XPS InSystem Flash (v1.01b)

DS698 May 16, 2009 **Product Specification**

Introduction

The In System Flash (ISF) is type of serial flash memory and it is present only on the Spartan™-3AN devices. The Spartan™-3AN FPGA family is the sub-family of Spartan™-3A FPGA devices. The XPS InSystem Flash IP Core is 32-bit slave peripheral that connects to PLBv46 (Processor Local Bus with Xilinx simplifications) and provides access to the ISF using Serial Peripheral Interface (SPI) protocol.

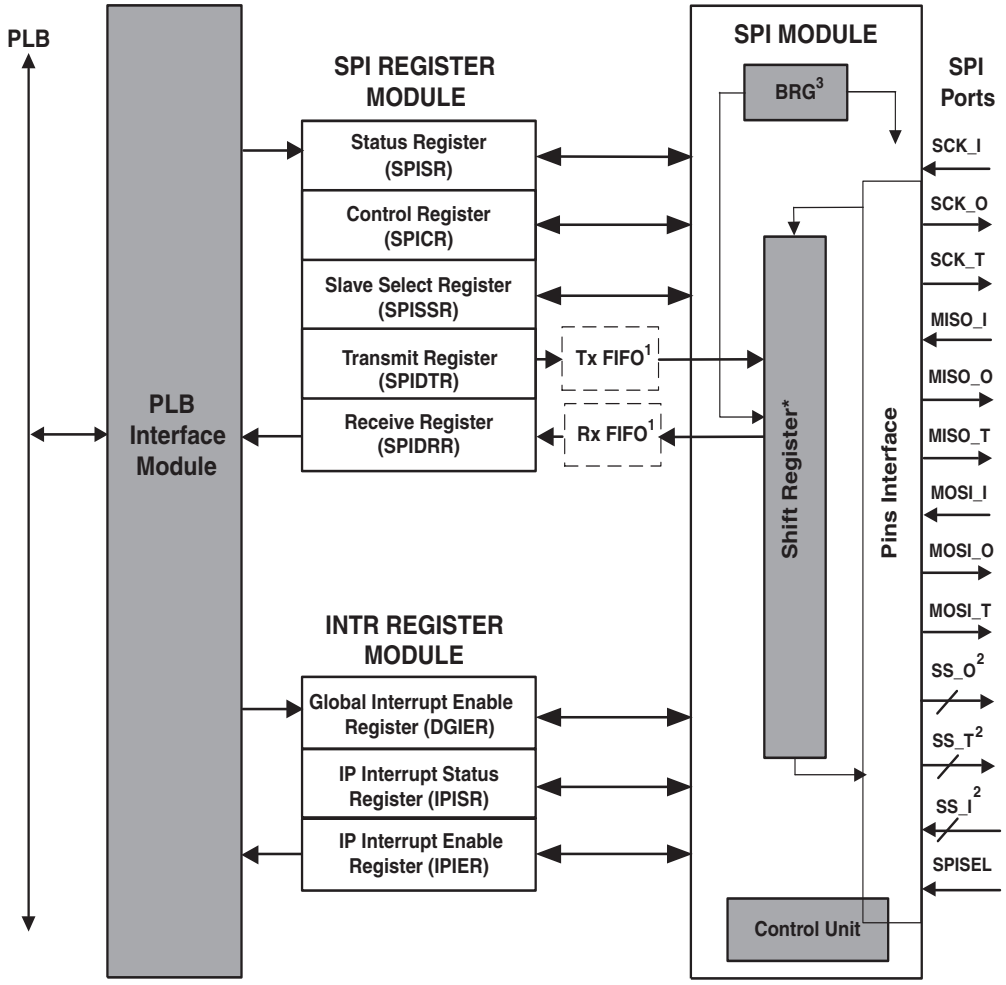
Features

- Connects as a 32-bit slave on PLB V4.6 buses of 32, 64 or 128 bits
- Supports four signal SPI interface (MOSI, MISO, SCK and $\overline{SS}$)
- Supports full-duplex operation
- Supports programable clock phase and polarity
- Supports manual slave select mode only
- Supports transfer length of 8-bits only
- Supports local loopback capability for internal testing of the core
- Supports single SPI master and single ISF slave environment
- Optional 16 element deep (an element is a byte) transmit and receive FIFOs

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Spartan™-3A	
Version of Core	xps_insystem_flash	v1.01b
Resources Used		
	Min	Max
Slices	Refer to the Table 16	
LUTs		
FFs		
Block RAMs	N/A	
Special Features	N/A	
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs & Application notes	N/A	
Additional Items	N/A	
Design Tool Requirements		
Xilinx Implementation Tools	ISE 11.1i or later	
Verification	ModelSim SE 6.4b or later	
Simulation	ModelSim SE 6.4b or later	
Synthesis	XST 11.1i or later	
Support		
Provided by Xilinx, Inc.		

Functional Description

The top level block diagram for the XPS InSystem Flash IP Core is shown in Figure 1.



1 = The width of Tx FIFO, Rx FIFO and *Shift Register depends on the value of generic C_NUM_TRANSFER_BITS
2 = The width of SS depends on the value of generic C_NUM_SS_BITS
3 = BRG stands for Baud Rate Generator
4= The above block diagram is same as XPS SPI core. In this core, apart from SPI ports, only SS_O, MOSI_O, MISO_I and SCK_O signals are useful. All remaming signals are connected with the default values,

Figure 1: **XPS InSystem Flash IP Core Top-Level Block Diagram**

The XPS InSystem Flash IP Core uses XPS SPI IP Core as base core. The ISF primitive is instantiated in the core along with the XPS SPI IP Core instantiation.

The XPS InSystem Flash IP Core provides a full-duplex synchronous channel that supports four-wire interface (receive, transmit, clock and slave-select) between PLBv46 master and ISF.

The XPS InSystem Flash IP Core supports Manual Slave Select Mode as the Default Mode of operation. This mode allows the user to manually control the slave select line by the data written to the slave select register. This allows transfers of an arbitrary number of elements without toggling the slave select line between elements. However, the user must toggle the slave select line before starting a new transfer.

The XPS InSystem Flash IP Core supports continuous transfer mode, when configured as master, the transfer continues till the data is available in transmit register/FIFO.

The XPS InSystem Flash IP Core is always configured as a SPI master and should not be configured as slave for its proper operation. The XPS InSystem Flash IP Core is targeted to access only one In System Flash (ISF) available on Spartan™-3AN family devices.

The XPS InSystem Flash IP Core can't communicate with off-chip SPI slave devices as the SPI interface signals won't come out of the core. The number of slaves is limited to 1, as this IP Core is targeted to be used only with single ISF present on Spartan™-3AN family devices.

All the other SPI and INTR registers are 32-bit wide. The XPS InSystem Flash IP Core supports only word access to all SPI and INTR register modules.

The XPS InSystem Flash IP Core modules are described in the sections below.

XPS SPI IP Core Module:

The XPS SPI IP Core is used as base core and it is instantiated in the core. The XPS SPI IP Core contains the following modules,-

- PLB Interface Module

The PLB Interface Module provides the interface to the PLB V4.6. The read and write transactions at the PLB are translated into equivalent IP Interconnect (IPIC) transactions. The register interfaces of the SPI connect to the IPIC. The PLB Interface Module also provides an address decoding service for XPS InSystem Flash Core.

- SPI Register Module

The SPI Register Module includes all memory mapped registers (as shown in Figure 1). It interfaces to the PLB. It consists of Status Register, Control Register, N-bit Slave Select Register ($N \leq 32$) and a pair of Transmit/Receive Registers.

- INTR Register Module

The INTR Register Module consists of interrupt related registers namely Device Global Interrupt Enable Register (DGIER), IP Interrupt Enable Register (IPIER) and IP Interrupt Status Register (IPISR).

- SPI Module

The SPI Module consists of a shift register, a parameterized baud rate generator (BRG) and a control unit. It provides the SPI interface, including the control logic and initialization logic. It is the heart of XPS SPI IP Core.

- Optional FIFOs

The Tx FIFO and Rx FIFO are implemented on both transmit and receive paths when enabled by the parameter C_FIFO_EXIST. The width of Tx FIFO and Rx FIFO depends on generic C_NUM_TRANSFER_BITS. The depth of these FIFO's is 16.

