Mailbox Client with Avalon® Streaming Interface Intel® FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **21.1**

IP Version: **1.0.1**

Subscribe

Send Feedback

UG-20254 | 2021.03.29

Latest document on the web: **PDF** | **HTML**

Contents

1. Mailbox Client with Avalon® Streaming Interface Intel FPGA IP Overview

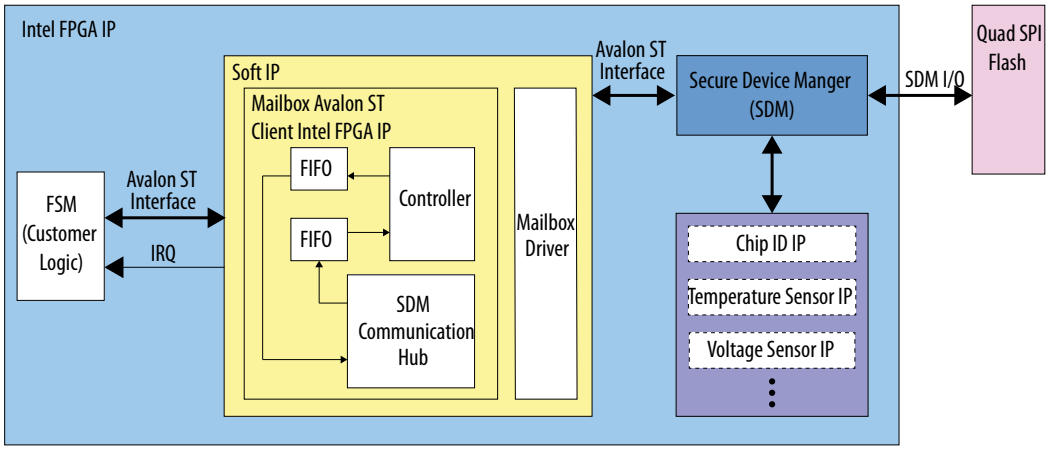
The Mailbox Client with Avalon® streaming interface Intel® FPGA IP (Mailbox Client with Avalon ST Client IP) provides a communication channel between your custom logic and the secure device manager (SDM). You can use the Mailbox Client with Avalon ST IP to send command packets and receive response packets from SDM peripheral modules. The Mailbox Client with Avalon ST IP defines functions that the SDM runs.

Your custom logic can use this communication channel to receive information and access flash memory from the following peripheral modules:

- The Chip ID
- The Temperature Sensor
- The Voltage Sensor
- Quad serial peripheral interface (SPI) flash memory

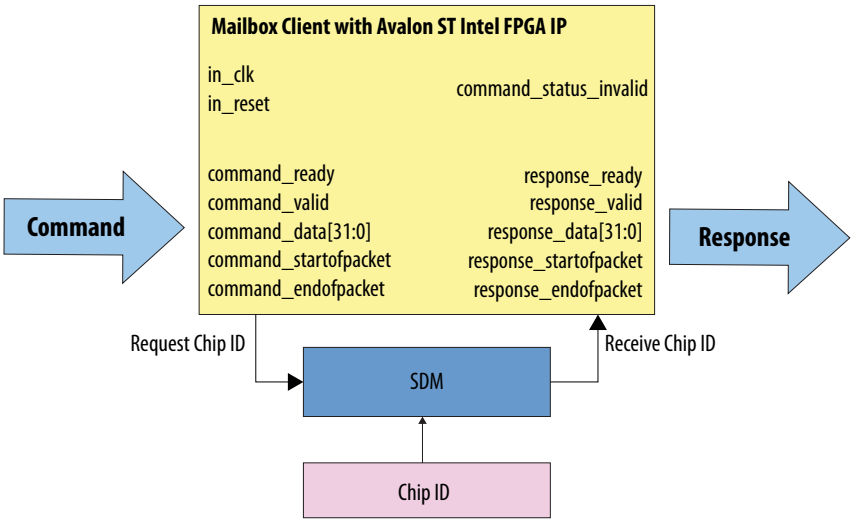
Note: Throughout this user guide, the term Avalon ST abbreviates the Avalon streaming interface or IP.

Figure 1. Mailbox Client with Avalon ST IP System Design



The following figure shows an application in which the Mailbox Client with Avalon ST IP reads the Chip ID.

Figure 2. Mailbox Client with Avalon ST IP Reads Chip ID



1.1. Device Family Support

The following lists the device support level definitions for Intel FPGA IPs:

- **Advance support** — The IP is available for simulation and compilation for this device family. Timing models include initial engineering estimates of delays based on early post-layout information. The timing models are subject to change as silicon testing improves the correlation between the actual silicon and the timing models. You can use this IP for system architecture and resource utilization studies, simulation, pinout, system latency assessments, basic timing assessments (pipeline budgeting), and I/O transfer strategy (data-path width, burst depth, I/O standards tradeoffs).
- **Preliminary support** — The IP is verified with preliminary timing models for this device family. The IP meets all functional requirements, but might still be undergoing timing analysis for the device family. It can be used in production designs with caution.
- **Final support** — The IP is verified with final timing models for this device family. The IP meets all functional and timing requirements for the device family and can be used in production designs.

