



Mailbox Client Intel[®] FPGA IP User Guide

Updated for Intel[®] Quartus[®] Prime Design Suite: **20.2**

IP Version: **20.0.0**



UG-20087 | 2020.06.30

Latest document on the web: [PDF](#) | [HTML](#)



Contents

1. Mailbox Client Intel FPGA IP User Guide.....	3
1.1. Device Family Support.....	4
1.2. Commands and Responses.....	5
1.2.1. Operation Commands.....	5
1.2.2. Error Code Responses.....	12
1.3. Mailbox Client Intel FPGA Core Signals.....	12
1.4. Mailbox Client Intel FPGA IP Avalon MM Memory Map.....	13
1.4.1. Interrupt Enable Register.....	13
1.4.2. Interrupt Status Register.....	14
1.4.3. Timer Registers.....	15
1.5. Specifying the Command and Response FIFO Depths.....	16
1.6. Using the Mailbox Client Intel FPGA IP.....	17
1.7. Mailbox Client Intel FPGA IP Core Use Case Examples.....	21
1.8. Mailbox Client Intel FPGA IP User Guide Archives.....	21
1.9. Document Revision History for the Mailbox Client Intel FPGA IP User Guide.....	22

1. Mailbox Client Intel FPGA IP User Guide

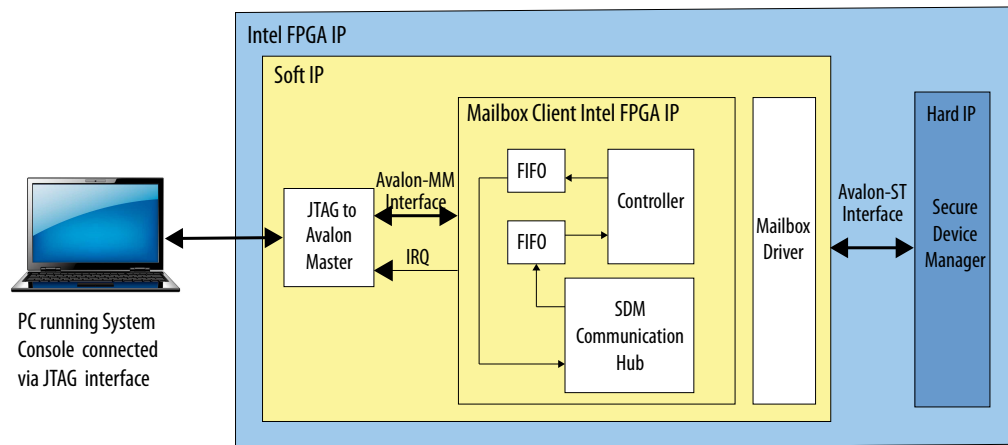
The Mailbox Client Intel FPGA IP is a bridge between a host and the Intel® Stratix® 10 secure device manager (SDM). You use the Mailbox Client Intel FPGA IP to send commands and receive status from SDM peripheral clients. The Mailbox Client defines functions that the SDM runs.

The following pre-defined functions are available:

- Reading the Chip ID
- Reading temperature sensors
- Reading voltage sensors
- Reading and writing external quad serial peripheral interface (SPI) flash memory
- Performing remote system updates (RSU)

The following block diagram shows how to use the Mailbox Client Intel FPGA IP in an interactive session.

Figure 1. Mailbox Client Intel FPGA IP System Block Diagram



This block diagram includes the following components:



- The System Console: provides a Tcl Console pane that you can use to run Mailbox Client Intel FPGA IP functions.
- The JTAG to Avalon® Master Bridge Intel FPGA IP: translates the commands it receives from the System Console to Avalon Memory-Mapped (Avalon MM) format that the Mailbox Client Intel FPGA IP requires.
- Mailbox Client Intel FPGA IP: drives commands and receives responses from the SDM. This component includes FIFOs with a maximum depth of 1024 entries to store commands and responses. The Mailbox Client Intel FPGA IP interrupt indicates when the input FIFO is full and when the output FIFO contains valid data. You can size these FIFOs to accommodate the commands that you intend to send.

Note: You can also use the Nios® II processor or custom logic to send commands to the Mailbox Client Intel FPGA IP. However, Intel does not provide IP to support these solutions.

Related Information

- [Avalon Interface Specifications](#)
- [Secure Device Manager in Intel Stratix 10 Devices](#)
- [Operation Commands](#) on page 5

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 Stratix 10	Final
Intel Agilex™	Advance

Related Information

[Mailbox Client Intel FPGA IP Release Notes](#)

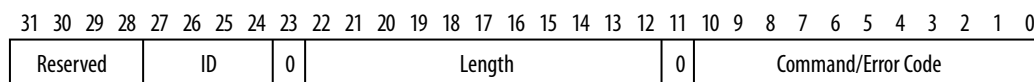


1.2. Commands and Responses

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 2. 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 2. 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 <i>Operation Commands</i> 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 <i>Error Code Responses</i> .

1.2.1. Operation Commands

Table 3. 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.
<i>continued...</i>				

