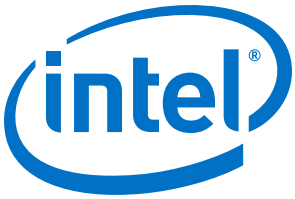


Mailbox Client Intel FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.3**

IP Version: **19.1**

Subscribe
Send Feedback

UG-20087 | 2020.03.17
Latest document on the web: **PDF** | **HTML**

Contents

1. Mailbox Client Intel FPGA IP User Guide

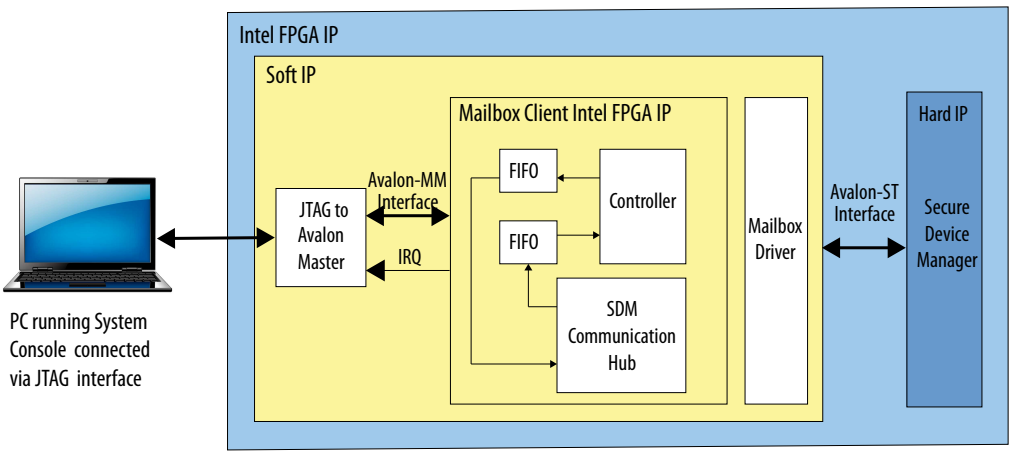
The Mailbox Client Intel FPGA IP is a bridge between a host and the Intel® Stratix® 10 secure device manager (SDM). You use the Mailbox Client Intel FPGA IP to send commands and receive status from SDM peripheral clients. The Mailbox Client defines functions that the SDM runs.

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 in full and when the output FIFO contains valid data. You can size these FIFOs to accommodate the commands the you intend to send.

Note: You can also use the Nios® II processor or custom logic to send commands 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

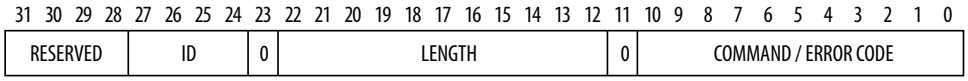
Related Information

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.

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 Format

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 for arguments following the header. The IP responds with an error if the LENGTH received does not match this value.
Reserved	[11]	Reserved. Must be set to 0.
COMMAND CODE or ERROR CODE	[10:0]	For the command header specifies the COMMAND CODE. For the response header specifies the ERROR CODE. If the command succeeds, the ERROR CODE is 0. If the command fails, refer to the error codes defined in the Table 5 on page 11, *Error Code Responses*.

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.
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.

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>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>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)</td></tr>
<tr><td>GET_TEMPERATURE</td><td>19</td><td>1</td><td>1</td><td>The GET_TEMPERATURE command has a single argument which is a bitmask indicating which temperature sensors to read. This argument is optional. If omitted, the command only reads sensor 0.
The response contains one word for each channel temperature requested. (4) (5) 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.
For Intel Stratix 10 devices, the channels return the temperatures for the following locations:
• 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-bandwith DRAM memory (HBM2) stacks.
For Intel Agilex devices, the channels return the temperatures for the following locations:
• Channel 0: Samples the temperature from the core fabric.
• Channel 1: Samples the temperature from the left tranasceiver tile.
• Channel 4: Samples the temperature from the right transceiver tile.
Information about the local temperature sensor channels is preliminary for Intel Agilex devices.</td></tr>
<tr><td>RSU_IMAGE_UPDATE</td><td>5C</td><td>2</td><td>0</td><td>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.
• Bit [63:32]: Reserved (write as 0).
• Bit [31:0]: The start address of an application image.
Returns a non-zero response if the device is already processing a configuration command.</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>Offset</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">SPT0 address in quad SPI flash.</td></tr>
<tr><td>3</td><td>SPT1[31:0]</td></tr>
</table></td></tr>
</table>

