International journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available onlne http://www.ijoer.in

Vol.1., Issue.2., 2013

RESEARCH ARTICLE

ISSN: 2321-7758

CUSTOMIZABLE UART TO SPI IP CORE

ABDUL IKRAM[1], T.MUNI KUMAR[2]

[1]Student of VLSI and ES Design, [2]Assistant Professor, ECE Department, ASIST, Paritala, Andhra Pradesh, Affiliated to JNTU-Kakinada

Article Received: 12/09/2013

Article Revised on: 16/09/2013

Article Accepted on:17/09/2013

ABDUL IKRAM

Author for Correspondence
Email:*ikram1984@gmail.com*

ABSTRACT: In this paper we proposed an algorithm for software implementation of an 'UART to SPI IP core'. It can be used as a module in implementation of SoC based design. The parameters of UART and SPI bus can be customized as per the requirement of user. The core is implemented in verilog HDL and verified the design using Xilinx ISE design suite 13.2.The area occupied and power consumed are evaluated. The simulation as well as test results are satisfactory.

KEYWORDS— UART, SPI, Verilog, IP Core

INTRODUCTION

The UART protocol has been used frequently for short distance off-board data transfer, software debugging etc. The ever increasing complexity of the processing system to design, the modular approach has become a must. With those goals in our mind, we have designed a UART-to SPI core which can be used as a module in building bigger systems incorporating the UART protocol as their serial communication protocol and SPI as Serial bus for data transfer.

Various parameters of this module can be customized by the user as suitable for him/her. The configurable parameters of our module are: i) System clock frequency (default-50Meg) ii) Baud-rate (any one is generally used, default-9600) iii) No. of data bits (7/8, default-8) iv) No. of stop bits(1.0/1.5/2.0, default-1) v)Over-sampling rate (8/16/24/32 etc. default16) vi) No. of buffers needed to cope up with the speed difference between the system using the UART and the rate at which data are coming (default-8)vii)SPI Interface

clock frequency(10kHz-100MHz,default 50MHz) viii)SPI Slave Interface(any device, default Flash ROM is Chosen for verification.

UART to SPI Architecture

The UART to SPI IP Core include a simple command parser that can be used to access an internal bus of SPI via a UART interface. This IP can be used understand the SPI transaction protocol. The internal bus is designed with address bus of 16 bits and data bus of 8 bits. The core implements a very basic UART transmit & receive blocks which share a common baud rate generator and a command parser. The parser supports text mode of command parsing. Text mode commands are designed to be used with hyper terminal software and enable easy access to the internal bus. The architecture includes a UART Core message handler and a SPI Core. The following figure depicts a block diagram of the core.

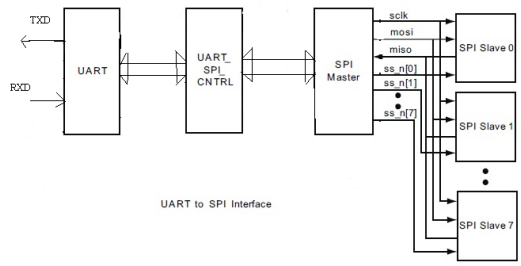

Fig1. block diagram of the UART to SPI core.

Start	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Parity (E/O/No)	Stop 1 or 2

Fig2.UART Frame format

A. UART Core

It translates data between parallel and serial forms. This core includes 3 parts. i) RXFSM**:** This block monitor the UART receives port (uart_rxd) and decode it into 8 bit data format. ii)

TXFSM**:** This block translate 8 Bit Data into serial UART bit frame format and drive it into UART Transmit Port (uart_txd).

iii) CLK-GEN: This block includes a clock divider circuit to generate required baud-clock required to sample/drive the UART interface signal.

16x Baud clock formula is = System-Clock (in Hz) / (2 + Configured Baud Register value).

B. UART_SPI CNTRL: This block includes a state machine and decodes i)wm <addr> <data> <newline> command into register write towards SPI core with <Addr> <Data> ii)rm <addr> <newline> command into a register read access towards SPI core and transmit back the read data received from SPI interface.

C. Serial Peripheral Interface(SPI): Serial to Peripheral Interface (SPI) is a hardware/firmware communications protocol developed by Motorola and later adopted by others in the industry. Microwire of National Semiconductor is same as SPI. Sometimes SPI is also called a "four wire" serial bus.

The Serial Peripheral Interface or SPI-bus is a simple 4-wire serial communications interface used by many microprocessor/microcontroller peripheral chips that enables the controllers and peripheral devices to communicate each other. Even though it is developed primarily for the communication between host processor and peripherals, a connection of two processors via SPI is just as well possible.

The SPI bus, which operates at full duplex (means, signals carrying data can go in both directions simultaneously), is a synchronous type data link setup With a Master / Slave interface and can support up to 1 mega baud or 10Mbps of speed.

