International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2396-2402 ISSN: 2249-6645

High speed customized serial protocol for IP integration on FPGA based SOC applications

[1]R. Renukanath, [2]D. R. Srinivas, M.S. (Ph.D)
[1](ECE Department, GPR College of Engineering, Kurnool, Indian)
[2](ECE Department, Associate Professor of GPR College of Engineering, Kurnool, Indian)

ABSTRACT: *The serial communication is very commonly used communication protocol between various peripherals and processor. The current trend is all high speed buses are built with serial communication interface. As the key processing equipment of comprehensive task processing system, Mission management computer needs to crosslink with various equipments and the types of communication interface are different, a serial communication interface based on FPGA (Field Programmable Gate Array) has been designed in this project used for data communication with other equipment . It guarantees the realization of the serial communication function under the condition of without any increasing in hardware resources.*

*The Xilinx **Micro Blaze** soft processor and PowerPC hard processor are widely used in FPGA based CSOC (configurable system on chip) applications. These processers do not have programmable serial links for interfacing with embedded peripherals which are mostly off chip. In this project it is proposed to implement dynamically configurable serial communication block in VHDL. The developed module shall be interfaced with **Micro Blaze** processor as a general purpose IO port. The serial interface blocks shall be implemented to handle high data rate serial links and provide parallel interface to the processor. The serial interface transceiver shall consist of PISO, SIPO shift registers, clock switching modules, counters and control logic.The Xilinx Embedded development kit (EDK) shall be used for developing the test application in C programming language. The serial interface blocks which are coded in VHDL shall be synthesized using Xilinx ISE. The Modelsim simulation software shall be used for simulation. The Spartan 3 family FPGA board shall be used for verifying the results on board.*

Key words: *Serial Communication, Parallel to Serial Conversion, FPGA.*

I. INTRODUCTION

As the key processing equipment of comprehensive task processing system, mission management computer implements comprehensive control and management of the system, data processing, information processing, data decoding and so on, it needs to crosslink with many equipments and the types of communication interface are various. Generally standard interface as RS232, RS422, RS485, ARINC429, simulation and on off, can satisfy the requirements, but there are some equipment which has special requirements.

In order to guarantee normal correspond, private communication interface design is needed. The implement of standard module guarantees the stability and the reliability of the system in largely, and reduces the design personnel repeat work and effectively improves the work efficiency. But the design requirements of mission management computer are diverse, how possible on the base of without any increasing in existing module kinds to satisfy increasingly rich design requirements is currently hardware design personnel need to consider problems. The FPGA (Field Programmable Gate Array) has the characteristics of the reconstruction, the rapidity, design flexibility and the high –density of logical resources [1]; we make full use of the programmable resources of FPGA on a great extent to module function expansion and to meet increasingly complex requirements[2]. The implement mode of private communication interface based on FPGA is presented in this paper; under the condition of without anyincreasing in original module Har dware resources, it has realized module function expansion, shortened the development cycle, and satisfied the module standardization requirements.

II. REALIZATION OF THE MODULES BLOCKS DIAGRAM

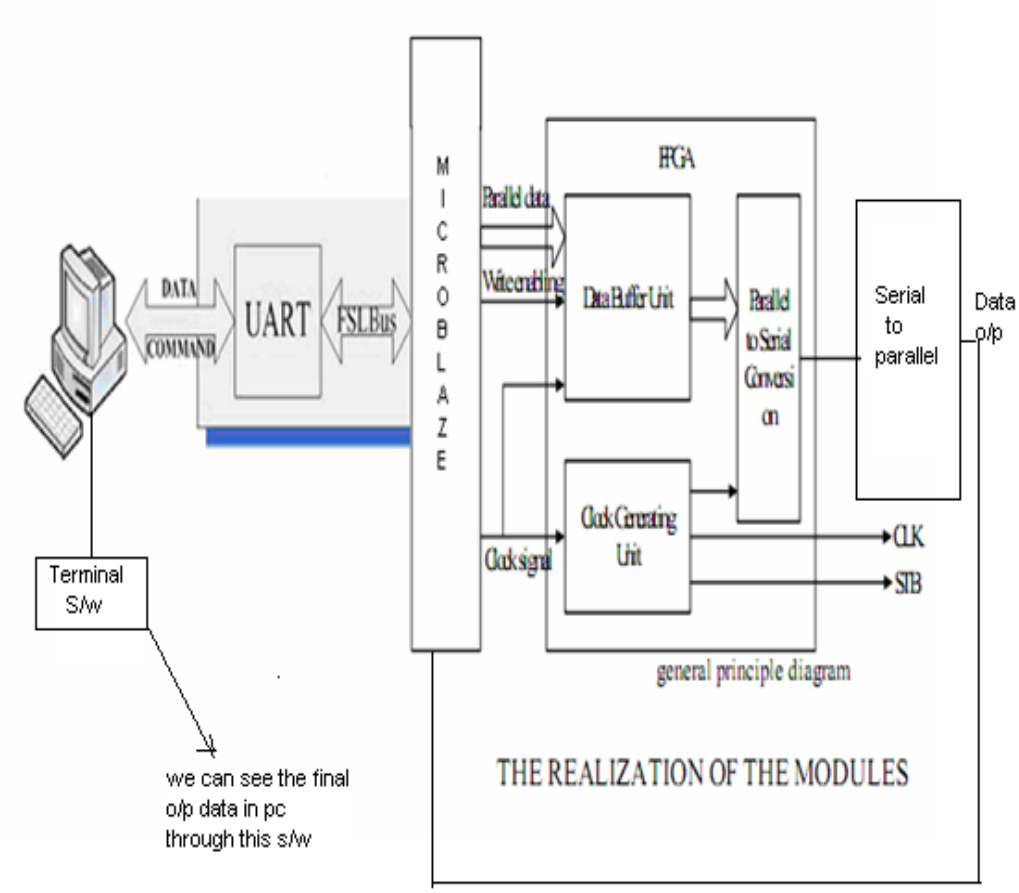


Fig.1. REALIZATION OF THE MODULES BLOCK DIAGRAM

2.1 IMPLEMENTATION OF THE TOTAL BLOCKS DIAGRAM:

The main theme of the project is to prove the serial communication is speed compared to the parallel communication. This is possible in our project by the new configuration board of fpga which is named as Spartan 3E and also the clock signal which is a 50 MHz clock. In this project whatever we used the modules and also the functions of the each and every block will be discussed in below paragraphs. First we are taking the information which is 32 bit from the pc through rs232 cable and also micro blaze is use to send the commands those are various 32 bits of inputs when we want to change the input data immediately it will automatically changed by the micro blaze commands and after receiving the 32 bits of data from the micro blaze this will passed to buffer unit for the storage purpose and after that depends on the parallel and serial clock signals this data will be send to the parallel to serial unit and here we get the output as serial and again this serial data is converted back into the paralle data by the serial to parallel conversion block and now this final parallel data will be sent back to the micro blaze port this data will appears in PC through terminal software and RS232 cable**.**

2.2 COMMUNICATIONS PROTOCOL:

Mission management computer sends control information and high-speed information to aviation administer transponder by CLK, STB and DI signal, CLK denotes data sending clock, DI denotes data(including control and state data), STB denotes data sending finish symbol . Figur2 and Figure 3 respectively shows the signal formats of CLK, STB and DI. Figure 2 shows the interval in figur1. The cycle of CLK is 13us, duty-cycle is 3/10, the 32 bits of data is sent in bytes every time, it has 32us interval between bytes, 4bytes have been sent 8us later, STB is effective; And after keeping 28us, STB is ineffective. A data sending is finished.