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



Command	Code (Hex)	Command Length ⁽¹⁾	Response Length ⁽¹⁾	Description
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) (3)}
GET_TEMPERATURE	19	1	1	The GET_TEMPERATURE command returns the temperature or temperatures of the core fabric or transceiver channel locations you specify. Use the <code>sensor_req</code> argument to specify the locations. The <code>sensor_req</code> includes the following fields: <ul style="list-style-type: none"> Bits[31:28]: Reserved. Bits[27:16]: <code>Sensor Location</code>. Specifies the TSD location. Bits[15:0]: <code>Sensor mask</code>. Specifies the sensors to read for the <code>sensor location</code> 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 0xFFFFE80. If the bitmask specifies an invalid <code>Location</code> , the command returns an error code which is any value in the range 0x8000000 -0x80000FF. For Intel Stratix 10 devices, the channels return the temperatures for the following locations: <ul style="list-style-type: none"> Channel 0: Samples the temperature from the core fabric. Channels 1- 6: Samples the temperature from the specified transceiver tile. Channels 7-8: Samples the temperature from the high-bandwidth DRAM memory (HBM2) stacks. For Intel Stratix 10 devices, refer to the <i>Temperature Sensor Channels and Locations</i> in the <i>Intel Stratix 10 Analog to Digital Converter User Guide</i> for more information about sensor locations. For Intel Agilex devices, refer to the <i>Intel Agilex E-Tile Transceiver Local Temperature Sensor Design Guidelines</i> in the <i>Intel Agilex Power Management User Guide</i> for more information about local build-in temperature sensors.
RSU_IMAGE_UPDATE	5C	2	0	Triggers reconfiguration from the data source which 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. If you do not provide this argument its value is assumed to be 0. <ul style="list-style-type: none"> Bit [63:32]: Reserved (write as 0). Bit [31:0]: The start address of an application image.

continued...

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

⁽²⁾ Refer to *Intel Stratix 10 Analog to Digital Converter User Guide* for more information about reading voltage sensors on Intel Stratix 10 devices.

⁽³⁾ Refer to *Intel Agilex Power Management User Guide* for more information about temperature sensor channels and locations.



Command	Code (Hex)	Command Length ⁽¹⁾	Response Length ⁽¹⁾	Description																					
				Returns a non-zero response if the device is already processing a configuration command.																					
RSU_GET_SPT	5A	0	4	<p>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:</p> <table border="1"> <thead> <tr> <th>Offset</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table>	Offset	Name	Description	0	SPT0[63:32]	SPT0 address in quad SPI flash.	1	SPT0[31:0]	2	SPT1[63:32]	SPT1 address in quad SPI flash.	3	SPT1[31:0]								
Offset	Name	Description																							
0	SPT0[63:32]	SPT0 address in quad SPI flash.																							
1	SPT0[31:0]																								
2	SPT1[63:32]	SPT1 address in quad SPI flash.																							
3	SPT1[31:0]																								
CONFIG_STATUS	4	0	6	<p>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:</p> <table border="1"> <thead> <tr> <th>Word</th> <th>Summary</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>State</td> <td> <p>Describes the most recent configuration related error. Returns 0 when there are no configuration errors.</p> <p>The error field has 2 fields:</p> <ul style="list-style-type: none"> Upper 16 bits: Major error code. Lower 16 bits: Minor error code. <p>Refer to the Table 4 on page 10 and Table 5 on page 11 for more information.</p> </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> <ul style="list-style-type: none"> 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> <p>Contains the value of each of the soft functions, even if you have not assigned the function to an SDM pin.</p> <ul style="list-style-type: none"> 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>4</td> <td>Error location</td> <td>Contains the error location. Returns 0 if there are no errors.</td> </tr> <tr> <td>5</td> <td>Error details</td> <td>Contains the error details. Returns 0 if there are no errors.</td> </tr> </tbody> </table>	Word	Summary	Description	0	State	<p>Describes the most recent configuration related error. Returns 0 when there are no configuration errors.</p> <p>The error field has 2 fields:</p> <ul style="list-style-type: none"> Upper 16 bits: Major error code. Lower 16 bits: Minor error code. <p>Refer to the Table 4 on page 10 and Table 5 on page 11 for more information.</p>	1	Version	The version of the RSU data structure.	2	Pin status	<ul style="list-style-type: none"> 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 	3	Soft function status	<p>Contains the value of each of the soft functions, even if you have not assigned the function to an SDM pin.</p> <ul style="list-style-type: none"> 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 	4	Error location	Contains the error location. Returns 0 if there are no errors.	5	Error details	Contains the error details. Returns 0 if there are no errors.
Word	Summary	Description																							
0	State	<p>Describes the most recent configuration related error. Returns 0 when there are no configuration errors.</p> <p>The error field has 2 fields:</p> <ul style="list-style-type: none"> Upper 16 bits: Major error code. Lower 16 bits: Minor error code. <p>Refer to the Table 4 on page 10 and Table 5 on page 11 for more information.</p>																							
1	Version	The version of the RSU data structure.																							
2	Pin status	<ul style="list-style-type: none"> 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 																							
3	Soft function status	<p>Contains the value of each of the soft functions, even if you have not assigned the function to an SDM pin.</p> <ul style="list-style-type: none"> 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 																							
4	Error location	Contains the error location. Returns 0 if there are no errors.																							
5	Error details	Contains the error details. Returns 0 if there are no errors.																							

continued...

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



Command	Code (Hex)	Command Length ⁽¹⁾	Response Length ⁽¹⁾	Description		
RSU_STATUS	5B	0	9	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:		
				Word	Summary	Description
				0-1	Current image	Flash offset of the currently running application image.
				2-3	Failing image	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. <i>Note:</i> 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.
				4	State	Failure code of the failing image. The error field has two parts: <ul style="list-style-type: none"> Upper 16 bits: Major error code. Lower 16 bits: Minor error code. Returns 0 for no failures. Refer to the Table 4 on page 10 and Table 5 on page 11 for more information.
				5	Version	The version of the RSU software.
				6	Error location	Stores the error location of the failing image. Returns 0 for no errors.
				7	Error details	Stores the error details for the failing image. Returns 0 if there are no errors.
				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: <code>set_global_assignment -name RSU_MAX_RETRY_COUNT 3</code> . 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.
				<i>continued...</i>		

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



Command	Code (Hex)	Command Length ⁽¹⁾	Response Length ⁽¹⁾	Description
RSU_NOTIFY	5D	1	0	<p>Clears all error information in the RSU_STATUS response and resets the retry counter. The one-word argument has the following fields:</p> <ul style="list-style-type: none"> 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. <p>This command is not available before version 19.3 of the Intel Quartus Prime Pro Edition Software.</p>
QSPI_OPEN	32	0	1	<p>Requests exclusive access to the quad SPI. 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. Returns the ALT_SDM_MBOX_RESP_DEVICE_BUSY when the quad SPI flash is busy.</p> <p><i>Note:</i> 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.</p>
QSPI_CLOSE	33	0	1	<p>Closes the exclusive access to the quad SPI interface.</p>
QSPI_SET_CS	34	1	1	<p>Specifies one of the attached quad SPI devices via the chip select lines. Takes a one-word argument as described below:</p> <ul style="list-style-type: none"> Bits[31:28]: Flash device to select. The value 4'b0000 selects the flash that corresponds to nCS0[0]. nCS0[0] is the only signal that the FPGA can use to access the quad SPI flash device. The HPS can use nCS0[3:1] to access HPS data. Bits[27:0]: Reserved (write as 0). The HPS can use nCS0[3:1] to access 3 additional quad SPI devices. <p>This command is optional for the AS x4 configuration scheme. Is required for all other configuration schemes.</p> <p>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 ST configuration scheme, you must connect QSPI flash memories to GPIO pins.</p>
QSPI_READ	3A	2	N	<p>Reads the attached quad SPI device. The maximum read size is 4 kilobytes (KB). Takes two arguments:</p> <ul style="list-style-type: none"> 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). <p>When successful returns OK followed by the read data from the quad SPI device. A failure response returns an error code.</p> <p>For a partially successful read, QSPI_READ may erroneously return the OK status.</p> <p><i>Note:</i> You cannot run the QSPI_READ command while device configuration is in progress.</p>
QSPI_WRITE	39	2+N	0	<p>Writes data to the quad SPI device. Takes three arguments:</p> <ul style="list-style-type: none"> The flash address offset (one word). The write address must be word aligned. The device returns error code 0x3FF for non-aligned addresses. The number of words to write (one word). The data to be written (one or more words). <p>A successful write returns the OK response code.</p>

continued...

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



Command	Code (Hex)	Command Length ⁽¹⁾	Response Length ⁽¹⁾	Description
				To prepare memory for writes, Intel recommends using the QSPI_ERASE command before issuing this command. <i>Note:</i> You cannot run the QSPI_WRITE command while device configuration is in progress.
QSPI_ERASE	38	2	0	Erases a sector of the quad SPI device. Takes two arguments: <ul style="list-style-type: none"> 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.
QSPI_READ_DEVICE_REG	35	2	N	Reads registers from the quad SPI device. The maximum read is 8 bytes. Takes two arguments. <ul style="list-style-type: none"> 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. Pads data that is not a multiple of 4 bytes to the next word boundary.
QSPI_WRITE_DEVICE_REG	36	2+N	0	Writes to registers of the quad SPI. The maximum write is 8 bytes. Takes three arguments: <ul style="list-style-type: none"> 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.
QSPI_SEND_DEVICE_OP	37	1	0	Sends a command opcode to the quad SPI. Takes one argument: <ul style="list-style-type: none"> The opcode to send the quad SPI device. A successful command returns the OK response code.

Table 4. CONFIG_STATUS and RSU_STATUS Major Error Code Descriptions

Major Error Code	Error Type	Description
0xF001	BITSTREAM_ERROR	Potential unsigned bitstream used. Ensure the bitstream is signed with the correct key.
0xF002	HARDWARE_ACCESS_FAILURE	Failure to communicate to PMBus-compliant voltage regulator. Check your power management and smart voltage identification (SmartVID) parameter settings and PMBus interface connections.
0xF003	BITSTREAM_CORRUPTION	Bitstream is corrupt. Ensure the bit stream in configuration device or flash is not corrupt.
<i>continued...</i>		

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



Major Error Code	Error Type	Description
0xF004	INTERNAL_ERROR	This major code may indicate the following error events: <ul style="list-style-type: none"> An error in the SDM Crypto IP task. An RSU operation error. Refer to Table 5 on page 11 for more information.
0xF005	DEVICE_ERROR	Indicates an SDM internal device error. The following errors are possible: <ul style="list-style-type: none"> A device cleaning failure An HPS configuration failure Contact your local Field Applications Engineer (FAE). Alternatively, submit a Service Request on the My Intel support page to get the support on capturing the error log for further debug.
0xF006	HPS_WATCHDOG_TIMEOUT	HPS watchdog timeout failure. Ensure that your design resets the watchdog timer correctly.
0xF007	INTERNAL_UNKNOWN_ERROR	Indicates an internal device error due to an unknown task. You can contact your local Field Applications Engineer (FAE). Alternatively, submit a Service Request on the My Intel support page to get the support on capturing the error log for further debug.

Table 5. CONFIG_STATUS and RSU_STATUS Minor Error Code Descriptions

Minor Error Code	Error Type	Description
0xD001	RSU_CMF_AUTH_ERR	Authentication failure for the firmware.
0xD002	RSU_USER_AUTH_ERR	Authentication failure for the design.
0xD003	RSU_CMF_DESC_SHA_MISMATCH	The SHA does not match for the firmware descriptor.
0xD004	RSU_POINTERS_NOT_FOUND_ERR	Unable to read data from boot ROM on first boot after the device exits power-on reset (POR).
0xD005	RSU_QSPI_REQ_CHANGE	Unable to configure the quad SPI flash during RSU initialization.
0xD006	RSU_FACTORY_IMAGE_FAILED	Failed to load any image, including the factory image.
0xD007	RSU_CMF_TYPE_ERR	The firmware version does not match the version that was previously loaded.

Related Information

- [Intel Stratix 10 Analog to Digital Converter User Guide](#)
For more information about the temperature sensor channel numbers and temperature sensing diodes (TSDs).
- [Intel Agilix Power Management User Guide](#)
For more information about the temperature sensor channel numbers and temperature sensing diodes (TSDs).



