
0x00000144	RW	0x0	*ADDR_TEMP_BASE*	Temporal buffer base address
0x00000148	RW	0x0	*TEMP_SIZE*	Temporal buffer size
0x0000014C	RW	0x0	*TEMP_PARAM*	Temporal buffer parameter
0x00000150	RW	0x0	*ADDR_SEC_AXI*	Secondary AXI base address
0x00000154	RW	0x0	*SEC_AXI_SIZE*	Seconary AXI memory size
0x00000158	RW	0x0	*USE_SEC_AXI*	Secondary AXI usage
0x0000015C	RW	0x0	*CMD_ENC_VUI_PARAM_FLAGS*	VUI parameter flag
0x00000160	RW	0x0	*CMD_ENC_VUI_ASPECT_RATIO_IDC*	VUI apsect ratio idc
0x00000164	RW	0x0	*CMD_ENC_VUI_SAR_SIZE*	VUI sar size
0x00000168	RW	0x0	*CMD_ENC_VUI_OVERSCAN_APPROPRIATE*	VUI overscan appropriate flag
0x0000016C	RW	0x0	*CMD_ENC_VUI_VIDEO_SIGNAL*	VUI video signal
0x00000170	RW	0x0	*CMD_ENC_VUI_CHROMA_SAMPLE_LOC*	VUI chroma sample location type
0x00000174	RW	0x0	*CMD_ENC_VUI_DISP_WIN_LEFT_RIGHT*	VUI display window left right offset
0x00000178	RW	0x0	*CMD_ENC_VUI_DISP_WIN_TOP_BOT*	VUI display window top bottom offset
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons

Confidential Starfive Inc.

1.2.2.2.3. ENC_PIC Parameter Registers

Table 1.7. Register summary

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x00000104	RW	0x0	*CORE_INDEX*	V-CORE index
0x00000108	RW	0x0	*INST_INDEX_COD_STD*	Codec standard/ Instance index
0x00000120	RW	0x0	*BS_START_ADDR*	Bitstream Buffer Start Address
0x00000124	RW	0x0	*BS_SIZE*	Bitstream Buffer Size
0x00000128	RW	0x0	*BS_PARAM*	Bitstream buffer parameters
0x0000012C	RW	0x0	*BS_OPTIONS*	Bistream buffer options
0x00000130	RW	0x0	*BS_RD_PTR*	Bistream buffer read pointer
0x00000134	RW	0x0	*BS_WR_PTR*	Bistream buffer write pointer
0x00000138	RW	0x0	*ADDR_WORK_BASE*	Work buffer base address
0x0000013C	RW	0x0	*WORK_SIZE*	Work buffer size

Offset	Type	Reset Value	Name	Description
0x00000140	RW	0x0	*WORK_PARAM*	Work buffer parameters
0x00000144	RW	0x0	*ADDR_TEMP_BASE*	Temporal buffer base address
0x00000148	RW	0x0	*TEMP_SIZE*	Temporal buffer size
0x0000014C	RW	0x0	*TEMP_PARAM*	Temporal buffer parameter
0x00000150	RW	0x0	*ADDR_SEC_AXI*	Secondary AXI base address
0x00000154	RW	0x0	*SEC_AXI_SIZE*	Seconary AXI memory size
0x00000158	RW	0x0	*USE_SEC_AXI*	Secondary AXI usage
0x0000015C	RW	0x0	*CMD_ENC_ADDR_REPORT_BASE*	Base address of report buffer
0x00000160	RW	0x0	*CMD_ENC_REPORT_SIZE*	Byte size of report buffer
0x00000164	RW	0x0	*CMD_ENC_REPORT_PARAM*	Report parameters
0x00000168	R/W	0x0	*CMD_ENC_CODE_OPTION*	NAL unit coding options
0x0000016C	RW	0x0	*CMD_ENC_PIC_PARAM*	HEVC encoder sequence parameters
0x00000170	RW	0x0	*CMD_ENC_SRC_PIC_IDX*	Buffer index of source picture
0x00000174	RW	0x0	*CMD_ENC_SRC_ADDR_Y*	Y component source address
0x00000178	RW	0x0	*CMD_ENC_SRC_ADDR_U*	Cb component source address
0x0000017C	RW	0x0	*CMD_ENC_SRC_ADDR_V*	Cr component source address
0x00000180	RW	0x0	*CMD_ENC_SRC_STRIDE*	Stride of source picture
0x00000184	RW	0x0	*CMD_ENC_SRC_FORMAT*	Format of source picture
0x00000188	RW	0x0	*CMD_ENC_PREFIX_SEI_NAL_ADDR*	Address of prefix SEI nal data
0x0000018C	RW	0x0	*CMD_ENC_PREFIX_SEI_INFO*	Information of prefix SEI nal data
0x00000190	RW	0x0	*CMD_ENC_SUFFIX_SEI_NAL_ADDR*	Address of suffix SEI nal data
0x00000190	RW	0x0	*CMD_ENC_SRC_TIMESTAMP_LOW*	LSB of timestamp
0x00000194	RW	0x0	*CMD_ENC_SUFFIX_SEI_INFO*	Information of suffix SEI nal data
0x00000194	RW	0x0	*CMD_ENC_SRC_TIMESTAMP_HIGH*	MSB of timestamp
0x00000198	RW	0x0	*CMD_ENC_LONGTERM_PIC*	longterm picture setting
0x000001A0	RW	0x0	*CMD_ENC_ROI_PARAM*	ROI parameters
0x000001A4	RW	0x0	*CMD_ENC_ROI_MAP_ADDR*	Start buffer address of ROI map
0x000001A8	RW	0x0	*CMD_ENC_CTU_MODE_MAP_ADDR*	Start buffer address of CTU mode map
0x000001AC	RW	0x0	*CMD_ENC_CTU_QP_MAP_ADDR*	Start buffer address of CTU QP map
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons
0x000001A8	RW	0x0	*RET_ENC_PIC_IDX*	Frame buffer index of encoded picture
0x000001AC	RW	0x0	*RET_ENC_PIC_SLICE_NUM*	Number of slice segments
0x000001B0	RW	0x0	*RET_ENC_PIC_SKIP*	A picture skip flag
0x000001B4	RW	0x0	*RET_ENC_PIC_NUM_INTRA*	Number of intra block
0x000001B8	RW	0x0	*RET_ENC_PIC_NUM_MERGE*	Number of merge block
0x000001BC	RW	0x0	*RET_ENC_PIC_RESERVED*	Reserved
0x000001C0	RW	0x0	*RET_ENC_PIC_NUM_SKIP*	Number of skip block
0x000001C4	RW	0x0	*RET_ENC_PIC_AVG_CU_QP*	CU QP on average
0x000001C8	RW	0x0	*RET_ENC_PIC_BYTE*	Byte size of encoded picture
0x000001CC	RW	0x0	*RET_ENC_GOP_PIC_IDX*	A picture index in GOP
0x000001D0	RW	0x0	*RET_ENC_PIC_POC*	A POC value of encoded picture
0x000001D4	RW	0x0	*RET_FRAME_CYCLE*	Frame processing cycle
0x000001D8	RW	0x0	*RET_ENC_USED_SRC_IDX*	Buffer index of source picture that is used for encoding

Confidential
Starfive Inc.

Offset	Type	Reset Value	Name	Description
0x000001DC	RW	0x0	*RET_ENC_PIC_NUM*	Encoded picture number
0x000001E0	RW	0x0	*RET_ENC_PIC_TYPE*	Encoded picture type

1.2.2.2.4. SET_FRAMEBUF Parameter Registers

Table 1.8. SET_FRAMEBUF Parameter Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x00000104	RW	0x0	*CORE_INDEX*	V-CORE index
0x00000108	RW	0x0	*INST_INDEX_COD_STD*	Codec standard/ Instance index
0x0000010C	RW	0x0	*SFB_OPTION*	Set Dpb Frame Option
0x00000120	RW	0x0	*COMMON_PIC_INFO*	Dpb Information
0x00000124	RW	0x0	*PIC_SIZE*	Decoded Picture Size
0x00000128	RW	0x0	*SET_FB_NUM*	Set frame Number
0x00000138	RW	0x0	*ADDR_WORK_BASE*	Work Buffer Base Address
0x0000013C	RW	0x0	*WORK_SIZE*	Work Buffer Size
0x00000140	RW	0x0	*WORK_PARAM*	Work Buffer Parameter
0x00000154	RW	0x0	*FBC_STRIDE*	Frame buffer setting for compressed frame
0x00000158	RW	0x0	*ADDR_SUB_SAMPLED_FB_BASE*	Base address of frame buffer for sub-sampled frame
0x0000015C	RW	0x0	*SUB_SAMPLED_ONE_FB_SIZE*	Size of frame buffer for sub-sampled frame
0x00000160	RW	0x0	*ADDR_LUMA_BASE0*	Luma base of index0
0x00000164	RW	0x0	*ADDR_CB_BASE0*	Cb base of index0
0x00000168	RW	0x0	*ADDR_CR_BASE0*	Cr base of index0
0x00000168	RW	0x0	*ADDR_FBC_Y_OFFSET0*	FBC luma offset base of index0
0x0000016C	RW	0x0	*ADDR_FBC_C_OFFSET0*	FBC chroma offset base of index0
0x00000170	RW	0x0	*ADDR_LUMA_BASE1*	Luma base of index1
0x00000174	RW	0x0	*ADDR_CB_BASE1*	Cb base of index1
0x00000178	RW	0x0	*ADDR_CR_BASE1*	Cr base of index1
0x00000178	RW	0x0	*ADDR_FBC_Y_OFFSET1*	FBC luma offset base of index1
0x0000017C	RW	0x0	*ADDR_FBC_C_OFFSET1*	FBC chroma offset base of index1
0x00000180	RW	0x0	*ADDR_LUMA_BASE2*	Luma base of index2
0x00000184	RW	0x0	*ADDR_CB_BASE2*	Cb base of index2
0x00000188	RW	0x0	*ADDR_CR_BASE2*	Cr base of index2
0x0000018C	RW	0x0	*ADDR_FBC_C_OFFSET2*	FBC chroma offset base of index2
0x00000190	RW	0x0	*ADDR_LUMA_BASE3*	Luma base of index3
0x00000194	RW	0x0	*ADDR_CB_BASE3*	Cb base of index3
0x00000198	RW	0x0	*ADDR_CR_BASE3*	Cr base of index3
0x00000198	RW	0x0	*ADDR_FBC_Y_OFFSET3*	FBC luma offset base of index3
0x0000019C	RW	0x0	*ADDR_FBC_C_OFFSET3*	FBC chroma offset base of index3
0x000001A0	RW	0x0	*ADDR_LUMA_BASE4*	Luma base of index4
0x000001A4	RW	0x0	*ADDR_CB_BASE4*	Cb base of index4
0x000001A8	RW	0x0	*ADDR_CR_BASE4*	Cr base of index4
0x000001A8	RW	0x0	*ADDR_FBC_Y_OFFSET4*	FBC luma offset base of index4
0x000001AC	RW	0x0	*ADDR_FBC_C_OFFSET4*	FBC chroma offset base of index4
0x000001B0	RW	0x0	*ADDR_LUMA_BASE5*	Luma base of index5

Confidential
StarFive Inc.

Offset	Type	Reset Value	Name	Description
0x000001B4	RW	0x0	*ADDR_CB_BASE5*	Cb base of index5
0x000001B8	RW	0x0	*ADDR_CR_BASE5*	Cr base of index5
0x000001B8	RW	0x0	*ADDR_FBC_Y_OFFSET5*	FBC luma offset base of index5
0x000001BC	RW	0x0	*ADDR_FBC_C_OFFSET5*	FBC chroma offset base of index5
0x000001C0	RW	0x0	*ADDR_LUMA_BASE6*	Luma base of index6
0x000001C4	RW	0x0	*ADDR_CB_BASE6*	Cb base of index6
0x000001C8	RW	0x0	*ADDR_CR_BASE6*	Cr base of index6
0x000001C8	RW	0x0	*ADDR_FBC_Y_OFFSET6*	FBC luma offset base of index6
0x000001CC	RW	0x0	*ADDR_FBC_C_OFFSET6*	FBC chroma offset base of index6
0x000001D0	RW	0x0	*ADDR_LUMA_BASE7*	Luma base of index7
0x000001D4	RW	0x0	*ADDR_CB_BASE7*	Cb base of index7
0x000001D8	RW	0x0	*ADDR_CR_BASE7*	Cr base of index7
0x000001D8	RW	0x0	*ADDR_FBC_Y_OFFSET7*	FBC luma offset base of index7
0x000001DC	RW	0x0	*ADDR_FBC_C_OFFSET7*	FBC chroma offset base of index7
0x000001E0	RW	0x0	*ADDR_MV_COL0*	Colocated mv buffer base of index 0
0x000001E4	RW	0x0	*ADDR_MV_COL1*	Colocated mv buffer base of index 1
0x000001E8	RW	0x0	*ADDR_MV_COL2*	Colocated mv buffer base of index 2
0x000001EC	RW	0x0	*ADDR_MV_COL3*	Colocated mv buffer base of index 3
0x000001F0	RW	0x0	*ADDR_MV_COL4*	Colocated mv buffer base of index 4
0x000001F4	RW	0x0	*ADDR_MV_COL5*	Colocated mv buffer base of index 5
0x000001F8	RW	0x0	*ADDR_MV_COL6*	Colocated mv buffer base of index 6
0x000001FC	RW	0x0	*ADDR_MV_COL7*	Colocated mv buffer base of index 7
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	0	0x0	*RET_FAIL_REASON*	Fail reasons

1.2.2.2.5. GET_FW_VERSION Parameter Registers

Table 1.9. GET_FW_VERSION Parameter Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run Command
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run Command Status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail Reason
0x00000118	RW	0x0	*RET_FW_VERSION*	Firmware Version
0x0000011C	RW	0x0	*RET_PRODUCT_NAME*	HW product name
0x00000120	RW	0x0	*RET_PRODUCT_VERSION*	HW product version
0x00000124	RW	0x0	*RET_STD_DEF0*	Standard Definition #0
0x00000128	RW	0x0	*RET_STD_DEF1*	Standard Definition #1
0x0000012C	RW	0x0	*RET_CODEC_STD*	Standard Definition
0x00000130	RW	0x0	*RET_CONF_DATE*	The date of H/W configuration
0x00000134	RW	0x0	*RET_CONF_REVISION*	The revision of H/W configuration
0x00000138	RW	0x0	*RET_CONF_TYPE*	The define value of H/W configur

Confidential StarFive Inc.

1.2.2.2.6. SLEEP_VPU Parameter Registers

Table 1.10. SLEEP_VPU Parameter Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons
0x00000124	RW	0x0	*RET_CUR_SP*	Return Current Stack Pointer

1.2.2.2.7. WAKEUP_VPU Parameter Registers

Table 1.11. WAKEUP_VPU Parameter Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x00000118	RW	0x0	*ADDR_CODE_BASE*	Code buffer base address
0x0000011C	RW	0x0	*CODE_SIZE*	Code buffer size
0x00000120	RW	0x0	*CODE_PARAM*	Code buffer parameters
0x00000124	RW	0x0	*CUR_SP*	Current Stack Pointer
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons

1.2.2.2.8. CREATE_INST Parameter Registers

Table 1.12. CREATE_INST Parameter Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x00000104	RW	0x0	*CORE_INDEX*	V-CORE index
0x00000108	RW	0x0	*INST_INDEX_COD_STD*	Codec standard/ Instance index
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons

1.2.2.2.9. CHANGE_INST Parameter Registers

Table 1.13. CHANGE_INST Parameter Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x00000104	RW	0x0	*CORE_INDEX*	V-CORE index
0x00000108	RW	0x0	*INST_INDEX_COD_STD*	Codec standard/ Instance index
0x00000138	RW	0x0	*ADDR_WORK_BASE*	Work buffer base address
0x0000013C	RW	0x0	*WORK_SIZE*	Work buffer size

Confidential
StarFive Inc.

Offset	Type	Reset Value	Name	Description
0x00000140	RW	0x0	*WORK_PARAM*	Work buffer parameters
OUTPUT RETURN				
0x00000110	RW	0x0	*RET_SUCCESS*	Run command status
0x00000114	RW	0x0	*RET_FAIL_REASON*	Fail reasons

1.2.2.2.10. UPDATE_BS Parameter Registers

Table 1.14. UPDATE_BS Parameter Registers

Offset	Type	Reset Value	Name	Description
INPUT ARGUMENT				
0x00000100	RW	0x0	*COMMAND*	Run command
0x0000012C	RW	0x0	*BS_OPTION*	Bistream Buffer Option
0x00000134	RW	0x0	*BS_WR_PTR*	Bistream buffer write pointer
OUTPUT RETURN				

Confidential StarFive Inc.

1.2.3. Register Descriptions

Detailed definitions of host interface registers are presented in the following sub-sections. It covers general description, bit assignment, and meaning of bit field of each host interface register.

1.2.3.1. Control Register Descriptions

1.2.3.1.1. VPU_PO_CONF (0x00000000)

Power On Configurations

Table 1.15. VPU_PO_CONF Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												USE_PO_CONF	RSVD		DEBUGMODE
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	RO	RO	WO

Table 1.16. VPU_PO_CONF Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3]	USE_PO_CONF	WO	USE PO_CONF Host processor should set 0 for this field which is required when VPU initialization.	0x0
[2:1]	RSVD	RO	Reserved	0x0
[0]	DEBUGMODE	WO	Power on with debug mode Host processor should set 0 for this field.	0x0

1.2.3.1.2. VCPU_CUR_PC (0x00000004)

Current program counter value of V-CPU

Table 1.17. VCPU_CUR_PC Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUR_PC																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.18. VCPU_CUR_PC Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CUR_PC	RO	PC value represents the address of instruction which is executed in V-CPU. (for debugging purpose only)	0x0

1.2.3.1.3. VPU_PDBG_STEP_MASK (0x0000000C)

Debugger control

Confidential StarFive Inc.

(for debugger only)

Table 1.19. VPU_PDBG_STEP_MASK Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															STEP_MASK_ENABLE
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW

Table 1.20. VPU_PDBG_STEP_MASK Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	STEP_MASK_ENABLE	RW	It is V-CPU interrupt mask for STEP mode	0x0

1.2.3.1.4. VPU_PDBG_CTRL (0x00000010)

Debugger control
(for debugger only)

Table 1.21. VPU_PDBG_CTRL Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												IMMBRK	STABLEBRK	RESUME	STEP
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO

Table 1.22. VPU_PDBG_CTRL Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3]	IMMBRK	WO	Immediate break It forces to stop V-CPU and enter in a debug mode to analysis hang-up situation. V-CPU cannot resume from the break point.	0x0
[2]	STABLEBRK	WO	Stable break It is an external break request when V-CPU is in stable (breakable) status.	0x0
[1]	RESUME	WO	Resume It resumes from the breakpoint.	0x0
[0]	STEP	WO	Step It runs V-CPU in step instruction mode.	0x0

1.2.3.1.5. VPU_PDBG_IDX_REG (0x00000014)

V-CPU debugger index register
(For debugger only)

Confidential StarFive Inc.

Table 1.23. VPU_PDBG_IDX_REG Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						RDDBG	WRDBG	DBGIDX							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

Table 1.24. VPU_PDBG_IDX_REG Field Description

Bit	Name	Type	Function	Reset Value
[31:10]	RSVD	RO	Reserved	0x0
[9]	RDDBG	WO	Read Operation Request RDDBG and WRDBG cannot be 1 at the same time.	0x0
[8]	WRDBG	WO	Write Operation Request RDDBG and WRDBG cannot be 1 at the same time.	0x0
[7:0]	DBGIDX	WO	Debug Index Debugger Register Index	0x0

1.2.3.1.6. VPU_PDBG_WDATA_REG (0x00000018)

V-CPU debugger write data register
(For debugger only)

Table 1.25. VPU_PDBG_WDATA_REG Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VPU_PDBG_WDATA_REG																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

Table 1.26. VPU_PDBG_WDATA_REG Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	VPU_PDBG_WDATA_REG	WO	To write data to the debugger, 1. Write some data to this register. 2. Write VCPU_PDBG_IDX_REG with WRDBG as 1 and DBGIDX as this register address. 3. After writing is completed, VPU_PDBG_WDATA_REG will be cleared.	0x0

1.2.3.1.7. VPU_PDBG_RDATA_REG (0x0000001C)

V-CPU debugger read data register
(For debugger only)

Table 1.27. VPU_PDBG_RDATA_REG Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

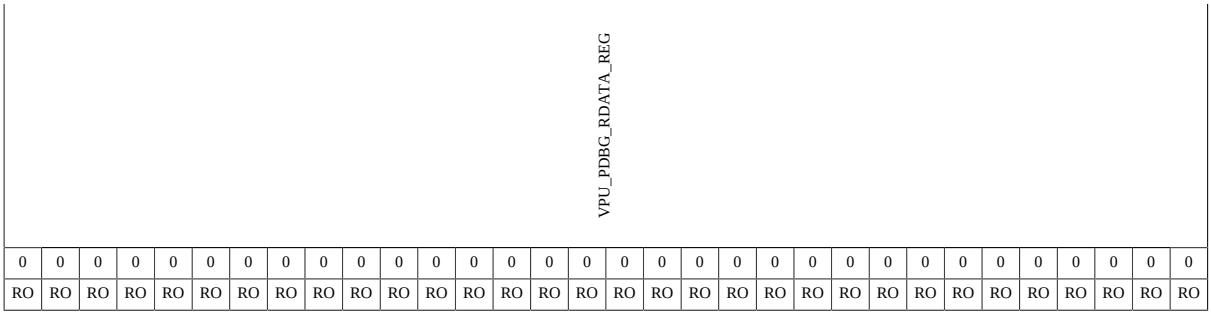


Table 1.28. VPU_PDBG_RDATA_REG Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	VPU_PDBG_RDATA_REG	RO	To read data to the debugger, 1. Write VCPU_PDBG_IDX_REG with RDDBG as 1 and DBGIDX as the register address to be read. 2. Read from the specified register to this register.	0x0

1.2.3.1.8. VPU_FIO_CTRL_ADDR (0x00000020)

FIO CRTL, READY, and Address for accessing FIO (FastIO) which is an internal peripheral bus in VPU.
By accessing FIO, user can check the internal status of some accelerators in VPU for debugging purpose in some cases.
FIO transaction is carried out when host writes this register.
CAUTON> Because accessing FIO can make unwanted behavior in VPU, please access it using API only.

Table 1.29. VPU_FIO_CTRL_ADDR Bit Assignment

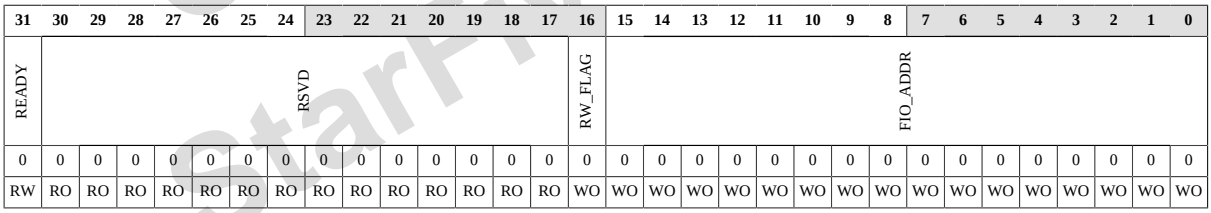


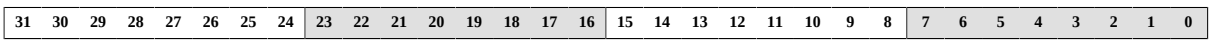
Table 1.30. VPU_FIO_CTRL_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31]	READY	RW	Ready for the transaction When writing, it should be 0.	0x0
[30:17]	RSVD	RO	Reserved	0x0
[16]	RW_FLAG	WO	Read/Write transaction control 0: read 1: write	0x0
[15:0]	FIO_ADDR	WO	FIO Address	0x0

1.2.3.1.9. VPU_FIO_DATA (0x00000024)

FIO data

Table 1.31. VPU_FIO_DATA Bit Assignment



Confidential StarFive Inc.

FIO_DATA																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.32. VPU_FIO_DATA Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FIO_DATA	RW	When writing, FIO data should be written to this register firstly. When reading, 1. Write data to this register. 2. Check whether READY flag of VPU_FIO_CTRL_ADDR register is 1. 3. When READY flag is asserted, VPU reads data from this VPU_FIO_DATA.	0x0

1.2.3.1.10. VPU_VINT_REASON_USR (0x00000030)

Interrupt Reason Check & Clear using user level previlege in the host

- When an interrupt is asserted, the value of VPU_VINT_REASON is ORing to VPU_VINT_REASON_USER.
- Even if interrupt service routine(ISR) clears VPU_INT_REASON by using VPU_VINT_REASON_CLR, an application in user mode can identify the reason of the interrupt signalling from VPU by accessing this register.
- Applications in user mode should clear this register just after perform processing for the interrupt.

Table 1.33. VPU_VINT_REASON_USR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																BSEMPTY_INTR_USER	CMDE_INTR_USER	RSVD	CMDC_INTR_USER	CMDB_INTR_USER	CMDA_INTR_USER	CMD9_INTR_USER	CMD8_INTR_USER	RSVD		CMD5_INTR_USER	CMD4_INTR_USER	CMD3_INTR_USER	CMD2_INTR_USER	CMD1_INTR_USER	CMD0_INTR_USER
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RO	RO	RW	RW	RW	RW	RW	RW

Table 1.34. VPU_VINT_REASON_USR Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15]	BSEMPTY_INTR_USER	RW	Bitstream empty(feeding request) interrupt	0x0
[14]	CMDE_INTR_USER	RW	CREATE_INSTANCE command done interrupt	0x0
[13]	RSVD	RW	Reserved	0x0
[12]	CMDC_INTR_USER	RW	CHANGE_INST command done interrupt	0x0
[11]	CMDB_INTR_USER	RW	WAKEUP_VPU command done interrupt	0x0
[10]	CMDA_INTR_USER	RW	SLEEP_VPU command done interrupt	0x0
[9]	CMD9_INTR_USER	RW	QUERY_DECODER command done interrupt	0x0
[8]	CMD8_INTR_USER	RW	GET_FW_VERSION command done interrupt	0x0
[7:6]	RSVD	RO	Reserved	0x0
[5]	CMD5_INTR_USER	RW	FLUSH_DECODER command done interrupt	0x0
[4]	CMD4_INTR_USER	RW	SET_FRAMEBUF command done interrupt	0x0

Confidential
CtarFive Inc.

Bit	Name	Type	Function	Reset Value
[3]	CMD3_INTR_USER	RW	DEC_PIC command done interrupt	0x0
[2]	CMD2_INTR_USER	RW	FINI_SEQ command done interrupt	0x0
[1]	CMD1_INTR_USER	RW	DEC_PIC_HDR/ENC_SET_PARAM command done interrupt	0x0
[0]	CMD0_INTR_USER	RW	INIT_VPU command done interrupt	0x0

1.2.3.1.11. VPU_VINT_REASON_CLR (0x00000034)

Interrupt Reason Clear
This register clears the interrupt reason in ISR when host processor catchs an interrupt. It is to notify that host processor has received the interrupt. If host processor wants to get task done interrupts or so from VPU, they should set the fields of VPU_VINT_ENABLE register as they wish. And when they recieve any of the command done interrupt from VPU (VPU_VINT_REASON_CLR is not zero), host processor should check the interrupt and do the following sequence for safe interrupt clear.

1. Clear the interrupt by setting the relevant bit of this VPU_VINT_REASON_CLR.
2. Set VPU_VINT_CLEAR to 1, which drops the interrupt signal.
3. Then VPU_VPU_INT_STS is automatically cleared.

In above sequence, 1 is important because VPU_VINT_CLEAR without VPU_VINT_REASON_CLR might lead to interrupt reassurtion. It was because interrupts happened concurrently and other interrupt is still remaining even though one was cleared.

Table 1.35. VPU_VINT_REASON_CLR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																BSEMPTY_CLR	CMDE_CLR	RSVD	CMDC_CLR	CMDB_CLR	CMDA_CLR	CMD9_CLR	CMD8_CLR	RSVD		CMD5_CLR	CMD4_CLR	CMD3_CLR	CMD2_CLR	CMD1_CLR	CMD0_CLR
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RO	RO	RW	RW	RW	RW	RW	RW

Table 1.36. VPU_VINT_REASON_CLR Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15]	BSEMPTY_CLR	RW	Bitstream empty(feeding request) interrupt clear	0x0
[14]	CMDE_CLR	RW	CREATE_INSTANCE command done interrupt clear	0x0
[13]	RSVD	RW	Reserved	0x0
[12]	CMDC_CLR	RW	CHANGE_INST command done interrupt clear	0x0
[11]	CMDB_CLR	RW	WAKEUP_VPU command done interrupt clear	0x0
[10]	CMDA_CLR	RW	SLEEP_VPU command done interrupt clear	0x0
[9]	CMD9_CLR	RW	QUERY_DECODER command done interrupt clear	0x0
[8]	CMD8_CLR	RW	GET_FW_VERSION command done interrupt clear	0x0
[7:6]	RSVD	RO	Reserved	0x0
[5]	CMD5_CLR	RW	FLUSH_DECODER command done interrupt clear	0x0
[4]	CMD4_CLR	RW	SET_FRAMEBUF command done interrupt clear	0x0
[3]	CMD3_CLR	RW	DEC_PIC command done interrupt clear	0x0
[2]	CMD2_CLR	RW	FINI_SEQ command done interrupt clear	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[1]	CMD1_CLR	RW	DEC_PIC_HDR/ENC_SET_PARAM command done interrupt clear	0x0
[0]	CMD0_CLR	RW	INIT_VPU command done interrupt clear	0x0

1.2.3.1.12. VPU_HOST_INT_REQ (0x00000038)

Interrupt request sent from host processor to VPU for the command and so on.

Table 1.37. VPU_HOST_INT_REQ Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															HINTREQ
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW

Table 1.38. VPU_HOST_INT_REQ Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	HINTREQ	RW	If this is set to 1, an interrupt named HOST interrupt is sent to VPU.	0x0

1.2.3.1.13. VPU_VINT_CLEAR (0x0000003C)

VPU interrupt clear

Table 1.39. VPU_VINT_CLEAR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															VINTCLR
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW

Table 1.40. VPU_VINT_CLEAR Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	VINTCLR	RW	Host processor can clear the VPU interrupt which has been pending through this register. As mentioned in VPU_VINT_REASON_CLR, this register setting definetely come after VPU_INT_REASON setting. Also, setting this VPU_VINT_CLEAR to 1 forces VPU_VPU_INT_STS INT_STS to be cleared automatically.	0x0

1.2.3.1.14. VPU_HINT_CLEAR (0x00000040)

Host interrupt clear

Table 1.41. VPU_HINT_CLEAR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															HINTCLR

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW

Table 1.42. VPU_HINT_CLEAR Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	HINTCLR	RW	VPU can clear the host interrupt which has been pending through this register. When VPU operation for the host interrupt is done, VPU sets this reigster to clear the host interrupt.	0x0

1.2.3.1.15. VPU_VPU_INT_STS (0x00000044)

Interrupt Status

Table 1.43. VPU_VPU_INT_STS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															VPU_VPU_INT_STS
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW

Table 1.44. VPU_VPU_INT_STS Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	VPU_VPU_INT_STS	RW	VPU Interrupt status which comes from VPU to Host This register turns 1 if VPU_VINT_REASON is not zero (any bit of VPU_VINT_REASON is on), and it becomes to be clear by setting VPU_VINT_CLEAR register. This is a way of interrupt check by register polling.	0x0

1.2.3.1.16. VPU_VINT_ENABLE (0x00000048)

Interrupt Enable for each interrupt reason(source)
Interrupt in LSB position shall be handled with higher priority.

Table 1.45. VPU_VINT_ENABLE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																CMDF_EN	CMDE_EN	RSVD	CMDC_EN	CMDB_EN	CMDA_EN	CMD9_EN	CMD8_EN	RSVD		CMD5_EN	CMD4_EN	CMD3_EN	CMD2_EN	CMD1_EN	CMD0_EN
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RO	RO	RW	RW	RW	RW	RW	RW

Table 1.46. VPU_VINT_ENABLE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15]	CMDF_EN	RW	UPDATE_BS command done interrupt enable	0x0

Confidential
StarFive Inc.

Bit	Name	Type	Function	Reset Value
[14]	CMDE_EN	RW	CREATE_INSTANCE command done interrupt enable	0x0
[13]	RSVD	RW	Reserved	0x0
[12]	CMDC_EN	RW	CHANGE_INST command done interrupt enable	0x0
[11]	CMDB_EN	RW	WAKEUP_VPU command done interrupt enable	0x0
[10]	CMDA_EN	RW	SLEEP_VPU command done interrupt enable	0x0
[9]	CMD9_EN	RW	QUERY_DECODER command done interrupt enable	0x0
[8]	CMD8_EN	RW	GET_FW_VERSION command done interrupt enable	0x0
[7:6]	RSVD	RO	Reserved	0x0
[5]	CMD5_EN	RW	FLUSH_DECODER command done interrupt enable	0x0
[4]	CMD4_EN	RW	SET_FRAMEBUF command done interrupt enable	0x0
[3]	CMD3_EN	RW	DEC_PIC/ENC_PIC command done interrupt enable	0x0
[2]	CMD2_EN	RW	FINI_SEQ command done interrupt enable	0x0
[1]	CMD1_EN	RW	DEC_PIC_HDR/ENC_SETPARAM command done interrupt enable	0x0
[0]	CMD0_EN	RW	INIT_VPU command done interrupt enable	0x0

1.2.3.1.17. VPU_VINT_REASON (0x0000004C)

VPU interrupt reason
This register Interrupt reasons are almost the same with the command. Interrupt in LSB position shall be handled with higher priority.

Table 1.47. VPU_VINT_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																BSEMPTY_INTR	CMDE_INTR	RSVD	CMDC_INTR	CMDB_INTR	CMDA_INTR	CMD9_INTR	CMD8_INTR	RSVD		CMD5_INTR	CMD4_INTR	CMD3_INTR	CMD2_INTR	CMD1_INTR	CMD0_INTR
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RO	RO	RW	RW	RW	RW	RW	RW

Table 1.48. VPU_VINT_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15]	BSEMPTY_INTR	RW	Bitstream empty (bitstream feeding request)	0x0
[14]	CMDE_INTR	RW	CREATE_INSTANCE command done	0x0
[13]	RSVD	RW	Reserved	0x0
[12]	CMDC_INTR	RW	CHANGE_INST command done	0x0
[11]	CMDB_INTR	RW	WAKEUP_VPU command done	0x0
[10]	CMDA_INTR	RW	SLEEP_VPU command done	0x0
[9]	CMD9_INTR	RW	QUERY_DECODER command done	0x0
[8]	CMD8_INTR	RW	GET_FW_VERSION command done	0x0
[7:6]	RSVD	RO	Reserved	0x0
[5]	CMD5_INTR	RW	FLUSH_DECODER command done	0x0
[4]	CMD4_INTR	RW	SET_FRAMEBUF command done	0x0
[3]	CMD3_INTR	RW	DEC_PIC/ENC_PIC command done	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[2]	CMD2_INTR	RW	FINI_SEQ command done	0x0
[1]	CMD1_INTR	RW	DEC_PIC_HDR/ENC_SETPARAM command done	0x0
[0]	CMD0_INTR	RW	INIT_VPU command done	0x0

1.2.3.1.18. VPU_RESET_REQ (0x00000050)

VPU reset request for each block of clock domain
0: reset request clear or do noting
1: reset request For more details, please refer to *clock and reset signals* section in datasheet.

Table 1.49. VPU_RESET_REQ Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					MRST_REQ	VRST_REQ	VARST_REQ	ARST_REQ								BRST_REQ								CRST_REQ							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.50. VPU_RESET_REQ Field Description

Bit	Name	Type	Function	Reset Value
[31:27]	RSVD	RO	Reserved	0x0
[26]	MRST_REQ	RW	Reset request for MCLK (low-latency processor AXI bus clock for V-CPU) domain	0x0
[25]	VRST_REQ	RW	Reset rquest for VCLK (V-CPU clock) domain	0x0
[24]	VARST_REQ	RW	Reset rquest for ACLK (high bandwidth AXI bus clock for V-CPU) domain	0x0
[23:16]	ARST_REQ	RW	Reset request for ACLK (high bandwidth AXI bus clock for each V-CORE) domain [23] : V-CORE7 [22] : V-CORE6 [21] : V-CORE5 [20] : V-CORE4 [19] : V-CORE3 [18] : V-CORE2 [17] : V-CORE1 [16] : V-CORE0	0x0
[15:8]	BRST_REQ	RW	Reset request for BCLK (BPU clock for each V-CORE) domain [15] : V-CORE7 [14] : V-CORE6 [13] : V-CORE5 [12] : V-CORE4 [11] : V-CORE3 [10] : V-CORE2 [9] : V-CORE1 [8] : V-CORE0	0x0
[7:0]	CRST_REQ	RW	Reset request for CCLK (VCE clock for each V-CORE) domain [7] : V-CORE7 [6] : V-CORE6 [5] : V-CORE5 [4] : V-CORE4 [3] : V-CORE3 [2] : V-CORE2	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			[1] : V-CORE1 [0] : V-CORE0	

1.2.3.1.19. VPU_RESET_STATUS (0x00000054)

Reset status for each clock domain
0: reset is released
1: on reset handling phase

Table 1.51. VPU_RESET_STATUS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					MRST_STS	VRST_STS	VARST_STS	ARST_STS								BRST_STS								CRST_STS							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.52. VPU_RESET_STATUS Field Description

Bit	Name	Type	Function	Reset Value
[31:27]	RSVD	RO	Reserved	0x0
[26]	MRST_STS	RO	MCLK domain reset status	0x0
[25]	VRST_STS	RO	VCLK domain reset status	0x0
[24]	VARST_STS	RO	ACLK domain (for vCPU) reset status	0x0
[23:16]	ARST_STS	RO	ACLK domain (for each vCORE) reset status	0x0
[15:8]	BRST_STS	RO	BCLK domain (for each vCORE) reset status	0x0
[7:0]	CRST_STS	RO	CCLK domain (for each vCORE) reset status	0x0

1.2.3.1.20. VCPU_RESTART (0x00000058)

V-CPU restart request

Table 1.53. VCPU_RESTART Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															VCPU_RESTART
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO

Table 1.54. VCPU_RESTART Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	VCPU_RESTART	WO	This register restarts V-CPU from the reset vector without clearing H/W logic. 0: DO NOTHING 1: RESTART REQUEST	0x0

Confidential
Starfive Inc.

1.2.3.1.21. VPU_CLK_MASK (0x0000005C)

Clock gating control register for each clock domain

Table 1.55. VPU_CLK_MASK Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					MCLK_EN	VCLK_EN	ACLK_CPU_EN	ACLK_EN								BCLK_EN								CCLK_EN							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.56. VPU_CLK_MASK Field Description

Bit	Name	Type	Function	Reset Value
[31:27]	RSVD	RO	Reserved	0x0
[26]	MCLK_EN	RW	MCLK domain gating	0x0
[25]	VCLK_EN	RW	VCLK domain gating	0x0
[24]	ACLK_CPU_EN	RW	ACLK domain (for V-CPU) gating	0x0
[23:16]	ACLK_EN	RW	ACLK domain (for each V-CORE) gating	0x0
[15:8]	BCLK_EN	RW	BCLK domain (for each V-CORE) gating	0x0
[7:0]	CCLK_EN	RW	CCLK domain (for each V-CORE) gating	0x0

1.2.3.1.22. VPU_REMAP_CTRL (0x00000060)

VPU remaps addresses it uses between physical memory and virtual memory for efficient address management. VPU works with virtual memory addresses that are translated to phsyical addresses. Host processor needs to assign vCPU code buffer to VPU_REMAP_PADDR and VPU_REMAP_VADDR to 0, so that VPU can start by reading from VPU_REMAP_PADDR considering it is its base address 0. This register has configuration fields for remapping.

Table 1.57. VPU_REMAP_CTRL Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REMAP_GLOBEN	RSVD	RSVD	RSVD					AXIID_PROC				ENDIAN				REMAP_IDX				REMAP_PAGE_SIZE_EN	REMAP_ATTR	REMAP_BERR	REMAP_PSIZE								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.58. VPU_REMAP_CTRL Field Description

Bit	Name	Type	Function	Reset Value
[31]	REMAP_GLOBEN	RW	Remap global configuration enable [23:16] is valid only if this flag is *1*. Thus, global configurations such as AXIID_PROC and ENDIAN is valid only if REMAP_GLOBEN is *1*. If not, the value in the fields are ignored.	0x0

Confidential Starlive Inc.

Bit	Name	Type	Function	Reset Value
[30]	RSVD	RW	Reserved	0x0
[29]	RSVD	RW	Reserved	0x0
[28:24]	RSVD	RW	Reserved	0x0
[23:20]	AXIID_PROC	RW	Upper AXI-ID for processor bus to distinguish guest OS For virtualization only. Use this value from higher bit of the 4 bits.	0x0
[19:16]	ENDIAN	RW	Endianness for memory access Endian control for processor memory transaction (except stream)	0x0
[15:12]	REMAP_IDX	RW	Remap index Only 0 to 3 are valid.	0x0
[11]	REMAP_PAGE_SIZE_EN	RW	Enable to update a Remap Page Size. In other words, Remap Page Size can change only if this bit is 1.	0x0
[10]	REMAP_ATTR	RW	Region Attribute - Bypass 0: Normal 1: Bypass region	0x0
[9]	REMAP_BERR	RW	Region Attribute - Bus Error 0: Normal 1: Make Bus Error for the region	0x0
[8:0]	REMAP_PSIZE	RW	Remap Page Size (page base should be aligned in 4K region) 0x001: 4K 0x002: 8K 0x004: 16K 0x008: 32K 0x010 : 64K 0x020 : 128K 0x040 : 256K 0x080: 512K 0x100: 1M	0x0

Confidential StarFive Inc.

1.2.3.1.23. VPU_REMAP_VADDR (0x00000064)

Remap region base address in virtual address space

Table 1.59. VPU_REMAP_VADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VPU_REMAP_VADDR																				RSVD											
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.60. VPU_REMAP_VADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:12]	VPU_REMAP_VADDR	RW	Virtual address space is an address generated by V-CPU. Section address should be aligned to 4KB boundary.	0x0

Bit	Name	Type	Function	Reset Value
			CAUTION> In case of CODE section (which has REMAP IDX 0), virtual address for the section should be 0x0, since V-CPU always start booting up from virtual address 0. If not, VPU can not boot-up.	
[11:0]	RSVD	RO	Reserved	0x0

1.2.3.1.24. VPU_REMAP_PADDR (0x00000068)

Remap region base address in physical address space

Table 1.61. VPU_REMAP_PADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VPU_REMAP_PADDR																				RSVD											
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.62. VPU_REMAP_PADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:12]	VPU_REMAP_PADDR	RW	Real address(physical address) as a pair of virtual address. It should aligned to 4KB boundary.	0x0
[11:0]	RSVD	RO	Reserved	0x0

1.2.3.1.25. VPU_REMAP_CORE_START (0x0000006C)

VPU Start Request

Table 1.63. VPU_REMAP_CORE_START Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															VPU_REMAP_CORE_START
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW

Table 1.64. VPU_REMAP_CORE_START Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	VPU_REMAP_CORE_START	RW	It starts VPU after initial setting has been done. After setting up remap or initial configuartion, host should set this register to init VPU.	0x0

Confidential StarFive Inc.

1.2.3.1.26. VPU_BUSY_STATUS (0x00000070)

VPU Busy Status

Table 1.65. VPU_BUSY_STATUS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															VPU_BUSY_STATUS
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW

Table 1.66. VPU_BUSY_STATUS Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	VPU_BUSY_STATUS	RW	Command status check - command reentrance check This flag should be set as *1* just before send a command. If this flag is *1*, another command can not sent to VPU. This flag is cleared by VPU.	0x0

1.2.3.1.27. VPU_HALT_STATUS (0x00000074)

Reserved for Report VPU status (not implemented yet)

Table 1.67. VPU_HALT_STATUS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																											VPU_HALT_STATUS					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.68. VPU_HALT_STATUS Field Description

Bit	Name	Type	Function	Reset Value
[31:5]	RSVD	RO	Reserved	0x0
[4:0]	VPU_HALT_STATUS	RO	For power control / status checking of V-CPU Remappable status	0x0

1.2.3.1.28. VPU_VCORE_PARSE_STATUS (0x00000078)

Reserved for VPU status reproting (Not implemented yet)

Table 1.69. VPU_VCORE_PARSE_STATUS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

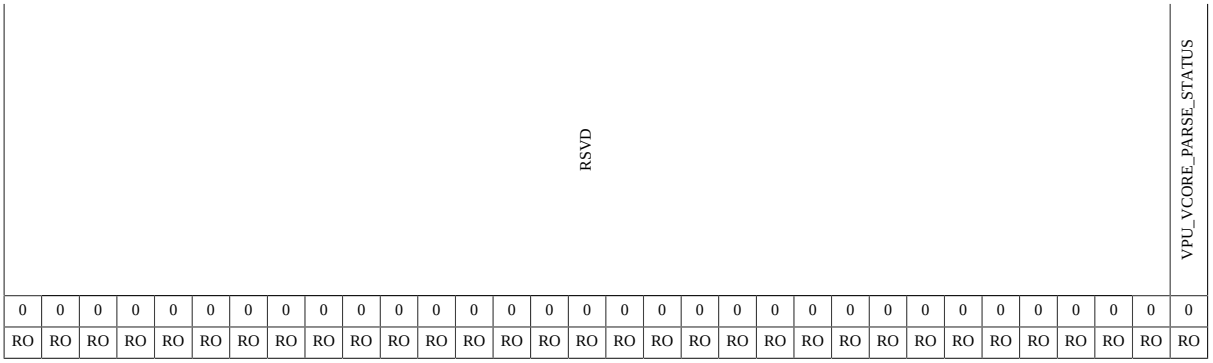


Table 1.70. VPU_VCORE_PARSE_STATUS Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	VPU_VCORE_PARSE_STATUS	RO	V-CORE pasring status - busy	0x0

1.2.3.1.29. VPU_VCORE_DEC_STATUS (0x0000007C)

Reserved for VPU status reproting (Not implemented yet)

Table 1.71. VPU_VCORE_DEC_STATUS Bit Assignment

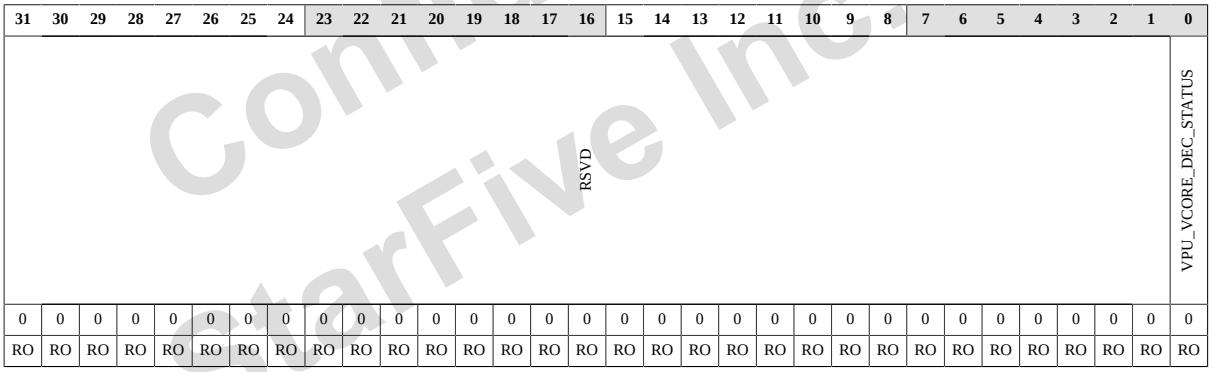


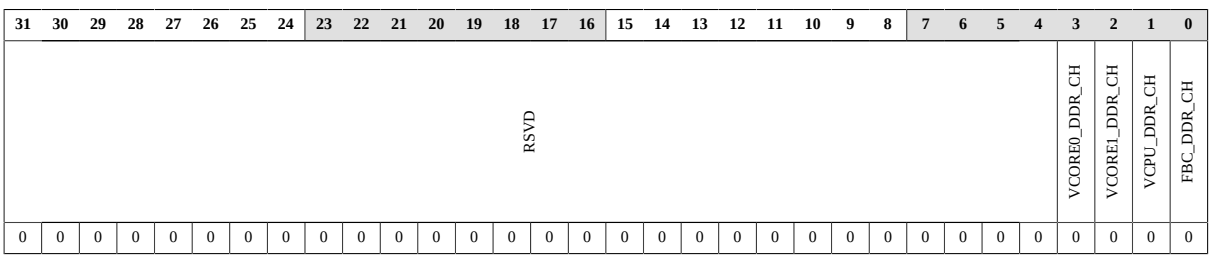
Table 1.72. VPU_VCORE_DEC_STATUS Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	VPU_VCORE_DEC_STATUS	RO	V-CORE decoding status - LF write-back	0x0

1.2.3.1.30. VCPU_DDR_CH_SELECT (0x000000F8)

DDR channel selector

Table 1.73. VCPU_DDR_CH_SELECT Bit Assignment



Confidential StarFive Inc.

RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW

Table 1.74. VCPU_DDR_CH_SELECT Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3]	VCORE0_DDR_CH	RW	V-CORE0 DDR channel 0: ch0 1: ch1	0x0
[2]	VCORE1_DDR_CH	RW	V-CORE1 DDR channel 0: ch0 1: ch1	0x0
[1]	VCPU_DDR_CH	RW	VCPU DDR channel 0: ch0 1: ch1	0x0
[0]	FBC_DDR_CH	RW	FBC/FBD DDR channel 0: ch0 1: ch1	0x0

Confidential
StarFive Inc.

1.2.3.2. Command I/O Interface Description

Note Basically, the reset values of command I/O interface are pseudo-values. But, it can be initialized as 0x0 when the host ran INIT_VPU command and cleared it.

1.2.3.2.1. INIT_VPU Command Parameter Registers

These are command I/O registers where host processor can set arguments for the INIT_VPU command or get return values.

1.2.3.2.1.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.75. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.76. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future • 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.1.2. INIT_OPTION (0x0000010C)

Reserved

Table 1.77. INIT_OPTION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INIT_OPTION																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.78. INIT_OPTION Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	INIT_OPTION	RW	VPU initialization option	0x0

1.2.3.2.1.3. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.79. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.80. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.1.4. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.81. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.82. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.1.5. ADDR_CODE_BASE (0x00000118)

Code Buffer Base Address

Table 1.83. ADDR_CODE_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE_BUF_BASE																				RSVD											
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.84. ADDR_CODE_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:12]	CODE_BUF_BASE	R/W	Base address of V-CPU micro code buffer It should be aligned to 4KB range.	0x0
[11:0]	RSVD	RO	Reserved	0x0

Confidential Starfive Inc.

1.2.3.2.1.6. CODE_SIZE (0x0000011C)

Code Buffer Size

Table 1.85. CODE_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE_BUF_SIZE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.86. CODE_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CODE_BUF_SIZE	R/W	Size of CODE buffer It should be aligned to 4KB range.	0x0

1.2.3.2.1.7. CODE_PARAM (0x00000120)

Parameter for code buffer

Table 1.87. CODE_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								CODE_AXIID				CODE_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.88. CODE_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:8]	RSVD	RO	Reserved	0x0
[7:4]	CODE_AXIID	R/W	AXI-ID for the V-CPU part (for virtualization)	0x0
[3:0]	CODE_ENDIAN	R/W	Endianness	0x0

1.2.3.2.1.8. HW_OPTION (0x00000124)

VPU hardware options

Table 1.89. HW_OPTION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																				UART_OPTION				RSVD						CACHE_DISABLE	USE_DEBUG

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.90. HW_OPTION Field Description

Bit	Name	Type	Function	Reset Value
[31:12]	RSVD	RO	Reserved	0x0
[11:8]	UART_OPTION	R/W	Reserved for debug UART option	0x0
[7:2]	RSVD	RO	Reserved	0x0
[1]	CACHE_DISABLE	R/W	It disables use of cache.	0x0
[0]	USE_DEBUG	R/W	It uses debug mode.	0x0

1.2.3.2.1.9. TIMEOUT_CNT (0x00000134)

Reserved

Table 1.91. TIMEOUT_CNT Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TIME_OUT_CNT															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.92. TIMEOUT_CNT Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	TIME_OUT_CNT	R/W	Watchdog timer setting value	0x0

Confidential
StarFive Inc.

1.2.3.2.2. SET_PARAM Command(COMMON) Parameter Registers

There are command I/O registers where host processor can set arguments for the SET_PARAM command or get return values. These are only when CMD_ENC_SET_PARAM_OPTION(0x10C) is set to 0, all of which work in the common setting mode.

1.2.3.2.2.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.93. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.94. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			Reserved for future • 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.2.2. CORE_INDEX (0x00000104)

V-CORE Index
This is a common parameter register for almost all commands, except VPU control commands such as INIT_VPU, GET_FW_VERSION, SLEEP_VPU, and WAKEUP_VPU.

Table 1.95. CORE_INDEX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VCORE_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.96. CORE_INDEX Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	VCORE_INDEX	RW	V-CORE index - reserved for future use, so please set this as 0.	0x0

1.2.3.2.2.3. INST_INDEX_COD_STD (0x00000108)

Codec Standard and Instance Index
This is a common parameter register for almost all commands (Reserved for future use)

Table 1.97. INST_INDEX_COD_STD Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODEC_STD																INST_INDEX															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.98. INST_INDEX_COD_STD Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CODEC_STD	R/W	Codec standard to run HEVC_DEC = 0x0 HEVC_ENC = 0x1	0x0
[15:0]	INST_INDEX	R/W	Instance Index VPU can decode more than one decoding instance simultaneously. If multiple instances are running, each instance must have their own process index that is assigned by this register. For example, two instances are running simultaneously, the first instance has the process index 0, the second instance has the process index 1. Instance index shall be in the range of 0 to 65535, inclusive.	0x0

1.2.3.2.2.4. CMD_ENC_SET_PARAM_OPTION (0x0000010C)

Host should set this as one of the following while using SET_PARAM command.
Depending on this, SET_PARAM command parameter registers from 0x15C have different meanings.

Table 1.99. CMD_ENC_SET_PARAM_OPTION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														SET_PARAM_OPTION	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Confidential
StarFive Inc.

Table 1.100. CMD_ENC_SET_PARAM_OPTION Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	SET_PARAM_OPTION	R/W	0: It enables to set COMMON registers. 1: It enables to set CUSTOM_GOP registers. 3: It enables to set VUI registers.	0x0

1.2.3.2.2.5. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.101. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.102. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.2.6. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.103. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.104. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.2.7. BS_START_ADDR (0x00000120)

Bitstream buffer start address

Table 1.105. BS_START_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS_START_ADDR																												RSVD			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

Table 1.106. BS_START_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	BS_START_ADDR	R/W	Start Address of Bitstream Buffer (fixed in ring buffer mode) It should be aligned in bus width (128bit/16 byte).	0x0

Confidential Starfive Inc.

Bit	Name	Type	Function	Reset Value
[3:0]	RSVD	RO	Reserved	0x0

1.2.3.2.2.8. BS_SIZE (0x00000124)

Bitstream buffer size

Table 1.107. BS_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS_BUF_SIZE																												RSVD			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

Table 1.108. BS_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	BS_BUF_SIZE	R/W	Size of Bistream buffer (fixed in ring buffer mode) It should be aligned in bus width (128bit/16byte).	0x0
[3:0]	RSVD	RO	Reserved	0x0

1.2.3.2.2.9. BS_PARAM (0x00000128)

Bitstream buffer parameters

Table 1.109. BS_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										BS_SLICE_WR_PTR_INTERRUPT_FLAG	BS_WRAP_AROUND_FLAG	BS_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.110. BS_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:6]	RSVD	RO	Reserved	0x0
[5]	BS_SLICE_WR_PTR_INTERRUPT_FLAG	R/W	It updates wr_ptr each slice.	0x0
[4]	BS_WRAP_AROUND_FLAG	R/W	A wrap around flag	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[3:0]	BS_ENDIAN	R/W	Endianness of bitstream buffer	0x0

1.2.3.2.2.10. BS_OPTIONS (0x0000012C)

Bistream buffer option

Table 1.111. BS_OPTIONS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														STREAM_END	EXPLICIT_END
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.112. BS_OPTIONS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	R/W	Reserved	0x0
[1]	STREAM_END	R/W	This field makes VPU assumes Stream End when there is no stream to feed in the buffer.	0x0
[0]	EXPLICIT_END	R/W	Explicit End When this field is set to 1, VPU assumes that bitstream buffer has one NAL unit or more for decoding a signle frame - it works based on the BS_NAL_LEVEL_PUMP register on. Then it follows the tasks below. 1. VPU decodes until the end of bitstream buffer (WR_PTR) even if the last 3 bytes are all 0. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU performs what it is supposed to do with the specified task in BS_SHORTAGE_OPTION of BS_PARAM. If this flag is 0, 1. VPU decodes to the almost end of bitstream buffer (WR_PTR), but not to some bytes (less than 3). It intentionally does not consume some a few bytes, because VPU is not sure if the last bytes are the start code of next NAL or they might not fill the offset in the CABAC. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU stops decoding and waits for more bitstream to be filled. Then host processor can do either feed bitstream more or set EXPLICIT_END to complete decoding anyhow. Bitstream_empty interrupt is asserted when EXPLITCT_END = 0 or when bitstream buffer is near empty not real empty for seamless decoding. **Caution** Host processor can set this register any time, but cannot clear during command processing.	0x0

1.2.3.2.2.11. BS_RD_PTR (0x00000130)

Bistream Buffer Read Pointer

Confidential Starfive Inc.

Table 1.113. BS_RD_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD_PTR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.114. BS_RD_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RD_PTR	R/W	Start address of bitstream for handling current command Host processor cannot update this register in the middle of decoding. Updating before start decoding is only allowed. VPU updates this RD_PTR as the scheme described below. • When BS_NAL_LEVEL_PUMP of BS_PARAM is 1, VPU updates RET_ADDR_BS_RD_PTR to the end address of the NAL. • When BS_NAL_LEVEL_PUMP of BS_PARAM is 0, VPU updates RET_ADDR_BS_RD_PTR as end address of the last NAL for a frame.	0x0

1.2.3.2.2.12. BS_WR_PTR (0x00000134)

Bistream Buffer Write Pointer

Table 1.115. BS_WR_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WR_PTR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.116. BS_WR_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WR_PTR	R/W	End address of bitstream for handling current command Host can update this register anytime. If a bitstream_empty interrupt is asserted, host processor should do either feed more bitstream and update WR_PTR or set EXPLICIT_END to complete decoding anyhow.	0x0

1.2.3.2.2.13. ADDR_WORK_BASE (0x00000138)

Work Buffer Base Address

Table 1.117. ADDR_WORK_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORK_BUF_BASE																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.118. ADDR_WORK_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WORK_BUF_BASE	R/W	Base address of work buffer for each instance This address should be aligned in 4K boundary.	0x0

1.2.3.2.2.14. WORK_SIZE (0x0000013C)

Work Buffer Size

Table 1.119. WORK_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORK_BUF_SIZE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.120. WORK_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WORK_BUF_SIZE	R/W	Size of work buffer (4K boundary) The size of work buffer should be larger than required minimum work buffer size. This address should be aligned in 4K boundary.	0x0

1.2.3.2.2.15. WORK_PARAM (0x00000140)

Work Buffer Parameter

Table 1.121. WORK_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												WORK_BUF_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W

Table 1.122. WORK_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	WORK_BUF_ENDIAN	R/W	Endianness of work buffer	0x0

Confidential Starnix Inc.

Bit	Name	Type	Function	Reset Value
			Not implemented yet - VPU uses work buffer as internal purpose so endianness is not important.	

1.2.3.2.2.16. ADDR_TEMP_BASE (0x00000144)

Temporal Buffer Base Address

Table 1.123. ADDR_TEMP_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEMP_BUF_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.124. ADDR_TEMP_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TEMP_BUF_BASE	R/W	Base address of temporal buffer for this frame **Note** Each V-CORE should set this field for its own temporal buffer.	0x0

1.2.3.2.2.17. TEMP_SIZE (0x00000148)

Temporal Buffer Size

Table 1.125. TEMP_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEMP_BUF_SIZE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.126. TEMP_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TEMP_BUF_SIZE	R/W	Temporal buffer size for this frame	0x0

1.2.3.2.2.18. TEMP_PARAM (0x0000014C)

Temporal Buffer Parameter

Table 1.127. TEMP_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

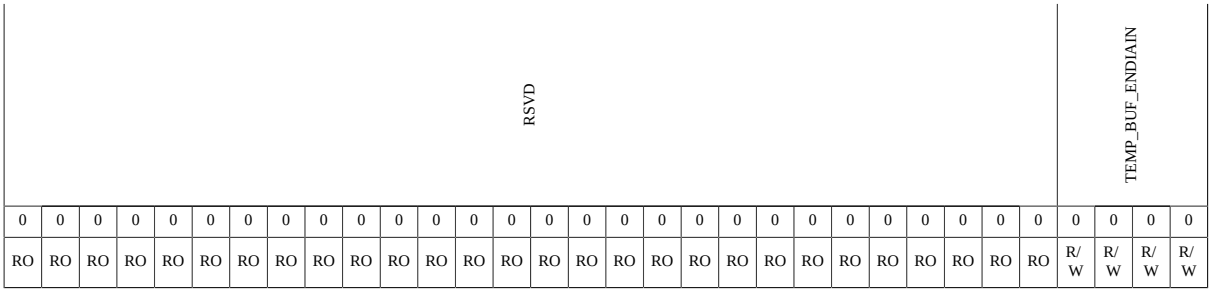


Table 1.128. TEMP_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	TEMP_BUF_ENDIAIN	R/W	Endianness of temporal buffer **Note** Each V-CORE should set this register for its own temporal buffer.	0x0

1.2.3.2.2.19. ADDR_SEC_AXI (0x00000150)

Secondary AXI Base Address
It is valid only when USE_SEC_AXI is not 0.

Table 1.129. ADDR_SEC_AXI Bit Assignment

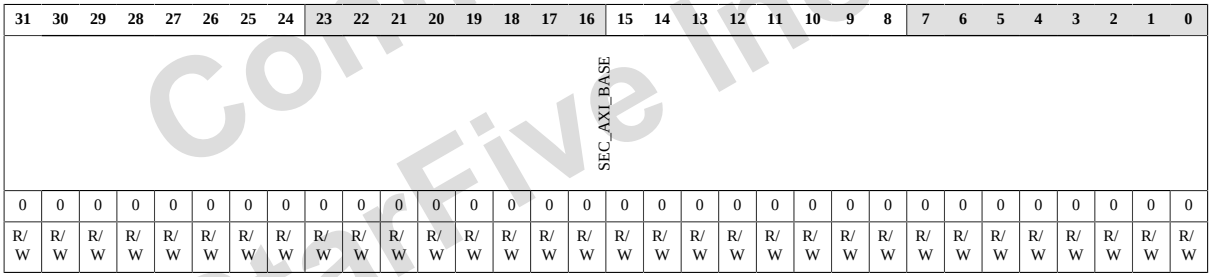


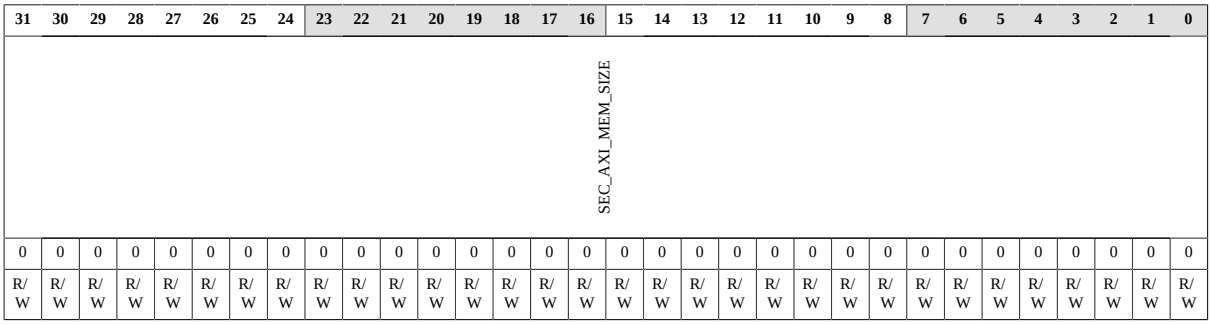
Table 1.130. ADDR_SEC_AXI Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SEC_AXI_BASE	R/W	The base address of secondary AXI memory	0x0

1.2.3.2.2.20. SEC_AXI_SIZE (0x00000154)

Seconary AXI Memory Size
It is valid only when USE_SEC_AXI is not 0.

Table 1.131. SEC_AXI_SIZE Bit Assignment



Confidential
StarFive Inc.

Table 1.132. SEC_AXI_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SEC_AXI_MEM_SIZE	R/W	The size of secondary AXI memory	0x0

1.2.3.2.2.21. USE_SEC_AXI (0x00000158)

Secondary AXI Usage option

Table 1.133. USE_SEC_AXI Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																SEC_AXI_LF_ROW	RSVD			SEC_AXI_RDO_ENABLE	RSVD										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Confidential
StarFive Inc.

Table 1.134. USE_SEC_AXI Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15]	SEC_AXI_LF_ROW	R/W	Use 2nd AXI temp buffer for Read channel of LF row buffer.	0x0
[14:12]	RSVD	RO	Reserved	0x0
[11]	SEC_AXI_RDO_ENABLE	R/W	Use 2nd AXI temp buffer for Read channel of RDO row buffer.	0x0
[10:0]	RSVD	RO	Reserved	0x0

1.2.3.2.2.22. CMD_ENC_SET_PARAM_ENABLE (0x0000015C)

Encoder Parameter Change Setting

Table 1.135. CMD_ENC_SET_PARAM_ENABLE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					ENABLE_SET_NR_WEIGHT	ENABLE_SET_NR_PARAM	ENABLE_SET_RESERVED	ENABLE_SET_RESERVED	ENABLE_SET_RC_TARGET_RATE	ENABLE_SET_RC_TRANS_RATE	ENABLE_SET_NUM_TICKS_POC_DIFF_ONE	ENABLE_SET_TIME_SCALE	ENABLE_SET_NUM_UNITS_IN_TICK	ENABLE_SET_RC_RESERVED1	ENABLE_SET_RC_RESERVED0	ENABLE_SET_RC_BIT_RATIO_LAYER_4_7	ENABLE_SET_RC_BIT_RATIO_LAYER_0_3	ENABLE_SET_RC_MIN_MAX_QP	ENABLE_SET_RC_PARAM	ENABLE_SET_ENC_RESERVED	ENABLE_SET_ENC_PARAM	ENABLE_SET_SEQ_INTRA_REFRESH	ENABLE_SET_SEQ_DEPENDENT_SLICE	ENABLE_SET_SEQ_INDEPENDENT_SLICE	ENABLE_SET_SEQ_FRAME_RATE	ENABLE_SET_SEQ_CONF_WIN_LEFT_RIGHT	ENABLE_SET_SEQ_CONF_WIN_TOP_BOT	ENABLE_SET_SEQ_INTRA_PARAM	ENABLE_SET_SEQ_GOP_PARAM	ENABLE_SET_SEQ_PARAM	ENABLE_SET_SEQ_SRC_SIZE

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.136. CMD_ENC_SET_PARAM_ENABLE Field Description

Bit	Name	Type	Function	Reset Value
[31:27]	RSVD	RO	Reserved	0x0
[26]	ENABLE_SET_NR_WEIGHT	R/W	It enables to set NR_WEIGHT.	0x0
[25]	ENABLE_SET_NR_PARAM	R/W	It enables to set NR_PARAM.	0x0
[24]	ENABLE_SET_RESERVED	R/W	Reserved	0x0
[23]	ENABLE_SET_RESERVED	R/W	Reserved	0x0
[22]	ENABLE_SET_RC_TARGET_RATE	R/W	It enables to set RC_TARGET_RATE.	0x0
[21]	ENABLE_SET_RC_TRANS_RATE	R/W	It enables to set RC_TRANS_RATE.	0x0
[20]	ENABLE_SET_NUM_TICKS_POC_DIFF_ONE	R/W	It enables to set NUM_TICKS_POC_DIFF_ONE.	0x0
[19]	ENABLE_SET_TIME_SCALE	R/W	It enables to set TIME_SCALE.	0x0
[18]	ENABLE_SET_NUM_UNITS_IN_TICK	R/W	It enables to set NUM_UNITS_IN_TICK.	0x0
[17]	ENABLE_SET_RC_RESERVED1	R/W	Reserved	0x0
[16]	ENABLE_SET_RC_RESERVED0	R/W	Reserved	0x0
[15]	ENABLE_SET_RC_BIT_RATIO_LAYER_4_7	R/W	It enables to set RC_BIT_RATIO_LAYER_4_7.	0x0
[14]	ENABLE_SET_RC_BIT_RATIO_LAYER_0_3	R/W	It enables to set RC_BIT_RATIO_LAYER_0_3.	0x0
[13]	ENABLE_SET_RC_MIN_MAX_QP	R/W	It enables to set RC_MIN_MAX_QP.	0x0
[12]	ENABLE_SET_RC_PARAM	R/W	It enables to set RC_PARAM.	0x0
[11]	ENABLE_SET_ENC_RESERVED	R/W	Reserved	0x0
[10]	ENABLE_SET_ENC_PARAM	R/W	It enables to set SEQ_ENC_PARAM.	0x0
[9]	ENABLE_SET_SEQ_INTRA_REFRESH	R/W	It enables to set SEQ_INTRA_REFRESH.	0x0
[8]	ENABLE_SET_SEQ_DEPENDENT_SLICE	R/W	It enables to set SEQ_DEPENDENT_SLICE.	0x0
[7]	ENABLE_SET_SEQ_INDEPENDENT_SLICE	R/W	It enables to set SEQ_INDEPENDENT_SLICE.	0x0
[6]	ENABLE_SET_SEQ_FRAME_RATE	R/W	It enables to set SEQ_FRAME_RATE.	0x0
[5]	ENABLE_SET_SEQ_CONF_WIN_LEFT_RIGHT	R/W	It enables to set SEQ_CONF_WIN_LEFT_RIGHT.	0x0
[4]	ENABLE_SET_SEQ_CONF_WIN_TOP_BOT	R/W	It enables to set SEQ_CONF_WIN_TOP_BOT.	0x0
[3]	ENABLE_SET_SEQ_INTRA_PARAM	R/W	It enables to set SEQ_INTRA_PARAM.	0x0
[2]	ENABLE_SET_SEQ_GOP_PARAM	R/W	It enables to set SEQ_GOP_PARAM.	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[1]	ENABLE_SET_SEQ_PARAM	R/W	It enables to set SEQ_PARAM.	0x0
[0]	ENABLE_SET_SEQ_SRC_SIZE	R/W	It enables to set SEQ_SRC_SIZE.	0x0

1.2.3.2.2.23. CMD_ENC_SEQ_SRC_SIZE (0x00000160)

A size of Source Picture

Table 1.137. CMD_ENC_SEQ_SRC_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_HEIGHT																SRC_WIDTH															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.138. CMD_ENC_SEQ_SRC_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	SRC_HEIGHT	R/W	A height of source picture in pixel (128 ~ 8192)	0x0
[15:0]	SRC_WIDTH	R/W	A width of source picture in pixel (256 ~ 8192)	0x0

1.2.3.2.2.24. CMD_ENC_SEQ_PARAM (0x0000016C)

Table 1.139. CMD_ENC_SEQ_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHROMA_CR_QP_OFFSET					CHROMA_CB_QP_OFFSET					CONSTRAINED_INTRA_PRED	USE_LOSSLESS_CODING	CHROMA_FORMAT_IDC		BIT_DEPTH		TIER_IDC		LEVEL_IDC										PROFILE_IDC			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.140. CMD_ENC_SEQ_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:27]	CHROMA_CR_QP_OFFSET	R/W	A value of chroma(Cr) QP offset (-10 ~ 10)	0x0
[26:22]	CHROMA_CB_QP_OFFSET	R/W	A value of chroma(Cb) QP offset (-10 ~ 10)	0x0
[21]	CONSTRAINED_INTRA_PRED	R/W	It enables constrained intra prediction.	0x0
[20]	USE_LOSSLESS_CODING	R/W	It enables lossless coding.	0x0
[19:18]	CHROMA_FORMAT_IDC	R/W	A chroma format indicator (0 - 4:2:0)	0x0
[17:14]	BIT_DEPTH	R/W	Bit depth (8 or 10)	0x0
[13:12]	TIER_IDC	R/W	A tier indicator	0x0

Confidential
StarFive Inc.

Bit	Name	Type	Function	Reset Value
			0 - main 1 - high	
[11:3]	LEVEL_IDC	R/W	A level indicator (level * 10) 0 - firmware determines a level.	0x0
[2:0]	PROFILE_IDC	R/W	A profile indicator 0 - firmware determines a profile. 1 - main 2 - main10	0x0

1.2.3.2.2.25. CMD_ENC_SEQ_GOP_PARAM (0x00000170)

Table 1.141. CMD_ENC_SEQ_GOP_PARAM Bit Assignment

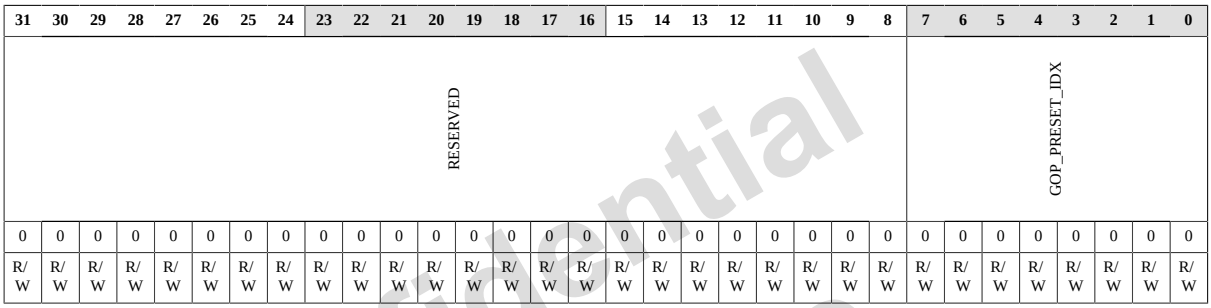


Table 1.142. CMD_ENC_SEQ_GOP_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:8]	RESERVED	R/W	Reserved	0x0
[7:0]	GOP_PRESET_IDX	R/W	A GOP structure option 0: Custom GOP 1 : I-I-I-I,..I (all intra, gop_size=1) 2 : I-P-P-P,… P (consecutive P, gop_size=1) 6 : I-P-P-P-P,… (consecutive P, gop_size=4)	0x0

1.2.3.2.2.26. CMD_ENC_SEQ_INTRA_PARAM (0x00000174)

Table 1.143. CMD_ENC_SEQ_INTRA_PARAM Bit Assignment

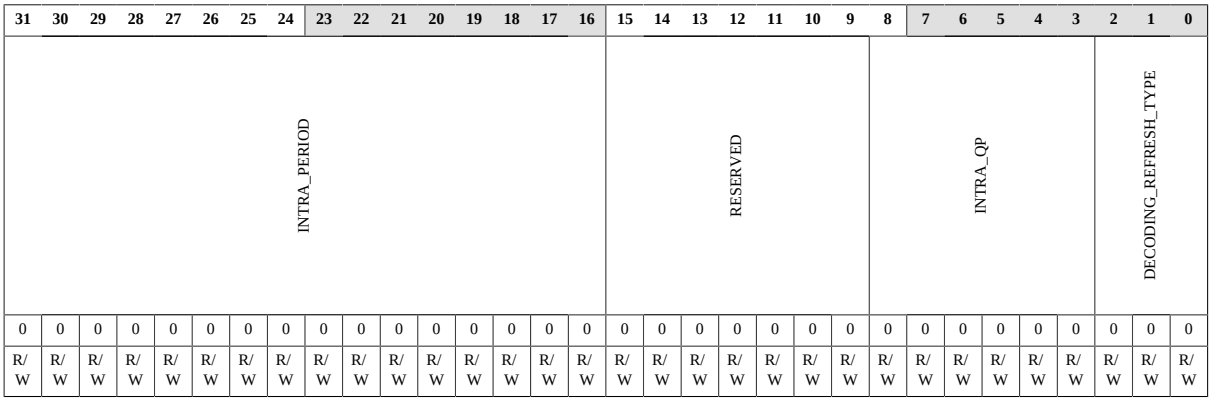


Table 1.144. CMD_ENC_SEQ_INTRA_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	INTRA_PERIOD	R/W	A period of intra picture (0 ~ 1024)	0x0

Confidential
StarFive Inc.

Bit	Name	Type	Function	Reset Value
			0 - implies an infinite period	
[15:9]	RESERVED	R/W	Reserved	0x0
[8:3]	INTRA_QP	R/W	A quantization parameter of intra picture (0 ~ 51)	0x0
[2:0]	DECODING_REFRESH_TYPE	R/W	A decoding refresh type of intra picture 0 - Non-IRAP 1 - CRA 2 - IDR	0x0

1.2.3.2.2.27. CMD_ENC_SEQ_CONF_WIN_TOP_BOT (0x00000178)

Table 1.145. CMD_ENC_SEQ_CONF_WIN_TOP_BOT Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONFORMACE_WINDOW_SIZE_BOTTOM																CONFORMACE_WINDOW_SIZE_TOP															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.146. CMD_ENC_SEQ_CONF_WIN_TOP_BOT Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CONFORMACE_WINDOW_SIZE_BOTTOM	R/W	A conformance bottom window size (0 ~ 8192)	0x0
[15:0]	CONFORMACE_WINDOW_SIZE_TOP	R/W	A conformance top window size (0 ~ 8192)	0x0

1.2.3.2.2.28. CMD_ENC_SEQ_CONF_WIN_LEFT_RIGHT (0x0000017C)

Table 1.147. CMD_ENC_SEQ_CONF_WIN_LEFT_RIGHT Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONFORMANCE_WINDOW_SIZE_RIGHT																CONFORMANCE_WINDOW_SIZE_LFET															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.148. CMD_ENC_SEQ_CONF_WIN_LEFT_RIGHT Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CONFORMANCE_WINDOW_SIZE_RIGHT	R/W	A conformance right window size (0 ~ 8192)	0x0
[15:0]	CONFORMANCE_WINDOW_SIZE_LFET	R/W	A conformance left window size (0 ~ 8192)	0x0

1.2.3.2.2.29. CMD_ENC_SEQ_FRAME_RATE (0x00000180)

Table 1.149. CMD_ENC_SEQ_FRAME_RATE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRAME_RATE_IDC																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.150. CMD_ENC_SEQ_FRAME_RATE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FRAME_RATE_IDC	R/W	A frame rate indicator (x 1024)	0x0

1.2.3.2.2.30. CMD_ENC_SEQ_INDEPENDENT_SLICE (0x00000184)

Table 1.151. CMD_ENC_SEQ_INDEPENDENT_SLICE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGUMENT																RESERVED													SLICE_MODE		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.152. CMD_ENC_SEQ_INDEPENDENT_SLICE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	ARGUMENT	R/W	A size of slice for the slice mode (2^16 - 1) • The number of CTU for slice mode of 1	0x0
[15:3]	RESERVED	R/W	Reserved	0x0
[2:0]	SLICE_MODE	R/W	An independent slice mode 0 - No multi-slice 1 - CTU based slice	0x0

Confidential Starfive Inc.

1.2.3.2.2.31. CMD_ENC_SEQ_DEPENDENT_SLICE (0x00000188)

Table 1.153. CMD_ENC_SEQ_DEPENDENT_SLICE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGUMENT																RESERVED													SLICE_MODE		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.154. CMD_ENC_SEQ_DEPENDENT_SLICE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	ARGUMENT	R/W	A size of slice for the slice mode (2^16 - 1) • The number of CTU for slice mode of 1 • The number of byte for slice mode of 2	0x0
[15:3]	RESERVED	R/W	Reserved	0x0
[2:0]	SLICE_MODE	R/W	A dependent slice mode 0 - No multi-slice 1 - CTU based slice 2 - Byte based slice	0x0

1.2.3.2.2.32. CMD_ENC_SEQ_INTRA_REFRESH (0x0000018C)

Table 1.155. CMD_ENC_SEQ_INTRA_REFRESH Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRA_REFRESH_ARGUMENT																RESERVED													INTRA_REFRESH_MODE		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.156. CMD_ENC_SEQ_INTRA_REFRESH Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	INTRA_REFRESH_ARGUMENT	R/W	An argument of intra refresh mode (2^16 - 1) • The number of consecutive CTU rows for intra refresh mode of 1 • The number of consecutive CTU columns for intra refresh mode of 2 • A step size in CTU for intra refresh mode of 3	0x0
[15:3]	RESERVED	R/W	Reserved	0x0
[2:0]	INTRA_REFRESH_MODE	R/W	An intra refresh mode	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			0 - No intra refresh 1 - Row 2 - Column 3 - A step size in CTU	

1.2.3.2.2.33. CMD_ENC_PARAM (0x00000190)

Display flag

Table 1.157. CMD_ENC_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			RESERVED	USE_INTRA_IN_INTER_SLICE	RESERVED	RESERVED	TC_OFFSET_DIV2				BETA_OFFSET_DIV2				USE_LF_CROSS_SLICE_BOUNDARY	DISABLE_DBK	USE_DYNAMIC_MERGE_32x32	USE_DYNAMIC_MERGE_16X16	USE_DYNAMIC_MERGE_8x8	MAX_NUM_MERGE			RESERVED	USE_TMVP	USE_CU_SIZE			USE_SCALING_LIST	ENABLE_CTU_QP_MAP	USE_RECOMMEND_ENC_PARAM		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.158. CMD_ENC_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:29]	RESERVED	R/W	Reserved	0x0
[28]	RESERVED	R/W	Reserved	0x0
[27]	USE_INTRA_IN_INTER_SLICE	R/W	It enables intra CU in inter slice.	0x0
[26]	RESERVED	R/W	Reserved	0x0
[25]	RESERVED	R/W	Reserved	0x0
[24:21]	TC_OFFSET_DIV2	R/W	TcOffsetDiv2 for deblocking filter	0x0
[20:17]	BETA_OFFSET_DIV2	R/W	BetaOffsetDiv2 for deblocking filter	0x0
[16]	USE_LF_CROSS_SLICE_BOUNDARY	R/W	It enables the use of in-loop filtering across slice boundaries.	0x0
[15]	DISABLE_DBK	R/W	It disables the in-loop deblocking filter.	0x0
[14]	USE_DYNAMIC_MERGE_32x32	R/W	It enables 32x32 CUs to be dynamic merge candidates.	0x0
[13]	USE_DYNAMIC_MERGE_16X16	R/W	It enables 16x16 CUs to be dynamic merge candidates.	0x0
[12]	USE_DYNAMIC_MERGE_8x8	R/W	It enables 8x8 CUs for dynamic merge candidates.	0x0
[11:9]	MAX_NUM_MERGE	R/W	Maximum number of merge candidates (0 ~ 2)	0x0
[8]	RESERVED	R/W	Reserved	0x0
[7]	USE_TMVP	R/W	It enables temporal motion vector prediction.	0x0
[6:4]	USE_CU_SIZE	R/W	It enables CU size.	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			[0] : 8x8 CU [1] : 16x16 CU [2] : fixed 0 **Note** Host should enable all CU sizes due to hardware constraint. The RDO block determines the best CU size for CTU on its own.	
[3]	USE_SCALING_LIST	R/W	It enables scaling list.	0x0
[2]	ENABLE_CTU_QP_MAP	R/W	It enables CTU QP map.	0x0
[1:0]	USE_RECOMMEND_ENC_PARAM	R/W	It uses recommended enc params. 0 : Custom 1 : Recommend enc params 2 ~ 3 : Reserved	0x0

1.2.3.2.2.34. CMD_ENC_RC_MIN_MAX_QP (0x00000194)

Table 1.159. CMD_ENC_RC_MIN_MAX_QP Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRA_QP_OFFSET														MAX_DELTA_QP						MAX_QP						MIN_QP					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.160. CMD_ENC_RC_MIN_MAX_QP Field Description

Bit	Name	Type	Function	Reset Value
[31:18]	INTRA_QP_OFFSET	R/W	A QP offset of intra picture	0x0
[17:12]	MAX_DELTA_QP	R/W	A maximum delta QP for HVS (0 ~ 12)	0x0
[11:6]	MAX_QP	R/W	A maximum QP for rate control (0 ~ 51)	0x0
[5:0]	MIN_QP	R/W	A minimum QP for rate control (0 ~ 51)	0x0

1.2.3.2.2.35. CMD_ENC_RC_PARAM (0x00000198)

Table 1.161. CMD_ENC_RC_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITIAL_DELAY												INITIAL_RC_QP						EN_SEQ_ROI	INIT_BUF_LEVELx8				BIT_ALLOC_MODE		HVS_QP_SCALE			EN_HVS_QP_SCALE	EN_HVS_QP	EN_CU_LEVEL_RC	EN_RATE_CONTROL
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential
Cnm Inc.

Table 1.162. CMD_ENC_RC_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:20]	INITIAL_DELAY	R/W	An inital cpb delay in msec (10 ~ 3000)	0x0
[19:14]	INITIAL_RC_QP	R/W	A value of initial RC QP by host. (0 ~ 51). If host wants VPU to use a firmware-generated initial RC QP, set any value from 52 to 63.	0x0
[13]	EN_SEQ_ROI	R/W	It enables ROI in sequence level.	0x0
[12:9]	INIT_BUF_LEVELx8	R/W	It specifies encoder initial delay (encoder initial delay = InitialDelay * InitBufLevelx8 / 8).	0x0
[8:7]	BIT_ALLOC_MODE	R/W	It specifies picture bits allocation mode.	0x0
[6:4]	HVS_QP_SCALE	R/W	It specifies QP scaling factor for CU QP adjustment.	0x0
[3]	EN_HVS_QP_SCALE	R/W	It enables HVS QP scaling function when EN_HVS_QP is 1.	0x0
[2]	EN_HVS_QP	R/W	It enables HVS (Human Visual System) function for subjective quality enhancement.	0x0
[1]	EN_CU_LEVEL_RC	R/W	It enables CU level rate control.	0x0
[0]	EN_RATE_CONTROL	R/W	It enables rate control.	0x0

1.2.3.2.2.36. CMD_ENC_RC_INTRA_MIN_MAX_QP (0x0000019C)

Table 1.163. CMD_ENC_RC_INTRA_MIN_MAX_QP Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																				MAX_INTRA_QP						MIN_INTRA_QP					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.164. CMD_ENC_RC_INTRA_MIN_MAX_QP Field Description

Bit	Name	Type	Function	Reset Value
[31:12]	RSVD	RO	Reserved	0x0
[11:6]	MAX_INTRA_QP	R/W	A maximum intra QP for rate control (0 ~ 51)	0x0
[5:0]	MIN_INTRA_QP	R/W	A minimum intra QP for rate control (0 ~ 51)	0x0

1.2.3.2.2.37. CMD_ENC_RC_BIT_RATIO_LAYER_0_3 (0x000001A0)

Table 1.165. CMD_ENC_RC_BIT_RATIO_LAYER_0_3 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIXED_BIT_RATIO_3								FIXED_BIT_RATIO_2								FIXED_BIT_RATIO_1								FIXED_BIT_RATIO_0							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confidential StarFive Inc

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.166. CMD_ENC_RC_BIT_RATIO_LAYER_0_3 Field Description

Bit	Name	Type	Function	Reset Value
[31:24]	FIXED_BIT_RATIO_3	R/W	It specifies the ratio of 4th picture bits in encoding order in a GOP (1~255).	0x0
[23:16]	FIXED_BIT_RATIO_2	R/W	It specifies the ratio of 3rd picture bits in encoding order in a GOP (1~255).	0x0
[15:8]	FIXED_BIT_RATIO_1	R/W	It specifies the ratio of 2nd picture bits in encoding order in a GOP (1~255).	0x0
[7:0]	FIXED_BIT_RATIO_0	R/W	It specifies the ratio of 1st picture bits in encoding order in a GOP (1~255).	0x0

1.2.3.2.2.38. CMD_ENC_RC_BIT_RATIO_LAYER_4_7 (0x000001A4)

Table 1.167. CMD_ENC_RC_BIT_RATIO_LAYER_4_7 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIXED_BIT_RATIO_7								FIXED_BIT_RATIO_6								FIXED_BIT_RATIO_5								FIXED_BIT_RATIO_4							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.168. CMD_ENC_RC_BIT_RATIO_LAYER_4_7 Field Description

Bit	Name	Type	Function	Reset Value
[31:24]	FIXED_BIT_RATIO_7	R/W	It specifies the ratio of 8th picture bits in encoding order in a GOP (1~255).	0x0
[23:16]	FIXED_BIT_RATIO_6	R/W	It specifies the ratio of 7th picture bits in encoding order in a GOP (1~255).	0x0
[15:8]	FIXED_BIT_RATIO_5	R/W	It specifies the ratio of 6th picture bits in encoding order in a GOP (1~255).	0x0
[7:0]	FIXED_BIT_RATIO_4	R/W	It specifies the ratio of 5th picture bits in encoding order in a GOP (1~255).	0x0

1.2.3.2.2.39. CMD_ENC_NR_PARAM (0x000001A8)

Table 1.169. CMD_ENC_NR_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NR_NOISE_SIGMA_CR								NR_NOISE_SIGMA_CB								NR_NOISE_SIGMA_Y				ENABLE_NR_EST_NOISE	ENABLE_NR_CR	ENABLE_NR_CB	ENABLE_NR_Y

Confidential
Start Five Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.170. CMD_ENC_NR_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:25]	RESERVED	R/W	Reserved	0x0
[24:18]	NR_NOISE_SIGMA_CR	R/W	It specifies Cr noise standard deviation if ENABLE_NR_EST_NOISE is 0. (0 ~ 255)	0x0
[17:11]	NR_NOISE_SIGMA_CB	R/W	It specifies Cb noise standard deviation if ENABLE_NR_EST_NOISE is 0. (0 ~ 255)	0x0
[10:4]	NR_NOISE_SIGMA_Y	R/W	It specifies Y noise standard deviation if ENABLE_NR_EST_NOISE is 0. (0 ~ 255)	0x0
[3]	ENABLE_NR_EST_NOISE	R/W	It enables VPU to conduct noise estimation for reduction. When this is disabled, noise estimation is carried out ouside VPU.	0x0
[2]	ENABLE_NR_CR	R/W	It enables noise reduction algorithm to Cr component.	0x0
[1]	ENABLE_NR_CB	R/W	It enables noise reduction algorithm to Cb component.	0x0
[0]	ENABLE_NR_Y	R/W	It enables noise reduction algorithm to Y component.	0x0

1.2.3.2.2.40. CMD_ENC_NR_WEIGHT (0x000001AC)

Table 1.171. CMD_ENC_NR_WEIGHT Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		NR_INTER_WEIGHT_CR						NR_INTER_WEIGHT_CB				NR_INTER_WEIGHT_Y				NR_INTRA_WEIGHT_CR						NR_INTRA_WEIGHT_CB					NR_INTRA_WEIGHT_Y				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.172. CMD_ENC_NR_WEIGHT Field Description

Bit	Name	Type	Function	Reset Value
[31:30]	RESERVED	R/W	Reserved	0x0
[29:25]	NR_INTER_WEIGHT_CR	R/W	Weight to CR noise level for inter picture (0 ~ 31)	0x0
[24:20]	NR_INTER_WEIGHT_CB	R/W	Weight to CB noise level for inter picture (0 ~ 31)	0x0
[19:15]	NR_INTER_WEIGHT_Y	R/W	Weight to Y noise level for inter picture (0 ~ 31) **Note** NR_INTER_WEIGHT/4 is multiplied to the noise level that has been estimated. This weight is put for inter frame to be filtered more strongly or more weakly than just with the estimated noise level.	0x0
[14:10]	NR_INTRA_WEIGHT_CR	R/W	Weight to CR noise level for intra picture (0 ~ 31)	0x0
[9:5]	NR_INTRA_WEIGHT_CB	R/W	Weight to CB noise level for intra picture (0 ~ 31)	0x0
[4:0]	NR_INTRA_WEIGHT_Y	R/W	Weight to Y noise level for intra picture (0 ~ 31)	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			Note NR_INTRA_WEIGHT/4 is multiplied to the noise level that has been estimated. This weight is put for intra frame to be filtered more strongly or more weakly than just with the estimated noise level.	

1.2.3.2.2.41. CMD_ENC_NUM_UNITS_IN_TICK (0x000001B0)

Table 1.173. CMD_ENC_NUM_UNITS_IN_TICK Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUM_UNITS_IN_TICK																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.174. CMD_ENC_NUM_UNITS_IN_TICK Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	NUM_UNITS_IN_TICK	R/W	It specifies the number of time units of a clock operating at the frequency time_scale Hz.	0x0

1.2.3.2.2.42. CMD_ENC_TIME_SCALE (0x000001B4)

Table 1.175. CMD_ENC_TIME_SCALE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME_SCALE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.176. CMD_ENC_TIME_SCALE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TIME_SCALE	R/W	It specifies the number of time units that pass in one second.	0x0

1.2.3.2.2.43. CMD_ENC_NUM_TICKS_POC_DIFF_ONE (0x000001B8)

Table 1.177. CMD_ENC_NUM_TICKS_POC_DIFF_ONE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

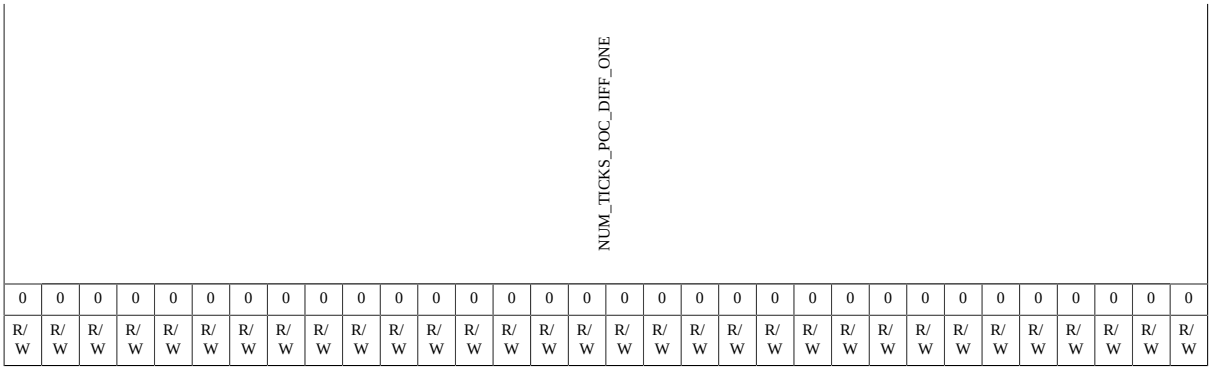


Table 1.178. CMD_ENC_NUM_TICKS_POC_DIFF_ONE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	NUM_TICKS_POC_DIFF_ONE	R/W	It specifies the number of clock ticks corresponding to a difference of picture order count values equal to 1.	0x0

1.2.3.2.2.44. CMD_ENC_RC_TRANS_RATE (0x000001BC)

Table 1.179. CMD_ENC_RC_TRANS_RATE Bit Assignment

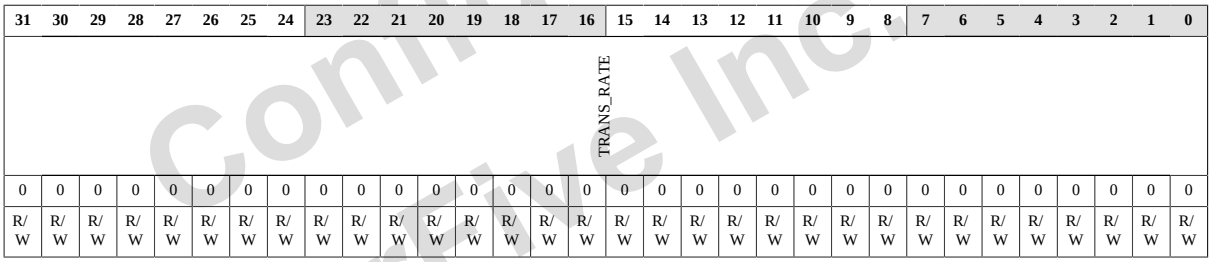


Table 1.180. CMD_ENC_RC_TRANS_RATE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TRANS_RATE	R/W	A peak transmission bitrate in bps (0 ~ 700000000)	0x0

1.2.3.2.2.45. CMD_ENC_RC_TARGET_RATE (0x000001C0)

Table 1.181. CMD_ENC_RC_TARGET_RATE Bit Assignment

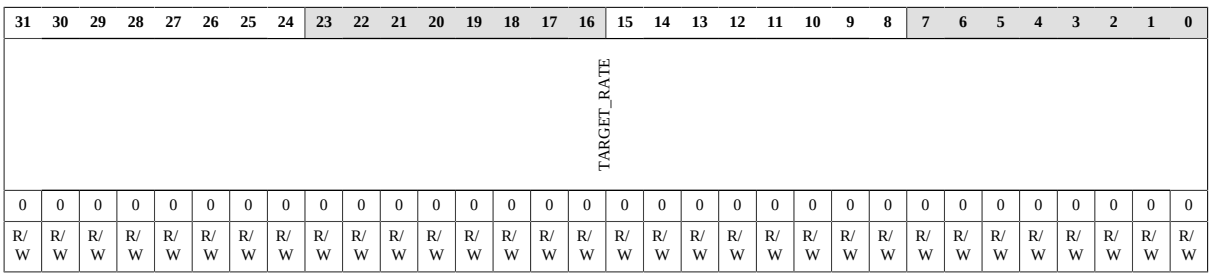


Table 1.182. CMD_ENC_RC_TARGET_RATE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TARGET_RATE	R/W	A target rate (EN_SEPARATE_BITRATE = 0) (0 ~ 700000000)	0x0

Confidential
Chips&Media Inc.

1.2.3.2.2.46. CMD_ENC_RESERVED (0x000001C4)

Table 1.183. CMD_ENC_RESERVED Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															RESERVED
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W

Table 1.184. CMD_ENC_RESERVED Field Description

Bit	Name	Type	Function	Reset Value
[31:1]	RSVD	RO	Reserved	0x0
[0]	RESERVED	R/W	Reserved	0x0

1.2.3.2.2.47. RET_ENC_MIN_FB_NUM (0x000001CC)

Table 1.185. RET_ENC_MIN_FB_NUM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_FB_NUM																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.186. RET_ENC_MIN_FB_NUM Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	MIN_FB_NUM	R/W	The minimun number of frame buffer for encoding	0x0

1.2.3.2.2.48. RET_ENC_NAL_INFO_TO_BE_ENCODED (0x000001D0)

User data flag

Table 1.187. RET_ENC_NAL_INFO_TO_BE_ENCODED Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAL_INFO_TO_BE_ENCODED																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential
StarFive Inc.