In System Flash Instance (Unisim Component):

The ISF (In System Flash) is a unisim library component named as SPI_ACCESS and it is instantiated in the core. This is available only on the Spartan™-3AN devices. The ISF provides the Mode-3 SPI protocol (CPHA = 1 and CPOL = 1) and it communicates using 8-bit mode. Please read the XPS InSystem Flash IP Core Parameter - Port Dependencies section carefully.

XPS InSystem Flash IP Core Design Parameters

To allow the user to obtain a XPS InSystem Flash IP Core that is uniquely tailored for the system, certain features can be parameterized. Parameterization affords a measure of control over the function, resource usage, and performance of the actually implemented XPS InSystem Flash IP Core. The features that can be parameterized are as shown in Table 1.

Table 1: **XPS InSystem Flash IP Core Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameters					
G1	Target FPGA family	C_FAMILY(1)	"spartan3a"	"spartan3a"	string
PLB Parameters					
G2	PLB base address	C_BASEADDR	Valid Address(2)	None(3)	std_logic_vector
G3	PLB high address	C_HIGHADDR	Valid Address(2)	None(3)	std_logic_vector
G4	PLB least significant address bus width	C_SPLB_AWIDTH	32	32	integer
G5	PLB data width	C_SPLB_DWIDTH	32, 64, 128	32	integer
G6	Shared bus topology	C_SPLB_P2P	0 = Shared bus topology(4)	0	integer
G7	PLB master ID bus Width	C_SPLB_MID_WIDTH	$\log_2$(C_SPLB_NUM_MASTERS) with a minimum value of 1	1	integer
G8	Number of PLB masters	C_SPLB_NUM_MASTERS	1 - 16	1	integer
G9	Width of the slave data bus	C_SPLB_NATIVE_DWIDTH	32	32	integer
G10	Burst support	C_SPLB_SUPPORT_BURSTS	0 = No burst support(5)	0	integer
XPS InSystem Flash IP Core Parameters					
G11	Include receive and transmit FIFOs	C_FIFO_EXIST	0 = FIFOs not included 1 = FIFOs included	1	integer

Table 1: **XPS InSystem Flash IP Core Design Parameters *(Contd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G12	SPI clock frequency ratio	C_SCK_RATIO	2, 4, 16, 32, Nx16 for N = 1, 2, 3,...,128	32	integer
G13	Total number of slave select bits	C_NUM_SS_BITS	1(6)	1	integer
G14	Select number of transfer bits as 8	C_NUM_ TRANSFER_BITS	8(6)	8	integer
G15	Select the subfamily of Spartan™-3AN devices	C_DEVICE (1)(7)	"3S50AN", "3S200AN", "3S400AN", "3S700AN", "3S1400AN"	"3S700AN"	string
G16	HEX file containing the memory initialization contents	C_SIM_MEM_FILE(1)(8)	Any file and directory name	"NONE"	string
G17	64-byte value	C_SIM_FACTORY_ ID(1)(9)	Any 64-byte value	all '0'	bit_ vector
G18	64-byte value	C_SIM_USER_ID(1)(10)	Any 64-byte value	all '1'	bit_ vector
G19	Choice of simulation delays for ISF	C_SIM_DELAY_TYPE	"SCALED"(11), "ACCURATE"	"SCALED"	string

Notes:

1. The in system flash (ISF) is present only on "spartan3an" FPGA family devices, which is considered to be the part of "spartan3a" FPGA family. So when "spartan3an" family is targeted, the user must use C_FAMILY = "spartan3a". Please refer the Spartan™-3AN FPGA In System Flash User Guide, UG333(v2.0) for more reference of ISF.
2. The range C_BASEADDR to C_HIGHADDR is the address range for the XPS InSystem Flash IP Core. This range is subject to restrictions to accommodate the simple address decoding scheme that is employed: The size, C_HIGHADDR - C_BASEADDR + 1, must be a power of two and must be at least 0x80 to accommodate all XPS InSystem Flash IP Core registers. However, a larger power of two may be chosen to reduce decoding logic. C_BASEADDR must be aligned to a multiple of the range size.
3. No default value will be specified to insure that an actual value appropriate to the system is set.
4. Point to point bus topology is not allowed.
5. Burst is not supported.
6. This parameter remains constant and user can't modify it. This parameter is added to support the XPS SPI IP Core parameter requirements.
7. The C_DEVICE parameter should be used along with C_FAMILY = "spartan3a". It will be auto picked up by the EDK tool while integrating the XPS InSystem Flash IP Core in EDK system. The value for the parameter should be one of the sub-family member of Spartan™-3AN.
8. The C_SIM_MEM_FILE is an optional generic. It represents the hex file containing the memory contents for ISF memory. Please refer the Spartan™-3AN FPGA In System Flash User Guide, UG333 for more reference.
9. The C_SIM_FACTORY_ID is an optional generic. The C_SIM_FACTORY_ID is a 64-byte factory-programmed unique indentifier number to be specified in device Security Register. Please refer the Spartan™-3AN FPGA In System Flash User Guide, UG333 for more reference.
10. The C_SIM_USER_ID is an optional generic. This a 64-byte factory-programmed unique indentifier number to be specified in device Security Register. Please refer the Spartan™-3AN FPGA In System Flash User Guide, UG333 for more reference.
11. The C_SIM_DELAY_TYPE is an simulation-only optional generic. It is recommended that user should leave it to the default value of "SCALED".

XPS InSystem Flash IP Core I/O Signals

The XPS InSystem Flash IP Core I/O signals are listed and described in Table 2.

Table 2: **XPS InSystem Flash IP Core I/O Signal Descriptions(1)**

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P1	SPLB_Clk	System	I	-	PLB clock
P2	SPLB_Rst	System	I	-	PLB reset, active high
P3	IP2INTC_Irpt	System	O	0	Interrupt control signal from SPI
PLB Master Interface Signals					
P4	PLB_ABus[0 : 31]	PLB	I	-	PLB address bus
P5	PLB_PAValid	PLB	I	-	PLB primary address valid
P6	PLB_masterID[0 : C_SPLB_MID_WIDTH - 1]	PLB	I	-	PLB current master identifier
P7	PLB_RNW	PLB	I	-	PLB read not write
P8	PLB_BE[0 : (C_SPLB_DWIDTH/8) - 1]	PLB	I	-	PLB byte enables
P9	PLB_size[0 : 3]	PLB	I	-	PLB size of requested transfer
P10	PLB_type[0 : 2]	PLB	I	-	PLB transfer type
P11	PLB_wrDBus[0 : C_SPLB_DWIDTH - 1]	PLB	I	-	PLB write data bus
Unused PLB Master Interface Signals					
P12	PLB_UABus[0 : 31]	PLB	I	-	PLB upper address bits
P13	PLB_SAValid	PLB	I	-	PLB secondary address valid
P14	PLB_rdPrim	PLB	I	-	PLB secondary to primary read request indicator
P15	PLB_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P16	PLB_abort	PLB	I	-	PLB abort bus request
P17	PLB_busLock	PLB	I	-	PLB bus lock
P18	PLB_MSize[0 : 1]	PLB	I	-	PLB data bus width indicator
P19	PLB_lockErr	PLB	I	-	PLB lock error
P20	PLB_wrBurst	PLB	I	-	PLB burst write transfer
P21	PLB_rdBurst	PLB	I	-	PLB burst read transfer
P22	PLB_wrPendReq	PLB	I	-	PLB pending bus write request
P23	PLB_rdPendReq	PLB	I	-	PLB pending bus read request
P24	PLB_wrPendPri[0 : 1]	PLB	I	-	PLB pending write request priority
P25	PLB_rdPendPri[0 : 1]	PLB	I	-	PLB pending read request priority

Table 2: **XPS InSystem Flash IP Core I/O Signal Descriptions(1)**

Port	Signal Name	Interface	I/O	Initial State	Description
P26	PLB_reqPri[0 : 1]	PLB	I	-	PLB current request priority
P27	PLB_TAttribute[0 : 15]	PLB	I	-	PLB transfer attribute
PLB Slave Interface Signals					
P28	Sl_addrAck	PLB	O	0	Slave address acknowledge
P29	Sl_SSize[0 : 1]	PLB	O	0	Slave data bus size
P30	Sl_wait	PLB	O	0	Slave wait
P31	Sl_rearbitrate	PLB	O	0	Slave bus rearbitrate
P32	Sl_wrDAck	PLB	O	0	Slave write data acknowledge
P33	Sl_wrComp	PLB	O	0	Slave write transfer complete
P34	Sl_rdDBus[0 : C_SPLB_DWIDTH - 1]	PLB	O	0	Slave read data bus
P35	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
P36	Sl_rdComp	PLB	O	0	Slave read transfer complete
P37	Sl_MBusy[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave busy
P38	Sl_MWrErr[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave write error
P39	Sl_MRdErr[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave read error
Unused PLB Slave Interface Signals					
P40	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P41	Sl_rdWdAddr[0 : 3]	PLB	O	0	Slave read word address
P42	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
P43	Sl_MIRQ[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave interrupt request

Notes:

1. Please note that the SPI related port signals will be consumed internally and will not be available outside.

XPS InSystem Flash IP Core Parameter - Port Dependencies

The dependencies between the XPS InSystem Flash IP Core design parameters and I/O signals are described in Table 3.

