

SOC VoIP-X-4K Network API Specification

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.

© 2008-2020 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 (MM/DD/YYYY)	Version	Notes
06/24/2020	1.0	SOC initial Release

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 CODEC 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_RX and UART_TX pins.

SOC Network status and control registers

(w) : writable bit(s)

(t) : toggle write bit(s) – write ‘1’ to toggle bit value

Note: Any register bit not listed is reserved for internal use only. All register bits are read-only unless marked with (w) or (t).

Addr (hex)	Information	Default (hex)	Access															
A0	<p>TX Target IP address is divided into 4 bytes (see example in the table below)</p> <p>Read: Bit[31:0]: TX Target IP address [31:0] (Byte 3 ~ Byte 0)</p> <p>Write: Bit[9:8](w): TX Target IP address byte index (3~0) Bit[7:0](w): TX Target IP address byte</p> <table border="1" style="margin-left: 40px;"> <tr> <td>Target IP Address</td> <td colspan="4">192.168.1.101</td> </tr> <tr> <td>Byte index</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>Target IP byte</td> <td>C0</td> <td>A8</td> <td>01</td> <td>65</td> </tr> </table>	Target IP Address	192.168.1.101				Byte index	3	2	1	0	Target IP byte	C0	A8	01	65	<p>Encoder: C0A80165</p> <p>Decoder: C0A80164</p>	R/W
Target IP Address	192.168.1.101																	
Byte index	3	2	1	0														
Target IP byte	C0	A8	01	65														
A1	Reserved	00000000	RO															
A2	Reserved	00000001	R/W															
A3	Reserved	00000000	RO															
A4	Bit[15:0](w): TX Target UDP Port.	000004D2	R/W															
A5	Bit[15:0](w): TX Packet Size (Bytes)	00000578	R/W															

Addr (hex)	Information	Default (hex)	Access																					
A6	<p>MAC address is divided into 6 bytes (see example in the table below)</p> <p>Read: Bit[31:0]: MAC address[31:0] (Byte 3 ~ Byte 0)</p> <p>Write: Bit[10:8](w): MAC address byte index (5~0) Bit[7:0](w): MAC address byte</p> <p>A Network Soft Reset is required for MAC address change (See Addr 0xAF)</p> <table border="1" data-bbox="280 785 1110 993"> <tr> <td>MAC Address</td> <td colspan="6">BA : 98 : 76 : 54 : 32 : 10</td> </tr> <tr> <td>Byte index</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>MAC byte</td> <td>BA</td> <td>98</td> <td>76</td> <td>54</td> <td>32</td> <td>10</td> </tr> </table>	MAC Address	BA : 98 : 76 : 54 : 32 : 10						Byte index	5	4	3	2	1	0	MAC byte	BA	98	76	54	32	10	Vary by board	R/W
MAC Address	BA : 98 : 76 : 54 : 32 : 10																							
Byte index	5	4	3	2	1	0																		
MAC byte	BA	98	76	54	32	10																		
A7	<p>Bit[31:16]: MAC address[47:32] (Byte 5 ~ Byte 4)</p> <p>Bit[15:14](w): Number of packets buffered in Network Stack minus 1.</p> <p>Bit[13:0](w): Inter-packet Delay (unit: cycles of network TX clock).</p>	<p>1Gbps: XXXX05DC</p> <p>100Mbps: XXXX0064</p>	R/W																					

Addr (hex)	Information	Default (hex)	Access															
A8	<p>Gateway IP address is divided into 4 bytes (see example in the table below)</p> <p>Read: Bit[31:0]: Gateway IP address [31:0] (Byte 3 ~ Byte 0)</p> <p>Write: Bit[9:8](w): Gateway IP address byte index (3~0) Bit[7:0](w): Gateway IP address byte Set to 0 if using automatic gateway from DHCP. A Network Soft Reset is required for any Gateway change (See Addr 0xAF).</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Gateway IP Address</td> <td colspan="4">192.168.1.1</td> </tr> <tr> <td>Byte index</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>Target IP byte</td> <td>C0</td> <td>A8</td> <td>01</td> <td>01</td> </tr> </table>	Gateway IP Address	192.168.1.1				Byte index	3	2	1	0	Target IP byte	C0	A8	01	01	00000000	R/W
Gateway IP Address	192.168.1.1																	
Byte index	3	2	1	0														
Target IP byte	C0	A8	01	01														
A9	Reserved	00000000	RO															
AA	<p>Board IP address is divided into 4 bytes (see example in the table below)</p> <p>Read: Bit[31:0]: Board IP address [31:0] (Byte 3 ~ Byte 0)</p> <p>Write: Bit[9:8](w): Board IP address byte index (3~0) Bit[7:0](w): Board IP address byte</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Target IP Address</td> <td colspan="4">192.168.1.100</td> </tr> <tr> <td>Byte index</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>Target IP byte</td> <td>C0</td> <td>A8</td> <td>01</td> <td>64</td> </tr> </table>	Target IP Address	192.168.1.100				Byte index	3	2	1	0	Target IP byte	C0	A8	01	64	Encoder: C0A80164 Decoder: C0A80165	R/W
Target IP Address	192.168.1.100																	
Byte index	3	2	1	0														
Target IP byte	C0	A8	01	64														

Addr (hex)	Information	Default (hex)	Access
AB	Reserved	00000000	RO
AC	Reserved	00000000	RO
AD	Bit[15:0](w): RX UDP Port	000004D2	RW
AE	Bit[0](w): Check TX packet format for Transport Stream compliance. If set to '1', only Transport Stream packets will be sent out.	00000001	R/W
AF	Bit[3](t): Bypass DHCP (Toggle Write) <ul style="list-style-type: none"> - 0: Obtain an IP from DHCP server - 1: Bypass DHCP and use the default IP Bit[2](t): Static IP Enable (Toggle Write) <ul style="list-style-type: none"> - 0: Only assign IP address on successful DHCP handshake - 1: Use static board IP address set in registers 0xAA~0xAD if DHCP fails Bit[1]: Network Speed <ul style="list-style-type: none"> - 0: 100Mbps - 1: 1Gbps Bit[0](w): Network Soft Reset (Self-clear)	1Gbps: 00000006 100Mbps: 00000004	R/W