UNIVERSITY OF CALIFORNIA AT BERKELEY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Fall 2003 Project
Checkpoint 1
Writing and Reading an external SDRAM

1. Objectives

1. To configure an external SDRAM,
2. To write and read from the SDRAM,
3. To use FIFOs as buffers,
4. To estimate the minimum time the SDRAM can hold the data without refresh (retention time).

2. Methodology

The Calinx board contains 2 SDRAM chips from **MICRON** (http://www.micron.com/products/category.jsp?path=/DRAM/SDRAM part number: **MT48LC8M16A2**). A 16-bit wide data bus is connected to each of these SDRAMs. The FPGA is connected to these 2 SDRAMs with a 32 bit wide bus.

You are given the following Verilog code:

1. Synchronous FIFO (SYNC_FIFO_V3_0.v, fifo_sync.v, fifo_sync.edn, fifo_sync.xco, fifo_sync.xcp)
2. Pseudo-random number generator, (pn_gen.v)
3. Initialization and configuration information for the SDRAM (sdram_cntrl.v)
4. Verilog model of the SDRAM (mt48lc8m16a2.v)
5. Other wrapper files (FPGA_top.v, sdram_top.v)

The files you should modify are mainly, sdram_cntrl.v and sdram_top.v.

The **sdram_cntrl.v** will contain all the circuits required to generate all the control signals for the SDRAM, the FIFOs, the pseudo-random number generators (pn-generators) as well as any other control signals you will need.

The **sdram_top.v** will instantiate the sdram_cntrl module that you created and the FIFOs and pn-generators. It will also generate the error counts.

The circuit should work as follows

1. Initialize and configure the SDRAM.
2. Generate 32-bit wide words using a pn-generator.
3. Write the random numbers to a FIFO and fill it up to 16 words.
4. Start reading from the FIFO and write contents into the SDRAM. Write continuously and stop after filling up the SDRAM completely,
5. After starting the write operation for the SDRAM, wait a programmable delay, and then start

reading from the SDRAM at the same rate as the write cycle. The write cycle must still be able to continue while you are reading.
6. Write the data that has been read from the SDRAM into a second FIFO.
7. Start reading from the FIFO as soon as it is not empty.
8. At the same time start a second pn-generator and compare its results to the output of the FIFO
9. If the output of the FIFO and the output of the second pn-generator are different, increment an error counter.
10. The value of the error counter will be shown on the LEDs

3. SDRAM (All reference pages refer to the SDRAM datasheet)

Read the datasheet for the SDRAM. For the SDRAM we are using **MT48LC8M16A2**. The FPGA will send to the SDRAM the following signals:

```
RAM_DQ[31:0] : data input and output
RAM_CLK  : CLK sent to the SDRAM
RAM_CLKE  : CKE sent to the SDRAM, always 1
RAM_DQMH  : DQMH part of command instructions for read/write
RAM_DQML  : DQML part of command instructions for read/write
RAM_CS  : CS# part of command instructions for read/write
RAM_RAS  : RAS# part of command instructions for read/write
RAM_CAS  : CAS# part of command instructions for read/write
RAM_WE  : WE# part of command instructions for read/write
RAM_BA  : BA1 BA0, bank address
RAM_A   : A11-A0, Address input and configuration mode register
```

What do the A11-A0 address represent in different modes of operation? What do BA1 and BA0 represent?

3a. Initialization and configuration (pages 9-14, page 37)

The initialization sequence is as follows:

Power up -> NOP for min 100us -> precharge -> NOP -> autorefresh -> NOP ->autorefresh -> NOP -> Load Mode Register -> NOP -> active for first active operation

Please refer to the timing waveforms on page 37 and the truth table on page 12 for the signals that correspond to the commands.

The Load Mode Register command will send configuration information to the SDRAM. See page 10 for information on the configuration data. Configure the SDRAM to the following modes:

SDRAM Configuration

1. Burst length of 4 for both read and write
2. Sequential order of bursts
3. CAS latency of 2
4. Standard operation

Draw the state diagram for the FSM that you will be creating.

3b. WRITE and READ sequence (page 12, pages16-27, page 43, page 50)

In order to achieve the required data rate for the project, we will use read and write with bursts of 4 words with auto-precharge. DRAM refresh is not necessary for the project as the frame buffer contents are constantly written from video decoder. This lab will demonstrate how long we can wait before the data in

the SDRAM becomes invalid (retention time). You should validate that the DRAM retention time is sufficiently long for the correct operation of the project.

