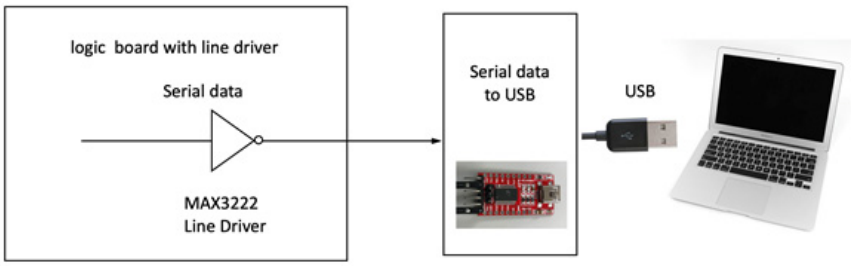
M A S S A C H U S E T T S I N S T I T U T E O F T E C H N O L O G Y
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.111 Introductory Digital Systems Laboratory
Fall 2019

Lecture PSet #2
Due: Tue, 09/17/19 2:30p
Upload as PDF

Problem 1 [3 points]. [As noted previously, MIT lpset problems are clearly and neatly structured. But in the real world, necessary information for a design is there but typically scattered across multiple sources. This problem is another sample of what you might encounter as an engineer.]

In lab 2, we'll send serial directly from the Nexys4 IO to a serial to USB chip. This method is fine over a few inches. A more robust method to is use a MAX32222 RS-232 line driver going to a USB interface chip such as a FTDI FT232H or Prologic PL2303. [Using Ethernet would be overkill and require significant computer resources.]



RS232 serial port driver using the MAX32222 IC
http://web.mit.edu/6.111/www/f2017/serial/max3222.pdf

The specifications for the PL2303 IC are shown in the datasheet
http://web.mit.edu/6.111/volume2/www/f2019/handouts/pl2303v1_4_4.pdf

For a project you want to transfer data as fast as possible. Based on the above information,
(a) What device limits the maximum data transfer rate?
(b) What is the maximum data transfer rate?

(over)

Problem 2. [3 points] Endianness refers to the ordering of the bytes. With big endian, the most significant byte is sent first. Endianness can also refer to the bit level. UART (lab2), USB and Ethernet use little endian at the bit level, meaning it transmits the least significant bit first. (Think of a shift register where the bits are shifting in left to right.) Other protocols are different, with I2C being most-significant bit first and SPI varying depending on the application.

To make things more complicated, the Ethernet protocol while it uses little endian at the bit level, uses big endian at the byte level! So, a 16-bit value with bits being labeled 0 to 15 for lsb to msb, respectively, is transmitted with Ethernet as bit order: 8-9-10-11-12-13-14-15 - 0-1-2-3-4-5-6-7. See IEEE 802.3 Clauses 3.1.1, 3.2.6, and 3.3 for the gruesome details.

At the end of Lab2, you write a serial_tx module that will send out a one-byte UART packet. To adhere to standard, this sends data out in lsb (little-endian byte format). For flexibility, it would be good to allow both ways. To handle this endianness, write a Verilog module that will load a register with the endianness determine by a switch name big endian.

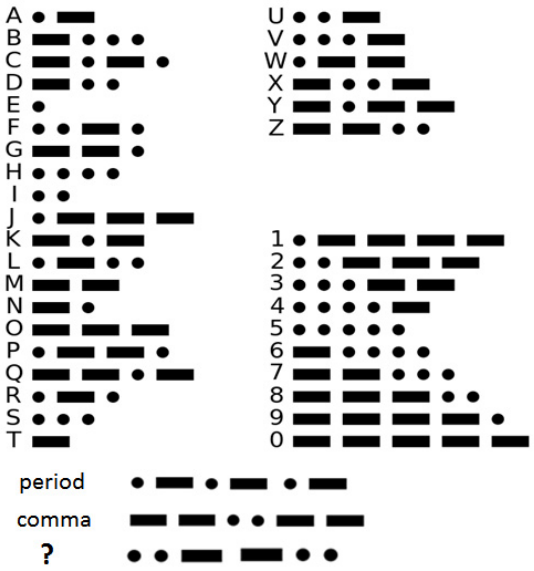
```
module endian(
    input [7:0] data,             //  data to be sent
    input big_endian,             //  programmable switch, if true = big endian
    output logic [7:0] send_data  //  send_data[0] is sent first.
    );

   //  data is a 8 bit Verilog register.
   //
   //  send_data is a shift register that will send out eight bits in serial
   //  fashion with send_data[0] first to a receiver.
   //
   //  You are unsure about whether the receiver at the other end is
   //  expecting big or little endian. So you have switch that load
   //  the send_data register with either little or big endian format.
   //
   //  Your Verilog here
   //
endmodule

   // here is  the top level with an instance of endian
   // sw[7:0] is the data being transmitted
    logic [7:0] ordered_data;
    endian my_endian(.data(sw[7:0]), .big_endian(sw[8]), .send_data(ordered_data));
    serial_tx my_tx (.clk_in(clk_100mhz), .trigger(edge_detect),
        data_in(ordered_data), .data_out(ja[0]));
```

