

H264 Encoder Codec

API Specification

12/20/2017

Revision 2.2

© 2017 SOC Technologies Inc.

SOC is disclosing this user manual (the "Documentation") to you solely for use in the development of designs to operate with SOC hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of SOC. SOC expressly disclaims any liability arising out of your use of the Documentation. SOC reserves the right, at its sole discretion, to change the Documentation without notice at any time. SOC assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. SOC expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. SOC MAKES NO OTHER WARRANTIES, WHETHER EXPRESSED, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL SOC BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© 2008-2017 SOC, Inc. All rights reserved.

SOC, the SOC logo, the Brand Window, and other designated brands included herein are trademarks of SOC, Inc.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision	Author
01/13/2017	SOC initial Release. Based on: H264 Encoder API v1_6.docx	1.0	Blake
04/27/2017	Corrected register 0x56	1.7	Blake
05/07/2017	Added Aspect Ratio Table	1.8	Chad
07/07/2017	Added CAVLC/CABAC switch	1.9	Mulong

09/14/2017	Added default values	2.0	Blake
09/26/2017	Added 8K support, register 0x63	2.1	Blake
12/20/2017	Added Crop register 0x68 and 0x69	2.2	Blake

Introduction

The API interface has 256 register (16 bit for writing and 32 bits for reading). Each register has an address to allow user-access. The registers accept command code for controlling the operation of the features within the FPGA. The registers may be read to acquire status and debugging information.

Some registers are allocated for user configurations and the rest are reserved for system functions such as debugging. The following table(s) specifies the address, definition, and access right (read/write) for the user-accessible registers. Note: registers of any API_TYPE that are NOT in the table are not designed for user access. If such a register is written to, the decoder may stop running. To recover, the core must be re-downloaded into the FPGA.

Control the API inside the FPGA

If it is necessary to control the API via the modules inside the FPGA, the ports of the API are available. Details, including the clock frequency, are provided in the corresponding integration manual.

Control the API outside the FPGA

If it is necessary to control the Encoder from outside the FPGA, the address, data, and R/W pins can be routed to the FPGA I/O pins. The desired FPGA I/O pins for the API may be selected or the SOC default pins may be used. An UART interface is available, upon request. The interface provides access to the API via `uart_tx` and `uart_rx` pins.

Default values

For writable register, default value is the value after power on. There is no default value for the states and information registers.

All default values depend on the build version and may be modified without notification.

H264 CODEC Registers

Addr (HEX)	Information	Access
92	Bit[2:0]: Multi-Channel/Time Sharing API Core Selection <ul style="list-style-type: none"> - 0 - Selects all cores for writing - 1-7 = Core # - Note only 1 core can be read at a time. A value of 0 should not be used for reading registers. - Default value: 0 	R/W

The following registers are unique to each time-shared core (when used)

0	Bit[0]: Enable H264 Encoder (0=Pause, 1=Run) Default value: 1	R/W
1	Bit[7:0]: H264 Profile Default value: 122(0x7A)	R/W
2	Bit[7:0]: H264 Level	R/W

	Default value: 51	
3	Bit[1:0]: Chroma Format - "01" = 4:2:0 - "10" = 4:2:2 - Default value: 01	R/W
4	Bit[1:0]: Bit Depth - "00" = 8 bit - "10" = 10 bit (Requires Special Licence). - Default value: "00" This is supported in only version higher than 10-bit version but not in 8-bit version. Please check your order information.	R/W

10	Bit[0]: CABAC/CAVLC switch <ul style="list-style-type: none"> - 0 = CAVLC - 1 = CABAC - This is supported in only version with CABAC support. - Default value: '0' 	R/W
17	Bit[0]: Constrained Intra Prediction <ul style="list-style-type: none"> - '1' = Intra prediction formed from Intra Neighbour blocks only - '0' = Intra prediction formed form any block type With '1', the refresh speed is faster but bit-rate higher.	R/W
	Default value: '0'	
23	Bit[0]: Disable Deblocking Filter <ul style="list-style-type: none"> - '1' = Disabled - '0' = Enabled - * Some build may have the deblocking feature permanently disabled - Default value: '0' *This is just a filtering, so there is no big difference in most case compared to without the filtering.	R/W