Write (page 50)

The write sequence is as follows:

Active -> NOP -> write, DQMH/L low -> NOP, DQMH/L low for 3 cycles

Refer to the write timing waveforms shown on page 50 and the truth table on page 12 of the datasheet for the signals that correspond to the commands.

The data must be present at the input to the SDRAM during the cycle when write command is given and for the next 3 clock cycles. The DQMH/L signals must be low as well when the data is sent to the SDRAM. The circuit will write into the SDRAM until the SDRAM is completely filled up (128 Mbits of data). What is the final address and what is the order of writing that you will be adopting?

Use a counter to generate the address of the SDRAM where the data is to be written. This counter should be incremented at the rate of writing.

Please note that the write timing waveforms shown on page 50 indicate that the write burst requires 9 cycles. However, because our clock cycle is relatively long, only 8 cycles are needed in our case (the final cycle is eliminated).

Read (page 43)

The read sequence is as follows:

Active -> NOP -> read, DQMH/L low -> NOP, DQMH/L low for 3 cycles

Refer to the write timing waveforms shown on page 43 and the truth table on page 12 of the datasheet for the signals that correspond to the commands.

The data will be present at the output of the SDRAM 2 clocks after the read command is sent to the SDRAM. The next 3 data will appear consecutively for the next 3 clock cycles. The SDRAM will be read from address 0 to the last possible address. The sequence of the address must follow that of write cycle.

Use a counter to generate the address of the SDRAM where the data is to be read. This counter should be incremented a the rate of reading

4. FIFO

The FIFO that you are given has width 32 and depth 64.

To use the FIFO, you will have to include the files SYNC_FIFO_V3_0.v, fifo_sync.v, fifo_sync.edn, fifo_sync.xco, fifo_sync.xcp in your project directory and use the 2 verilog files, SYNC_FIFO_V3_0.v and fifo_sync.v for simulation and synthesis. When implementing in XILINX, remember to choose the macro path to point to the location of the fifo_sync.* files.

The FIFO is generated using XILINX's COREGEN utilities. Please read the datasheet (sync_fifo.pdf).

You will instantiate 2 FIFOs (we'll call them FIFO_1 and FIFO_2 here). One will be used as a buffer for the data to be written to the SDRAM and the other will be a buffer for the data read from the SDRAM.