Table 1. Device Family Support

Device Family	Support
Intel Agilex™	Advance

Note: Intel does not provide simulation modes for the Mailbox Client with Avalon streaming interface Intel FPGA IP.

Related Information

Mailbox Client with Avalon Streaming Interface Intel FPGA IP Release Notes

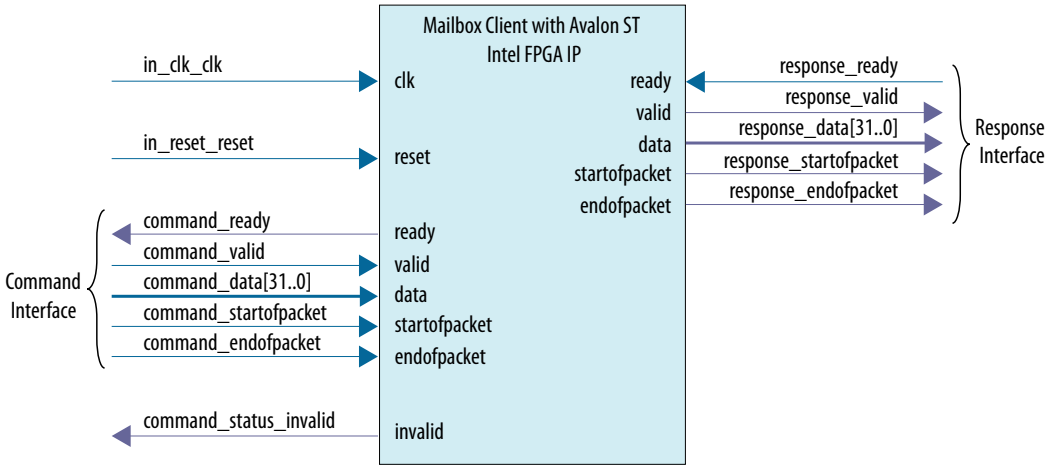
1.1.1. Parameters

Parameter Name	Value	Description
Enable status interface	**On** **Off**	When you enable this interface, the Mailbox Client with Avalon streaming interface Intel FPGA IP includes the `command_status_invalid` signal. When `command_status_invalid` asserts, you must reset the IP.

1.1.2. Interfaces

The following figure illustrates the Mailbox Client with Avalon Streaming Interface Intel FPGA IP interfaces:

Figure 3. Mailbox Client with Avalon Streaming Interface Intel FPGA IP Interfaces



For more information about Avalon streaming interfaces, refer to the *Avalon Interface Specifications*.

Related Information

Avalon Interface Specifications

1.1.2.1. Clock and Reset Interfaces

Table 2. Clock and Reset Interfaces

Signal Name	Direction	Description
`in_clk`	Input	This is the clock for the Avalon streaming interfaces. The maximum frequency in 250 MHz.
`in_reset`	Input	This is an active high reset. Assert `in_reset` to reset the Mailbox Client with Avalon streaming interface Intel FPGA IP (Mailbox Client with Avalon ST IP). When the `in_reset` signal asserts, the SDM must flush any pending activity from the Mailbox Client with Avalon ST IP. The SDM continues to process commands from other clients. To ensure the Mailbox Client with Avalon ST IP functions correctly when the device enters user mode, your design must include the Reset Release Intel FPGA IP to hold the reset until the FPGA fabric entered user mode. Intel recommends using a reset synchronizer when connecting the user reset or output of the Reset Release IP to

continued...

Signal Name	Direction	Description
		the reset port of the Mailbox Client with Avalon ST IP. To implement the reset synchronizer, use the Reset Bridge Intel FPGA IP available in the Platform Designer. *Note:* For IP instantiation and connection guidelines in the Platform Designer, refer to the *Required Communication and Host Components for the Remote System Update Design Example* figure in the *Intel Agilex Configuration User Guide*.

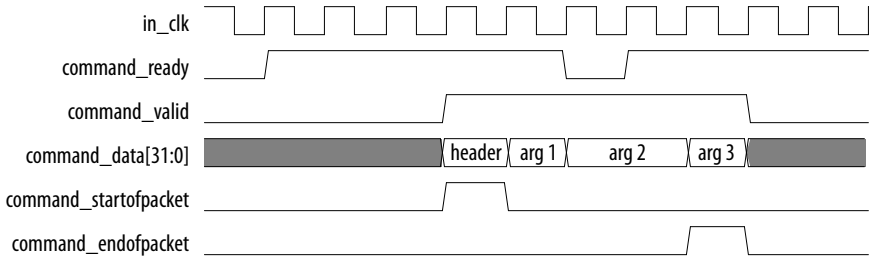
1.1.2.2. Command Interface

Use the Avalon Streaming (Avalon ST) interface to send commands to the SDM.

