University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

CS 150
Fall 1996

J. Wawrzynek
N. Weaver

Checkpoint 3
Buffer Management

1 Objectives

For this checkpoint, you will

1. Build some of the computer interface
2. Manage memory access among various components.

2 Introduction

In your final project, the SRAM which you wired up will serve as a buffer, for temporary storage of messages between when a packet is sent to the board via the serial port or the IR link, and when the packet is passed on through the corresponding link. Thus, there are two devices which need to read memory (the serial and IR sender) and two devices which need to write memory (the serial and IR receiver).

Since modular design is a good idea, it makes sense to create a scheme to arbitrate memory accesses among components. Most of this checkpoint involves building such a component. The other part of this checkpoint requires you to build part of the computer interface.

3 Buffer Management

Since four devices need to read and write from the SRAM, it is up to an arbitrator to determine which devices get access when. The arbitrator handles this access. Probably the best way to divide the task up is by time. Since no device needs to read data on a really urgent basis, a time multiplexed scheme is sufficient. Every device is simply given access to the memory at different periods of time, and are notified when their request is successful. Another potential method is to have each device request access to the memory. In either case, the arbitrator should notify the device when it's request is complete. Also, you need to make sure that writes are handled correctly.

We logically divide the 8k SRAM into 64 separate 128 byte buffers, although we will only use a small number of these buffers in the project itself. When the computer wants to send data, it loads 127 bytes into a buffer, which is then sent out over the IR link. For receiving data from the network, the Xilinx will send the first 126 bytes to the computer.

You will need to build a circuit which recieves 127 bytes and puts it into a buffer specified by the DIP switches. It should then signal that the packet is recieves. To check that this works, the signal should toggle one of the decimal points on the 7 segment displays.

You also need to build a circuit which sends the first 126 bytes from a buffer specified by the DIP switches when the spare button is pressed, back to the computer.

Just about every circuit other then the serial communication circuits should run at 7 MHz. Since the low level communication operates at a slower speed, you will need to take this into account when designing your circuits.

Test routines on the computer will be provided, and information on these will be posted to the newsgroup.

Name: _____ Name: _____
Lab Section (Check one)
M: ☐11 ☐12 T: ☐13 W: ☐14 ☐15 Th: ☐16

4 Checkoffs

1. Send packet to buffer TA: _____(40%)

2. Return buffer to computer TA: _____(50%)

3. Turned in on time TA: _____(10%)