Table 3: **XPS InSystem Flash IP Core Parameter-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G5	C_SPLB_DWIDTH	P8, P11, P34	-	Affects the number of bits in data bus
G7	C_SPLB_MID_WIDTH	P6	G8	This value is calculated as: $\log_2$(C_SPLB_NUM_MASTERS) with a minimum value of 1
G8	C_SPLB_NUM_MASTERS	P37, P38, P39, P43	-	Affects the number of PLB masters
I/O Signals				
P6	PLB_masterID[0 : C_SPLB_MID_WIDTH - 1]	-	G7	Width of the PLB_mastedID varies according to C_SPLB_MID_WIDTH
P8	PLB_BE[0 : (C_SPLB_DWIDTH/8) -1]	-	G5	Width of the PLB_BE varies according to C_SPLB_DWIDTH
P11	PLB_wrDBus[0 : C_SPLB_DWIDTH - 1]	-	G5	Width of the PLB_wrDBus varies according to C_SPLB_DWIDTH
P34	Sl_rdDBus[0 : C_SPLB_DWIDTH - 1]	-	G5	Width of the Sl_rdDBus varies according to C_SPLB_DWIDTH
P37	Sl_MBusy[0 : C_SPLB_NUM_MASTERS - 1]	-	G8	Width of the Sl_MBusy varies according to C_SPLB_NUM_MASTERS
P38	Sl_MWrErr[0 : C_SPLB_NUM_MASTERS - 1]	-	G8	Width of the Sl_MWrErr varies according to C_SPLB_NUM_MASTERS
P39	Sl_MRdErr[0 : C_SPLB_NUM_MASTERS - 1]	-	G8	Width of the Sl_MRdErr varies according to C_SPLB_NUM_MASTERS
P43	Sl_MIRQ[0 : C_SPLB_NUM_MASTERS - 1]	-	G8	Width of the Sl_MIRQ varies according to C_SPLB_NUM_MASTERS

XPS InSystem Flash IP Core Register Descriptions

There are no separate registers for XPS InSystem Flash IP Core. All the XPS SPI IP Core registers are accessed through the XPS InSystem Flash IP Core.

Table 4 gives a summary of the XPS SPI IP Core internal registers, accessed through the XPS InSystem Flash IP Core. The transmit FIFO occupancy register and the receive FIFO occupancy register exists only when C_FIFO_EXIST = 1.

Table 4: **XPS SPI IP Core Registers**

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
SPI Core Grouping				
C_BASEADDR + 40	SRR	Write	N/A	Software Reset Register
C_BASEADDR + 60	SPICR	R/W	0x180	SPI Control Register
C_BASEADDR + 64	SPISR	Read	0x5	SPI Status Register
C_BASEADDR + 68	SPIDTR	Write	0x0	SPI Data Transmit Register A single register or a FIFO
C_BASEADDR + 6C	SPIDRR	Read	0x0	SPI Data Receive Register A single register or a FIFO
C_BASEADDR + 70	SPISSR	R/W	No slave is selected	SPI Slave Select Register
C_BASEADDR + 74	SPI Transmit FIFO Occupancy Register(1)	Read	0x0	Transmit FIFO Occupancy Register
C_BASEADDR + 78	SPI Receive FIFO Occupancy Register(1)	Read	0x0	Receive FIFO Occupancy Register
Interrupt Controller Grouping				
C_BASEADDR + 1C	DGIER	R/W	0x0	Device Global Interrupt Enable Register
C_BASEADDR + 20	IPISR	R/TOW(2)	0x0	IP Interrupt Status Register
C_BASEADDR + 28	IPIER	R/W	0x0	IP Interrupt Enable Register

Notes:

1. This register does not exist if C_FIFO_EXIST = 0.
2. TOW = Toggle On Write. Writing a 1 to a bit position within the register causes the corresponding bit position in the register to toggle.

Details of XPS InSystem Flash IP Core Registers

The XPS InSystem Flash IP Core uses XPS SPI IP Core as base core. Therefore, all the XPS SPI IP Core registers are listed and supported here. As the XPS SPI IP Core embedded in to XPS InSystem Flash and must be configured in the SPI Master Mode only, as the XPS SPI IP Core must not be configured in slave mode, some of the slave mode related register bits are non-applicable and must not be accessed while writing the write-only or read-write registers. While reading those un-used slave register bits from the read-only or read-write registers the particular slave operation related bit value should be either ignored or should be left to default values. Please read the notes mentioned for such bits in register bit description area.

Software Reset Register (SRR)

The Software Reset Register permits the programmer to reset the XPS InSystem Flash IP Core independent of other devices in the system. To activate software generated reset, the value 0x0000_000A must be written to this register. Any other write access generates an error condition with undefined results and generates an error. The bit assignment in the software reset register is shown in Figure 2 and described in Table 5. The effect of an attempt to read this register is undefined.

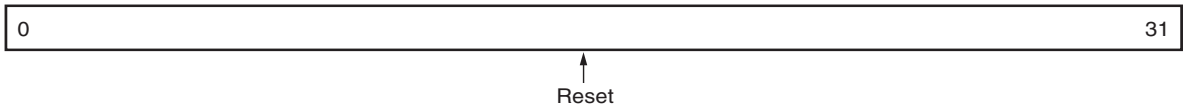


Figure 2: **Software Reset Register (C_BASEADDR + 0x40)**

Table 5: **Software Reset Register (SRR) Description (C_BASEADDR + 0x40)**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 31	Reset	Write only	N/A	The only allowed operation on this register is a write of 0x0000000A, which resets the XPS InSystem Flash IP Core. Read is not recommended.

After reset all the internal registers of the XPS SPI IP Core will be initialized to their own default values.

SPI Control Register (SPICR)

The SPI Control Register (SPICR) gives the programmer control over various aspects of the IP Core. The bit assignment in the SPICR is shown in Figure 3 and described in Table 6. While accessing ISF, please make sure that the XPS SPI IP Core is always configured in master mode only. The ISF supports the SPI Mode-3 transfer protocol, so care should be taken while initializing the SPI Control Register. It's bit-27 (CPHA) and bit-28 (CPOL) must always be written with '1' to support SPI Mode-3 protocol.

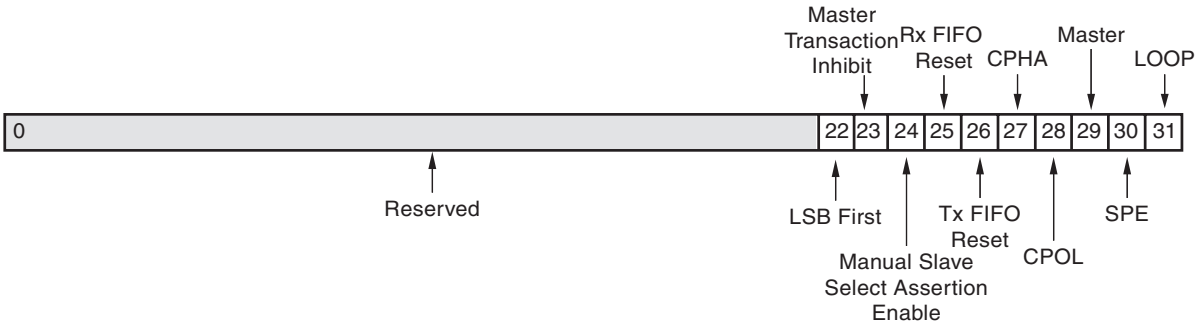


Figure 3: **SPI Control Register (C_BASEADDR + 0x60)**

Table 6: **SPI Control Register (SPICR) Description (C_BASEADDR + 0x60)**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 21	Reserved	N/A	N/A	Reserved
22	LSB First	R/W	'0'	**LSB First.** This bit selects LSB first data transfer format. The default transfer format is MSB first. '0' = MSB first transfer format '1' = LSB first transfer format **Note:** The ISF supports only MSB-to-LSB bit format. So this bit should always be set to default '0'.
23	Master Transaction Inhibit	R/W	'1'	**Master Transaction Inhibit.** This bit inhibits master transactions. This bit has no effect on slave operation. '0' = Master transactions enabled '1' = Master transactions disabled
24	Manual Slave Select Assertion Enable	R/W	'1'	**Manual Slave Select Assertion Enable.** This bit forces the data in the slave select register to be asserted on the slave select output anytime the device is configured as a master and the device is enabled (SPE asserted). This bit has no effect on slave operation. '1' = Slave select output follows data in slave select register **Note:** Please note that Automatic Slave Select Mode is not supported due to In system Flash specific requirements.
25	Rx FIFO Reset	R/W	'0'	**Receive FIFO Reset.** When written to '1', this bit forces a reset of the Receive FIFO to the empty condition. One PLB clock cycle after reset, this bit is again set to '0'. This bit is unassigned when the XPS InSystem Flash IP Core is not configured with FIFOs. '0' = Receive FIFO normal operation '1' = Reset receive FIFO pointer

Table 6: **SPI Control Register (SPICR) Description (C_BASEADDR + 0x60) *(Contd)***

Bit(s)	Name	Core Access	Reset Value	Description
26	Tx FIFO Reset	R/W	'0'	**Transmit FIFO Reset.** When written to '1', this bit forces a reset of the Transmit FIFO to the empty condition. One PLB clock cycle after reset, this bit is again set to '0'. This bit is unassigned when the XPS InSystem Flash IP Core is not configured with FIFOs. '0' = Transmit FIFO normal operation '1' = Reset transmit FIFO pointer
27	CPHA	R/W	'0'	**Clock Phase.** Setting this bit selects one of two fundamentally different transfer formats. **Note:** The ISF supports SPI Mode-3 protocol. To operate the ISF properly, it is required to set this bit to '1' during the core configuration.
28	CPOL	R/W	'0'	**Clock Polarity.** Setting this bit defines clock polarity. '0' = Active high clock; SCK idles low '1' = Active low clock; SCK idles high **Note:** The ISF supports SPI Mode-3 protocol. To operate the ISF properly, it is required to set this bit to '1' during the core configuration.
29	Master	R/W	'0'	**Master.** Setting this bit configures the SPI device as a master or a slave. '0' = Slave configuration '1' = Master configuration **Note:** To operate the core in master SPI mode, it is must to set this bit to '1' during the core configuration. No SPI slave mode is supported.
30	SPE	R/W	'0'	**SPI System Enable.** Setting this bit to '1' enables the SPI devices as noted below. '0' = SPI system disabled. '1' = SPI system enabled. Master outputs active (e.g. MOSI and SCK in idle state) and slave outputs will become active if $\overline{SS}$ becomes asserted. Master will start transfer when transmit data is available. **Note:** To operate the ISF properly, please follow configuration steps mentioned in SPI Registers Flow Description.
31	LOOP	R/W	'0'	**Local Loopback Mode.** Enables local loopback operation and is functional only in master mode. '0' = Normal operation '1' = Loopback mode. The transmitter output is internally connected to the receiver input. The receiver and transmitter operate normally, except that received data (from remote slave) is ignored. **Note:** Please make sure that the ISF is is not selected during the internal loop-back mode, else it will result in error condition. Note that the interrupt enable bit which resides at this bit position of the M68HC11 specification resides in the interrupt enable register in this implementation; see Specification Exceptions.

SPI Status Register (SPISR)

The SPI Status Register (SPISR) is a read-only register that gives the programmer visibility of the status of some aspects of the XPS InSystem Flash IP Core. Internally this register is a part of XPS SPI IP Core. The bit assignment in the SPISR is shown in Figure 4 and described in Table 7. Writing to the SPISR is not recommended and if it is done by mistake, then no change will be there in register contents.

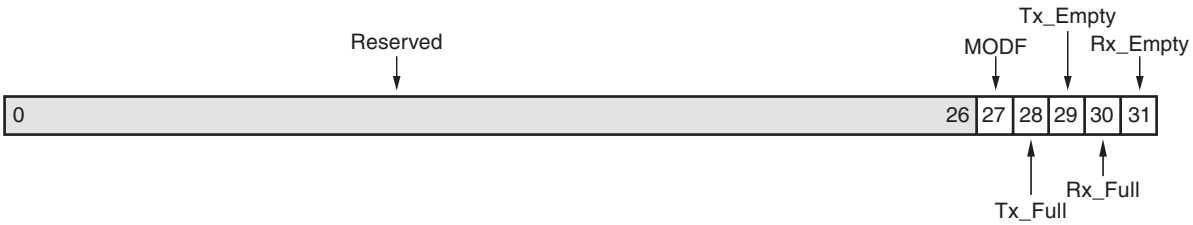


Figure 4: **SPI Status Register (C_BASEADDR + 0x64)**

Table 7: **SPI Status Register (SPISR) Description (C_BASEADDR + 0x64)**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 26	Reserved	N/A	N/A	Reserved
27	MODF	Read	'0'	**Mode-Fault Error Flag.** This flag is set if the $\overline{SS}$ signal goes active while the SPI device is configured as a master. MODF is automatically cleared by reading the SPISR. MODF does generate an interrupt with a single cycle strobe when the MODF bit transitions from a low to high. '0' = No error '1' = Error condition detected **Note:** As XPS InSystem Flash IP Core operates in the master SPI mode only, this bit will always be '0'.
28	Tx_Full	Read	'0'	**Transmit Full.** When a transmit FIFO exists, this bit will be set high when the transmit FIFO is full. When FIFOs don't exist, this bit is set high when an PLB write to the register has been made. This bit is cleared when the SPI transfer is completed.
29	Tx_ Empty	Read	'1'	**Transmit Empty.** When a transmit FIFO exists, this bit will be set high when the transmit FIFO is empty. The occupancy of the FIFO is decremented with the completion of each SPI transfer. When FIFOs don't exist, this bit is set with the completion of an SPI transfer. Either with or without FIFOs, this bit is cleared upon a PLB write to the FIFO or transmit register.
30	Rx_Full	Read	'0'	**Receive Full.** When a receive FIFO exists, this bit will be set high when the receive FIFO is full. The occupancy of the FIFO is incremented with the completion of each SPI transaction. When FIFOs don't exist, this bit is set high when an SPI transfer has completed. Rx_Empty and Rx_Full are complements in this case.
31	Rx_ Empty	Read	'1'	**Receive Empty.** When a receive FIFO exists, this bit will be set high when the receive FIFO is empty. The occupancy of the FIFO is decremented with each FIFO read operation. When FIFOs don't exist, this bit is set high when the receive register has been read. This bit is cleared at the end of a successful SPI transfer.

SPI Data Transmit Register (SPIDTR)

This register is written with a data to be transmitted on the SPI bus. Once the SPE bit is set to '1' in master mode the data is transferred from the SPIDTR to the shift register.

The SPIDTR is shown in Figure 5, while Table 8 shows specifics of the data format.

When a transmit FIFO exists, data is written directly in the FIFO and the first location in the FIFO is treated as the SPIDTR. The pointer is decremented after completion of each SPI transfer.

This register should not be read and used only for writing when it is known that space for the data is available. If an attempt to write is made on a full register or FIFO, then the PLB write transaction completes with an error condition. Reading to the SPIDTR is not allowed and the read transaction will result in undefined data.

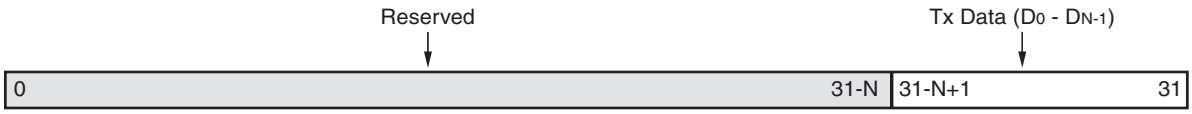


Figure 5: **SPI Data Transmit Register (C_BASEADDR + 0x68)**

Table 8: **SPI Data Transmit Register (SPIDTR) Description (C_BASEADDR + 0x68)**

Bit(s)	Name	Core Access	Reset Value	Description
0 - [31-N]	Reserved	N/A	N/A	Reserved
[31-N+1] - 31	Tx Data(1) (D_0 - D_{N-1})	Write only	0	N-bit SPI transmit data. N is only 8. The bit postion 31 represents N-1 data bit. The ISF supports 8-bit transfer mode only. N = 8 as C_NUM_TRANSFER_BITS = 8

Notes:
1. The D_{N-1} bit will always represent the MSB bit irrespective of "LSB first" or "MSB first" transfer selection.