Table 3. Command Interface

Signal Name	Direction	Description
command_ready	Output	The Mailbox Client with Avalon ST Intel FPGA IP asserts command_ready when it is ready to receive commands from the application. The ready_latency is 0 cycles. The Mailbox Client with Avalon ST can accept command_data[31:0] in the same cycle that command_ready asserts.
command_valid	Input	The command_valid signal asserts to indicate that command_data is valid.
command_data[31:0]	Input	The command_data bus drives commands to the SDM. Refer to *Command List and Description* for definitions of the commands.
command_startofpacket	Input	The command_startofpacket asserts in the first cycle of a command packet.
command_endofpacket	Input	The command_endofpacket asserts in the last cycle of command a packet.

Figure 4. Timing for Avalon ST Command Packet



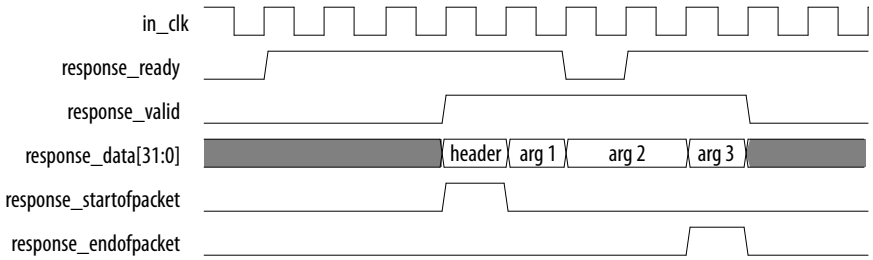
1.1.2.3. Response Interface

The SDM Avalon ST Client IP sends responses to your application using the response interface.

Table 4. Response Interface

Signal 5	Direction	Description
response_ready	Input	Application logic can assert the response_ready signal whenever it is able to receive a response.
response_valid	Output	The SDM asserts response_valid to indicate that response_data is valid.
response_data[31:0]	Output	The SDM drives response_data to provide the requested information. The first word of the response is a header that identifies the command that the SDM is providing. Refer to *Command List and Description* for definitions of the commands.
response_startofpacket	Output	The response_startofpacket asserts in the first cycle of a response packet.
response_endofpacket	Output	The response_endofpacket asserts in the last cycle of a response packet.

Figure 5. Timing for Avalon ST Response Packet

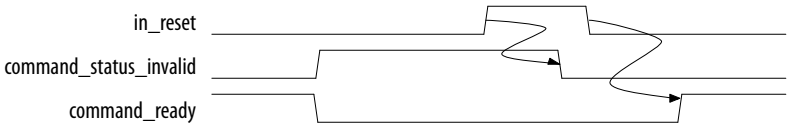


1.1.2.4. Command Status Interface

Table 5. Command Status Interface

Signal Name	Direction	Description
`command_status_invalid`	Output	The `command_status_invalid` asserts to indicate an error. This signal typically asserts to indicate that the length of the command specified in the command header does not match the length of the command sent. When `command_status_invalid` asserts, your application logic must assert `in_reset` to restart the Mailbox Client with Avalon streaming interface Intel FPGA IP.

Figure 6. Reset After command_status_invalid Asserts

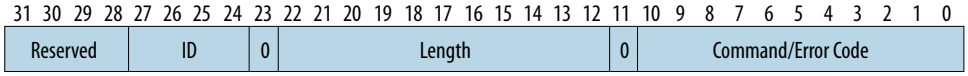


1.2. Commands and Responses

The host controller communicates with the SDM using command and response packets via the Mailbox Client Intel FPGA IP.

The first word of the command and response packets is a header that provides basic information about the command or response.

Figure 7. Command and Response Header Format



Note: The LENGTH field in the command header must match the command length of corresponding command.

The following table describes the fields of the header command.

Table 6. Command and Response Header Description

Header	Bit	Description
Reserved	[31:28]	Reserved.
`ID`	[27:24]	The command ID. The response header returns the ID specified in the command header. Refer to *Operation Commands* for command descriptions.
`0`	[23]	Reserved.
`LENGTH`	[22:12]	Number of words of arguments following the header. The IP responds with an error if a wrong number of words of arguments is entered for a given command.
Reserved	[11]	Reserved. Must be set to 0.
`Command Code/Error Code`	[10:0]	`Command Code` specifies the command. The `Error Code` indicates whether the command succeeded or failed. In the command header, these bits represent command code. In the response header, these bits represent error code. If the command succeeds, the `Error Code` is 0. If the command fails, refer to the error codes defined in the *Error Code Responses*.

1.2.1. Operation Commands

Resetting Quad SPI Flash

Important: For Intel Agilex devices, you must connect the serial flash or quad SPI flash reset pin to the AS_nRST pin. The SDM must fully control the QSPI reset. Do not connect the quad SPI reset pin to any external host.

