Mailbox Client Intel® Stratix® 10 FPGA IP Core User Guide

Subscribe

Send Feedback

UG-20087 | 2018.02.14

Latest document on the web: PDF | HTML

Contents

1. Mailbox Client Intel® Stratix® 10 FPGA IP Core User Guide

The Mailbox Client Intel® Stratix® 10 FPGA IP core converts the Avalon-ST interface to Avalon-MM interface for clients such as JTAG, FPGA mailbox, HPS mailbox, to communicate with the secure device manager (SDM) in Intel Stratix 10 devices.

Related Information

- Avalon Interface Specifications
- Secure Device Manager in Intel Stratix 10 Devices

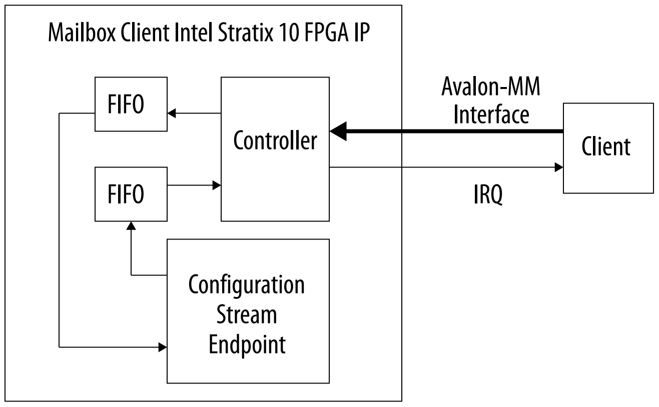
1.1. Feature Description

The Mailbox Client Intel Stratix 10 FPGA IP core features commands and responses for communication with the SDM.

The Mailbox Client Intel Stratix 10 FPGA IP core supports:

- Command and response access
- Urgent access[1]
- Depth-adjustable FIFO to buffer command, response and urgent packet
- Configurable interrupt source:
  - Indication that command FIFO is not full and client can send more commands
  - Indication that there is valid response in FIFO and client can begin to read out the response.

Figure 1. Mailbox Client Intel Stratix 10 FPGA IP Core Block Diagram


Mailbox Client Intel Stratix 10 FPGA IP
FIFO
FIFO
Controller
Configuration Stream Endpoint
Avalon-MM Interface
IRQ
Client

[1] This feature will be available in the future releases of Intel Quartus® Prime software

1.2. Command & Error Code

The command and response packets start with one word of header, followed by zero or more words of argument. The following figure describes the header word format.

Figure 2. Command and Error Code Header Format

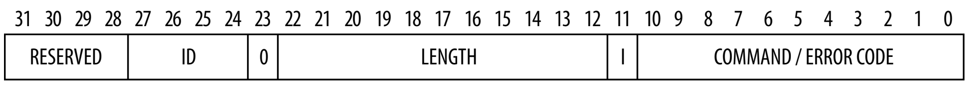


Table 1. Command and Error Code Header Description

Header	Bit	Description
Reserved	[31:28]	Reserved.
ID	[27:24]	A response header returns the same ID value specified in the command header. If you set different ID values in each command, you can use the ID value to match responses with commands if they are out of order.
0	[23]	Reserved
Length	[22:12]	Number of words of arguments following the header.
I	[11]	Always set this bit to `0` when sending commands using the Mailbox Client Intel Stratix 10 FPGA IP core.
Command Code/Error Code	[10:0]	Command identifier

1.2.1. Commands

Table 2. Command List and Description

Command	Code (Hex)	Number of Command Word(2)	Number of Response Word(2)	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 provided to the device by the configuration bitstream.
GET_DESIGNHASH	14	1	3	Command has one argument which is the PR region number. The response contains 3 words (least significant word first), 96-bits of the DESIGN HASH of the region which has been provided to the device by the configuration bitstream. • If the PR region number is 0 then this command returns the design hash for the static region (if a core region is loaded). • If the PR region number is non-zero and a persona is loaded into that region, then this command returns the design hash for the PR persona loaded into that region.

continued...

(2) The number does not include the command and response header.