SPI Data Receive Register (SPIDRR)

This register is used to read data that is received from the SPI bus. This is a double buffered register. The received data is placed in this register after each complete transfer. The SPI architecture does not provide any means for a slave to throttle traffic on the bus; consequently, the SPIDRR is updated following each completed transaction only if the SPIDRR was read prior to the last SPI transfer. If the SPIDRR was not read (i.e. is full), then the most recently transferred data will be lost and a receive over-run interrupt will occur. The same condition can occur with a master SPI device as well.

For the XPS SPI IP Core with a receive FIFO, the incoming data is buffered in the FIFO. The receive FIFO is a read only buffer. If an attempt to read an empty receive register or FIFO is made, then the PLB read transaction completes with an error condition. The effect is undefined if an attempt is made to write the SPIDRR. The SPIDRR is shown in Figure 6, while the specifics of the data format is described in Table 9.

Figure 6: **SPI Data Receive Register (C_BASEADDR + 0x6C)**

Table 9: **SPI Data Receive Register (SPIDRR) Description (C_BASEADDR + 0x6C)**

Bit(s)	Name	Core Access	Reset Value	Description
0 - [31-N]	Reserved	N/A	N/A	Reserved
[31-N+1] - 31	Rx Data(1) (D_0 - D_{N-1})	Read only	0	N-bit SPI receive data. N is only 8. The bit postion 31 represents N-1 data bit. The ISF supports 8-bit transfer mode only. N = 8 as C_NUM_TRANSFER_BITS = 8

Notes:
1. The D_{N-1} bit will always represent the MSB bit irrespective of "LSB first" or "MSB first" transfer selection.

SPI Slave Select Register (SPISSR)

This register contains an active-low, one-hot encoded Slave Select vector $\overline{SS}$ of length N, where N is the number of ISF slaves. In the XPS InSystem Flash IP Core the ISF is the only slave. So this register will always be of 1-bit length. The default value is tied to '1'. The bits of $\overline{SS}$ occupy the right-most bits of the register.

The bit assignment in the SPISSR is shown in Figure 7 and described in Table 10.

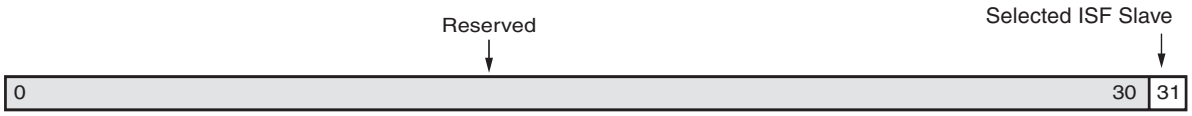


Figure 7: **SPI Slave Select Register (C_BASEADDR + 0x70)**

Table 10: **SPI Slave Select Register (SPISSR) Description (C_BASEADDR + 0x70)**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 30	Reserved	N/A	N/A	Reserved
31	Selected Slave	R/W	1	Active-low. In order to select the ISF as slave, this bit should be programmed to '0'.

SPI Transmit FIFO Occupancy Register (Tx_FIFO_OCY)

The SPI Transmit FIFO Occupancy Register is present if and only if XPS InSystem Flash IP Core is configured with FIFOs (C_FIFO_EXIST = 1). If it is present and if the Transmit FIFO is not empty, the register contains a four-bit, right-justified value that is one less than the number of elements in the FIFO (occupancy minus one).

This register is a read only. The effect of a write to it (or of a read when the FIFO is empty) will not affect the register contents. The only reliable way to determine that the FIFO is empty is by reading the Tx_Empty status bit in the SPI Status Register or the DTR Empty bit in the Interrupt Status Register.

The Transmit FIFO Occupancy register is shown in Figure 8, while the specifics of the data format is described in Table 11.

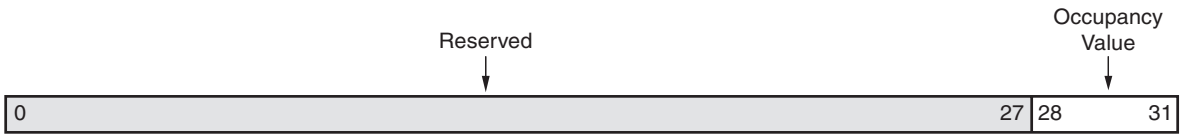


Figure 8: **SPI Transmit FIFO Occupancy Register (C_BASEADDR + 0x74)**

Table 11: **SPI Transmit FIFO Occupancy Register Description (C_BASEADDR + 0x74)**

Bit(s)	Name	Core Access	Reset Value (hex)	Description
0 - 27	Reserved	N/A	N/A	Reserved
28 - 31	Occupancy Value	Read	0	Bit 28 is the MSB. The binary value plus 1 yields the occupancy.

SPI Receive FIFO Occupancy Register (Rx_FIFO_OCY)

The SPI Receive FIFO Occupancy Register is present if and only if XPS InSystem Flash IP Core is configured with FIFOs (C_FIFO_EXIST = 1). If it is present and if the Receive FIFO is not empty, the register contains a four-bit, right-justified value that is one less than the number of elements in the FIFO (occupancy minus one).

This register is a read only. The effect of a write to it (or of a read when the FIFO is empty) will not affect the register contents. The write operation is not recommended on this register. The only reliable way to determine that the FIFO is empty is by reading the Rx_Empty status bit in the SPI Status Register.

The Receive FIFO Occupancy register is shown in Figure 9, while the specifics of the data format is described in Table 12.

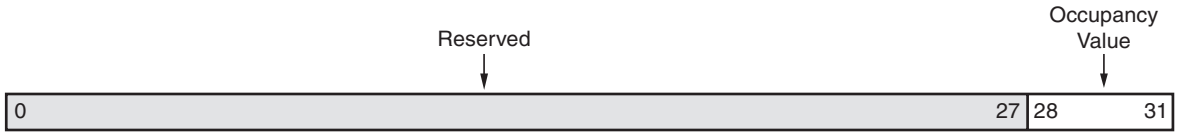


Figure 9: **SPI Receive FIFO Occupancy Register (C_BASEADDR + 0x78)**

Table 12: **SPI Receive FIFO Occupancy Register Description (C_BASEADDR + 0x78)**

Bit(s)	Name	Core Access	Reset Value (hex)	Description
0 - 27	Reserved	N/A	N/A	Reserved
28 - 31	Occupancy Value	Read	0	Bit 28 is the MSB. The binary value plus 1 yields the occupancy.

XPS InSystem Flash IP Core Interrupt Descriptions

The XPS InSystem Flash IP Core has instantiated XPS SPI IP Core which supports number of distinct interrupts that are sent to the interrupt controller module which is one of the sub-modules of XPS SPI IP Core. The Interrupt controller module allows each interrupt to be enabled independently (via the IP interrupt enable register (IPIER)).

The interrupt registers are in the interrupt module. The XPS InSystem Flash IP Core permits multiple conditions for an interrupt, or an interrupt strobe which occurs only after the completion of a transfer.

Setting the parameter C_FIFO_EXIST = 1 makes available all the interrupts shown in Table 14.

Setting the parameter C_FIFO_EXIST = 0 makes available all the interrupts except bit(25), i.e. Tx FIFO Half Empty, which is not present in this case.

Device Global Interrupt Enable Register (DGIER)

The Device Global Interrupt Enable Register is used to globally enable the final interrupt output from the Interrupt controller as shown in Figure 10 and described in Table 13. This bit is a read/write bit and is cleared upon reset.

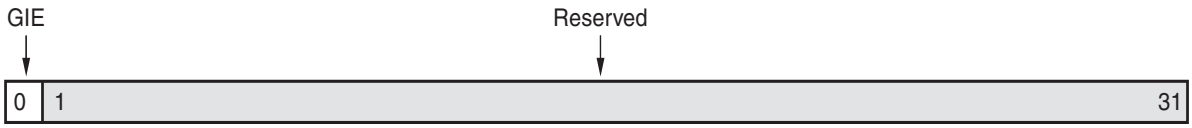


Figure 10: **Device Global Interrupt Enable Register (C_BASEADDR + 0x1C)**

Table 13: **Device Global Interrupt Enable Register (DGIER) Description(C_BASEADDR + 0x1C)**

Bit(s)	Name	Access	Reset Value	Description
0	GIE	R/W	'0'	**Global Interrupt Enable.** It enables all individually enabled interrupts to be passed to the interrupt controller. '0' = Disabled '1' = Enabled
1 - 31	Reserved	N/A	N/A	Reserved

IP Interrupt Status Register (IPISR)

Up to seven unique interrupt conditions are possible depending upon whether the system is configured with FIFOs or not. A system without FIFOs has six interrupts.

