Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.111 Introductory Digital Systems Laboratory (Spring 2009)

Laboratory 3 – Infrared Remote Checkoff Sheet

Student Name:____

TA Signature/Date:______________________________________

Must show to TA at beginning of checkoff:

- Finite State Machine diagram ☐
- Logic analyzer display of input waveform and FSM states ☐
- Verilog code available ☐

Be able to demonstrate your working lab:

- Demonstrate the “Learning” of four remote commands. ☐
- Demonstrate the transmission of the learned commands. ☐

Be able to respond to any of the following questions and possibly others: ☐

- What are possible problems encountered during the “Learn” mode?
- Describe your timing of your receiver and how wide of a tolerance in bit timing is acceptable with your design.
- Describe how you would implement the IR transmitter FSM.

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.111 Introductory Digital Systems Laboratory - Spring 2009

Laboratory 3 – Infrared Remote Control

Handed Out: 3/4/2009
Checkoff (see page 7 for details on what to turn in): 3/12/2009

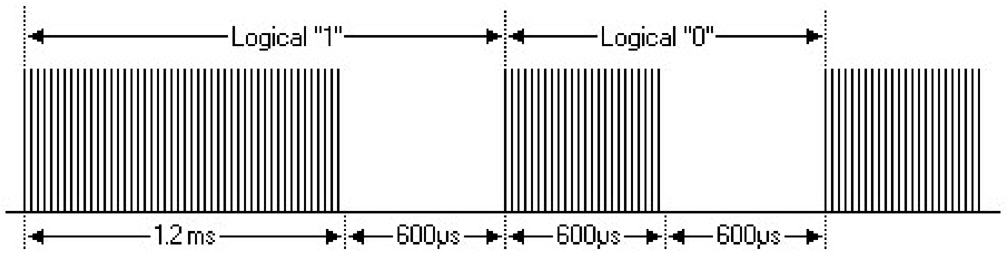
Introduction

In this lab, you will design finite state machine(s) to "learn" four Sony Infrared Command (SIRC) and use it to control a Sony television.

Serial communications, where data is transmitted over a single channel one bit at a time, is used every where. Examples are PS/2 mouse and keyboard, USB port and PCI-Express in newer PC's. Another example is the wireless infrared remote control used for TV's and stereo. Serial communications is in contrast to sending many bits in parallel, for example the parallel port on older PCs. Serial communications saves on cable pins, wires and cost. However, to implement serial communications, you need a "serializer" at the transmitter end and a "deserializer" at the receiver. (See http://web.mit.edu/6.111/www/f2008/handouts/L14.pdf for more details.)

Serial communications can be asynchronous (no clock line) such as RS-232 or the wireless infrared remote control. USB controllers and the PS/2 interface are synchronous and use a separate line for the clock.

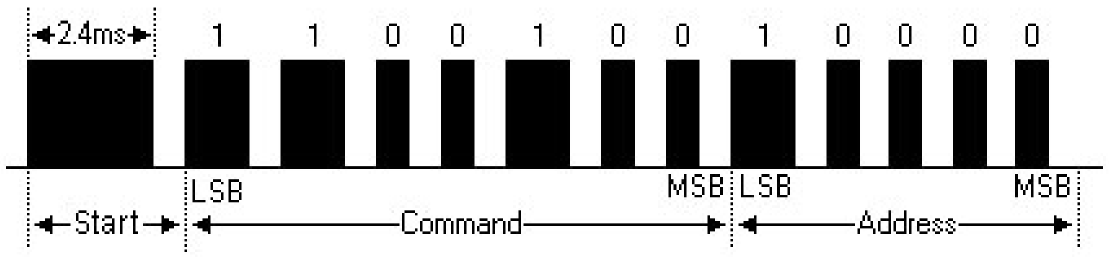
For the wireless remote, the communications channel/medium is infrared light at 950 nm wavelength. Sony uses a 12 bit, 15 bit and 20 bit protocol. We will be using the 12 bit pulse-width modulated serial protocol. With this protocol, a data "1" is 1200 us and a data "0" is 600 us. Bits are separated by 600us[1]



During the "on" time, the signal is a 40 kHz square wave. For simplicity, we will not show the 40 kHz carrier in future timing waveforms.

