Fast Simplex Link Channel (FSL)

DS449 (v1.1) Aug 06, 2003

Product Specification

Summary

This document provides the design specification for the Fast Simplex Link Channel V1.0. This document applies to the following bus infrastructure cores:

fsl_v20	v1.00.b

Features

The FSL_V20 module is used as the Fast Simplex Link (FSL) interconnect for Xilinx FPGA based embedded processor systems. The FSL is a fast uni-directional point to point communication channel with the following features.

- Uni-directional point to point communication
- Unshared non-arbitrated communication mechanism
- Control and Data communication support
- FIFO based communication
- Configurable data size
- 600 MHz standalone operation

The FSL bus is driven by one master and drives one slave.

FSL_V20 Design Parameters

The features that you can parameterize in the Xilinx FSL_V20 design are shown in the Table 1.

FSL Bus Width

Specifies the width in bits of the master and slave connected to this core. The default is 32 bits.

FSL FIFO Depth

Specifies the depth of the FIFO implemented by the FSL bus.

External Reset Active Level

Specifies the active level of external reset. This field corresponds to the VHDL generic C_EXT_RESET_HIGH.

FSL_V20 Signals

The I/O signals for the FSL_V20 are listed in Table 2 Note: *Currently the FSL_M_Clk and FSL_S_Clk signals that add support for asynchronous read and write is not supported. These ports should not be used*.

Table 1: **FSL_V20 Design Parameters**

Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
FSL Bus Width	C_FSL_DWIDTH	8,16,32	32	integer
FSL FIFO Depth	C_FSL_DEPTH	16	16	integer
Level of external reset	C_EXT_RESET_HIGH	0 = Low-true external reset 1 = High-true external reset	1	integer

Table 2: **LMB_V10 I/O Signals**

Signal Name	MSB:LSB	I/O	Description
FSL_Clk		I	FSL Clock
SYS_Rst		I	External System Reset
FSL_Rst		O	FSL Reset
FSL_M_Clk		I	Master FSL Clock
FSL_M_Data	0:C_FSL_DWIDTH-1	I	Master FSL Data
FSL_M_Control		I	Master FSL Control Signal
FSL_M_Write		I	Master FSL Write Signal

Table 2: **LMB_V10 I/O Signals *(Continued)***

Signal Name	MSB:LSB	I/O	Description
FSL_M_Full		O	Master FSL Full Signal
FSL_S_Clk		I	Slave FSL Clock
FSL_S_Data	0:C_FSL_DWIDTH-1	O	Slave FSL Data
FSL_S_Control		O	Slave FSL Control Signal
FSL_S_Read		I	Slave FSL Read Signal
FSL_S_Existsl		O	Slave FSL Data Exists Signal

FSL_Clk

Input clock signal to the FSL bus.

SYS_Rst

Input system reset signal to the FSL bus.

FSL_Rst

Output reset signal generated by the FSL reset logic. Any peripherals connected to the FSL bus must use this reset signal to operate the peripheral reset.

FSL_M_Clk

Non-synchronous FSL master clock signal to asynchronously control the master writes to the FSL channel

FSL_M_Data

The data bus written to the FSL FIFO

FSL_M_Control

Single bit control signal that is transmitted together with the data at each clock edge

FSL_M_Write

Input signal that controls the write enable signal of the FIFO. When set to ’1’, the values of FSL_M_Data and FSL_M_Control are pushed into the FIFO on a rising clock edge.

FSL_M_Full

Output signal from the FIFO indicating that the FIFO is full.

FSL_S_Clk

Non-synchronous FSL slave read clock to asynchronously read the FSL FIFO.

FSL_S_Data

Output bus that indicates the data available at the read end of the FIFO

FSL_S_Control

Output signal that Indicates the control bit associated with the data at the read end of the FIFO

FSL_S_Read

Input signal that controls the read acknowledge signal of the FIFO. When set to ’1’, the values of FSL_S_Data and FSL_S_Control are popped from the FIFO on a rising clock edge.

FSL_S_Exists

Output signal indicating that FIFO contains valid data.

.

