EEE 491 Lab Assignment
Lab-DEBUG: DEBUGGER with UART

1. Technical Specifications/Requirements

A debugger shall be designed and implemented on the FPGA of Basys-3 board. The debugger shall be capable of transmitting multiple 32-bit data to MATLAB program running on a PC via USB connection. The debugger design shall be only the transmitter part of a legacy Universal Asynchronous Receiver and Transmitter (UART).

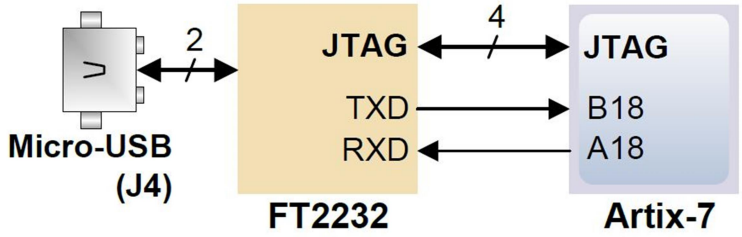
The debugger shall be capable of accessing read-only port of a memory inside the FPGA. Memory address size (N-bit) shall be set in VHDL code as a constant definition. Hence the debugger shall transmit 2^N 32-bit data to the PC via USB port.

The debugger shall initially transmit a 32-bit start data (55AACC03) in order to indicate the start of 2^N data transmission. In addition, the debugger shall transmit another 32-bit end data (AA5503CC) after the transmission of the 2^N data in order to indicate the end of the transmission. Therefore, a complete data transmission looks like:

Header for Start data				First 32-bit data transmission				Second 32-bit data transmission					Header for Stop data			
55	AA	CC	03	1st 8-bit of First 32-bit data	2nd 8-bit of First 32-bit data	3rd 8-bit of First 32-bit data	4th 8-bit of First 32-bit data	1st 8-bit of Second 32-bit data	2nd 8-bit of Second 32-bit data	3rd 8-bit of Second 32-bit data	4th 8-bit of Second 32-bit data	...	AA	55	03	CC

At the PC side, on MATLAB, received data (which shall be the data between start code and the stop code) shall be processed and each four 8-bit datum shall be converted to 32-bit integer value by concatenation. The Lab-DEBUG will become an essential tool for other lab assignments that will be used to transfer data from FPGA to MATLAB.

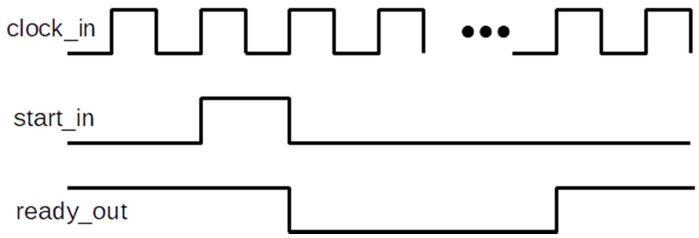
The UART in the FPGA shall interface with the FT2232HQ USB-UART Bridge on the Basys-3 board. A PC can be connected to Basys-3 via USB port, as shown below. The transfer rate of the UART shall be set to at least 115200 baud rate if supported by the FT2232HQ device.



Lab-DEBUG in FPGA shall have the following input/output (IO) signals:

- “reset_in” input control signal, which resets the FSM registers.
- ”clock_in” input clock signal of the 100MHz clock oscillator on the Basys-3 board.
- “mem_addr_out” N-bit address bus output signal that addresses a memory to read data. N is defined in a constant.
- “mem_data_in” 32-bit bus input signal which is the data output of the memory.
- “txd_out” output signal, which is serial data output of UART component in the Lab-DEBUG.

- "start_in" input control signal, which starts the transfer of data from the memory to MATLAB. This signal is active high asserted on a single clock pulse (refer to waveform below).
- "ready_out" output status signal, that indicates if Lab-DEBUG transferred all the data from memory to MATLAB. This signal is active high and asserted low just after the start signal is asserted (refer to waveform below).



Write a test-bench in VHDL for the simulation of your design. Verify by using your test-bench simulation results that your design satisfies the technical specifications.