Please note that these are total number of interrupts XPS SPI IP Core supports internally. As for XPS InSystem Flash core XPS SPI IP Core is configured only in master SPI mode, some of the slave operation related interrupt signals may not be useful at all.

The Interrupt controller has a register that can enable each interrupt independently. Bit assignment in the Interrupt register for a 32-bit data bus is shown in Figure 11 and described in Table 14. The interrupt register is a read/toggle on write register and by writing a '1' to a bit position within the register causes the corresponding bit position in the register to 'toggle'. All register bits are cleared upon reset.

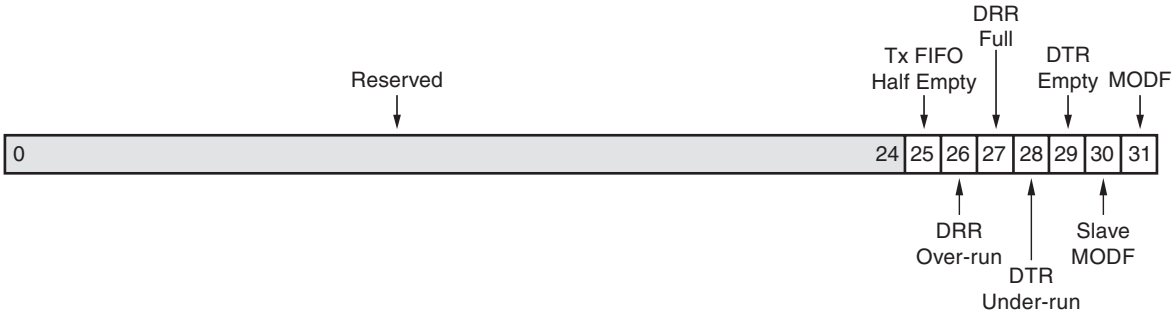


Figure 11: **IP Interrupt Status Register (IPISR) (C_BASEADDR + 0x20)**

Table 14: **IP Interrupt Status Register (IPISR) Description (C_BASEADDR + 0x20)**

Bit(s)	Name	Access	Reset Value	Description
0 - 24	Reserved	N/A	N/A	Reserved
25	Tx FIFO Half Empty	R/TOW(1)	'0'	**Transmit FIFO Half Empty.** IPISR bit(25) is the transmit FIFO half empty interrupt. This bit is set by a one-clock period strobe to the interrupt register when the occupancy value is decremented from "1000" to "0111". Note that "0111" means there are 8 elements in the FIFO to be transmitted. This interrupt exists only if the XPS InSystem Flash IP Core is configured with FIFOs.
26	DRR Over-run	R/TOW(1)	'0'	**Data Receive Register/FIFO Over-run.** IPISR bit(26) is the data receive FIFO over-run interrupt. This bit is set by a one-clock period strobe to the interrupt register when an attempt to write data to a full receive register or FIFO is made by the SPI core logic in order to complete an SPI transfer. This can occur when the SPI device is in master mode.
27	DRR Full	R/TOW(1)	'0'	**Data Receive Register/FIFO Full.** IPISR bit(27) is the data receive register full interrupt. Without FIFOs, this bit is set at the end of an SPI element (An element is a byte) transfer by a one-clock period strobe to the interrupt register. With FIFOs, this bit is set at the end of the SPI element transfer when the receive FIFO has been filled by a one-clock period strobe to the interrupt register.
28	DTR Under-run	R/TOW(1)	'0'	**Data Transmit Register/FIFO Under-run.** IPISR bit(28) is the data transmit register/FIFO under-run interrupt. This bit is set at the end of an SPI element transfer by a one-clock period strobe to the interrupt register when data is requested from an "empty" transmit register/FIFO by the SPI core logic in order to perform an SPI transfer. **Note:** As XPS InSystem Flash IP Core operates in the master SPI mode only, this bit will always be '0'.

Table 14: **IP Interrupt Status Register (IPISR) Description (C_BASEADDR + 0x20) *(Contd)***

Bit(s)	Name	Access	Reset Value	Description
29	DTR Empty	R/TOW(1)	'0'	**Data Transmit Register/FIFO Empty.** IPISR bit(29) is the data transmit register/FIFO empty interrupt. Without FIFOs, this bit is set at the end of an SPI element transfer by a one-clock period strobe to the interrupt register. With FIFOs, this bit is set at the end of the SPI element transfer when the transmit FIFO is emptied by a one-clock period strobe to the interrupt register. In the context of the M68HC11 reference manual, when configured without FIFOs, this interrupt is equivalent in information content to the complement of SPI transfer complete flag ($\overline{\text{SPIF}}$) interrupt bit. In master mode if this bit is set to '1' no more SPI transfers are permitted.
30	Slave MODF	R/TOW(1)	'0'	**Slave Mode-Fault Error.** IPISR bit(30) is the slave mode-fault error flag. This interrupt is generated if the $\overline{\text{SS}}$ signal goes active while the SPI device is configured as a slave but is not enabled. This bit is set immediately upon $\overline{\text{SS}}$ going active and continually set if $\overline{\text{SS}}$ is active and the device is not enabled. **Note:** As XPS InSystem Flash IP Core operates in the master SPI mode only, this bit will always be set to '0'.
31	MODF	R/TOW(1)	'0'	**Mode-Fault Error.** IPISR bit(31) is the mode-fault error flag. This interrupt is generated if the $\overline{\text{SS}}$ signal goes active while the SPI device is configured as a master. This bit is set immediately upon $\overline{\text{SS}}$ going active. **Note:** As XPS InSystem Flash IP Core operates in the master SPI mode only, this bit will always be set to '0'.

Notes:
1. TOW = Toggle On Write. Writing a '1' to a bit position within the register causes the corresponding bit position in the register to toggle.

IP Interrupt Enable Register (IPIER)

The IPIER has an enable bit for each defined bit of the IPISR as shown in Figure 12 and described in Table 15. All bits are cleared upon reset.

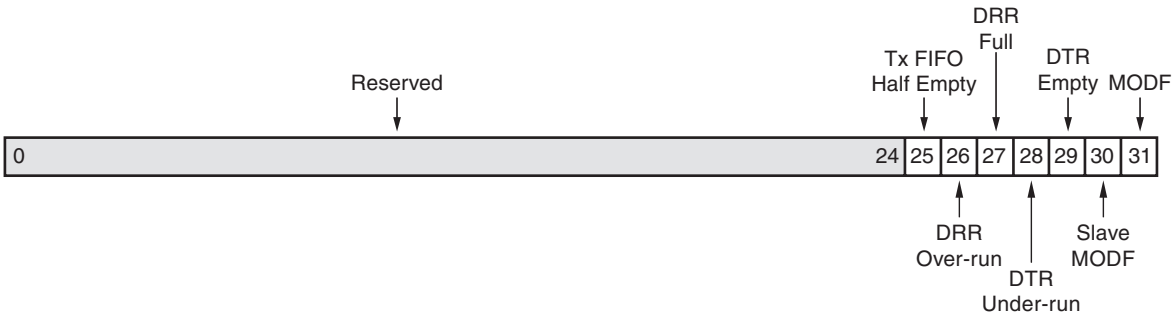


Figure 12: **IP Interrupt Enable Register (IPIER) (C_BASEADDR + 0x28)**

Table 15: **IP Interrupt Enable Register (IPIER) Description (C_BASEADDR + 0x28)**

Bit(s)	Name	Access	Reset Value	Description
0 - 24	Reserved	N/A	N/A	Reserved
25	Tx FIFO Half Empty	R/W	'0'	**Transmit FIFO Half Empty.** '0' = Disabled '1' = Enabled
26	DRR Over-run	R/W	'0'	**Receive FIFO Over-run.** '0' = Disabled '1' = Enabled **Note:** Please make sure that the XPS InSystem Flash IP Core is configured only in the master SPI mode.
27	DRR Full	R/W	'0'	**Data Receive Register/FIFO Full.** '0' = Disabled '1' = Enabled
28	DTR Under-run	R/W	'0'	**Data Transmit FIFO Under-run.** '0' = Disabled '1' = Enabled **Note:** As XPS InSystem Flash IP Core operates in the master SPI mode only, this bit must be always tied to '0'.
29	DTR Empty	R/W	'0'	**Data Transmit Register/FIFO Empty.** '0' = Disabled '1' = Enabled
30	Slave MODF	R/W	'0'	**Slave Mode-Fault Error Flag.** '0' = Disabled '1' = Enabled **Note:** As XPS InSystem Flash IP Core operates in the master SPI mode only, this bit must be always set to '0'.
31	MODF	R/W	'0'	**Mode-Fault Error Flag.** '0' = Disabled '1' = Enabled **Note:** As XPS InSystem Flash IP Core operates in the master SPI mode only, this bit must be always set to '0'.