Send Feedback

Command	Code (Hex)	Number of Command Word[2]	Number of Response Word[2]	Description
`GET_VOLTAGE`	18	1	1	Command has a single argument which is a bitmask of which channels to read. Bit 0 is set to read channel 0, bit 1 to read channel 1 and so on. The response contains one word argument for each bit set in the bitmask. Each return value is a 32-bit value. The voltage returned is an unsigned fixed point number with 16 bits below the binary point. Example: A voltage of 0.75V returns 0x0000C000.
`GET_TEMPERATURE`	19	1	1	Command has a single argument which is a bitmask indicating which temperature sensors to read. The response contains one word for each channel[3] temperature requested. The temperature returned as a signed fixed value with 8 bits below the binary point. • Channel 0: Samples the temperature value from the core fabric. • Channels 1 to 6: Samples the temperature value from the specified transceiver tile. Example: A Temperature of 10°C returns 0x00000A00 and a of temperature -1.5°C returns 0xFFFFFE80.
`QSPI_OPEN`	32	0	1	Requests exclusive access to the quad-serial peripheral interface (QSPI). If the SDM accepts the request (if the QSPI is not already in use or the SDM is not in the process of a device configuration), it returns the OK response, else it returns the error response. *Note:* The exclusive access is granted only to the client using this mailbox. Other clients are not be able to access the QSPI until it is closed by this client.
`QSPI_CLOSE`	33	0	1	Closes the exclusive access to the QSPI interface.
`QSPI_SET_CS`	34	1	1	Select which of the attached QSPI device via the chip select lines. Takes on one word argument as described below: • Bit [31:28]: Flash device to be selected. Bit setting `0000` is used to select flash that attached to `nCSO[0]`. • Bit [27:0]: Reserved (write as 0).
`QSPI_READ`	3A	2	N	Reads the attached QSPI device and takes two parameters: • The flash address offset from where you want to start reading from the QSPI device (one word). • Number of words to read (one word). A successful response returns an OK response code followed by the data read from the QSPI device. A failure response is either: • Returns an error code • Returns OK but partial of the data read from QSPI device is incorrect.

continued...

(2) The number does not include the command and response header.

(3) The availability of each transceiver tile varies among devices. For the temperature sensor channel numbers, refer to the related information.

Command	Code (Hex)	Number of Command Word[2]	Number of Response Word[2]	Description
				Note: The maximum transfer size is limited to 4K bytes and cannot be called while a configuration is in progress.
`QSPI_WRITE`	39	2+N	0	Writes data on the attached QSPI device and takes three parameters: • The flash address offset from where you want the command to start writing to the QSPI device (one word). • The number of words to write (one word). • The data to be written (one or more words). A successful write returns an OK response code. The client may need to issue `QSPI_ERASE` command before issuing this command to prepare the memory for writing. *Note:* The maximum transfer size is limited to 4K bytes and cannot be called while a configuration is in progress.
`QSPI_ERASE`	38	2	0	Erases sector on the attached QSPI device and takes two parameters: • The flash address offset within the device to start erasing from (one word). The address must be the start address of a sector within the flash memory. • The number of bytes to erase. The erase size is the multiple of 64K bytes. A successful erase returns an OK response code.
`QSPI_READ_DEVICE_REG`	35	2	N	Reads registers from the attached QSPI device and takes two parameters. • The opcode for the read command. • The number of bytes to read (the maximum size is 8 bytes). A successful read returns an OK response code followed by the data read from the device. If the data is not an exact multiple of 4 bytes then it is padded with 0 bytes until the next word boundary.
`QSPI_WRITE_DEVICE_REG`	36	2+N	0	Writes to registers on the attached QSPI and takes three arguments: • The opcode for the write command. • The number of bytes to write (the maximum size is 8 bytes). • The data to write (maximum 2 words, padded with 0 to word boundary). A successful write returns an OK response code.
`QSPI_SEND_DEVICE_OP`	37	1	0	Sends a command opcode to the QSPI and takes one argument: • The opcode to send the attached QSPI device. A successful command returns an OK response code.

[2] The number does not include the command and response header.