continued...

(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.

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

Command	Code (Hex)	Command Length (1)	Response Length (1)	Description
CONFIG_STATUS	4	0	6	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:

Word	Summary	Description
0	State	Describes the most recent configuration related error. Returns 0 when there are no configuration errors. The error field hCONFIG_STATUSas 2 fields: • Upper 16 bits: Major error code. • Lower 16 bits: Minor error code. Refer to Table 4 on page 10, *Error Codes for the CONFIG_STATUS and RSU_STATUS* for more information.
1	Version	The version of the RSU software.
2	Pin status	• 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	Contains the value of each of the soft functions, even if you have not assigned the function to an SDM pin. • 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.

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

continued...

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

(4) For Intel Stratix 10 devices, refer to the *Temperature Sensor Channels and Locations* in the *Intel Stratix 10 Analog to Digital Converter User Guide* for more information about sensor locations.

(5) For Intel Agilex devices, refer to the *Temperature Sensor Channels and Locations* in the Intel Agilex Power Management User Guide for more information on sensor locations.

Command	Code (Hex)	Command Length [1]	Response Length [1]	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 1s 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.
				4	State	Failure code of the failing image. The error field has two parts: • Upper 16 bits: Major error code. • Lower 16 bits: Minor error code. Returns 0 for no failures. Refer to Table 4 on page 10, *Error Codes for the CONFIG_STATUS and RSU_STATUS* 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: `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 `RSU_MAX_RETRY_COUNT -1`
`QSPI_OPEN`	32	0	1	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. *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.		
`QSPI_CLOSE`	33	0	1	Closes the exclusive access to the quad SPI interface.		
`QSPI_SET_CS`	34	1	1	Specifies one of the attached quad SPI devices via the chip select lines. Takes a one-word argument as described below:		

continued...

[1] This number does not include the command or response header.

Command	Code (Hex)	Command Length (1)	Response Length (1)	Description
				• 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[3:1] 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.
QSPI_READ	3A	2	N	Reads the attached quad SPI device. The maximum read size is 4 kilobytes (KB). 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.
QSPI_WRITE	39	2+N	0	Writes data to the quad SPI device. Takes three arguments: • 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). A successful write returns the OK response code. To prepare memory for writes, Intel recommends using the QSPI_ERASE command before issuing this command. *Note:* 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: • 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. • 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: • The opcode for the write command. • The number of bytes to write. • The data to write.

continued...

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

Command	Code (Hex)	Command Length (1)	Response Length (1)	Description
				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: • The opcode to send the quad SPI device. A successful command returns the OK response code.

Table 4. Error Codes for the CONFIG_STATUS and RSU_STATUS Commands

Major Error Code	Description	Possible failure
0xF001	BITSTREAM_ERROR	The configuration bitstream may not be signed. Ensure the bitstream is signed with the correct key.
0xF002	HARDWARE_ACCESS_FAILURE	Communication with the Power Management BUS (PMBus*) voltage regulator failed. Check your power management and smart voltage identification (SmartVID) parameter settings and PMBus interface connections.
0xF003	BITSTREAM_CORRUPTION	The configuration bitstream is corrupt. Check the configuration bitstream stored in the configuration device or flash for corruption.
0xF004	INTERNAL_ERROR	This major code indicates one of the following error events: • An SDM Crypto IP task error • An RSU failure

Minor	Name	Description
0xD001	RSU_CMF_AUTH_ERR	Authentication failed for the firmware signature.
0xD002	RSU_USER_AUTH_ERR	Authentication failed for the firmware in the owner design.
0xD003	RSU_CMF_DESC_SHA_MISMATCH	The SHA does not match for the firmware signature.
0xD004	RSU_POINTERS_NOT_FOUND_ERR	Unable to read data from BootROM on first boot after exiting power-on reset (POR).

continued...