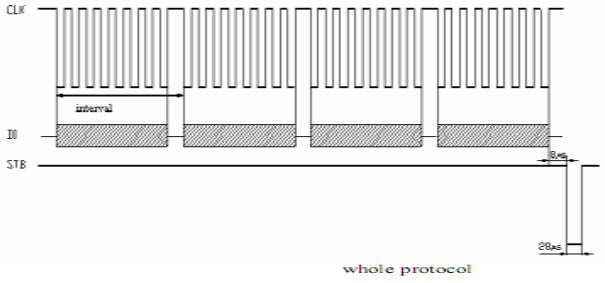


Fig.2

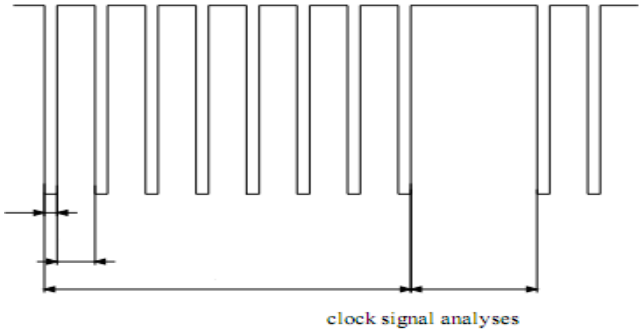


Fig.3

This interface control logic is located on data processing module, FPGA communicates with MicroBlaze processor by internal bus . Figure 1 shows the integral design diagram of this interface logic. FPGA adopts the Xilinx Company's Spartan3e series, the input signal includes 50MHz clock, data/address multiplexing bus and write enabling signals. The output signal is CLK, STB, DI which are the signal the protocol requires**(in our case clk and serial enable and final parallel data ports in our vhdl logic**). The above communication protocol will be changed in our project for the effectiveness of the output that is we are receiving the 8 bit data from the uart and at a time we will receive the 32 bits of by receiving the 4 bytes of data from the uart .then the microblaze will form the 32 bits of data and sends it to the data buffer unit for the storing purpose.

2.2.1 THE INTERFACE LOGIC MAINLY INCLUDES THREE FUNCTION MODULES:

The data buffer unit: The data buffer unit is the functional unit that is mainly responsible for the storage control of parallel data; at the same time it receives upper software order; if there are sending requirements, it sends the parallel data in buffer to parallel to serial conversion unit to process. The clock generating unit: The clock generating unit is the functional unit that is mainly responsible for generating various required clock signal according to protocol requirements; input is the interval bus clock signal which Microblaze output; frequency is 50MHz; at the same time it provides sampling basic clock for parallel to serial conversion unit and generates sending finish signal STB(serial_en signal).The parallel to serial conversion unit: The parallel to serial conversionunit is the functional unit that is mainly responsible for theconversion of parallel data is to serial data; the data is output on the CLK clock edge according to protocol requirements. After the module are electrified, firstly the data which will be sent is wrote in data buffer unit; then the module receives upper software order to send data; the data buffer unit sends the data to parallel to serial conversion unit to data transform and produces serial DI data; After waiting serial data ready, clock generating unit produces CLK signal according to protocol requirements and produces STB signal after data sending has been finished for some time , this data sending has been finishedAnd also we have added another unit in this paper that is serial to parallel unit which is used to convert the serial output of the block parallel to serial into the form of parallel. why we need this block is to just to see the output of out project in separate software called terminal and this final parallel data will be again sent back to the microblaze to see the output in terminal in pc through rs232 cable.

2.2.2 THE DATA BUFFER UNIT:

The data buffer unit is the functional unit that is responsible for the storage of the parallel data in internal bus through FPGA built-in "distributed double-port RAM" resource to achieve ; the principle diagram is shown in Figure 4. This module's input signal includes write enabling (WR_EN), read enabling(RD_EN), write clock(WR_CLK), read clock(RD_CLK), write address(WR_ADDR), read address(RD_ADDR) and write data(WR_DATA), Output port uses synchronized output RD_sDATA. After the power is reset, if WR_EN is effective, then data is wrote to WR_ADDR corresponding data unit under the action of the WR_CLK clock edge; if WR_EN is ineffective, then write port is closed. When RD_EN is effective, the data in RD_ADDR address space is read under the action of the RD_CLK clock edge as the initial data in the parallel to serial conversion unit.