1.2.2. Error Code Response

Table 3. Error Code Response

Value	Error Code Response	Description
0	OK	Indicates that the command completed successfully. This does not mean that everything is necessarily ok, for example a pin to read status might return OK if it has successfully read the status. Further examination of the response arguments might be needed to see the error.
1	INVALID_COMMAND	Indicates that the command is in an incorrect format.
2	UNKNOWN_BR	Indicates that the command code is not understood. The device is running in the boot rom, load firmware for advanced command support.
3	UNKNOWN	Indicates that the command code is not understood by the currently loaded firmware.
256	NOT_CONFIGURED	Indicates that the device is not configured.
511	ALT_SDM_MBOX_RESP_DEVICE_BUSY	Indicates that the device is busy.
767	ALT_SDM_MBOX_RESP_NO_VALID_RESP_AVAILABLE	Indicates that there is no valid response available.
1023	ALT_SDM_MBOX_RESP_ERROR	General Error

1.3. Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-MM Interface

Table 4. Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-MM Interface

Signal Role	Width	Description
avmm_address	4	Avalon-MM address
avmm_write	1	Avalon-MM write request
avmm_read	1	Avalon-MM read request
avmm_writedata	32	Avalon-MM writedata bus
avmm_readdata	32	Avalon-MM readdata bus
avmm_readdatavalid	1	Avalon-MM readdata valid

1.4. Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-Memory Map

Table 5. Mailbox Client Intel Stratix 10 FPGA IP Core Avalon-Memory Map

Offset (word)	R/W	31	1	0
Base + 0	W	Command		
Base + 1	W	Command last word (eop)		
Base + 2	R	Command FIFO empty space		
Base + 3	W	Urgent command		

continued...

Offset (word)	R/W	31	1	0
Base + 4	R	Urgent FIFO empty space		
Base + 5	R	Response data		
Base + 6	R	Response FIFO fill level	EOP	SOP
Base + 7	R/W	Interrupt enable register (IER)		
Base + 8	R	Interrupt status register (ISR)		

1.4.1. Interrupt Enable Register

Table 6. Interrupt Enable Register

Having the enable bit cleared disregards the corresponding interrupt status bit from causing interrupt output assertion (IRQ).

Note: These enable bits does not prevent the value of interrupt status bit from showing up in ISR, it only prevents the interrupt status bit from causing interrupt output assertion.

Bit	Fields	Access	Default Value	Description
31:2	Reserved			
1	EN_CMD_FIFO_NOT_FULL	R/W	0x0	The enable bit interrupt of command FIFO is not full. • 1: Enable the corresponding interrupt • 0: Disable the corresponding interrupt
0	EN_DATA_VALID	R/W	0x0	The enable bit of data valid in response FIFO. • 1: Enable the corresponding interrupt • 0: Disable the corresponding interrupt

1.4.2. Interrupt Status Register

Table 7. Interrupt Status Register

Bit	Fields	Access	Default Value	Description
31:2	Reserved			
1	CMD_FIFO_NOT_FULL	R	0x0	Command FIFO is not full interrupt.[4] • 1: Indicates command FIFO is not full, the client can send more data into it. • 0: Indicates FIFO is full.
0	DATA_VALID	R	0x0	Data valid interrupt.[4] • 1: Indicates data exist in FIFO, master can read it out. • 0: Indicates the FIFO is full.

1.5. Using the Mailbox Client Intel Stratix 10 FPGA IP Core

Writing Command Packet

Command with one word (header only): Write to base + 1.

(4) This bit is cleared by operations on the FIFO. You do not need to clear this bit manually.

Command with more than one word (header + argument):

1. Write the header & argument (except the last one) to base + 0
2. Write last argument to base + 1

You can read from base + 2 which shows the remaining available free space in the FIFO for command. The FIFO fills up when the SDM is busy.

The behavior of the IP is undefined if you write to base + 0 and base + 1 while the FIFO is full. The write data is discarded.

Reading Response packet

1. Read base + 6 to check SOP, EOP & fill level of response FIFO. Proceed to step 2 if SOP = 1.

   For example, a return value of `0x00000007` indicates that a word is returned by SDM to fill up the response FIFO where the word represents the start of packet and also is the end of packet.
2. Read base + 5 to read response header. The *length* value contains the number (eg:n) of argument a response has.
3. Read base + 5 n times to retrieve all the response data.

1.6. Mailbox Client Intel Stratix 10 FPGA IP Core Use Case Examples

To use the Mailbox Client Intel Stratix 10 FPGA IP Core, an Avalon master is required and the simplest Avalon master is JTAG-to-Avalon Master.

The following examples are based on a Platform Designer system consist of JTAG-to-Avalon Master and the Mailbox Client Intel Stratix 10 FPGA IP core.