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

Major Error Code	Description	Possible failure		
		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 previously loaded.
0xF005	DEVICE_ERROR	Indicates an internal device error in an SDM task such as a device cleaning failure or configuration via HPS failure. Contact your local Field Applications Engineer (FAE) or submit a Service Request at the My Intel support page for help capturing the error log for further debugging.		
0xF006	HPS_WATCHDOG_TIMEOUT	An HPS watchdog timeout failure. Ensure your design reset for the watchdog timer in functioning correctly.		
0xF007	INTERNAL_UNKNOWN_ERROR	Indicates an internal device error due to unknown task. Contact your local Field Applications Engineer (FAE) or submit a Service Request at the My Intel support page for help capturing the error log for further debugging.		

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 Agilex 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 5. 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.
2	UNKNOWN_BR	Indicates that the command code is not understood.
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 or SD MMC access does not have exclusive access.
9	INVALID_ADDRESS	The address is invalid. This error indicates one of the following conditions:

continued...

Value (Hex)	Error Code Response	Description
		• 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 6. 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.
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 reset signal high for at least 2 clk cycles. *Note:* For IP instantiation guidelines, refer to the *AN 891: Using the Reset Release Intel FPGA IP*.
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
- AN 891: Using the Reset Release Intel FPGA IP

1.4. Mailbox Client Intel FPGA IP Avalon MM Memory Map

Table 7. Avalon MM Memory Map

Offset (word)	R/W	31	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)		

1.4.1. Interrupt Enable Register

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

Table 8. Interrupt Enable Register

Bit	Fields	Access	Default Value	Description
31:2	Reserved			
3	EN_COMMAND_INVALID	R/W	0x0	The enable interrupt bit for COMMAND_INVALID. • 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. • 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. • 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. You logic must assert reset for a minimum of 10 clock cycles, assuming an f_{MAX} frequency of 1 GHz.

Table 9. Interrupt Status Register

Bit	Fields	Access	Default Value	Description
31:2	Reserved			
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. • 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. • 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. • 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.