1.2.2. Error Code Responses

Table 6. Error Codes

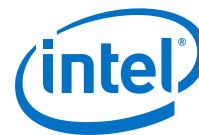
Value (Hex)	Error Code Response	Description
0	OK	Indicates that the command completed successfully. A command may erroneously return the OK status if a command, such as QSPI_READ is partially successful.
1	INVALID_COMMAND	Indicates that the command is incorrectly formatted.
3	UNKNOWN	Indicates that the currently loaded firmware cannot decode the command code.
4	INVALID_LENGTH	Indicates that 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 requesting quad SPI access does not have exclusive access.
9	INVALID_ADDRESS	The address is invalid. This error indicates one of the following conditions: <ul style="list-style-type: none"> An unaligned address An address range problem A read permission problem
A	AUTHENTICATION_FAIL	Indicates the configuration bitstream signature authentication failure.
B	TIMEOUT	The command timed out.
C	HW_NOT_READY	The hardware is not ready. Can indicate either an initialization or configuration problem.
100	NOT_CONFIGURED	Indicates that the device is not configured.
1FF	ALT_SDM_MBOX_RESP_DEVICE_BUSY	Indicates that the device is busy.
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.

1.3. Mailbox Client Intel FPGA Core Signals

The host communicates with the Mailbox Client Intel FPGA over its Avalon Memory-Mapped (Avalon MM) interface. The Avalon MM interface is standard memory-mapped interface. For detailed definitions of these signals, refer to the *Avalon Memory-Mapped Interfaces* chapter in the *Avalon Interface Specifications*.

Table 7. Mailbox Client Intel FPGA Signal Descriptions

Signal Role	Width	Direction	Description
Avalon-MM Interface Signals			
avmm_address	4	Input	Avalon MM address.
avmm_write	1	Input	Avalon MM write request.
avmm_read	1	Input	Avalon MM read request.
<i>continued...</i>			



Signal Role	Width	Direction	Description
avmm_writedata	32	Input	Avalon MM writedata bus.
avmm_readdata	32	Output	Avalon MM readdata bus.
avmm_readdatavalid	1	Output	Avalon MM readdata valid.
Clock and Reset			
clk	1	Input	Input clock to clock the IP. The maximum frequency is 250 MHz.
reset	1	Input	Reset that resets the IP. To reset the IP, assert the <code>reset</code> signal high for at least 2 <code>clk</code> cycles. <i>Note:</i> For IP instantiation guidelines, refer to the <i>Configuration User Guide</i> .
irq	1	Output	Interrupt signal. Drives the value of the AND of the interrupt status and interrupt enable registers.

Related Information

- [Avalon Memory-Mapped Interface Signal Roles](#)
- [Intel Stratix 10 Configuration User Guide](#)
For information about including the Reset Release IP in your design.

1.4. Mailbox Client Intel FPGA IP Avalon MM Memory Map

Table 8. Avalon MM Memory Map

Offset (word)	R/W	31	30:2	1	0
Base address + 0	W	Command			
Base address + 1	W	Command last word (eop)			
Base address + 2	R	Command FIFO empty space			
Base address + 3	N/A	Reserved			
Base address + 4	N/A	Reserved			
Base address + 5	R	Response data			
Base address + 6	R	Response FIFO fill level		EOP	SOP
Base address + 7	R/W	Interrupt enable register (IER)			
Base address + 8	R	Interrupt status register (ISR)			
Base address + 9	R/W	Timer 1 enable	Timer 1 period		
Base address + 10	R/W	Timer 2 enable	Timer 2 period		

1.4.1. Interrupt Enable Register

Use the `Interrupt Enable` register to enable or disable interrupts.

Table 9. Interrupt Enable Register

Bit	Fields	Access	Default Value	Description
31:6	Reserved			
5	EN_BACKPRESSURE_TIMEOUT	R/W	0x0	The enable interrupt bit for SDM backpressure timeout. <ul style="list-style-type: none"> 1: Enable the SDM backpressure timeout interrupt 0: Disable the SDM backpressure timeout interrupt
4	EN_EOP_TIMEOUT	R/W	0x0	The enable interrupt bit for EN_EOP_TIMEOUT. <ul style="list-style-type: none"> 1: Enable the EOP timeout interrupt 0: disable the EOP timeout interrupt
3	EN_COMMAND_INVALID	R/W	0x0	The enable interrupt bit for COMMAND_INVALID. <ul style="list-style-type: none"> 1: Enable the command invalid interrupt 0: Disable the command invalid interrupt
2	Reserved	—	—	You can use this bit to implement an enable for a custom interrupt status bit.
1	EN_CMD_FIFO_NOT_FULL	R/W	0x0	The enable for the command FIFO full interrupt. <ul style="list-style-type: none"> 1: Enable the FIFO full interrupt 0: Disable the FIFO full interrupt
0	EN_DATA_VALID	R/W	0x0	The enable for the data valid interrupt. <ul style="list-style-type: none"> 1: Enable the data valid interrupt 0: Disable the data valid interrupt

1.4.2. Interrupt Status Register

Use the `interrupt_status` register to monitor the status of the FIFO and identify invalid commands.

Your logic can poll the error bits of the `interrupt_status` register. Or, you can configure the `EN_COMMAND_INVALID` bit of the interrupt enable register to interrupt when an error occurs.

When an error occurs, the Mailbox Client IP clears all pending responses. Your logic should not expect any response from Mailbox Client IP after an error occurs. Your logic must assert reset for a minimum of 10 clock cycles to reset the Mailbox Client IP.

Table 10. Interrupt Status Register

Bit	Fields	Access	Default Value	Description
31:6	Reserved			
5	BACKPRESSURE_TIMEOUT	R	0x0	SDM backpressure timer interrupt. <ul style="list-style-type: none"> 1: The SDM backpressure timer has timeout. Indicates that a fatal error occurred in SDM. You must reset the device. 0: The SDM backpressure timer has not timeout.
4	EOP_TIMEOUT	R	0x0	End of Packet (EOP) timer interrupt.

continued...