1.6.1. Example 1: Reading Intel Stratix 10 IDCODE and Voltage

```
#set base address according to Platform Designer system
set base 0x00000000
#set the variables to their respective offset
set b0    [expr {$base + 0x0}]
set b1    [expr {$base + 0x4}]
set b2    [expr {$base + 0x8}]
set b3    [expr {$base + 0xc}]
set b4    [expr {$base + 0x10}]
set b5    [expr {$base + 0x14}]
set b6    [expr {$base + 0x18}]
set b7    [expr {$base + 0x1c}]
set b8    [expr {$base + 0x20}]

#assign variable mp to the string that is the 0th element in the list returned
by get_service_paths master
set mp [lindex [get_service_paths master] 0]

#procedure to open the connection to the master module
proc start_service_master {} {
    global omp
    set omp [claim_service master $mp demo]
}

#procedure to close the connection to the master module
proc stop_service_master {} {
    global omp
    close_service master $omp
    set omp {}
}

#calling the start_service_master procedure
start_service_master

#writing a command without argument
#writing the command header to offset 1 of the SDM Mailbox IP (eg Get_IDCODE)
master_write_32 $omp $b1 0x00000010

#read offset 8 (Interrupt service register) to determine if there is data in
the FIFO
master_read_32 $omp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $omp $b6 1

#read offset 5 for the first packet of data (this will be the response header),
the length field will notify user how many packets of argument that is to follow
master_read_32 $omp $b5 1

#read offset 5 again to retrieve the response argument (in this case, this
command only has one response argument, which is the IDCODE)
master_read_32 $omp $b5 1


#writing a command with argument (eg: GET_VOLTAGE)
#writing the command header to offset 0 of the SDM Mailbox IP
master_write_32 $omp $b0 0x00001018

#writing the command argument to offset 1 (the voltage of interest is of
channel 0)
master_write_32 $omp $b1 0x00000001

#read offset 8 (Interrupt service register) to determine if there is data in
the FIFO
master_read_32 $omp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $omp $b6 1

#read offset 5 for the first packet of data (this will be the response header),
the length field will notify user how many packets of argument that is to follow
master_read_32 $omp $b5 1

#read offset 5 again to retrieve the response argument (in this case, this
command only has one response argument, which is the voltage of channel 0)
master_read_32 $omp $b5 1
stop_service_master
```

1.6.2. Example 2: Read and Write EPCQ-L or QSPI Devices

The following example shows the sequences and commands to read 1 word of data stored in the EPCQ-L or QSPI device.

1. Request an access to QSPI interfaces using QSPI_OPEN command:

```
#writing the command header to offset 1 of the SDM Mailbox IP by using
QSPI_OPEN command code
master_write_32 $mp $b1 0x00000032

#read offset 8 (Interrupt service register) to determine if there is valid
data in the FIFO
master_read_32 $mp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $mp $b6 1
```

```
#read offset 5 for the first packet of data (this will be the response
header). You are expecting to get the response packet to return
OK(0x0000000)
master_read_32 $mp $b5 1
```

2. Select the QSPI device using QSPI_SET_CS command:

```
#writing the command header to offset 0 of the SDM Mailbox IP by using the
QSPI_CS command code
master_write_32 $mp $b0 0x00001034

#writing the command argument to offset 1 (select QSPI flash attached to
nCSO[0])
master_write_32 $mp $b1 0x00000000

#read offset 8 (Interrupt service register) to determine if there is valid
data in the FIFO
master_read_32 $mp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $mp $b6 1

#read offset 5 for the first packet of data (this will be the response
header), the length field will notify user how many packets of argument
that is to follow.
master_read_32 $mp $b5 1

#read offset 5 again to retrieve the response argument (in this case, this
command only has one response argument, which is one word of data
0x62294895 stored in address offset 0x00000000)
master_read_32 $mp $b5 1
```