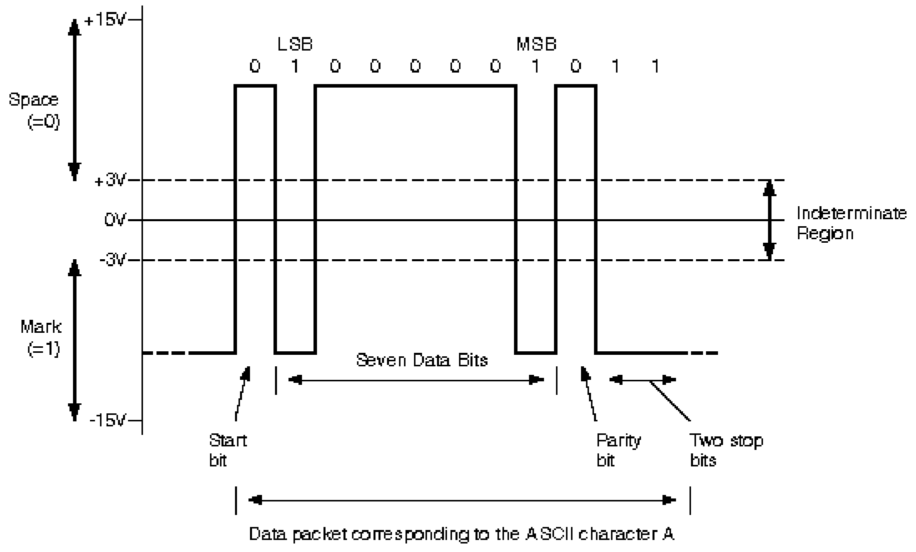
[1] Picture from www.sbprojects.com/ir/sirc.htm

The format of the data stream is shown below. In this example a command 19 (volume lower) is issued for the TV. [1] A partial listing of the commands is attached at the end.



With this protocol, the length (duration) of each command is a function of the data.

The protocol used by the IR remote control is similar to the protocol used by RS-232, a common serial bus standard used in computer serial ports. The RS-232 protocol differs from the remote control protocol in that it uses fixed width encoding. 0's and 1's are encoded with high and low voltages, rather than by the length of pulses. Thus the data rate is independent of the data and is purely a function of the bit (baud) rate. The waveform for RS-232 is shown below:
(fromhttp://www.arcelect.com/rs232.htm)