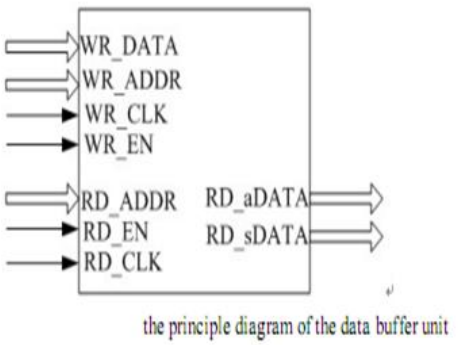


Fig.4

2.3UART BLOCK DIAGRAM:

The UART block diagram for the fpga implementation is shown in Fig. It consists of 4 blocks namely transmitter, receiver, enable generator and baud generator. The uart transmitter and receiver are deigned in the same block which is shown in below. The UART is a serial interface with a frame format of start bit of active low '0'at the beginning and 8 bit of information with a stop bit of active high'1' signal at the end. The operation of UART is controlled by Clock signal which is fed from external crystal.

Baud Generator:-Baud generation section is a clock divider ckt, FPGA board clock runs at 50MHz, but UART transfer data at predefined standards that hade to be maintained, in present system is designed for a rate of 9600/sec(i.e $50x10^6$ is scaled down for 9600). Generates a 9600 pluses for a sec, this implies the speed of UART is 9600 bits per sec. another clock with a 16 times faster is required to the receiver section so that the data will not be corrupted, baud out is given to the enable generator section.

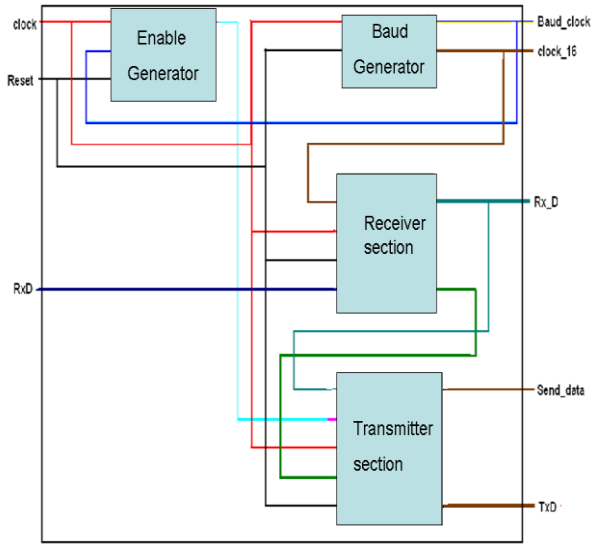


Fig.5

Enable Generator:- this section receives baud_clock signal as a enable signal and gives enable out signal to the transmitter section as a enable input signal. This signal is used to synchronize the transmitter section when ever the data is to transferred.

Transmitter: - The transmitter block is responsible for the transmission of serial data from UART. It takes 8-bit data from the receiver section (in this architecher it takes data after processing image operation block) in parallel and send data in serial form. Data inserted between start and stop bits. An optional parity bit also may be used for error detection. state machine for transmitter is shown in Fig. Transmitter stays in IDLE state unless transmit enable (tx_enable) is made as '1'.The data transmission starts with tx_enable = 1. As mandated by the protocol, a '0' is transmitted to indicate start of transmission or start bit. This is done in START state. Then data bits 0 to 7 are transmitted in states DATA0 to DATA7. If parity is enabled in configuration register, the data is attached with parity in PARITY state. Then transmitter enters STOP state and sends a '1'. This indicates the completion of transmission. Then the transmitter enters the IDLE state and waits for next data transmission.

Receiver: - UART receiver handles reception of data from RS232 port. main functions of receiver block are to convert the serial data to parallel data, check the correctness of data from parity and store the received data.

UART receiver state machine is shown in Fig. The receiver is in IDLE state by default. When the serial data pin goes low, indicating the start bit, the state machine enters DATA0 state. The data is received, one bit at a time from LSB to MSB in states DATA0 to DATA7.