Table 7. Command List and Description

Command	Code (Hex)	Command Length (1)	Response Length (1)	Description
NOOP	0	0	0	Sends an OK status response.
GET_IDCODE	10	0	1	The response contains one argument which is the JTAG IDCODE for the device
GET_CHIPID	12	0	2	The response contains 64-bit CHIPID value with the least significant word first.
GET_USERCODE	13	0	1	The response contains one argument which is the 32-bit JTAG USERCODE that the configuration bitstream writes to the device.
GET_VOLTAGE	18	1	1	The GET_VOLTAGE command has a single argument which is a bitmask specifying the channels to read. Bit 0 specifies channel 0, bit 1 specifies channel 1, and so on. The response includes a one-word argument for each bit set in the bitmask. The voltage returned is an unsigned fixed-point number with 16 bits below the binary point. For example, a voltage of 0.75V returns 0x0000C000. (2) Intel Agilex devices have a single voltage sensor. Consequently, the response is always one word.
GET_ TEMPERATURE	19	1	n(3)	The GET_TEMPERATURE command returns the temperature or temperatures of the core fabric or transceiver channel locations you specify. For Intel Agilex devices, use the sensor_req argument to specify the locations. The sensor_req includes the following fields: • Bits[31:28]: Reserved. • Bits[27:16]: Sensor Location. Specifies the TSD location. • Bits[15:0]: Sensor mask. Specifies the sensors to read for the sensor location specified. The response contains one word for each temperature requested. If omitted, the command reads channel 0. The least significant bit (lsb) corresponds to sensor 0. The most significant bit (msb) corresponds to channel 15. The temperature returned is a signed fixed value with 8 bits below the binary point. For example, a temperature of 10°C returns 0x00000A00. A of temperature -1.5°C returns 0xFFFFFE80. If the bitmask specifies an invalid Location, the command returns an error code which is any value in the range 0x80000000 -0x800000FF. For Intel Agilex devices, refer to the *Intel Agilex Power Management User Guide* for more information about local build-in temperature sensors.
RSU_IMAGE_ UPDATE	5C	2	0	Triggers reconfiguration from the data source that can be either the factory or an application image. This command takes an optional 64-bit argument that specifies the reconfiguration data address in the flash. When sending the argument to the IP, you first send bits [31:0] followed by bits [63:32]. If you do not provide this argument its value is assumed to be 0.

continued...

(1) This number does not include the command or response header.

(2) Refer to *Intel Agilex Power Management User Guide for more information* about temperature sensor channels and locations.

(3) Index n depends on the number of sensor masks.

<table>
<tr><th>Command</th><th>Code (Hex)</th><th>Command Length (1)</th><th>Response Length (1)</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>• Bit [31:0]: The start address of an application image.
• Bit [63:32]: Reserved (write as 0).
Once the device processes this command, it returns the response header to response FIFO before it proceeds to reconfigure the device. Ensure the host PC or host controller stops servicing other interrupts and focuses on reading the response header data to indicate the command completed successfully. Otherwise, the host PC or host controller may not be able to receive the response once the reconfiguration process started.
Once the device proceeds with reconfiguration, the link between the external host and FPGA is lost. If you use PCIe in your design, you need to re-enumerate the PCIe link.
Important: When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.</td></tr>
<tr><td>RSU_GET_SPT</td><td>5A</td><td>0</td><td>4</td><td>RSU_GET_SPT retrieves the quad SPI flash location for the two sub-partition tables that the RSU uses: SPT0 and SPT1.
The 4-word response contains the following information:
<table>
<tr><th>Word</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>SPT0[63:32]</td><td rowspan="2">SPT0 address in quad SPI flash.</td></tr>
<tr><td>1</td><td>SPT0[31:0]</td></tr>
<tr><td>2</td><td>SPT1[63:32]</td><td rowspan="2">SPT1 address in quad SPI flash.</td></tr>
<tr><td>3</td><td>SPT1[31:0]</td></tr>
</table></td></tr>
<tr><td>CONFIG_STATUS</td><td>4</td><td>0</td><td>6</td><td>Reports the status of the last reconfiguration. You can use this command to check the configuration status during and after configuration. The response contains the following information:
<table>
<tr><th>Word</th><th>Summary</th><th>Description</th></tr>
<tr><td>0</td><td>State</td><td>Describes the most recent configuration related error. Returns 0 when there are no configuration errors.
The error field has 2 fields:
• Upper 16 bits: Major error code.
• Lower 16 bits: Minor error code.
Refer to Appendix: CONFIG_STATUS and RSU_STATUS Error Code Descriptions in the Mailbox Client Intel FPGA IP User Guide for more information.</td></tr>
<tr><td>1</td><td>Version</td><td>The version of the RSU data structure.</td></tr>
<tr><td>2</td><td>Pin status</td><td>• Bit [31]: Current nSTATUS output value (active low)
• Bit [30]: Detected nCONFIG input value (active low)
• Bit [29:3]: Reserved
• Bit [2:0]: The MSEL value at power up</td></tr>
<tr><td>3</td><td>Soft function status</td><td>Contains the value of each of the soft functions, even if you have not assigned the function to an SDM pin.</td></tr>
</table></td></tr>
</table>

continued...