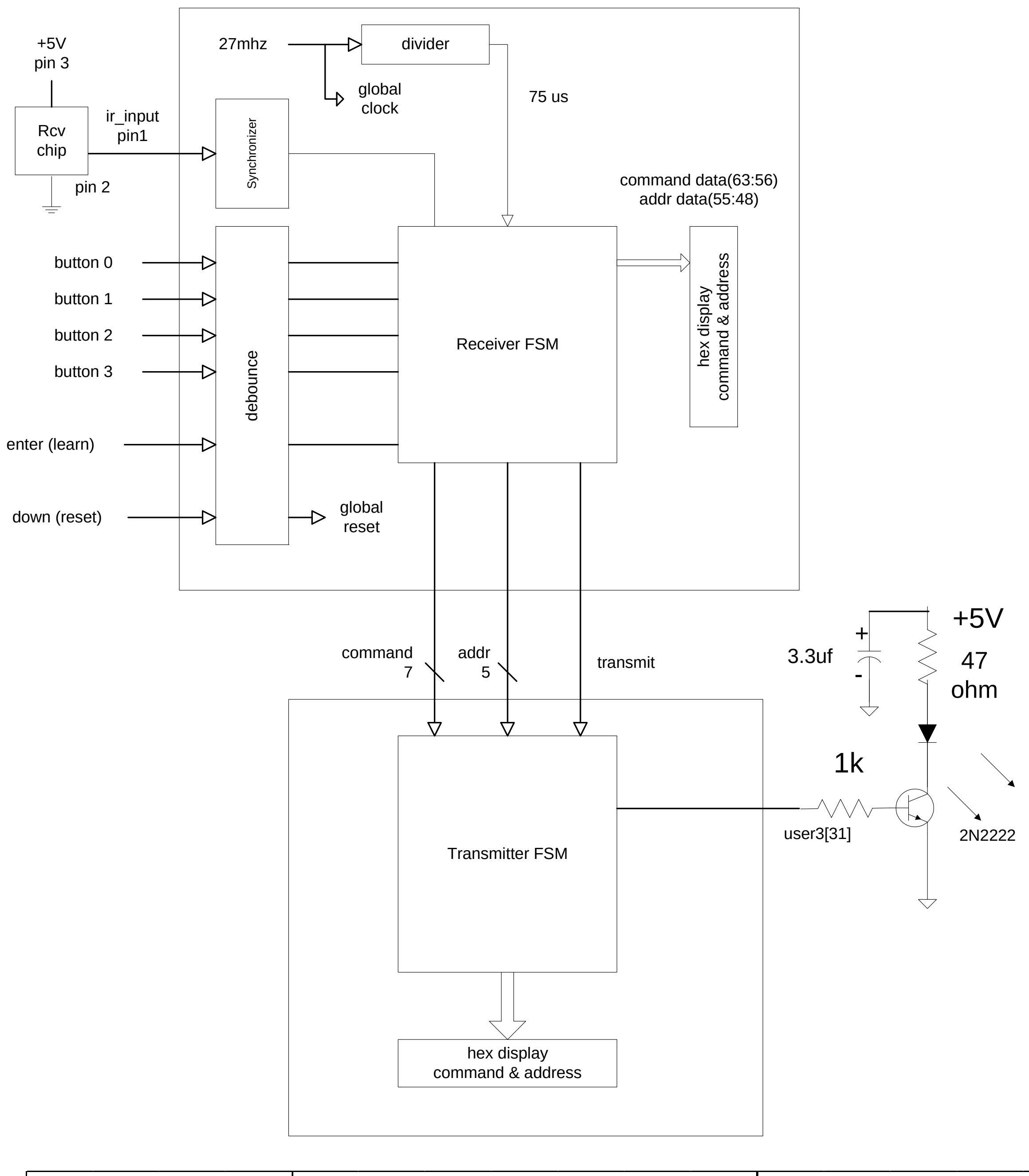


Procedure

The Lab exercise consists of two parts. You will wire up a simple circuit with an infrared LED transmitter, compile a Verilog source file that we provide and verify that the circuit controls the TV. The source file contains the FSM that creates the 40 kHz carrier and serializes the data stream to the infrared LED. In the second part you will wire an infrared receiver and design a FSM that learns new Sony commands, stores the command for use. While the data rate for IR controls is such that a FGPA is not really needed and can be implemented with software, the design principles are applicable at much higher data when hardware must be used.

1 Picture from www.sbprojects.com/ir/sirc.htm

The following diagram illustrates a possible organization of your design. This diagram is a high level view and does not show every necessary signal.

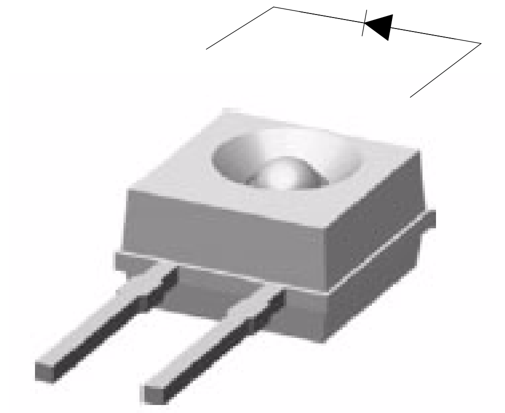


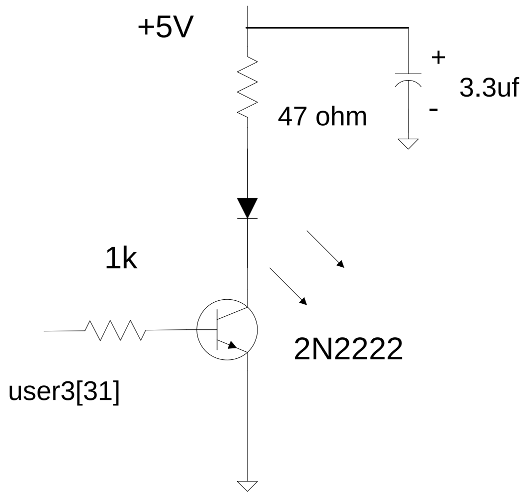
decoding				for your debugging use							transmit				
addr MSB	addr LSB	command MSB	command LSB								1=transmit	addr MSB	addr LSB	command MSB	command LSB

16 digit hex display

Part 1

Using your protoboard (not the labkit), wire up the infrared transmitter (Vishay TSKS5400S Infrared Emitting Diode, 950 nm, GaAs). The led is the blue component. In order to avoid damage to the user i/o when driving the high currents required for the infrared LED, an external bipolar junction transistor (BJT) 2N2222, is used. When user3[31] is high, current flows through the LED. Use the +5V from the labkit. A 3.3uf capacitor is added for noise filtering.





The pin out for the 2N2222 BJT is shown below. Note that the collector (pin 3) is connected electrically to the case!