The peripherals can be a Real Time Clocks, converters like ADC and DAC, memory modules like EEPROM and FLASH, sensors like temperature sensors and pressure sensors, or some other devices like signal-mixer, potentiometer, LCD controller, UART, CAN controller, USB controller and amplifier.

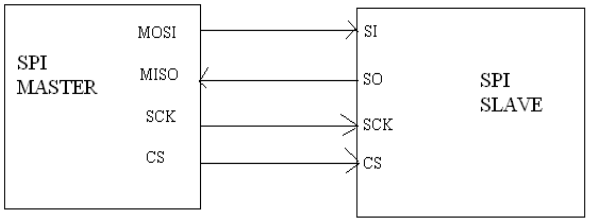

Fig 3.SPI Master-Slave Configuration

VERIFICATION

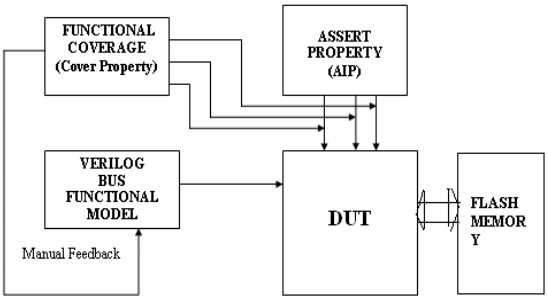

FIG 4.Verification Flow

UART Bus Functional Model (BFM): This block manages the UART protocol format translation from 8bit data to serial format and vice-versa.

DUT: It stands for Device Under Test. Which is 'UART to SPI Core'.

Flash Memory: It act a SPI Slave device, which is connected through SPI Bus.

Assert Property: The assert statement is used to enforce a property as a checker. When the property for the assert statement is evaluated to be true, the pass statements of the action block are executed. Otherwise, the fail statements of the action block are executed. Assert Property has Tasks.2 types of task in this project are,

SPI Task: This module includes various tasks to configure the on-chip SPI module.

UART Task: This module includes various tasks to configure the on-chip UART module.

RESULTS

A. *SYNTHESIS BLOCK DIAGRAMS:*

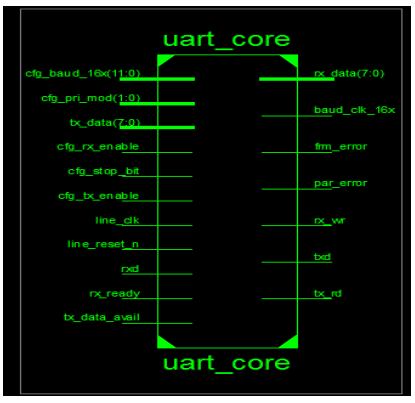

Fig 5.UART Core top module

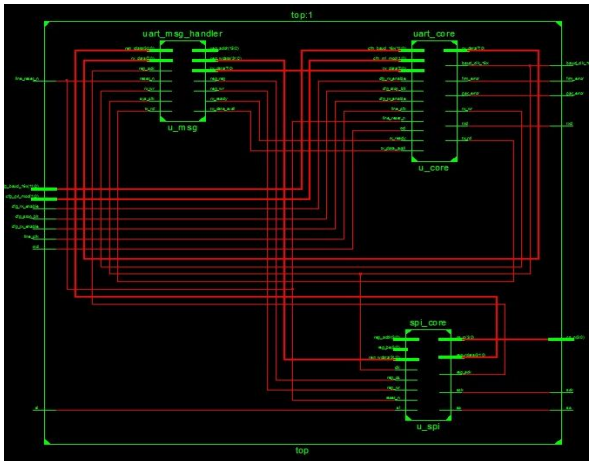

Fig 6.UART to SPI Core Top level module

B. *SIMULATION DIAGRAMS:*

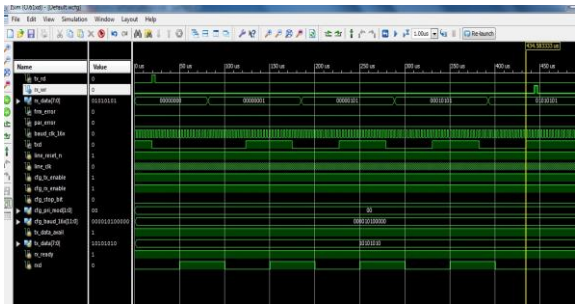

Fig 7.Uart Receiver (without parity)