Integrate Lab-DEBUG with a sub-system in order to transfer the RAM data of the sub-system to MATLAB.

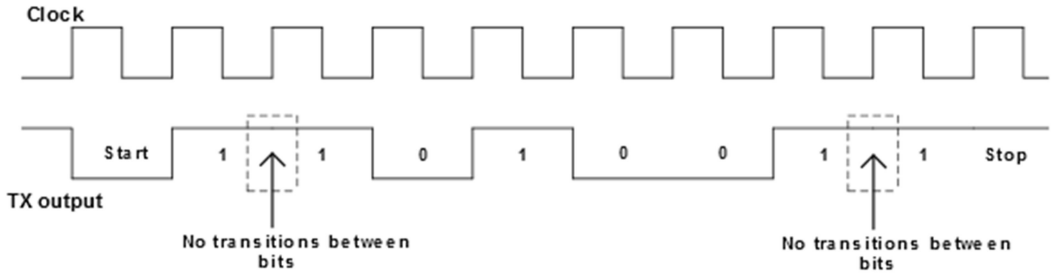
2. Demonstration

- Show that your design is implemented on FPGA.
- Present your test bench simulation results: inputs, outputs (waveforms). Start, ready, UART transmit signals, transactions must be shown clearly on the waveforms. Observe the input, output waveforms, and verify the technical specifications listed above.
- Design a demo design in which Lab-DEBUG is integrated with the Lab-CTRL and a read-only memory (ROM). Set the memory address bits and the parameter N in the debugger design to 14. Define connections between the memory and the Lab-DEBUG such that Lab-DEBUG can access the ROM. Select ROM from "Block Memory Generator" from Vivado "IP Catalog", configure it with 14-bit address and 32-bit data. Prepare a ROM content file in coefficient (COE) file format for test data and then synthesize your design.
- Configure your FPGA on Basys-3 board and then transfer the ROM content by your Lab-DEBUG to MATLAB. Verify the results on MATLAB.