28	<p>Maximum skip control (for latency control)</p> <p>Bit[15]: Skip control type.</p> <ul style="list-style-type: none"> - '0' = MB row based, each MB row is 16 pixel rows. - '1' = MB based type. Each MB is a block of 16x16 pixels. <p>Bit[7:0] = maximum skipped MB rows and used when bit[15]='0'.</p> <p>Bit[14:0] = maximum skipped MBs and used when bit[15] = '1'.</p> <p>Note: If maximum skipped is set to 0, all possible are skipped.</p> <p>Default value: 0x8000</p>	R/W
29	<p>Bit[7:0] = Aspect Ratio See Aspect Ratio Table</p> <p>Default value: 0x01</p>	R/W
32-3F	<p>See Note 1</p> <p>Default GOP: One Intra frame + 15 P frames</p>	R/W

The following registers are shared between all cores even when multi/time-sharing cores are used

52	<p>Bit[15:8]: Random Intra Block Size X</p> <p>Bit[7:0]: Random Intra Block Size Y</p> <ul style="list-style-type: none"> - 1 Block size is 16x16 pixels - This register only has an effect when 'Random Intra Mode 0x5B' is enabled. <p>Default value: 0x0703</p>	R/W
55	<p>Bit[6:0]: Intra MB qP adjustment, two's complement value.</p> <p>*Negative value makes the I-block and I-slice better quality.</p> <p>*Use this register along with register 0x56 together.</p> <p>Default value: 0x7C (-4 in decimal)</p>	R/W
56	<p>Bit[5:0]: Maximum intra MB qP.</p> <p>Bit[13:8]: Minimum intra MB qP.</p> <p>*Use this register along with register 0x55 together.</p> <p>Default value: 0x062A. Max_qP=42, Min_qP=6</p>	R/W
5B	<p>Bit[0]: Random Intra Mode Enable</p> <ul style="list-style-type: none"> - '1' = Enabled - '0' = Disabled 	R/W

	<ul style="list-style-type: none"> - When Enabled and not Intra frames exist, 'Constrained Intra Prediction 0x17' must be enabled for this to function as intended. <p>Default value: 0 (Disabled)</p>	
5D	<p>Bit[0]: Enable Interlaced Encoding.</p> <p>'1', The encoder use interlacing when the input is interlaced.</p> <p>'0', The encoder do not use interlacing even the input is.</p> <p>Bit[1]: Force Encoding as Interlaced</p> <p>'1', The encoder use interlacing even the input is not.</p> <p>'0', The encoder is controlled by bit[0] only.</p> <p><i>* The interlaced encoder is not supported by input with scalar.</i></p> <p>Default value: 0x01. (Interlaced is not forced but enabled)</p>	R/W
5F	<p>Bit[7:0]: Disable Video Channel</p> <ul style="list-style-type: none"> - Can Disable(1)/Enable(0) each video encoding channel - <i>*For Time-Sharing Encoder Core only*</i> - Example - "00001010" = Cores 1 and 3 are disabled - Note bit 0 has no effect <p>Default value: 0x0. All channels are enabled.</p>	R/W
61	Bit[31:0]: H264 Build Date	R

62	Bit[31:0]: H264 Build Time Stamp	R
----	----------------------------------	---

63	Bit[27:20]: 0x80 for 8K encoder 0x40 for 4K encoder 0x20 for 2K or HD encoder	
64	H264 Encoder core production number(used by SOC only) Value: 0x53000008 for 2K-8-bit encoder Value: 0x5300000A for 2K-10-bit encoder	R
65	H264 git version (used by SOC only) Example: 20160602	R
68	crop_top_row, default 0. Note: see note in next register crop_bottom_row.	R/W
69	crop_bottom_row, default 0. Note: The H264 encoder crops pixel rows on top edge and bottom edge of the original input frame with these two registers. The final encoded frame height would be: Final Encoded Frame Height = Input Height - crop_top_row - crop_bottom_row. The input frame height is final in register 0xEF, named "Encoded Height" in manual "API - Encoder System API.pdf". When any one of the two registers is zero, then no cropping on the regarding edge.	R/W

70	Bit[15]: use_I_frame_balace, Only when use_I_frame_balace = '1', the bit[2:0], Intra_frame_scale takes effect.	R/W*
----	--	------