XPS InSystem Flash IP Core Design Description

SPI Device Features

In addition to the features listed in the Features section, the SPI device also includes the following standard features:

- Single-master environment supported
- Single-slave environment supported. Only one ISF SPI slave device is supported.
- Supports maximum SPI clock rates up to one-half of the PLB clock rate. (Please check the ISF timing requirements from Spartan™-3AN FPGA In System Flash User Guide, UG333).
- Parameterizable baud rate generator.

- Back to Back transactions are supported, which means there can be multiple byte transfers taking place without interruption provided the transmit FIFO never gets empty and receive FIFO never gets full.
- All SPI transfers are full-duplex where an 8-bit data character is transferred from the master to the slave and an independent 8-bit data character is transferred from the slave to the master. This can be viewed as a circular 16-bit shift register; an 8-bit shift register in the SPI master device and another 8-bit shift register in a SPI slave device that are connected.

Optional FIFOs

Please note that the implementation of FIFO is a part of XPS SPI IP Core. So following explaination holds true for XPS InSystem Flash IP Core.

The user has the option to include FIFOs in the XPS InSystem Flash IP Core as shown in Figure 1. These FIFO's are internally configured from XPS SPI IP Core. Since SPI is full-duplex, both transmit and receive FIFOs are instantiated as a pair.

When FIFOs are implemented, the slave select address is required to be the same for all data buffered in the FIFOs. This is required because a FIFO for the slave select address is not implemented. Both transmit and receive FIFOs are 16 elements deep and are accessed via single PLB transactions since burst mode is not supported.

The transmit FIFO is write-only. When data is written in the FIFO, the occupancy number is incremented and when an SPI transfer is completed, the number is decremented. As a consequence of this operation, aborted SPI transfers still has the data available for the transmission retry. The transfers can only be aborted in the master mode by setting Master Transaction Inhibit bit, bit(23) of SPICR to '1' during a transfer. Setting this bit in the slave mode has no affect on the operation of the slave. These aborted transfers are on the SPI interface. The occupancy number is a read-only register.

If a write is attempted when the FIFO is full, then acknowledgement is given along with an error signal generation. Interrupts associated with the transmit FIFO include data transmit FIFO empty, transmit FIFO half empty and transmit FIFO under-run. See the section on XPS InSystem Flash IP Core Interrupt Descriptions for details.

The receive FIFO is read-only. When data is read from the FIFO, the occupancy number is decremented and when the SPI transfer is completed, the number is incremented. If a read is attempted when the FIFO is empty, then acknowledgement is given along with an error signal generation. When the receive FIFO becomes full, the receive FIFO full interrupt is generated.

Data is automatically written to the FIFO from the SPI module shift register after the completion of the SPI transfer. If the receive FIFO is full and more data is received, then a receive FIFO overflow interrupt is issued. When this happens, all data attempted to be written to the full receive FIFO by the SPI module is lost.

SPI transfers, when the XPS InSystem Flash IP Core is configured with FIFOs, can be started in two different ways depending on when the enable bit in the SPICR is set. If the enable bit is set prior to the first data being loaded in the FIFO, then the SPI transfer begins immediately after the write to the master transmit FIFO. If the FIFO is emptied via SPI transfers before additional elements are written to the transmit FIFO, an interrupt will be asserted. When the PLB to SPI SCK frequency ratio is sufficiently small, this scenario is highly probable.

Alternatively, the FIFO can be loaded up to 16 elements and then the enable bit can be set which starts the SPI transfer. In this case, an interrupt is issued after all elements are transferred. In all cases, more

data can be written to the transmit FIFOs to increase the number of elements transferred before emptying the FIFOs.

Local Master Loopback Operation

Local master loopback operation, although not included in the M68HC11 reference manual, has been implemented to expedite core internal testing. This operation is selected via setting the loop bit in the SPICR, the transmitter output is internally connected to the receiver input. The receiver and transmitter operate normally, except that received data (from remote slave) is ignored.

Hardware Error Detection

Under-run and over-run conditions error detection is provided as well. Under-run conditions can happen only if XPS SPI IP Core is configured in slave mode operation. As for XPS InSystem Flash IP Core, the XPS SPI IP Core is configured in master mode always, the under-run error will not occur. so no under-run related interrupts will be generated. Over-run can happen to both master and slave devices where a transfer occurs when the receive register or FIFO is full. During an over-run condition, the data received in that transfer is not registered (i.e. it is lost) and the IPISR over-run interrupt bit(26) is asserted.

Precautions to be Taken while Assigning the C_SCK_RATIO Parameter

XPS InSystem Flash IP Core is tested in hardware with the ISF (on Spartan™-3AN family devices only). Please read the data sheet of targeted Spartan™-3AN FPGA family device for ISF size and other timing related parameters. It is user's responsibility to mention the correct values while deciding the PLB clock and selecting the C_SCK_RATIO parameter of the core. The PLB clock and the C_SCK_RATIO will decide the clock at SCK pin of XPS InSystem Flash IP Core.

SPI Protocol Slave Select Assertion Modes

The SPI protocol is designed to have manual slave select assertion which is described in the following sections. In XPS InSystem Flash IP Core only the manual slave mode is allowed.

SPI Protocol with Manual Slave Select Assertion

This section briefly describes the SPI protocol where slave select ($\overline{SS}$) is manually asserted by the user (i.e. SPICR bit(24) = 1).

This is the configuration mode provided to permit transfers of an arbitrary number of elements without toggling slave select until all the elements are transferred. In this mode, the data in the SPISSR register appears directly on the $\overline{SS}$ output.

As described earlier, SCK must be stable before the assertion of slave select. Therefore, when manual slave select mode is utilized, the SPI master must be enabled first (SPICR bit(24) = 1) to assert SCK to the idle state prior to asserting slave select.

Note that the master transfer inhibit (SPICR bit(23)) can be utilized to inhibit master transactions until the slave select is asserted manually and all data registers of FIFOs are initialized as desired. This can be utilized before the first transaction and after any transaction that is allowed to complete.

SPI Registers Flow Description

This section provides information on setting the SPI registers to initiate and complete bus transactions.

XPS InSystem Flash IP Core in SPI master mode with FIFOs where the ISF is selected as slave using Slave Select Register and enabled via SPICR bit(24) assertion

This flow permits the transfer of N number of bytes by toggling of the slave select line just once. This is the default mode of operation. Please note that only Manual Slave Select mode must be configured for the proper operation of the core. Follow these steps to successfully complete an SPI transaction:

1. Configure DGIER and IPIER registers as desired.
2. Insure SPISSR register has all ones.
3. Write configuration data to core's SPICR as desired including setting bit(24) for manual asserting of $\overline{SS}$ and setting both enable bit and master transfer inhibit bit to logic '1'. This initializes SCK and MOSI but inhibits transfer. Make sure that the bit-27 (CPHA) and bit-28 (CPOL) of the SPICR register are set to '1', this will make XPS SPI IP Core to operate in SPI Mode-3 protocol.
4. Write initial data to core's FIFO at SPIDTR register address. At this moment it is assumed that the SPI master is disabled.
5. Write to SPISSR to manually assert $\overline{SS}$ (Slave Select active low).
6. Write the above configuration data to master SPI device SPICR, but clear Master Inhibit Bit which starts transfer.
7. Wait for interrupt (typically IPISR bit(30)) or poll status for completion. Wait time depends on SPI clock ratio.
8. Set Master Transaction Inhibit bit in SPICR so that the core can service interrupt request.
9. Write new data to FIFO at SPIDTR address and then clear master transaction inhibit bit in SPICR to continue N 8-bit element transfer. Note that an overrun of the SPIDRR FIFO can occur if the SPIDRR FIFO is not read properly. Also note that SCK will have "stretched" idle levels between element transfers (or groups of element transfers if utilizing FIFOs) and that MOSI can transition at end of a element transfer (or group of transfers) but will be stable at least one-half SCK period prior to sampling edge of SCK.
10. Repeat above steps until all data is transferred.
11. Write all ones to SPISSR or exit manual slave select assert mode to deassert $\overline{SS}$ vector while SCK and MOSI are in the idle state.
12. Disable XPS InSystem Flash IP Core as desired.