(1) This number does not include the command or response header.

<table>
<tr><th>Command</th><th>Code (Hex)</th><th>Command Length (1)</th><th>Response Length (1)</th><th colspan="3">Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>• Bit [31:6]: Reserved
• Bit [5]: HPS_WARMRESET
• Bit [4]: HPS_COLDRESET
• Bit [3]: SEU_ERROR
• Bit [2]: CVP_DONE
• Bit [1]: INIT_DONE
• Bit [0]: CONF_DONE</td></tr>
<tr><td></td><td></td><td></td><td></td><td>4</td><td>Error location</td><td>Contains the error location. Returns 0 if there are no errors.</td></tr>
<tr><td></td><td></td><td></td><td></td><td>5</td><td>Error details</td><td>Contains the error details. Returns 0 if there are no errors.</td></tr>
<tr><td>RSU_STATUS</td><td>5B</td><td>0</td><td>9</td><td colspan="3">Reports the current remote system upgrade status. You can use this command to check the configuration status during configuration and after it has completed. This command returns the following responses:</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Word</td><td>Summary</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td></td><td>0-1</td><td>Current image</td><td>Flash offset of the currently running application image.</td></tr>
<tr><td></td><td></td><td></td><td></td><td>2-3</td><td>Failing image</td><td>Flash offset of the highest priority failing application image. If multiple images are available in flash memory, stores the value of the first image that failed. A value of all 0s indicates no failing images. If there are no failing images, the remainder of the remaining words of the status information do not store valid information.
Note: A rising edge on nCONFIG to reconfigure from ASx4, does not clear this field. Information about failing image only updates when the Mailbox Client receives a new RSU_IMAGE_UPDATE command and successfully configures from the update image.</td></tr>
<tr><td></td><td></td><td></td><td></td><td>4</td><td>State</td><td>Failure code of the failing image.
The error field has two parts:
• Bit [31:16]: Major error code
• Bit [15:0]: Minor error code
Returns 0 for no failures. Refer to Appendix: CONFIG_STATUS and RSU_STATUS Error Code Descriptions in the Mailbox Client Intel FPGA IP User Guide for more information.</td></tr>
<tr><td></td><td></td><td></td><td></td><td>5</td><td>Version</td><td>The version of the RSU software.</td></tr>
<tr><td></td><td></td><td></td><td></td><td>6</td><td>Error location</td><td>Stores the error location of the failing image. Returns 0 for no errors.</td></tr>
<tr><td></td><td></td><td></td><td></td><td>7</td><td>Error details</td><td>Stores the error details for the failing image. Returns 0 if there are no errors.</td></tr>
</table>

continued...

(1) This number does not include the command or response header.

Command	Code (Hex)	Command Length (1)	Response Length (1)	Description		
				8	Current image retry counter	Count of the number of retries that have been attempted for the current image. The counter is 0 initially. The counter is set to 1 after the first retry, then 2 after a second retry. Specify the maximum number of retries in your Intel Quartus® Prime Settings File (`.qsf`). The command is: `set_global_assignment -name RSU_MAX_RETRY_COUNT 3`. Valid values for the MAX_RETRY counter are 1-3. The actual number of available retries is `MAX_RETRY -1` This field was added in version 19.3 of the Intel Quartus Prime Pro Edition software.
`RSU_NOTIFY`	5D	1	0	Clears all error information in the RSU_STATUS response and resets the retry counter. The one-word argument has the following fields: • 0x00050000: Clear current reset retry counter. Resetting the current retry counter sets the counter back to zero, as if the current image was successfully loaded for the first time. • 0x00060000: Clear error status information. • All other values are reserved. This command is not available before version 19.3 of the Intel Quartus Prime Pro Edition software.		
`QSPI_OPEN`	32	0	0	Requests exclusive access to the quad SPI. You issue this request before any other QSPI requests. The SDM accepts the request if the quad SPI is not in use and the SDM is not configuring the device. Returns OK if the SDM grants access. *Note:* The SDM grants exclusive access to the client using this mailbox. Other clients cannot access the quad SPI until the active client relinquishes access using the `QSPI_CLOSE` command. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.		
`QSPI_CLOSE`	33	0	0	Closes the exclusive access to the quad SPI interface. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.		
`QSPI_SET_CS`	34	1	0	Specifies one of the attached quad SPI devices via the chip select lines. Takes a one-word argument as described below: • Bits[31:28]: Flash device to select. The value `4'b0000` selects the flash that corresponds to `nCSO[0]`. `nCSO[0]` is the only signal that the FPGA can use to access the quad SPI flash device. The HPS can use `nCSO[3:0]` to access HPS data. • Bits[27:0]: Reserved (write as 0). The HPS can use `nCSO[3:1]` to access 3 additional quad SPI devices. This command is optional for the AS x4 configuration scheme. Is required for all other configuration schemes. Access to the QSPI flash memory devices using SDM_IO pins is only available for the AS x4 configuration scheme, JTAG configuration, and a design compiled for ASx4 configuration. For the Avalon streaming interface (Avalon ST) configuration scheme, you must connect QSPI flash memories to GPIO pins. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.		

continued...