Table 1.188. RET_ENC_NAL_INFO_TO_BE_ENCODED Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	NAL_INFO_TO_BE_ENCODED	R/W	NAL unit types that will be encoded while run the next ENC_PIC command	0x0

1.2.3.2.2.49. RET_FRAME_CYCLE (0x000001D4)

A frame processing cycle

Table 1.189. RET_FRAME_CYCLE Bit Assignment

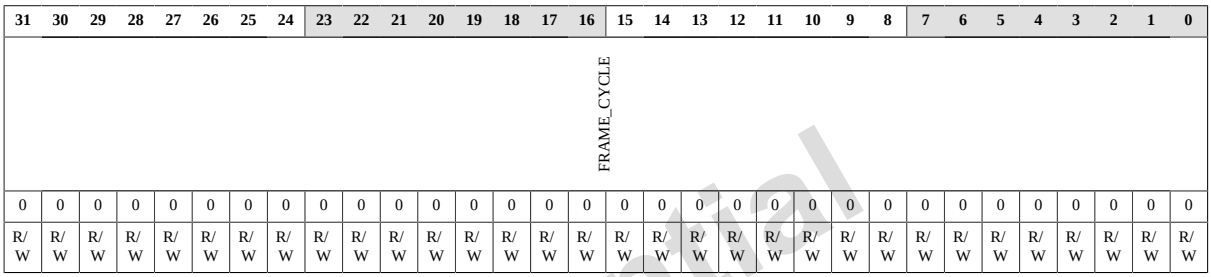


Table 1.190. RET_FRAME_CYCLE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FRAME_CYCLE	R/W	The number of processing cycle for a frame (1 ~ 16)	0x0

1.2.3.2.2.50. RET_MIN_SRC_BUF_NUM (0x000001D8)

DPB out flag

Table 1.191. RET_MIN_SRC_BUF_NUM Bit Assignment

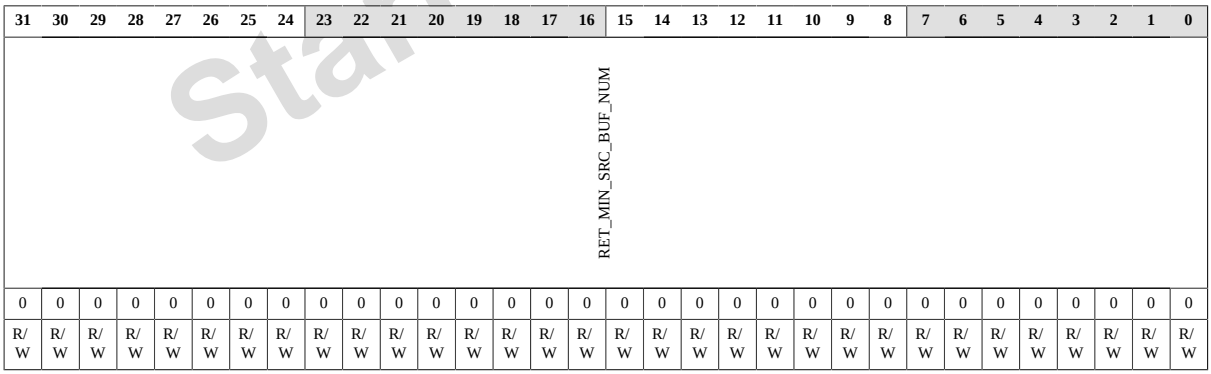


Table 1.192. RET_MIN_SRC_BUF_NUM Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RET_MIN_SRC_BUF_NUM	R/W	The minimum number of source frame buffer for encoding (1 ~ 16)	0x0

Confidential StarFive Inc.

1.2.3.2.3. SET_PARAM Command(GOP) Parameter Registers

There are command I/O registers where host processor can set arguments for the SET_PARAM command or get return values. These are only when CMD_ENC_SET_PARAM_OPTION(0x10C) is set to 1, all of which work in the GOP setting mode.

1.2.3.2.3.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.193. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.194. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			Reserved for future • 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.3.2. CORE_INDEX (0x00000104)

V-CORE Index
This is a common parameter register for almost all commands, except VPU control commands such as INIT_VPU, GET_FW_VERSION, SLEEP_VPU, and WAKEUP_VPU.

Table 1.195. CORE_INDEX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VCORE_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.196. CORE_INDEX Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	VCORE_INDEX	RW	V-CORE index - reserved for future use, so please set this as 0.	0x0

1.2.3.2.3.3. INST_INDEX_COD_STD (0x00000108)

Codec Standard and Instance Index
This is a common parameter register for almost all commands (Reserved for future use)

Table 1.197. INST_INDEX_COD_STD Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODEC_STD																INST_INDEX															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.198. INST_INDEX_COD_STD Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CODEC_STD	R/W	Codec standard to run HEVC_DEC = 0x0 HEVC_ENC = 0x1	0x0
[15:0]	INST_INDEX	R/W	Instance Index VPU can decode more than one decoding instance simultaneously. If multiple instances are running, each instance must have their own process index that is assigned by this register. For example, two instances are running simultaneously, the first instance has the process index 0, the second instance has the process index 1. Instance index shall be in the range of 0 to 65535, inclusive.	0x0

1.2.3.2.3.4. CMD_ENC_SET_PARAM_OPTION (0x0000010C)

Host should set this as one of the following while using SET_PARAM command.
Depending on this, SET_PARAM command parameter registers from 0x15C have different meanings.

Table 1.199. CMD_ENC_SET_PARAM_OPTION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														SET_PARAM_OPTION	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.200. CMD_ENC_SET_PARAM_OPTION Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	SET_PARAM_OPTION	R/W	0: It enables to set COMMON registers. 1: It enables to set CUSTOM_GOP registers. 2: It enables to set SEI registers. 3: It enables to set VUI registers.	0x0

1.2.3.2.3.5. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.201. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.202. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.3.6. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.203. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.204. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.3.7. BS_START_ADDR (0x00000120)

Bitstream buffer start address

Table 1.205. BS_START_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS_START_ADDR																												RSVD			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

Table 1.206. BS_START_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	BS_START_ADDR	R/W	Start Address of Bitstream Buffer (fixed in ring buffer mode) It should be aligned in bus width (128bit/16 byte).	0x0
[3:0]	RSVD	RO	Reserved	0x0

1.2.3.2.3.8. BS_SIZE (0x00000124)

Bitstream buffer size

Confidential StarFive Inc.

Table 1.207. BS_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS_BUF_SIZE																												RSVD			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

Table 1.208. BS_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	BS_BUF_SIZE	R/W	Size of Bistream buffer (fixed in ring buffer mode) It should be aligned in bus width (128bit/16byte).	0x0
[3:0]	RSVD	RO	Reserved	0x0

1.2.3.2.3.9. BS_PARAM (0x00000128)

Bitstream buffer parameters

Table 1.209. BS_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										BS_SLICE_WR_PTR_INTERRUPT_FLAG	BS_WRAP_AROUND_FLAG	BS_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.210. BS_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:6]	RSVD	RO	Reserved	0x0
[5]	BS_SLICE_WR_PTR_INTERRUPT_FLAG	R/W	It updates wr_ptr each slice.	0x0
[4]	BS_WRAP_AROUND_FLAG	R/W	A wrap around flag	0x0
[3:0]	BS_ENDIAN	R/W	Endianness of bitstream buffer	0x0

1.2.3.2.3.10. BS_OPTIONS (0x0000012C)

Bistream buffer option

Table 1.211. BS_OPTIONS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

RSVD																												STREAM_END	EXPLICIT_END
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.212. BS_OPTIONS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	R/W	Reserved	0x0
[1]	STREAM_END	R/W	This field makes VPU assumes Stream End when there is no stream to feed in the buffer.	0x0
[0]	EXPLICIT_END	R/W	Explicit End When this field is set to 1, VPU assumes that bitstream buffer has one NAL unit or more for decoding a signle frame - it works based on the BS_NAL_LEVEL_PUMP register on. Then it follows the tasks below. 1. VPU decodes until the end of bitstream buffer (WR_PTR) even if the last 3 bytes are all 0. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU performs what it is supposed to do with the specified task in BS_SHORTAGE_OPTION of BS_PARAM. If this flag is 0, 1. VPU decodes to the almost end of bitstream buffer (WR_PTR), but not to some bytes (less than 3). It intentionally does not comsume some a few bytes, because VPU is not sure if the last bytes are the start code of next NAL or they might not fill the offset in the CABAC. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU stops decoding and waits for more bitstream to be filled. Then host processor can do either feed bitstream more or set EXPLICIT_END to complete decoding anyhow. Bitstream_empty interrupt is asserted when EXPLITCIT_END = 0 or when bitstream buffer is near empty not real empty for seamless decoding. **Caution** Host processor can set this register any time, but cannot clear during command processing.	0x0

Confidential
StarFive Inc.

1.2.3.2.3.11. BS_RD_PTR (0x00000130)

Bistream Buffer Read Pointer

Table 1.213. BS_RD_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD_PTR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.214. BS_RD_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RD_PTR	R/W	Start address of bitstream for handling current command Host processor cannot update this register in the middle of decoding. Updating before start decoding is only allowed. VPU updates this RD_PTR as the scheme described below. • When BS_NAL_LEVEL_PUMP of BS_PARAM is 1, VPU updates RET_ADDR_BS_RD_PTR to the end address of the NAL. • When BS_NAL_LEVEL_PUMP of BS_PARAM is 0, VPU updates RET_ADDR_BS_RD_PTR as end address of the last NAL for a frame.	0x0

1.2.3.2.3.12. BS_WR_PTR (0x00000134)

Bistream Buffer Write Pointer

Table 1.215. BS_WR_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WR_PTR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.216. BS_WR_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WR_PTR	R/W	End address of bitstream for handling current command Host can update this register anytime. If a bitstream_empty interrupt is asserted, host processor should do either feed more bitstream and update WR_PTR or set EXPLICIT_END to complete decoding anyhow.	0x0

1.2.3.2.3.13. ADDR_WORK_BASE (0x00000138)

Work Buffer Base Address

Table 1.217. ADDR_WORK_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORK_BUF_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.218. ADDR_WORK_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WORK_BUF_BASE	R/W	Base address of work buffer for each instance	0x0

Confidential
Starfive Inc.

Bit	Name	Type	Function	Reset Value
			This address should be aligned in 4K boundary.	

1.2.3.2.3.14. WORK_SIZE (0x0000013C)

Work Buffer Size

Table 1.219. WORK_SIZE Bit Assignment

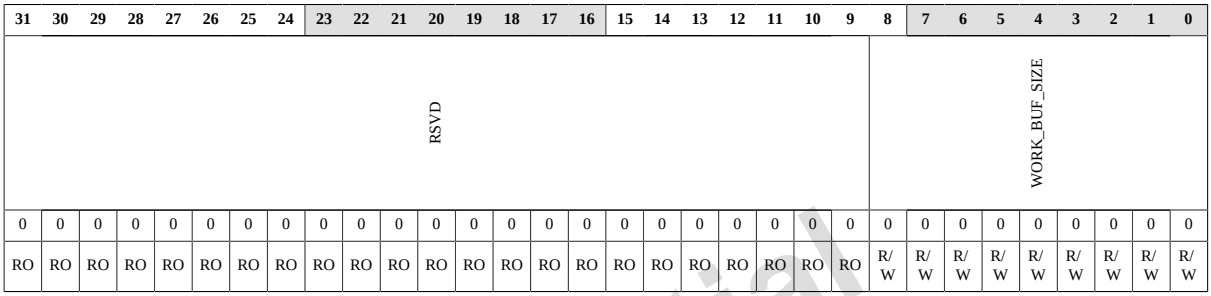


Table 1.220. WORK_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:9]	RSVD	RO	Reserved	0x0
[8:0]	WORK_BUF_SIZE	R/W	Size of work buffer (4K boundary) The size of work buffer should be larger than required minimum work buffer size. This address should be aligned in 4K boundary.	0x0

1.2.3.2.3.15. WORK_PARAM (0x00000140)

Work Buffer Parameter

Table 1.221. WORK_PARAM Bit Assignment

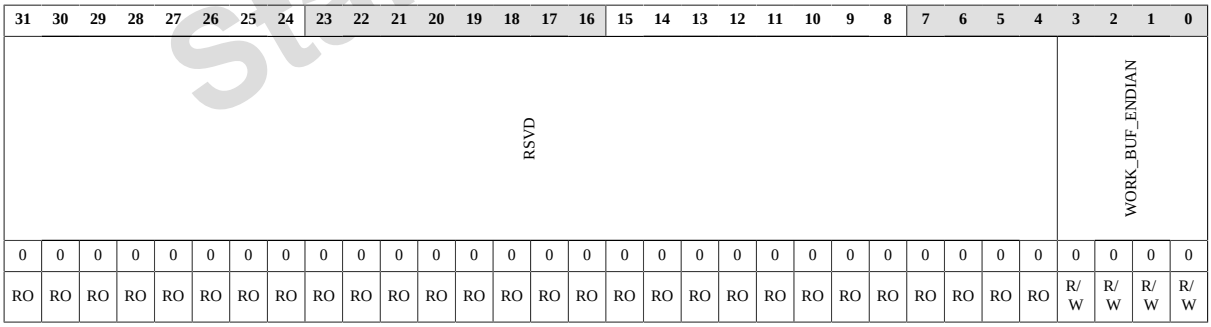


Table 1.222. WORK_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	WORK_BUF_ENDIAN	R/W	Endianness of work buffer Not implemented yet - VPU uses work buffer as internal purpose so endianness is not important.	0x0

1.2.3.2.3.16. ADDR_TEMP_BASE (0x00000144)

Temporal Buffer Base Address

Confidential StarFive Inc.

Table 1.223. ADDR_TEMP_BASE Bit Assignment

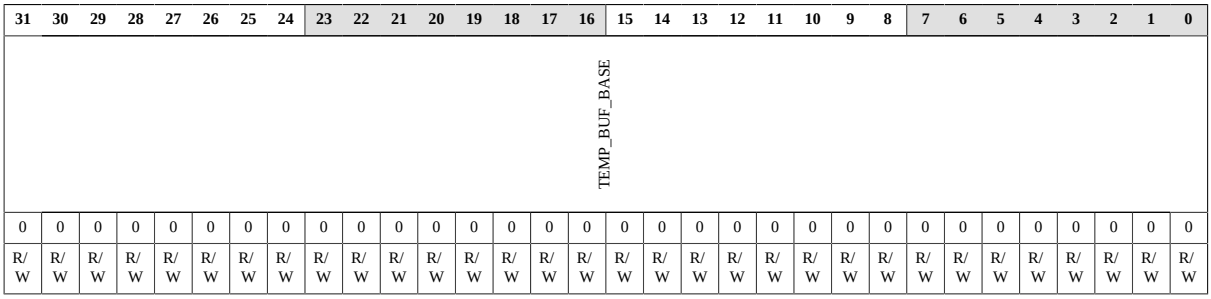


Table 1.224. ADDR_TEMP_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TEMP_BUF_BASE	R/W	Base address of temporal buffer for this frame **Note** Each V-CORE should set this field for its own temporal buffer.	0x0

1.2.3.2.3.17. TEMP_SIZE (0x00000148)

Temporal Buffer Size

Table 1.225. TEMP_SIZE Bit Assignment

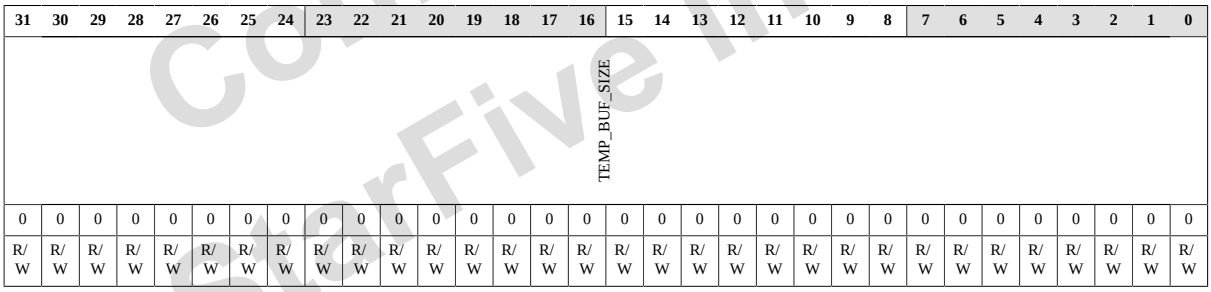


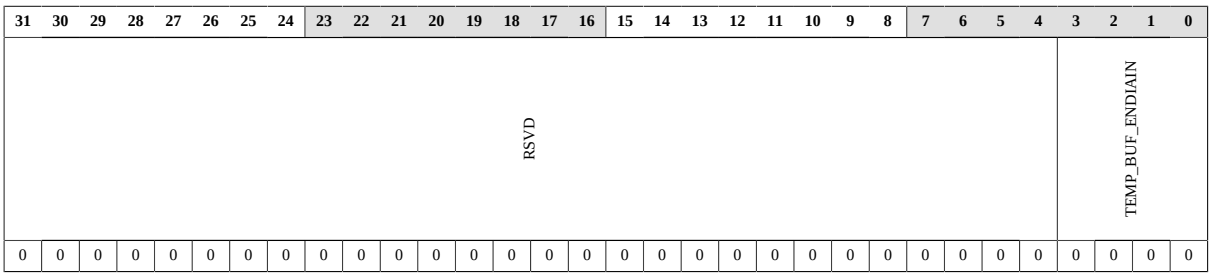
Table 1.226. TEMP_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TEMP_BUF_SIZE	R/W	Temporal buffer size for this frame	0x0

1.2.3.2.3.18. TEMP_PARAM (0x0000014C)

Temporal Buffer Parameter

Table 1.227. TEMP_PARAM Bit Assignment



Confidential StarFive Inc.

RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W

Table 1.228. TEMP_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	TEMP_BUF_ENDIAIN	R/W	Endianness of temporal buffer **Note** Each V-CORE should set this register for its own temporal buffer.	0x0

1.2.3.2.3.19. ADDR_SEC_AXI (0x00000150)

Secondary AXI Base Address
It is valid only when USE_SEC_AXI is not 0.

Table 1.229. ADDR_SEC_AXI Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_AXI_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.230. ADDR_SEC_AXI Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SEC_AXI_BASE	R/W	The base address of secondary AXI memory	0x0

1.2.3.2.3.20. SEC_AXI_SIZE (0x00000154)

Seconary AXI Memory Size
It is valid only when USE_SEC_AXI is not 0.

Table 1.231. SEC_AXI_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_AXI_MEM_SIZE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.232. SEC_AXI_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SEC_AXI_MEM_SIZE	R/W	The size of secondary AXI memory	0x0

Confidential Starfive Inc.

1.2.3.2.3.21. USE_SEC_AXI (0x00000158)

Secondary AXI Usage option

Table 1.233. USE_SEC_AXI Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																SEC_AXI_LF_ROW	RSVD			SEC_AXI_RDO_ENABLE	RSVD										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.234. USE_SEC_AXI Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15]	SEC_AXI_LF_ROW	R/W	Use 2nd AXI temp buffer for Read channel of LF row buffer.	0x0
[14:12]	RSVD	RO	Reserved	0x0
[11]	SEC_AXI_RDO_ENABLE	R/W	Use 2nd AXI temp buffer for Read channel of RDO row buffer.	0x0
[10:0]	RSVD	RO	Reserved	0x0

1.2.3.2.3.22. CMD_ENC_SET_CUSTOM_GOP_ENABLE (0x0000015C)

Custom GOP setting

Table 1.235. CMD_ENC_SET_CUSTOM_GOP_ENABLE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														CUSTOM_GOP_PIC_LAMBDA_7_EN	CUSTOM_GOP_PIC_LAMBDA_6_EN	CUSTOM_GOP_PIC_LAMBDA_5_EN	CUSTOM_GOP_PIC_LAMBDA_4_EN	CUSTOM_GOP_PIC_LAMBDA_3_EN	CUSTOM_GOP_PIC_LAMBDA_2_EN	CUSTOM_GOP_PIC_LAMBDA_1_EN	CUSTOM_GOP_PIC_LAMBDA_0_EN	RESERVED	CUSTOM_GOP_PIC_PARAM_7_EN	CUSTOM_GOP_PIC_PARAM_6_EN	CUSTOM_GOP_PIC_PARAM_5_EN	CUSTOM_GOP_PIC_PARAM_4_EN	CUSTOM_GOP_PIC_PARAM_3_EN	CUSTOM_GOP_PIC_PARAM_2_EN	CUSTOM_GOP_PIC_PARAM_1_EN	CUSTOM_GOP_PIC_PARAM_0_EN	CUSTOM_GOP_PARAM_EN
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.236. CMD_ENC_SET_CUSTOM_GOP_ENABLE Field Description

Bit	Name	Type	Function	Reset Value
[31:18]	RSVD	RO	Reserved	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[17]	CUSTOM_GOP_PIC_LAMBDA_7_EN	R/W	It enables to set CUSTOM_GOP_PIC_LAMBDA_7.	0x0
[16]	CUSTOM_GOP_PIC_LAMBDA_6_EN	R/W	It enables to set CUSTOM_GOP_PIC_LAMBDA_6.	0x0
[15]	CUSTOM_GOP_PIC_LAMBDA_5_EN	R/W	It enables to set CUSTOM_GOP_PIC_LAMBDA_5.	0x0
[14]	CUSTOM_GOP_PIC_LAMBDA_4_EN	R/W	It enables to set CUSTOM_GOP_PIC_LAMBDA_4.	0x0
[13]	CUSTOM_GOP_PIC_LAMBDA_3_EN	R/W	It enables to set CUSTOM_GOP_PIC_LAMBDA_3.	0x0
[12]	CUSTOM_GOP_PIC_LAMBDA_2_EN	R/W	It enables to set CUSTOM_GOP_PIC_LAMBDA_2.	0x0
[11]	CUSTOM_GOP_PIC_LAMBDA_1_EN	R/W	It enables to set CUSTOM_GOP_PIC_LAMBDA_1.	0x0
[10]	CUSTOM_GOP_PIC_LAMBDA_0_EN	R/W	It enables to set CUSTOM_GOP_PIC_LAMBDA_0.	0x0
[9]	RESERVED	R/W	Reserved	0x0
[8]	CUSTOM_GOP_PIC_PARAM_7_EN	R/W	It enables to set CUSTOM_GOP_PARAM_7.	0x0
[7]	CUSTOM_GOP_PIC_PARAM_6_EN	R/W	It enables to set CUSTOM_GOP_PARAM_6.	0x0
[6]	CUSTOM_GOP_PIC_PARAM_5_EN	R/W	It enables to set CUSTOM_GOP_PARAM_5.	0x0
[5]	CUSTOM_GOP_PIC_PARAM_4_EN	R/W	It enables to set CUSTOM_GOP_PARAM_4.	0x0
[4]	CUSTOM_GOP_PIC_PARAM_3_EN	R/W	It enables to set CUSTOM_GOP_PARAM_3.	0x0
[3]	CUSTOM_GOP_PIC_PARAM_2_EN	R/W	It enables to set CUSTOM_GOP_PARAM_2.	0x0
[2]	CUSTOM_GOP_PIC_PARAM_1_EN	R/W	It enables to set CUSTOM_GOP_PARAM_1.	0x0
[1]	CUSTOM_GOP_PIC_PARAM_0_EN	R/W	It enables to set CUSTOM_GOP_PARAM_0.	0x0
[0]	CUSTOM_GOP_PARAM_EN	R/W	It enables to set CUSTOM_GOP_PARAM.	0x0

1.2.3.2.3.23. CMD_ENC_CUSTOM_GOP_PARAM (0x00000160)

Table 1.237. CMD_ENC_CUSTOM_GOP_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																											DERIVE_LAMBDA_WEIGHT	CUSTOM_GOP_SIZE				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.238. CMD_ENC_CUSTOM_GOP_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:5]	RSVD	RO	Reserved	0x0
[4]	DERIVE_LAMBDA_WEIGHT	R/W	It derives a lambda_weight internally instead of using a lambda weight specified.	0x0
[3:0]	CUSTOM_GOP_SIZE	R/W	The size of cyclic GOP (1 ~ 8) This should be the same as the number of CUSTOM_GOP_PIC_PARAM set by host	0x0

1.2.3.2.3.24. CMD_ENC_CUSTOM_GOP_PIC_PARAM_0 (0x00000164)

Table 1.239. CMD_ENC_CUSTOM_GOP_PIC_PARAM_0 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TEMPORAL_ID				REF_POC_L1					REF_POC_L0					RESERVED		PIC_QP						POC_OFFSET				PIC_TYPE	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.240. CMD_ENC_CUSTOM_GOP_PIC_PARAM_0 Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RESERVED	R/W	Reserved	0x0
[27:24]	TEMPORAL_ID	R/W	A temporal ID of 0th picture in the custom GOP	0x0
[23:19]	REF_POC_L1	R/W	A poc offset of reference L1 of 0th picture in the custom GOP	0x0
[18:14]	REF_POC_L0	R/W	A poc offset of reference L0 of 0th picture in the custom GOP	0x0
[13:12]	RESERVED	R/W	Reserved	0x0
[11:6]	PIC_QP	R/W	A quantization parameter of 0th picture in the custom GOP	0x0
[5:2]	POC_OFFSET	R/W	A poc offset of 0th picture in the custom GOP	0x0
[1:0]	PIC_TYPE	R/W	A picture type of 0th picture in the custom GOP 0 : I picture 1 : P picture 2 : B picture	0x0

1.2.3.2.3.25. CMD_ENC_CUSTOM_GOP_PIC_PARAM_1 (0x00000168)

Table 1.241. CMD_ENC_CUSTOM_GOP_PIC_PARAM_1 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TEMPORAL_ID				REF_POC_L1					REF_POC_L0					RESERVED		PIC_QP						POC_OFFSET				PIC_TYPE	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.242. CMD_ENC_CUSTOM_GOP_PIC_PARAM_1 Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RESERVED	R/W	Reserved	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[27:24]	TEMPORAL_ID	R/W	A temporal ID of 1st picture in the custom GOP	0x0
[23:19]	REF_POC_L1	R/W	A poc offset of reference L1 of 1st picture in the custom GOP	0x0
[18:14]	REF_POC_L0	R/W	A poc offset of reference L0 of 1st picture in the custom GOP	0x0
[13:12]	RESERVED	R/W	Reserved	0x0
[11:6]	PIC_QP	R/W	A quantization parameter of 1st picture in the custom GOP	0x0
[5:2]	POC_OFFSET	R/W	A poc offset of 1st picture in the custom GOP	0x0
[1:0]	PIC_TYPE	R/W	A picture type of 1st picture in the custom GOP 0 : I picture 1 : P picture 2 : B picture	0x0

1.2.3.2.3.26. CMD_ENC_CUSTOM_GOP_PIC_PARAM_2 (0x0000016C)

Table 1.243. CMD_ENC_CUSTOM_GOP_PIC_PARAM_2 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TEMPORAL_ID				REF_POC_L1					REF_POC_L0			RESERVED				PIC_QP						POC_OFFSET				PIC_TYPE	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.244. CMD_ENC_CUSTOM_GOP_PIC_PARAM_2 Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RESERVED	R/W	Reserved	0x0
[27:24]	TEMPORAL_ID	R/W	A temporal ID of 2nd picture in the custom GOP	0x0
[23:19]	REF_POC_L1	R/W	A poc offset of reference L1 of 2nd picture in the custom GOP	0x0
[18:14]	REF_POC_L0	R/W	A poc offset of reference L0 of 2nd picture in the custom GOP	0x0
[13:12]	RESERVED	R/W	Reserved	0x0
[11:6]	PIC_QP	R/W	A quantization parameter of 2nd picture in the custom GOP	0x0
[5:2]	POC_OFFSET	R/W	A poc offset of 2nd picture in the custom GOP	0x0
[1:0]	PIC_TYPE	R/W	A picture type of 2nd picture in the custom GOP 0 : I picture 1 : P picture 2 : B picture	0x0

1.2.3.2.3.27. CMD_ENC_CUSTOM_GOP_PIC_PARAM_3 (0x00000170)

Table 1.245. CMD_ENC_CUSTOM_GOP_PIC_PARAM_3 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TEMPORAL_ID				REF_POC_L1					REF_POC_L0			RESERVED				PIC_QP						POC_OFFSET				PIC_TYPE	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential Starfive Inc.

Table 1.246. CMD_ENC_CUSTOM_GOP_PIC_PARAM_3 Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RESERVED	R/W	Reserved	0x0
[27:24]	TEMPORAL_ID	R/W	A temporal ID of 3rd picture in the custom GOP	0x0
[23:19]	REF_POC_L1	R/W	A poc offset of reference L1 of 3rd picture in the custom GOP	0x0
[18:14]	REF_POC_L0	R/W	A poc offset of reference L0 of 3rd picture in the custom GOP	0x0
[13:12]	RESERVED	R/W	Reserved	0x0
[11:6]	PIC_QP	R/W	A quantization parameter of 3rd picture in the custom GOP	0x0
[5:2]	POC_OFFSET	R/W	A poc offset of 3rd picture in the custom GOP	0x0
[1:0]	PIC_TYPE	R/W	A picture type of 3rd picture in the custom GOP 0 : I picture 1 : P picture 2 : B picture	0x0

1.2.3.2.3.28. CMD_ENC_CUSTOM_GOP_PIC_PARAM_4 (0x00000174)

Table 1.247. CMD_ENC_CUSTOM_GOP_PIC_PARAM_4 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TEMPORAL_ID				REF_POC_L1					REF_POC_L0					RESERVED		PIC_QP						POC_OFFSET				PIC_TYPE	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.248. CMD_ENC_CUSTOM_GOP_PIC_PARAM_4 Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RESERVED	R/W	Reserved	0x0
[27:24]	TEMPORAL_ID	R/W	A temporal ID of 4th picture in the custom GOP	0x0
[23:19]	REF_POC_L1	R/W	A poc offset of reference L1 of 4th picture in the custom GOP	0x0
[18:14]	REF_POC_L0	R/W	A poc offset of reference L0 of 4th picture in the custom GOP	0x0
[13:12]	RESERVED	R/W	Reserved	0x0
[11:6]	PIC_QP	R/W	A quantization parameter of 4th picture in the custom GOP	0x0
[5:2]	POC_OFFSET	R/W	A poc offset of 4th picture in the custom GOP	0x0
[1:0]	PIC_TYPE	R/W	A picture type of 4th picture in the custom GOP 0 : I picture 1 : P picture 2 : B picture	0x0

1.2.3.2.3.29. CMD_ENC_CUSTOM_GOP_PIC_PARAM_5 (0x00000178)

Table 1.249. CMD_ENC_CUSTOM_GOP_PIC_PARAM_5 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TEMPORAL_ID				REF_POC_L1					REF_POC_L0					RESERVED		PIC_QP						POC_OFFSET				PIC_TYPE	

Confidential
StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.250. CMD_ENC_CUSTOM_GOP_PIC_PARAM_5 Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RESERVED	R/W	Reserved	0x0
[27:24]	TEMPORAL_ID	R/W	A temporal ID of 5th picture in the custom GOP	0x0
[23:19]	REF_POC_L1	R/W	A poc offset of reference L1 of 5th picture in the custom GOP	0x0
[18:14]	REF_POC_L0	R/W	A poc offset of reference L0 of 5th picture in the custom GOP	0x0
[13:12]	RESERVED	R/W	Reserved	0x0
[11:6]	PIC_QP	R/W	A quantization parameter of 5th picture in the custom GOP	0x0
[5:2]	POC_OFFSET	R/W	A poc offset of 5th picture in the custom GOP	0x0
[1:0]	PIC_TYPE	R/W	A picture type of 5th picture in the custom GOP 0 : I picture 1 : P picture 2 : B picture	0x0

1.2.3.2.3.30. CMD_ENC_CUSTOM_GOP_PIC_PARAM_6 (0x0000017C)

Table 1.251. CMD_ENC_CUSTOM_GOP_PIC_PARAM_6 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TEMPORAL_ID				REF_POC_L1					REF_POC_L0			RESERVED				PIC_QP						POC_OFFSET				PIC_TYPE	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.252. CMD_ENC_CUSTOM_GOP_PIC_PARAM_6 Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RESERVED	R/W	Reserved	0x0
[27:24]	TEMPORAL_ID	R/W	A temporal ID of 6th picture in the custom GOP	0x0
[23:19]	REF_POC_L1	R/W	A poc offset of reference L1 of 6th picture in the custom GOP	0x0
[18:14]	REF_POC_L0	R/W	A poc offset of reference L0 of 6th picture in the custom GOP	0x0
[13:12]	RESERVED	R/W	Reserved	0x0
[11:6]	PIC_QP	R/W	A quantization parameter of 6th picture in the custom GOP	0x0
[5:2]	POC_OFFSET	R/W	A poc offset of 6th picture in the custom GOP	0x0
[1:0]	PIC_TYPE	R/W	A picture type of 6th picture in the custom GOP 0 : I picture 1 : P picture 2 : B picture	0x0

1.2.3.2.3.31. CMD_ENC_CUSTOM_GOP_PIC_PARAM_7 (0x00000180)

Table 1.253. CMD_ENC_CUSTOM_GOP_PIC_PARAM_7 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential
StarFive Inc.

RESERVED				TEMPORAL_ID				REF_POC_L1					REF_POC_L0					RESERVED		PIC_QP						POC_OFFSET				PIC_TYPE	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.254. CMD_ENC_CUSTOM_GOP_PIC_PARAM_7 Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RESERVED	R/W	Reserved	0x0
[27:24]	TEMPORAL_ID	R/W	A temporal ID of 7th picture in the custom GOP	0x0
[23:19]	REF_POC_L1	R/W	A poc offset of reference L1 of 7th picture in the custom GOP	0x0
[18:14]	REF_POC_L0	R/W	A poc offset of reference L0 of 7th picture in the custom GOP	0x0
[13:12]	RESERVED	R/W	Reserved	0x0
[11:6]	PIC_QP	R/W	A quantization parameter of 7th picture in the custom GOP	0x0
[5:2]	POC_OFFSET	R/W	A poc offset of 7th picture in the custom GOP	0x0
[1:0]	PIC_TYPE	R/W	A picture type of 7th picture in the custom GOP 0 : I picture 1 : P picture 2 : B picture	0x0

1.2.3.2.3.32. CMD_ENC_CUSTOM_GOP_RESERVED (0x00000184)

Table 1.255. CMD_ENC_CUSTOM_GOP_RESERVED Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.256. CMD_ENC_CUSTOM_GOP_RESERVED Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RESERVED	R/W	Reserved	0x0

1.2.3.2.3.33. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_0 (0x00000188)

Table 1.257. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_0 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAMBDA_WEIGHT																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential
StarFive Inc.

Table 1.258. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_0 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LAMBDA_WEIGHT	R/W	A lamda weight of 0th picture in the custom GOP	0x0

1.2.3.2.3.34. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_1 (0x0000018C)

Table 1.259. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_1 Bit Assignment

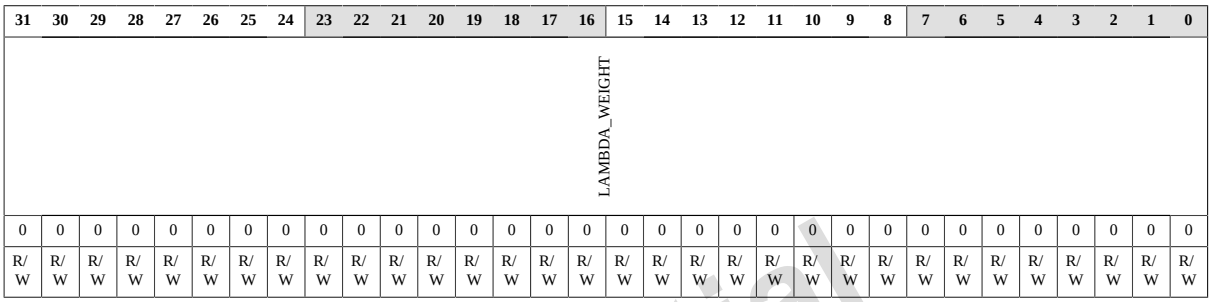


Table 1.260. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_1 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LAMBDA_WEIGHT	R/W	A lamda weight of 1st picture in the custom GOP	0x0

1.2.3.2.3.35. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_2 (0x00000190)

Table 1.261. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_2 Bit Assignment

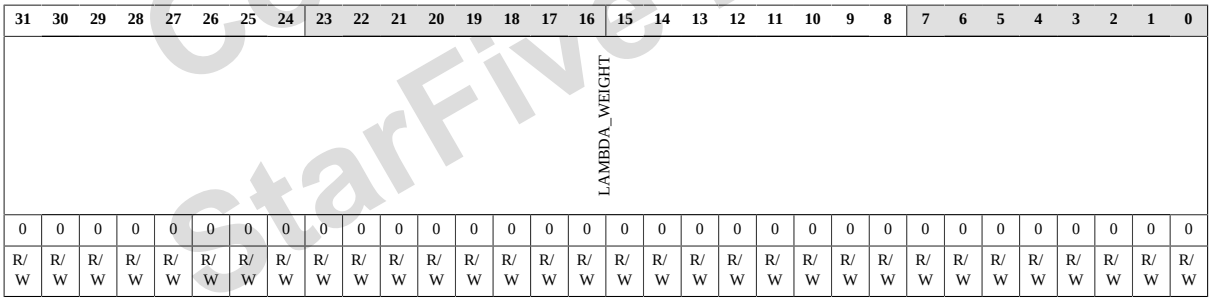
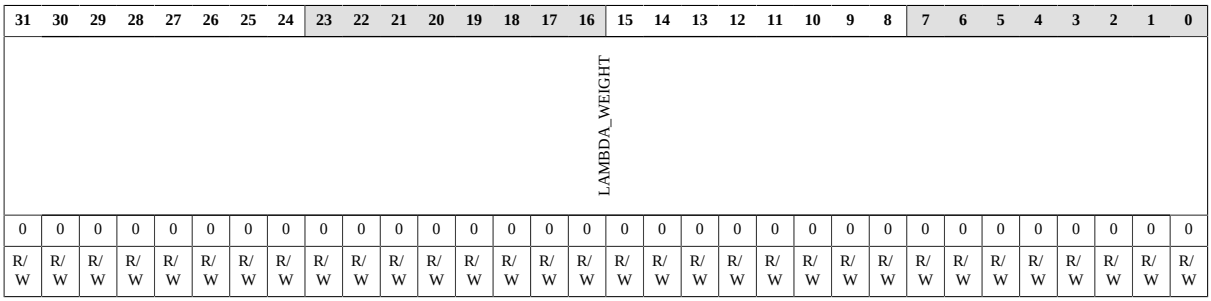


Table 1.262. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_2 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LAMBDA_WEIGHT	R/W	A lamda weight of 2nd picture in the custom GOP	0x0

1.2.3.2.3.36. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_3 (0x00000194)

Table 1.263. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_3 Bit Assignment



Confidential StarFive Inc.

Table 1.264. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_3 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LAMBDA_WEIGHT	R/W	A lamda weight of 3rd picture in the custom GOP	0x0

1.2.3.2.3.37. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_4 (0x00000198)

Table 1.265. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_4 Bit Assignment

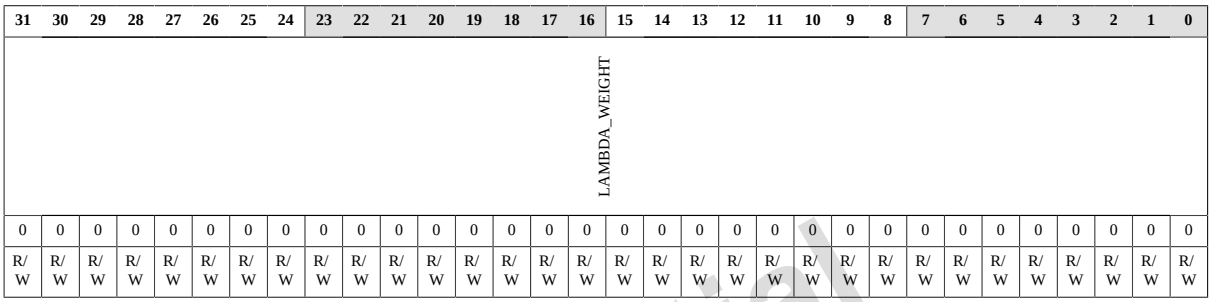


Table 1.266. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_4 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LAMBDA_WEIGHT	R/W	A lamda weight of 4th picture in the custom GOP	0x0

1.2.3.2.3.38. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_5 (0x0000019C)

Table 1.267. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_5 Bit Assignment

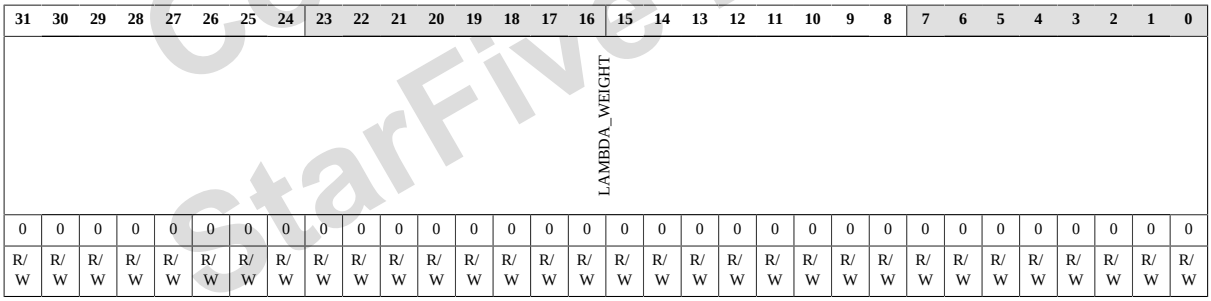
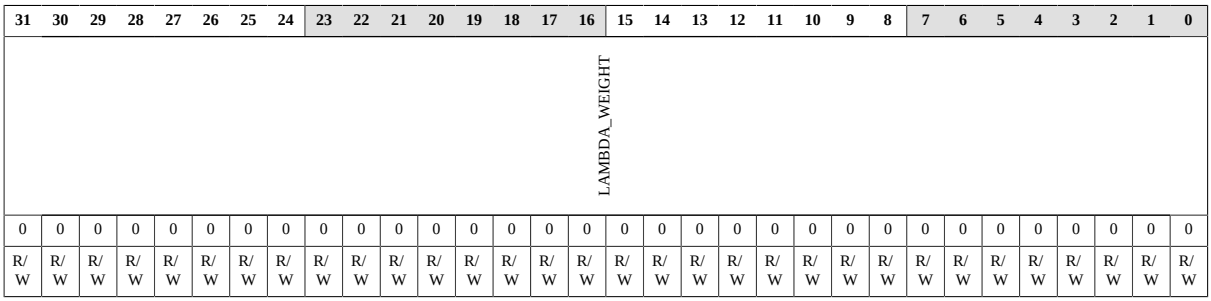


Table 1.268. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_5 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LAMBDA_WEIGHT	R/W	A lamda weight of 5th picture in the custom GOP	0x0

1.2.3.2.3.39. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_6 (0x000001A0)

Table 1.269. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_6 Bit Assignment



Confidential StarFive Inc.

Table 1.270. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_6 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LAMBDA_WEIGHT	R/W	A lamda weight of 6th picture in the custom GOP	0x0

1.2.3.2.3.40. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_7 (0x000001A4)

Table 1.271. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_7 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAMBDA_WEIGHT																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.272. CMD_ENC_CUSTOM_GOP_PIC_LAMBDA_7 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LAMBDA_WEIGHT	R/W	A lamda weight of 7th picture in the custom GOP	0x0

1.2.3.2.3.41. RET_ENC_MIN_FB_NUM (0x000001CC)

Table 1.273. RET_ENC_MIN_FB_NUM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_FB_NUM																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.274. RET_ENC_MIN_FB_NUM Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	MIN_FB_NUM	R/W	The minimun number of frame buffer for encoding	0x0

1.2.3.2.3.42. RET_FRAME_CYCLE (0x000001D4)

A frame processing cycle

Table 1.275. RET_FRAME_CYCLE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRAME_CYCLE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.276. RET_FRAME_CYCLE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FRAME_CYCLE	R/W	The Number of processing cycle for a frame	0x0

1.2.3.2.3.43. RET_MIN_SRC_BUF_NUM (0x000001D8)

A DPB out flag

Table 1.277. RET_MIN_SRC_BUF_NUM Bit Assignment

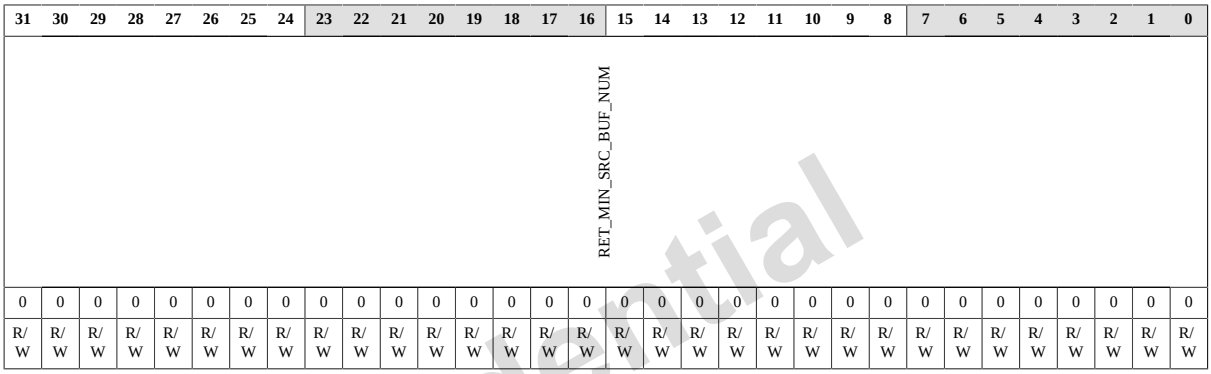


Table 1.278. RET_MIN_SRC_BUF_NUM Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RET_MIN_SRC_BUF_NUM	R/W	The minimum number of source frame buffer for encoding	0x0

Confidential StarFive Tech

1.2.3.2.4. SET_PARAM Command(VUI) Parameter Registers

There are command I/O registers where host processor can set arguments for the SET_PARAM command or get return values. These are only when CMD_ENC_SET_PARAM_OPTION(0x10C) is set to 3, all of which work in the VUI setting mode.

1.2.3.2.4.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.279. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.280. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			Reserved for future • 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.4.2. CORE_INDEX (0x00000104)

V-CORE Index
This is a common parameter register for almost all commands, except VPU control commands such as INIT_VPU, GET_FW_VERSION, SLEEP_VPU, and WAKEUP_VPU.

Table 1.281. CORE_INDEX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VCORE_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.282. CORE_INDEX Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	VCORE_INDEX	RW	V-CORE index - reserved for future use, so please set this as 0.	0x0

1.2.3.2.4.3. INST_INDEX_COD_STD (0x00000108)

Codec Standard and Instance Index
This is a common parameter register for almost all commands (Reserved for future use)

Table 1.283. INST_INDEX_COD_STD Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODEC_STD																INST_INDEX															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.284. INST_INDEX_COD_STD Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CODEC_STD	R/W	Codec standard to run HEVC_DEC = 0x0 HEVC_ENC = 0x1	0x0
[15:0]	INST_INDEX	R/W	Instance Index VPU can decode more than one decoding instance simultaneously. If multiple instances are running, each instance must have their own process index that is assigned by this register. For example, two instances are running simultaneously, the first instance has the process index 0, the second instance has the process index 1. Instance index shall be in the range of 0 to 65535, inclusive.	0x0

1.2.3.2.4.4. CMD_ENC_SET_PARAM_OPTION (0x0000010C)

Host should set this as one of the following while using SET_PARAM command.
Depending on this, SET_PARAM command parameter registers from 0x15C have different meanings.

Table 1.285. CMD_ENC_SET_PARAM_OPTION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														SET_PARAM_OPTION	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.286. CMD_ENC_SET_PARAM_OPTION Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	SET_PARAM_OPTION	R/W	0: It enables to set COMMON registers. 1: It enables to set CUSTOM_GOP registers. 2: It enables to set SEI registers. 3: It enables to set VUI registers.	0x0

1.2.3.2.4.5. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.287. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.288. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.4.6. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.289. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.290. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.4.7. BS_START_ADDR (0x00000120)

Bitstream buffer start address

Table 1.291. BS_START_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS_START_ADDR																												RSVD			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

Table 1.292. BS_START_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	BS_START_ADDR	R/W	Start Address of Bitstream Buffer (fixed in ring buffer mode) It should be aligned in bus width (128bit/16 byte).	0x0
[3:0]	RSVD	RO	Reserved	0x0

Confidential Starfive Inc.

1.2.3.2.4.8. BS_SIZE (0x00000124)

Bitstream buffer size

Table 1.293. BS_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS_BUF_SIZE																												RSVD			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

Table 1.294. BS_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	BS_BUF_SIZE	R/W	Size of Bistream buffer (fixed in ring buffer mode) It should be aligned in bus width (128bit/16byte).	0x0
[3:0]	RSVD	RO	Reserved	0x0

1.2.3.2.4.9. BS_PARAM (0x00000128)

Bitstream buffer parameters

Table 1.295. BS_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										BS_SLICE_WR_PTR_INTERRUPT_FLAG	BS_WRAP_AROUND_FLAG	BS_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.296. BS_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:6]	RSVD	RO	Reserved	0x0
[5]	BS_SLICE_WR_PTR_INTERRUPT_FLAG	R/W	It updates wr_ptr each slice.	0x0
[4]	BS_WRAP_AROUND_FLAG	R/W	A wrap around flag	0x0
[3:0]	BS_ENDIAN	R/W	Endianness of bitstream buffer	0x0

1.2.3.2.4.10. BS_OPTIONS (0x0000012C)

Bistream buffer option

Confidential
StarFive Inc.