Bit	Fields	Access	Default Value	Description
				<ul style="list-style-type: none"> 1: Indicates that the EOP timer has timeout. You must reset the Mailbox Client IP. 0: The EOP timer has not timeout. Indicates that the Mailbox Client IP did not receive the full command with EOP due to: <ul style="list-style-type: none"> Mailbox did not receive the last argument with EOP. Mailbox already received all arguments without the EOP in it.
3	COMMAND_INVALID	R	0x0	Invalid command interrupt. Indicates a mismatch between the command length specified in the command header and the number of words sent. Hardware clears this bit. <ul style="list-style-type: none"> 1: Indicates that the command is invalid. You must reset the Mailbox client. 0: The command is valid.
2	Reserved	—	—	You can use this bit to implement a custom interrupt.
1	CMD_FIFO_NOT_FULL	R	0x0	Command FIFO is not full interrupt. <ul style="list-style-type: none"> 1: Indicates command FIFO is not full. The client can drive data. 0: Indicates the FIFO is full. The FIFO automatically clears this bit. You do not need to clear this bit manually.
0	DATA_VALID	R	0x0	Data valid interrupt. <ul style="list-style-type: none"> 1: Indicates that valid data is available. The master can read. 0: Indicates the FIFO is empty. The FIFO automatically clears this bit. You do not need to clear this bit manually.

1.4.3. Timer Registers

Use `timer` registers to monitor and address incomplete transactions between host and the Mailbox Client IP.

Incomplete Command Transaction Error

When a host fails to send the last command word to the Mailbox Client IP or the system stops sending data before the last word, the incomplete command transaction error occurs. Timer 1 allows you to set a specific transaction time period to complete each command. When the timer's timeout occurs, `ISR[4]` is set to indicate the error. To recover the system, you need to reset the Mailbox Client IP.

Table 11. Timer 1 Register

Bit	Fields	Access	Default Value	Description
31	Timer 1 enable	R/W	0x0	The enable bit for Timer 1 period.

continued...



Bit	Fields	Access	Default Value	Description
				<ul style="list-style-type: none">• 1: Enable timer 1• 0: Disable timer 1
30:0	Timer 1 period	R/W	0x7FF_FFFF	When enabled, it counts down the specified period as the maximum number of clock cycles the system has not received a valid command. The timer starts the count down as soon as the SOP tag is set. The timer resets when the Mailbox Client IP receives complete command transaction, indicated by EOP set to 1.

SDM Backpressure Error

When SDM is back pressuring for some time period not allowing you to write any data into the Mailbox fabric and SDM, the SDM backpressure error occurs. Timer 2 allows you to set the wait time to monitor this scenario. When a timer's timeout occurs, `ISR[5]` is set to indicate an error. Note that this is a fatal error received from SDM, possibly indicating a system error. Resetting the Mailbox Client won't recover the system.

Table 12. Timer 2 Register

Bit	Fields	Access	Default Value	Description
31	Timer 2 enable	R/W	0x0	The enable bit for Timer 2 period. <ul style="list-style-type: none">• 1: Enable timer 2• 0: Disable timer 2
30:0	Timer 2 period	R/W	0x7FF_FFFF	When enabled, it counts down the specified period as the maximum number of clock cycles the system has not received a valid command.

1.5. Specifying the Command and Response FIFO Depths

The optimal depth of the command and response FIFOs depends on the specific application. You should size these FIFOs to accommodate the maximum command and responses that your application requires.

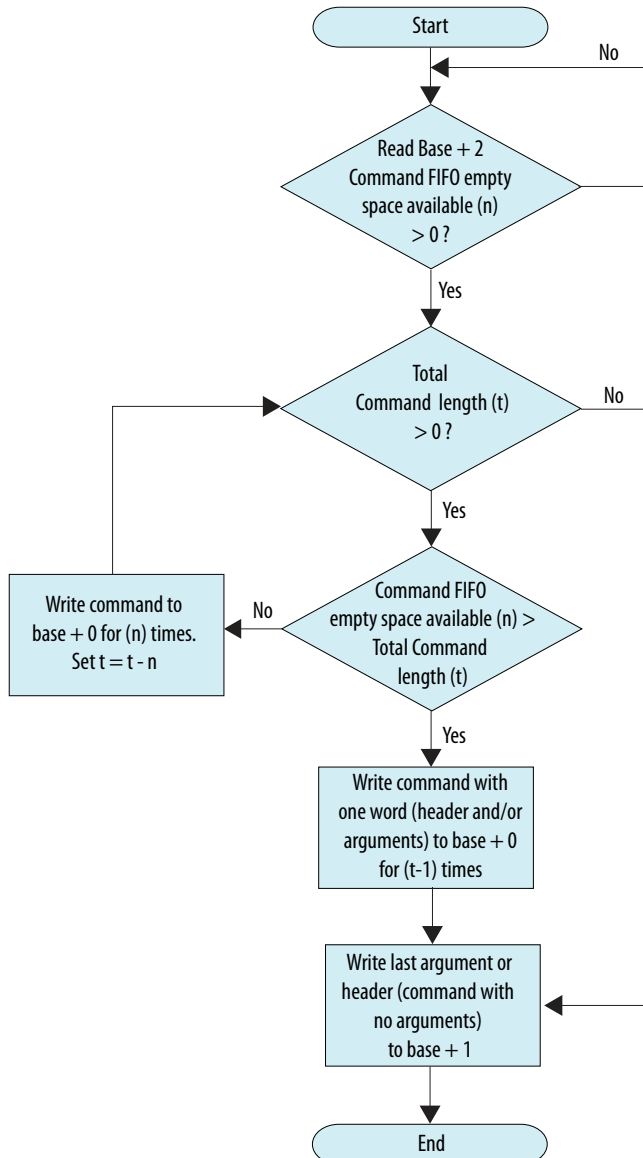
The following example illustrates this point. Consider an application sends a maximum of eight back-to-back `GET_TEMPERATURE` commands to the FPGA core and one transceiver bank. Each command consists of a header and one command argument, specifying the mask of the temperature sensors to read. The optimal setting for the command FIFO is 16 words.