(1) This number does not include the command or response header.

Command	Code (Hex)	Command Length (1)	Response Length (1)	Description
QSPI_READ	3A	2	N	Reads the attached quad SPI device. The maximum transfer size is 4 kilobytes (KB) or 1024 words. Takes two arguments: • The quad SPI flash address (one word). The address must be word aligned. The device returns the 0x1 error code for non-aligned addresses. • Number of words to read (one word). When successful, returns OK followed by the read data from the quad SPI device. A failure response returns an error code. For a partially successful read, QSPI_READ may erroneously return the OK status. *Note:* You cannot run the QSPI_READ command while device configuration is in progress. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.
QSPI_WRITE	39	2+N	0	Writes data to the quad SPI device. The maximum transfer size is 4 kilobytes (KB) or 1024 words. Takes three arguments: • The flash address offset (one word). The write address must be word aligned. • The number of words to write (one word). • The data to be written (one or more words). A successful write returns the OK response code. To prepare memory for writes, use the QSPI_ERASE command before issuing this command. *Note:* You cannot run the QSPI_WRITE command while device configuration is in progress. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.
QSPI_ERASE	38	2	0	Erases a sector of the quad SPI device. Takes two arguments: • The flash address offset to start the erase (one word). The address must be the start address of a sector within the flash memory; consequently, the address must be 64 KB aligned. Returns an error for non-64 KB aligned addresses. • The number of words to erase specified in multiples of 0x4000 words. A successful erase returns the OK response code. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.
QSPI_READ_DEVICE_REG	35	2	N	Reads registers from the quad SPI device. The maximum read is 8 bytes. Takes two arguments: • The opcode for the read command. • The number of bytes to read. A successful read returns the OK response code followed by the data read from the device. The read data return is in multiple of 4 bytes. If the bytes to read is not an exact multiple of 4 bytes, it is padded with multiple of 4 bytes until the next word boundary and the padded bit value is zero. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.
QSPI_WRITE_DEVICE_REG	36	2+N	0	Writes to registers of the quad SPI. The maximum write is 8 bytes. Takes three arguments:

continued...

(1) This number does not include the command or response header.

Command	Code (Hex)	Command Length (1)	Response Length (1)	Description
				• The opcode for the write command. • The number of bytes to write. • The data to write. To perform a sector erase or sub-sector erase, you must specify the serial flash address in most significant byte (MSB) to least significant byte (LSB) order as the following example illustrates. To erase a sector of a Micron 2 gigabit (Gb) flash at address 0x04FF0000 using the QSPI_WRITE_DEVICE_REG command, write the flash address in MSB to LSB order as shown here: Header: 0x00003036 Opcode: 0x000000DC Number of bytes to write: 0x00000004 Flash address: 0x0000FF04 A successful write returns the OK response code. This command pads data that is not a multiple of 4 bytes to the next word boundary. The command pads the data with zero. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.
QSPI_SEND_DEVICE_OP	37	1	0	Sends a command opcode to the quad SPI. Takes one argument: • The opcode to send the quad SPI device. A successful command returns the OK response code. *Important:* When resetting quad SPI, you must follow instructions specified in Resetting Quad SPI Flash on page 9.
REBOOT_HPS	47	0	0	Triggers the HPS cold reset. Firmware loads the new bitstream from the boot source based on MSEL settings. The loaded bitstream and your device must use the same firmware version. When loading the new bitstream, SDM wipes HPS and loads the HPS bootloader to HPS OCRAM and releases MPU. Refer to *Intel Agilex Hard Processor System Technical Reference Manual* and *Intel Agilex SoC FPGA Boot User Guide* for more details on HPS reset.

For CONFIG_STATUS and RSU_STATUS major and minor error code descriptions, refer to *Appendix: CONFIG_STATUS and RSU_STATUS Error Code Descriptions* in the *Mailbox Client Intel FPGA IP User Guide*.

Related Information

- Mailbox Client Intel FPGA IP User Guide: CONFIG_STATUS and RSU_STATUS Error Code Descriptions
  For more information about the CONFIG_STATUS and RSU_STATUS error codes.
- Intel Agilex Power Management User Guide
  For more information about the temperature sensor channel numbers and temperature sensing diodes (TSDs).
- Intel Agilex Hard Processor System Technical Reference Manual

(1) This number does not include the command or response header.

