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

ASSIGNED:	Friday, October 24th
DUE:	Friday, November 7th, 2:10pm sharp

Checkpoint 3
SDRAM Arbiter

1.0 Motivation

This checkpoint serves two purposes:

1. Wrap your SDRAM Controller with the Ready/Valid FIFO interface.
    a. Translate a messy interface used by the SDRAM Controller and checkpoint0bb into a clean Ready/Valid interface.
2. Allow multiple pieces of your design to communicate to SDRAM.

The Arbiter is in many ways the core of your design: its job is to allow every memory-needy device to talk to an SDRAM that only has a single DQ line. In addition to getting many requests, the SDRAM may get many requests at the exact same time. As such, it is also the Arbiter's job to prioritize read and write requests depending on the module in need and the current state of the system.

2.0 Introduction

This checkpoint will be split into two parts.

First, you must move your SDRAM Controller from simulation to a state of hardware verification. This means that **at the very least** you must have a working SDRAM Controller with the TA MT48LC16M16A2.V-based test bench and also have a 0 error count implementation of the checkpoint0bb framework. **If you do not finish your SDRAM Arbiter in time, you can get partial credit by proving that your SDRAM Controller works with checkpoint0bb.** Checkpoint0bb, once again, is not mandatory if you complete your SDRAM Arbiter and prove that it works correctly with the TA-streamed video that will be given as a blackbox to verify checkpoint 3. That said, **it is very unlikely that your Arbiter will work if you do not first have a working SDRAM Controller under checkpoint0bb's test suite.** While the Arbiter is a relatively simple module, trying to debug it without a working SDRAM Controller is futile. For this reason, it will behoove you to be able to correctly interface with checkpoint0bb.

The second part of the checkpoint is to actually build the Arbiter (SDRAMArbiter.v). The Arbiter will be required to interface with your SDRAM Controller and expose four Ready/Valid FIFO handshaking interfaces to the rest of your design. Two of these four interfaces (one for an Address and one for Data) will be used by an audio or video module to write data to SDRAM. The other (again, an

Address/Data pair) will be used to read data from SDRAM and show it on the TV through the Video Encoder.

3.0 Prelab

Please make sure to complete the prelab before you attend your lab section. **You will not be able to finish this checkpoint in 3hrs! By this time, work might be starting to pile up. Stay on top of your game!**

1. **Read this handout thoroughly**.
    a. Pay particular attention to section **4.0 Lab Procedure** as it describes what you will be doing in detail.
2. **Examine the Verilog** provided for this checkpoint.
    a. As with checkpoint 1 and 2, code will be released on Sunday.
    b. There isn't much, so it should be pretty clear.
    c. **You will use the FPGA_TOP2 from checkpoint 2 in this checkpoint and for later checkpoints.**
3. **Start your design ahead of time**.
    a. Begin with **schematics** and **bubble-and-arc** diagrams
        i. Know how to coordinate timing between the Arbiter and the SDRAM Controller.
        ii. Be able to demonstrate what happens during reads and writes.
    b. **Draw block diagrams of how the FIFO Ready/Valid interface will interact with the rest of your design.**
        i. From this checkpoint onward, the pieces that you design will feature this interface.
        ii. Knowing how to glue different pieces together with the interface is **essential** (and thanks to the interface, very simple).
        iii. If you are unclear about how the handshake works, refer to the FIFO Interface Document on the website.

4.0 Lab Procedure

Remember to **manage your Verilog, projects and folders well**. Doing a poor job of managing your files can cost you **hours of rewriting code**, if you accidentally delete your files.

4.1 SDRAMArbiter.v

This is the main module you will need to build for this checkpoint. The port specifications are given below. The essential functions of your module are listed in section 2.0 Introduction.

In the port specification for the SDRAM Arbiter, the interface to the SDRAM Controller is given in blue, and the interface to the read and write ports (connected to the rest of your design) are given in red and green, respectively.

Signal	Width	Dir	Description
Clock	1	I	System clock signal
Reset	1	I	Reset signal
RAMReadRequest	1	O	Indicates a read request (only high for one cycle).
RAMWriteRequest	1	O	Indicates a write request (only high for one cycle).
OutputEnable	1	O	Tri-state enable signal to put WriteData on the RAM_DQ line.
DataValid	1	I	Indicates to the Arbiter that the data on RAMDataIn is valid.
RAMDataOut	32	O	Data to be written to SDRAM.
RAMDataIn	32	I	Data to be read from SDRAM.
RAMAddress	24	O	**{RowAddress, BankAddress, ColumnAddress}** Corresponds to the address that will be read from or written to.
RAMInitDone	1	I	**Indicates that the SDRAM Controller has finished initializing. (Added in RevB)**
Done	1	I	**Indicates that the SDRAM Controller is not currently servicing a request. (Added in RevB)**
ReadData	32	O	The data that will be read from SDRAM to the Video Encoder and Waveform Generator side of your design.
ReadDataValid	1	O	Read the FIFO Interface Document.
ReadDataReady	1	I	Read the FIFO Interface Document.
ReadAddress	24	I	**{RowAddress, BankAddress, ColumnAddress}** The starting address corresponding to the next burst of data that will be read from SDRAM.
ReadAddressValid	1	I	Read the FIFO Interface Document.
ReadAddressReady	1	O	Read the FIFO Interface Document.
WriteData	32	I	The data that will be written to SDRAM from the audio/camera side of your design.
WriteDataValid	1	I	Read the FIFO Interface Document.
WriteDataReady	1	O	Read the FIFO Interface Document.
WriteAddress	24	I	**{RowAddress, BankAddress, ColumnAddress}** The starting address corresponding to the next burst of data that will be written to SDRAM.
WriteAddressValid	1	I	Read the FIFO Interface Document.
WriteAddressReady	1	O	Read the FIFO Interface Document.

Table 1 SDRAM Arbiter port specification