Table 1.297. BS_OPTIONS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														STREAM_END	EXPLICIT_END
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.298. BS_OPTIONS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	R/W	Reserved	0x0
[1]	STREAM_END	R/W	This field makes VPU assumes Stream End when there is no stream to feed in the buffer.	0x0
[0]	EXPLICIT_END	R/W	Explicit End When this field is set to 1, VPU assumes that bitstream buffer has one NAL unit or more for decoding a signle frame - it works based on the BS_NAL_LEVEL_PUMP register on. Then it follows the tasks below. 1. VPU decodes until the end of bitstream buffer (WR_PTR) even if the last 3 bytes are all 0. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU performs what it is supposed to do with the specified task in BS_SHORTAGE_OPTION of BS_PARAM. If this flag is 0, 1. VPU decodes to the almost end of bitstream buffer (WR_PTR), but not to some bytes (less than 3). It intentionally does not consume some a few bytes, because VPU is not sure if the last bytes are the start code of next NAL or they might not fill the offset in the CABAC. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU stops decoding and waits for more bitstream to be filled. Then host processor can do either feed bitstream more or set EXPLICIT_END to complete decoding anyhow. Bitstream_empty interrupt is asserted when EXPLITCT_END = 0 or when bitstream buffer is near empty (not real empty) for seamless decoding. **Caution** Host processor can set this register any time, but cannot clear during command processing.	0x0

1.2.3.2.4.11. BS_RD_PTR (0x00000130)

Bistream Buffer Read Pointer

Table 1.299. BS_RD_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD_PTR																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.300. BS_RD_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RD_PTR	R/W	Start address of bitstream for handling current command Host processor cannot update this register in the middle of decoding. Updating before start decoding is only allowed. VPU updates this RD_PTR as the scheme described below. • When BS_NAL_LEVEL_PUMP of BS_PARAM is 1, VPU updates RET_ADDR_BS_RD_PTR to the end address of the NAL. • When BS_NAL_LEVEL_PUMP of BS_PARAM is 0, VPU updates RET_ADDR_BS_RD_PTR as end address of the last NAL for a frame.	0x0

1.2.3.2.4.12. BS_WR_PTR (0x00000134)

Bistream Buffer Write Pointer

Table 1.301. BS_WR_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WR_PTR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.302. BS_WR_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WR_PTR	R/W	End address of bitstream for handling current command Host can update this register anytime. If a bitstream_empty interrupt is asserted, host processor should do either feed more bitstream and update WR_PTR or set EXPLICIT_END to complete decoding anyhow.	0x0

1.2.3.2.4.13. ADDR_WORK_BASE (0x00000138)

Work Buffer Base Address

Table 1.303. ADDR_WORK_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORK_BUF_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.304. ADDR_WORK_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WORK_BUF_BASE	R/W	Base address of work buffer for each instance	0x0

Confidential
Starfive Inc.

Bit	Name	Type	Function	Reset Value
			This address should be aligned in 4K boundary.	

1.2.3.2.4.14. WORK_SIZE (0x0000013C)

Work Buffer Size

Table 1.305. WORK_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								WORK_BUF_SIZE							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.306. WORK_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:9]	RSVD	RO	Reserved	0x0
[8:0]	WORK_BUF_SIZE	R/W	Size of work buffer (4K boundary) The size of work buffer should be larger than required minimum work buffer size. This address should be aligned in 4K boundary.	0x0

1.2.3.2.4.15. WORK_PARAM (0x00000140)

Work Buffer Parameter

Table 1.307. WORK_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												WORK_BUF_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W

Table 1.308. WORK_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	WORK_BUF_ENDIAN	R/W	Endianness of work buffer Not implemented yet - VPU uses work buffer as internal purpose so endianness is not important.	0x0

1.2.3.2.4.16. ADDR_TEMP_BASE (0x00000144)

Temporal Buffer Base Address

Confidential StarFive Inc.

Table 1.309. ADDR_TEMP_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEMP_BUF_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.310. ADDR_TEMP_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TEMP_BUF_BASE	R/W	Base address of temporal buffer for this frame **Note** Each V-CORE should set this field for its own temporal buffer.	0x0

1.2.3.2.4.17. TEMP_SIZE (0x00000148)

Temporal Buffer Size

Table 1.311. TEMP_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEMP_BUF_SIZE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.312. TEMP_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TEMP_BUF_SIZE	R/W	Temporal buffer size for this frame	0x0

1.2.3.2.4.18. TEMP_PARAM (0x0000014C)

Temporal Buffer Parameter

Table 1.313. TEMP_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												TEMP_BUF_ENDIAIN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.314. TEMP_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	TEMP_BUF_ENDIAIN	R/W	Endianness of temporal buffer **Note** Each V-CORE should set this register for its own temporal buffer.	0x0

1.2.3.2.4.19. ADDR_SEC_AXI (0x00000150)

Secondary AXI Base Address
It is valid only when USE_SEC_AXI is not 0.

Table 1.315. ADDR_SEC_AXI Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_AXI_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.316. ADDR_SEC_AXI Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SEC_AXI_BASE	R/W	The base address of secondary AXI memory	0x0

1.2.3.2.4.20. SEC_AXI_SIZE (0x00000154)

Seconary AXI Memory Size
It is valid only when USE_SEC_AXI is not 0.

Table 1.317. SEC_AXI_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_AXI_MEM_SIZE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.318. SEC_AXI_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SEC_AXI_MEM_SIZE	R/W	The size of secondary AXI memory	0x0

1.2.3.2.4.21. USE_SEC_AXI (0x00000158)

Secondary AXI Usage option

Confidential StarFive Inc.

Table 1.319. USE_SEC_AXI Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																SEC_AXI_LF_ROW	RSVD			SEC_AXI_RDO_ENABLE	RSVD												
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.320. USE_SEC_AXI Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15]	SEC_AXI_LF_ROW	R/W	Use 2nd AXI temp buffer for Read channel of LF row buffer.	0x0
[14:12]	RSVD	RO	Reserved	0x0
[11]	SEC_AXI_RDO_ENABLE	R/W	Use 2nd AXI temp buffer for Read channel of RDO row buffer.	0x0
[10:0]	RSVD	RO	Reserved	0x0

1.2.3.2.4.22. CMD_ENC_VUI_PARAM_FLAGS (0x0000015C)

Table 1.321. CMD_ENC_VUI_PARAM_FLAGS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						BITSTREAM_RESTRICTION_FLAG	DEFAULT_DISPLAY_WINDOW_FLAG	CHROMA_LOC_INFO_PRESENT_FLAG	COLOUR_DESCRIPTION_PRESENT_FLAG	VIDEO_SIGNAL_TYPE_PRESENT_FLAG	OVERSCAN_INFO_PRESENT_FLAG	ASPECT_RATIO_INFO_PRESENT_FLAG	RSVD	RSVD	NEUTRAL_CHROMA_INDICATION_FLAG
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.322. CMD_ENC_VUI_PARAM_FLAGS Field Description

Bit	Name	Type	Function	Reset Value
[31:10]	RSVD	R/W	Reserved	0x0
[9]	BITSTREAM_RESTRICTION_FLAG	R/W	A flag whether to insert bitstream_restriction_flag syntax of VUI parameters	0x0
[8]	DEFAULT_DISPLAY_WINDOW_FLAG	R/W	A flag whether to insert default_display_window_flag syntax of VUI parameters	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[7]	CHROMA_LOC_INFO_PRESENT_FLAG	R/W	A flag whether to insert chroma_loc_info_present_flag syntax of VUI parameters	0x0
[6]	COLOUR_DESCRIPTION_PRESENT_FLAG	R/W	A flag whether to insert colour_description_present_flag syntax of VUI parameters	0x0
[5]	VIDEO_SIGNAL_TYPE_PRESENT_FLAG	R/W	A flag whether to insert video_signal_type_present_flag syntax of VUI parameters	0x0
[4]	OVERSCAN_INFO_PRESENT_FLAG	R/W	A flag whether to insert overscan_info_present_flag syntax of VUI parameters	0x0
[3]	ASPECT_RATIO_INFO_PRESENT_FLAG	R/W	A flag whether to insert aspect_ratio_info_present_flag syntax of VUI parameters	0x0
[2]	RSVD	R/W	Reserved	0x0
[1]	RSVD	R/W	Reserved	0x0
[0]	NEUTRAL_CHROMA_INDICATION_FLAG	R/W	A flag whether to insert neutral_chroma_indication_flag syntax of VUI parameters	0x0

1.2.3.2.4.23. CMD_ENC_VUI_ASPECT_RATIO_IDC (0x00000160)

Table 1.323. CMD_ENC_VUI_ASPECT_RATIO_IDC Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASPECT_RATIO_IDC																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.324. CMD_ENC_VUI_ASPECT_RATIO_IDC Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	ASPECT_RATIO_IDC	R/W	aspect_ratio_idc	0x0

1.2.3.2.4.24. CMD_ENC_VUI_SAR_SIZE (0x00000164)

Table 1.325. CMD_ENC_VUI_SAR_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR_HEIGHT																SAR_WIDTH															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential
StarFive Inc.

Table 1.326. CMD_ENC_VUI_SAR_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	SAR_HEIGHT	R/W	sar_height (valid when aspect_ratio_idc is equal to 255)	0x0
[15:0]	SAR_WIDTH	R/W	sar_width (valid when aspect_ratio_idc is equal to 255)	0x0

1.2.3.2.4.25. CMD_ENC_VUI_OVERSCAN_APPROPRIATE (0x00000168)

Table 1.327. CMD_ENC_VUI_OVERSCAN_APPROPRIATE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVERSCAN_APPROPRIATE_FLAG																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.328. CMD_ENC_VUI_OVERSCAN_APPROPRIATE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	OVERSCAN_APPROPRIATE_FLAG	R/W	overscan_appropriate_flag	0x0

1.2.3.2.4.26. CMD_ENC_VUI_VIDEO_SIGNAL (0x0000016C)

Table 1.329. CMD_ENC_VUI_VIDEO_SIGNAL Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				MATRIX_COEFFS								TRANSFER_CHARACTERISTICS				COLOUR_PRIMARIES								VIDEO_FULL_RANGE_FLAG	VIDEO_FORMAT						
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.330. CMD_ENC_VUI_VIDEO_SIGNAL Field Description

Bit	Name	Type	Function	Reset Value
[31:28]	RSVD	R/W	Reserved	0x0
[27:20]	MATRIX_COEFFS	R/W	matrix_coeffs	0x0
[19:12]	TRANSFER_CHARACTERISTICS	R/W	transfer_characteristics	0x0
[11:4]	COLOUR_PRIMARIES	R/W	colour_primaries	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[3]	VIDEO_FULL_RANGE_FLAG	R/W	video_full_range_flag	0x0
[2:0]	VIDEO_FORMAT	R/W	video_format	0x0

1.2.3.2.4.27. CMD_ENC_VUI_CHROMA_SAMPLE_LOC (0x00000170)

Table 1.331. CMD_ENC_VUI_CHROMA_SAMPLE_LOC Bit Assignment

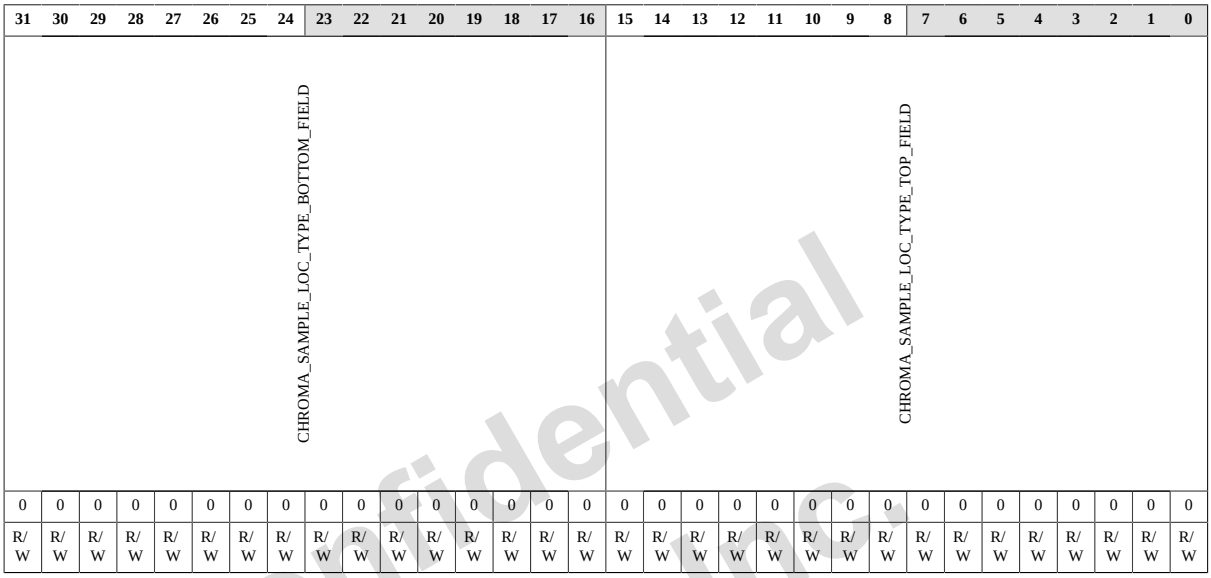


Table 1.332. CMD_ENC_VUI_CHROMA_SAMPLE_LOC Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CHROMA_SAMPLE_LOC_TYPE_BOTTOM_FIELD	R/W	chroma_sample_loc_type_bottom_field	0x0
[15:0]	CHROMA_SAMPLE_LOC_TYPE_TOP_FIELD	R/W	chroma_sample_loc_type_top_field	0x0

1.2.3.2.4.28. CMD_ENC_VUI_DISP_WIN_LEFT_RIGHT (0x00000174)

Table 1.333. CMD_ENC_VUI_DISP_WIN_LEFT_RIGHT Bit Assignment

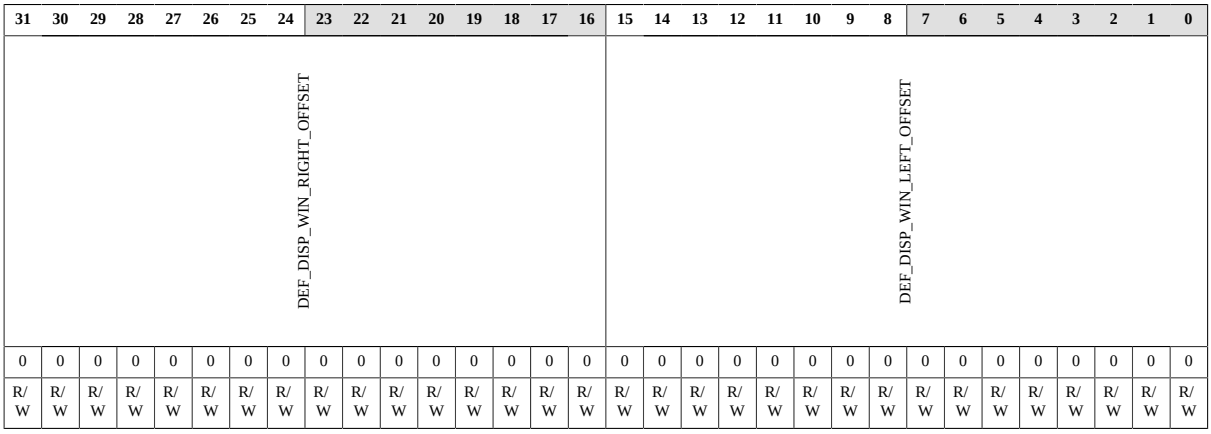


Table 1.334. CMD_ENC_VUI_DISP_WIN_LEFT_RIGHT Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	DEF_DISP_WIN_RIGHT_OFFSET	R/W	def_disp_win_right_offset	0x0

Confidential Starlive Inc.

Bit	Name	Type	Function	Reset Value
[15:0]	DEF_DISP_WIN_LEFT_OFFSET	R/W	def_disp_win_left_offset	0x0

1.2.3.2.4.29. CMD_ENC_VUI_DISP_WIN_TOP_BOT (0x00000178)

Table 1.335. CMD_ENC_VUI_DISP_WIN_TOP_BOT Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEF_DISP_WIN_BOTTOM_OFFSET																DEF_DISP_WIN_TOP_OFFSET															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.336. CMD_ENC_VUI_DISP_WIN_TOP_BOT Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	DEF_DISP_WIN_BOTTOM_OFFSET	R/W	def_disp_win_bottom_offset	0x0
[15:0]	DEF_DISP_WIN_TOP_OFFSET	R/W	def_disp_win_top_offset	0x0

Confidential StarFive Inc

1.2.3.2.5. ENC_PIC Command Parameter Registers

These are command I/O registers where host processor can set arguments for the ENC_PIC command or get return values.

1.2.3.2.5.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.337. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.338. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			• 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.5.2. CORE_INDEX (0x00000104)

V-CORE Index
This is a common parameter register for almost all commands, except VPU control commands such as INIT_VPU, GET_FW_VERSION, SLEEP_VPU, and WAKEUP_VPU.

Table 1.339. CORE_INDEX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VCORE_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.340. CORE_INDEX Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	VCORE_INDEX	RW	V-CORE index - reserved for future use, so please set this as 0.	0x0

1.2.3.2.5.3. INST_INDEX_COD_STD (0x00000108)

Codec Standard and Instance Index
This is a common parameter register for almost all commands (Reserved for future use)

Table 1.341. INST_INDEX_COD_STD Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODEC_STD																INST_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confidential
StarFive Inc.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.342. INST_INDEX_COD_STD Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CODEC_STD	R/W	Codec standard to run HEVC_DEC = 0x0 HEVC_ENC = 0x1	0x0
[15:0]	INST_INDEX	R/W	Instance Index VPU can decode more than one decoding instance simultaneously. If multiple instances are running, each instance must have their own process index that is assigned by this register. For example, two instances are running simultaneously, the first instance has the process index 0, the second instance has the process index 1. Instance index shall be in the range of 0 to 65535, inclusive.	0x0

1.2.3.2.5.4. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.343. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.344. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.5.5. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.345. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.346. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.5.6. BS_START_ADDR (0x00000120)

Bitstream buffer start address

Table 1.347. BS_START_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS_START_ADDR																												RSVD			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

Table 1.348. BS_START_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	BS_START_ADDR	R/W	Start Address of Bitstream Buffer (fixed in ring buffer mode) It should be aligned in bus width (128bit/16 byte).	0x0
[3:0]	RSVD	RO	Reserved	0x0

1.2.3.2.5.7. BS_SIZE (0x00000124)

Bitstream buffer size

Table 1.349. BS_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS_BUF_SIZE																												RSVD			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO

Table 1.350. BS_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	BS_BUF_SIZE	R/W	Size of Bistream buffer (fixed in ring buffer mode) It should be aligned in bus width (128bit/16byte).	0x0
[3:0]	RSVD	RO	Reserved	0x0

1.2.3.2.5.8. BS_PARAM (0x00000128)

Bitstream buffer parameters

Table 1.351. BS_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

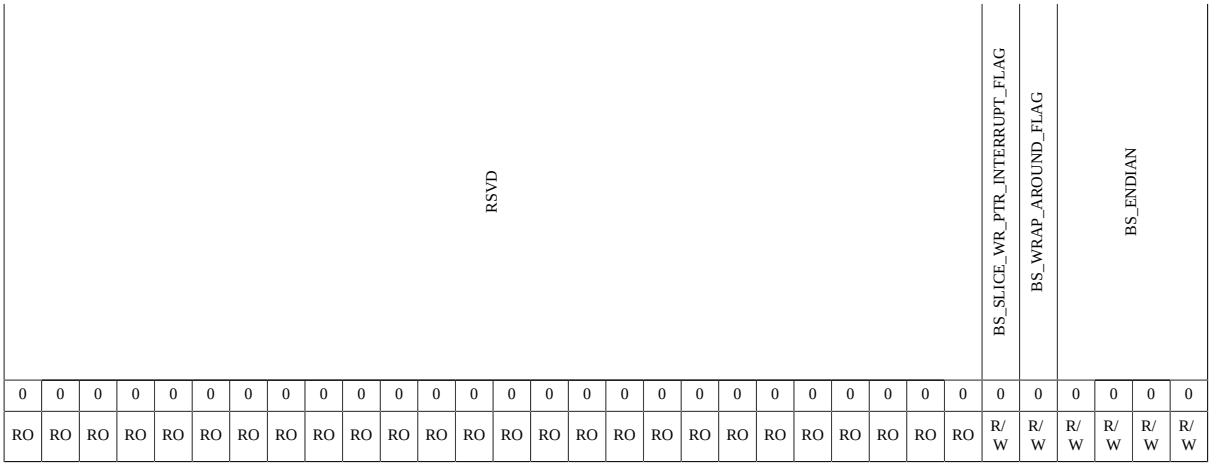


Table 1.352. BS_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:6]	RSVD	RO	Reserved	0x0
[5]	BS_SLICE_WR_PTR_INTERRUPT_FLAG	R/W	It updates wr_ptr each slice.	0x0
[4]	BS_WRAP_AROUND_FLAG	R/W	A wrap around flag	0x0
[3:0]	BS_ENDIAN	R/W	Endianness of bitstream buffer	0x0

1.2.3.2.5.9. BS_OPTIONS (0x0000012C)

Bistream buffer option

Table 1.353. BS_OPTIONS Bit Assignment

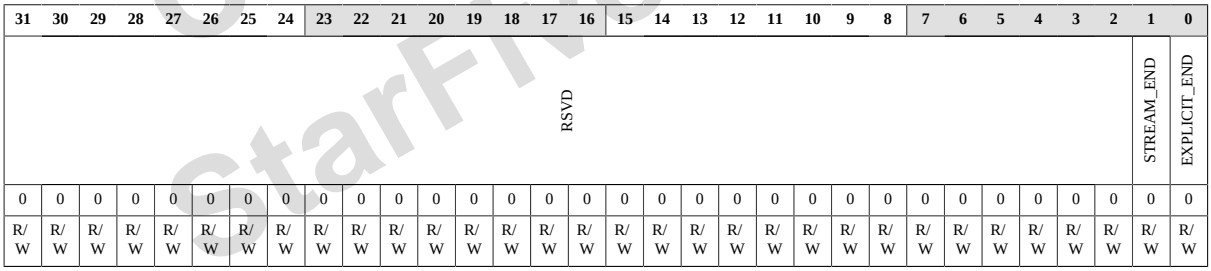


Table 1.354. BS_OPTIONS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	R/W	Reserved	0x0
[1]	STREAM_END	R/W	This field makes VPU assumes Stream End when there is no stream to feed in the buffer.	0x0
[0]	EXPLICIT_END	R/W	Explicit End When this field is set to 1, VPU assumes that bitstream buffer has one NAL unit or more for decoding a signle frame - it works based on the BS_NAL_LEVEL_PUMP register on. Then it follows the tasks below. 1. VPU decodes until the end of bitstream buffer (WR_PTR) even if the last 3 bytes are all 0. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU performs what it is supposed to do with the specified task in BS_SHORTAGE_OPTION of BS_PARAM.	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			If this flag is 0, 1. VPU decodes to the almost end of bitstream buffer (WR_PTR), but not to some bytes (less than 3). It intentionally does not consume some a few bytes, because VPU is not sure if the last bytes are the start code of next NAL or they might not fill the offset in the CABAC. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU stops decoding and waits for more bitstream to be filled. Then host processor can do either feed bitstream more or set EXPLICIT_END to complete decoding anyhow. Bitstream_empty interrupt is asserted when EXPLITCIT_END = 0 or when bitstream buffer is near empty (not real empty) for seamless decoding. **Caution** Host processor can set this register any time, but cannot clear during command processing.	

1.2.3.2.5.10. BS_RD_PTR (0x00000130)

Bistream Buffer Read Pointer

Table 1.355. BS_RD_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD_PTR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.356. BS_RD_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RD_PTR	R/W	Start address of bitstream for handling current command Host processor cannot update this register in the middle of decoding. Updating before start decoding is only allowed. VPU updates this RD_PTR as the scheme described below. • When BS_NAL_LEVEL_PUMP of BS_PARAM is 1, VPU updates RET_ADDR_BS_RD_PTR to the end address of the NAL. • When BS_NAL_LEVEL_PUMP of BS_PARAM is 0, VPU updates RET_ADDR_BS_RD_PTR as end address of the last NAL for a frame.	0x0

1.2.3.2.5.11. BS_WR_PTR (0x00000134)

Bistream Buffer Write Pointer

Table 1.357. BS_WR_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WR_PTR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential
Starfive Inc.

Table 1.358. BS_WR_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WR_PTR	R/W	End address of bitstream for handling current command Host can update this register anytime. If a bitstream_empty interrupt is asserted, host processor should do either feed more bitstream and update WR_PTR or set EXPLICIT_END to complete decoding anyhow.	0x0

1.2.3.2.5.12. ADDR_WORK_BASE (0x00000138)

Work Buffer Base Address

Table 1.359. ADDR_WORK_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORK_BUF_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.360. ADDR_WORK_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WORK_BUF_BASE	R/W	Base address of work buffer for each instance This address should be aligned in 4K boundary.	0x0

1.2.3.2.5.13. WORK_SIZE (0x0000013C)

Work Buffer Size

Table 1.361. WORK_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								WORK_BUF_SIZE							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.362. WORK_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:9]	RSVD	RO	Reserved	0x0
[8:0]	WORK_BUF_SIZE	R/W	Size of work buffer (4K boundary) The size of work buffer should be larger than required minimum work buffer size. This address should be aligned in 4K boundary.	0x0

1.2.3.2.5.14. WORK_PARAM (0x00000140)

Work Buffer Parameter

Confidential StarFive Inc.

Table 1.363. WORK_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												WORK_BUF_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W

Table 1.364. WORK_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	WORK_BUF_ENDIAN	R/W	Endianness of work buffer Not implemented yet - VPU uses work buffer as internal purpose so endianness is not important.	0x0

1.2.3.2.5.15. ADDR_TEMP_BASE (0x00000144)

Temporal Buffer Base Address

Table 1.365. ADDR_TEMP_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEMP_BUF_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.366. ADDR_TEMP_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TEMP_BUF_BASE	R/W	Base address of temporal buffer for this frame **Note** Each V-CORE should set this field for its own temporal buffer.	0x0

1.2.3.2.5.16. TEMP_SIZE (0x00000148)

Temporal Buffer Size

Table 1.367. TEMP_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEMP_BUF_SIZE																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.368. TEMP_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	TEMP_BUF_SIZE	R/W	Temporal buffer size for this frame	0x0

1.2.3.2.5.17. TEMP_PARAM (0x0000014C)

Temporal Buffer Parameter

Table 1.369. TEMP_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												TEMP_BUF_ENDIAIN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W

Table 1.370. TEMP_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	TEMP_BUF_ENDIAIN	R/W	Endianness of temporal buffer **Note** Each V-CORE should set this register for its own temporal buffer.	0x0

1.2.3.2.5.18. ADDR_SEC_AXI (0x00000150)

Secondary AXI Base Address
It is valid only when USE_SEC_AXI is not 0.

Table 1.371. ADDR_SEC_AXI Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_AXI_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.372. ADDR_SEC_AXI Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SEC_AXI_BASE	R/W	The base address of secondary AXI memory	0x0

Confidential StarFive Inc.

1.2.3.2.5.19. SEC_AXI_SIZE (0x00000154)

Seconary AXI Memory Size
It is valid only when USE_SEC_AXI is not 0.

Table 1.373. SEC_AXI_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_AXI_MEM_SIZE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.374. SEC_AXI_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SEC_AXI_MEM_SIZE	R/W	The size of secondary AXI memory	0x0

1.2.3.2.5.20. USE_SEC_AXI (0x00000158)

Secondary AXI Usage option

Table 1.375. USE_SEC_AXI Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																SEC_AXI_LF_ROW	RSVD			SEC_AXI_RDO_ENABLE	RSVD										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	RO	RO	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.376. USE_SEC_AXI Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15]	SEC_AXI_LF_ROW	R/W	Use 2nd AXI temp buffer for Read channel of LF row buffer.	0x0
[14:12]	RSVD	RO	Reserved	0x0
[11]	SEC_AXI_RDO_ENABLE	R/W	Use 2nd AXI temp buffer for Read channel of RDO row buffer.	0x0
[10:0]	RSVD	RO	Reserved	0x0

1.2.3.2.5.21. CMD_ENC_ADDR_REPORT_BASE (0x0000015C)

Table 1.377. CMD_ENC_ADDR_REPORT_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

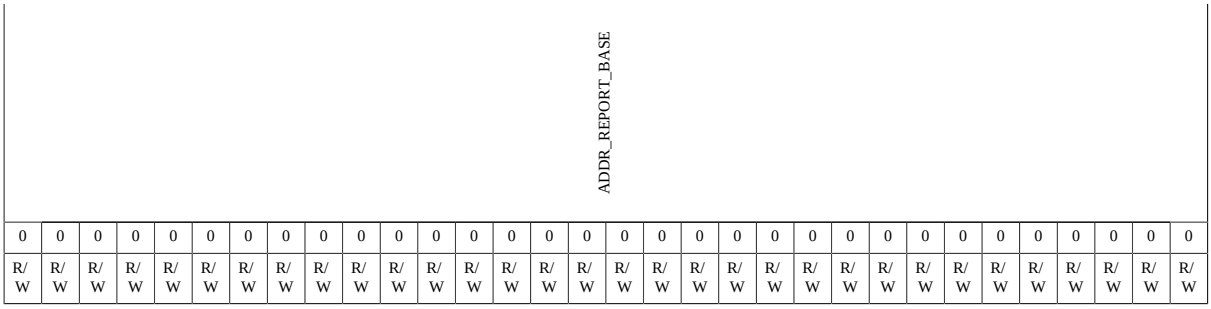


Table 1.378. CMD_ENC_ADDR_REPORT_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	ADDR_REPORT_BASE	R/W	An address of report buffer	0x0

1.2.3.2.5.22. CMD_ENC_REPORT_SIZE (0x00000160)

Table 1.379. CMD_ENC_REPORT_SIZE Bit Assignment

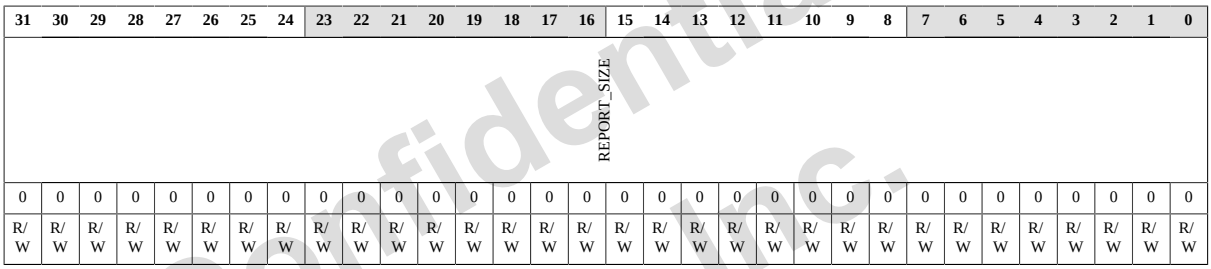


Table 1.380. CMD_ENC_REPORT_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	REPORT_SIZE	R/W	A reporting size in byte	0x0

1.2.3.2.5.23. CMD_ENC_REPORT_PARAM (0x00000164)

Table 1.381. CMD_ENC_REPORT_PARAM Bit Assignment

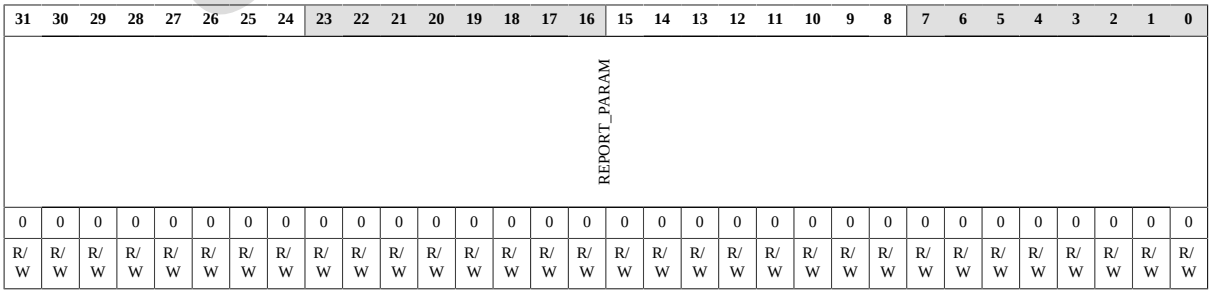
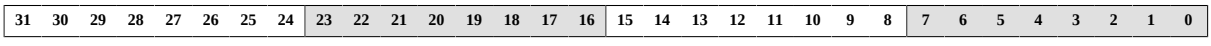


Table 1.382. CMD_ENC_REPORT_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	REPORT_PARAM	R/W	TBD	0x0

1.2.3.2.5.24. CMD_ENC_CODE_OPTION (0x00000168)

Table 1.383. CMD_ENC_CODE_OPTION Bit Assignment



Confidential
Startive Inc.

RSVD																					VUI	SEI	EOB	EOS	AUD	PPS	SPS	VPS	VCL	IMPLICITLY_HEADER_ENCODE
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.384. CMD_ENC_CODE_OPTION Field Description

Bit	Name	Type	Function	Reset Value
[31:10]	RSVD	RO	Reserved	0x0
[9]	VUI	R/W	A flag to encode VUI nal unit explicitly	0x0
[8]	SEI	R/W	A flag to encode SEI nal unit explicitly	0x0
[7]	EOB	R/W	A flag to encode EOB nal unit explicitly	0x0
[6]	EOS	R/W	A flag to encode EOS nal unit explicitly	0x0
[5]	AUD	R/W	A flag to encode AUD nal unit explicitly	0x0
[4]	PPS	R/W	A flag to encode PPS nal unit explicitly	0x0
[3]	SPS	R/W	A flag to encode SPS nal unit explicitly	0x0
[2]	VPS	R/W	A flag to encode VPS nal unit explicitly	0x0
[1]	VCL	R/W	A flag to encode VCL nal unit explicitly	0x0
[0]	IMPLICITLY_HEADER_ENCODE	R/W	A flag to enable to encode a header or headers (VPS, SPS, PPS) implicitly for generating bit-streams conforming to specification	0x0

1.2.3.2.5.25. CMD_ENC_PIC_PARAM (0x0000016C)

Table 1.385. CMD_ENC_PIC_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	FORCE_SRC_PIC_TYPE_FLAG	FORCE_SRC_PIC_QP_FLAG	RSVD					FORCE_PIC_TYPE			USE_FORCE_PIC_TYPE	FORCE_PIC_QP_B				FORCE_PIC_QP_P								FORCE_PIC_QP_I						USE_FORCE_PIC_QP	PIC_SKIP_FLAG
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.386. CMD_ENC_PIC_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31]	RSVD	R/W	Reserved	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[30]	FORCE_SRC_PIC_TYPE_FLAG	R/W	A flag to use a force picture type by encoded/source index	0x0
[29]	FORCE_SRC_PIC_QP_FLAG	R/W	A flag to use a force qp by encoded/source index	0x0
[28:24]	RSVD	R/W	Reserved	0x0
[23:21]	FORCE_PIC_TYPE	R/W	A force picture type (I, P, B, IDR, CRA) or an irap type 0 : I picture 1 : P picture 2 : B picture 3 : IDR picture 4 : CRA picture	0x0
[20]	USE_FORCE_PIC_TYPE	R/W	A flag whether to use a force picture type or an irap type	0x0
[19:14]	FORCE_PIC_QP_B	R/W	A force picture quantization parameter for B picture (0 ~ 51)	0x0
[13:8]	FORCE_PIC_QP_P	R/W	A force picture quantization parameter for P picture (0 ~ 51)	0x0
[7:2]	FORCE_PIC_QP_I	R/W	A force picture quantization parameter for I picture (0 ~ 51)	0x0
[1]	USE_FORCE_PIC_QP	R/W	A flag to use a force picture quantization parameter	0x0
[0]	PIC_SKIP_FLAG	R/W	A flag to skip the current picture	0x0

1.2.3.2.5.26. CMD_ENC_SRC_PIC_IDX (0x00000170)

Table 1.387. CMD_ENC_SRC_PIC_IDX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_IDX																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.388. CMD_ENC_SRC_PIC_IDX Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SRC_IDX	R/W	A buffer index of a source picture (-2 : no more source picture)	0x0

1.2.3.2.5.27. CMD_ENC_SRC_ADDR_Y (0x00000174)

Table 1.389. CMD_ENC_SRC_ADDR_Y Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_ADDR_Y																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.390. CMD_ENC_SRC_ADDR_Y Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SRC_ADDR_Y	R/W	An address of Y component of a source picture	0x0

1.2.3.2.5.28. CMD_ENC_SRC_ADDR_U (0x00000178)

Table 1.391. CMD_ENC_SRC_ADDR_U Bit Assignment

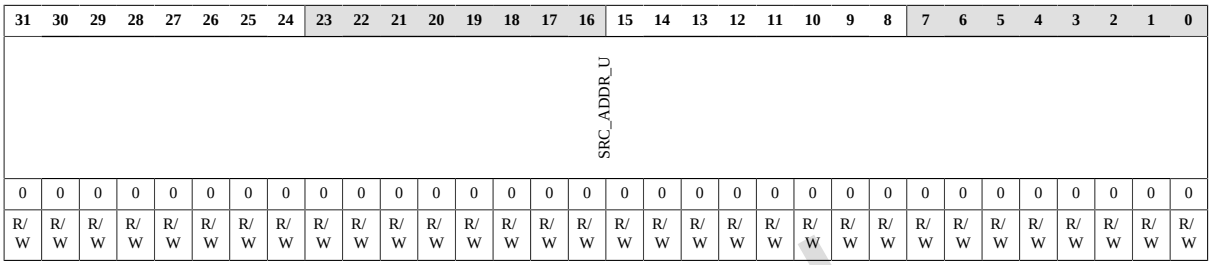


Table 1.392. CMD_ENC_SRC_ADDR_U Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SRC_ADDR_U	R/W	An address of U component of a source picture	0x0

1.2.3.2.5.29. CMD_ENC_SRC_ADDR_V (0x0000017C)

Table 1.393. CMD_ENC_SRC_ADDR_V Bit Assignment

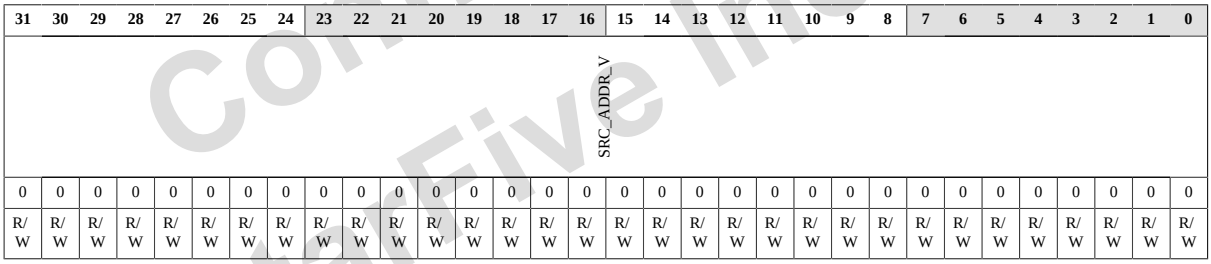


Table 1.394. CMD_ENC_SRC_ADDR_V Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SRC_ADDR_V	R/W	An address of V component of a source picture	0x0

1.2.3.2.5.30. CMD_ENC_SRC_STRIDE (0x00000180)

Table 1.395. CMD_ENC_SRC_STRIDE Bit Assignment

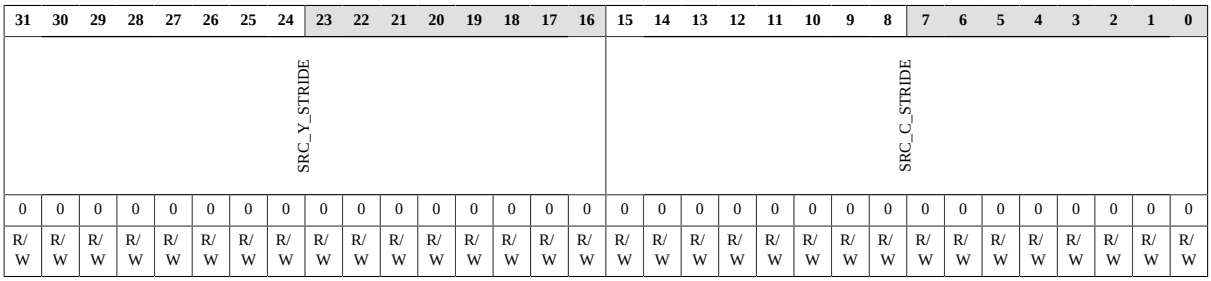


Table 1.396. CMD_ENC_SRC_STRIDE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	SRC_Y_STRIDE	R/W	A luma stride of source picture in pixel	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[15:0]	SRC_C_STRIDE	R/W	A chroma stride of source picture in pixel	0x0

1.2.3.2.5.31. CMD_ENC_SRC_FORMAT (0x00000184)

Table 1.397. CMD_ENC_SRC_FORMAT Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						SRC_ENDIAN				SRC_PIXEL_FORMAT			SRC_FRAME_FORMAT		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.398. CMD_ENC_SRC_FORMAT Field Description

Bit	Name	Type	Function	Reset Value
[31:10]	RSVD	RO	Reserved	0x0
[9:6]	SRC_ENDIAN	R/W	Endianess of source picture	0x0
[5:3]	SRC_PIXEL_FORMAT	R/W	[2] left justified [1:0] 0 : 8bit 1 : 16bit (1 pixel / 2 byte) 2 : 32bit (3 pixel / 4 byte)	0x0
[2:0]	SRC_FRAME_FORMAT	R/W	0 : Planar 1 : Tiled Sub-CTU frame map - within a sub-CTU(32x32), the first row of 16x32 is read in vertical direction and then the second row of 16x32 is read. 2 : CbCr interleaved (NV12) 3 : CrCb interleaved (NV21) 4 : Packed mode (YUYV) 5 : Packed mode (YVYU) 6 : Packed mode (UYVY) 7 : Packed mode (VYUY)	0x0

1.2.3.2.5.32. CMD_ENC_PREFIX_SEI_NAL_ADDR (0x00000188)

Table 1.399. CMD_ENC_PREFIX_SEI_NAL_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PREFIX_SEI_NAL_DATA_ADDR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.400. CMD_ENC_PREFIX_SEI_NAL_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	PREFIX_SEI_NAL_DATA_ADDR	R/W	The start address of the total prefix SEI NALs to be encoded	0x0

1.2.3.2.5.33. CMD_ENC_PREFIX_SEI_INFO (0x0000018C)

Table 1.401. CMD_ENC_PREFIX_SEI_INFO Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PREFIX_SEI_NAL_DATA_SIZE																RSVD														PREFIX_SEI_TIMING_FLAG	PREFIX_SEI_NAL_DATA_ENABLE_FLAG
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.402. CMD_ENC_PREFIX_SEI_INFO Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	PREFIX_SEI_NAL_DATA_SIZE	R/W	The total byte size of the prefix SEI	0x0
[15:2]	RSVD	RO	Reserved	0x0
[1]	PREFIX_SEI_TIMING_FLAG	R/W	A flag whether to encode PREFIX_SEI_DATA with a picture of this command or with a source picture of the buffer at the moment 0 : encode PREFIX_SEI_DATA when a source picture is encoded. 1 : encode PREFIX_SEI_DATA at this command.	0x0
[0]	PREFIX_SEI_NAL_DATA_ENABLE_FLAG	R/W	It enables to encode the prefix SEI NAL which is given by host.	0x0

1.2.3.2.5.34. CMD_ENC_SUFFIX_SEI_NAL_ADDR (0x00000190)

Table 1.403. CMD_ENC_SUFFIX_SEI_NAL_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUFFIX_SEI_NAL_DATA_ADDR																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.404. CMD_ENC_SUFFIX_SEI_NAL_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SUFFIX_SEI_NAL_DATA_ADDR	R/W	The start address of the total suffix SEI NALs to be encoded	0x0

1.2.3.2.5.35. CMD_ENC_SRC_TIMESTAMP_LOW (0x00000190)

Table 1.405. CMD_ENC_SRC_TIMESTAMP_LOW Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_TIMESTAMP_LOW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.406. CMD_ENC_SRC_TIMESTAMP_LOW Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SRC_TIMESTAMP_LOW	R/W	An LSB of timestamp of a source picture	0x0

1.2.3.2.5.36. CMD_ENC_SUFFIX_SEI_INFO (0x00000194)

Table 1.407. CMD_ENC_SUFFIX_SEI_INFO Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUFFIX_SEI_NAL_DATA_SIZE																RSVD														SUFFIX_SEI_TIMING_FLAG	SUFFIX_SEI_NAL_DATA_ENABLE_FLAG
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.408. CMD_ENC_SUFFIX_SEI_INFO Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	SUFFIX_SEI_NAL_DATA_SIZE	R/W	The total byte size of the suffix SEI	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[15:2]	RSVD	RO	Reserved	0x0
[1]	SUFFIX_SEI_TIMING_FLAG	R/W	A flag whether to encode SUFFIX_SEI_DATA with a picture of this command or with a source picture of the buffer at the moment 0 : encode SUFFIX_SEI_DATA when a source picture is encoded. 1 : encode SUFFIX_SEI_DATA at this command.	0x0
[0]	SUFFIX_SEI_NAL_DATA_ENABLE_FLAG	R/W	It enables to encode the suffix SEI NAL which is given by host.	0x0

1.2.3.2.5.37. CMD_ENC_SRC_TIMESTAMP_HIGH (0x00000194)

Table 1.409. CMD_ENC_SRC_TIMESTAMP_HIGH Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_TIMESTAMP_HIGH																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.410. CMD_ENC_SRC_TIMESTAMP_HIGH Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SRC_TIMESTAMP_HIGH	R/W	An MSB of timestamp of a source picture	0x0

1.2.3.2.5.38. CMD_ENC_LONGTERM_PIC (0x00000198)

Table 1.411. CMD_ENC_LONGTERM_PIC Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REF_LONGTERM_PIC_POC																RSVD														USE_REF_LONGTERM_PIC	USE_SRC_LONGTERM_PIC
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.412. CMD_ENC_LONGTERM_PIC Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	REF_LONGTERM_PIC_POC	R/W	A POC of a longterm picture to be used as a reference picture	0x0

Confidential
Starfive Inc.