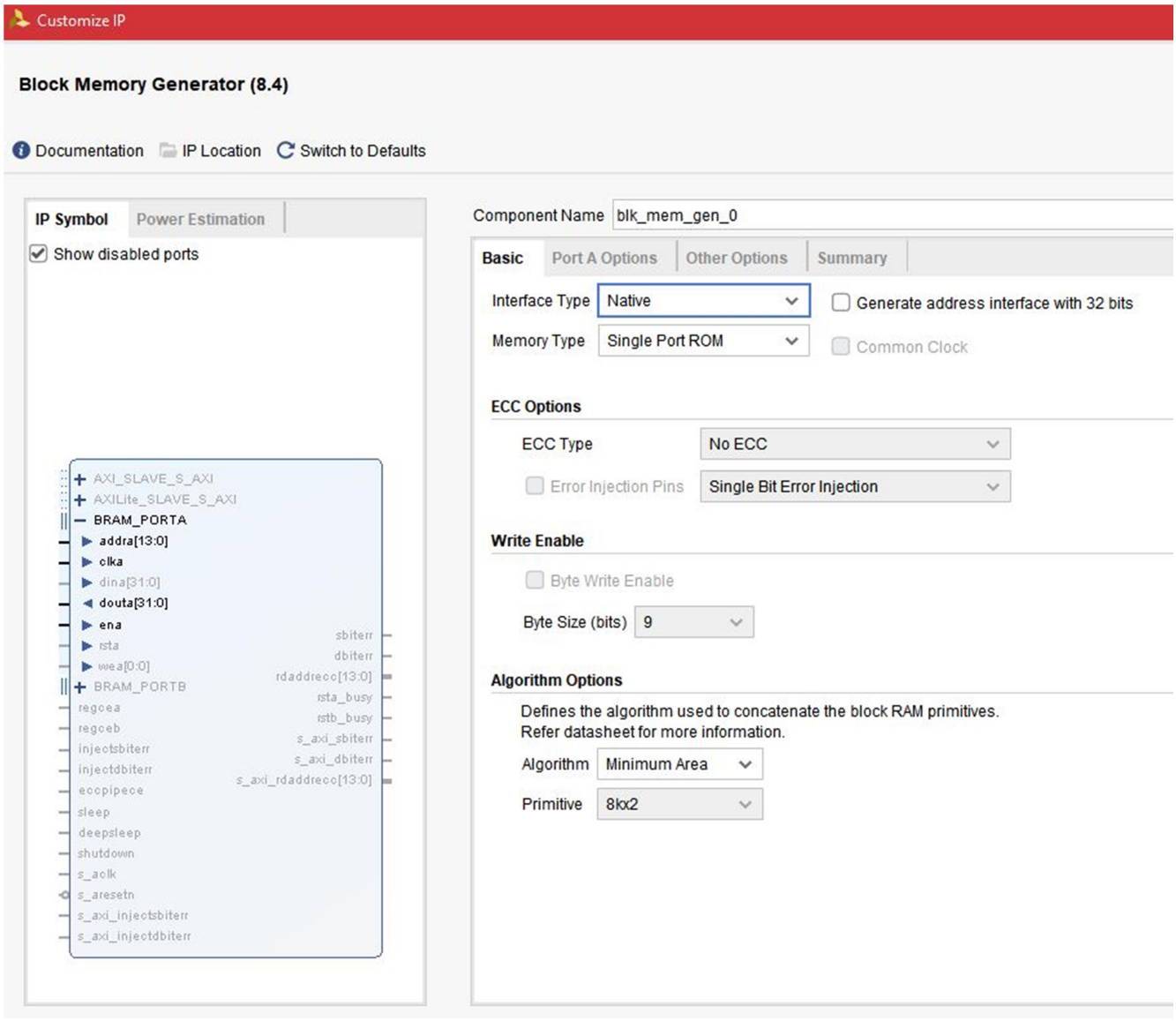
3. Guidance

- During an FPGA design, always write a test-bench code in order to simulate your design. Note that a VHDL test-bench module is not implementable i.e. you cannot describe a hardware behavior using test-bench module. Test-bench module is only used for testing a designed block (VHDL module) in simulation environment.
- UART can transmit 8-bit data in a transaction. Therefore in order to transmit a 32-bit data, a solution may be to use a Finite-State Machine (FSM) in the design that triggers the UART component four times.
- For UART to USB converter, read the related section of "Basys3 FPGA Board Reference Manual" document.
- Configure "Basys3_Master.xdc" Xilinx Design Constraints (XDC) file provided by the Basys-3 board manufacturer (Digilent). The file includes FPGA pin assignments on Basys-3 board. Uncomment the lines corresponding to used pins and rename the used ports (in each line, after "get_ports") according to the top level signal names in your project.
- Download and use the data sheet of the FT2232HQ for UART interfacing, and operating speed.
- Build your Lab-DEBUG module/block progressively. For example, start with the design and test of only 8-bit fixed data transfer with low baud rates.
- In order to establish USB connection between Basys-3 board (via FT2232HQ) and your PC, you need to install "Virtual Com Port (VCP)" driver of FTDI2232H. Depending on the operating system of your computer follow the link as described in the data sheet.