XPS InSystem Flash IP Core in SPI master mode without FIFOs and ISF as slave device performing single command (example: ISF status read command, which is of 4 bytes) transfer.

Follow these steps to successfully complete an SPI transaction:

1. Configure master DGIER and IPIER. Also configure slave DGIER and IPIER registers as desired.
2. Write configuration data to master SPI device SPICR as required. Do not enable the core at this moment. This can be done by setting the Master Transaction Inhibit bit to 1.
3. Write data to SPIDTR register.
4. Write into the SPICR to set the SPE bit to '1' to start the transmission.
5. Write the active-low, one-hot encoded ISF slave select address to the master SPISSR.
6. Write the SPICR (disable the Master Transaction Inhibit)to start the data transmission.
7. Update the SPIDTR with the next byte of data and Read the SPIDRR.
8. Wait for interrupt (typically IPISR bit(30)) or poll status for completion.
9. Read IPISR of the core.

10. Perform interrupt request service routine as required.
11. Repeat the step from 4 till the complete 4 bytes are transferred.
12. Perform actions as required or dictated by SPISR data, which also includes reading the data from SPIDRR register.

Design Constraints

Note: An example UCF for this core is available and must be modified for use in the system. Please refer to the *EDK Getting Started Guide* for the location of this file.

Timing Constraints

For complete timing coverage, it is recommended that Flip-Flop edge to edge constraint is specified along with SPLB_Clk input constraint. An example is shown in Figure 13.

```
NET "SPLB_CLK" TNM_NET = "splb_clk";
TIMESPEC "TS_splb_clk" = PERIOD "splb_clk" 10 ns HIGH 50%;

NET "*XPS_InSystem_FLASH_I/XPS_SPI_MODULE_I/I_SPI_MODULE/Serial_Dout" TIG;
NET "*XPS_InSystem_FLASH_I/XPS_SPI_MODULE_I/I_SPI_MODULE/SS_O<0>" TIG;
NET  "XPS_InSystem_FLASH_I/miso_from_isf" TIG;
```

Figure 13: **Timing Constraints**

Design Implementation

Target Technology

The intended target technology is Spartan™-3an family FPGAs.

Device Utilization and Performance Benchmarks

Core Performance

Since the XPS InSystem Flash IP Core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When the XPS InSystem Flash IP Core is combined with other designs in the system, the utilization of FPGA resources and timing of the XPS InSystem Flash IP Core design will vary from the results reported here.

The XPS InSystem Flash IP Core resource utilization for various parameter combinations measured with Spartan™-3A as the target device are detailed in Table 16.

Table 16: **Performance and Resource Utilization Benchmarks on Spartan-3a (xc3s700an-fgg484-4)**

Parameter Values (other parameters at default values)				Device Resources			Performance
C_FIFO_EXIST	C_SCK_RATIO	C_NUM_SS_BITS	C_NUM_TRANSFER_BITS	Slices	Slice Flip-Flops	LUTs	Fmax (MHz)
0	2	1	8	220	178	213	104
1	2	1	8	240	179	267	103
0	4	1	8	217	183	207	108
1	4	1	8	229	181	261	104
0	32	1	8	215	186	211	102
1	32	1	8	244	184	265	102

System Performance

To measure the system performance (Fmax) of this core, this core was added to a Spartan-3A system as the Device Under Test (DUT) as shown in Figure 14.

Because the XPS InSystem Flash IP Core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When this core is combined with other designs in the system, the utilization of FPGA resources and timing of the core design will vary from the results reported here.

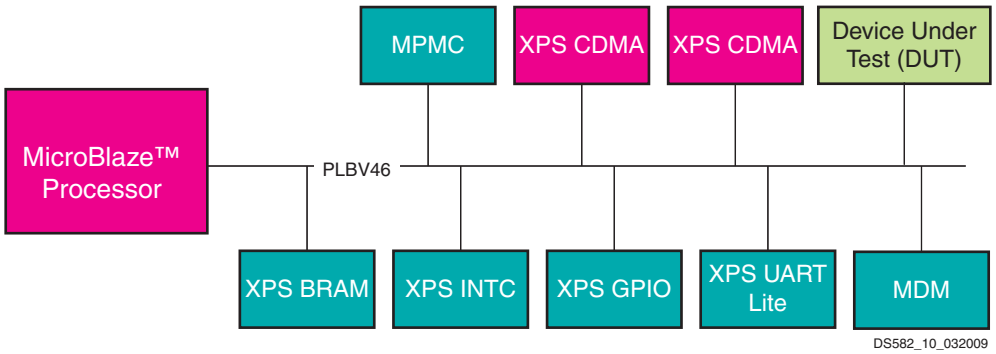


Figure 14: **Spartan-3A System**

The target FPGA was then filled with logic to drive the LUT and BRAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target FMax numbers are shown in Table 17.

Table 17: **XPS InSystem Flash IP Core System Performance**

Target FPGA	Target Fmax (MHz)
S3A700an-4	90

The target Fmax is influenced by the exact system and is provided for guidance. It is not a guaranteed value across all systems.

Specification Exceptions

Exceptions from the Motorola M68HC11-Rev. 4.0 Reference Manual

All the design exceptions of XPS SPI IP Core will remain same for the XPS InSystem Flash IP Core as well. For reference only, all the possible exceptions are listed here.

1. A slave mode-fault error interrupt is added to provide an interrupt if a SPI device is configured as a slave and is selected when not enabled.
2. In this design, the SPIDTR and SPIDRR registers have independent addresses. This is an exception to the M68HC11 specification which calls for two registers to have the same address.
3. All $\overline{SS}$ signals are required to be routed between SPI devices internally to the FPGA. This is because toggling of the $\overline{SS}$ signal is utilized in slaves to minimize FPGA resources.
4. Manual control of the $\overline{SS}$ signals is provided by setting bit(24) in the SPICR register. When the device is configured as a master and is enabled and bit(24) of the SPICR register is set, the vector in the SPISSR register is asserted. When this mode is enabled, multiple elements can be transferred without toggling the $\overline{SS}$ vector.

5. A control bit is provided to inhibit master transfers. This bit is effective in any master mode, but has main utility in manual control of the $\overline{SS}$ signals.
6. In the M68HC11 implementation, the transmit register is transparent to the shift register which necessitates the write collision error (WCOL) detection hardware. This is not implemented in this design.
7. The interrupt enable bit (SPIE) defined by the M68HC11 specifications which resides in the M68HC11 control register has been moved to the IPIER register. In the position of the SPIE bit, there is a bit to select local master loopback mode for testing.
8. An option is implemented in this FPGA design to implement FIFOs on both transmit and receive (Full Duplex only) mode.
9. M68HC11 implementation supports only byte transfer.
10. The baud rate generator is specified by Motorola to be programmable via bits in the control register; however, in this FPGA design the baud rate generator is programmable via parameters in the VHDL implementation. Thus, in this implementation run time configuration of baud rate is not possible. Furthermore, in addition to the ratios of 2, 4, 16 and 32, all integer multiples of 16 up to 2048 are allowed.

Reference Documents

The following documents contain reference information important to understanding the XPS InSystem Flash IP Core design:

- Motorola M68HC11-Rev. 4.0 Reference Manual
- *Motorola MPC8260 PowerQUICC II™ Users Manual* 4/1999 Rev. 0
- IBM CoreConnect™ *128-Bit Processor Local Bus, Architectural Specification* (v4.6).
- DS561 PLBV46_Slave_Single (latest version)
- DS570 XPS SPI (latest version)
- Spartan-3A Generation FPGA User Guide, UG331 (latest version)
- Spartan-3AN FPGA In-System Flash User Guide,UG333 (latest version)

Revision History

Date	Version	Revision
05/22/2008	1.0	Initial version
01/05/2009	1.1	Updated version
05/16/09	1.2	Updated minor version as its base core xps_spi goes through minor version update