1.2.2. Error Code Responses

Table 8. Error Codes

Value (Hex)	Error Code Response	Description
0	OK	Indicates that the command completed successfully. A command may erroneously return the OK status if a command, such as QSPI_READ is partially successful.
1	INVALID_COMMAND	Indicates that the currently loaded boot ROM cannot decode or recognize the command code.
3	UNKNOWN_COMMAND	Indicates that the currently loaded firmware cannot decode the command code.
4	INVALID_COMMAND_PARAMETERS	Indicates that the command is incorrectly formatted. For example, the length field setting in header is not valid.
6	COMMAND_INVALID_ON_SOURCE	Indicates that the command is from a source for which it is not enabled.
8	CLIENT_ID_NO_MATCH	Indicates that the Client ID cannot complete the request to close the exclusive access to quad SPI. The Client ID does not match the existing client with the current exclusive access to quad SPI.
9	INVALID_ADDRESS	The address is invalid. This error indicates one of the following conditions: • An unaligned address • An address range problem • A read permission problem • An invalid chip select value, displaying value of more than 3 • An invalid address in RSU case • An invalid bitmask value for GET_VOLTAGE command • An invalid page selection for GET_TEMPERATURE command
A	AUTHENTICATION_FAIL	Indicates the configuration bitstream signature authentication failure.
B	TIMEOUT	This error indicates time out due to the following conditions: • Command • Waiting for QSPI_READ operation to complete • Waiting for the requested temperature reading from one of the temperature sensors. May indicate a potential hardware error in the temperature sensor.
C	HW_NOT_READY	Indicates one of the following conditions: • The hardware is not ready. Can indicate either an initialization or configuration problem. The hardware may refer to quad SPI. • RSU image is not used to configure the FPGA.
D	HW_ERROR	Indicates that the command completed unsuccessfully due to unrecoverable hardware error.
80 - 8F	COMMAND_SPECIFIC_ERROR	Indicates a command specific error due to an SDM command you used.

SDM Command	Error Name	Error code	Description
GET_CHIPID	EFUSE_SYSTEM_FAILURE	0x82	Indicates that the eFuse cache pointer is invalid.
QSPI_OPEN/ QSPI_CLOSE/ QSPI_SET_CS/ QSPI_READ_DEVICE_REG/	QSPI_HW_ERROR	0x80	Indicates QSPI flash memory error. This error indicates one of the following conditions:

continued...

Value (Hex)	Error Code Response	Description			
		QSPI_WRITE_DEVICE_REG/ QSPI_SEND_DEVICE_OP/ QSPI_READ			• A QSPI flash chip select setting problem • A QSPI flash initialization problem • A QSPI flash resetting problem • A QSPI flash settings update problem
			QSPI_ALREADY_OPEN	0x81	Indicates that the client's exclusive access to QSPI flash via QSPI_OPEN command is already open.
100	NOT_CONFIGURED	Indicates that the device is not configured.			
1FF	ALT_SDM_MBOX_RESP_DEVICE_ BUSY	Indicates that the device is busy due to following use cases: • RSU: Firmware is unable to transition to different version due to an internal error. • HPS: HPS is busy when in HPS reconfiguration process or HPS cold reset.			
2FF	ALT_SDM_MBOX_RESP_NO_VALID_RESP_AVAILABLE	Indicates that there is no valid response available.			
3FF	ALT_SDM_MBOX_RESP_ERROR	General Error.			

1.2.3. Error Code Recovery

The table below describes possible steps to recover from an error code. Error recovery depends on specific use case.

Table 9. Error Code Recovery for known Error Codes

Value	Error Code Response	Error Code Recovery
4	INVALID_COMMAND_PARAMETERS	Resend the command header or header with arguments with corrected parameters. For example, ensures that the length field setting in header is sent with the correct value.
6	COMMAND_INVALID_ON_SOURCE	Resend the command from valid source such as JTAG, HPS, or core fabric.
8	CLIENT_ID_NO_MATCH	Wait for the client who opened the access to quad SPI to complete its access and then closes the exclusive access to quad SPI.
9	INVALID_ADDRESS	Possible error recovery steps: For GET_VOLTAGE command: Send command with a valid bitmask. For GET_TEMPERATURE command: Send command with valid sensor location and sensor mask. For QSPI operation: • Send command with a valid chip select. • Send command with a valid QSPI flash address. For RSU: Send command with a valid start address of the factory image or application.
B	TIMEOUT	Possible recovery steps: For GET_TEMPERATURE command: Retry to send the command again. If problem persists, reconfigure or power cycle the device.

continued...

Value	Error Code Response	Error Code Recovery
		For QSPI operation: Check signal integrity of QSPI interfaces and attempt command again. For HPS restart operation: Retry to send the command again.
C	HW_NOT_READY	Possible recovery steps: For QSPI operation: Reconfigure the device via source. Ensure that IP used to build your design allows access to the QSPI flash. For RSU: Configure the device with RSU image.
80	QSPI_HW_ERROR	Check the QSPI interface signal integrity and to ensure the QSPI device is not damaged.
81	QSPI_ALREADY_OPEN	Client already opened QSPI. Continue with the next operation.
82	EFUSE_SYSTEM_FAILURE	Attempt reconfiguration or power cycle. If error persists after reconfiguration or power cycle, the device may be damaged and unrecoverable.
100	NOT_CONFIGURED	Send a bitstream that configures the HPS.
1FF	ALT_SDM_MBOX_RESP_ DEVICE_ BUSY	Possible error recovery steps: For QSPI operation: Wait for ongoing configuration or other client to complete operation. For RSU: Reconfigure device to recover from internal error. For HPS restart operation: Wait for reconfiguration via HPS or HPS Cold Reset to complete.