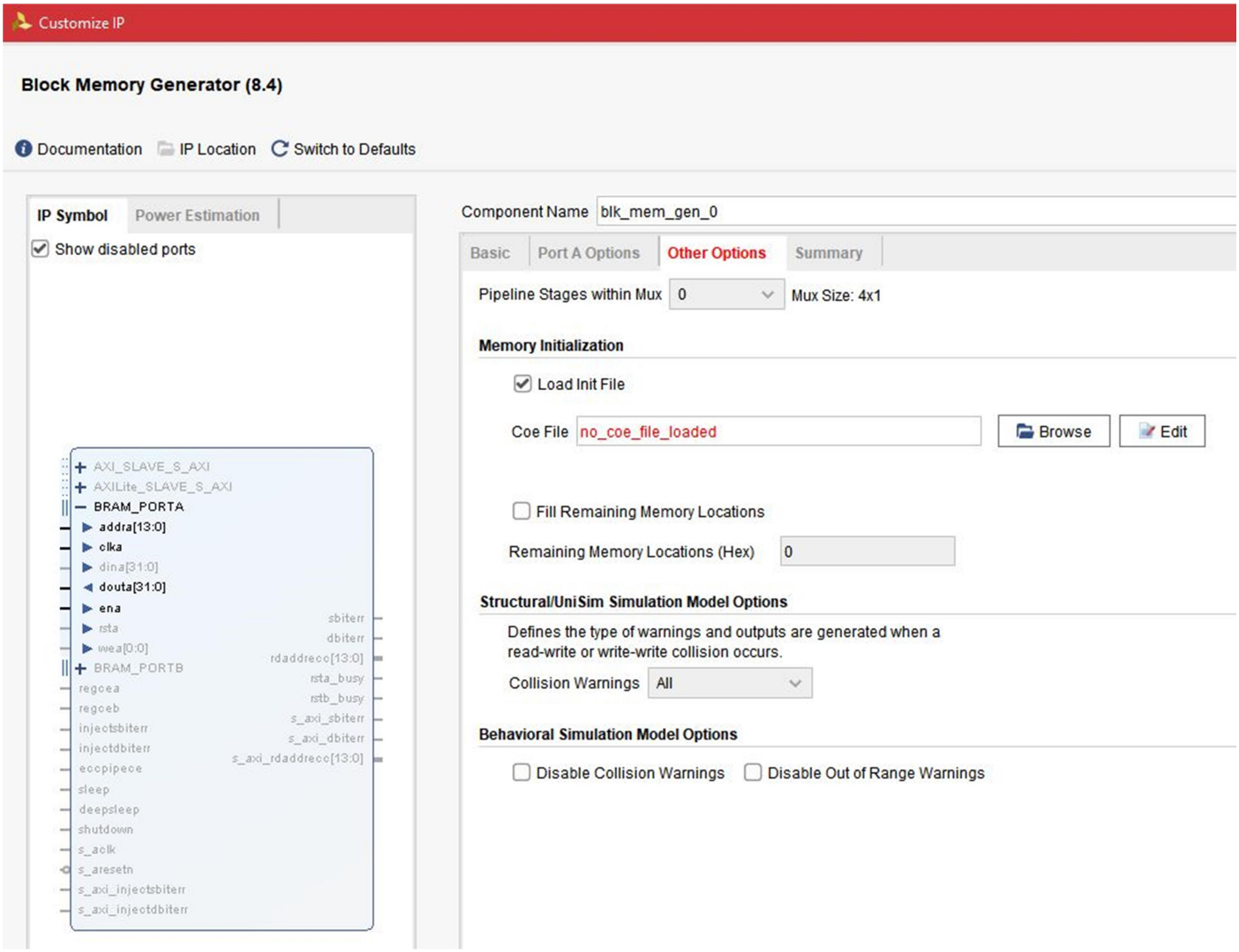
- Use “AN232B-05_BaudRates.pdf” application note in order set the baud rate you will use. Minimum baud rate must be 115200. An example for the transmission of 8-bit data with a single stop bit without parity bit. The Clock frequency is at the baudrate. The first bit to be transmitted is the least significant bit (LSB) of the data.



- When Lab-DEBUG is used for data widths (words) that are less than 32 bits, you may just insert logic-0 for unused most significant bits (MSBs) of 32-bit word.
- An example for a ROM memory configuration is shown below. Be careful for the memory block data latency at the data out (check the summary tab) which is usually one or two clock cycles.
- Use a terminal program in order to see and observe the transferred bytes on the screen.
- Write a MATLAB code on your PC in order to receive the data from USB port.



- ROM content data by specifying COE File is shown below.



An example for “COE” file content for block memory with data width of 32-bit and data depth of 4:

MEMORY_INITIALIZATION_RADIX=16;
MEMORY_INITIALIZATION_VECTOR=
FFAA01E3,
001122CD,
A0A1B298,
7123054D;