1.5. Specifying the Command and Response FIFO Depths

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

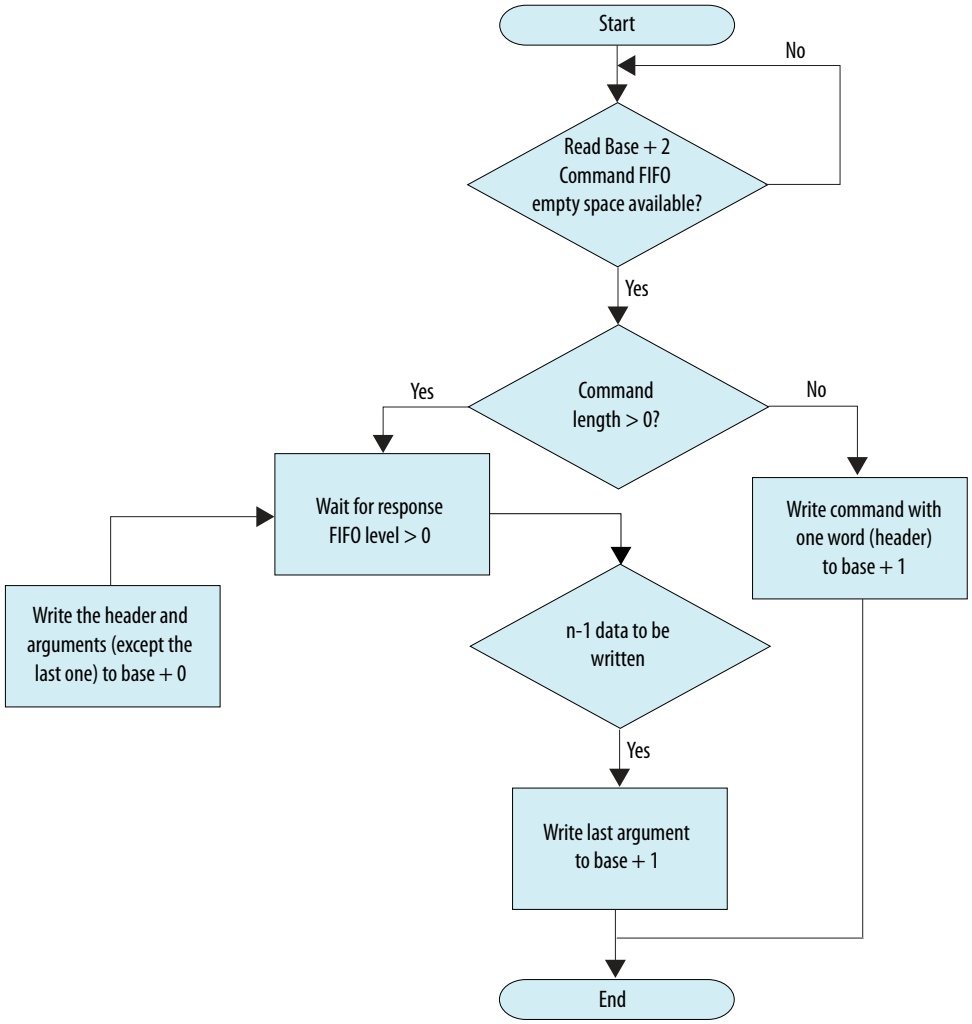
The following example illustrates this point. Consider an application sends a maximum of eight back-to-back `GET_TEMPERATURE` commands to the FPGA core and one transceiver bank. Each command consists of a header and one command argument, specifying the mask of the temperature sensors to read. The optimal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



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, 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 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.

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).