If parity is enabled, the state machine checks the parity bit received against the parity obtained from received data. If the data received is fine, the data_rx (data_rx_done) bit is set to '1' and the receiver goes back to IDLE state again.

Top Module: - top module is a combination of uart and a selected image processing applications. Type of operation is selected by slider switches on the FPGA board each and every block is explained in detail below.

2.4 IMPLEMENTATION OF MICROBLAZE:

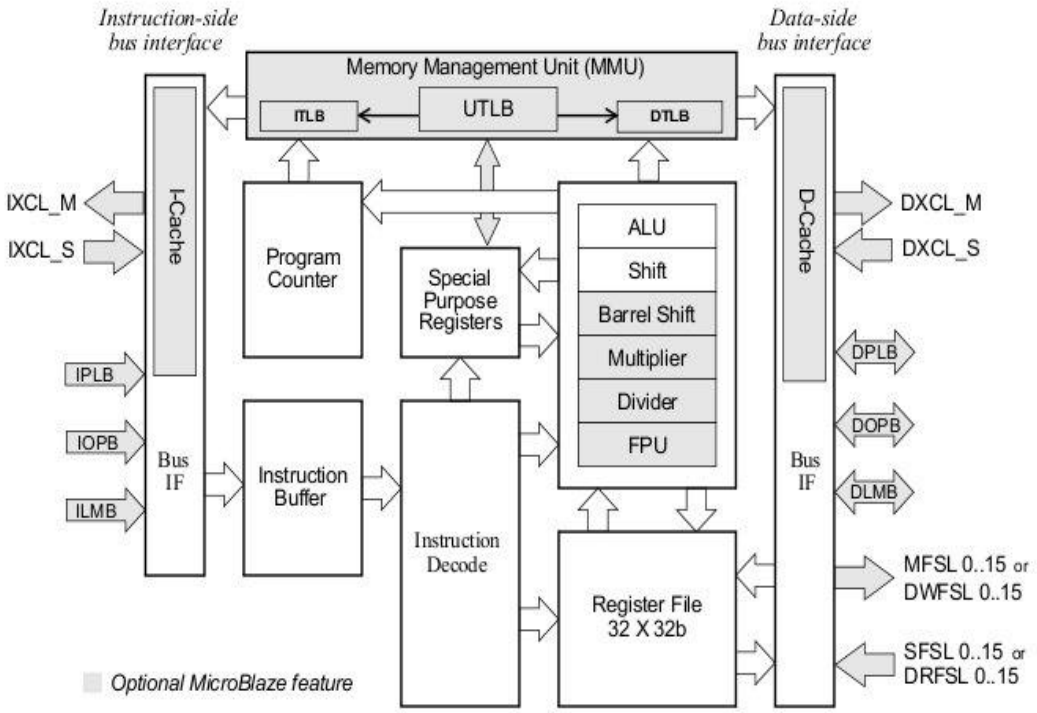


Fig.6

2.4.1.MICROBLAZE: Which is a soft core processor, the part of the FPGAwill acts as a micro blaze processor by implementing the hardware description language means by the vhdl code we are making a act the part of fpga as micro blaze. So it is called as soft core processor. No need of any external hardware circuitry. if we compared with the power pc hard core processor we won't require any separate memory block in FPGA for the micro blaze so the FPGA memory will acts as a micro blaze memory for to build a processor. we will get this soft core processor micro blaze by the ip core generation of Xilinx tool. And sdk tool of Xilinx tool.

2.4.2.POWER PC: It is hard core processor, means it is different than microblaze. In this we should require the some space in fpga hardware to build a power pc that means in if we want to use the power pc for the processor based application then that particular fpga will consist the power pc hardware in the memory block means in the memory space some memory will be accessed for power pc.So in this project we are implementing Microblaze soft core processor only. And this processor will take the information from the Pc by using UART. So from the we receive the commands in hexa,ascii format. By using these commands only we can change our parameters of modulation techniques like setting frequency ,setting azimuth angle, changing work mode, self testing and output RF signal. This function we will implementing by coding. Acoording to these commands the modulation techniques are select in the synthesized unit.

III. THE PARALLEL TO SERIAL CONVERSION UNIT

The parallel to serial conversion Unit mainly implements the conversion of parallel data to serial data. Spartan 3E provides plenty of parallel to serial conversion resources, so it can implement 2 frequency doubling to 8 frequency doubling parallel to serial conversion, and this is suitable for high-speed continuous conversion situation. This module conversion rate is quite low, and the quantity of data is small. Adopting the combination of data buffer and shift register realize the conversion of parallel data to serial data. This method is simple and suitable for low-speed and little data situation, and it can transplant to implement the control of serial interface AD/DA. The example realization code is following:

```
process (sampling clock, output control)
begin
if (output control is ineffective) then
serial output<='0';
count<="0000";
pdata_tmp<=parallel data input;
elseif( the falling edge of the sampling clock) then
if (count="1001") then
count<="0000";
else
count<=count+1;
end if;
pdata_tmp<=pdata_tmp(6 downto 0) &
pdata_tmp(7);--shift register
```

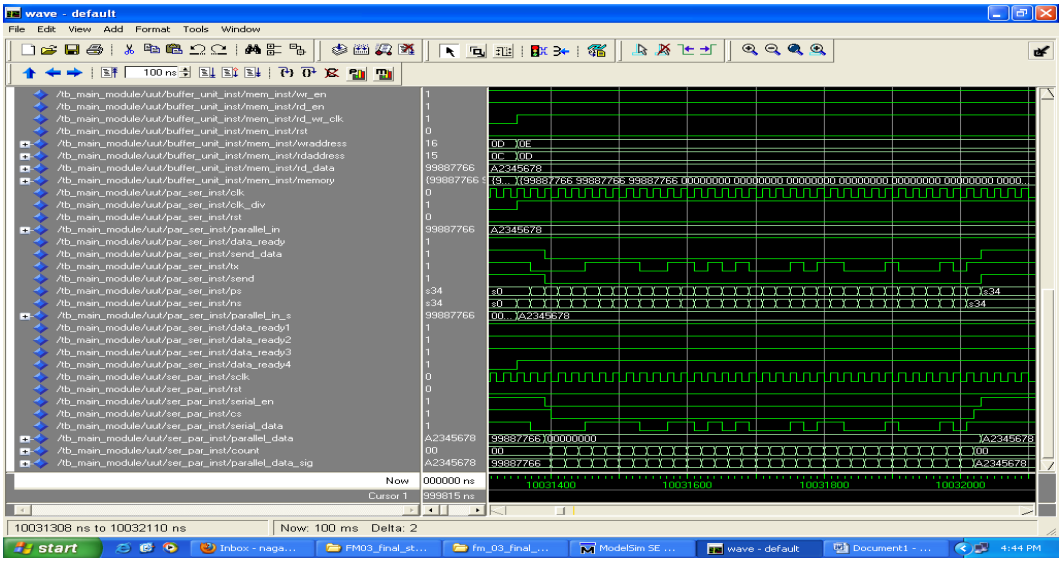
```
sdata_tmp<=pdata_tmp(7);--high bit as
serial data output
if (count="0000") then
pdata_tmp<=pdata_tmp1;-- after
8 bits are been converted, then load the new parallel data
end if;
end if;
end process;
```