3. Start to read from selected QSPI device using QSPI_READ command:

```
#writing the command header to offset 0 of the SDM Mailbox IP (Specify the
QSPI_READ command code)
master_write_32 $mp $b0 0x0000203A

#writing the command argument to offset 0 (Specify the flash address offset
in one word)
master_write_32 $mp $b0 0x00000000

#writing the command argument to offset 1 (reading one word from flash
address offset 0x00000000)
master_write_32 $mp $b1 0x00000001

#read offset 8 (Interrupt service register) to determine if there is valid
data in the FIFO
master_read_32 $mp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $mp $b6 1

#read offset 5 for the first packet of data (this will be the response
header), the length field will notify user how many packets of argument
that is to follow. You are expecting to get the response packet to return
OK(0x0000000)
master_read_32 $mp $b5 1

#read offset 5 again to retrieve the response argument (in this case, this
command only has one response argument, which is one word of data
0x62294895 stored in address offset 0x00000000)
master_read_32 $mp $b5 1
```

4. Close the access to the QSPI interfaces using QSPI_CLOSE command:

```
#writing the command header to offset 1 of the SDM Mailbox IP by using
QSPI_CLOSE command code
master_write_32 $mp $b1 0x00000033
```

```
#read offset 8 (Interrupt service register) to determine if there is valid
data in the FIFO
master_read_32 $mp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $mp $b6 1
```

The following example shows the sequences and commands to write 1 word of data to an EPCQ-L or QSPI flash device.

1. Repeat step 1 and 2 in the example above to request exclusive access to QSPI interface and select QSPI flash attached to `nCSO[0]`.
2. Ensure you erase the content stored in the address that you would like to write to:
   a. Set the Write enable latch bit to 1:

```
#writing the command header to offset 0 of the SDM Mailbox IP by using
QSPI_SEND_DEVICE_OP command code
master_write_32 $mp $b1 0x00001037

#writing the command argument to offset 1 (the opcode for write enable
0x00000006)
master_write_32 $mp $b1 0x00000006

#read offset 8 (Interrupt service register) to determine if there is
valid data in the FIFO
master_read_32 $mp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $mp $b6 1

#read offset 5 for the first packet of data (this will be the response
header). You are expecting to get the response packet to return
OK(0x0000000)
master_read_32 $mp $b5 1
```

   b. Start erasing the contents:

```
#writing the command header to offset 0 of the SDM Mailbox IP (Specify
the QSPI_ERASE command code)
master_write_32 $mp $b0 0x00002038

#writing the command argument to offset 0 (Specify the flash address
offset in one word)
master_write_32 $mp $b0 0x00800000

#writing the command argument to offset 1 (Specify the number bytes to
erase in the multiple of 64K bytes)
master_write_32 $mp $b1 0x00004000

#read offset 8 (Interrupt service register) to determine if there is
valid data in the FIFO
master_read_32 $mp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $mp $b6 1

#read offset 5 for the first packet of data (this will be the response
header). You are expecting to get the response packet to return
OK(0x0000000)
master_read_32 $mp $b5 1
```

3. Start to write to the selected QSPI device using `QSPI_WRITE` command.

```
#writing the command header to offset 0 of the SDM Mailbox IP (Specify the
QSPI_WRITE command code)
master_write_32 $mp $b0 0x00003039
```

Send Feedback

```
#writing the command argument to offset 0 (Specify the flash address offset
in one word)
master_write_32 $mp $b0 0x00800000

#writing the command argument to offset 0 (To write a single word, specify
to 1)
master_write_32 $mp $b0 0x00000001

#writing the command argument to offset 1 (Specify the data to be written
for example like 0xdeadbeef)
master_write_32 $mp $b1 0xdeadbeef

#read offset 8 (Interrupt service register) to determine if there is valid
data in the FIFO
master_read_32 $mp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $mp $b6 1

#read offset 5 for the first packet of data (this will be the response
header). You are expecting to get the response packet to return
OK(0x0000000). This means that the data is written into the flash.
master_read_32 $mp $b5 1
```

Alternatively, you can do a QSPI_READ at address 0x0080000 to verify the data has been written properly into the targeted flash address.

4. Close the access to the QSPI interfaces using QSPI_CLOSE command:

```
#writing the command header to offset 1 of the SDM Mailbox IP by using
QSPI_CLOSE command code
master_write_32 $mp $b1 0x00000033

#read offset 8 (Interrupt service register) to determine if there is valid
data in the FIFO
master_read_32 $mp $b8 1

#read offset 6 for SOP and EOP of response packet
master_read_32 $mp $b6 1
```

1.7. Document Revision History for Mailbox Client Intel Stratix 10 FPGA IP Core User Guide

Document Version	Changes
2018.02.14	Initial release.