-

SPI Master IP

Hardware-Software Interface Document (HSID)

Serial Peripheral Interface (SPI) Master

For

Faster FPGA Local Bus to SPI access

July 2021 – Initial version

TABLE OF CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

Sital Technology provides IP cores and boards that include SPI targets. These SPI targets are designed to be accessed and controlled by a SPI master.

In many cases, the SPI Master controlling these targets is embedded in an ASIC or an FPGA. In many cases, these are SoC or MPSoC type FPGAs.

The major motivation driving Sital to develop this IP core was that commonly available SPI Master controllers don't provide the level of performance needed for managing a real time SPI slave such as Sital's BRM1553SPI IP core. There are two factors affecting the performance of a SPI interface. These include the SPI clock rate and the cycle-to-cycle time. While available SPI masters are able to operate well at high clock rates including up to 50 MHz, their cycle-to-cycle delay times can be unacceptably long, sometimes on the order of milliseconds.

Sital's SPI Master IP core is optimized to allow an on-chip processor on an SoC or MPSoC FPGA to be able to access the core at low-level through on-chip data buses such as AXI for Xilinx. AXI is a high-performance bus, enabling 100 MHz, 32-bit continuous burst transfers.

An important issue that sometimes arises is that many of the real time transactions for accessing the BRM1553SPI IP core consist of blocks of data, such as 32 16-bit words (64 bytes).

In SPI, if 64 bytes are transferred in 64 individual SPI cycles, transferring a single byte with each cycle, then each such cycle requires an address + command which are sometimes 4 bytes, resulting in an inefficiency of at least 5x overhead. Sital's SPI Master IP is optimized for burst mode support for working with API read and write operations for supporting block mode transfers.

An additional enhancement of Sital's SPI Master IP is to provide two similar SPI controllers operating through a common external SPI interface that can operate in a "Ping Pong" (double buffered) approach. The Host CPU may program one of the controllers to write 32 16-bit words, and before this operation is completed, also configure the second controller to read 256 words of 16 bits from the Monitor. Using this configuration, the second controller will begin its read operation immediately after the first controller finishes its write operation. This will provide far better performance than can be achieved by host software implementing these two operations sequentially.

The depth of the SPI RAM block is 4096 bytes (2048 16-bit words), enabling the user to send or receive blocks of any length up to 4096 bytes.

1.2 ABOUT THIS MANUAL

The purpose of this document is to define the software-to-hardware interface with the SPI Master IP.

It is intended for the device driver software developer.

2 Block Interface

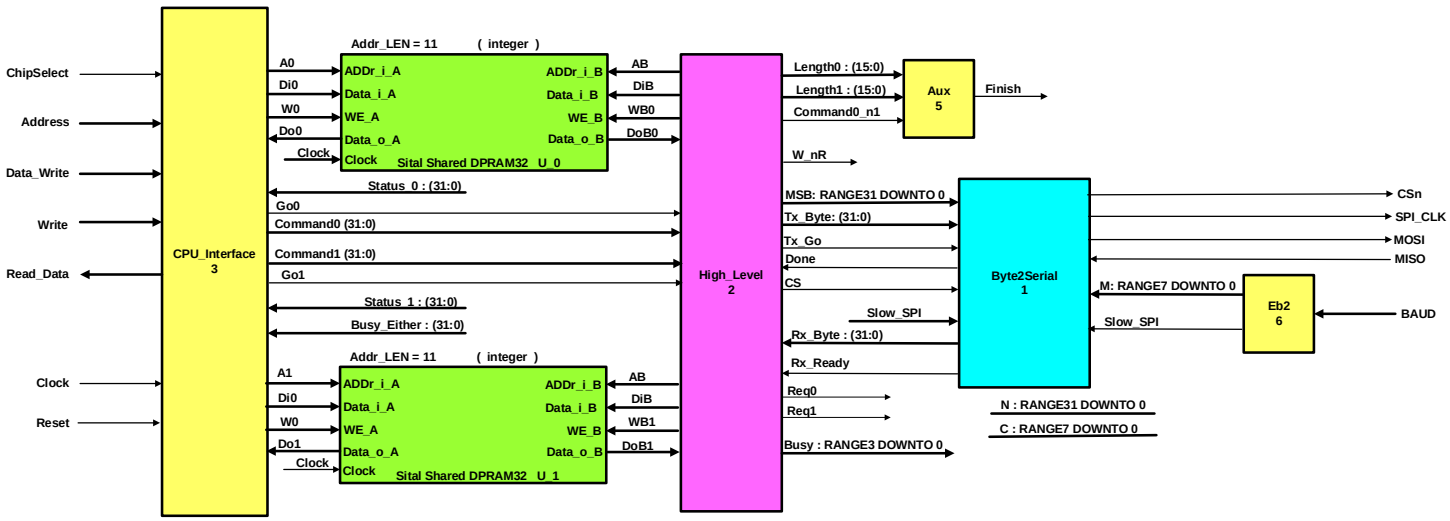

Figure 1 – Block Diagram

The CPU Interface block interfaces between the Host and the core's two memory buffers.

Each memory buffer is 32 bits wide and 1024 long, supporting 2048 x 16-bit data.

The "High_level" state machine manages which of the two buffers is active on the SPI lines, and which is waiting it's turn.