For the response FIFO, each response has one header word, and one word each for the core temperature and transceiver bank temperature. Each response is three words. The optimal setting for the response FIFO Depth is 24 words.

1.6. Using the Mailbox Client Intel FPGA IP

Writing Command Packet

Figure 3. Flow Chart for Writing Command Packet



Agenda:
n: Command FIFO empty space
t: Total Command Length

Write Command Description

When you send a command to the SDM, write the command word into command register, which is the base address. To stay in sync with the hardware, while the command length (t) is greater than zero, write the header and arguments in the Command register which is (base address + 0). Continue writing the header and/or



arguments, one word at the time, in the `Command` register (base address + 0) while there is available free space in the FIFO for commands ($n > t$). Write the last word to the `Command last word` register which is (base address +1). For commands with no arguments, write the header to the `Command last word` register, (base address +1).

Reading from (base address + 2) shows the remaining available free space in the FIFO for commands. The command FIFO can become full when the SDM is busy. The IP requires 3 clock cycles to update the `Command FIFO empty space` value. You can begin reading the `Command FIFO empty space` value 3 clock cycles after writing the command to the IP.

You must check the `Command FIFO empty space` register, (base address + 2) before proceeding to write into the `Command/Command last word` registers. The behavior of the IP is undefined if you write to (base address + 0) and (base address + 1) while the FIFO is full. The write data is discarded.

Unexpected or undefined behavior may occur if you send more commands than required. For example, send the following commands to read the Chip ID value:

- Write the command header to (base address + 0).
- Write again the command header to (base address + 1).

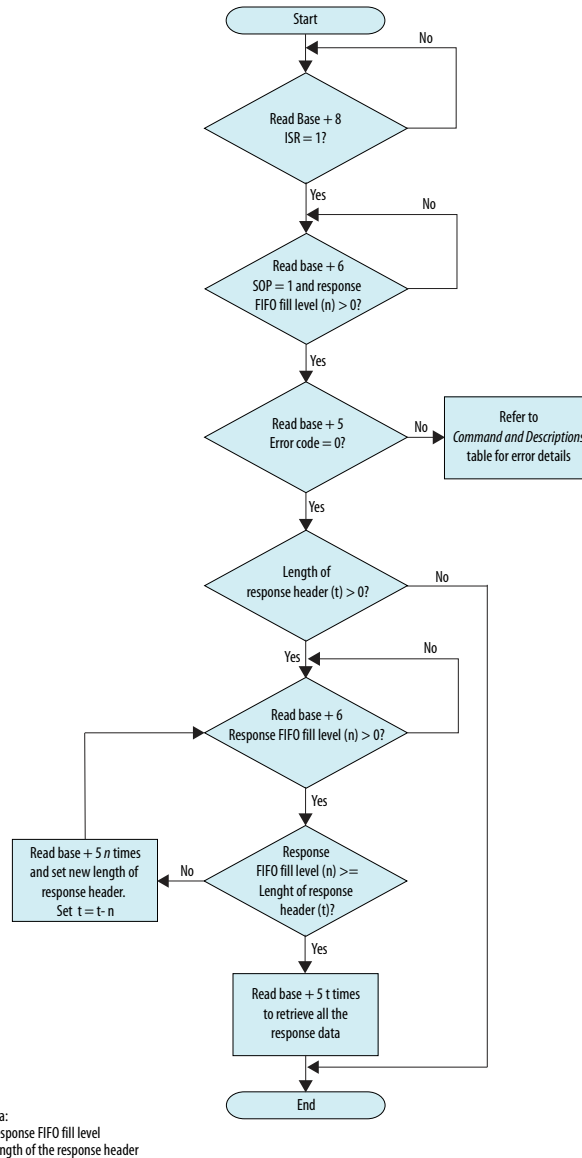
In the above scenario, the IP core expects a 3-word response (command header and 2 data words). However, the SDM only returns a one-word response, which is the error response code.

You must send commands in the correct order to the `Command` or `Command last word` register, as described in the [Writing Command Packet](#) on page 17. Failure to send commands in the correct order can result in loss of services for all mailbox clients, including the following standalone IP cores:

- Temperature Sensor Intel FPGA IP
- Voltage Sensor Intel FPGA IP
- Chip ID Intel FPGA IP
- Advanced SEU Detection Intel IP
- Partial Reconfiguration Controller Intel IP
- Partial Reconfiguration External Configuration Controller Intel FPGA IP

Reading Response Packet

Figure 4. Flow Chart for Reading Response Packet



Read Command Description

1. Read (base address + 8) to check if bit 0 of Interrupt status register is 1, to indicate the valid data is available for the master to read. You can poll the Interrupt status register continuously until bit 0 is 1.
2. Read (base address + 6) to check the SOP (start of packet), EOP (end of packet), and the Response FIFO fill level (n).

To read multiple words complete the following steps:

- a. If `SOP = 1` and `EOP = 0`, the response has multiple words.
- b. If the `Response FIFO fill level (n)` is non-zero, the FIFO has valid data.
- c. For example, if you perform a `QSPI_READ` operation to read 10 words from quad SPI flash, a return value of `0x0000002d` indicates that the SDM wrote 11 words to the response FIFO. The 11 words comprise a response header word and 10 data words.

To read a single word complete the following steps:

- a. If `SOP = 1` and `EOP = 1`, the response has a single word.
 - b. If the `Response FIFO fill level` is non-zero, the FIFO has valid data.
 - c. A return value of `0x00000007` indicates that the SDM wrote a single word to the response FIFO. This single is both the start and end of the single-cycle packet.
3. Read the response header at `(base address + 5)`. The `LENGTH` value specifies the number of words in the response. Proceed to step 4 if the response error code is zero. The response error code is non-zero for unsuccessful commands. Refer to [Table 6](#) on page 12 for more information.
 4. When the length of the response header (`t`) is greater than zero (`LENGTH > 1`), read `(base address + 5)` to retrieve the response data. While continuously reading the response data, you must also continuously poll `(base address + 6)` to check the `Response FIFO fill level (n)`. For the final word of the packet, the `Response FIFO fill level (n)` is 0 and the `EOP` is 1.

Note: If the response FIFO is empty, the return data is undefined. You must check the `Interrupt status` register to ensure that valid data is available. You must verify that the `Response FIFO fill level (n)` is non-zero before reading the response data.

Ensure that you read or flush out the content in the response FIFO before issuing a new command to the mailbox. Continuously sending commands without reading back the valid data from the response FIFO gradually fills the response FIFO. When the response FIFO overflows the SDM freezes.

If the SDM freezes you must reconfigure the device. The Intel Quartus Prime software supports device reconfiguration starting in version 19.1. For earlier versions of the Intel Quartus Prime software, power cycle the device to recover.

Restrictions

1. You can only issue one request and read back the response before issuing a new request to the Mailbox Client IP. Wait 10 ms between back-to-back commands to the SDM mailbox.
2. Do not instantiate more than six mailbox clients in your design. For designs requiring more than six mailbox clients, use the Mailbox Client IP to replace the following standalone IP cores:
 - Voltage Sensor Intel FPGA IP
 - Chip ID Intel FPGA IP
 - Serial Flash Mailbox Client Intel FPGA IP
 - Temperature Sensor Intel FPGA IP



1.7. Mailbox Client Intel FPGA IP Core Use Case Examples

The Mailbox Client Intel FPGA IP is an Avalon MM slave component that must connect to an Avalon MM master. The simplest Avalon MM master is the JTAG-to-Avalon Master.

The `rsu1.tcl` script provides examples to perform all the available command functions. You can run the functions available in the `rsu1.tcl` script via System Console of the Intel Quartus Prime software.

The following example shows how to access the quad SPI flash memory. Follow this sequence to prevent errors. Refer to [Table 3](#) on page 5 for more information about these commands.

1. QSPI_OPEN
2. QSPI_SET_CS
3. Any of the following quad SPI operations:
 - QSPI_READ
 - QSPI_WRITE
 - QSPI_ERASE
 - QSPI_READ_DEVICE_REG
 - QSPI_WRITE_DEVICE_REG
 - QSPI_SEND_DEVICE_OP
4. QSPI_CLOSE

Related Information

[Example of Tcl Script](#)

A Tcl script that implements all the Mailbox Client operations.

1.8. Mailbox Client Intel FPGA IP User Guide 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
20.1	20.0.0	Mailbox Client Intel FPGA IP User Guide
19.3	19.1	Mailbox Client Intel FPGA IP User Guide
18.1	18.1	Mailbox Client Intel FPGA IP User Guide
17.1	17.1	Mailbox Client Intel FPGA IP User Guide



1.9. Document Revision History for the Mailbox Client Intel FPGA IP User Guide

Document Version	Intel Quartus Prime Version	Changes
2020.06.30	20.2	<p>Made the following changes:</p> <ul style="list-style-type: none"> Revised LENGTH and Command Code/Error Code descriptions in the <i>Command and Response Header Description</i> table. Revised GET_TEMPERATURE command description in the <i>Command List and Description</i> table. Removed UNKNOWN_BR command from the <i>Error Codes</i> table. Added new timer feature to handle the error detection for the incomplete transaction timeout error and the SDM backpressure timeout fatal error. Added support for an EOP_TIMEOUT interrupt which indicates that the full command did not include the EOP. Added support for an BACKPRESSURE_TIMEOUT interrupt which indicates that an error within the SDM occurred. Removed SD/MMC text from the CLIENT_ID_NO_MATCH description in the <i>Error Codes</i> table. Updated write and read command descriptions in the <i>Using the Mailbox Client Intel FPGA IP</i> section.
2020.04.13	20.1	<p>Made the following changes:</p> <ul style="list-style-type: none"> Added the following restriction to the definition of QSPI_SET_CS: <i>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 ST configuration scheme, you must connect QSPI flash memories to GPIO pins.</i> Added the following text to the definition of the Failing image field of the RSU_STATUS command: <ul style="list-style-type: none"> <i>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.</i> Added RSU_NOTIFY command in the <i>Command List and Description</i> table. Revised the <i>Flow Chart for Writing Command Packet</i> and <i>Flow Chart for Reading Response Packet</i> to include the correct sequence for writing commands into a command FIFO and reading response packets from a response FIFO. Updated corresponding <i>Write Command Description</i> and <i>Read Command Description</i> sections.
2020.03.17	19.3	<p>Made the following changes:</p> <ul style="list-style-type: none"> Updated the <i>Error Codes</i> table: <ul style="list-style-type: none"> 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	<p>Made the following changes:</p>

continued...



Document Version	Intel Quartus Prime Version	Changes
		<ul style="list-style-type: none"> • Added device support for the Intel Agilex device. • Added support for an <code>COMMAND_INVALID</code> interrupt which indicates the command length specified in the header does not match the actual command sent. • Changed name of the IP from Mailbox Client Intel Stratix 10 FPGA IP to Mailbox Client Intel FPGA IP. • Revised introduction including the <i>Figure 1: Mailbox Client Intel FPGA IP System Block Diagram</i>. • Revised the <i>Flow Chart for Writing Command Packet</i> and <i>Flow Chart for Reading Response Packet</i> to include logic to handle multiple word commands and responses. • Changed references to names of all mailbox client IPs. The mailbox clients IP no longer include the Intel Stratix 10 FPGA in their names. • Added reference to <i>AN 891: Using the Reset Release Intel FPGA IP</i>. • Added reference to the <i>Intel Agilex Power Management User Guide</i>. • Updated the description of the <code>GET_TEMPERATURE</code> command to say the mask argument is optional. When omitted, the command returns the temperature for sensor 0. • Updated the <code>RSU_STATUS</code> command to say the highest priority failing image, not the last failing image. The error information is for the first failing image which is the highest priority failing image. • Added descriptions for <code>CONFIG_STATUS</code> and <code>RSU_STATUS</code> major and minor error codes. • Added <code>HPS_COLDRESET</code> and <code>HPS_WARMRESET</code> to the list of soft functions for the <code>CONFIG_STATUS</code> command. • Added <i>Mailbox Client Intel FPGA IP User Guide Archives</i> topic. • Added the following Intel FPGA IPs to the list of IPs that require proper use of the Command and Command <code>last</code> registers: <ul style="list-style-type: none"> – Advanced SEU Detection Intel IP – Partial Reconfiguration Controller Intel IP – Partial Reconfiguration External Configuration Controller Intel FPGA IP – Edited entire user guide for clarity and style.