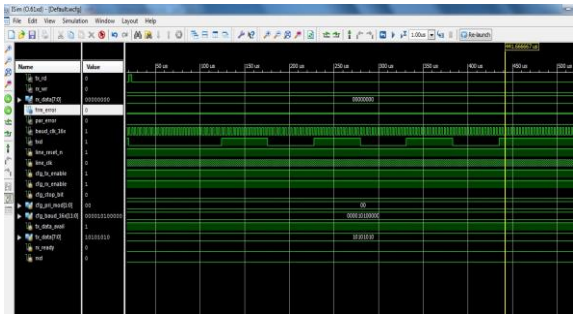

Fig 8.Uart Transmitter(without Parity)

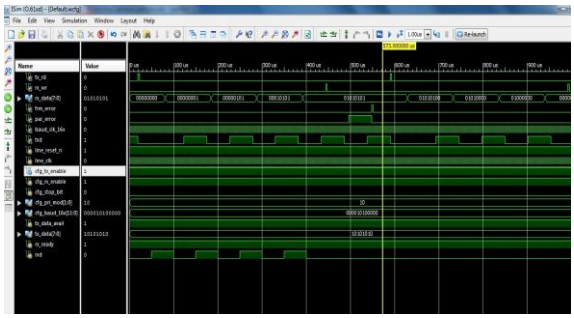

Fig 9.Uart with even Parity

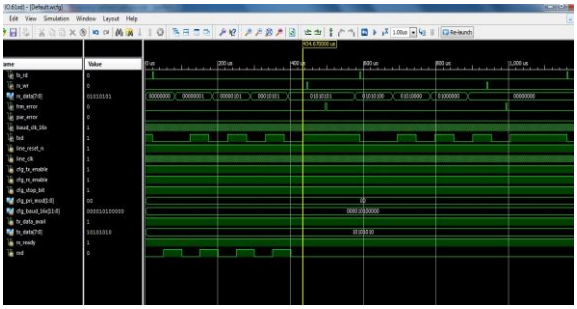

Fig 10.Uart with 2 stop bits

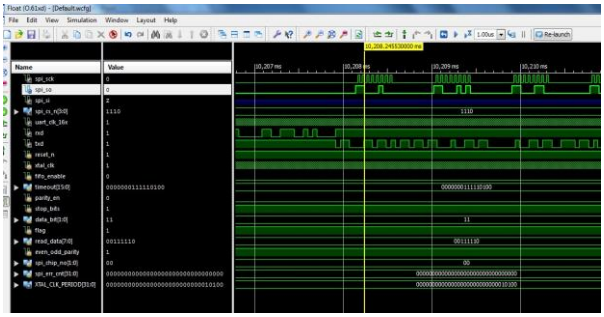

Fig 11.Uart to SPI Memory Write Operation

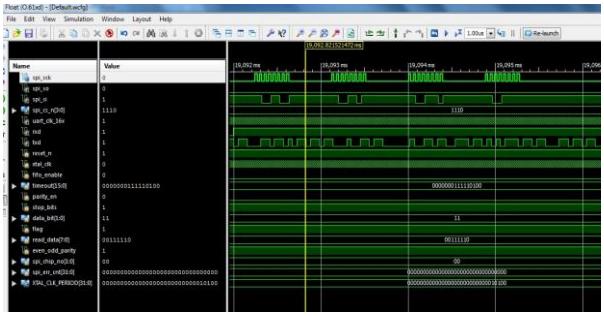

Fig 12.Uart to SPI Memory Read Operation

CONCLUSION

We have designed our UART to SPI IP Core which is operating fine and can be customized to make it free from overrun error with the capability provided and so can be made available as IP which can be used in building larger systems like SoC.

REFERENCES

[1]. B.Roy,Platform-Independent Customizable UART Soft-Core ,Intelligent Systems, Modelling and Simulation (ISMS), IEEE Third International Conference 2012.p 692-694.

[2]. Dinesh Annayya,UART to SPI Specification available :http://www.opencores.org.

[3]. MICROSEMI Application notes:AC 327 UART to SPI Interface Design example available: *www.actel.com/documents/UART_to_SPI_AN.pdf*

[4]. Verilog HDL by Samir Palnitkar Publisher: Prentice Hall PTR (January 15, 1996)

[5]. Parag k.Lala Introduction to Logic Circuit Testing Morgan and Claypool Publishers 2009.

[6]. Xilinx Synthesis Overview available at *www.**xilinx**.com/support/.../**xilinx**11/**ise**_c_using_xst_for_**synthesis**.htm*

[7]. ISIM in-depth available : *www.**xilinx**.com/itp/**xilinx**10/isehelp/ism_p_running_**simulation**_ise.htm.*

[8]. Blessington, T.P; Murthy, B.B. ; Ganesh, G.V. ; Prasad, T.S.R. Devices, Circuits and Systems (ICDCS), 2012 International Conference.p:673-677

[9]. http://opencores.org/usercontent,doc,1359617335

[10]. http://opencores.org/project,uart2spi

[11]. http://www.eeherald.com/section/design-guide/esmod12.html