Once a new SPI cycle is initiated by the host, the High_level state machine sends 32-bit wide commands to the Bytes2serial state machine which serializes these bits to and from the SPI bus.

The SPI baud rate is programmable via the Baud input. This input is in the range of 0 to 3. When 0, the baud rate of the SPI would be ½ of the clock input. When 1, the baud would be 1/6 of the clock input. When 2, the baud would be 1/10 of the clock input and when 3, the baud would be 1/16 of the clock input.

Example: If the clock input is 100Mhz, and the "BAUD" input is 0, the SPI clock rate would be 50 Mhz.

If the clock input is 100Mhz, and the "BAUD" input is 2, the SPI clock rate would be 10 Mhz.

3 SPI MASTER CONTROLLER CONTENT

3.1 ADDRESS SPACE

The internal address space of the SPI master is presented in the table below. The address is word offset (byte offset/2) with respect to base address, which is defined inside the FPGA:

Word Offset	Name	Comments
0...2047	Data Buffer	Organized as 32-bit width. Supports both 16 and 32-bit accesses.
2K..32K-1	Spare	Reserved for future use.
32K-4 (0x7FFC)	Command	Bits 31…16: 16-bit address of WORDS(!) not of bytes.
		Bit 15: Command for this cycle. '1' for write, '0' for read.
		Bits 10...0: 11 bits that define the block length in words for the next cycle. Value = 0...2047. The actual length is 1 more than specified. That is, 0 is actually 1…and 2047 is actually 2048.
32K…32K+4K-1	Data Buffer2	Same as above but for buffer 2
64K-4 (0xFFFC)	Command2	Same as above but for buffer 2

The SPI cycle is triggered when a command is written to 0x7FFC or 0xFFFC.

The user can operate in single buffer mode, working only with either controller, or working with both.

3.2 COMMAND DETAILS

The command is 32 bits and defined as follows:

Bits 31..16	Bit 15	Bit 14	Bits 13..11	Bits 10..0
Word Address of block	"0" – Register access "1" – Memory access	"0" – Read "1" – Write	"0000" for future expansion.	Length = # of words in block minus 1. 0 means 1 word 2047 means 2048 words (0x800)…

3.3 STATUS DETAILS

Reading the status words from the same address as the command word. The reply is as follows:

Bits 31..16	Bit 15	Bits 14..11	Bits 10..0
Version date code: 0xDDMY	'1' if any channel is busy	"0000" for future expansion	0 – means all words transferred, IP is IDLE. N – number of words that still remain to be sent.

4 PINOUT OF SPI IP

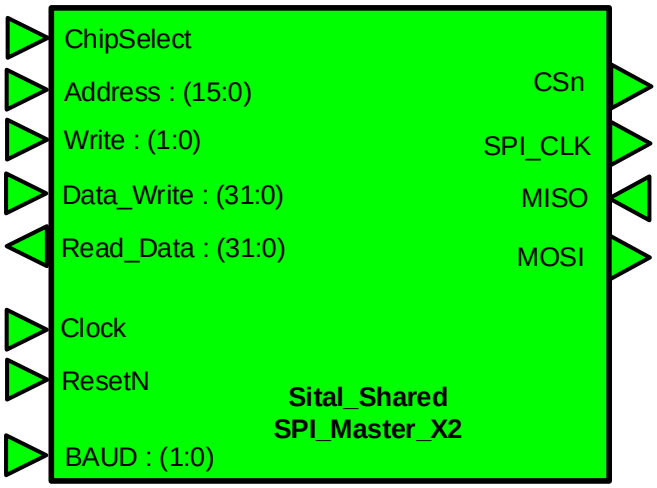

Figure 2 – Top Module Diagram

Name		Comments
ChipSelect	Input	'1' – enables writing to IP.
Address	Input	16-bit address, of words.
Data_Write	Input	32-bit data input. Arranged as LSB in 15..0, and MSB in 31..16.
Write	Input	2 bits. Write(0) when '1' enables write Data_write(15..0) bits. Write(1) when '1' enables write Data_write(31..16) bits.
Read_Data	Output	32-bit data output. Arranged as LSB in 15..0, and MSB in 31..16.
Clock	Input	The IP toggles on this clock. This clock also determine the SPI clock.
ResetN	Input	Active low reset signal, should be synchronized with clock.
Baud	Input	2 bits. Define the SPI clock rate based on: 0 – SPI clock = Clock/2. 1 – SPI clock = Clock/6. 2 – SPI clock = Clock/10. 3 – SPI clock = Clock/16.
CSn	Output	SPI Enable signal. When '1' resets the cycle. When '0' allows communication.
SPI_CLK	Output	SPI Clock. Data is sampled on rising edge, and output on falling edge.
MOSI	Output	Master Output Slave Input – Data transmitted to target.
MISO	Input	Master Input Slave Output – Data received from target.

Figure 3 – Detailed Pinout

Sital Technology USA
Bob McBride; 949-212-5922, robert.mcbride@sitaltech.com
Mike Glass; 716-626-7059; mike.glass@sitaltech.com
Email: info@sitaltech.com
Web: www.sitaltech.com

The information provided in this User's Guide is believed to be accurate; however, no responsibility is assumed by Sital Technology for its use, and no license or rights are granted by implication or otherwise in connection therewith. Specifications are subject to change without notice.

Please visit our Web site at www.sitaltech.com for the latest information.