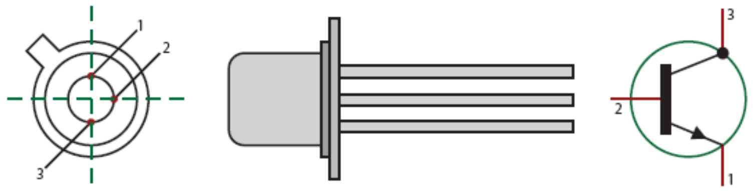
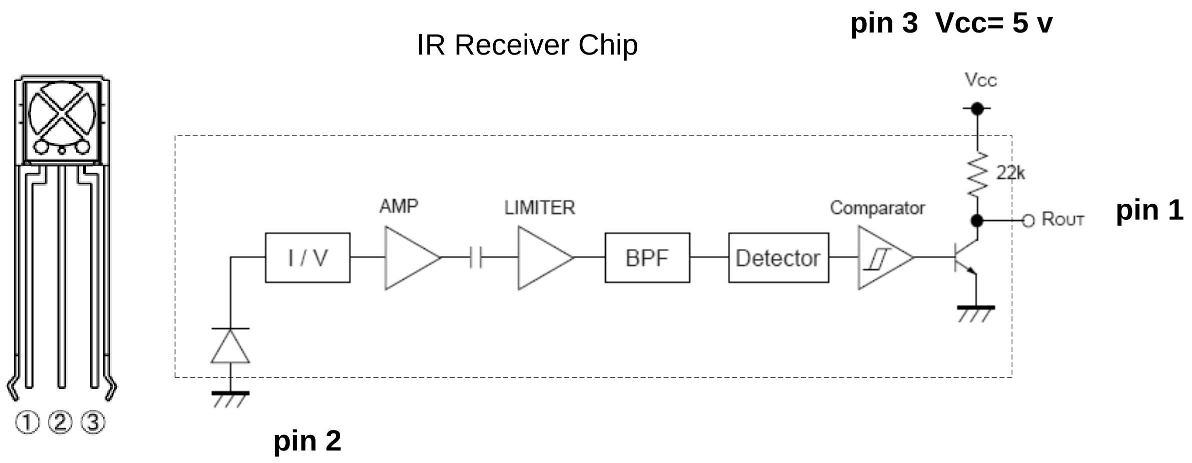


Figure by MIT OpenCourseWare.

Compile and load the bit file. Verify that this changes the channel on the TV using the *up button*. The command and the address are displayed on the led display. Observe the signal from the user output on an oscilloscope.

Part 2
Again, using the protoboard and not the labkit, wire up the infrared receiver (RPM7140-R) on your protoboard. The receiver is an integrated infrared digital chip that demodulates the 40 kHz signal and provides an inverted output signal to the labkit. The output of the receiver (pin 1) is the input signal to the labkit. All that is required is to supply Vcc = 5V and ground. Use the +5 and ground from the labkit. Do not use the external power supply. This will minimize grounds and noise. (You may keep the components in this lab for your personal use.)



Design and implement a digital system that will read the data from the receiver, learn four commands and then transmit the commands using button 0, 1, 2 and 3 with your transmitter.

Receiver FSM Design Requirements

With serial communications, incoming data may be glitchy. The glitch could be caused by noise over the incoming channel or noise from current spikes in the detector circuit. For these reasons, it is not good design practice to use an edge of the data as a trigger [example: always @ (posedge data)]. ***For this lab, your receiver FSM must be implemented by over sampling the incoming data.*** With sampling, the probability of an error is greatly reduced since the glitch must coincide with the sampling. (You can also do additional digital filtering to further remove noise.) Implement your "Learn" mode as follows:

1. press enter to go into learn mode.
2. press button 0,1,2, or 3 to store the command in the button 0,1,2 or 3 respectively.
3. send the command to be learned with the TV remote (you may swap step 2 and step 3 for your implementation). Use the "up channel" command provided in part 1 for debugging your receiver FSM Verilog.
4. display the address and command decoded on the left most 4 digits on the hex display.
5. send the learned command with the transmitter and verify that it controls the Sony TV.

Design and Implementation suggestions.

1. Carefully design the FSM and use ModelSim to check out the design. Consult with staff on your design.
2. Use your transmitter which sends a known command as the data source to be learned. Use the switches as input tot the transmitter to change the command.
3. Display the input waveform on the logic analyzer.

4. Use the 7 unused digits in the hex display to display your FSM states, timer value, etc. During transmission, a “1”, the address, and command is displayed the right side of the hex display.

What to Turn In:

You are not required to write a detailed report for this lab. However, you must turn in the following:

1) Check off Sheet signed by a member of the 6.111 teaching staff
2) Verilog Code

Appendix

A subset of the Sony commands and address is listed below.

Address	Device
1	TV
2	VCR
3	VCR
17	CD Player

Command (decimal)	Function
0	Digit 1
1	Digit 2
2	Digit 3
3	Digit 4
4	Digit 5
5	Digit 6
6	Digit 7
7	Digit 8
8	Digit 9
9	Digit 0
16	Channel +
17	Channel -
18	Volume +
19	Volume -
20	Mute
21	Power

Acknowledgment
We would like to thank Adam Lerer for his help in the development and debugging of this lab and Dr. Chris Terman for his notes on serial communications. This lab was based in part on Wolfe Styke’s Fall, 2008 6.111 project.