Bit	Name	Type	Function	Reset Value
[15:2]	RSVD	RO	Reserved	0x0
[1]	USE_REF_LONGTERM_PIC	R/W	A flag to use a longterm picture in DPB to encode a picture	0x0
[0]	USE_SRC_LONGTERM_PIC	R/W	A flag to use a source picture as a longterm reference picture later	0x0

1.2.3.2.5.39. CMD_ENC_ROI_PARAM (0x000001A0)

Table 1.413. CMD_ENC_ROI_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTU_MAP_STRIDE																CTU_MAP_ENDIAN				RSVD		CTU_QP_MAP_ENABLE_FLAG	CTU_MODE_ENABLE_FLAG	ROI_DELTA_QP							ROI_ENABLE_FLAG
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.414. CMD_ENC_ROI_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CTU_MAP_STRIDE	R/W	Stride of CTU map Set this with (Width + CTB_SIZE - 1) / CTB_SIZE	0x0
[15:12]	CTU_MAP_ENDIAN	R/W	Endianess of CTU map	0x0
[11:10]	RSVD	R/W	Reserved	0x0
[9]	CTU_QP_MAP_ENABLE_FLAG	R/W	It enables CTU QP map that allows CTUs to be encoded with the given QPs.	0x0
[8]	CTU_MODE_ENABLE_FLAG	R/W	It enables CTU mode map that allows CTUs to be encoded with force intra or to be skipped.	0x0
[7:1]	ROI_DELTA_QP	R/W	It specifies the delta QP for ROI important level.	0x0
[0]	ROI_ENABLE_FLAG	R/W	It enables ROI map.	0x0

1.2.3.2.5.40. CMD_ENC_ROI_MAP_ADDR (0x000001A4)

Table 1.415. CMD_ENC_ROI_MAP_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROI_MAP_ADDR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.416. CMD_ENC_ROI_MAP_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	ROI_MAP_ADDR	R/W	Start buffer address of ROI map The memory size is the number of CTUs of picture in bytes. For example if there are 64 CTUs within a picture, the size of ROI map is 64 bytes. All CTUs have their ROI importance level (0 ~ 8 ; 1 byte) in raster order. The higher an ROI level is, the more important the region is with a lower QP.	0x0

1.2.3.2.5.41. CMD_ENC_CTU_MODE_MAP_ADDR (0x000001A8)

Table 1.417. CMD_ENC_CTU_MODE_MAP_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTU_MODE_MAP_ADDR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.418. CMD_ENC_CTU_MODE_MAP_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CTU_MODE_MAP_ADDR	R/W	Start buffer address of CTU mode map The memory size is the number of CTUs of picture in bytes. For example if there are 64 CTUs within a picture, the size of CTU map is 64 bytes. It should be given when CTU_MODE_ENABLE_FLAG is 1.	0x0

1.2.3.2.5.42. RET_ENC_PIC_IDX (0x000001A8)

Table 1.419. RET_ENC_PIC_IDX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIC_IDX								RSVD																							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.420. RET_ENC_PIC_IDX Field Description

Bit	Name	Type	Function	Reset Value
[31:24]	PIC_IDX	R/W	A frame buffer index of the encoded picture -2 : encoding delay (default) -1 : encoding end	0x0
[23:0]	RSVD	RO	Reserved	0x0

1.2.3.2.5.43. CMD_ENC_CTU_QP_MAP_ADDR (0x000001AC)

Table 1.421. CMD_ENC_CTU_QP_MAP_ADDR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

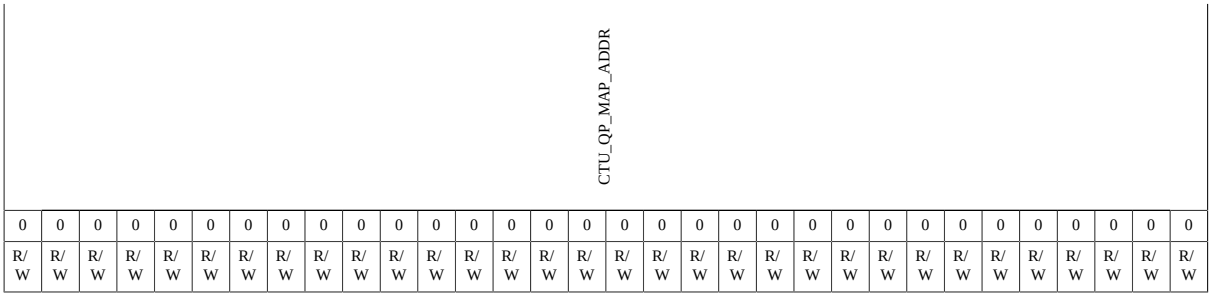


Table 1.422. CMD_ENC_CTU_QP_MAP_ADDR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CTU_QP_MAP_ADDR	R/W	Start buffer address of CTU qp map The memory size is the number of CTUs of picture in bytes. For example if there are 64 CTUs within a picture, the size of CTU map is 64 bytes. It should be given when CTU_QP_MAP_ENABLE_FLAG is 1.	0x0

1.2.3.2.5.44. RET_ENC_PIC_SLICE_NUM (0x000001AC)

Table 1.423. RET_ENC_PIC_SLICE_NUM Bit Assignment

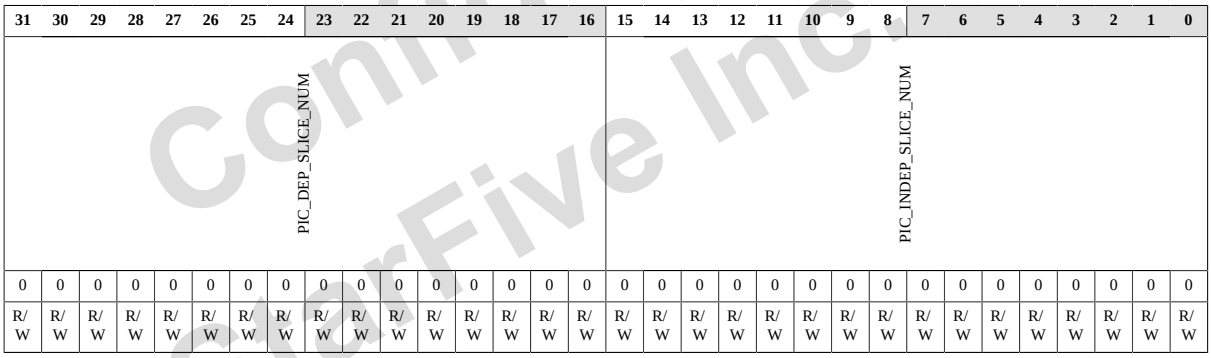
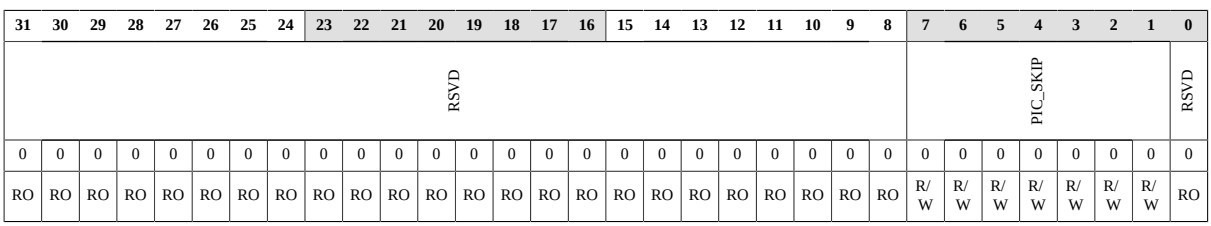


Table 1.424. RET_ENC_PIC_SLICE_NUM Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	PIC_DEP_SLICE_NUM	R/W	The total dependent slice segment number in the encoded picture	0x0
[15:0]	PIC_INDEP_SLICE_NUM	R/W	The total independent slice segment number in the encoded picture	0x0

1.2.3.2.5.45. RET_ENC_PIC_SKIP (0x000001B0)

Table 1.425. RET_ENC_PIC_SKIP Bit Assignment



Confidential StarFive Inc.

Table 1.426. RET_ENC_PIC_SKIP Field Description

Bit	Name	Type	Function	Reset Value
[31:8]	RSVD	RO	Reserved	0x0
[7:1]	PIC_SKIP	R/W	A picture skip flag	0x0
[0]	RSVD	RO	Reserved	0x0

1.2.3.2.5.46. RET_ENC_PIC_NUM_INTRA (0x000001B4)

Table 1.427. RET_ENC_PIC_NUM_INTRA Bit Assignment

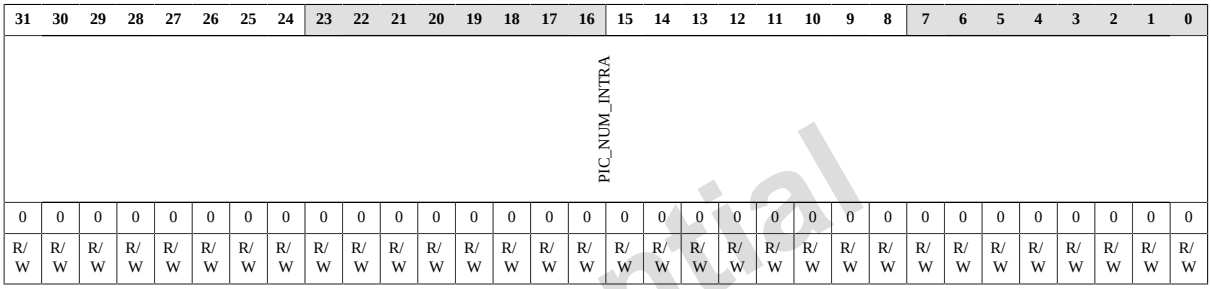


Table 1.428. RET_ENC_PIC_NUM_INTRA Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	PIC_NUM_INTRA	R/W	The number of intra block in 8x8	0x0

1.2.3.2.5.47. RET_ENC_PIC_NUM_MERGE (0x000001B8)

Table 1.429. RET_ENC_PIC_NUM_MERGE Bit Assignment

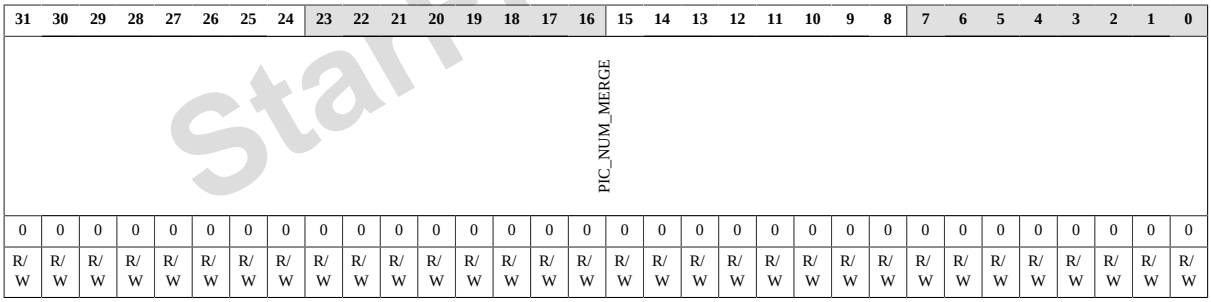
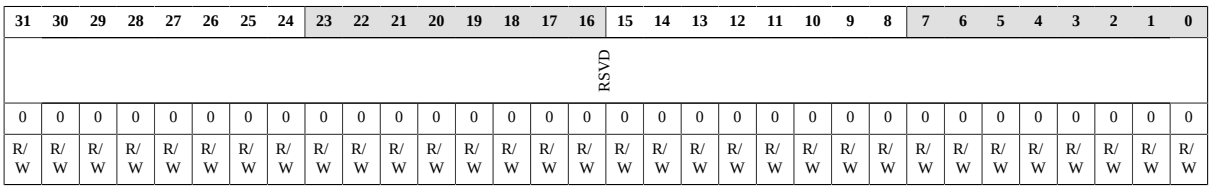


Table 1.430. RET_ENC_PIC_NUM_MERGE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	PIC_NUM_MERGE	R/W	The number of merge block in 8x8	0x0

1.2.3.2.5.48. RET_ENC_PIC_RESERVED (0x000001BC)

Table 1.431. RET_ENC_PIC_RESERVED Bit Assignment



Confidential StarFive Inc

Table 1.432. RET_ENC_PIC_RESERVED Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RSVD	R/W	Reserved	0x0

1.2.3.2.5.49. RET_ENC_PIC_NUM_SKIP (0x000001C0)

Table 1.433. RET_ENC_PIC_NUM_SKIP Bit Assignment

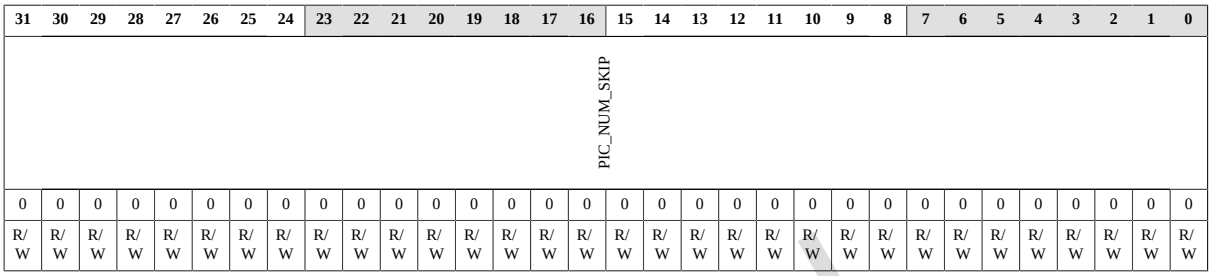
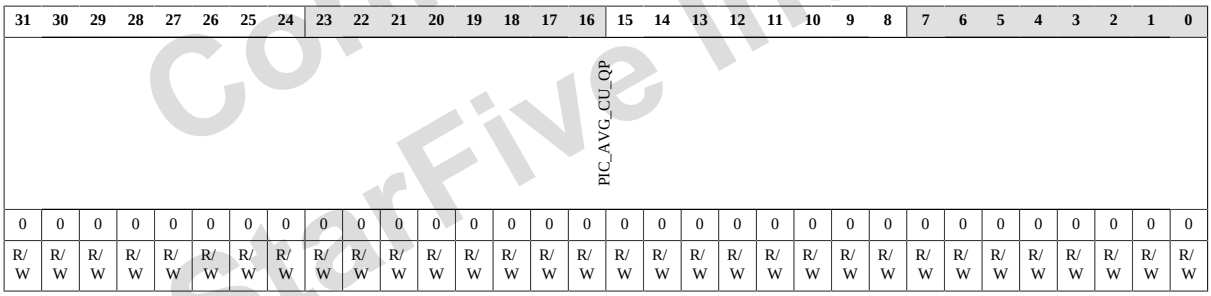


Table 1.434. RET_ENC_PIC_NUM_SKIP Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	PIC_NUM_SKIP	R/W	The number of skip block in 8x8	0x0

1.2.3.2.5.50. RET_ENC_PIC_AVG_CU_QP (0x000001C4)

Table 1.435. RET_ENC_PIC_AVG_CU_QP Bit Assignment



Confidential StarFive Inc

Table 1.436. RET_ENC_PIC_AVG_CU_QP Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	PIC_AVG_CU_QP	R/W	An average value of CU QPs	0x0

1.2.3.2.5.51. RET_ENC_PIC_BYTE (0x000001C8)

Table 1.437. RET_ENC_PIC_BYTE Bit Assignment

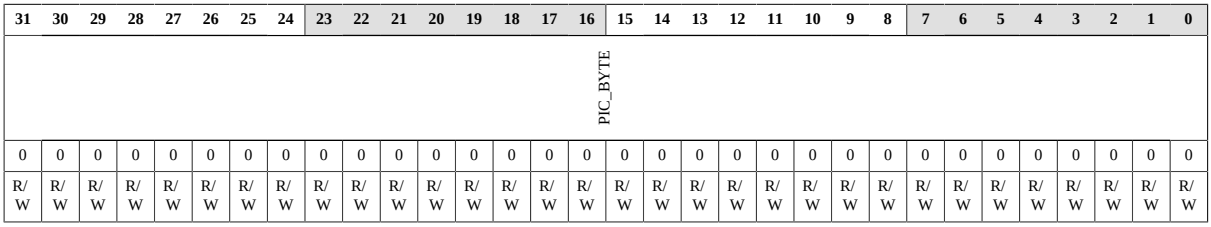


Table 1.438. RET_ENC_PIC_BYTE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	PIC_BYTE	R/W	The size of encoded picture in byte	0x0

1.2.3.2.5.52. RET_ENC_GOP_PIC_IDX (0x000001CC)

Table 1.439. RET_ENC_GOP_PIC_IDX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GOP_PIC_IDX																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.440. RET_ENC_GOP_PIC_IDX Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	GOP_PIC_IDX	R/W	A picture index in GOP	0x0

1.2.3.2.5.53. RET_ENC_PIC_POC (0x000001D0)

Table 1.441. RET_ENC_PIC_POC Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIC_POC																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.442. RET_ENC_PIC_POC Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	PIC_POC	R/W	A POC value of the current picture	0x0

1.2.3.2.5.54. RET_FRAME_CYCLE (0x000001D4)

A frame processing cycle

Table 1.443. RET_FRAME_CYCLE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRAME_CYCLE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.444. RET_FRAME_CYCLE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FRAME_CYCLE	R/W	The number of processing cycle for a frame (1 ~ 16)	0x0

Confidential StarFive Inc.

1.2.3.2.5.55. RET_ENC_USED_SRC_IDX (0x000001D8)

Table 1.445. RET_ENC_USED_SRC_IDX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USED_SRC_PIC_IDX																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.446. RET_ENC_USED_SRC_IDX Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	USED_SRC_PIC_IDX	R/W	A source buffer index of the encoded picture (-2 : encoding delay)	0x0

1.2.3.2.5.56. RET_ENC_PIC_NUM (0x000001DC)

Table 1.447. RET_ENC_PIC_NUM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIC_NUM																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.448. RET_ENC_PIC_NUM Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	PIC_NUM	R/W	The encoded picture number	0x0

1.2.3.2.5.57. RET_ENC_PIC_TYPE (0x000001E0)

Table 1.449. RET_ENC_PIC_TYPE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																PIC_TYPE															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.450. RET_ENC_PIC_TYPE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	PIC_TYPE	R/W	The encoded picture type 0 : I slice 1 : P slice	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			2 : B slice	

Confidential
StarFive Inc.

1.2.3.2.6. SET_FRAMEBUF Command Parameter Registers

These are command I/O registers where host processor can set arguments for the SET_FRAMEBUF command or get return values.

1.2.3.2.6.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.451. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.452. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			• 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.6.2. CORE_INDEX (0x00000104)

V-CORE Index
This is a common parameter register for almost all commands, except VPU control commands such as INIT_VPU, GET_FW_VERSION, SLEEP_VPU, and WAKEUP_VPU.

Table 1.453. CORE_INDEX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VCORE_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.454. CORE_INDEX Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	VCORE_INDEX	RW	V-CORE index - reserved for future use, so please set this as 0.	0x0

1.2.3.2.6.3. INST_INDEX_COD_STD (0x00000108)

Codec Standard and Instance Index
This is a common parameter register for almost all commands (Reserved for future use)

Table 1.455. INST_INDEX_COD_STD Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODEC_STD																INST_INDEX															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.456. INST_INDEX_COD_STD Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CODEC_STD	R/W	Codec standard to run HEVC_DEC = 0x0 HEVC_ENC = 0x1	0x0
[15:0]	INST_INDEX	R/W	Instance Index VPU can decode more than one decoding instance simultaneously. If multiple instances are running, each instance must have their own process index that is assigned by this register. For example, two instances are running simultaneously, the first instance has the process index 0, the second instance has the process index 1. Instance index shall be in the range of 0 to 65535, inclusive.	0x0

1.2.3.2.6.4. SFB_OPTION (0x0000010C)

Set frame buffer option (Reserved)

Table 1.457. SFB_OPTION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						FBC_MODE						FB_ENDIAN				RSVD											SETUP_DONE	SETUP_START	SFB_OPTION		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Table 1.458. SFB_OPTION Field Description

Bit	Name	Type	Function	Reset Value
[31:26]	RSVD	RO	Reserved	0x0
[25:20]	FBC_MODE	R/W	FBC prediction control 0x00: Best predection (best for bandwidth, some performance overhead) 0x0C: Normal prediction (good for bandwidth and performance) 0x3C: Basic predcition (best for performance) WARNING) Do not set this as 0x3F, which might cause unpredictable operation.	0x0
[19:16]	FB_ENDIAN	R/W	Framebuffer endianness	0x0
[15:5]	RSVD	RO	Reserved	0x0
[4]	SETUP_DONE	R/W	Setup Done Please set this flag as 1 when setup for frame buffer is done	0x0
[3]	SETUP_START	R/W	Setup Start Please set this flag as 1 when host issues SET_FRAME_BUF command for the first time. In other words, this is 0 when giving VPU the second SET_FRAME_BUF command for unregistered frame buffers. (only 8 framebuffers possible to register at once)	0x0
[2:0]	SFB_OPTION	R/W	0x0 : SET_FRAME_BUF	0x0

Confidential
StarFive Inc.

Bit	Name	Type	Function	Reset Value
			0x1~0x7 : Reserved	

1.2.3.2.6.5. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.459. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.460. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.6.6. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.461. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.462. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.6.7. COMMON_PIC_INFO (0x00000120)

Dpb Information

Table 1.463. COMMON_PIC_INFO Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential StarFive Inc.

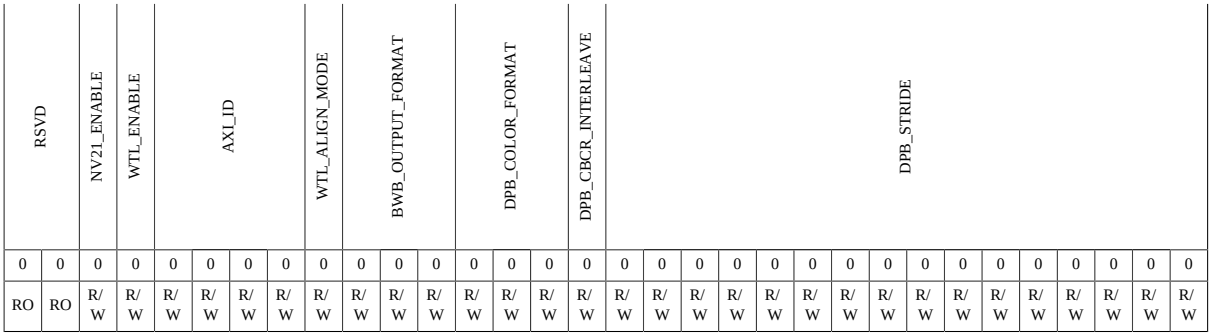


Table 1.464. COMMON_PIC_INFO Field Description

Bit	Name	Type	Function	Reset Value
[31:30]	RSVD	RO	Reserved	0x0
[29]	NV21_ENABLE	R/W	This field enables NV21 format for BWB output. If this field is 0, BWB output color format is NV12. Otherwise, BWB output color format is NV21. • The value of NV21 is valid when both BWB and Cb/Cr interleave are enabled. • This field is valid only if WTL_ENABLE = 1	0x0
[28]	WTL_ENABLE	R/W	WTL_ENABLE This fields indicates target frame buffers are used for uncompressed linear frame. When this field is 0, target frame buffers are used for CFrame - frame buffer compression.	0x0
[27:24]	AXI_ID	R/W	AXI_ID to distinguish guest OS For virtualization only. Set this value in highest bit order.	0x0
[23]	WTL_ALIGN_MODE	R/W	Pixel Alignment Mode (Not Implemented Yet) 0: place lower position pixel in MSB part in the bus - MSB like pixel alignment. 1: place lower position pixel in LSB part in the bus - LSB like pixel alignment This field is valid only if WTL_ENABLE = 1	0x0
[22:20]	BWB_OUTPUT_FORMAT	R/W	BWB output format 000: 8bit down sampling (1 byte/pixel) 001: A pixel is stored in 16 bit and aligned to MSB part (2 bytes/pixel) 010: 3 pixels are stored in 32 bit and aligned to MSB part (4 bytes/3 pixels) 011: Reserved 100: 8bit down sampling (1 byte/pixel) 101: A pixel is stored in 16 bit and aligned to LSB part (2 bytes/pixel) 110: 3 pixels are stored in 32 bit and aligned to LSB part (4 bytes/3 pixels) 111: Reserved • This field is valid only if sequence is MAIN10 profile and bit-depth is bigger than 8. • This field is valid only if WTL_ENABLE = 1.	0x0
[19:17]	DPB_COLOR_FORMAT	R/W	Color format 000: YUV420 • This field is valid only if WTL_ENABLE = 1. • This field ALWAYS should have 000.	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[16]	DPB_CBCR_INTERLEAVE	R/W	Cb/Cr interleaved 1 : set linear frame buffer output as Chroma inter-leaved format (semi-planar format) 0 : set linear frame buffer output as planar format (separated format) This flag is valid only if WTL_ENABLE = 1	0x0
[15:0]	DPB_STRIDE	R/W	Stride for the storing alignment	0x0

1.2.3.2.6.8. PIC_SIZE (0x00000124)

Decoded Picture Size

Table 1.465. PIC_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPB_WIDTH																DPB_HEIGHT															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.466. PIC_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	DPB_WIDTH	R/W	Dpb Picture Width	0x0
[15:0]	DPB_HEIGHT	R/W	Dpb Picture Height	0x0

1.2.3.2.6.9. SET_FB_NUM (0x00000128)

Option of set frame buffer

Table 1.467. SET_FB_NUM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																			FB_NUM_START					RSVD				FB_NUM_END			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	R/W	R/W	R/W	R/W	R/W

Table 1.468. SET_FB_NUM Field Description

Bit	Name	Type	Function	Reset Value
[31:13]	RSVD	RO	Reserved	0x0
[12:8]	FB_NUM_START	R/W	The number of start frame buffer to set using SET_FRAME_BUFFER comamnd. This value is valid only if SFB_OPTION is set as 0 (SET_FRAME_BUF).	0x0
[7:5]	RSVD	RO	Reserved	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
[4:0]	FB_NUM_END	R/W	The number of end frame buffer to set using SET_FRAME_BUFFER comamnd. This value is valid only if SFB_OPTION is set as 0 (SET_FRAME_BUF).	0x0

1.2.3.2.6.10. ADDR_WORK_BASE (0x00000138)

Work Buffer Base Address

Table 1.469. ADDR_WORK_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORK_BUF_BASE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.470. ADDR_WORK_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WORK_BUF_BASE	R/W	Base address of work buffer for each instance This address should be aligned in 4K boundary.	0x0

1.2.3.2.6.11. WORK_SIZE (0x0000013C)

Work Buffer Size

Table 1.471. WORK_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								WORK_BUF_SIZE							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.472. WORK_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:9]	RSVD	RO	Reserved	0x0
[8:0]	WORK_BUF_SIZE	R/W	Size of work buffer (4K boundary) The size of work buffer should be larger than required minimum work buffer size. This address should be aligned in 4K boundary.	0x0

1.2.3.2.6.12. WORK_PARAM (0x00000140)

Work Buffer Parameter

Confidential StarFive Inc.

Table 1.473. WORK_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												WORK_BUF_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W

Table 1.474. WORK_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	WORK_BUF_ENDIAN	R/W	Endianness of work buffer Not implemented yet - VPU uses work buffer for internal purpose so endianness is not important.	0x0

1.2.3.2.6.13. FBC_STRIDE (0x00000154)

Table 1.475. FBC_STRIDE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_Y_STRIDE																FBC_C_STRIDE															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.476. FBC_STRIDE Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	FBC_Y_STRIDE	R/W	A stride of fbc for luma component	0x0
[15:0]	FBC_C_STRIDE	R/W	A stride of fbc for chroma component	0x0

1.2.3.2.6.14. ADDR_SUB_SAMPLED_FB_BASE (0x00000158)

Table 1.477. ADDR_SUB_SAMPLED_FB_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_SUB_SAMPLED_FB																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential
StarFive Inc.

Table 1.478. ADDR_SUB_SAMPLED_FB_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	ADDR_SUB_SAMPLED_FB	R/W	A base address of frame buffer for sub sampled decoded frames	0x0

1.2.3.2.6.15. SUB_SAMPLED_ONE_FB_SIZE (0x0000015C)

Table 1.479. SUB_SAMPLED_ONE_FB_SIZE Bit Assignment

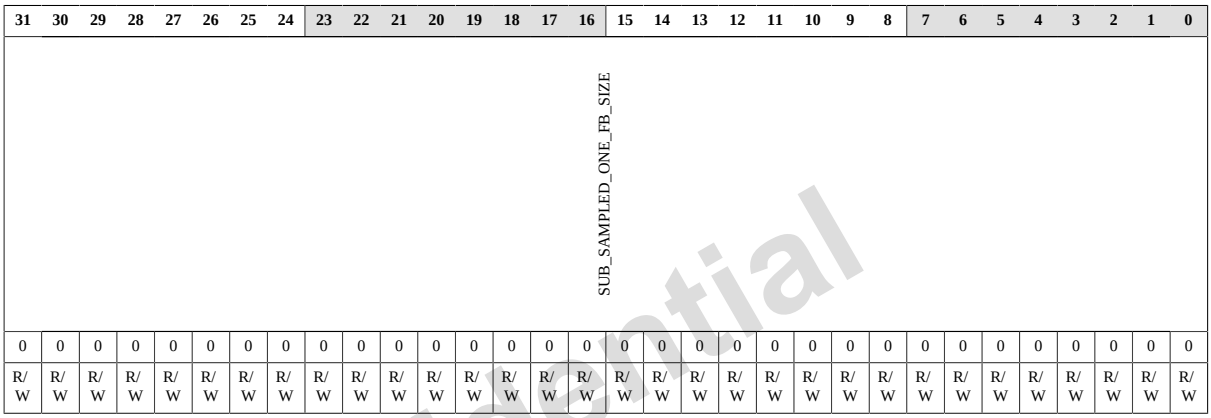


Table 1.480. SUB_SAMPLED_ONE_FB_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	SUB_SAMPLED_ONE_FB_SIZE	R/W	A size of one sub-sampled decoded frame	0x0

1.2.3.2.6.16. ADDR_LUMA_BASE0 (0x00000160)

Luma base of index0

Table 1.481. ADDR_LUMA_BASE0 Bit Assignment

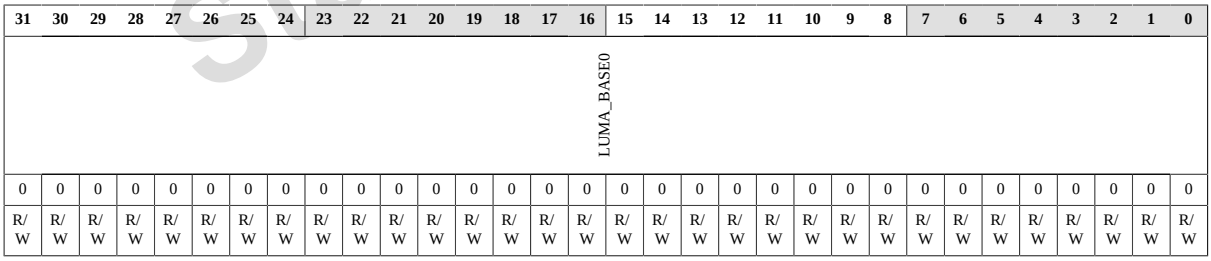


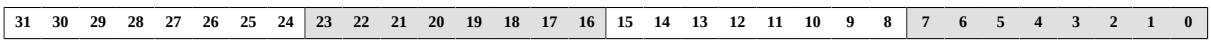
Table 1.482. ADDR_LUMA_BASE0 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LUMA_BASE0	R/W	Luma base address of DPB index 0 unless FBC is used. If FBC is used, this is compressed Luma base address of DPB index 0.	0x0

1.2.3.2.6.17. ADDR_CB_BASE0 (0x00000164)

Cb base of index0

Table 1.483. ADDR_CB_BASE0 Bit Assignment



Confidential StarFive Inc.

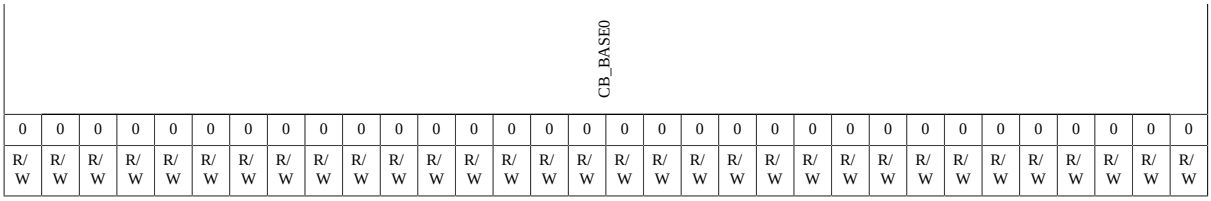


Table 1.484. ADDR_CB_BASE0 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CB_BASE0	R/W	Cb base address of DPB index 0 unless FBC is used. If FBC is used, this is compressed Cb and Cr base address of DPB index 0.	0x0

1.2.3.2.6.18. ADDR_CR_BASE0 (0x00000168)

Cr base of index0

Table 1.485. ADDR_CR_BASE0 Bit Assignment

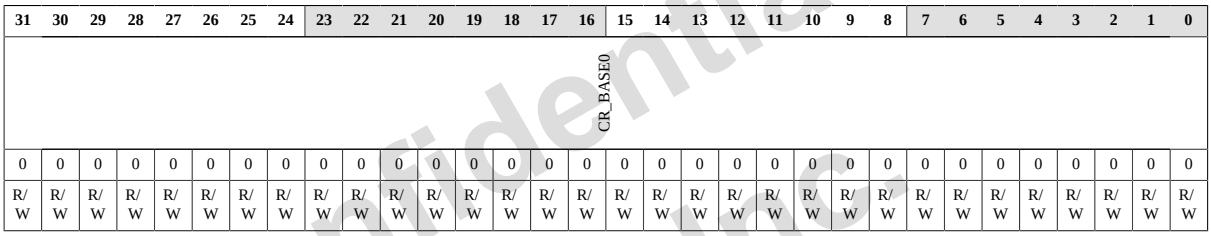


Table 1.486. ADDR_CR_BASE0 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE0	R/W	Cr base address of DPB index 0 unless FBC is used.	0x0

1.2.3.2.6.19. ADDR_FBC_Y_OFFSET0 (0x00000168)

Cr base of index0

Table 1.487. ADDR_FBC_Y_OFFSET0 Bit Assignment

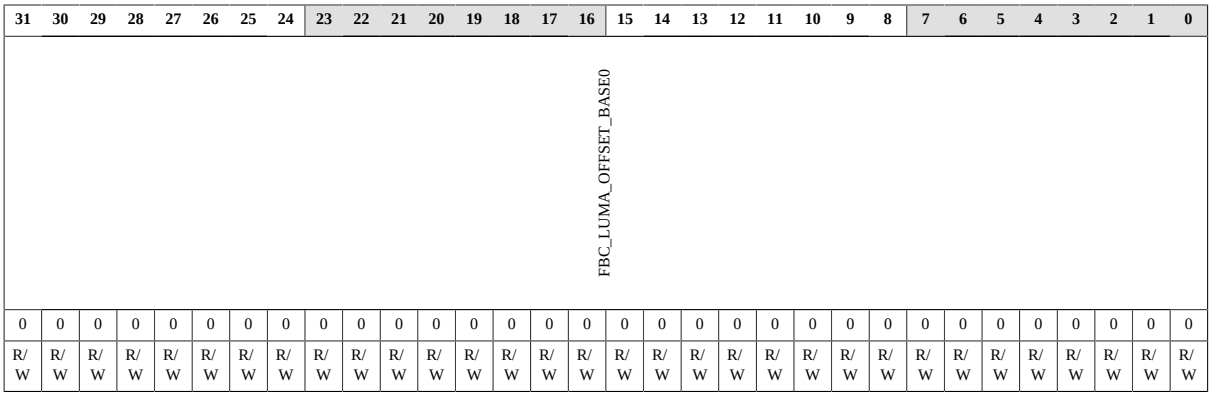


Table 1.488. ADDR_FBC_Y_OFFSET0 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_LUMA_OFFSET_BASE0	R/W	Compressed Luma offset table base address of DPB index 0 when FBC is used	0x0

Confidential
StarFive Inc.

1.2.3.2.6.20. ADDR_FBC_C_OFFSET0 (0x0000016C)

FBC Chroma offset base of index0

Table 1.489. ADDR_FBC_C_OFFSET0 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_CHROMA_OFFSET_BASE0																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.490. ADDR_FBC_C_OFFSET0 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_CHROMA_OFFSET_BASE0	R/W	If FBC is used, this is chroma offset base address of DPB index 0. Otherwise, it is ignored.	0x0

1.2.3.2.6.21. ADDR_LUMA_BASE1 (0x00000170)

Luma base of index1

Table 1.491. ADDR_LUMA_BASE1 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUMA_BASE1																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.492. ADDR_LUMA_BASE1 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LUMA_BASE1	R/W	Luma base address of DPB index 1 unless FBC is used. If FBC is used, this is compressed Luma base address of DPB index 1.	0x0

1.2.3.2.6.22. ADDR_CB_BASE1 (0x00000174)

Cb base of index1

Table 1.493. ADDR_CB_BASE1 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB_BASE1																															

Confidential
StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.494. ADDR_CB_BASE1 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CB_BASE1	R/W	Cb base address of DPB index 1 unless FBC is used. If FBC is used, this is compressed Cb and Cr base address of DPB index 1.	0x0

1.2.3.2.6.23. ADDR_CR_BASE1 (0x00000178)

Cr base of index1

Table 1.495. ADDR_CR_BASE1 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE1																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.496. ADDR_CR_BASE1 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE1	R/W	Cr base address of DPB index 1 unless FBC is used.	0x0

1.2.3.2.6.24. ADDR_FBC_Y_OFFSET1 (0x00000178)

FBC luma offset base of index1

Table 1.497. ADDR_FBC_Y_OFFSET1 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_LUMA_OFFSET_BASE1																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.498. ADDR_FBC_Y_OFFSET1 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_LUMA_OFFSET_BASE1	R/W	Compressed Luma offset table base address of DPB index 1 when FBC is used.	0x0

1.2.3.2.6.25. ADDR_FBC_C_OFFSET1 (0x0000017C)

FBC chroma offset base of index1

Confidential StarFive Inc.

Table 1.499. ADDR_FBC_C_OFFSET1 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_CHROMA_OFFSET_BASE1																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.500. ADDR_FBC_C_OFFSET1 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_CHROMA_OFFSET_BASE1	R/W	If FBC is used, this is chroma offset base address of DPB index 1. Otherwise, it is ignored.	0x0

1.2.3.2.6.26. ADDR_LUMA_BASE2 (0x00000180)

Luma base of index2

Table 1.501. ADDR_LUMA_BASE2 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUMA_BASE2																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.502. ADDR_LUMA_BASE2 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LUMA_BASE2	R/W	Luma base address of DPB index 2 unless FBC is used. If FBC is used, this is compressed Luma base address of DPB index 2.	0x0

1.2.3.2.6.27. ADDR_CB_BASE2 (0x00000184)

Cb base of index2

Table 1.503. ADDR_CB_BASE2 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB_BASE2																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.504. ADDR_CB_BASE2 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CB_BASE2	R/W	Cb base address of DPB index 2 unless FBC is used. If FBC is used, this is compressed Cb and Cr base address of DPB index 2.	0x0

1.2.3.2.6.28. ADDR_CR_BASE2 (0x00000188)

Cr base of index2

Table 1.505. ADDR_CR_BASE2 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE2																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.506. ADDR_CR_BASE2 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE2	R/W	Cr base address of DPB index 2 unless FBC is used.	0x0

1.2.3.2.6.29. ADDR_FBC_C_OFFSET2 (0x0000018C)

FBC chroma offset base of index2

Table 1.507. ADDR_FBC_C_OFFSET2 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_CHROMA_OFFSET_BASE2																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.508. ADDR_FBC_C_OFFSET2 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_CHROMA_OFFSET_BASE2	R/W	If FBC is used, this is chroma offset base address of DPB index 2. Otherwise, it is ignored.	0x0

1.2.3.2.6.30. ADDR_LUMA_BASE3 (0x00000190)

Luma base of index3

Table 1.509. ADDR_LUMA_BASE3 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Confidential
StarFive Inc.

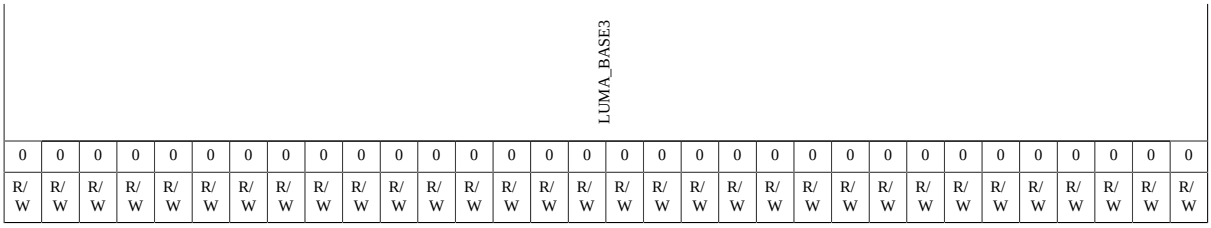


Table 1.510. ADDR_LUMA_BASE3 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LUMA_BASE3	R/W	Luma base address of DPB index 3 unless FBC is used. If FBC is used, this is compressed Luma base address of DPB index 3.	0x0

1.2.3.2.6.31. ADDR_CB_BASE3 (0x00000194)

Cb base of index3

Table 1.511. ADDR_CB_BASE3 Bit Assignment

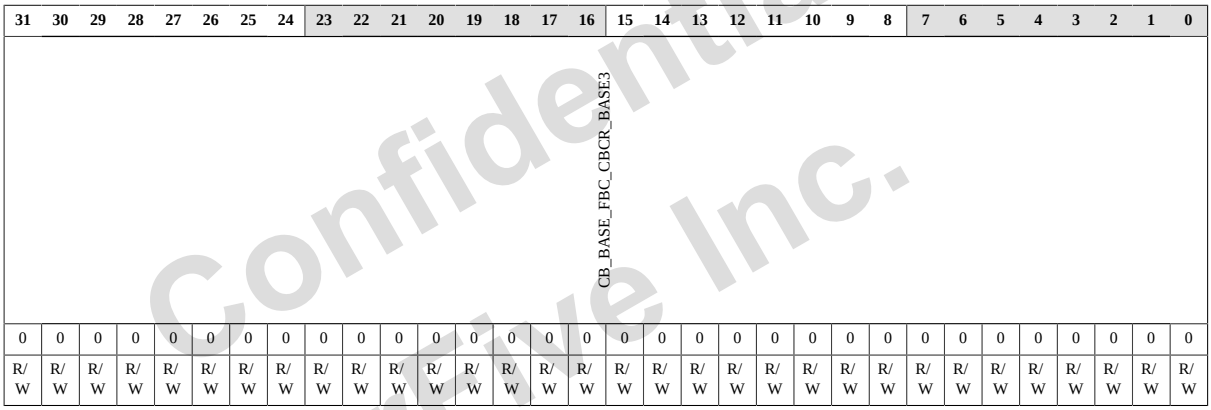


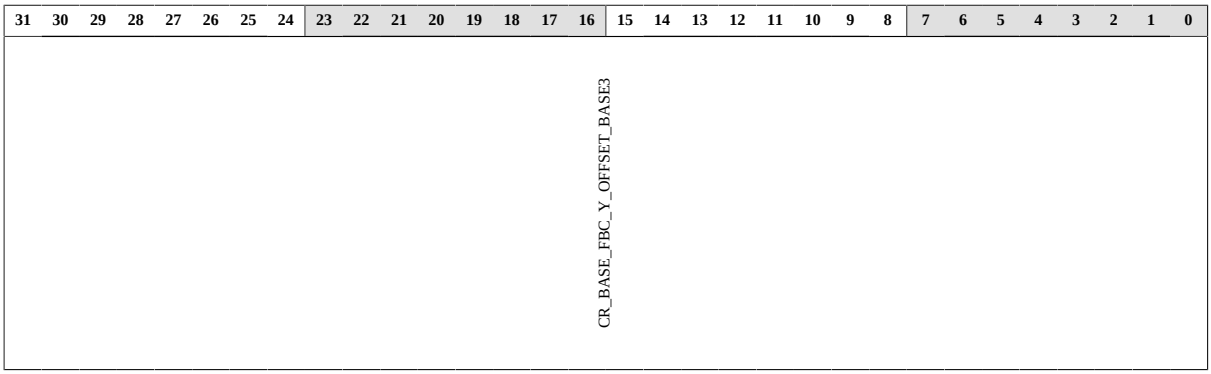
Table 1.512. ADDR_CB_BASE3 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CB_BASE_FBC_CBCR_BASE3	R/W	Cb base address of DPB index 3 unless FBC is used. If FBC is used, this is compressed Cb and Cr base address of DPB index 3.	0x0

1.2.3.2.6.32. ADDR_CR_BASE3 (0x00000198)

Cr base of index3

Table 1.513. ADDR_CR_BASE3 Bit Assignment



Confidential
Starfive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.514. ADDR_CR_BASE3 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE3	R/W	Cr base address of DPB index 3 unless FBC is used. If FBC is used, this is compressed Luma offset table base address of DPB index 3.	0x0

1.2.3.2.6.33. ADDR_FBC_Y_OFFSET3 (0x00000198)

FBC luma offset base of index3

Table 1.515. ADDR_FBC_Y_OFFSET3 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE_FBC_Y_OFFSET_BASE3																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential
StarFive Inc.

Table 1.516. ADDR_FBC_Y_OFFSET3 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE3	R/W	Cr base address of DPB index 3 unless FBC is used.	0x0

1.2.3.2.6.34. ADDR_FBC_C_OFFSET3 (0x0000019C)

FBC Chroma offset base of index3

Table 1.517. ADDR_FBC_C_OFFSET3 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_CHROMA_OFFSET_BASE3																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.518. ADDR_FBC_C_OFFSET3 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_CHROMA_OFFSET_BASE3	R/W	If FBC is used, this is chroma offset base address of DPB index 3. Otherwise, it is ignored.	0x0

1.2.3.2.6.35. ADDR_LUMA_BASE4 (0x000001A0)

Luma base of index4

Table 1.519. ADDR_LUMA_BASE4 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUMA_BASE4																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.520. ADDR_LUMA_BASE4 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LUMA_BASE4	R/W	Luma base address of DPB index 4 unless FBC is used. If FBC is used, this is compressed Luma base address of DPB index 4.	0x0

1.2.3.2.6.36. ADDR_CB_BASE4 (0x000001A4)

Cb base of index4

Table 1.521. ADDR_CB_BASE4 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB_BASE_FBC_CBCR_BASE4																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.522. ADDR_CB_BASE4 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CB_BASE_FBC_CBCR_BASE4	R/W	Cb base address of DPB index 4 unless FBC is used. If FBC is used, this is compressed Cb and Cr base address of DPB index 4.	0x0

1.2.3.2.6.37. ADDR_CR_BASE4 (0x000001A8)

Cr base of index4

Confidential StarFive Inc.

Table 1.523. ADDR_CR_BASE4 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE_FBC_Y_OFFSET_BASE4																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.524. ADDR_CR_BASE4 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE4	R/W	Cr base address of DPB index 4 unless FBC is used. If FBC is used, this is compressed Luma offset table base address of DPB index 4.	0x0

1.2.3.2.6.38. ADDR_FBC_Y_OFFSET4 (0x000001A8)

Cr base of index4

Table 1.525. ADDR_FBC_Y_OFFSET4 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE_FBC_Y_OFFSET_BASE4																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.526. ADDR_FBC_Y_OFFSET4 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE4	R/W	Cr base address of DPB index 4 unless FBC is used. If FBC is used, this is compressed Luma offset table base address of DPB index 4.	0x0

1.2.3.2.6.39. ADDR_FBC_C_OFFSET4 (0x000001AC)

FBC Chroma offset base of index4

Confidential StarFive Inc.

Table 1.527. ADDR_FBC_C_OFFSET4 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_CHROMA_OFFSET_BASE4																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.528. ADDR_FBC_C_OFFSET4 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_CHROMA_OFFSET_BASE4	R/W	If FBC is used, this is chroma offset base address of DPB index 4. Otherwise, it is ignored.	0x0

1.2.3.2.6.40. ADDR_LUMA_BASE5 (0x000001B0)

Luma base of index5

Table 1.529. ADDR_LUMA_BASE5 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUMA_BASE5																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.530. ADDR_LUMA_BASE5 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LUMA_BASE5	R/W	Luma base address of DPB index 5 unless FBC is used. If FBC is used, this is compressed Luma base address of DPB index 5.	0x0

1.2.3.2.6.41. ADDR_CB_BASE5 (0x000001B4)

Cb base of index5

Table 1.531. ADDR_CB_BASE5 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB_BASE_FBC_CBCR_BASE5																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.532. ADDR_CB_BASE5 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CB_BASE_FBC_CBCR_BASE5	R/W	Cb base address of DPB index 5 unless FBC is used. If FBC is used, this is compressed Cb and Cr base address of DPB index 5.	0x0

1.2.3.2.6.42. ADDR_CR_BASE5 (0x000001B8)

Cr base of index5

Table 1.533. ADDR_CR_BASE5 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE_FBC_Y_OFFSET_BASE5																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.534. ADDR_CR_BASE5 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE5	R/W	Cr base address of DPB index 5 unless FBC is used. If FBC is used, this is compressed Luma offset table base address of DPB index 5.	0x0

1.2.3.2.6.43. ADDR_FBC_Y_OFFSET5 (0x000001B8)

FBC luma offset base of index5

Table 1.535. ADDR_FBC_Y_OFFSET5 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE_FBC_Y_OFFSET_BASE5																															

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.536. ADDR_FBC_Y_OFFSET5 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE5	R/W	Cr base address of DPB index 5 unless FBC is used.	0x0

1.2.3.2.6.44. ADDR_FBC_C_OFFSET5 (0x000001BC)

FBC chroma offset base of index5

Table 1.537. ADDR_FBC_C_OFFSET5 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_CHROMA_OFFSET_BASE5																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential
Star Five Inc.

Table 1.538. ADDR_FBC_C_OFFSET5 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_CHROMA_OFFSET_BASE5	R/W	If FBC is used, this is chroma offset base address of DPB index 5. Otherwise, it is ignored.	0x0

1.2.3.2.6.45. ADDR_LUMA_BASE6 (0x000001C0)

Luma base of index6

Table 1.539. ADDR_LUMA_BASE6 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUMA_BASE6																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.540. ADDR_LUMA_BASE6 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LUMA_BASE6	R/W	Luma base address of DPB index 6 unless FBC is used. If FBC is used, this is compressed Luma base address of DPB index 6.	0x0

1.2.3.2.6.46. ADDR_CB_BASE6 (0x000001C4)

Cb base of index6

Table 1.541. ADDR_CB_BASE6 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB_BASE_FBC_CBCR_BASE6																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.542. ADDR_CB_BASE6 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CB_BASE_FBC_CBCR_BASE6	R/W	Cb base address of DPB index 6 unless FBC is used. If FBC is used, this is compressed Cb and Cr base address of DPB index 6.	0x0

1.2.3.2.6.47. ADDR_CR_BASE6 (0x000001C8)

Cr base of index6

Table 1.543. ADDR_CR_BASE6 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE_FBC_Y_OFFSET_BASE6																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.544. ADDR_CR_BASE6 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE6	R/W	Cr base address of DPB index 6 unless FBC is used.	0x0

1.2.3.2.6.48. ADDR_FBC_Y_OFFSET6 (0x000001C8)

FBC luma offset base of index6

Confidential StarFive Inc.

