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

CS 150 J. Wawrzynek
Fall 1996 N. Weaver

Checkpoint 4
IR Send and CRC

1 Objectives

For this checkpoint, you will

1. Build the low and high level interface to send over the Infra Red link
2. Build and implement the CRC check.

2 Introduction

In this checkpoint, you will build the interface for sending packets over the infrared link, and the necessary hardware to detect bad packets and respond to collisions.

The first part handles the sending of bits over the IR link. The ir sender is active low, and bits are pulse time encoded (meaning that the shape and timing of the sent waveform, not just the presence or absence of a signal). Also, the IR sends the MSB of a byte first, NOT the LSB!

The second part involves the implementation of a Cyclic Redundancy Check, or CRC, which forms a fingerprint of the data. A CRC is based on a linear feedback shift register, and can be used as a sophisticated checksum on the data. The CRC is computed for a packet within your hardware, and then is sent at the end. The reciever will then recompute the CRC and compare it with the one sent along. If they match, the packet is good. Otherwise, there is a problem.

3 IR sender

The IR sender sends the data one bit at a time in a continuous stream of data, without pausing, until the entire 129 bytes (127 byte packet + 2 byte CRC) is sent. THere should be a signal which tells the sender that it should send the contents of the buffer which the computer reciever wrote into. It has 2 additional signaling inputs, an OKTOSEND signal which is high if no other device is transmitting, and a COLLISION signal if a collision is detected. It never sends data unless OKTOSEND is high, and stops if COLLISION ever goes high. When it is done, it should output a DONE signal.

When told to start sending data, it waits until a slowly clocked (at the UART reciever clock rate) 4 bit LFSR reads all 1s. Then it sends out the first byte, most significant bit first, and then, without pausing, sends the rest. The sender operates at the 7.4 MHz clock, and takes 16 cycles to output each bit. The waveforms show how the bits are encoded. The transmitter will be on pin 15, and will be ACTIVE LOW, so you will need to invert your output first.

If the reciever gets another packet before the IR sender DONE signal goes high, it should be ignored, and the packet should be dropped. So the general procedure is: Serial port reciever gets one packet (of 127 bytes), computes the CRC, and signals the IR sender. The IR sender reads each byte out of the buffer and sends it over the IR serial link one bit at a time, MSB first. It does not pause between bytes[1]. When all 129 bytes are sent, it notifies the serial port reciever that it is OK to accept another packet into the buffer.

[1] Thus, the IR sender should request the next byte BEFORE it needs it, in order to insure that it is ready in time

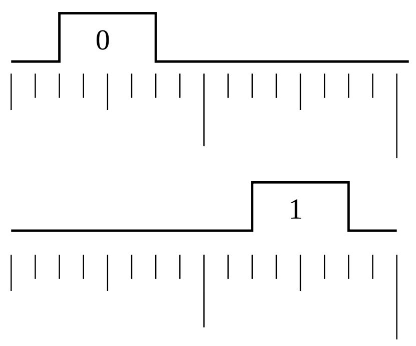

Figure 1: Pulse encoding of the bits.

4 CRC

Details will be posted to the newsgroup. Note that you should not include the first bit the computer sends in the CRC. You send the CRC after all the rest of the bytes are sent.

Name: ____________________ Name: ____________________

Lab Section (Check one)

M: ☐11 ☐12 T: ☐13 W: ☐14 ☐15 Th: ☐16

5 Checkoffs

1. Correct CRC TA: __________(40%)

2. Correctly sends over IR. (Verify with logic analyser) TA: __________(50%)

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

If you can read this, you're too close. :)