Problem 3. [4 points] In engineering, the straightforward solution to a design problem generally is not optimal in terms of gates, power, number of wires, etc. Using this first pass is fine for proof of concept but for production, the design may need to be optimized to meet the design constraints.

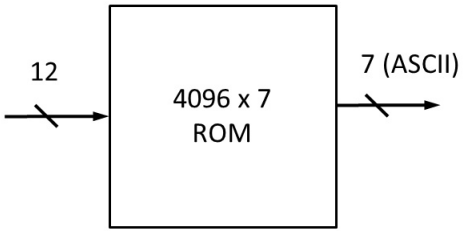
[This problem is based on the instructor's 6.111 project, a Morse code to ASCII[1] text translator: as Morse code is sent, the 6.111 system prints the characters on an old Teletype printer. The solution to this problem saved countless hours by reducing the size of the Karnaugh map.]



Morse code is a variable length encoding scheme to represent letters, numbers and punctuation marks. (There is no lower case in Morse code.) The encoding length is one to six symbols. A symbol is a dot or a dash. As an example the mapping for the letter E is one dot while a period is dot-dash-dot-dash-dot-dash. (Only three of the many punctuation marks are shown.) The tone (length) for a dash is three times that of a dot with long periods of silence between characters.

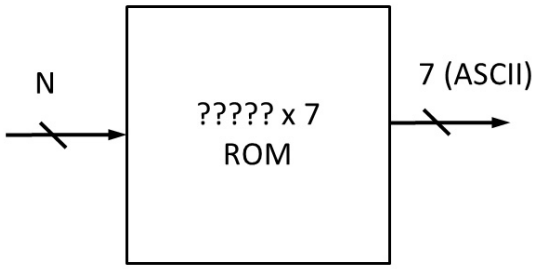
The first thought is to design a finite state machine (FSM) which outputs a fix length six element register representing the Morse code input. Each element is two bits with a dot encoded as 01, a dash as 11 and empty as 00.

Using the letter A as an example, the FSM will see a dot followed by a dash followed by a period silence. So 01-11-00-00-00-00 is stored in the register. To send the letter A to the printer, 01-11-00-00-00-00 is mapped to the ASCII code (7 bit encoding) for A and 0x41 is transmitted. In essence it's a mapping from 12 bit to 7 bits, i.e. a 4096 x 7 ROM or a giant Karnaugh map can be created. [In the project a Karnaugh map was used – flash was not yet invented.]



[1] ASCII, American Standard Code for Information Interchange, is a character encoding standard for electronic communication. Text is represented with a 7 bit code: http://www.asciitable.com. Later extensions allow for an eighth bit for symbols and other non-text characters.

The FSM can be redesigned and optimized using fewer bits to represent the incoming Morse code thus reducing the size of the ROM. Describe the new encoding. Using this optimized encoding show the register representation for the characters A, B, D, E, and a period. How many bits are required for this encoding scheme? Of course, N<12! Full credit for N(min), partial credit for N<12. [The solution for N(min) is not unique.]



Revised encoding

A (dot dash) = _____

B (dash dot dot dot) = _____

D (dash dot dot) = _____

E (dot) = _____

Period (dot dash dot dash dot dash) = _____

N = _____