	<p>Bit[14]: Reserved, don't change, W1RB*</p> <p>Bit[13]: Reserved, don't change, W1RB*</p> <p>Bit[12]: control_i_slice: control bit-rate for I slice. When '1', the I slice is not in countered in the bit-rate control.</p> <p>Bit[11]: control_Intra_mb_adj, W1RB*, default '1'</p> <p>Bit[10]: Reserved, don't change, W1RB*</p> <p>Bit[9]: control_Inter_mb_adj, W1RB*, default '1'</p> <p>Bit[8]: Reserved, don't change, W1RB*</p> <p>Bit[7]: Reserved, don't change, W1RB*</p> <p>Bit[6]: Reserved, don't change, W1RB*</p> <p>Bit[5]: Reserved, don't change, W1RB*</p> <p>Bit[4]: Reserved, don't change, W1RB*</p> <p>Bit[3]: Reserved, don't change, W1RB*</p> <p>Bit[2:0]: Intra_frame_scale, the I frame size scale than the P frame when use_I_frame_balace='1'.</p> <p>*Basically, the default value should be the best configuration But the user is allowed to change it.</p> <p>Default value: 0xAFD6.</p> <p>*To change I-slice quality, use register 0x55.</p>	
79	Bit[6:0]: current macroblock Qp value. Informative.	R

7B	Bit[5:0]: Minimum QP Value *Note: There is no register for Max-QP value because it is not necessary to have one. Default value: 6.	R/W
----	--	-----

7F	<p>Bit[15]: Intra(I) Constant QP Mode</p> <p>Bit[14:8]: Intra(I) Constant QP Value-26</p> <p>Bit[7]: Inter(P) Constant QP Mode</p> <p>Bit[6:0]: Inter(P) Constant QP Value-26</p> <ul style="list-style-type: none"> - Bit[15] / Bit[7]: Write '1' to the bit to clear the bit. <p>When reading, if bit[15]='1', the intra block uses bit[14:8]+26 as QP vlue.</p> <p>When reading, if bit[15]='0', the encoder use dynamic QP controlled by register 0x97 to encode intra blocks.</p> <p>When reading, if bit[7]='1', the inter block uses bit[6:0]+26 as QP vlue.</p> <p>When reading, if bit[7]='0', the encoder use dynamic QP controlled by register 0x97 to encode inter blocks.</p> <p>For example: write 0515H to this register, the register read value will be: 8595H. So, the encoder use QP=31(=26+5) to encode intra(I) blocks, and QP=47(=26+0x15) to encode inter(P) blocks.</p>	R/W
----	--	-----

	<p>To clear the constant QP, write '1' to bit[7] or bit[15].</p> <p>For example, write 8080H to this register, the register read value will be: 0000H, means the constant QP was cleared.</p> <ul style="list-style-type: none"> - Bitrate control must be disabled for proper operation (0x97=0) - If failed to change this value, try to write again until succeed. <p>Default value: 0x0000. The qP is controlled by bit-rate. See system register 0x97.</p>	
--	---	--

Note:

- (1) All the default values depend on the build version. So you may have different values for defaults.
- (2) W1RB*: Write 1 Reverse Bit. The bit is reversed by writing logic '1'.

H264 Profiles:

- Baseline: 66
- Main: 77
- High: 100
- High 10: 110

- High 4:2:2: 122

Level:

The level is dependent on the Frame Size, frame rate, and encoding bit rate. The level is used to ensure the Decoder is capable of decoding the encoded stream. A higher level can be used even if the maximum rate is not required. Refer to Annex A of ITU-T H264.

The Level parameter is a decimal value. The register is this value without the decimal place. Ex. 4.2 would be written as 42.

Recommended Level support:

- 3.2 – Up to 720p60
- 4.2 – Up to 1080p60
- 5.1 – Up to 4k30

Note1:

The registers 32~3F control the encoding frame type or the Group Of Pictures (GOP). The encoder scans the 14 registers in order before encoding next frame/frames/unit.

For each register in the range of 32-3F:

Bit[15:8] indicates the repeat time of the frame type

Bit[7:0] indicates the unit/frame/sequence type as below.

