A System Bus – SPI Bridge Architecture for Wireless Radio

Mohamad Yusri Mohamad Yusof, Devi Prasad, Smruti Santosh Palai
Microelectronics Department, MIMOS Berhad.
Technology Park Malaysia, Kuala Lumpur, Malaysia-57000.
myusri.myusof@mimos.my, devi.prasad@mimos.my, santosh.palai@mimos.my http://www.mimos.my

Abstract: - Validation of a wireless radio on a standard prototyping platform is a major part of the development process. Typically the hardware is predefined in these platforms, which sometime requires bridge designs for bus protocol translation. This paper presents a system bus-to–SPI bridge design and its implementation.

Key-Words: - FPGA, SPI, Prototyping, System Bus.

1 Introduction
Prototyping of is an essential part of the development process of a wireless radio. Implementation on a practical test bed not only validates the theoretical and simulation work but also verifies the design performance in real-time. Most of these platforms comprise a DSP processor, FPGA for PHY and MAC layer processing in a general purpose processor (GPP) and an analog frontend comprises A/D and D/A converters and RF module [1][2].

One of the common problems in prototyping the radio is interfacing the baseband module with the RF-frontend. Many of these problems can be addressed easily during the final product release. However the problem still remains during the prototyping of the radio.

The proposed interface method is based on the standard 4 wire SPI interface to the baseband processor [3][4]. In this paper we propose a bridge design to interface SPI compliant RF module with baseband processor in a shared bus interface system. This design has been verified on a prototyping platform (SPTWIMAXCX1E Multi-Standard Baseband AMC Channel Card) from Freescale Semiconductor [5].

2 SYSTEM ARCHITECTURE
The SPTWIMAXCX1E development board is a channel card designed for wireless broadband access applications. Among other things, it contains one Freescale MPC8555E processor, running at 833MHz, two StarCore MSC8126 multi-core DSPs (8 DSP cores in total), running at 500MHz, and one Stratix II Altera FPGA. The FPGA is connected to the each of the two DSPs, DSP A and DSP B, with separate system bus interface.

The targeted application was a prototype of a broadband radio on the above platform. The architecture of the system is shown in Fig.1 where the Receiver (RX) path of the radio has been depicted. The system consists of IQ sample collection from A/D convertor and the forwarding of the samples to the DSP without significant latency for further processing. A portion of the PHY processing is done by FPGA. A RF transceiver module has been used for up/down conversion.

The RF module down converts the received signal in analog IQ form. Two A/D converters (one each for I and Q) digitize them and send them to the FPGA buffer for temporary storage. Subsequently, baseband processor (DSP) reads the stored IQ samples through System Bus Interface from FPGA.

Fig.1 RX path of an implemented radio (Partial.)

2.1 BASEBAND PROCESSOR
The baseband processing algorithms are implemented on a MSC8126 DSP. The 64-bit System Bus running at 166 MHz connected to FPGA. It is a bidirectional shared bus. This bus is used to transfer the IQ samples (continuously), command and control signals for RF and for data transactions for some Physical layer processing inside the FPGA.

2.2 FPGA
The Altera Stratix II FPGA is used for system Interfacing and for some lower physical layer processing [6]. The System Bus Interface from Freescale is a memory map decoder which enables the DSP to read or
write into the selected device or memory [7]. This FPGA also has a FIFO to store IQ samples temporarily before they get transferred to the DSP. This is to facilitate the DSP to use the bus not only for the transfer of IQ samples but also for other purposes.

2.3 SYSTEM BUS INTERFACE

The system bus communication between MSC8126 and the FPGA is supported by either a User-Programmable Machine (UPM) or a General-Purpose Chip-Select Machine (GPCM) controller [8]. Although UPM allows for arbitrary waveform patterns to be defined, for example, to devices supporting burst access, GPCM was chosen in our implementation because it is simpler and it meets our bandwidth requirement. In addition, a 32-bit data bus width was selected.

2.3 RF MODULE

This RF module comprises of a Zero-I/F transceiver chip. It requires 32 internal registers to be configured by a 4 wire SPI interface. These registers are responsible for changing the operating frequency, gain and other RF parameters. One of the major parameters is the LNA gain register which needs to be updated through AGC loop to maintain the signal level within a valid range. This requires frequent updating of these register based on the values calculated by the DSP.

As shown in Fig.1, DSP is not directly connected to the RF module through the 4 wire SPI interface. The only option is to access these registers from DSP over the System Bus through the FPGA. Hence a System Bus-SPI bridge module has been developed to address this interface issue.

3 SYSTEM BUS-SPI BRIDGE

A system bus module has been developed using Verilog HDL. This module gives access to 5 memory locations to DSP namely: rf_config_wr (It is a 16 bit FIFO onto which DSP writes the desired register address(6bits), parameter (10bits) to be written to the RF module through SPI), rf_start_spi (to start the SPI read/ write process), rf_config_readaddr_wr (It is 6 bits address FIFO onto which DSP writes the desired address to read from RF module ), rf_config_read_addr_data (It is a 16 bit FIFO from which DSP can read the values of the registers requested by rf_config_readaddr_wr), RF_status_SPI (It’s a 2 bit register to show the status of the current SPI operation).

Fig.2. Write SPI.

Fig.3. Read SPI.

This module also connects the RF module through 4 wires namely, SCLK, CSB, DIN, DOUT. Typical read and write operation shown in Fig 2 and Fig 3 respectively. Register data is shifted in MSB first and is framed by CSB. When CSB is low, the SCLK is active, and input data is shifted with the rising edge of the SCLK. Output data is used for read access and is shifted out to the registers in the falling edges of the SCLK. Verilog HDL is used in designing the module and the top level test bench to verify the functionality of the design.

As shown in Fig.1., the SPI BRIDGE contains 32 word depth and 16 bit wide Write-FIFO, to store the address (6 msbs) and configuration parameter (10lsbs) up to 32 registers. It has a 32 word depth 6 bit wide FIFO to store the address of the register for read back. It also has a Read FIFO identical to Write-FIFO to store the read back values (to be read by DSP later).

A FSM controller detects the DSP command to start the SPI operation. It checks for the empty status of the Read address and Write FIFO to distinguish between the read and write operation. For write operation, first it sets the SPI-Status register value to busy. It then reads one 16 bit value from the write FIFO into a register. Next the register is shifted 1 bit at a time to the DIN pin for the RF module in MSB first order. After completion of all the shift operations it again checks for the FIFO empty status. If it is not empty then repeats the above operation. Otherwise it changes the state to DONE by changing SPI-Status register value to “Write done”. Similarly for read operation, first it sets the Status register value to busy. It then reads 6 bit address from the FIFO and shifts the register to the DIN pin of the RF module. It then presents the tri-stated value for next 10 clocks. The serial bits coming out of the DOUT pin of the RF module are stored in a 10 bit register. Finally this value is stored in the Read FIFO. This process continues repeatedly till the read address FIFO gets empty. Once the read address FIFO gets empty, the state changes to DONE by changing SPI-Status register value to “Read done”.

A top level test- model was developed to simulate the DSP action to verify the functionality as shown in Fig.4. After successful functional verification this design was then implemented on Stratix-II FPGA. Quartus-II8.0 from ALTERA was used for synthesis and implementation/fitting. The design was successfully
tested on the SPTWIMXCC1E platform and it was verified that the RF module could be configured at SCLK running at 25MHz without any timing violations.

4 Conclusion

This paper explains a System bus- SPI bridge module that has been developed and implemented in an ALTERA Stratix-II FPGA to overcome difficulties in interfacing the System Bus to a SPI complaint RF module.

References: