

# SOC H264 Decoder IP Core

## API Specification

© 2016 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.

© 2008-2016 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
01/27/2016	SOC initial Release.	1.0	Chad
04/05/2017	Register Corrections	1.1	Chad

## 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 Decoder 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.

**Table-1 H264 Core API registers description**

<b>Addr (HEX)</b>	<b>Information</b>	<b>Access</b>
00	Bit[3]: Trick Mode Bit[1]: Pause Bit[0]: Reset	R/W

01	Bit[4]: Bit Depth (0=8bit, 1=10bit) Bit[3]: Compression Type (0=MPEG2, 1= H264) Bit[2]: Chroma Format (0=4:2:0, 1=4:2:2) Bit[1]: Aspect Ratio (0=4:3, 1=16:9) Bit[0]: Color Space (0=601, 1=709)	R
02	Bit[13:12]: Frame Format - 13 = '1' = MBaff (Adaptive Frame Field Coding) - 12 = '1' = Field Coding (Interlaced), '0' = Progressive Bit[8:0]: Frame Rate	R
04	Bit[11:0]: Horizontal Resolution	R
06	Bit[11:0]: Vertical Resolution	R
07	Bit[7:0]: Profile IDC	R
08	Bit[7:0]: Level IDC	R
09	Bit[7]: Pipeline Reset Bit[6]: Sequence Parameter Set Present Bit[5]: Picture Parameter Set Present Bit[4]: Byte Stream Read Enable Bit[3]: Macroblock Controller Idle Bit[2]: DDR Frame Not Available Bit[1]: Byte Stream Fifo Empty Bit[0]: Decoding	R
0A	Bit[5]: Deblocking Buffer Ready Bit[4]: Macroblock PL Input Buffer Ready Bit[2]: Decoded Macroblock Buffer Ready Bit[1]: Temporal Macroblock Ready Bit[0]: Spatial Macroblock Ready	R
0E	Bit[31:0]: Decoded Frame Count	R
0F	Bit[31:28]: Max Ref Line Index Bit[27:24]: Max Macroblock PL Buffer Index Bit[23:20]: Max Resolution Support - 6 = 1k (1024x1024) - 7 = 2k (2048x2048) - 8 = 4k (4096x4096) Bit[20:16]: Max Bit Depth Supported	R

	Bit[5]: Intra 4x4 Support Enabled Bit[4]: Deblocking Support Enabled Bit[3]: Dual DDR Reference Support Enabled Bit[2]: Weighted Prediction Support Enabled Bit[1]: CABAC Support Enabled Bit[0]: B-Frame Support Enabled	
10	Bit[31:0]: Build Date	R
11	Bit[31:0]: Build Time	R
12	Bit[31:16]: Frame Expired Count Bit[15:0]: Frame Error Count	R
13	Bit[31:16]: Pipeline Timeout Count Bit[6:4]: Current Frame Type Bit[3]: Scalable Frame Present Bit[2]: B Frame Present Bit[1]: P Frame Present Bit[0]: I Frame Present	R
14	Bit[31:0]: Stream Bytes Read	R