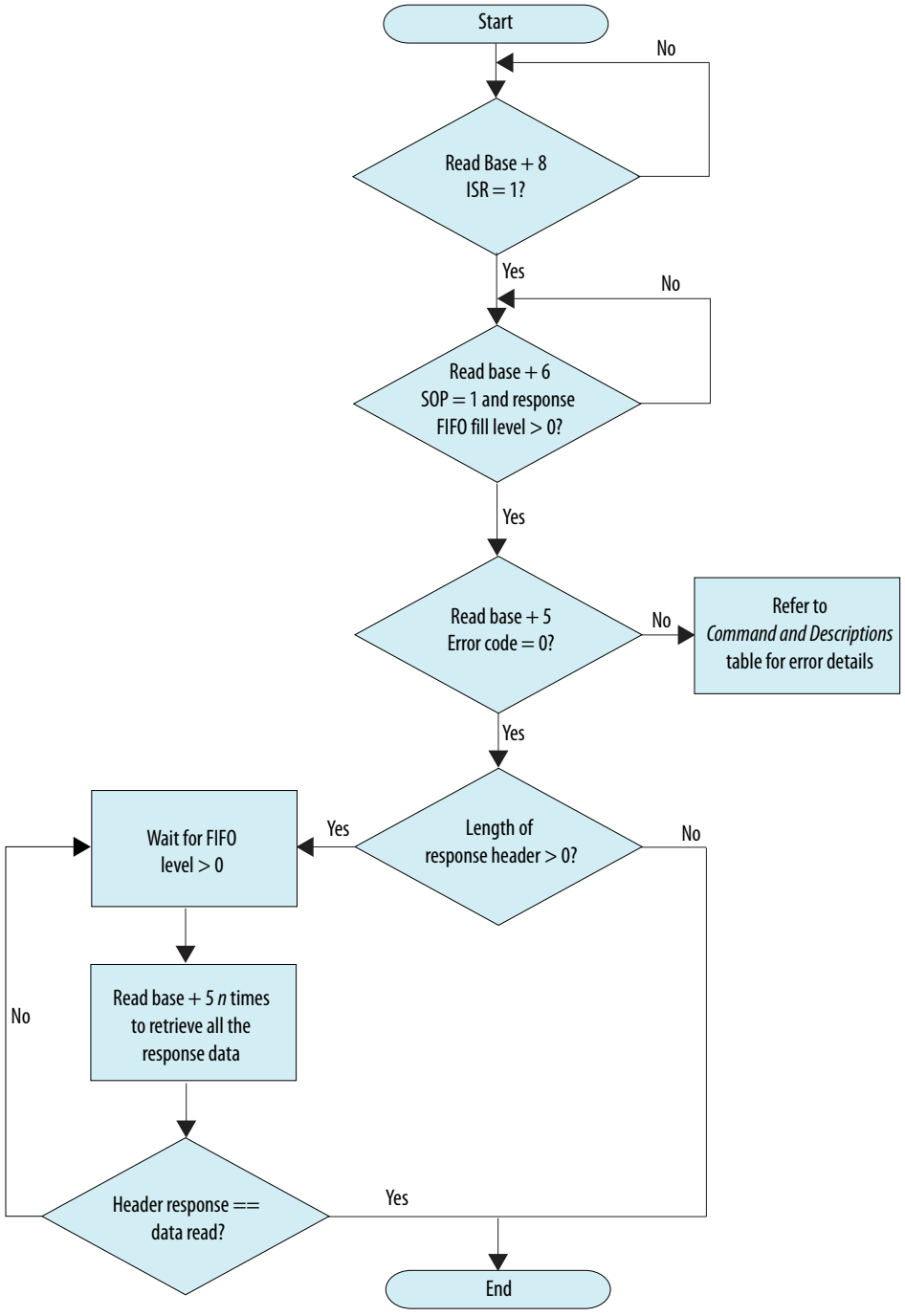
In the above scenario, the IP core expects a 3-word response (command header and 2 data words). However, the SDM only returns a one-word response, which is the error response code.

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 15. 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, meaning the FIFO is not full. You can poll the `Interrupt status` register continuously until bit 0 is 1 if the FIFO is full.
2. Read (base address + 6) to check the SOP (start of packet), EOP (end of packet), and the `Response FIFO fill level`.

   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` 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 5 on page 11 for more information.
4. When (*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`. For the final word of the packet, the `Response FIFO fill level` 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` 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.

IP Core Version	User Guide
18.1	Mailbox Client Intel FPGA IP User Guide
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
2020.03.17	19.3	Made the following changes: • 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	Made the following changes: • Added device support for the Intel Agilex device. • Added support for an COMMAND_INVALID 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 *Figure 1: Mailbox Client Intel FPGA IP System Block Diagram*. • Revised the *Flow Chart for Writing Command Packet* and *Flow Chart for Reading Response Packet* 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 *AN 891: Using the Reset Release Intel FPGA IP*. • Added reference to the *Intel Agilex Power Management User Guide*. • Updated the description of the GET_TEMPERATURE command to say the mask argument is optional. When omitted, the command returns the temperature for sensor 0. • Updated the RSU_STATUS 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 CONFIG_STATUS and RSU_STATUS major and minor error codes. • Added HPS_COLDRESET and HPS_WARMRESET to the list of soft functions for the CONFIG_STATUS command. • Added *Mailbox Client Intel FPGA IP User Guide Archives* topic. • Added the following Intel FPGA IPs to the list of IPs that require proper use of the Command and Command last registers: — 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	• Updated the *Feature Description* topic. • Added a note to Figure: *Command and Response Header Format*. • Updated Table: *Mailbox Client Intel Stratix 10 FPGA IP Command and Response Header Description* to update the description for bit[11] of the command and response header.

continued...

