

Digital System Design

Fall 2011 Preliminary Exam Questions

Name: _____ Date: _____

Each problem is worth 10 points.

Problem	Max Points	Student Points
Problem 1	10	
Problem 2	10	
Problem 3	10	
Problem 4	10	
Problem 5	10	
Problem 6	10	
Total	60	

$$\text{Score} = \frac{\quad}{60} * 100\% = \underline{\quad}$$

1. Design a combinational circuit that compares two four-bit numbers to check if they are equal. The circuit output is equal to 1 if the two numbers are equal to 0 otherwise.

2. Design a combinational circuit using a ROM. The circuit accepts a 3-bit number and generates an output binary number equal to the square of the input number.

3. A 32K X 8 RAM chip uses coincident decoding by splitting the internal decoder into row select and column select.
- (a) Assuming that the RAM cell array is square, what is the size of each decoder, and how many AND gates are required for decoding an address?
- (b) Determine the row and column selection lines that are enabled when the input address is the binary equivalent of $(21000)_{10}$

4. A traffic metering system for controlling the release of traffic from an entrance ramp onto a superhighway has the following specifications for a part of its controller. There are three parallel metering lanes, each with its own stop (red) - go (green) light. One of these lanes, the car pool lane, is given priority for a green light over the other two lanes. Otherwise, a "round robin" scheme in which the green lights alternate is used for the other two (left and right) lanes. The part of the controller that determines which light is to be green (rather than red) is to be designed. The specifications for the controller follow:

Inputs

- PS Car pool lane sensor (car present – 1; car absent-0)
LS Left lane sensor (car present – 1; car absent-0)
RS Right lane sensor (car present – 1; car absent-0)
RR Round robin signal (select left – 1; select right -0)

Outputs

- PL Car pool lane light (green-1; red – 0)
LL Left lane light (green-1; red – 0)
RL Right lane light (green-1; red – 0)

Operation

1. If there is a car in the car pool lane, PL is 1.
2. If there are no cars in the car pool lane and the right lane, and there is a car in the left lane, LL is 1.
3. If there are no cars in the car pool lane and in the left lane, and there is a car in the right lane, RL is 1.
4. If there is no car in the car pool lane, there are cars in both the left and right lanes and RR is 1, then LL=1.
5. If there is no car in the car pool lane, there are cars in both the left and right lanes, and RR is 0, then RL =1.
6. If any PL, LL or RL is not specified to be 1 above, then it has value 0
 - a. Find the minimum multiple-level gate implementation with minimum gate input cost using AND gates, OR gates and inverters.

5. A pair of signals Request (R) and Acknowledge (A) is used to coordinate transactions between a CPU and its I/O system. The interaction of these signals is often referred to as a “handshake.” These signals are synchronous with the clock and, for a transaction, are to have their transitions always appear in the order shown in Figure 6.1. A handshake checker is to be designed that will verify the transition order. The checker has inputs, R and A, asynchronous reset signal, RESET, and output, Error (E). If the transitions in a handshake are in order, E=0. If the transitions are out of order, then E becomes 1 and remains at 1 until an asynchronous reset signal (RESET=1) is applied to the CPU.
- Find the state diagram for the handshake checker.
 - Find the state table for the handshake checker.

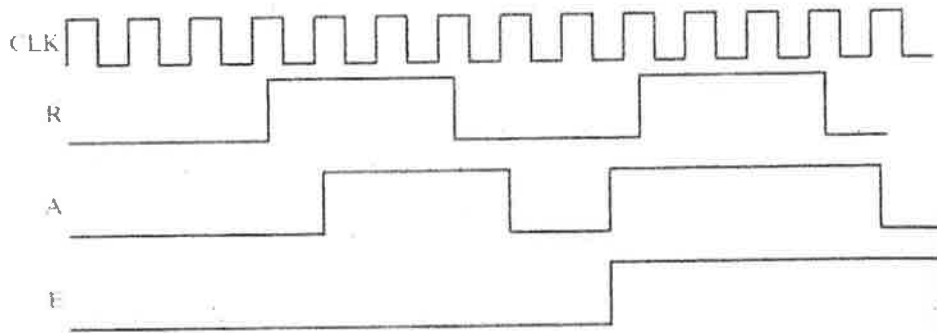


Figure 6.1: Signals

6. Design a digital system with three 16-bit registers AR, BR, and CR and 16-bit data input IN to perform the following operations, assuming a 2s complement representation and ignoring overflow:
- a. Transfer two 16-bit signed numbers to AR and BR on successive clock cycles after a go signal G becomes 1.
 - b. If the number in AR is positive but nonzero, multiply the contents of BR by two and transfer the result to a register CR.
 - c. If the number AR is negative, multiply the contents of AR by two and transfer the result to register CR.