1.3. Mailbox Client with Avalon Streaming Interface Intel FPGA IP User Guide Document Archives

IP versions are the same as the Intel Quartus Prime Design Suite software versions up to v19.1. From Intel Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme.

If an IP core version is not listed, the user guide for the previous IP core version applies.

Intel Quartus Prime Version	IP Core Version	User Guide
20.4	1.0.1	Mailbox Client with Avalon Streaming Interface Intel FPGA IP User Guide
20.3	1.0.1	Mailbox Client with Avalon Streaming Interface Intel FPGA IP User Guide
20.2	1.0.0	Mailbox Avalon Streaming Interface Client Intel FPGA IP User Guide
20.1	1.0.0	Mailbox Avalon Streaming Interface Client Intel FPGA IP User Guide
19.3	1.0.0	Mailbox Avalon Streaming Interface Client Intel FPGA IP User Guide

1.4. Document Revision History for the Mailbox Client with Avalon Streaming Interface Intel FPGA IP User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2021.03.29	21.1	1.0.1	Made the following changes: • Revised RSU_IMAGE_UPDATE description in the *Command List and Description* table. • Restructured *Operation Commands*. Removed major and minor error code descriptions for the CONFIG_STATUS and RSU_STATUS commands. The major and minor error codes are now documented as an appendix in the *Mailbox Client Intel FPGA IP User Guide.*
2020.12.14	20.4	1.0.1	Made the following changes: • Added important note about resetting QSPI flash in the *Operation Commands* topic. • Updated the *Command List and Description* table: — Revised GET_TEMPERATURE command description. — Revised RSU_IMAGE_UPDATE command description. • Added text about resetting QSPI flash. • Added text describing behavior between the external host and FPGA. • Removed text: *Returns a non-zero response if the device is already processing a configuration command.* — Updated QSPI_WRITE and QSPI_READ descriptions to specify that the maximum transfer size is 4 kilobytes or 1024 words. — Corrected response length from 1 to 0 for the QSPI_OPEN, QSPI_CLOSE and QSPI_SET_CS command. — Revised QSPI_OPEN, QSPI_WRITE, QSPI_READ_DEVICE_REG, and QSPI_WRITE_DEVICE_REG descriptions. — Added a new command: REBOOT_HPS. • Added new topic: Error Code Recovery.
2020.10.05	20.3	1.0.1	• Changed the title of this user guide from *Mailbox Avalon Streaming Interface Client Intel FPGA IP User Guide* to *Mailbox Client with Avalon Streaming Interface Intel FPGA IP User Guide* due to the IP name change in the Intel Quartus Prime IP Catalog. • Globally updated all IP name instances. • Revised GET TEMPERATURE command description for Intel Agilex devices in the *Command List and Description* table. • Added recommendation about the reset synchronizer in the *Clock and Reset Interfaces* table. • Updated the *Error Codes* table. Added new error code responses: — HW_ERROR — COMMAND_SPECIFIC_ERROR • Removed the *Temperature Sensor Locations* topic. The temperature sensor information is available in the *Intel Agilex Power Management User Guide*.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2020.06.30	20.2	1.0.0	• Changed the title of this user guide from *Mailbox Avalon ST Client Intel FPGA IP User Guide* to *Mailbox Avalon Streaming Interface Client Intel FPGA IP User Guide*. • Renamed topic title *Command and Response Header* to *Commands and Responses*. • Revised `ID`, `LENGTH`, and `Command Code/Error Code` descriptions in the *Command and Response Header Description* table. • Renamed topic title *Supported Commands* to *Operation Commands*. • Revised the following commands description in the *Command List and Description* table: — `GET_TEMPERATURE` — `RSU_STATUS` — `QSPI_SET_CS` • Renamed topic title *Error Codes* to *Error Code Responses*. • Removed `UNKNOWN_BR` command from the *Error Code* table.
2020.04.13	20.1	1.0.0	Made the following changes: • Added information about the temperature sensors for the `GET_TEMPERATURE` command, including figures illustrating TSD locations. • Added `RSU_NOTIFY` command in the *Command Code List and Description* table. • Updated the *Error Codes* table: — Renamed `INVALID_COMMAND_PARAMETERS` to `INVALID_LENGTH`. — Changed `COMMAND_INVALID_ON_SOURCE` hex value from 5 to 6. — Changed `CLIENT_ID_NO_MATCH` hex value from 6 to 8. — Changed `INVALID_ADDRESS` hex value from 7 to 9. — Added `AUTHENTICATION_FAIL` command. — Changed `TIMEOUT` hex value from 8 to B. — Changed `HW_NOT_READY` hex value from 9 to C.
2019.09.30	19.3	1.0.0	Initial release.