Table 1.545. ADDR_FBC_Y_OFFSET6 Bit Assignment

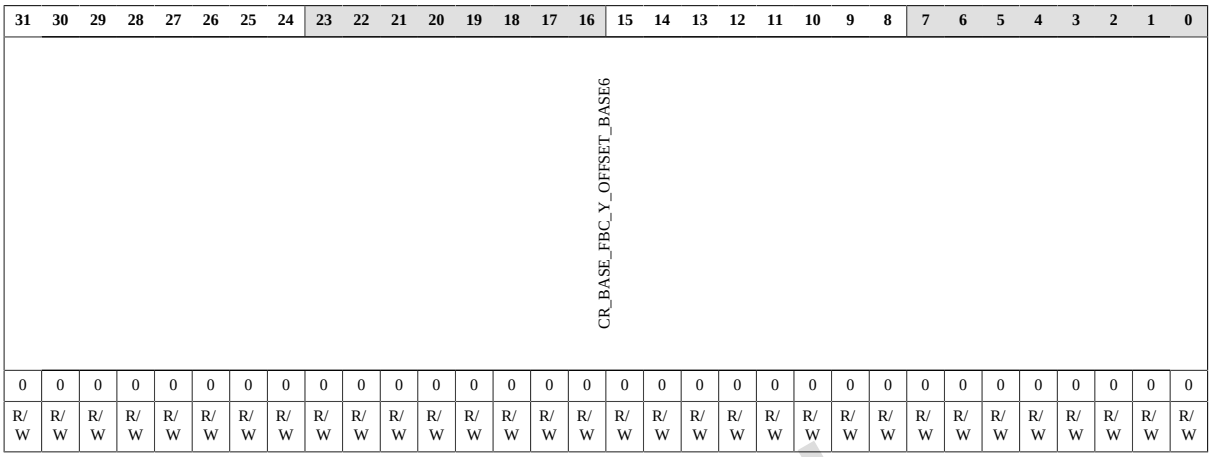


Table 1.546. ADDR_FBC_Y_OFFSET6 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE6	R/W	Compressed Luma offset table base address of DPB index 6 when FBC is used.	0x0

1.2.3.2.6.49. ADDR_FBC_C_OFFSET6 (0x000001CC)

FBC chroma offset base of index6

Table 1.547. ADDR_FBC_C_OFFSET6 Bit Assignment

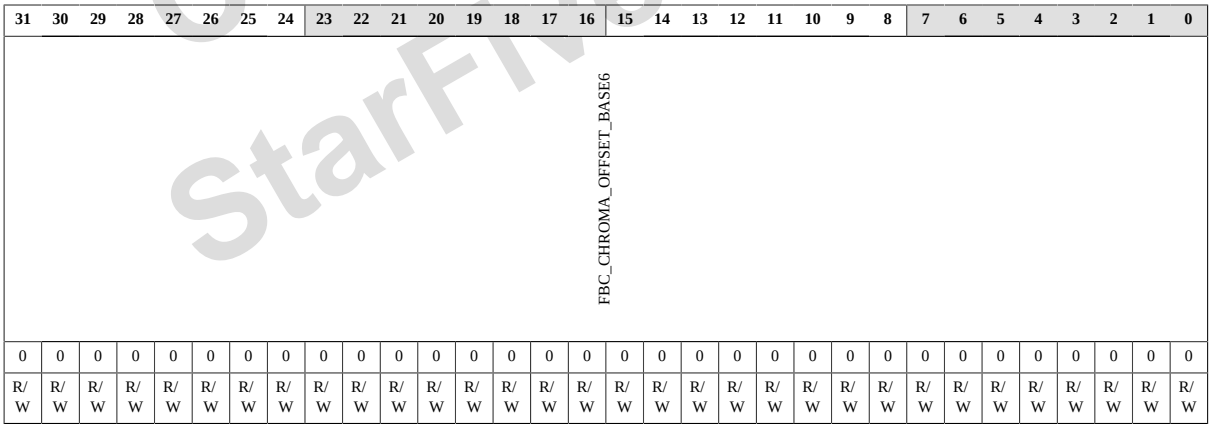


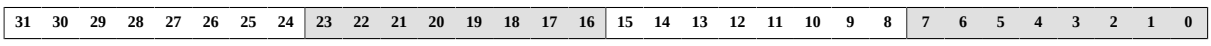
Table 1.548. ADDR_FBC_C_OFFSET6 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_CHROMA_OFFSET_BASE6	R/W	If FBC is used, this is chroma offset base address of DPB index 6. Otherwise, it is ignored.	0x0

1.2.3.2.6.50. ADDR_LUMA_BASE7 (0x000001D0)

Luma base of index7

Table 1.549. ADDR_LUMA_BASE7 Bit Assignment



Confidential StarFive Inc.

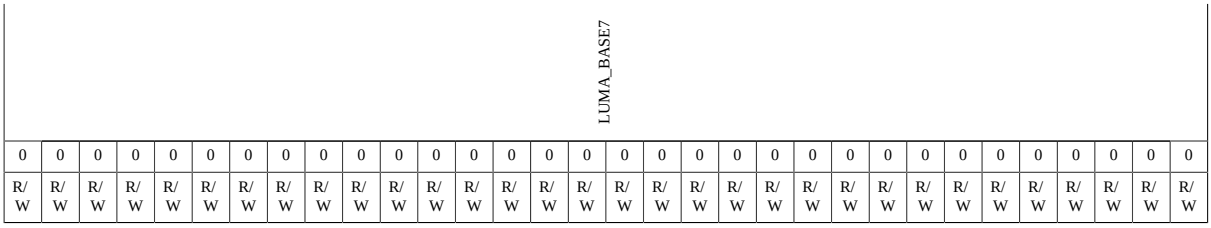


Table 1.550. ADDR_LUMA_BASE7 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	LUMA_BASE7	R/W	Luma base address of DPB index 7 unless FBC is used. If FBC is used, this is compressed Luma base address of DPB index 7.	0x0

1.2.3.2.6.51. ADDR_CB_BASE7 (0x000001D4)

Cb base of index7

Table 1.551. ADDR_CB_BASE7 Bit Assignment

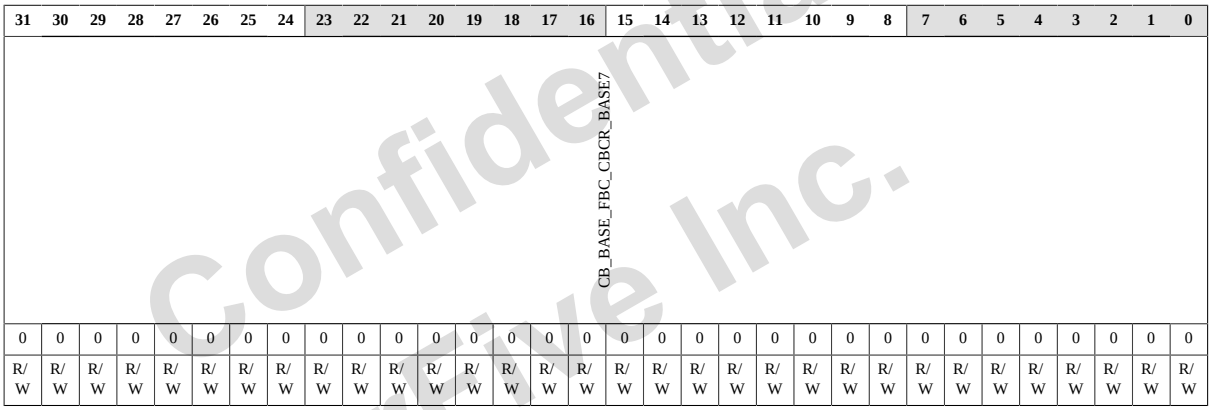


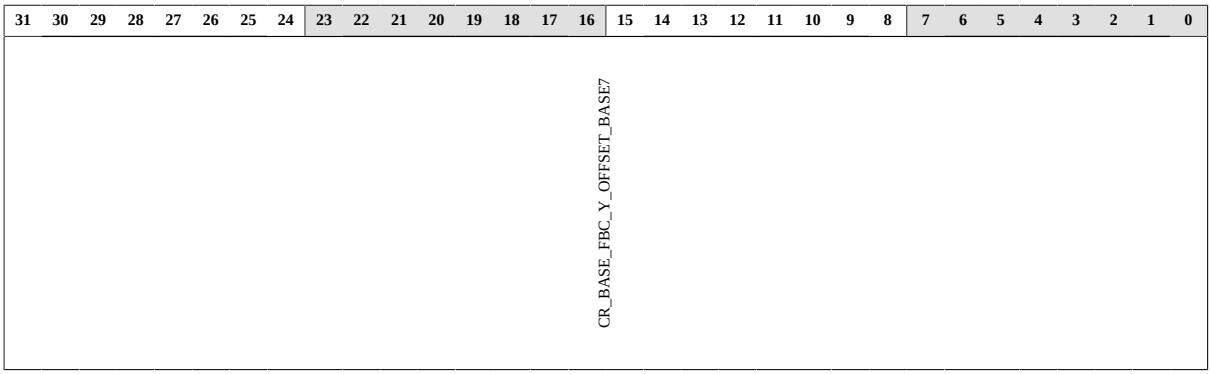
Table 1.552. ADDR_CB_BASE7 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CB_BASE_FBC_CBCR_BASE7	R/W	Cb base address of DPB index 7 unless FBC is used. If FBC is used, this is compressed Cb and Cr base address of DPB index 7.	0x0

1.2.3.2.6.52. ADDR_CR_BASE7 (0x000001D8)

Cr base of index7

Table 1.553. ADDR_CR_BASE7 Bit Assignment



Confidential
Starfive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.554. ADDR_CR_BASE7 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE7	R/W	Cr base address of DPB index 7 unless FBC is used. If FBC is used, this is compressed Luma offset table base address of DPB index 7.	0x0

1.2.3.2.6.53. ADDR_FBC_Y_OFFSET7 (0x000001D8)

FBC luma offset base of index7

Table 1.555. ADDR_FBC_Y_OFFSET7 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR_BASE_FBC_Y_OFFSET_BASE7																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.556. ADDR_FBC_Y_OFFSET7 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CR_BASE_FBC_Y_OFFSET_BASE7	R/W	Compressed Luma offset table base address of DPB index 7 when FBC is used.	0x0

1.2.3.2.6.54. ADDR_FBC_C_OFFSET7 (0x000001DC)

FBC chroma offset base of index7

Table 1.557. ADDR_FBC_C_OFFSET7 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBC_CHROMA_OFFSET_BASE7																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.558. ADDR_FBC_C_OFFSET7 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FBC_CHROMA_OFFSET_BASE7	R/W	If FBC is used, this is chroma offset base address of DPB index 7. Otherwise, it is ignored.	0x0

1.2.3.2.6.55. ADDR_MV_COL0 (0x000001E0)

Colocated mv buffer base of index 0

Table 1.559. ADDR_MV_COL0 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COL_MV_BUF_BASE0																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.560. ADDR_MV_COL0 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COL_MV_BUF_BASE0	R/W	Base address of colocated motion vecotors buffer of DPB index 0	0x0

1.2.3.2.6.56. ADDR_MV_COL1 (0x000001E4)

Colocated mv buffer base of index 1

Table 1.561. ADDR_MV_COL1 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COL_MV_BUF_BASE1																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.562. ADDR_MV_COL1 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COL_MV_BUF_BASE1	R/W	Base address of colocated motion vecotors buffer of DPB index 1	0x0

1.2.3.2.6.57. ADDR_MV_COL2 (0x000001E8)

Colocated mv buffer base of index 2

Confidential StarFive Inc.

Table 1.563. ADDR_MV_COL2 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COL_MV_BUF_BASE2																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.564. ADDR_MV_COL2 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COL_MV_BUF_BASE2	R/W	Base address of colocated motion vecotors buffer of DPB index 2	0x0

1.2.3.2.6.58. ADDR_MV_COL3 (0x000001EC)

Colocated mv buffer base of index 3

Table 1.565. ADDR_MV_COL3 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COL_MV_BUF_BASE3																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.566. ADDR_MV_COL3 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COL_MV_BUF_BASE3	R/W	Base address of colocated motion vecotors buffer of DPB index 3	0x0

1.2.3.2.6.59. ADDR_MV_COL4 (0x000001F0)

Colocated mv buffer base of index 4

Table 1.567. ADDR_MV_COL4 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COL_MV_BUF_BASE4																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.568. ADDR_MV_COL4 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COL_MV_BUF_BASE4	R/W	Base address of colocated motion vecotors buffer of DPB index 4	0x0

1.2.3.2.6.60. ADDR_MV_COL5 (0x000001F4)

Colocated mv buffer base of index 5

Table 1.569. ADDR_MV_COL5 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COL_MV_BUF_BASE5																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.570. ADDR_MV_COL5 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COL_MV_BUF_BASE5	R/W	Base address of colocated motion vecotors buffer of DPB index 5	0x0

1.2.3.2.6.61. ADDR_MV_COL6 (0x000001F8)

Colocated mv buffer base of index 6

Table 1.571. ADDR_MV_COL6 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COL_MV_BUF_BASE6																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.572. ADDR_MV_COL6 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COL_MV_BUF_BASE6	R/W	Base address of colocated motion vecotors buffer of DPB index 6	0x0

Confidential Starlive Inc.

1.2.3.2.6.62. ADDR_MV_COL7 (0x000001FC)

Colocated mv buffer base of index 7

Table 1.573. ADDR_MV_COL7 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COL_MV_BUF_BASE7																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.574. ADDR_MV_COL7 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COL_MV_BUF_BASE7	R/W	Base address of colocated motion vecotors buffer of DPB index 7	0x0

Confidential StarFive Inc.

1.2.3.2.7. GET_FW_VERSION Command Parameter Registers

These are command I/O registers where host processor can set arguments for the GET_FW_VERSION command or get return values.

1.2.3.2.7.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.575. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.576. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			• 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.7.2. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.577. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.578. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.7.3. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.579. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.580. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.7.4. RET_FW_VERSION (0x00000118)

Table 1.581. RET_FW_VERSION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RET_FW_VERSION																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 1.582. RET_FW_VERSION Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RET_FW_VERSION	R	Firmware version (internal use only)	0x0

1.2.3.2.7.5. RET_PRODUCT_NAME (0x0000011C)

Table 1.583. RET_PRODUCT_NAME Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												HW_NAME			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R	R	R	R

Table 1.584. RET_PRODUCT_NAME Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	HW_NAME	R	VPU hardware product name always returns "WAVE" for WAVE product series.	0x0

1.2.3.2.7.6. RET_PRODUCT_VERSION (0x00000120)

Table 1.585. RET_PRODUCT_VERSION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												HW_VERSION			

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R	R	R	R

Table 1.586. RET_PRODUCT_VERSION Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	HW_VERSION	R	VPU hardware product version always returns "4200" for WAVE420 product.	0x0

1.2.3.2.7.7. RET_STD_DEF0 (0x00000124)

Table 1.587. RET_STD_DEF0 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 1.588. RET_STD_DEF0 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RSVD	R	Configuration Information #0 (internal use only)	0x0

1.2.3.2.7.8. RET_STD_DEF1 (0x00000128)

Table 1.589. RET_STD_DEF1 Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 1.590. RET_STD_DEF1 Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RSVD	R	Configuration Information #1 (internal use only)	0x0

1.2.3.2.7.9. RET_CODEC_STD (0x0000012C)

Table 1.591. RET_CODEC_STD Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODEC_STD																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 1.592. RET_CODEC_STD Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CODEC_STD	R	General Configuration Information (internal use only)	0x0

Confidential StarFive Inc.

1.2.3.2.7.10. RET_CONF_DATE (0x00000130)

Table 1.593. RET_CONF_DATE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HW_DATE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 1.594. RET_CONF_DATE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	HW_DATE	R	The date that the hardware has been configured (YYYYmmdd in digit)	0x0

1.2.3.2.7.11. RET_CONF_REVISION (0x00000134)

Table 1.595. RET_CONF_REVISION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HW_REVISION																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 1.596. RET_CONF_REVISION Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	HW_REVISION	R	The revision number when the hardware has been configured	0x0

1.2.3.2.7.12. RET_CONF_TYPE (0x00000138)

Table 1.597. RET_CONF_TYPE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HW_TYPE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 1.598. RET_CONF_TYPE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	HW_TYPE	R	The define value used in hardware configuration (Reserved)	0x0

Confidential
Starfive Inc.

1.2.3.2.8. SLEEP_VPU Command Parameter Registers

These are command I/O registers where host processor can set arguments for the SLEEP_VPU command or get return values.

1.2.3.2.8.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.599. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.600. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			• 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.8.2. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.601. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.602. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.8.3. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.603. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.604. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.8.4. RET_CUR_SP (0x00000124)

Current Stack Pointer

Table 1.605. RET_CUR_SP Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RET_CUR_SP																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.606. RET_CUR_SP Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	RET_CUR_SP	R/W	Current stack pointer to backup When VPU power is going off by SLEEP_VPU command, the stack pointer should be backed up. This register returns the current supervisor stack pointer for backing up.	0x0

Confidential StarFive Inc.

1.2.3.2.9. WAKEUP_VPU Command Parameter Registers

These are command I/O registers where host processor can set arguments for the WAKEUP_VPU command or get return values.

1.2.3.2.9.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.607. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.608. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			• 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.9.2. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.609. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.610. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.9.3. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.611. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															

Confidential StarFive Inc.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.612. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.9.4. ADDR_CODE_BASE (0x00000118)

Code Buffer Base Address

Table 1.613. ADDR_CODE_BASE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE_BUF_BASE																				RSVD											
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Table 1.614. ADDR_CODE_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:12]	CODE_BUF_BASE	R/W	Base address of V-CPU micro code buffer It should be aligned to 4KB range.	0x0
[11:0]	RSVD	RO	Reserved	0x0

1.2.3.2.9.5. CODE_SIZE (0x0000011C)

Code Buffer Size

Table 1.615. CODE_SIZE Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE_BUF_SIZE																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.616. CODE_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CODE_BUF_SIZE	R/W	Size of CODE buffer It should be aligned to 4KB range.	0x0

1.2.3.2.9.6. CODE_PARAM (0x00000120)

Parameter for code buffer

Confidential Starfive Inc.

Table 1.617. CODE_PARAM Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								CODE_AXIID				CODE_ENDIAN			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.618. CODE_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:8]	RSVD	RO	Reserved	0x0
[7:4]	CODE_AXIID	R/W	AXI-ID for the V-CPU part (for virtualization)	0x0
[3:0]	CODE_ENDIAN	R/W	Endianness	0x0

1.2.3.2.9.7. CUR_SP (0x00000124)

Current Stack Pointer

Table 1.619. CUR_SP Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUR_SP																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.620. CUR_SP Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	CUR_SP	R/W	It restores the stack pointer back to the previous one that was backed-up. When VPU resumes by WAKEUP_VPU command, the stack pointer should be restored.	0x0

Confidential StarFive Inc.

1.2.3.2.10. CREATE_INST Command Parameter Registers

These are command I/O registers where host processor can set arguments for the CREATE_INST command or get return values.

1.2.3.2.10.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.621. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.622. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			• 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.10.2. CORE_INDEX (0x00000104)

V-CORE Index
This is a common parameter register for almost all commands, except VPU control commands such as INIT_VPU, GET_FW_VERSION, SLEEP_VPU, and WAKEUP_VPU.

Table 1.623. CORE_INDEX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VCORE_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.624. CORE_INDEX Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	VCORE_INDEX	RW	V-CORE index - reserved for future use, so please set this as 0.	0x0

1.2.3.2.10.3. INST_INDEX_COD_STD (0x00000108)

Codec Standard and Instance Index
This is a common parameter register for almost all commands (Reserved for future use)

Table 1.625. INST_INDEX_COD_STD Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODEC_STD																INST_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confidential
StarFive Inc.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.626. INST_INDEX_COD_STD Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CODEC_STD	R/W	Codec standard to run HEVC_DEC = 0x0 HEVC_ENC = 0x1	0x0
[15:0]	INST_INDEX	R/W	Instance Index VPU can decode more than one decoding instance simultaneously. If multiple instances are running, each instance must have their own process index that is assigned by this register. For example, two instances are running simultaneously, the first instance has the process index 0, the second instance has the process index 1. Instance index shall be in the range of 0 to 65535, inclusive.	0x0

1.2.3.2.10.4. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.627. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.628. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.10.5. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.629. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.630. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

Confidential
StarFive Inc.

1.2.3.2.11. CHANGE_INST Command Parameter Registers

These are command I/O registers where host processor can set arguments for the CHANGE_INST command or get return values.

1.2.3.2.11.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.631. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.632. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			• 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.11.2. CORE_INDEX (0x00000104)

V-CORE Index
This is a common parameter register for almost all commands, except VPU control commands such as INIT_VPU, GET_FW_VERSION, SLEEP_VPU, and WAKEUP_VPU.

Table 1.633. CORE_INDEX Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VCORE_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.634. CORE_INDEX Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	RO	Reserved	0x0
[15:0]	VCORE_INDEX	RW	V-CORE index - reserved for future use, so please set this as 0.	0x0

1.2.3.2.11.3. INST_INDEX_COD_STD (0x00000108)

Codec Standard and Instance Index
This is a common parameter register for almost all commands (Reserved for future use)

Table 1.635. INST_INDEX_COD_STD Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODEC_STD																INST_INDEX															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Confidential
StarFive Inc.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.636. INST_INDEX_COD_STD Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	CODEC_STD	R/W	Codec standard to run HEVC_DEC = 0x0 HEVC_ENC = 0x1	0x0
[15:0]	INST_INDEX	R/W	Instance Index VPU can decode more than one decoding instance simultaneously. If multiple instances are running, each instance must have their own process index that is assigned by this register. For example, two instances are running simultaneously, the first instance has the process index 0, the second instance has the process index 1. Instance index shall be in the range of 0 to 65535, inclusive.	0x0

1.2.3.2.11.4. RET_SUCCESS (0x00000110)

Result of the run command

Table 1.637. RET_SUCCESS Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														RUN_CMD_STATUS	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W

Table 1.638. RET_SUCCESS Field Description

Bit	Name	Type	Function	Reset Value
[31:2]	RSVD	RO	Reserved	0x0
[1:0]	RUN_CMD_STATUS	R/W	00: FAIL 01: SUCCESS 10: SUCCESS_WITH_WARNIG	0x0

1.2.3.2.11.5. RET_FAIL_REASON (0x00000114)

Fail reason of the run command

Table 1.639. RET_FAIL_REASON Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAIL_REASON																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Confidential StarFive Inc.

Table 1.640. RET_FAIL_REASON Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	FAIL_REASON	R/W	Please refer to *Section A.1, "System Error"* defining errors that VPU might have.	0x0

1.2.3.2.11.6. ADDR_WORK_BASE (0x00000138)

Work Buffer Base Address

Table 1.641. ADDR_WORK_BASE Bit Assignment

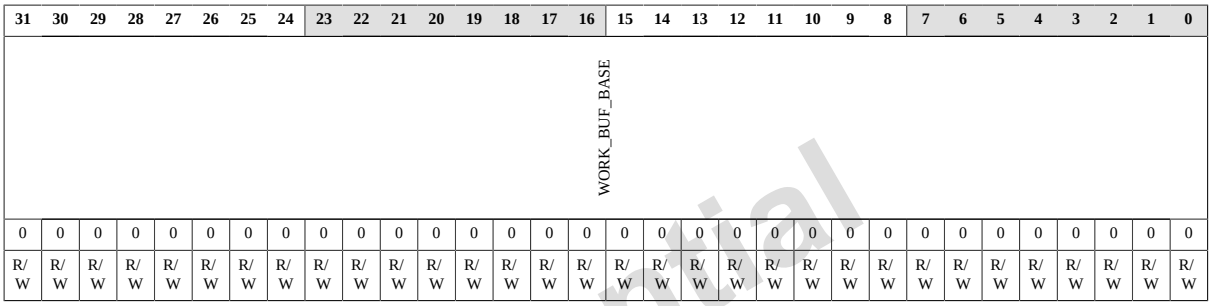


Table 1.642. ADDR_WORK_BASE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WORK_BUF_BASE	R/W	Base address of work buffer for each instance This address should be aligned in 4K boundary.	0x0

1.2.3.2.11.7. WORK_SIZE (0x0000013C)

Work Buffer Size

Table 1.643. WORK_SIZE Bit Assignment

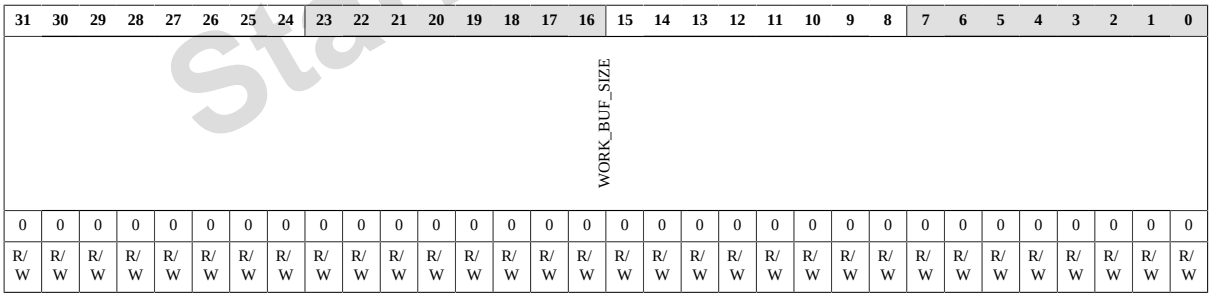


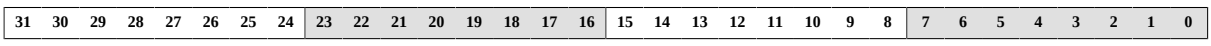
Table 1.644. WORK_SIZE Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WORK_BUF_SIZE	R/W	Size of work buffer (4K boundary) The size of work buffer should be larger than required minimum work buffer size. This address should be aligned in 4K boundary.	0x0

1.2.3.2.11.8. WORK_PARAM (0x00000140)

Work Buffer Parameter

Table 1.645. WORK_PARAM Bit Assignment



Confidential StarFive Inc.

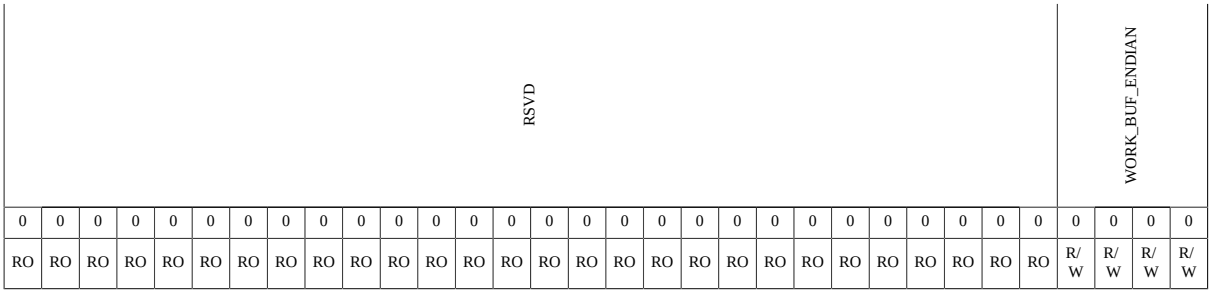


Table 1.646. WORK_PARAM Field Description

Bit	Name	Type	Function	Reset Value
[31:4]	RSVD	RO	Reserved	0x0
[3:0]	WORK_BUF_ENDIAN	R/W	Endianness of work buffer Not implemented yet - VPU uses work buffer as internal purpose so endianness is not important.	0x0

Confidential
StarFive Inc.

1.2.3.2.12. UPDATE_BS Command Parameter Registers

These are command I/O registers where host processor can set arguments for the UPDATE_BS command or get return values.

1.2.3.2.12.1. COMMAND (0x00000100)

Command to run VPU
Depending on the value of command, host interface registers from 0x104 to 0x1fc might mean different things.

Table 1.647. COMMAND Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMAND																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 1.648. COMMAND Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	COMMAND	RW	• 0x0001 : INIT_VPU Boot up vCPU • 0x0002 : SET_PARAM / DEC_PIC_HDR Encode/decode sequence initialize. In encode case, VPU analysis encoding parameter and encode sequence header. In decode case, VPU decode sequence header and report sequence header information. • 0x0004 : FINI_SEQ Terminate encoding/decoding of video sequence. • 0x0008 : ENC_PIC / DEC_PIC Encode/decode one picture. VPU encodes/decodes one picture. • 0x0010 : SET_FRAMEBUF Set decoded/reconstructed frame buffer SDRAM address and maximum frame buffer number. Before decode picture run command, host must inform framebuffer SDRAM address to VPU then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of decoding picture. • 0x0020 : FLUSH_DECODER Return framebuffer indexes remaining in DPB in display order and initialize the DPB. This command is used when a sequence changes or random access to a sequence is required in the middle of decoding. • 0x0100 : GET_FW_VERSION Return thre revision information of the firmware in 32bit integer. • 0x0200 : QUERY_DECODER Reserved for future	0x0

Confidential StarFive Inc.

Bit	Name	Type	Function	Reset Value
			• 0x0400 : SLEEP_VPU Save context (all internal variables, which are needed for the recovery) to the working buffer. • 0x0800 : WAKEUP_VPU Load the context from the working buffer which have been saved after decoding the previous picture. • 0x1000 : CHANGE_INST Reserved for future • 0x4000 : CREATE_INSTANCE Have VPU allocate an instance and initialize internal data for the instance. • 0x8000 : UPDATE_BS Reserved for future	

1.2.3.2.12.2. BS_OPTION (0x0000012C)

Table 1.649. BS_OPTION Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																PERIODIC_EMPTY_INT_CNT								RSVD					PERIODIC_EMPTY_INT_EN	STREAM_END	EXPLICIT_END
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W

Table 1.650. BS_OPTION Field Description

Bit	Name	Type	Function	Reset Value
[31:16]	RSVD	R/W	Reserved	0x0
[15:8]	PERIODIC_EMPTY_INT_CNT	R/W	Reserved for controlling periodic emtpy interrupt (please set as 0)	0x0
[7:3]	RSVD	RO	Reserved	0x0
[2]	PERIODIC_EMPTY_INT_EN	R/W	Reserved for controlling periodic emtpy interrupt (please set as 0)	0x0
[1]	STREAM_END	R/W	This field makes VPU assumes Stream End when there is no stream to feed in the buffer.	0x0
[0]	EXPLICIT_END	R/W	Explicit End When this field is set to 1, VPU assumes that bitstream buffer has one NAL unit or more for decoding a signle frame - it works based on the BS_NAL_LEVEL_PUMP register on. Then it follows the tasks below.	0x0

Confidential
StarFive Inc.

Bit	Name	Type	Function	Reset Value
			1. VPU decodes until the end of bitstream buffer (WR_PTR) even if the last 3 bytes are all 0. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU performs what it is supposed to do with the specified task in BS_SHORTAGE_OPTION of BS_PARAM. If this flag is 0, 1. VPU decodes to the almost end of bitstream buffer (WR_PTR), but not to some bytes (less than 3). It intentionally does not comsume some a few bytes, because VPU is not sure if the last bytes are the start code of next NAL or they might not fill the offset in the CABAC. 2. VPU returns success if it has decoded a frame successfuly. If bitstream is insufficient to complete decoding a frame, VPU stops decoding and waits for more bitstream to be filled. Then host processor can do either feed bitstream more or set EXPLICIT_END to complete decoding anyhow. Bitstream_empty interrupt is asserted when EXPLITCIT_END = 0 or when bitstream buffer is near empty (not real empty) for seamless decoding. **Caution** Host processor can set this register any time, but cannot clear during command processing.	

1.2.3.2.12.3. BS_WR_PTR (0x00000134)

Bistream Buffer Write Pointer

Table 1.651. BS_WR_PTR Bit Assignment

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WR_PTR																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 1.652. BS_WR_PTR Field Description

Bit	Name	Type	Function	Reset Value
[31:0]	WR_PTR	R/W	End address of bitstream for handling current command Host can update this register anytime. If a bitstream_empty interrupt is asserted, host processor should do either feed more bitstream and update WR_PTR or set EXPLICIT_END to complete decoding anyhow.	0x0

Confidential StarFive Inc.

Chapter 2
APPLICATION INTERFACE

2.1. VPU API-based Control Mechanism

Host applications can control VPU at an API level through pre-defined API set. They can send a command with arguments, receive an interrupt indicating a requested operation has completed, or get a result of the command by using API functions as shown in *Figure 2.1, "SW control model of VPU from host application"*.

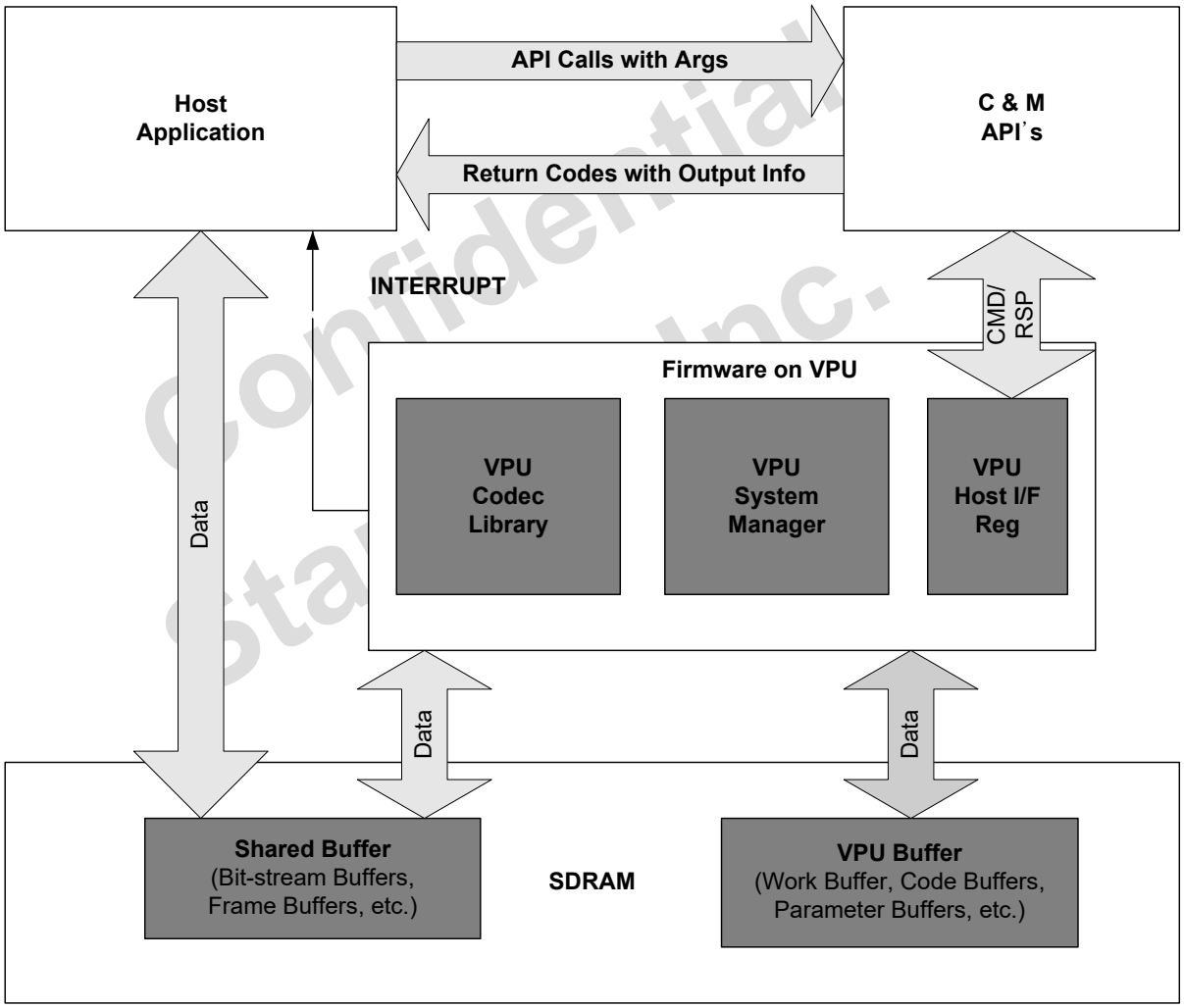


Figure 2.1. SW control model of VPU from host application

Each API definition includes the requested command as well as input and output data structure. The given command from API function is always written on a dedicated I/O register, and the input and output data structure are transmitted on a set of command I/O registers which contain input arguments and output results. So application programmers do not need to struggle with learning many of the host interface registers and their usage.

2.2. VPU API Reference Software

We provide customers with API reference software which is an implementation of codec application using VPU API functions. It helps application programmers develop their video applications by simply referring to or by porting it to fit their target CPU and OS.

There are five hierarchical layers in VPU API reference software, and *Figure 2.2, "API Reference Software Architecture"* shows the architectural layers of VPU API reference software.

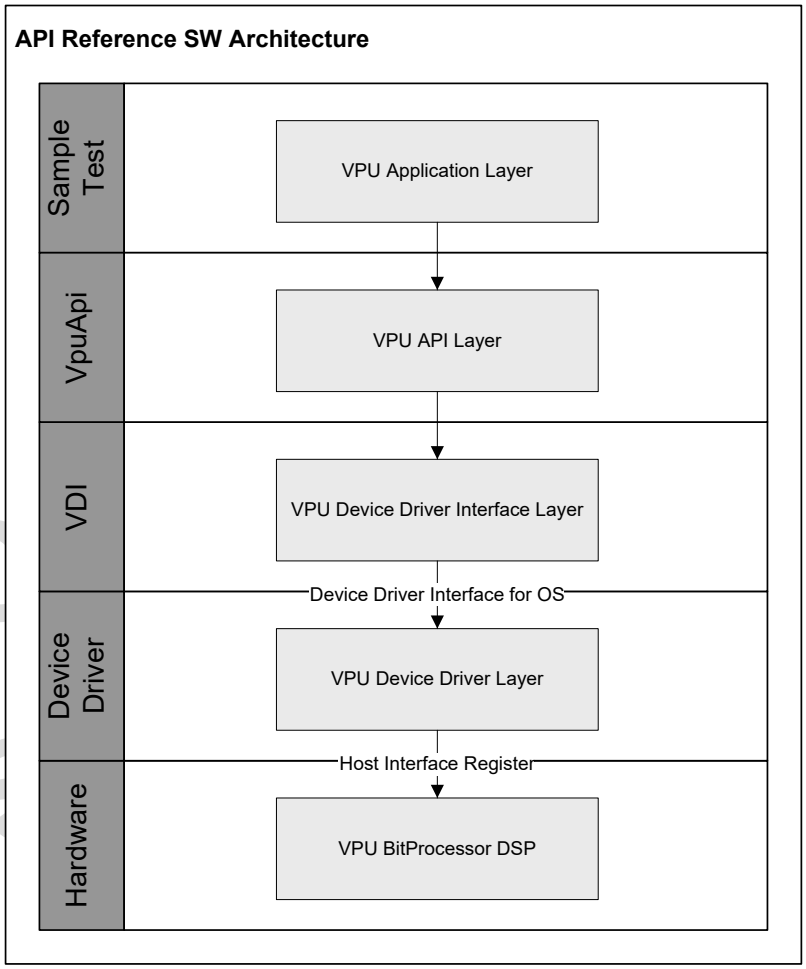


Figure 2.2. API Reference Software Architecture

- The VPU Application Layer is a kind of sample application stuff for decoding bitstream (packetized in general containers) and encoding image data by calling the VPU APIs.
- The VPU API Layer is the application-interface software stack to communicate with VPU hardware architecture. It can work on various OS platforms through VPU Device Driver Interface (VDI) that is able to get OS function calls.
- The VDI layer has some OS dependent codes for each OS that we support, and if there is a kernel mode in the OS, the VDI layer can communicate with the Device Driver Layer. Current API Reference Software implements three different types of VDI and their device drivers for Linux and NonOS and has a plan for extension and support for other OS.

2.2.1. Source Tree

Figure 2.3, "Source Tree of VPU API Reference Software" shows the source tree structure of the reference software package that we release.

```
<Root>/
    ├── cfg                 ; encoder configuration files (.cfg)
    ├── cmd                 ; configuration file list
    ├── ffmpeg              ; FFMPEG parser library for Linux and Windows
    ├── mmf
    │   ├── hmft
    │   └── openmax
    ├── prebuilt            ; encoded bitstreams from reference c-model
    ├── sample              ; Application c sources & helper
    ├── theoraparser        ; Theora VLD parser module
    ├── vdi
    │   ├── linux
    │   ├── nonos
    │   └── windows
    ├── vpuapi              ; VPU API C and headers
    │   ├── coda9           ; sources for product CODA960, CODA980, WAVE320
    │   └── wave4           ; sources for product WAVE410
    └── yuv                 ; encoding source YUV
```

Figure 2.3. Source Tree of VPU API Reference Software

2.2.2. Architecture

2.2.2.1. Application Layer

As a topmost layer of reference software, the VPU Application Layer has three functionalities. First, it contains video control sequences for decoding and displaying with given arguments. Second, it has a helper module that reads and writes a result from decoding and/or encoding and that interfaces with hardware resources through VDI module. And lastly, it also has user interface functions to communicate with application user through console menu.

To support these functions, the VPU Application Layer consists of Sample Test module, Parser module, VPUHelper module, and Mixer Module. *Figure 2.4, "Application Layer"* shows the modules of the Application Layer.

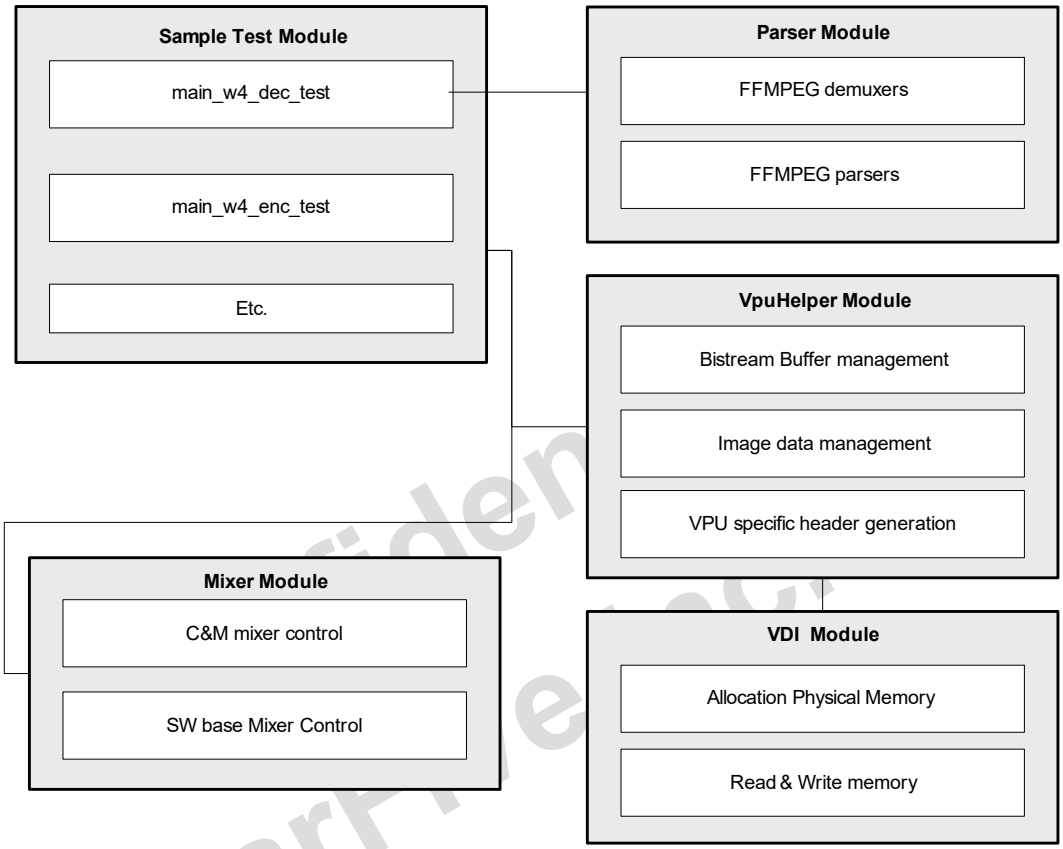


Figure 2.4. Application Layer

The relevant codes are in the `sample` directory , and they have simple functions such as Init, Open, SeqInit, Decode/Encode, Close, Deinit to control VPU hardware using VPU APIs, and also have system related codes that allocate decoder PP frame buffers, encoder source frame buffers, and bitstream buffer like user dependent memory. There is neither platform dependent code nor porting layer.

The following shortly describes responsibilities of each module and related files

- Sample Test module: main function in main_w4_enc_test.c or main_w4_dec_test.c
  - Start entry point
  - argc, argv parameter base
  - hevc_dec_test function with elementary stream
  - hevc_enc_test function with encoder option cfg file and user input encode option
  - MultiInstanceTest function: Multiple instance testing with multiple tasks that have different codec standards.
  - Calling VPU APIs and VDI to handle VPU
- **Parser module: ffmpeg open source library**

– To make API reference software capable of extracting elementary stream from various multimedia containers.

- **VpuHelper module: sample/helper directory**
  – Helper functions for video codec interface
  – Bitstream directory: A code that reads and writes bitstream from/to the memory.
  – comparator: A function that compares result data between c-model and FPGA/SoC conformance test
  – display: display module
  – misc: Other miscellaneous codes such as getopt for FPGA setting and md5
  – yuv: Codes that read and write YUV data from/to the memory
- **Mixer module: mixer.c**
  – Handle Chips&Media display module
  – Software based display support, including a yuv2rgb converter for display OS frame buffer

2.2.2.2. API Layer

The VPU API Layer is responsible for access to VPU hardware registers and direct control. The task of Sample Test module from the upper Application Layer calls the VPU API functions in this API layer to control the VPU Hardware. There is neither platform dependent code nor porting layer.

Figure 2.5, "VPU API Layer" illustrates the modules of the VPU API Layer, hierarchy, and interactions.

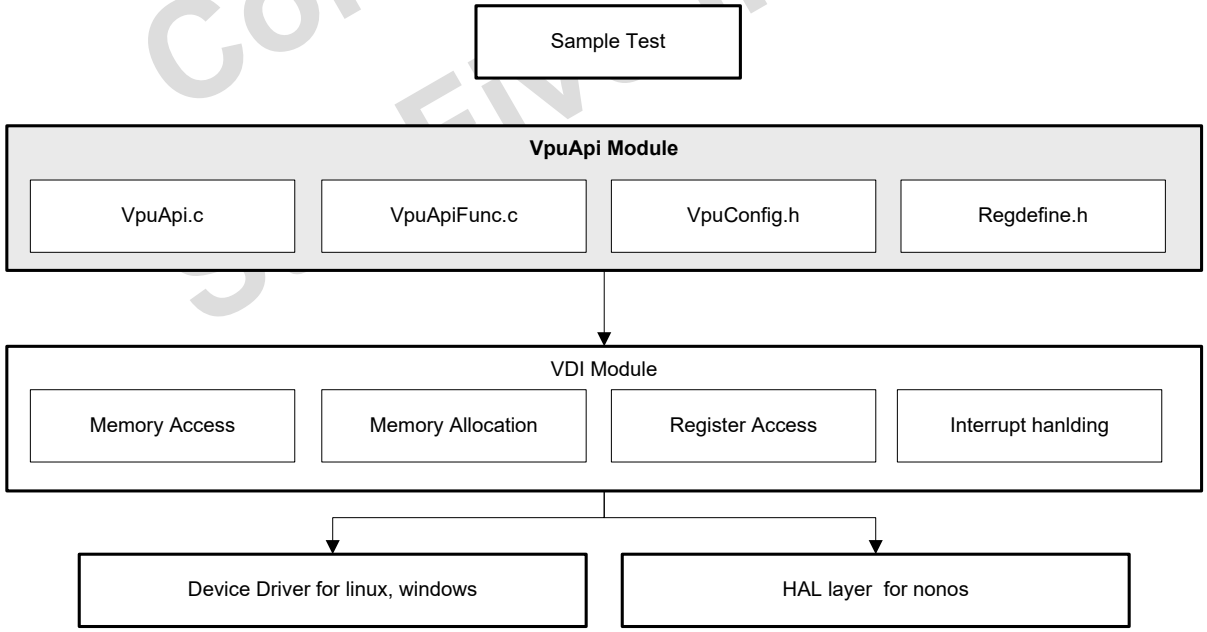


Figure 2.5. VPU API Layer

The following shortly describes responsibilities of each module and related files.

- **Main API module: VpuApi.c**
  – Decoder API function bodies to control the VPU Hardware and instance handle

– All the codec standard implementations share Host Interface on VPU hardware, so that these API functions can have simple architecture.

– To support multiple instances, the opened instance handle pool resides under VpuApiFunc.c.

– Some API functions access VPU Hardware registers directly through VDI functions.

– Some API functions access VDI functions for allocating codec dependent memory and waiting for VPU interrupt

– Some API functions jump to their sub-functions under VpuApiFunc.c to access the VPU hardware registers step by step.

2.2.2.3. VDI Layer

The VPU device driver interface (VDI) can interface with OS layer or VPU directly. If the target system has a kernel mode such as Linux, VDI calls its device driver. For other OS like RTOS or Non OS platform, VDI can interface with the OS layer or VPU directly.

The relevant codes are in vdi directory and they have some porting layer (platform dependent codes) to integrate to a new target system. There is not any VPU dependency.

Figure 2.6, "VDI Layer" shows which components or function modules are in the VDI layer.

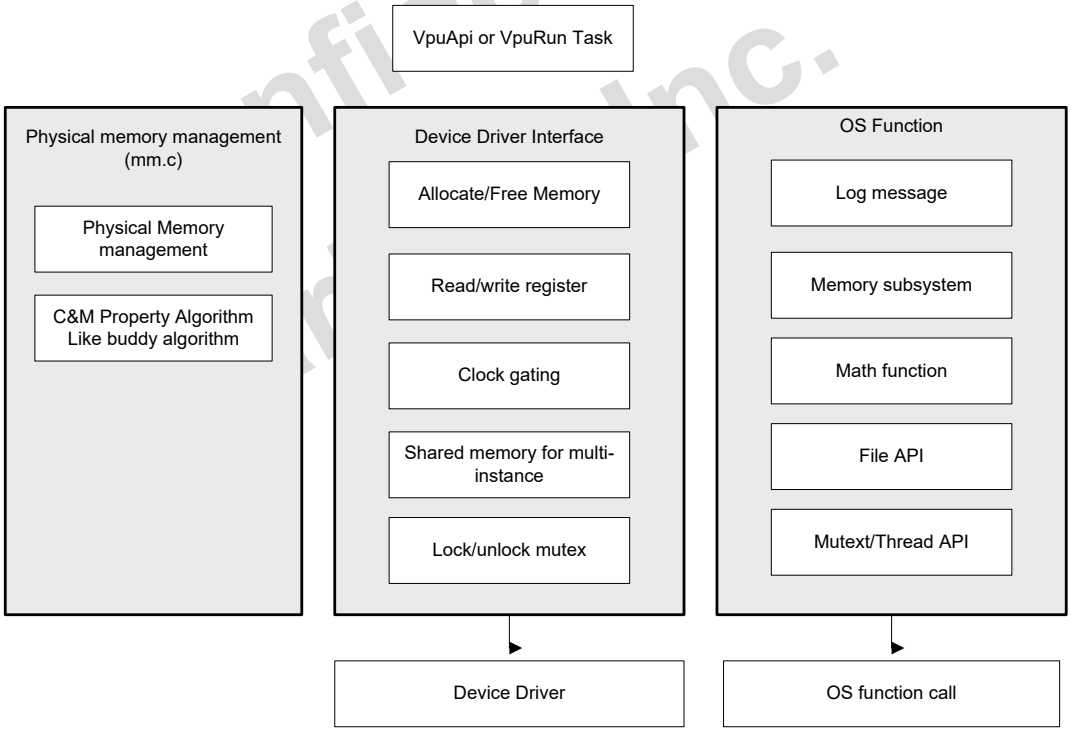


Figure 2.6. VDI Layer

The VDI Layer is taking charge of the following things.