Document Version	Changes
	• Updated Table: *Command List and Description* to update the descriptions for CONFIG_STATUS and RSU_STATUS. • Renamed topic title *Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-MM Interface* to *Mailbox Client Intel Stratix 10 FPGA IP Core Signals*. • Renamed table title *Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-MM Interface* to *Mailbox Client Intel Stratix 10 FPGA IP Core Signal Description*. • Updated Table: *Mailbox Client Intel Stratix 10 FPGA IP Core Signal Description* to include information on clock and reset signals. • Updated Table: *Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-Memory Map* to remove urgent command and urgent FIFO empty space. • Updated the *Using the Mailbox Client Intel Stratix 10 FPGA IP Core* topic: — Added new Figures: *Flow Chart for Writing Command Packet* and *Flow Chart for Reading Response Packet*. — Added a new section—*Restrictions*. — Updated the description in the *Writing Command Packet* section. • Updated the *Mailbox Client Intel Stratix 10 FPGA IP Core Use Case Examples* topic. • Made editorial updates through out the document.
2019.03.14	• Updated the *Mailbox Client Intel Stratix 10 FPGA IP Core User Guide* topic. • Updated Figure: *Mailbox Client Intel Stratix 10 FPGA IP Core and System Block Diagram*. • Updated Table: *Command List and Description*: — Updated the column name *Number of Commands* to *Command Length*. — Updated the column name *Number of Responses* to *Respond Length*. — Corrected the description for QSPI_READ, QSPI_WRITE, and QSPI_ERASE.
2019.02.25	• Updated the description in the *Mailbox Client Intel Stratix 10 FPGA IP Core User Guide* topic. • Updated Figure: *Mailbox Client Intel Stratix 10 FPGA IP Core User Guide*. • Updated Table: *Interrupt Status Register* to update the description for DATA_VALID. • Renamed the following topic titles: — *Commands and Error Codes* to *Commands and Responses* — *Commands* to *Operation Commands*. • Updated Table: *Mailbox Client Intel Stratix 10 FPGA IP Command and Response Header Description* to update the descriptions for Length and Command Code/Error Code. • Updated Table: *Command List and Description*: — Updated the number of responses and description for CONFIG_STATUS. — Updated the number of responses for RSU_STATUS. — Updated the descriptions for QSPI_READ, QSPI_WRITE, and QSPI_ERASE. • Updated Table: *Mailbox Client Intel Stratix 10 FPGA IP Error Code Responses and Description* to update the description for UNKNOWN_BR.

continued...

Document Version	Changes
	• Updated the *Writing Command Packet* and *Reading Command Packet* sections in the *Using the Mailbox Client Intel Stratix 10 FPGA IP Core* topic. • Updated the *Mailbox Client Intel Stratix 10 FPGA IP Core Use Case Examples* topic. • Removed the following topics: — *Example 1: Reading Intel Stratix 10 IDCODE and Voltage* — *Example 2: Read and Write EPCQ-L or QSPI Devices*
2018.10.15	• Updated Table: *Command List and Description* to include the following commands: — Updated the descriptions for `GET_TEMPERATURE`. — Added new commands: • `RSU_IMAGE_UPDATE` • `CONFIG_STATUS` • `RSU_STATUS` — Removed the command `GET_DESIGNHASH`. • Updated Table: *Error Code Responses and Description* to update the value of the following error code responses: — `NOT_CONFIGURED` — `ALT_SDM_MBOX_RESP_DEVICE_BUSY` — `ALT_SDM_MBOX_RESP_NO_VALID_RESP_AVAILABLE` — `ALT_SDM_MBOX_RESP_ERROR` • Added a note to Figure: *Mailbox Client Intel Stratix 10 FPGA IP Core Block Diagram*. • Made minor editorial updates.
2018.02.14	Initial release.