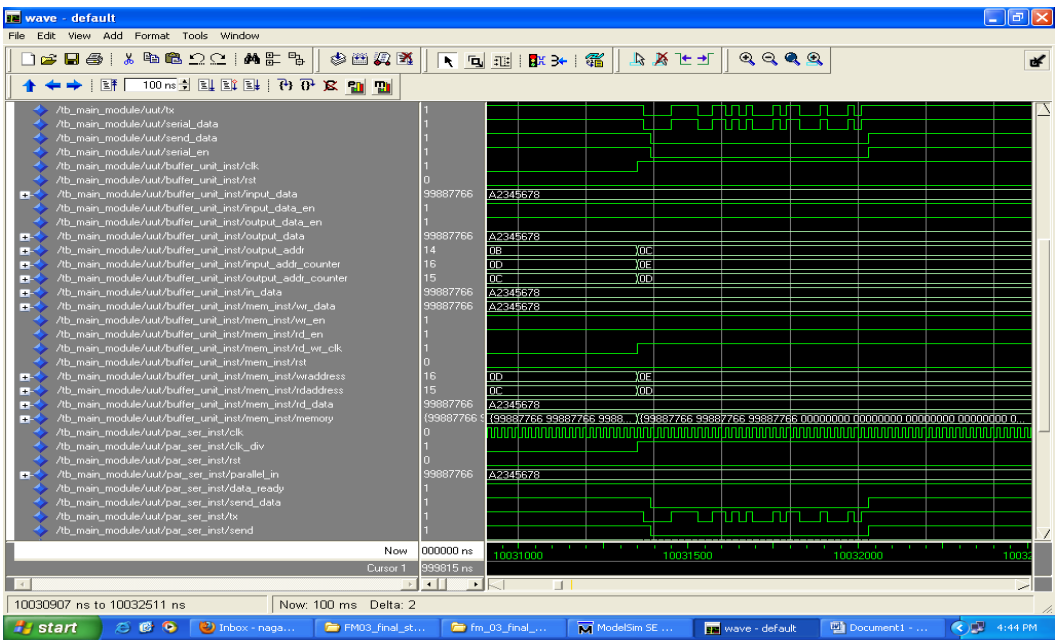
IV. THE PARALLEL TO SERIAL CONVERSION UNIT

Already we had mention in above session about the purpose of the serial to parallel unit. In the main IEEE paper they won't tell about this unit but we had added as a extra feature for the effectiveness of the final output And we know the main working function of the serial to parallel conversion unit it takes the serial information from the parallel to serial unit and it again converts into the parallel data which is sends back to the micro blaze to observe in pc through the terminal software.

V. FINAL SIMULATION RESULTS

After the design has been completed, we carry on the emulation to the function. The parallel input data is in turn"11111111", "00001111", "01010101", and "10101010".Figure 8 shows the waveform diagram. From this waveform diagram, we can see that this program has realized the extraction of effective data bits to input data, and carried on serial output according to certain baud rate. Data transmission is steady; data output satisfy the protocol requirements; the specific function and capability have been validated in system-test.





VI. APPLICATIONS

The Xilinx Microblaze soft processor and PowerPC hard processor are widely used in FPGA based CSOC (configurable system on chip) applications.

VII. ADVANTAGES

- High speed devices such as serial ADC/DACs can be interfaced to Microblaze
- Serial processing on high speed sensors is enabled with our application

VIII. CONCLUSIONS

With the enhancement of the comprehensive mission management system integration rate, the equipments with which mission management computer needs to cross-link are more and more. In this paper the design method of the private serial interface based on FPGA is shown, it has realized the new function, shortenned the development cycle, reduced manpower investment and adhered to the principle of the module standardization in the case of without increasing original module kind. This design method is worth promoting in future design.

REFERENCES

[1] Design of Serial Communication Interface Based on FPGA Peng Hui-ling; Nie Ya-lin; Dept. of Comput. & Inf. Eng., Luoyang Inst. of Sci. & Technol., Luoyang, China.
[2] Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference.
[3] LIU Wan, HE Dao-jun, TAN Ming. The Design and Application of FPGA. Beijing: Tsinghua University Press, China, June 2006.
[4] Guo Shu-tao, JING Yong-zhi. Serial Communication Based on FPGA. The transaction of Beijing Electronic Science and Technology Institute China, Vol .14, No.4, December 2006.
[5] LIU-An-jie, HOU lv, DUAN Zhi-jiao. VGA Display and Serial Communication Based on FPGA. Indurstry Control computer, China, Vol. 19, No.6, 2006.