- Read/Write VPU Register
- Read/Write physical memory
- Convert physical address to virtual address

- Allocate/Free memory
- Management of memory map for VPU in case not to use dynamic allocation, but to use system call
- Management of instances with shared memory in kernel mode
- SoC specified feature (HW reset, clock gating, interrupt)

2.2.2.4. Device Driver Layer

The Device Driver Layer accesses and handles VPU hardware, running in kernel mode if the OS has a framework for device driver. The relevant codes reside in VDI/[OS]/driver directory. It allocates physical buffer by using OS function calls, handles an interrupt, manages shared memory that is used for vpuapi handles for multi-process application.

2.2.3. Supported Operating Systems

2.2.3.1. Linux

VPU Reference Software is ported to Embedded Linux kernel version 2.6.35. The relevant codes are in VDI/linux folder.

- Build Environment
  - Toolchain: Sourcery CodeBench Lite 2011.09-65 for GNU/Linux
  - makefile in root folder: BUILD_CONFIGURATION variable should be Debug_Embedded_Linux. Also configure CC, CXX, and AR variables in case that cross compiler needs to be changed.
- Verification platform
  - ARM + FPGA platform board of SOCLE inc.

2.2.3.2. Non OS

VPU Reference Software is ported to ARM core base without a special OS. The relevant codes are in VDI/nonos folder.

- Build Environment
  - Sourcery CodeBench Lite 2011.09-65 for ARM EABI
  - makefile in root folder: BUILD_CONFIGURATION variable should be `Debug NonOS`.
- Verification Platform
  - MQX RTOS for certain ARCH platform
  - UCOS RTOS for certain ARM platform

2.3. Porting to Target System

2.3.1. Identifying Build Tool (c - compiler)

Example 2.1. config.h

```
--------------------------------------------------------------------------------
```

Confidential StarFive Inc.

```
#       define PLATFORM_NON_OS

#  error "Unknown compiler."
--------------------------------------------------------------------------------
```

Application should choose their target OS among PLATFORM_WIN32, PLATFORM_LINUX, PLATFORM_NON_OS

- PLATFORM_LINUX: when target platform is either embedded Linux or Android
- PLATFORM_WIN32: when target platform is Windows Embedded Compact 7 or Windows8
- PLATFORM_QNX: when target platform is Unix
- PLATFORM_NON_OS: when not in use of OS

For any other OS that is not defined here, new definition and platform dependent codes for that OS might be added.

2.3.2. Porting VDI

2.3.2.1. Creating a New VDI Folder

The first thing to port VDI is creating a new VDI folder and adapting the vdi.c and vdi_osal.c file to a new target system.

2.3.2.2. vdi Function Prototype

Example 2.2. vdi.h

```
--------------------------------------------------------------------------------
int vdi_probe(unsigned long core_idx);
int vdi_init(unsigned long core_idx);
int vdi_release(unsigned long core_idx);    //this function may be called only
                                              at system off.

vpu_instance_pool_t * vdi_get_instance_pool(unsigned long core_idx);
int vdi_get_common_memory(unsigned long core_idx, vpu_buffer_t *vb);
int vdi_allocate_dma_memory(unsigned long core_idx, vpu_buffer_t *vb);
void vdi_free_dma_memory(unsigned long core_idx, vpu_buffer_t *vb);
int vdi_get_sram_memory(unsigned long core_idx, vpu_buffer_t *vb);
int vdi_wait_interrupt(unsigned long core_idx, int timeout, unsigned long
addr_bit_int_reason);
int vdi_hw_reset(unsigned long core_idx);
int vdi_set_clock_gate(unsigned long core_idx, int enable);
int vdi_get_clock_gate(unsigned long core_idx);
int vdi_get_instance_num(unsigned long core_idx);
void vdi_write_register(unsigned long core_idx, unsigned long addr, unsigned int data);
unsigned long vdi_read_register(unsigned long core_idx, unsigned long addr);
int vdi_write_memory(unsigned long core_idx, unsigned long addr,
unsigned char *data, int len, int endian);
int vdi_read_memory(unsigned long core_idx, unsigned long addr,
unsigned char *data, int len, int endian);
int vdi_lock(unsigned long core_idx);
void vdi_unlock(unsigned long core_idx);
int vdi_disp_lock(unsigned long core_idx);
void vdi_disp_unlock(unsigned long core_idx);
int vdi_wait_vpu_busy(unsigned long coreIdx, int timeout, unsigned long addr_bit_busy_flag);
void vdi_log(unsigned long coreIdx, int cmd, int step);
--------------------------------------------------------------------------------
```

2.3.2.3. Implementation of vdi.c

#define VDI_SYSTEM_ENDIAN, VDI_LITTLE_ENDIAN

Sets an endian mode according to SoC's bus endian. For example, SoC uses AXI bus of ARM processor, VDI_LITTLE_ENDIAN should be defined.

#define VPU_BIT_REG_BASE 0x10000000
Sets the base address for VPU hardware register in SoC's memory map. When there is a device driver on the target platform, the device driver does set this address and so application does not need to set.

#define VDI_SRAM_BASE_ADDR 0x00
Sets the base address for VPU Secondary AXI bus (SRAM).

Note Set VDI_DRAM_PHYSICAL_BASE instead when you use VDI for Linux or Windows.

#define VDI_SRAM_SIZE 0x25000
Sets the size of VPU Secondary AXI bus (SRAM).

#define VDI_DRAM_PHYSICAL_BASE 0x00
Assigns the base address of memory that VPU uses. When there is a device driver on the target platform, application should make the device driver get the base address through the IOCTRL, VDI_IOCTL_GET_RESERVED_VIDEO_MEMORY_INFO.

#define VDI_DRAM_PHYSICAL_SIZE (1024*1024*1024)
Sets total size of memory that VPU uses

Note When you use VDI for Linux or Windows, set VPU_INIT_VIDEO_MEMORY_SIZE_IN_BYTE instead which is found vdi/linux(or windows)/driver/vpu.c.

#define SUPPORT_MULTI_CORE_IN_ONE_DRIVER
Enable this when VPU cores are in a same SOC using one DRAM. Comment out this code when VPU cores are in different SOCs using their own DRAM.

vdi_init()
This function creats a new vdi handle. For the OS with device driver, this function loads the device driver. It creates VPU instance mutexes for each OS and display lock mutex handles.

vdi_hw_reset()
This function is for VPU hardware reset. It should include some codes to issue a reset signal for VPU according to SoC environment.

vdi_lock(), vdi_unlock()
In multi-thread (multi-task) environment, a pair of these functions prevent VPU as a critical section from being concurrently accessed by threads. Add a mutex lock/unlock function according to target system. For example, application can use `pthread_mutex_lock()` and `pthread_mutex_unlock()` function when its target system is Linux.

vdi_write_register(), vdi_read_register()
These functions write to and read from Host Interface Register of VPU. Register access function is required to implement accoring to target system. It is supposed to direct access to VPU register address with volatile keyword as a default setting.

vdi_write_memory(), vdi_read_memory()
This function transfers data between CPU memory and VPU memory. `vdi_write_register()` copies CPU memory data (buffer pointer as an input argument) to the VPU target address. Vice versa `vdi_read_register()` copies data from VPU memory to CPU memory. As default a `memcpy()` function is used for this, but it might be possible to replace with a system dma copy function.

vdi_set_clock_gate(), vdi_get_clock_gate()
This function enables VPU clock gating. Application should implement a function to gate or ungate the VPU clock according to SoC environment. It is an option for low power, not mandatory to implement.

vdi_get_sram_memory()
This function returns the address of VPU secondary memory. Application does not need to port it, but simply assign the base address to VDI_SRAM_BASE_ADDR.

Confidential StarFive Inc.

vdi_get_common_memory()
This function returns common DRAM memory address that is used by VPU. Porting is not desired.

vdi_wait_interrupt
This function waits for an interrupt. If SoC supports an interrupt, application should add an interrupt waiting function for the target OS. We provide a polling function that reads out VPU status register periodically for the SoC not using an interrupt, and application should add the Sleep(1) code in it according to your target system because timeout is calculated in unit of 1ms.

2.3.2.4. Implementation of vdi_osal.c

LogMsg function()
This function sets print of debug string such as UART print.

osal_memcpy(), osal_memset(), osal_malloc(), osal_free()
This function manages virtual memory of OS. VDI controls the physical memory used for VPU DMA.

osal_fxxx()
This function is for file processing. Application (VPURUN) code load or save bitstream and image output in a file format, so there is need for application to work with file system .

link system library
Since VDI calls system related functions (malloc, memcpy, and so forth), libc libaries for the target OS should be prepared and linked.

2.3.3. Configuration of VPU

2.3.3.1. VPUAPI/vpuconfig.h

MAX_INST_HANDLE_SIZE
It is the size of variable holding information of instances that are managed inside driver. It must not be changed.

MAX_NUM_INSTANCE
Configures the number of instances to run if application wants to operate more than one instance. Instances could be set as many as possible within DRAM size available.

MAX_NUM_VPU_CORE
Sets the number of VPU cores if target SoC does have more than one VPU core to enable parallel processing or multi-channel.

VPU_BUSY_CHECK_TIMEOUT
Sets waiting time until a certain command is executed. This is millisecond unit and if there is not any response from it, VPU returns the error code, RETCODE_VPU_RESPONSE_TIMEOUT.

VPU_FRAME_ENDIAN
Sets an endian mode for frame buffer (image) data. i.e. VDI_128BIT_LITTLE_ENDIAN is set when ARM processor and AXI Bus is used.

VPU_STREAM_ENDIAN
Sets an endian mode for bitstream buffer data VDI_128BIT_LITTLE_ENDIAN is set when ARM processor and AXI Bus is used.

CBCR_INTERLEAVE
Sets a CBCR interleave mode
- 1: CBCR interleaved
- 0: CBCR separated

i.e. Set this to 1 when FOURCC value is NV12. Or set this to 0 when FOURCC value is YV12.

Confidential StarFive Inc.

NV21_ENABLE
Specifies the chroma interleave format.
- 1: NV21
- 0: NV12

SIZE_COMMON
It is the total size of code memory and stack memory which all are used by VPU.

2.3.3.2. VPUAPI/wave4/wave4_vpuconfig.h

WAVE4_STACK_SIZE
It is the size of stack memory for VPU. 8KB or more should be set for this.

WAVE4_MAX_CODE_BUF_SIZE
It is the size of firmware binary file. 1MB or less should be allocated for this code and stack memory.

WAVE4_WORKBUF_SIZE
It is the size of work buffer where some variables for running VPU are saved. More than 512KB of memory region should be allocated for this.

WAVE4_NUM_BPU_CORE
It is the number of VCE core inside VPU. It should set to 1.

2.3.4. Porting Device Driver

When user mode and kernel mode are separated on the OS such as Linux, it is expected to do porting the VDI part to fit into a new target OS. Application should implement a device driver for framework of the target OS.

2.3.4.1. IO Control Codes

Porting IOCTL Codes for Linux driver

```
VDI_IOCTL_MAGIC  'V'

VDI_IOCTL_WAIT_INTERRUPT                    IO(VDI_IOCTL_MAGIC, 2)
VDI_IOCTL_SET_CLOCK_GATE                   _IO(VDI_IOCTL_MAGIC, 3)
VDI_IOCTL_RESET                            _IO(VDI_IOCTL_MAGIC, 4)
VDI_IOCTL_GET_INSTANCE_POOL                _IO(VDI_IOCTL_MAGIC, 5)
VDI_IOCTL_GET_RESERVED_VIDEO_MEMORY_INFO    _IO(VDI_IOCTL_MAGIC, 8)
```

VDI_IOCTL_WAIT_INTERRUPT
A waiting function based on interrupt scheduling for the device driver framework.

VDI_IOCTL_SET_CLOCK_GATE
Clock gating on/off

VDI_IOCTL_RESET
HW reset signal on/off

VDI_IOCTL_GET_INSTANCE_POOL
Returns shared memory which is used by VDI and VPU API This memory should always be returned with same region.

VDI_IOCTL_GET_RESERVED_VIDEO_MEMORY_INFO
Allocates the entire size of memory that is used for VPU hardware and returns the address. Video processing demands considerable of memory space and most of the memory allocations should be done at booting time, which is also dependent on OS. Therefore, device driver actually allocates memory for VPU at booting time, and VDI returns the memory information by the VDI_IOCTL_GET_RESERVED_VIDEO_MEMORY_INFO io-control.

Confidential
Chips&Media

Note For an example of making a new device driver, please refer to sample driver codes in vdi/linux/driver folder.

2.3.4.2. Device Driver Configuration

#define VPU_SUPPORT_CLOCK_CONTROL
Enable Clock Gating control by using clock library in Linux kernel.

#define VPU_SUPPORT_ISR
Enable the use of interrupt service routine.

#define VPU_SUPPORT_PLATFORM_DRIVER_REGISTER
Get a base address and IRQ number from platform driver library.

#define VPU_INIT_VIDEO_MEMORY_SIZE_IN_BYTE
Set the memory size for memory allocation for running VPU. It must be larger than REQUIRED_VPU_MEMORY_SIZE that is calculated by vpuconfig.h.

#define VPU_SUPPORT_RESERVED_VIDEO_MEMORY
Enable the use of reserved memory region for VPU memory. When this is disabled, application should allocate memory by calling kernel API for device driver's non-cache/continuous physical memory allocation. For example, Linux driver uses a dma_alloc_coherent function.

#define VDI_DRAM_PHYSICAL_BASE
Set the physical base address of reserved memory if VPU_SUPPORT_RESERVED_VIDEO_MEMORY is enabled.

Confidential StarFive Inc.

2.4. Verification

For information on how to run sample decode/encode test using Chips&Media API reference software, please refer to the *Chapter 4. Software Verification Environment* of *Verification Guide*.

Confidential
StarFive Inc.

Chapter 3
HOW TO CONTROL VPU

3.1. VPU Initialization

When host processor turns on VPU for the first time, host processor needs to follow the steps below to activate VPU, which is called an initialization process. All these manual procedures including VPU hardware reset, firmware load, interrupt enable, and VPU start can be replaced with simply calling a single API function `VPU_Init()` that we provide.

1. Call Wave4VpuReset() function to reset all hardware blocks.
   a. Disable bus request.
   b. Waiting for completion of bus transaction.
   c. Reset all hardware blocks by setting VPU_RESET_REQ register to 0x3fffffff.
   d. Wait until VPU_RESET_STATUS register becomes 0.
   e. Enable bus request.
2. Set VPU_PO_CONF register to 0.
3. Place the binary file of firmware in the SDRAM where is accessible by VPU.
   a. Set ADDR_CODE_BASE register to SDRAM address where the firmware is placed. ADDR_CODE_BASE must be aligned in 4KB boundary.
   b. Set CODE_SIZE to the size of firmware.
   c. Set CODE_PARAM with endian. This should be same with the firmware's endian.
4. VPU starts booting from 0x0 of virtual address. Therefore, the code region should be remapped to virtual address for VPU,
   a. Set VPU_REMAP_CTRL register. (Set GLOB_EN flag to 1 for global setting such as Endianness and enter the values of ENDIAN.)
   b. Set VPU_REMAP_VADDR to 0x0 (Code Section always starts from 0)
   c. Set VPU_REMAP_PADDR with ADDR_CODE_BASE.
5. Set HW_OPTION register for activating hardware options such as UART or debug option. Generally, it is 0.
6. Set VPU_VINT_ENABLE register to enable an initial interrupt if necessary.
7. Write VPU_BUSY_STATUS register to 1.
8. Set COMMAND(0x100) register to 1 for INIT_VPU command.
9. Finally set VPU_REMAP_CORE_START to 1 to start VPU.

Confidential StarFive Inc.

Note | A total code size of firmware could be varied according to VPU configuration and customization.

3.1.1. Version Check of VPU hardware and Firmware

Application can check the version information of VPU hardware and firmware by using GET_FW_VERSION command. The version number of firmware is presented by a 32 bit value.

- Revision[31:0] - F/W revision number

The dedicated command is used for this version check, and also this is supported by calling `VPU_GetVersionInfo()` function after initialization.

3.1.2. Data Buffer Management

VPU requires a certain amount of SDRAM space for decoding or encoding operation. This dedicated memory space for VPU includes a code buffer, a working buffer, and a temp buffer.

- A code buffer is a memory region where firmware binary is stored. Host processor should set the base address of code buffer so that VPU can start boot-up from the address. VPU has one code buffer regardless of number of instances opened.
- A working buffer is a memory region where the parameters and information of a sequence are saved. An instance has one working buffer.
- A temp buffer is a memory region that holds temporarily required information for performing actual encoding/decoding process such as intra prediction and deblocking filter. Picture size affects the size of temp buffer. Part of temp buffer might be connected to on-chip SRAM through secondary AXI bus to help reducing external bandwidth. VPU has its own temp buffer that can be shared by all the instances opened. However, in multi-VPU environment there are temp buffers as many as the number of cores.

Each buffer size might vary according to codec standard and picture size of stream that an instance uses. Thus, the minimum required sizes of the buffers except code buffer are returned by DEC_PIC_HDR command, respectively. The size of code buffer is larger than the sum of the size of firmware binary and stack for supervisor, because it also has uninitialized global variables (BSS sections) and additional area for stacks of codec space. One more thing to note is the size of code buffer should be fit to 2^N (for example, 256KB, 512KB, or 1MB) for remapping.

Basically, all the buffers should be aligned with bus width (128-bit/16-byte), except for some buffers in 4K alignment.

Allocation of Buffers

Host processor can assign a code buffer and temp buffer simply by calling `VPU_Init()` which automatically sets and manages individual buffers at once through VDI module.

A working buffer is assigned whenever an instance opens and it is as large as the size defined in `VPU_DecOpen()` or `VPU_EncOpen()`. Through the VPUAPI functions, the address of each buffer is set to the dedicated registers of Host Interface register.

3.1.3. Source Buffer Management (for encoder)

VPU calculates the minimum number of source buffer required for encoding with host-given information such as GOP structure, whether to use frame-parallel encoding, etc. VPU returns the information to host processor through RET_MIN_SRC_BUF_NUM register of SET_PARAM command. Host processor should allocate source buffers for VPU with the information RET_MIN_SRC_BUF_NUM and source raw data size.

Confidential
Star Five Inc.

Note RET_MIN_SRC_BUF_NUM is just the least number of source buffer for VPU. It does not include buffers that might be needed for feeding source buffer from CMOS sensor. We recommend using dual buffering scheme with more than RET_MIN_SRC_BUF_NUM so that VPU should not wait for source input picture to be completely received.

3.1.4. Bitstream Buffer Management

A bitstream buffer, known as compressed picture buffer (CPB), is a memory region that stores coded picture data which is known as bitstream. Host processor and VPU share the bitstream buffer.

For bitstream buffer mode, there is a linear buffer mode. A linear buffer mode allocates bitstream buffer dynamically with start and end region of bitstream in byte unit.

Host processor should set BS_START_ADDR and BS_SIZE register in alignment with 128-bit bus width, which is to inform VPU of the start address of bitstream buffer and its size. With setting the BS_START_ADDR and BS_SIZE register dynamically for every ENC_PIC command, bitstream buffer can run in linear buffer mode.

Host processor can set bitstream buffer parameters such as endianness, way of bitstream pumping or handling a bitstream shortage case on BS_PARAM register. VPU does not allow change of bitstream buffer parameters in a same instance. In other words, different parameters might be set for another instance.

3.1.4.1. Allocation of Bitstream Buffer

Bitstream buffer should be aligned to memory bus width. For example, if a host processor uses a 128-bit bus, bitstream should be aligned to a 128-bit boundary. Thus, host processor should set carefully BS_START_ADDR and BS_SIZE register to meet this requirement.

Even though there are no particular restrictions on bitstream buffer size, the bitstream buffer must be assigned with one or more coded picture data which is affected by bitrate.

3.1.4.2. Pointers for Reading Bitstream Buffer (Encoder)

To access encoded bitstream, there are two pointers to bitstream buffer, a read pointer (BS_RD_PTR) and a write pointer (BS_WR_PTR). When it comes to controlling these pointers, host processor can basically manipulate both pointers before sending a command to VPU. However, host processor is not allowed to control pointers while encoding.

BS_RD_PTR indicates the start of stream, and BS_WR_PTR indicates the end of stream. Once picture encoding is completed, host processor can get bitstream from BS_RD_PTR to the encoded byte size (RET_ENC_PIC_BYTE) or to BS_WR_PTR.

3.1.4.3. Pointers for Bitstream Pumping Operation (Decoder)

There are two pointers - a read pointer (BS_RD_PTR) indicating where VPU is reading stream data from the buffer and a write pointer(BS_WR_PTR) indicating where the buffer is being filled up. This scheme is preferred in packet-based video communication and streaming applications such as broadcasting or video conference.

When it comes to control of these pointers, host processor can basically manipulate both pointers before sending a command to VPU. However, host processor cannot change a read pointer while VPU is decoding and can manipulate a write pointer after feeding bitstream to bitstream buffer.

When applications are going to feed a new chunk of bitstream, they need to check if there is available space to write in bitstream buffer, which can be computed simply with a read pointer, a write pointer and a buffer size.

The API VPU_DecGetBitStreamBuffer() is a dedicated function for that, informing applications of location of read pointer and write pointer and available space in bitstream buffer. With the return value of this API function, applications can download a new chunk of bitstream whose size is smaller than available buffer space to bitstream

Confidential StarFive Inc.

buffer. There is another API, VPU_DecUpdateBitStreamBuffer() that allows applications to get informed the amount of bits transferred into bitstream buffer (BS_WR_PTR).

Host processor can use API functions to manipulate BS_RD_PTR and BS_WR_PTR.

- VPU_DecSetRdPtr() updates BS_RD_PTR
- VPU_DecUpdateBitstreamBuffer() updates BS_WR_PTR

3.1.4.4. Bitstream Handling Modes (Decoder)

When VPU starts decoding, VPU executes prescan to find if a complete NAL exists in bitstream buffer and loads 4KB chunk of bitstream from the buffer. If it turns out there is enough NALs to decode one frame, VPU does not load any more bitstream.

When VPU fails to decode from bitstream shortage, VPU behaves according to either of two bitstream handling modes, interrupt mode and PicEnd mode.

- In the interrupt mode, VPU does not do anything and waits until host processor fills bitstream in the buffer.
- In the PicEnd mode, VPU acts according to the given BS_SHORTAGE_OPTION flag of BS_PARAM register. It is either concealment or error report.

Host processor can select either Interrupt mode or PicEnd mode with the EXPLICIT_END flag of BS_OPTION register.

The bitstream handling mode should be set before issuing DEC_PIC command. Host processor cannot change from interrupt mode to PicEnd mode while an instance is running. However, to disable EXPLICIT_END flag after setting up that in interrupt mode, host processor should wait until VPU has completed ongoing DEC_PIC command.

One more thing that host processor needs to be careful is use of BS_WR_PTR in PicEnd mode. Host processor must not update BS_WR_PTR until one picture decoding is finished. WR_PTR should be static while decoding for safe operation.

Confidential StarFive Inc.

3.1.4.4.1. Interrupt Mode

Interrupt mode is a basic mode for handling bitstream in VPU. If remaining bitstream data in the bitstream buffer is less than 512 bytes, VPU sends an interrupt to host processor to ask for more bitstream. In this case, the interrupt reason is set as bitstream buffer empty that is 14th bit in BIT_INT_REASON. After receiving the interrupt, host processor should fill bitstream to the bitstream buffer with new coded picture data or set EXPLICIT_END flag. Meanwhile, VPU waits for more bitstream to be fed up or the EXPLICIT_END flag to be updated if there exist not enough bitstream to decode.

Note We recommend that host processor should feed bistream larger than 1024 bytes into the bitstream buffer when it is right after SEQ_INIT command for safe decoder operation.

NAL level pumping

In the interrupt mode, VPU tries to decode to the almost end of bitstream buffer (WR_PTR), but last three bytes or less in the bitstream buffer are intended not to be consumed during decoding. It is because they might be part of start code for the next NAL, or they might not be filled into the offset in the CABAC; CABAC needs more than 4-byte chunk.

To overcome this problem, VPU has NAL level pumping mode. The size of a NAL unit can be determined in host parser by checking start code of the next NAL unit, by checking STOP pattern, or by using size information in container header. In interrupt mode, VPU is hard to detect the end of NAL thus some of bytes cannot be decoded. In NAL level pumping, VPU assume the position of BS_WR_PTR is always the end of a NAL unit; thus, it can

consume all the byte in the bitstream buffer if host processor can guarantee. User can enable this mode by setting NAL_END and EXPLICIT_END flags.

This mode is the simplest and efficient way of bitstream management, but it has some weak points as follows.

- If streaming coded picture data for a frame is much slower than consuming, it might lead unacceptable latency in instance switch (multi-instance use-case), because VPU switches a current instance with another only if the current instance finishes its decoding.

To handle these situations, host processor can do mode-switch into PicEnd mode, which continues decoding with mere small bitstream.

3.1.4.4.2. PicEnd Mode

In this mode, VPU decodes until the end of bitstream that host processor fills in. After the decoding, VPU might encounter either of the following two cases:

- If it was end of bitstream for a picture and VPU finishes decoding a whole frame successfully, VPU returns the result of picture done as it normally does.
- If it was not enough to complete decoding a frame or sequence header, VPU assumes that this is bitstream shortage case, and it performs predefined operation by the BS_SHORTAGE_OPTION@BS_OPTION

Assume Error on bitstream shortage

In this option, VPU assumes error in the end of stream and try to conceal error for the remaining pixels in the picture. After concealment, VPU return RET_SUCCESS as 0x2 (Success with warning) and write the FAIL_REASON. This mode is very useful in low-latency application and with small size of bitstream buffer.

File-Play Emulation using PicEnd mode

The PicEnd mode can emulate file play operation by designating where a certain file data begins and ends with BIT_RD_PTR and BIT_WR_PTR respectively. VPU can decode a stored coded picture data directly from BIT_RD_PTR to BIT_WR_PTR, without copying to bitstream buffer for decoding. To work this way, BIT_RD_PTR and BIT_WR_PTR should stay in bitstream buffer region. In other words, host processor should set bitstream buffer large enough to hold many files. Generally, host processor sets almost all of external memory as bitstream buffer for file-play emulation using PicEnd mode.

Report Error on bitstream shortage without addition handling

In this mode, VPU returns RET_SUCCESS as 1 immediately without any additional operation. In this case, host processor can drop the frame or conceal in host side.

3.1.4.5. Considering Multiple Instances

With multi-instance feature, host processor needs to manage its own list of BS_RD_PTR and BS_WR_PTR as many as instances are created with RunIndexes to identify each instance. VCPU changes the pointers based on the instance and sets the RD_PTR (in decoder case) and WR_PTR (in encoder case) using the current RunIndex, also known as instanceID. Host processor can get informed about how much bitstream data has been read in decoder and written in encoder for a specific instance by using QUERY_RD_PTR/QUERY_WR_PTR command and UPDATE_BS command.

3.1.5. Interrupt Signaling Management

To achieve maximum efficiency in VPU control, VPU basically provides interrupt signaling for completion of a requested operation as well as stream buffer empty/full. But for some commands returning quickly, interrupt signaling is not provided because interrupt signaling is not helpful in that case.

Confidential
Chips&Media Inc.

Currently, C&M provides interrupt signaling for the following commands:

- VPU_INIT: VPU processor initialization has been completed after setting VPU_REMAP_CORE_START by 1.
- DEC_PIC_HDR (SET_PARAM for encoder): VPU sequence initialization has been completed.
- FINI_SEQ: VPU sequence termination has been completed.
- DEC_PIC (ENC_PIC for encoder): VPU picture processing has been completed.
- SET_FRAMEBUF: Registration of frame buffer has been completed.
- FLUSH_DECODER: Forcing decoder to flush bitstream buffer and frame buffers has been completed.
- GET_FW_VERSION: Retrieving F/W version has been completed.
- QUERY_DECODER: Reserved.
- SLEEP_VPU: SLEE_VPU command has been completed.
- WAKEUP_VPU: WAKEUP_VPU command has been completed.
- CREATE_INSTANCE: Instance creation has been completed.

Each interrupt could be easily enabled or disabled by writing 0 or 1 to the corresponding bit field of Interrupt Enable Register. And when an interrupt is signaled, applications can check the source of interrupt by checking the value of Interrupt Reason Register.

All these kinds of interrupt signaling could be replaced by a polling scheme of reading VPU_BUSY_STATUS register in VPU when interrupt signaling is not easily applicable.

Confidential
StarFive Inc.

3.2. Encoder Control

3.2.1. Overall Encoder Sequence

Figure 3.1, "Encoder Control Flow with APIs" gives a summary of encoder control flow that shows the relation between encoder control flow and API function. And it also represents short description of each stage.

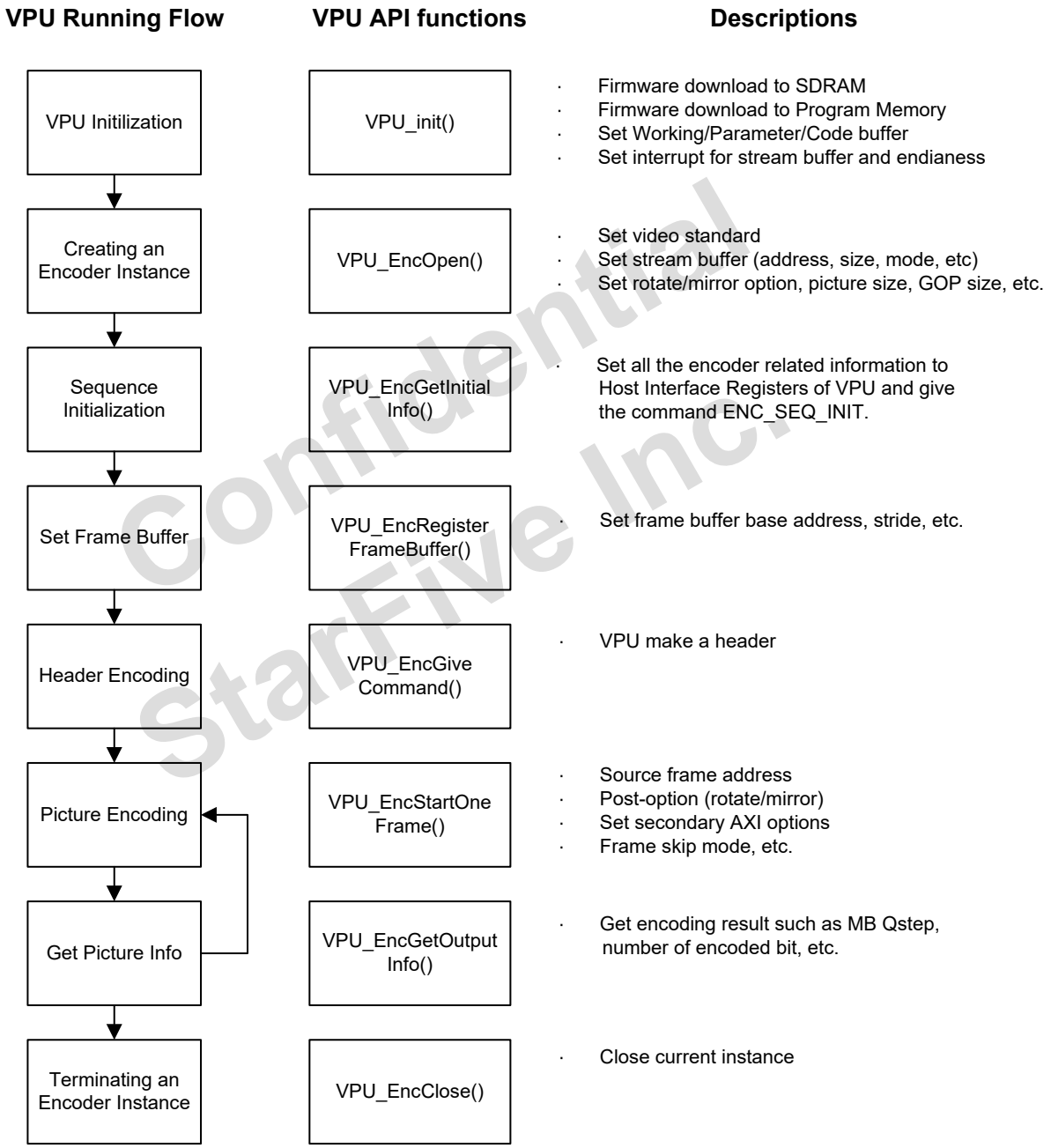


Figure 3.1. Encoder Control Flow with APIs

Confidential StarFive Inc.

3.2.2. Creating an Encoder Instance

After initialization of VPU, the first step to run encoder operation is the creation of an encoder instance, and the acquisition of the instance handle so that VPU can identify which instance is now running in multiple instance environment. It could be easily done by using a single API function called `VPU_EncOpen()`.

When creating a new encoder instance, applications should specify the internal features of this encoder instance through `EncOpenParam` structure. This structure includes the following information about the new encoder instance:

- Bitstream buffer address & size

  The physical start addresses of bitstream buffer and its size

- Codec standard

  A video codec standard such as H.265/HEVC

- Picture size

  A width and height of picture to encode

- Target frame rate and target bitrate with initialDelay

- Dynamic buffer allocation enable

  Applications can allocate picture bitstream buffer dynamically by enabling dynamic buffer allocation. The address and the size of picture bitstream buffer can be dynamically given by applications when issuing picture encoding operation. For use of static bitstream buffer, the address and the size of bitstream buffer retrieved by `VPU_EncGetInitialInfo()` are used in picture encoding.

- Video standard specific parameters

  Specify standard-specific parameters for each video codec standard such as error resilience tools in MPEG-4, Annexes in H.263, and deblocking parameters in H.264/AVC. For example, MPEG-4 encoder has many encoding parameters including data partitioning, RVLC, version ID, and so forth. For details about standard specific parameters, please refer to the API reference manual.

  Most of the H.265/HEVC parameters for encoding are defined in EncHevcParam data structure.

- Profile, Level, and Tier

  VPU supports both main profile and main 10 profile. Host processor can add profile information to SPS by setting the profile register. However, if you set 0 or have done nothing to the register, VPU automatically encodes a profile by using the bit depth of source picture - main profile with 8bit and main10 profile with 10bit.

  VPU can support up to the level 5.0 with main and high tier. The level and tier information can also be given to SPS by setting the level and tier register, repectively. When the level or tier register has 0, VPU can determine level and tier on its own and add a valid value to SPS.

- BitDepth and ChromaFormatIdc

  Host processor should give VPU basic picture information such as pixel bitdepth and chroma format of source picture. Currently, VPU can take 8/10 bit 4:2:0/4:2:2 picture as its input.

Note VPU actually converts 4:2:2 format into 4:2:0 format inside it and then encodes them, so 4:2:0 is an only output format by VPU.

- Lossless mode and constrained intra prediction mode

Confidential Start-Five Inc.

VPU can encode in lossless mode for the applications where no distortion is allowed in reconstructed frames. Also it supports constrained intra prediction.

- Cyclic GOP related parameters

  VPU provides cyclic GOP parameters that are configurable for composing a wide range of coding structures. CyclicGopSize indicates the number of repeated picture(s) of coding structure possibly up to 8. VPU delivers CyclicGopPresetIdx for host processor to use commonly used coding structures (IPP, IBP, IBBP, etc).

- Temporal scalability

  VPU supports temporal scalability offering up to 7 temporal layers.

- Other varied encoding tools

  VPU offers a variety of encoding tools like CuSizeMode, TmvpEnable, SkipIntraTrans, etc. so that applications can use whatever fits their requirements and needs by setting the relevant registers.

  In addition, VPU has a suggested combination of encoding tools called a preset for easy configuration.

- Rate control parameters

  VPU is capable of rate control that takes bitrate, video quality, and buffer status into consideration at hierarchical levels - GOP, CTU, and Sub-CTU (32x32) level. For customizability, there are many parameters for rate control - TransRate, RcTargetRateLayer, CuLevelRcEnable, HvsQpEnable, HvsLambdaEnable, HvsQpScaleEnable, HvsQpScale, MinQp, MaxQp, and MaxDeltaQp.

  It also has other parameters such as HierachicalBitEnable and RcTargetRateLayer allowing accurate rate control based on temporal scalability using multiple layers.

- Slice enable/disable, slice mode and size

  VPU can encode a picture into bitstream with mulitple slices as many as configured. Also there can be slice segments within a slice.

  Host processor can define a slice with the number of CTB and a slice segment with the number of generated bits as well as CTB count. VPU generates multiple slices or slice segments unless the slice size register is 0.

- IntraRefresh

  IntraRefresh can be enabled for error robustness. Host application can specify the number of intra CTBs in a non-intra picture and IntraRefresh mode.

- ConformanceWindowOffsets

  VPU provides parameter for specifing the number of pixels to crop named ConformanceWindowOffsets (ConfWinLeft, ConfWinRight, ConfWinTop, ConfWinBot). It is for decoder to only display a rectangular region given by host processor .

By using these options, applications can get the well-optimized encoding output that fits requirements of target applications.

For example, in case that packet size is fixed, applications might need to insert one slice to a certain number of bits. However, it is hard for the application to make sure that output slice size is smaller than the given size because of variable length characteristics of encoding process. For above reason, many applications should divide a slice into two packets frequently, which might cause big inefficiency in packetization. To achieve an easy packetization and effectiveness, applications can set a slice size by (`packet_size` : N) with a certain margin of N, which allows output slice size to be less than the packet size. Then applications can easily put a slice into a packet by just referring the slice boundary information provided by VPU as an encoder output.

Confidential StarFive Inc.

After creating an encoder instance with these parameters, applications cannot change these initial parameters specified in this stage. In principle, if applications want to change any of these basic parameters, it should close this instance and re-create another encoder instance with new initial parameters. But in practice, applications need to change some of these initial parameters depending on target application environment. So by providing a dynamic configuration command, VPU API enables applications to configure part of these initial parameters dynamically. For the details, refer to the `VPU_EncGiveCommand()` description in the API reference manual.

The API function, `VPU_EncOpen()`, does not require any operations on VPU side but declares all the internal parameters to be used in later stage as well as bitstream buffer information.

3.2.3. Configuring VPU for Encoder Instance

3.2.3.1. Sequence Initialization

After registering all the required information for a new encoder instance, host applications should configure VPU to be ready for supporting the new encoder instance. This procedure is done by setting all the encoder related information to Host Interface Registers of VPU and giving the command, `SET_PARAM` to the VPU for initiating internal configuration operation in VPU.

There are three modes of running SET_PARAM command - COMMON, GOP, and SEI. For sequence initialization, first host applications should set SET_PARAM command (COMMON) parameter registers and give VPU the SET_PARAM command.

After then if host applications want to use a customized cyclic GOP instead of cyclic GOP preset, they should configure the relevant SET_PARAM command (GOP) parameter registers and give VPU the SET_PARAM command again.

All those processes are mainly done by an API function, `VPU_EncGetInitialInfo()` and this function returns very crucial output parameters for encoder operation , the minimum number of frame buffers and the minimum number of source buffers. Mostly this process does not require so much time, and it should be done only once at the beginning of encoder instance. So it is not recommended to use interrupt signaling for this function, but interrupt signaling is allowed after completion of this operation by enabling the corresponding bit on Interrupt Enable Register.

Confidential
Starfive Inc.

3.2.3.2. Registering Frame Buffers

This configuring process is finished by registering frame buffers to VPU for its future picture encoding operation. In this final stage of configuration, the returned parameter from `VPU_EncGetInitialInfo()`, the minimum number of frame buffer, has a very important meaning. This parameter means that applications should reserve at least the same number of frame buffers to VPU for proper encoding operation. This configuration can be done by the API function `VPU_EncRegisterFrameBuffer()`.

3.2.3.3. Generating High-level Header Syntaxes

When opening an encoder instance is completed by calling `VPU_EncGetInitialInfo()`, applications MUST generate the high-level header syntaxes such as VPS/SPS/PPS in HEVC from VPU by using `VPU_EncGiveCommand()`.

There are two possible ways for generating these header syntaxes. The recommended way for getting header syntaxes is to use `ENC_PUT_VIDEO_HEADER`. If applications use this command, then the header syntaxes as a result are stored into the bitstream buffer according to the given endian setting.

The other way is creation of header syntaxes while picture encoding is performed. 1 of implicitHeaderEncode allows to generate not only slice data but also header syntaxes when ENC_PIC command is executed. If nothing

changes ever since header syntaxes have been created, header syntaxes are not anymore be generated on later picture encoding.

3.2.4. Running Picture Encoder on VPU

3.2.4.1. YUV Input Loading

Before running a picture encoder operation, host applications should provide an input YUV image in 4:2:0/4:2:2 format (8bit or 10bit per pixel) with pre-defined image size for H.265/HEVC.

The new input image might be coming from external video input device like CMOS sensor. In order to prevent VPU from being idle while waiting for completion of receiving input picture, it is recommended to use dual buffering scheme of input image. Then encoder does not spend any cycles for idling before starting its operation.

3.2.4.2. Initiating Picture Encoding

When activating picture encoding operation, applications should provide the following information to VPU:

- Source frame address

  A source frame address contains the base address of each component of input YUV picture, which includes luminance and chrominance (or Cb and Cr) frame buffer base address.

- Forced frame skip option

  Forced frame skip is to skip encoding a current frame unconditionally.

- Output information report

  Applications can enable or disable whether VPU generates the informative output data such as CU Info (CU QP, slice boundary) and Slice Info during encoding.

- Pre-processing option (Rotating/mirroring)

  Set rotating and mirroring option to VPU. VPU reads the source frame and rotate it to encode.

- forcePicQpI, forcePicQpP, forcePicQpB

  Applications can force to set a quantization parameter for I, P, or B picture when rate control is disabled.

- Picture type

  A picture type (I, P, or B) is configurable for encoding a current picture. When use of cyclic GOP, the picture type pre-determined in cyclic GOP is ignored.

- IRAP picture (forceIPicture)

  A current picture can be set as IRAP picture (IDR or CRA) to work as random-access point. This option changes a picture type of the current picture into an I picture.

Note It might not be available for the application where encoding order is different from display order (delay exists).

- Secondary AXI

  Set a secondary AXI option for bandwidth saving. There are secondary AXI options for encoding, which are useEncImdEnable, useEncLfEnable, and useEncRdoEnable

- Frame buffer

Confidential StarFive Inc.

Set a frame buffer mode such as CbCr interleave , NV21, and endianness.

- Implicit Header Encoding

  Set an implicit header encoding mode for generating bitstream conforming to HEVC standard. Receiving a new SET_PARAM command during encoding with change of the encoding tool associated parameters like disableDbk, VPU encodes and adds the new high level syntax (SPS or PPS) before bistream of the current picture.

After setting all these information to the VPU, host processor can initiate picture encoding operation by sending ENC_PIC command to the VPU. All of these pre-defined processes can be performed by just calling a single API function, `VPU_EncStartOneFrame()` with `EncParam` structure. This API function initiates picture encoding operation. Returning from this API does not mean that picture encoding is completed.

The quantization step size given to the VPU with ENC_PIC_RUN is meaningful only when rate control option is disabled. This additional feature is provided to support application-specific VBR encoder operations.

The forced frame skip option is so useful when encoding a new picture is not allowed temporarily due to the external situation of encoder. Forced frame skip is used by applications when encoding a picture is problematic under a certain external situation. For example, we can assume that channel condition is temporarily too bad and transmitting encoded stream is impossible. Then the applications can suspend encoding operation for a while by using this Forced frame skip option.

3.2.4.3. Completion of Picture Encoding

Picture encoder operation takes a certain amount of time, and applications could go on other tasks while waiting for the completion of picture encoding operation such as packetization of encoded stream for transmission. Application can use two different types of schemes for detecting completion of picture encoding operation: polling a status register and interrupt signaling. When applications are going to use a polling scheme, they just need to check the BusyFlag in the corresponding register. Calling `VPU_IsBusy()` gives the same result.

Interrupt signaling could be the most efficient way to check the completion of a given command. An interrupt signal for ENC_PIC command is mapped on bit[3] of Interrupt Enable Register. So applications could easily know the completion of picture encoder operation from this dedicated interrupt signal from the VPU.

3.2.4.4. Encoder Stream Handling

When an encoder bitstream buffer is big enough to store any big size of picture stream, then the encoder does not need to get any bitstream during picture encoder operation. After encoder operation is finished, host applications can read the encoded bitstream according to the requirements of packetization.

But when an encoder bitstream buffer is not big enough to store a whole picture stream, encoder buffer-full can happen. Unless this buffer-full situation is resolved, the encoder task running on VPU might be hanged. So while picture encoding goes on, applications should keep reading out encoded bitstream from the bitstream buffer in order to avoid this undesirable encoder hanging. Therefore when bistream buffer becomes full, VPU asserts an interrupt to inform host processor of this situation.

When using a limited size of encoder bitstream buffer, bitstream reading during encoder operation is recommended. By using two dedicated functions, `VPU_EncGetBitStreamBuffer()` and `VPU_EncUpdateBitStreamBuffer()`, applications can easily handle the read pointer in encoder case while accessing the encoder bitstream buffer.

Note Host application needs to set at least 96 Kbytes of bitstream buffer for use of ring-buffer scheme due to VPU hardware constraint.

But if there is a big bitstream buffer (line buffer is used) enough to store one encoded picture data in order to avoid VPU from being hanged, then host processor could read encoded bitstream only after each picture

Confidential
StarFive Inc.

encoding has been completed. In this case, applications could do safely other tasks while picture encoding is running on VPU. When use of this frame-based streaming option, `VPU_EncGetBitStreamBuffer()` and `VPU_EncUpdateBitStreamBuffer()` are useless.

3.2.4.5. Acquiring Encoder Results

Whenever a picture encoding is completed, host applications can get the encoded output such as encoded bytes and encoded source index, recon frame index, etc.

The API provides a function VPU_EncGetOutputInfo() for getting output results of picture encoder. When using the API, applications should always use two functions VPU_EncStartOneFrame() and VPU_EncGetOutputInfo() as a pair. This means that without calling the result acquisition function, VPU_EncGetOutputInfo(), the next picture encoding operation is not be initiated by calling VPU_EncStartOneFrame().

This constraint might be very useful to protect the encoded results from overwritten from other thread by mistake in multi-instance environment. So applications should regard VPU_EncGetOutputInfo() function as a releasing command of VPU from current picture encoding operation.

- Encoded source picture index

  With support of encoding into B pictures, there might be a gap between encoded order and display order - an encoded picture might not be from the source picture when ENC_PIC command is executed. For that reason, VPU manages a souce picture index and an encoded picture index separately as a frame buffer identifier and accesses or controls the frame buffers with each index.

  Also, source pictures are likely to wait to be encoded due to delay if B picture is used. VPU returns an encoded source picture index for the current ENC_PIC command so that host processor can recognize which source picture has been encoded this time.

  – If there was no encoded picture due to intial encoding delay, VPU returns -2 of source picture index.

  – If there is no more source picture, host processor should set 1 to srcEndFlag, by which VPU returns -1 of source picture index for ENC_PIC command.

- Reconstructed picture index

  By ENC_PIC command, VPU not only encodes a picture into bitstream, but also reconstructs the picture and saves it to the frame buffer for later use of refererence.

  Note VPU embeds FBC (Frame Buffer Compression) hardware block for bandwidth saving which compresses and stores the compressed format of recon picture to the frame buffer of external memory.)

  VPU returns a recon picture index indicating an index of the frame buffer where the reconstructed frame of current encoded picture is stored.

  – If there was no encoded picture due to intial encoding delay, VPU returns -2 of the recon picture indexes.

  – If VPU returns -1 of the recon picture indexes, it indicates the end of encoding.

- Encoded cyclic GOP picture index

  Report on a picture index of cyclic GOP structure for the currently encoded picture when use of cyclic GOP

- Encoded picture POC

  A POC value of the currently encoded picture

- Encoded picture type

Confidential
StarFive Inc.

A picture type of the currently encoded picture
– 0: I picture
– 2: P picture
– 4: B picture

- Encoded picture number

  The number of pictures that have been encoded so far

- picture skip info

  Report on whether the current picture has been skipped or not

- Number of slice segments

  A picture can be composed of one or more slice segments. It reports how many slice segments are included in the currently encoded picture.

- Stream size of encoded picture in bytes

  Report on the nal unit byte size after completion of one picture encoding which is the total size of all nal units including VCL, SPS, PPS, etc.

- Number of Intra block

  The number of intra block (8x8 unit) within the currently encoded picture

- Number of skip block

  The number of skipped block (8x8 unit) within the currently encoded picture

- Average CTU QP

  The average Qp value for all CTUs within the currently encoded picture

3.2.5. Terminating an Encoder Instance

When host applications want to finish encoder operation and terminate an encoder instance, they just can release the handle of the instance and terminate the instance by using the SEQ_END command. It can be done simply by calling `VPU_EncClose()` function.

3.2.6. Dynamic Configuration Commands

While running picture encoding operation sequentially, sometimes applications need to give a certain special commands to VPU such as change of parameter, insertion of high layer header syntaxes(VUI or HRD), etc. The VPU API provides a set of commands to support these kinds of special requests from host applications.