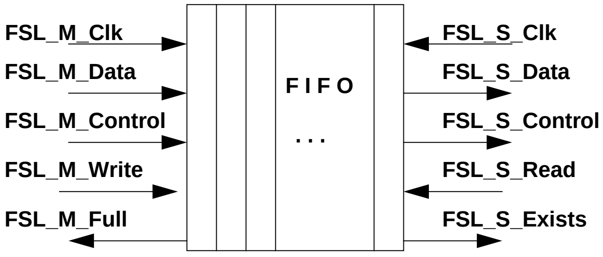

Figure 1: **FSL Interface Signals**

FSL Bus Operation

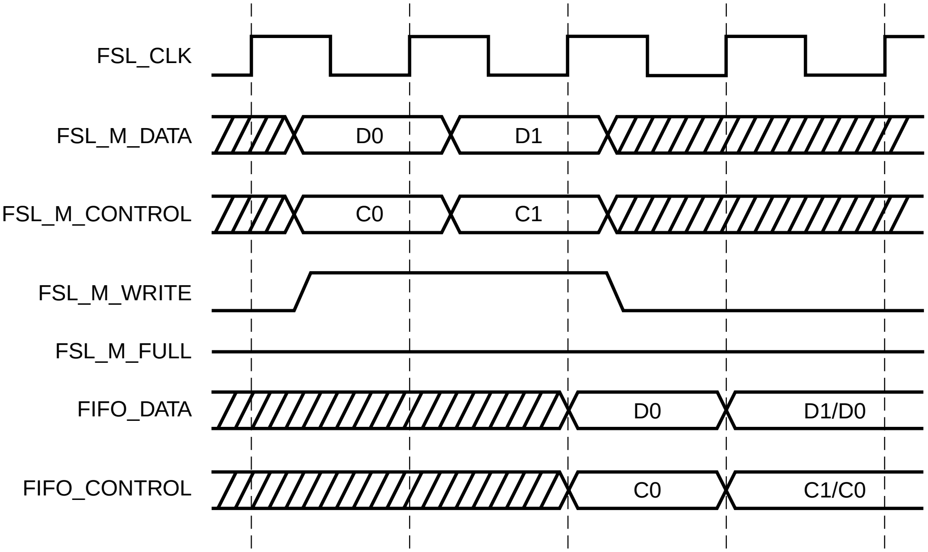

Figure 2: **FSL Back-to-Back Write Operation**

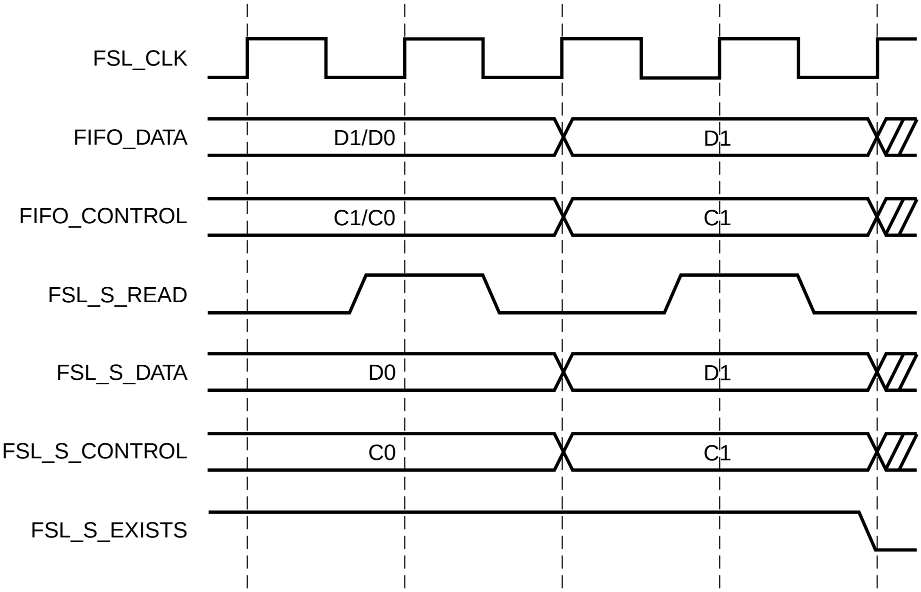

Figure 3: **FSL Back-to-Back Read Operation**

FSL Bus Write Operation

The write to the FSL bus is controlled by the FSL_M_Write signal. When FSL_M_Write is set to ’1’ at the current clock edge, the data and control signals, FSL_M_Data and FSL_M_Control, are pushed onto the FSL FIFO on the next rising clock edge. The timing diagram in Figure 2 depicts a back-to-back write operation on the FSL bus. The FIFO_Data and FIFO_Control signals denote the data and control signals of the FSL FIFO. When the FSL FIFO is full, the FSL_M_Full signal is set to ’1’.

FSL Bus Read Operation

The read end of the FIFO always contains the last unpopped data and control signals, FSL_S_Data and FSL_S_Control. When the FIFO is empty the FSL_S_Exists signal is set to ’0’ else it is set to ’1’. When the slave side peripheral of the FSL has finished reading the data, the FSL_S_Read acknowledge signal must be set to ’1’ for one clock cycle. The data and control values are then popped out of the top of the FIFO. See Figure 3 for timing diagram.

FSL Interconnect Mechanism

FSL peripherals may be created as a Master or a Slave to the FSL bus.

A peripheral connected to the master ports of the FSL bus pushes data and control signals onto the FSL. All peripherals that act as a master to the FSL bus should create a bus interface of the type MASTER for the bus standard FSL in the Microprocessor Peripheral Description (MPD) file. Further, the peripheral must form default connections to all master ports of the FSL bus and must follow the FSL bus write operation timing requirements as specified in Fig.

A peripheral connected to the slave ports of the FSL bus reads and pops data and control signals from the FSL. All peripherals that are a slave to the FSL bus should create a bus interface of the type SLAVE for the bus standard FSL in the MPD file. Further, the peripheral must form default connections to all slave port of the FSL bus and must follow the FSL bus read operation timing requirements as specified in Fig.

An example MPD of a simple peripheral having a master bus interface FSL_OUT and a slave bus interface FSL_IN is given below.

```
BEGIN my_fsl_peripheral
OPTION IPTYPE = PERIPHERAL
OPTION IMP_NETLIST = TRUE

BUS_INTERFACE BUS = FSL_IN,  BUS_STD = FSL,
BUS_TYPE = SLAVE
BUS_INTERFACE BUS = FSL_OUT, BUS_STD = FSL,
BUS_TYPE = MASTER

## Ports
PORT CLK = "", DIR = IN, SIGIS=CLK
PORT RESET = "", DIR = IN

PORT FSL_S_CLK     = FSL_S_Clk,     DIR=out,
SIGIS=CLOCK, BUS=FSL_IN,
PORT FSL_S_READ    = FSL_S_Read,    DIR=out,
BUS=FSL_IN
PORT FSL_S_DATA    = FSL_S_Data,    DIR=in,
VEC=[0:31], BUS=FSL_IN
PORT FSL_S_CONTROL = FSL_S_Control, DIR=in,
BUS=FSL_IN
PORT FSL_S_EXISTS  = FSL_S_Exists,  DIR=in,
BUS=FSL_IN

PORT FSL_M_CLK     = FSL_M_Clk,     DIR=out,
SIGIS=CLOCK, BUS=FSL_OUT
PORT FSL_M_WRITE   = FSL_M_Write,   DIR=out,
BUS=FSL_OUT
PORT FSL_M_DATA    = FSL_M_Data,    DIR=out,
VEC=[0:31], BUS=FSL_OUT
PORT FSL_M_CONTROL = FSL_M_Control, DIR=out,
BUS=FSL_OUT
PORT FSL_M_FULL    = FSL_M_Full,    DIR=in,
BUS=FSL_OUT

END
```

MicroBlaze SoftProcessor FSL Interface

The MicroBlaze32 soft processor consists of upto 8 input and 8 output FSL interfaces. The put and get instructions of MicroBlaze may be used to transfer the contents of a MicroBlaze register onto the FSL bus and vice-versa. The FSL bus configuration of MicroBlaze can be used in conjuction with any of the other bus configurations. Below is a brief overview of the FSL related instructions available in MicroBlaze. Please refer to the MicroBlaze processor reference guide for further details on the variants and uses of these instructions.

- **get, put** : Blocking Read and Blocking Write of data to the FSL. The control signal is set to ’0’.
- **nget, nput** : Non-blocking Read and Non-blocking Write of data to the FSL. The control signal is set to ’0’.
- **cget, cput** : Blocking Read and Blocking Write of data to the FSL. The control signal is set to ’1’.
- **ncget, ncput** : Non-blocking Read and Non-blocking Write of words to the FSL. The control signal is set to ’1’.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
2/11/03	1.0	Initial Xilinx release.
06/08/03	1.1	Updated document with FSL timing diagrams and requirements.