Document Version	Changes
2019.04.19	<ul style="list-style-type: none"> • Updated the <i>Feature Description</i> topic. • Added a note to Figure: <i>Command and Response Header Format</i>. • Updated Table: <i>Mailbox Client Intel Stratix 10 FPGA IP Command and Response Header Description</i> to update the description for bit[11] of the command and response header. • Updated Table: <i>Command List and Description</i> to update the descriptions for <code>CONFIG_STATUS</code> and <code>RSU_STATUS</code>. • Renamed topic title <i>Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-MM Interface</i> to <i>Mailbox Client Intel Stratix 10 FPGA IP Core Signals</i>. • Renamed table title <i>Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-MM Interface</i> to <i>Mailbox Client Intel Stratix 10 FPGA IP Core Signal Description</i>. • Updated Table: <i>Mailbox Client Intel Stratix 10 FPGA IP Core Signal Description</i> to include information on clock and reset signals. • Updated Table: <i>Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-Memory Map</i> to remove urgent command and urgent FIFO empty space.

continued...



Document Version	Changes
	<ul style="list-style-type: none"> • Updated the <i>Using the Mailbox Client Intel Stratix 10 FPGA IP Core</i> topic: <ul style="list-style-type: none"> – Added new Figures: <i>Flow Chart for Writing Command Packet</i> and <i>Flow Chart for Reading Response Packet</i>. – Added a new section—<i>Restrictions</i>. – Updated the description in the <i>Writing Command Packet</i> section. • Updated the <i>Mailbox Client Intel Stratix 10 FPGA IP Core Use Case Examples</i> topic. • Made editorial updates through out the document.
2019.03.14	<ul style="list-style-type: none"> • Updated the <i>Mailbox Client Intel Stratix 10 FPGA IP Core User Guide</i> topic. • Updated Figure: <i>Mailbox Client Intel Stratix 10 FPGA IP Core and System Block Diagram</i>. • Updated Table: <i>Command List and Description</i>: <ul style="list-style-type: none"> – Updated the column name <i>Number of Commands</i> to <i>Command Length</i>. – Updated the column name <i>Number of Responses</i> to <i>Respond Length</i>. – Corrected the description for <code>QSPI_READ</code>, <code>QSPI_WRITE</code>, and <code>QSPI_ERASE</code>.
2019.02.25	<ul style="list-style-type: none"> • Updated the description in the <i>Mailbox Client Intel Stratix 10 FPGA IP Core User Guide</i> topic. • Updated Figure: <i>Mailbox Client Intel Stratix 10 FPGA IP Core User Guide</i>. • Updated Table: <i>Interrupt Status Register</i> to update the description for <code>DATA_VALID</code>. • Renamed the following topic titles: <ul style="list-style-type: none"> – <i>Commands and Error Codes</i> to <i>Commands and Responses</i> – <i>Commands</i> to <i>Operation Commands</i>. • Updated Table: <i>Mailbox Client Intel Stratix 10 FPGA IP Command and Response Header Description</i> to update the descriptions for <code>Length</code> and <code>Command Code/Error Code</code>. • Updated Table: <i>Command List and Description</i>: <ul style="list-style-type: none"> – Updated the number of responses and description for <code>CONFIG_STATUS</code>. – Updated the number of responses for <code>RSU_STATUS</code>. – Updated the descriptions for <code>QSPI_READ</code>, <code>QSPI_WRITE</code>, and <code>QSPI_ERASE</code>. • Updated Table: <i>Mailbox Client Intel Stratix 10 FPGA IP Error Code Responses and Description</i> to update the description for <code>UNKNOWN_BR</code>. • Updated the <i>Writing Command Packet</i> and <i>Reading Command Packet</i> sections in the <i>Using the Mailbox Client Intel Stratix 10 FPGA IP Core</i> topic. • Updated the <i>Mailbox Client Intel Stratix 10 FPGA IP Core Use Case Examples</i> topic. • Removed the following topics: <ul style="list-style-type: none"> – <i>Example 1: Reading Intel Stratix 10 IDCODE and Voltage</i> – <i>Example 2: Read and Write EPCQ-L or QSPI Devices</i>
2018.10.15	<ul style="list-style-type: none"> • Updated Table: <i>Command List and Description</i> to include the following commands: <ul style="list-style-type: none"> – Updated the descriptions for <code>GET_TEMPERATURE</code>. – Added new commands: <ul style="list-style-type: none"> • <code>RSU_IMAGE_UPDATE</code> • <code>CONFIG_STATUS</code> • <code>RSU_STATUS</code> – Removed the command <code>GET_DESIGNHASH</code>. • Updated Table: <i>Error Code Responses and Description</i> to update the value of the following error code responses: <ul style="list-style-type: none"> – <code>NOT_CONFIGURED</code> – <code>ALT_SDM_MBOX_RESP_DEVICE_BUSY</code> – <code>ALT_SDM_MBOX_RESP_NO_VALID_RESP_AVAILABLE</code> – <code>ALT_SDM_MBOX_RESP_ERROR</code> • Added a note to Figure: <i>Mailbox Client Intel Stratix 10 FPGA IP Core Block Diagram</i>. • Made minor editorial updates.
2018.02.14	Initial release.