Applications can change encoding options or high layer header syntaxes by calling VPU_EncGiveCommand() with ENC_SET_PARA_CHANGE.

Confidential StarFive Inc.

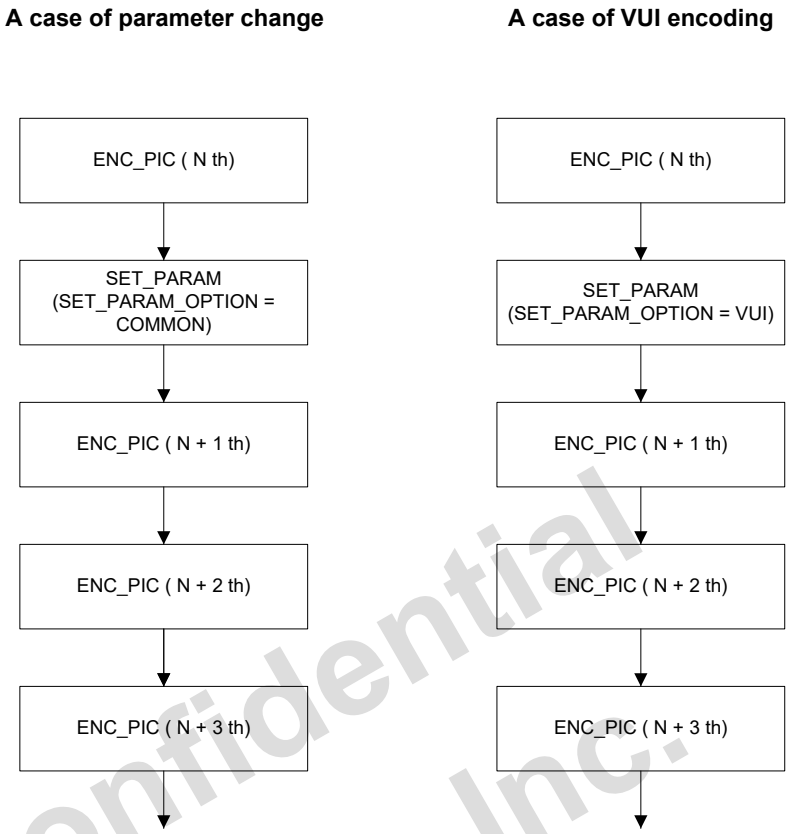


Figure 3.2. Change of Parameter While Encoding

The left figure of *Figure 3.2, "Change of Parameter While Encoding"* represents an example flow of encoding in which the command for adding a high level syntax (parameter change) is given in the middle of encoding.

After ENC_PIC #t command is issued, host processor gives VPU a new SET_PARAM command for changing encoding parameters. Then on ENC_PIC # t + 1,

- If implicit header encoding is 1, VPU decides high level syntax conforming to H.265/HEVC standard and inserts the SPS or PPS into stream.
- If implicit header encoding is 0, VPU reads high level syntax information from the code option register that is given by host processor and performs explicit header encoding.

Note In fact, SET_PARAM command only writes parameters to VPU internal memory, and actual header encoding with new parameters is done on next ENC_PIC command. If host processor wants to add a high level syntax without issuing a new SET_PARAM command, specify a header type (SPS or PPS or VPS) on code option register on ENC_PIC command.

Also the right figure of *Figure 3.2, "Change of Parameter While Encoding"* is an example of encoding with VUI. When host processor wants to put VUI information before or after encoded bitstream during encoding, VUI information should be written first on the relevant host interface register and the SET_PARAM command (w/ option = VUI) should be issued to VPU. Then on ENC_PIC # t + 1,

Confidential
Starfive Inc.

Confidential
StarFive Inc.

3.3. Other VPU Controls

3.3.1. Multiple Instances

In the multi-channel codec application, an instance is a handler to identify each task running a specific decoder or encoder, so it can seem that a number of tasks are running at the same time, even though VPU in fact allows these multiple tasks to take turns running in a time-sharing manner.

When host processor controls VPU by using VPU API layer the VPU API layer can handle the control of instance transition. Therefore host processor does not need to take care of anything regarding to multiple instances.

However in case that host processor controls VPU directly by using host interface registers, some global registers should be managed by host processor when instances are switched. Before host processor gives a new command, it needs to backup the content of host interface registers for previously executed instance and restore the content of host interface registers for an instance to resume.

3.3.1.1. Example of Running Instances

Figure 3.3, "Example Flow of Operating 2 Instances" shows when two instances are running how VPU serves them from one to another.

Confidential
StarFive Inc.

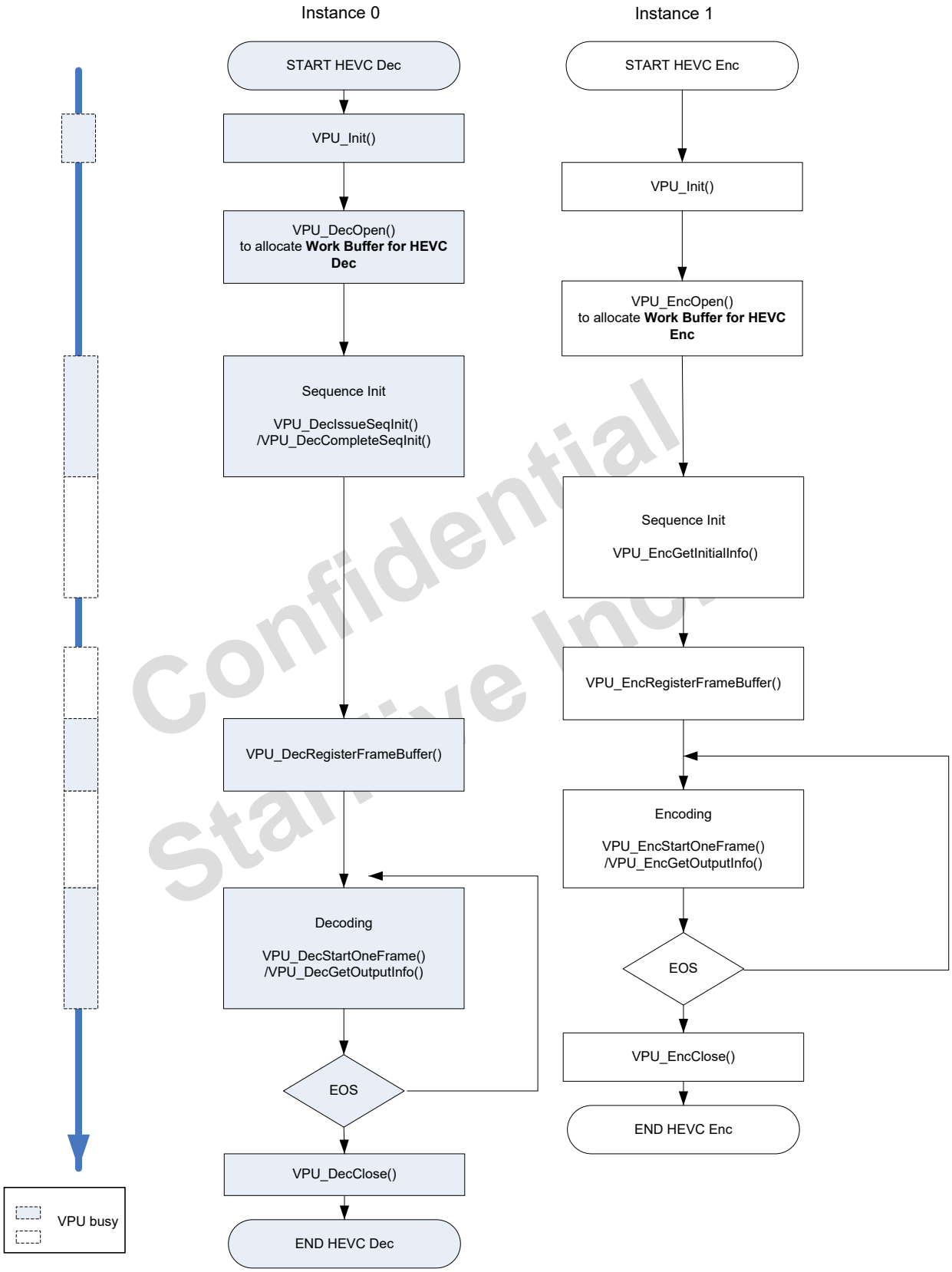


Figure 3.3. Example Flow of Operating 2 Instances

As an example, VPU basically can work with instances as the following sequence. Here let us say that instance 0 is HevcDec and instance 1 is HevcEnc.

1. In Instance 0, VPU_Init() downloads firmware from host processor to the internal program memory of VPU.
2. When VPU_Init() is issued, it is not executed for Instance 1, because VPU has already been initiated in Instance 0.
3. VPU_DecOpen() allocates a work buffer for Instance 0. The information of Instance 0 are saved in work buffer.
4. In the same manner with instance 0, VPU_EncOpen() allocates a work buffer for Instance 1.
5. Sequence Init is performed for Instance 0, which returns the information of image size and DPB number.
6. Sequence Init is performed for Instance 1, which returns the information of image size and DPB number.
7. VPU_DecRegisterFrameBuffer() registers frame buffers for Instance 0 with information of the frame buffer and mvCol buffer that host processor allocates.
8. VPU_EncRegisterFrameBuffer() registers frame buffers for Instance 1 with information of the frame buffer and mvCol buffer that host processor allocates.
9. Decoding a frame starts for Instance 0.
10. Encoding a frame starts for Instance 1.

The arrow pointing downwards in the left of *Figure 3.3, "Example Flow of Operating 2 Instances"* represents busy state of VPU. Two instances seem to take place in the same time. However, VPU in fact works for a single instance at the moment on a VPU command basis such as SEQ_INIT or PIC_RUN. So host application should make sure that they cannot call a VPU function in a VPU busy state. They can call another when previous function call is completed.

Note Chips&Media's reference software running on FPGA environment defines four instances for its maximum number of instances as the below code in vpuconfig.h. User can change it and need as much as memory size described in *Section 3.3.1.2, "Memory size for One Instance"* for each instance.

3.3.1.2. Memory size for One Instance

Please refer to the *Buffer Size* chapter in the Datasheet.

Confidential StarFive Inc.

3.3.2. Sequence Change

Host processor should set CMD_SEQ_CHANGE_ENABLE_FLAG to get informed about sequence change from VPU. If any of the following SPS information changes, VPU reports the sequence change to host processor through RET_SEQ_CHANGE_RESULT register.

- General profile idc
- pic_width_in_luma_sample
- pic_height_in_luma_sample
- sps_max_dec_pic_buffering_minus1
- sps_max_reorder_pics
- sps_max_latency_increase_plus1

Note In the VPUAPI, DecOutputInfo::sequenceChanged saves these information. For more details, please refer to the *WAVE410 API Reference manual*.

Host application using VPUAPI should follow the steps of *Figure 3.4, "Flow Diagram of Sequence Change"* when they detect DecOutputInfo::sequenceChanged has a non-zero value after VPU_DecGetOutputInfo() is called.

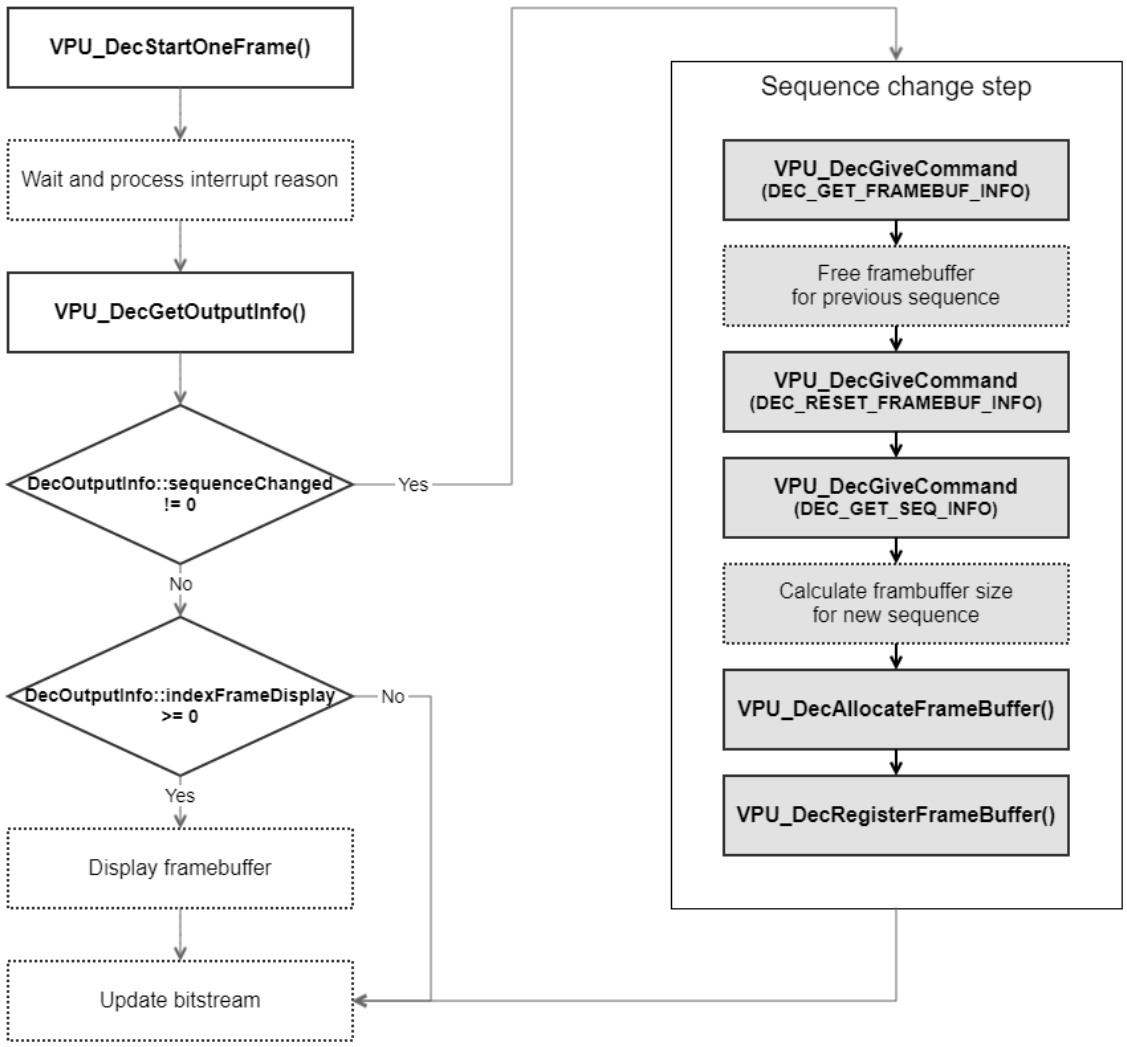


Figure 3.4. Flow Diagram of Sequence Change

Appendix A
ERROR DEFINITION

A.1. System Error

The following indicates system errors that might occur while INIT_VPU command is executed.

Table A.1. System Errors

Bit Pos	Bit Position (error_reason)	Description
0	0x0000_0001	Codec FW error
2	0x0000_0004	Coprocessor0 exception (INSTRUCTION_ACCESS_VIOLATION)
3	0x0000_0008	Coprocessor0 exception (PRIVILEDGE_VIOLATION)
4	0x0000_0010	Coprocessor0 exception (DATA_ADDR_ALIGNMENT_ERR)
5	0x0000_0020	Coprocessor0 exception (DATA_ACCESS_VIOLATION)
6	0x0000_0040	Coprocessor0 exception (WRITE_PROTECTION_ERR)
7	0x0000_0080	Coprocessor0 exception (INSTRUCTION_ADDR_ALIGNMENT_ERR)
8	0x0000_0100	Unknown error
9	0x0000_0200	Bus error
10	0x0000_0400	Double fault error
11	0x0000_0800	Reserved
12	0x0000_1000	HW access violation error
13	0x0000_2000	Reserved
14	0x0000_4000	Reverved
15	0x0000_8000	Watchdog time-out error
16	0x0001_0000	Reserved
17	0x0002_0000	Reserved
18	0x0004_0000	Reserved

Confidential
Star Five Inc.

Appendix B
POWER MANAGEMENT

For efficient power management, power-gating scheme is incorporated into the video processing unit (VPU) design. This paper describes how to suspend and resume VPU in a seamless, safe way when it powers down or up by host processor's power management policy. It also covers shortly the VPU_SleepWake function in VPUAPI and implementation with OS.

B.1. Introduction

There are two terms regarding power saving techniques, power gating and clock gating that reader might get confused with. Power gating is that VPU is not completely provided with power, while clock gating only limits the clock from being given to certain modules of VPU.

When VPU is turned off, registers and internal memory in V-CPU pmem/dmem are removed. Then V-CPU processor stops working and gets into the state ready for restart. This is basically how VPU responds to power off.

The following chapters show a little detail of how to control VPU safely at power down or up with some related codes.

Confidential
Starfive Inc.

B.2. At Power Down

1. Enable the VPU clock.

2. If VPU is running, wait until decoding or encoding operation is finished completely.

```
while (ReadVpuRegister(W4_VPU_BUSY_STATUS))
```

3. Send SLEEP_VPU command to let V-CPU flush the data of internal memory to external DRAM.

```
Wave4BitIssueCommand(core, W4_CMD_SLEEP_VPU);
```

4. Disable the VPU clock.

5. Power off the system.

B.3. At Power Up

1. Power up the system.

2. Enable the VPU clock source.

3. Reset VPU for returning to stable status since the power-up.

```
WriteVpuRegister(W4_VPU_RESET_REQ, 0x7ffffff) /* Reset All blocks */
```

4. Initialize the V-CPU Processor by issuing the INIT_VPU command so that VPU can be ready to get any command.

```
Wave4BitIssueCommand(core, W4_CMD_INIT_VPU);
```

5. Remap the code buffer.

6. Start the V-CPU Processor.

B.4. VPU_SleepWake() in VPUAPI

We offer VPU_SleepWake() function in the VPUAPI. But when target OS platform is Linux, device driver does directly handle this sort of function so that the VPU_SleepWake() function is not used. For implementation, host processor can refer to the linux driver code that we provide.

The VPU_SleepWake() function in VPUAPI can be used for the non OS platform or other OS platform where our application layer is able to handle a suspend/resume callback function.

B.5. Implementation with OS

On Linux OS platform, when VPU is suspended or resumed, power management module in kernel calls a callback function of device driver that has already been registered.

This is an example of implementation for power management in the Linux VPU device driver. Kernel scheduling API cannot be called in power management code.

Example B.1. Power Management Example Code in Linux Device Driver

```
static int vpu_suspend(struct platform_device *pdev, pm_message_t state)
{
	int i;
	int core;
	unsigned long timeout = jiffies + HZ;	/* vpu wait timeout to 1sec */
	int product_code;

	DPRINTK("[VPUDRV] vpu_suspend\n");

	vpu_clk_enable(s_vpu_clk);

	if (s_vpu_open_ref_count > 0) {
		for (core = 0; core < MAX_NUM_VPU_CORE; core++) {
			if (s_bit_firmware_info[core].size == 0)
				continue;
			product_code = ReadVpuRegister(VPU_PRODUCT_CODE_REGISTER);
		    if (PRODUCT_CODE_W_SERIES(product_code)) {
				while (ReadVpuRegister(W4_VPU_BUSY_STATUS)) {
					if (time_after(jiffies, timeout)) {
						DPRINTK("SLEEP_VPU BUSY timeout");
						goto DONE_SUSPEND;
					}
				}
				Wave4BitIssueCommand(core, W4_CMD_SLEEP_VPU);

				while (ReadVpuRegister(W4_VPU_BUSY_STATUS)) {
					if (time_after(jiffies, timeout)) {
						DPRINTK("SLEEP_VPU BUSY timeout");
						goto DONE_SUSPEND;
					}
				}
				if (ReadVpuRegister(W4_RET_SUCCESS) == 0) {
					DPRINTK("SLEEP_VPU failed [0x%x]",
					ReadVpuRegister(W4_RET_FAIL_REASON));
					goto DONE_SUSPEND;
				}
			} else {
				while (ReadVpuRegister(BIT_BUSY_FLAG)) {
					if (time_after(jiffies, timeout))
						goto DONE_SUSPEND;
				}

				for (i = 0; i < 64; i++)
				s_vpu_reg_store[core][i] =
				ReadVpuRegister(BIT_BASE+(0x100+(i * 4)));
			}
		}
	}

	vpu_clk_disable(s_vpu_clk);
	return 0;

DONE_SUSPEND:
```

```
	vpu_clk_disable(s_vpu_clk);

	return -EAGAIN;

}
static int vpu_resume(struct platform_device *pdev)
{
	int i;
	int core;
	u32 val;
	unsigned long timeout = jiffies + HZ;	/* vpu wait timeout to 1sec */
	int product_code;

	unsigned long	code_base;
	u32			code_size;
	u32			remap_size;
	int			 regVal;
	u32			hwOption  = 0;

	DPRINTK("[VPUDRV] vpu_resume\n");

	vpu_clk_enable(s_vpu_clk);

	for (core = 0; core < MAX_NUM_VPU_CORE; core++) {

		if (s_bit_firmware_info[core].size == 0) {
			continue;
		}

		product_code = ReadVpuRegister(VPU_PRODUCT_CODE_REGISTER);
	if (PRODUCT_CODE_W_SERIES(product_code)) {
			code_base = s_common_memory.phys_addr;
			/* ALIGN TO 4KB */
			code_size = (W4_MAX_CODE_BUF_SIZE&~0xfff);
			if (code_size < s_bit_firmware_info[core].size*2) {
				goto DONE_WAKEUP;
			}

			regVal = 0;
			WriteVpuRegister(W4_PO_CONF, regVal);

			/* Reset All blocks */
			regVal = 0x7ffffff;
			WriteVpuRegister(W4_VPU_RESET_REQ, regVal);	/*Reset All blocks*/

			/* Waiting reset done */
			while (ReadVpuRegister(W4_VPU_RESET_STATUS)) {
				if (time_after(jiffies, timeout))
					goto DONE_WAKEUP;
			}

			WriteVpuRegister(W4_VPU_RESET_REQ, 0);

			/* remap page size */
			remap_size = (code_size >> 12) & 0x1ff;
			regVal = 0x80000000 | (W4_REMAP_CODE_INDEX<<12) | (0 << 16) |
			(1<<11) | remap_size;
			WriteVpuRegister(W4_VPU_REMAP_CTRL,	regVal);
			WriteVpuRegister(W4_VPU_REMAP_VADDR,	0x00000000);	/* DO NOT CHANGE! */
			WriteVpuRegister(W4_VPU_REMAP_PADDR,	code_base);
			WriteVpuRegister(W4_ADDR_CODE_BASE,	 code_base);
			WriteVpuRegister(W4_CODE_SIZE,		  code_size);
			WriteVpuRegister(W4_CODE_PARAM,		 0);
			WriteVpuRegister(W4_INIT_VPU_TIME_OUT_CNT,   timeout);

			WriteVpuRegister(W4_HW_OPTION, hwOption);

			/* Interrupt */
			regVal  = (1<<W4_INT_DEC_PIC_HDR);
			regVal |= (1<<W4_INT_DEC_PIC);
			regVal |= (1<<W4_INT_QUERY_DEC);
			regVal |= (1<<W4_INT_SLEEP_VPU);
			regVal |= (1<<W4_INT_BSBUF_EMPTY);

			WriteVpuRegister(W4_VPU_VINT_ENABLE,  regVal);

			Wave4BitIssueCommand(core, W4_CMD_INIT_VPU);
			WriteVpuRegister(W4_VPU_REMAP_CORE_START, 1);
```

```
                        while (ReadVpuRegister(W4_VPU_BUSY_STATUS)) {
                                if (time_after(jiffies, timeout))
                                        goto DONE_WAKEUP;
                        }

                        if (ReadVpuRegister(W4_RET_SUCCESS) == 0) {
                                DPRINTK("WAKEUP_VPU failed [0x%x]",
                                ReadVpuRegister(W4_RET_FAIL_REASON));
                                goto DONE_WAKEUP;
                        }
                } else {

                        WriteVpuRegister(BIT_CODE_RUN, 0);

                        /*---- LOAD BOOT CODE*/
                        for (i = 0; i < 512; i++) {
                                val = s_bit_firmware_info[core].bit_code[i];
                                WriteVpuRegister(BIT_CODE_DOWN, ((i << 16) | val));
                        }

                        for (i = 0 ; i < 64 ; i++)
                                WriteVpuRegister(BIT_BASE+(0x100+(i * 4)),
                                s_vpu_reg_store[core][i]);

                        WriteVpuRegister(BIT_BUSY_FLAG, 1);
                        WriteVpuRegister(BIT_CODE_RESET, 1);
                        WriteVpuRegister(BIT_CODE_RUN, 1);

                        while (ReadVpuRegister(BIT_BUSY_FLAG)) {
                                if (time_after(jiffies, timeout))
                                        goto DONE_WAKEUP;
                        }

                }

        }


        if (s_vpu_open_ref_count == 0)
                vpu_clk_disable(s_vpu_clk);

DONE_WAKEUP:

        if (s_vpu_open_ref_count > 0)
                vpu_clk_enable(s_vpu_clk);

        return 0;
}
```

Appendix C
RATE CONTROL

WAVE encoder IP supports rate control which finds a quantization parameter (QP) adaptively to produce constant encoded bits. The rate control functions in three levels: picture level, CTU level, and subCTU level. The picture level RC determines a picture QP, the CTU level RC determines a 64x64 CTU QP, and the subCTU level RC determines a 32x32 block QP.

Besides normal rate control operation, WAVE encoder IP provides region of interest (ROI) rate control option in the picture level.

C.1. Picture-level RC

Figure C.1, "Diagram of Picture-level Rate Control" represents how the rate control is performed in the picture level.

Confidential StarFive Inc.

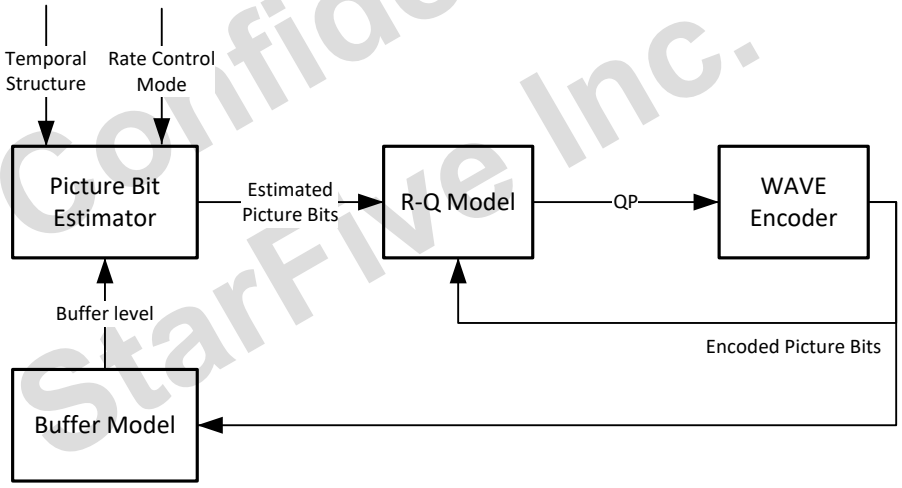


Figure C.1. Diagram of Picture-level Rate Control

A picture bit count is determined using buffer level, temporal structure, and rate control mode. R-Q model generates a QP with the determined picture bit count. WAVE encoder produces actually encoded picture bits which are fed both to R-Q model and to buffer model to refine the model parameters by comparing gap between estimated bits and encoded bits.

C.1.1. Buffer Model

The picture-level rate control is based on buffer model. The buffer model is controlled by three parameters: average bit rate, buffer size, and initial buffer level.

The buffer model size is determined by bitrate and InitialDelay which are sequence level parameters.

```
Buffer size = bitrate * InitialDelay
```

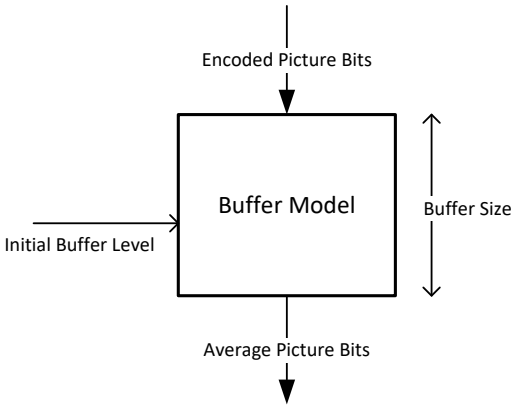


Figure C.2. Buffer Model

The buffer level is set to the initial buffer level at initial time. After encoding every picture, encoded bits are poured into the buffer, and average picture bits are drained from the buffer. The rate control controls QP so that the buffer does not suffer overflow or underflow.

C.1.2. Picture Bit Estimator

Picture Bit Estimator determines picture bits using buffer level, temporal structure, and rate control mode. When buffer level is around full, it decreases picture bits to prevent buffer overflow, while it increases picture bits when buffer level is around empty to prevent buffer underflow. By analyzing temporal structure, Picture Bit Estimator allocates more picture bits to more important or referenced pictures.

C.1.3. Rate Control Modes

There are three kinds of rate control modes: CBR, VBR and ABR. Difference of their behaviors comes from interpretation of buffer model. For CBR, buffer level is clipped to range [0, buffer size]. For VBR, buffer level clipping is the same as CBR but buffer size is amplified by ratio that is equal to [transmission bit rate/ average encoding bit rate]. ABR uses the same buffer size as CBR, but it does not use clipping of buffer level.

- CBR is good for application that requires tight end-to-end delay through constant bit rate channel, because it never suffers from buffer overflow or underflow and therefore CBR guarantees constant delay.
- VBR is good for application that transmission bit rate is higher than encoding bit rate such as internet environments. Because VBR uses lager buffer size than CBR, image quality is better than CBR.
- ABR is good for storage application where transmission bit rate is much higher than encoding bit rate. ABR shows best image quality than other modes but bitrate fluctuation is higher than other modes.

C.1.4. R-Q Model

R-Q model generates a picture QP by using R-Q equation inside. With the picture bit count from Picture Bit Estimator, the R-Q equation calculates a picture QP.

Also R-Q model gets feedback of actually encoded picture bits from Encoder and uses it to refine R-Q equation parameters for better QP estimation.

C.1.5. ROI

Host processor specifies foreground 64x64 blocks and their delta QPs. Then, R-Q model calculates the background QP so that picture bits summing all background bits and foreground bits should be equal to target picture bits. Each foreground QP is calculated by just adding the given delta QP to the background QP.

Confidential
Starfive Inc.

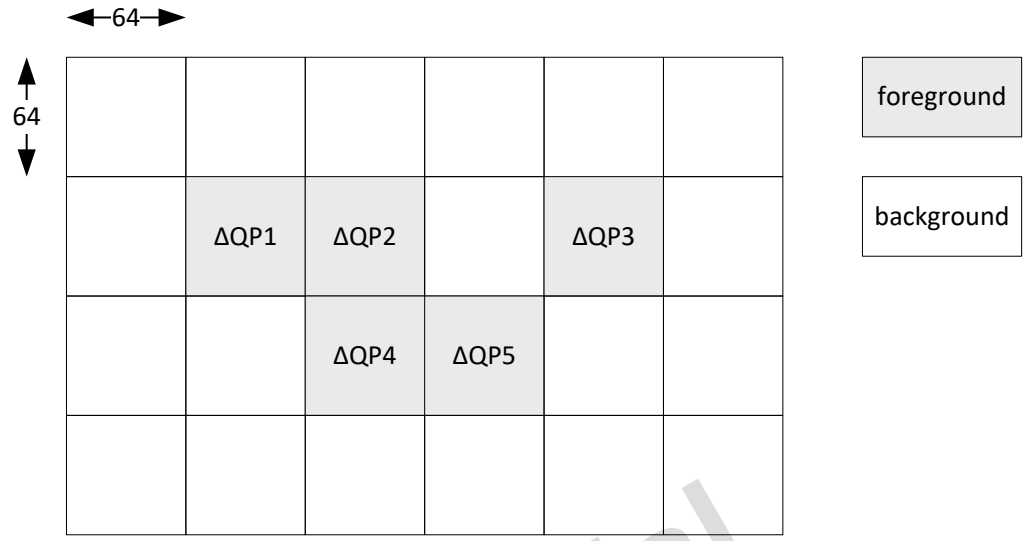


Figure C.3. ROI

C.2. CTU-level RC

The CTU-level RC can regulate the buffer level more accurately than the picture-level RC does. CTU target bit is derived from the following formula. QP keeps changing for every 64x64 block when CTU-level rate control is enabled.

```
CTU_target_bit = picture_target_bit / num_CTU_in_picture
```

Figure C.4, "CTU-level RC with RC Buffer" describes how CTU-level rate control works with RC buffer.

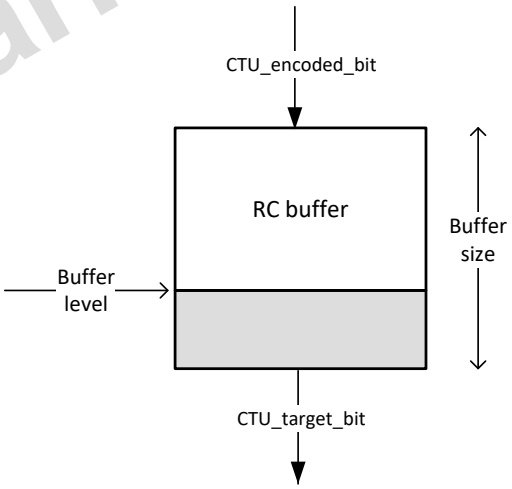


Figure C.4. CTU-level RC with RC Buffer

When CTU-level RC is enabled, CTU QP is decided based on the buffer level. CTU QP is linearly proportional to buffer level. In other words, CTU QP increases as buffer level goes up and CTU QP decreases as buffer level becomes low. The buffer level is controlled by accumulated encoded bit count of CTUs and complexity measure of CTUs.

Confidential StarFive Inc.

When buffer level is full, CTU QP is assigned with max. QP. When buffer level is empty, CTU QP is assigned with min QP. Max QP and Min QP are sequence level parameters.

With small buffer size, i.e. small end-to-end delay, rate control with only picture level has possibility that does not meet the small delay constraint. CTU-level rate control is the solution to solve this problem. However, the CTU-level RC might bring undesirable video quality than picture-level RC due to the frequent QP change.

This function can be enabled by setting the EN_CU_LEVEL_RC register.

C.3. subCTU-Level RC

When subCTU-level rate control is used, QP is changed for every 32x32 block. It allocates a high QP value to complex block and a low QP value to static block, because human eyes are more sensitive to static area than complex area. Block complexity is measured by hardware block in the VCE which calculates variance of an original 32x32 block.

The subCTU-Level RC aims to improve subjective video quality while keeping a constant bitrate. So, if this is used, SSIM score can be increased, but PSNR score might be decreased. Host processor can enable this by setting the relevant registers EN_HVS_QP and HVS_QP_SCALE.

C.4. Constrained VBR

Constrained VBR(CVBR) is another way of rate control to adapt scene change effectively. If CVBR is enabled, rate control operates in VBR mode (constant QP) in static scenes while it runs in CBR mode producing constant bit rates in complex scenes. The CVBR is very useful for surveillance application where there are mostly a lot of static or motionless scenes.

The CVBR is activated when MinQP is given with normal CBR setting. *Figure C.5, "Comparison between CBR and CVBR"* reveals how CBR and CVBR work differently in terms of bitrates and QP.

Confidential StarFive Inc.

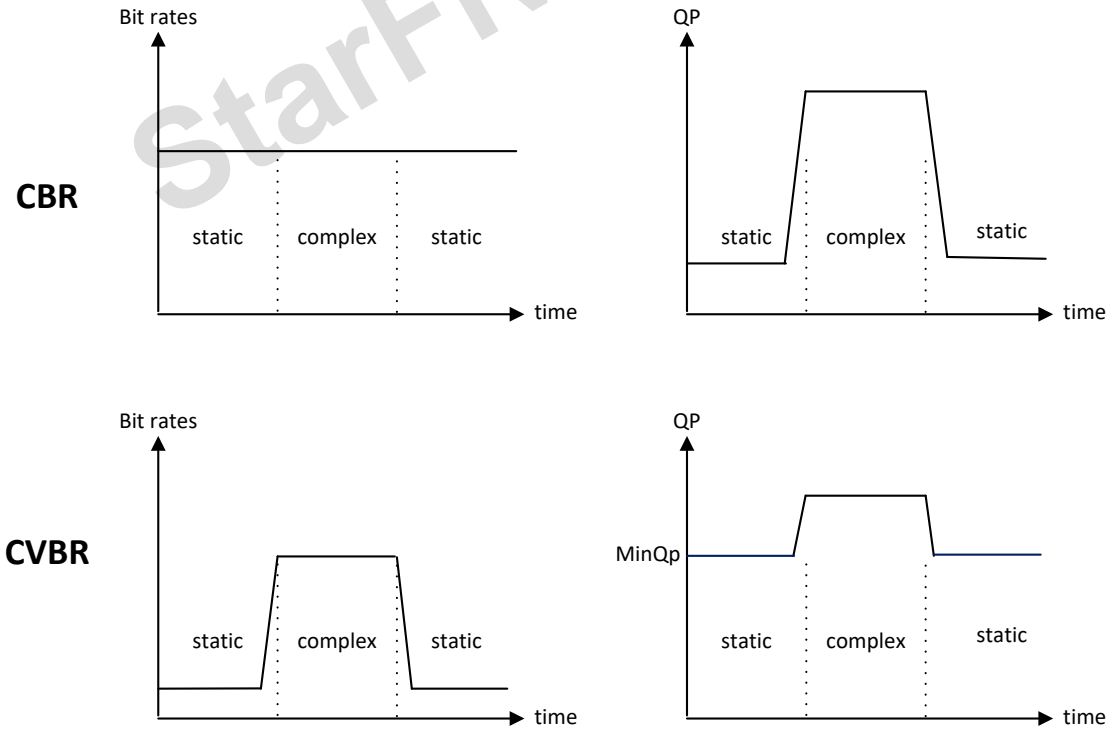


Figure C.5. Comparison between CBR and CVBR

Unlike CBR, CVBR can keep bitrate from being wasted in static scenes, because it does not allow QP to go below the specified MinQP.

C.5. Interface between Rate Control Levels

Figure C.6, "Interface between Rate Control Levels" describes each level of rate control and its interface on the architectural layer of WAVE encoder IP.

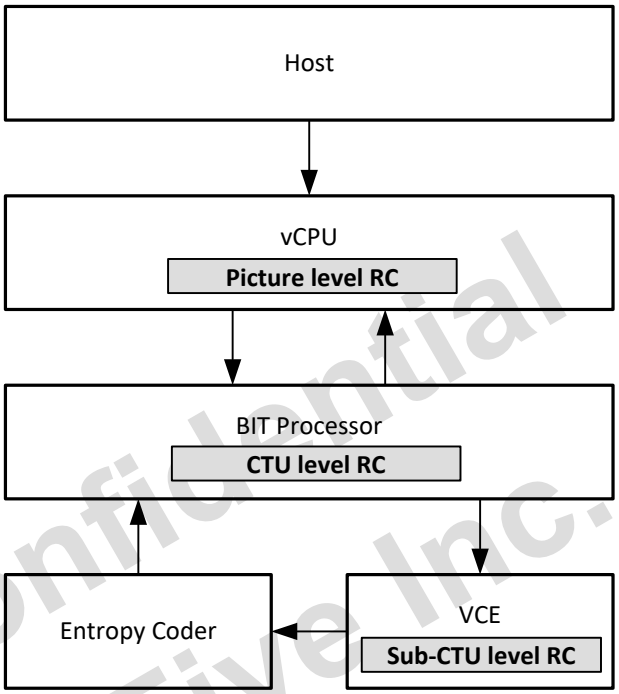


Figure C.6. Interface between Rate Control Levels

vCPU is a 32-bit processor that processes picture-level rate control. It is programmed using C code. BIT processor is a small programmable controller that processes CTU-level rate control. It is programmed using assembly code. VCE is the hardware unit that processes subCTU-level rate control and all pixel processing like motion estimation, transform, quantization and so on. Entropy coder is the hardware unit that processes syntax encoding.

C.5.1. Interface of Picture-level Rate Control

Table C.1. Interface Parameters from Host to vCPU

Parameter	Description
EN_RATE_CONTROL	Enables rate control.
EN_CU_LEVEL_RC	Enables CU-level rate control.
EN_HVS_QP	Enables HVS-based QP adjustment.
EN_HVS_QP_SCALE	Enable QP scaling for HVS_QP.
HVS_QP_SCALE	Specifies QP scaling factor for HVS_QP.
INIT_BUF_LEVELx8	Specifies encoder initial buffer level.
INITIAL_DELAY	Initial CPB delay in milliseconds (RC buffer size = INITIAL_DELAY * TRANS_RATE)
TARGET_RATE	Target encoding bitrate in bps

Confidential StarFive Inc.

Parameter	Description
TRANS_RATE	Peak transmission bitrate in bps
MIN_QP Minimum	QP for rate control
MAX_QP Maximum	QP for rate control
INTRA_QP_OFFSET	QP offset of intra picture
INTRA_PERIOD	Period of intra picture
GOP structure Information	GOP structure information such as GOP period, reference pictures and slice types and so on.
EN_SEQ_ROI	Enables ROI in sequence level.
ROI_ENABLE_FLAG	Enables ROI in picture level.
ROI_DELTA_QP	Delta QP of a CTU block is calculated by ROI_DELTA_QP multiplied by a value in the CTU position in ROI CTU map.
ROI_CTU_MAP_ ENDIAN	Endian of ROI CTU map
ROI_STRIDE	Stride of ROI CTU map
ROI_ADDR_CTU_MAP	Start address of ROI CTU map

Table C.2. Interface Parameters from BIT Processor to vCPU

Parameter	Description
Avg_qp	Average QP for a picture. It is used for R-Q parameter update in picture-level RC.
Pic_bit	Bit count of a picture. It is used for R-Q parameter update and RC buffer level update in picture-level RC.

Confidential
Chips&Media

C.5.2. Interface of CTU-Level Rate Control

Table C.3. Interface Parameters from vCPU to BIT processor

Parameter	Description
Picture_QP	QP from picture-level RC
Min_QP	MIN_QP from host processor
Max_QP	MAX_QP from host processor
Hrd_buf_size	RC buffer size of picture-level RC
Hrd_buf_level	RC buffer level of picture-level RC
Picture_target_bit	Picture target bit calculated from picture-level RC
ROI information	ROI map and delta QP

Table C.4. Interface Parameters From Entropy Coder to BIT Processor

Parameter	Description
CTU_rd_cost	RD cost of a CTU. It is used for complexity measure in CTU-level RC. VCE generates it which bypasses Entropy Coder and is delivered to BIT processor.
CTU_var	Variance of a CTU. Averaging CTU_var of all CTUs in a picture is Pic_var. VCE generates it which bypasses Entropy Coder and is delivered to BIT processor.

Parameter	Description
Ctu_bit	Bit count of a CTU. It is used for buffer level update in CTU-level RC.

C.5.3. Interface of subCTU-level Rate Control

Table C.5. Interface Parameters From BIT processor to VCE

Parameter	Description
CTU_QP	QP for each CTU
En_HVS_QP	Enable HVS QP.
Pic_var	Average variance of a picture

C.6. Porting Customer's Rate Contol to WAVE Encoder

C.6.1. Porting Picture-level Rate Control

Figure C.7, "Porting Picture-level Rate Control" shows two ways of porting customer's picture-level rate control to the IP.

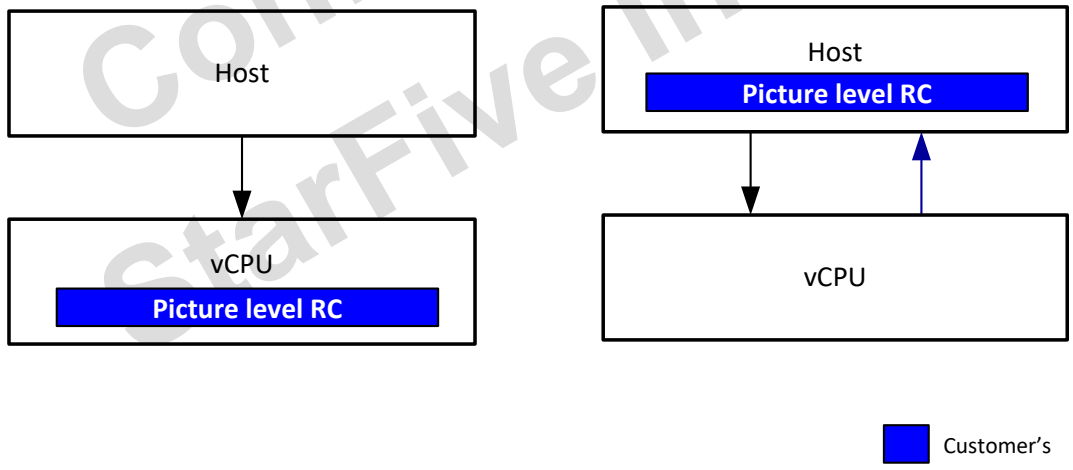


Figure C.7. Porting Picture-level Rate Control

- One is to implement the algorithm in vCPU. Customer delivers their algorithm and Chips&Media ports it to vCPU.
- The other is to implement the algorithm in host processor instead of vCPU. Host processor calculates picture QP using parameters from vCPU.

C.6.2. Porting CTU-level Rate Control

Host processor cannot be responsible for CTU-level rate control, because CTU-level interaction between host processor and the IP causes a lot of performance burden. *Figure C.8, "Porting CTU-level Rate Control"* depicts two feasible ways of porting customer's CTU-level rate control to the IP.

Confidential
StarFive Inc.

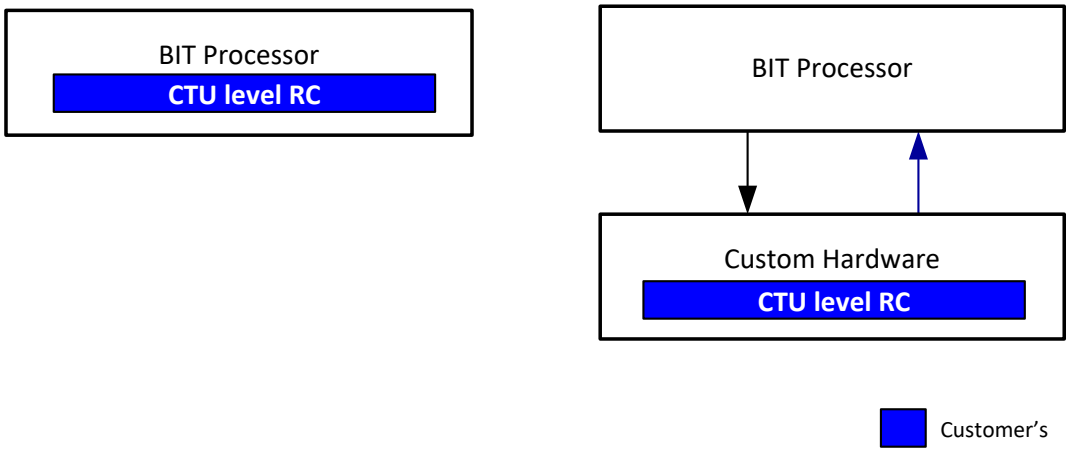


Figure C.8. Porting CTU-level Rate Control

Customer delivers their algorithm and Chips&Media ports it to BIT processor assembly code. If BIT processor is not fast enough to run the algorithm, custom hardware block for the algorithm might be necessary.

C.6.3. Porting subCTU-level Rate Control

There are two options for porting subCTU-level rate control to WAVE: designing custom hardware or implementing it in BIT processor. To implement it in BIT processor, the algorithm should be simple enough for BIT processor to run on time.

C.7. Quality Evaluation with C model

Chips&Media provides C model (.exe) that executes bit-exact operation as WAVE IP. With the C model, customer can simulate and evaluate their rate control algorithm before the algorithm is really ported to the IP.

There are two possible ways of allocating job and role between Chips&Media and customer for evalution of customer-specific RC algorithm on the WAVE C model.

- Customer explains their algorithm to Chips&Media. Then Chips&Media implements the algorithm and delivers an executable file to the customer.
- Chips&Media delivers the C code library for WAVE IP which excludes rate control. Then customer implements the C code for their rate control and builds an executable file by linking the C library with their C code. To work that way Chip&Media needs to get prototypes of rate control functions from customer.

Confidential Starfive Inc.

Confidential
StarFive Inc.

About Chips&Media

Chips&Media, Inc. is a leading video IP provider, headquartered in Seoul, South Korea. The company was established in 2003 by leading members with years of profound experiences in video standard technologies and the semiconductor industry. Our extensive catalogue of IP solutions includes video postprocessing, video frame buffer compression as well as video codecs covering vast range of video standards from MPEG-2, MPEG-4, H.263, Sorenson, H.264, VC-1, AVS/AVS+, VP8/9, HEVC(H.265), and AV1 for HD to UHD (4K/8K) resolution.

Chips&Media has been developing a line of reliable, high-quality IP solutions that allow our customers and partners to satisfy the growing consumer demand for high-performance multi-media digital devices. Especially as a leading multi-standard video codec solution provider, we have been providing our advanced ultra-low power multi-codec video IPs to top-tier semiconductor companies.

With a mission to become a global top SoC IP provider, Chips&Media has introduced Image Signal Processing (ISP) IP and Deep learning-based super resolution IP to the market and continues to widen our solution portfolio to cope with growing demand on image processing and related solutions. To find further information, please visit the company's web site at *http://www.chipsnmedia.com*