

Homework 5 v1.1  
EECS150, Fall 2012  
Due Friday, 6 pm, Sept 28, outside 125 Cory

1. You have a null-terminated string at location 0x10001234. Write the MIPS assembly code necessary to send that string out the serial port. The serial port data address is 0xffff000c, and the control address is 0xffff0008. The UART will put a 0x1 in the control register when it is ready for another byte. Writing to the data address when the serial port is not ready will result in unexpected behavior. You may use whatever registers you like.
2. Design a programmable 11 bit timer. The timer has two 11 bit inputs, T0 and T1, a SEL input, and a START input, and a DING output. When START is asserted, the timer should begin counting to either T0 or T1 according to the SEL input. When the count is reached, DING should be asserted for one clock cycle.
3. Using multiplexers and other simple gates, and D-FFs (for example, the one in Figure 3.13) design a 10 bit shift register with parallel in and out as well as serial in and out, a LD input that controls the parallel load, and a SH input that controls the shifting.
4. Using the shift register from the previous problem, design a serial receiver/transmitter that runs at a fixed baud rate of 9600. Assume that you have a clock at  $2^{10} \cdot 9600$  Hz. The system should provide a ready/valid interface on both the TX and RX port, with two 8 bit wide data buses, and a serial in and a serial out line.
  - a. sketch the datapath. Identify all inputs and outputs from your datapath.
  - b. Draw a diagram with two boxes labeled "CONTROL" and "DATAPATH" and show which signals go in which directions between the boxes. There should also be serial and parallel input and output lines coming out of the datapath, and ready/valid lines coming out of your control FSM.
  - c. Using the timer above, design a control FSM for the RX portion of your datapath.
5. Golden Bear Circuits is designing an electrical outlet monitoring chip. The analog gurus have figured out to get a 16 bit sample of both the voltage and current through the outlet. These values are synchronously clocked on two 16 bit buses at a rate of  $2^5 \cdot 60$  Hz. You have a clock at twice that frequency available to you. You need to calculate the real power being drawn by the outlet every second. The result for the previous second should be available in a 16-bit register while the next result is being calculated. Assuming that you have an ALU with both ADD and MUL instructions, draw the datapath and list the control signals.