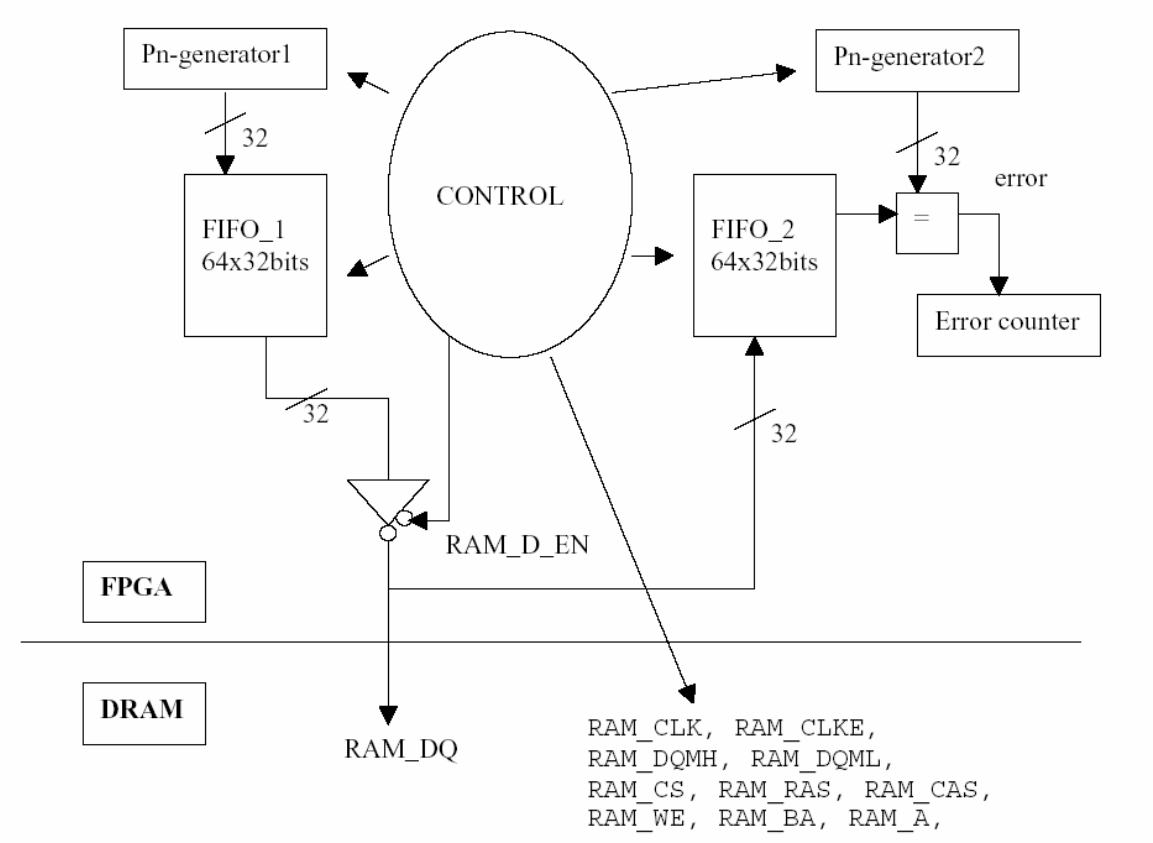
5. Pseudo-Random Number generator

This is a shift register where the output is feedback and XOR to a few of the registers in the chain of shift register. The output is a 32 bit pseudo-random number.

The number is pseudo-random because it has a probability distribution that resembles that of a random number. However, the pattern is repeating and deterministic. This means that if you have 2 identical pn-generator and they are initialized with the same value, the numbers that are generated will be identical. Also, after a certain number of cycles, the pattern is repeated.

You will instantiate 2 pn-generators, one for generating the data that will be written into the FIFO_1 and the second for comparing with the output of the FIFO_2.

6. System



6a. Important Points to Note

The SDRAM uses the same ports for data during read and write. You have to use a tristate buffer to put the output to high impedance when we want to read the data. Explain what happens if you don't do that. Locate the tri-state buffer in the Verilog codes.

6b. Control Block

Your main task is to create all the control signals.

7. Testing the retention time of the SDRAM

We will test how long the SDRAM can retain the data without refresh. You can do this is done by using the dipswitches (you can use all 16 switches) to program a delay between the first writing into the SDRAM and the first reading from the SDRAM. Pn-generator1 Pn-generator2

The data read from the SDRAM will be written into FIFO_2. When FIFO_2 is read for the first time, the pn-generator2 has to be initialized at the precise moment whereby the first data from pn-generator2 is equal to the first data that was generated by pn-generator1 and written into FIFO_1.

Each time there is an error in the comparison, the error counter will be incremented. The output of the error counter is mapped to the LED segments in hexadecimal format.

Start with a programmable delay of zero to test your circuit. There should be no error. Increase the programmable delay until you see an error.

You should use a resolution of around 10 ms in your programmable delay. You can expect the retention time at room temperature to be in the order of seconds to tens of seconds.

What is the approximate minimum retention time of the SDRAM? Is it sufficient for the needs of this project?

A Verilog model of the SDRAM is provided for your testbench (mt48lc8m16a2.v).

8. PreLab

1. Read the write-up and the datasheets and answer the questions
2. Understand the FSM given to you
3. Design your circuit.
4. Design a test bench testing your circuit.

9. Acknowledgements

Original lab by Prof. John Wawryznek and L. T. Pang
With help from – Norm Zhou and Yatish Patel
Modifications by Greg Gibeling

Name: _____ Name: _____
Lab Section (Check one)
T: AM PM W: AM PM Th: AM

Checkoffs: Checkpoint 1 Part 1

1. Answers to all questions. Description of FSM and design.
What is the minimum retention time needed for this project? _____

2. Working simulation of SDRAM read and write tests. _____

3. Valid Verilog that appears semi-readable _____

Checkoffs: Checkpoint 1 Part 2

4. No significant warnings in Synplify. _____
(warnings from FPGA_TOP are allowed)

4. Logic analyzer view of the write process and read process. _____

5. Retention time of the SDRAM at room temperature: _____

Total Score:: _____
TA: _____