0x00 - NULL. The encoder will skip to next register without encode any unit/frame.

0x67 - Sequence Parameter Set (SPS unit)

0x68 - Picture Parameter Set (PPS unit)

0xE5 - Intra IDR Picture (I-Frame)

0x65 - Temporary Intra IDR Picture Request - Inserts only 1 Intra Picture. Will be cleared to 0x00 after the Intra frame has been coded.

0x45 - Intra (non-IDR) Picture

0x41 - Inter (P) Picture

0xC1 - Inter (P) Picture sequence with Random Intra-Row P frame sequence.

Random Intra-Row P frame sequence (type 0xC1)

Generally, the Random Intra-Row P frame sequence (called I-slice mode) is for smooth bit rate and low latency purpose. In case of regular I frame, the I frame size may be much bigger than the expected P frame size. With the I-slice mode, the bit-rate will be more constant and so the latency is lower.

Here is the detail for I-slice mode. When the encoder encounters a 0xC1 type sequence, it put one I-slice row (16 pixel height in vertical) in each P frame of the sequence, with different

vertical position for different frame. After a number (depending the frame size) of P frames, all vertical position would be recovered by I-slice so that the entire picture get recovered.

Encoder running sequence

Here is how the encoder runs. The encoder reads the current position and finishes it. For the example above:

1. The encoder reads the position 32 (which is 0067, a sequence parameter), and then the encoder encodes a sequence parameter unit.
2. The encoder reads the position 33(which is 0068, a picture parameter), and then encodes a picture parameter unit.
3. The encoder reads the position 34 which is 00E5, and start encoding the next frame as I frame.
4. The encoder reads the position 35, and begins to encode 15 P frames.
5. The encoder reads position 36 which is 0000, so no frame for this position.
6. The encoder reads position 37 until 3F, all 0000, so no frame or unit encoded for these positions.
7. The encoder backs to 32 and repeats the steps 1-6.

While the encoder is encoding any frame, user could change any of the 14 registers. But until the encoder reads the changed position again, it does not take effect.

Examples

Example 1: The regular GOP:

	Address(hex)	Value(hex)
32	0067	
33	0068	
34	00E5	
35	0F41	
	36-3F	0000

The GOP sequence will be like this:

SPS + PPS + I-Frame + 15 P frames

Example 2: The GOP with I slice mode:

	Address(hex)	Value(hex)
32	0067	
33	0068	
34	0000	
35	00C1	
	36-3F	0000

The GOP sequence will be like this:

SPS + PPS + N * (P with I slice) frames.

Where N = frame height in pixel / 16

Aspect Ratio Table

Table E-1 – Meaning of sample aspect ratio indicator

aspect_ratio_idc	Sample aspect ratio	(informative) Examples of use
0	Unspecified	
1	1:1 ("square")	1280x720 16:9 frame without horizontal overscan 1920x1080 16:9 frame without horizontal overscan (cropped from 1920x1088) 640x480 4:3 frame without horizontal overscan
2	12:11	720x576 4:3 frame with horizontal overscan 352x288 4:3 frame without horizontal overscan
3	10:11	720x480 4:3 frame with horizontal overscan 352x240 4:3 frame without horizontal overscan
4	16:11	720x576 16:9 frame with horizontal overscan 528x576 4:3 frame without horizontal overscan
5	40:33	720x480 16:9 frame with horizontal overscan 528x480 4:3 frame without horizontal overscan
6	24:11	352x576 4:3 frame without horizontal overscan 480x576 16:9 frame with horizontal overscan
7	20:11	352x480 4:3 frame without horizontal overscan 480x480 16:9 frame with horizontal overscan
8	32:11	352x576 16:9 frame without horizontal overscan
9	80:33	352x480 16:9 frame without horizontal overscan
10	18:11	480x576 4:3 frame with horizontal overscan
11	15:11	480x480 4:3 frame with horizontal overscan
12	64:33	528x576 16:9 frame without horizontal overscan
13	160:99	528x480 16:9 frame without horizontal overscan
14	4:3	1440x1080 16:9 frame without horizontal overscan
15	3:2	1280x1080 16:9 frame without horizontal overscan
16	2:1	960x1080 16:9 frame without horizontal overscan
17..254	Reserved	