***King Fahd University of Petroleum and Minerals***

***College of Computer Science and Engineering***

***Computer Engineering Department***

**COE 202: Digital Logic Design (3-0-3)**

**Term 131 (Fall 2013)**

**Final Exam**

**Monday December 30, 2013**

**7:00 p.m. – 9:30 p.m.**

**Time: 150 minutes, Total Pages:**

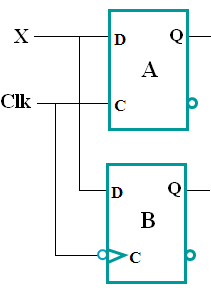
**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID: \_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_**

**Notes:**

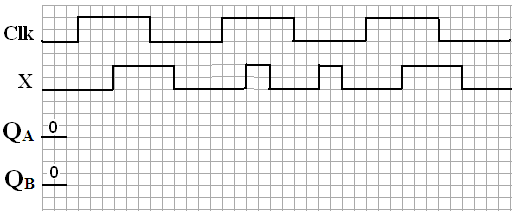
* Do not open the exam book until instructed
* Calculators are not allowed (basic, advanced, cell phones, etc.)
* Answer all questions
* All steps must be shown
* Any assumptions made must be clearly stated

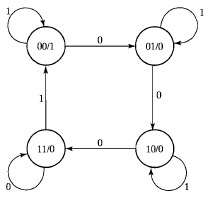
|  |  |  |
| --- | --- | --- |
| **Question** | **Maximum Points** | **Your Points** |
| **1** | **13** |  |
| **2** | **12** |  |
| **3** | **10** |  |
| **4** | **10** |  |
| **5** | **10** |  |
| **6** | **10** |  |
| **7** | **15** |  |
| **Total** | **80** |  |

Question 1. (**13 points)**

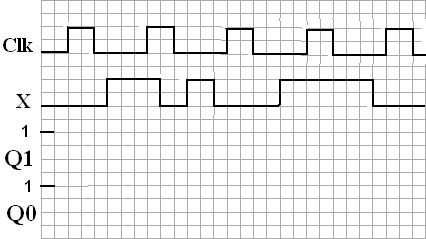
1. In the circuit shown, A is a D-type latch and B is a D-type flip flop. For the input waveforms given for the clock signal (Clk) and the input X, accurately draw the resulting waveforms at outputs **QA** and **QB**.

Assume that both QA and QB are initially at 0.





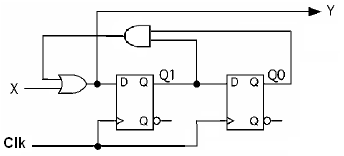
1. The state diagram shown is for a sequential circuit that has a single input X and a single output Y. The circuit uses two positive edge triggered D-type flip flops Q1 and Q0.
2. Starting with the circuit in state Q1Q0=11, complete the missing waveforms in the timing diagram below.



1. Let the circuit be in state 00 with input X held permanently at 0. The circuit will end up being stuck at state \_\_\_\_\_\_\_. With the circuit operating at a clock frequency of 2 KHz, this state transition takes a minimum time duration of \_\_\_\_\_\_\_ ms.

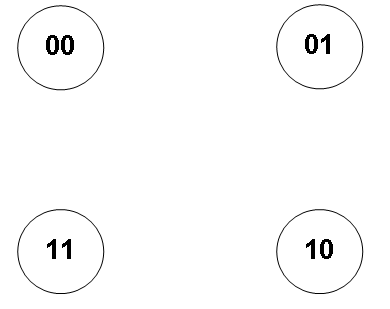
**Question 2. (12 Points)**

Consider the sequential circuit opposite and then answer the following questions:

a. Is the circuit Mealy or Moore?

b. Provide logical expressions for the flip flop D inputs and the external output

c. Give both the **state table** and the **state diagram**. Use the layout given below for the state diagram. Note: Q0 represents the LSB of the binary value of the state.



**Question 3. (10 Points)**

# It is required to design a synchronous sequential circuit that receives a serial sequence of **3-bit codes** through input **X** and produces **1** through output **Y** when the received 3-bit code equals either 010 or 110 (i.e., either 0 followed by 1 followed by 0, or 1 followed by 1 followed by 0). Assume the availability of an asynchronous reset input to reset the machine to a reset state. Draw the state diagram of the circuit assuming a **Mealy** model with **minimum** number of states. *You are not required to derive the equations and the circuit*. The following is an example of an input and output sequence:

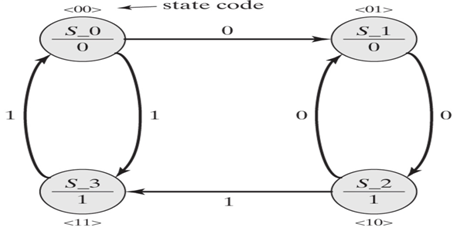
Example:

**Time**

|  |  |  |
| --- | --- | --- |
| Input | **X** | 0 1 0 0 0 1 0 0 1 1 0 1 1 1 0 |
| Output | **Y** | 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 |

**Question 4. (10 Points)**

The following state diagram represents a synchronous sequential circuit having a single input **X** and a single output **Y.** Note that the unspecified (missing) transitions in the state diagram do not occur (i.e. don’t care). The states are assigned the following state codes **S\_0=00, S\_1=01, S\_2=10** and **S\_3=11**. Assume the existence of an **asynchronous reset** input to reset the machine to state S\_0.



# 

## Draw the state transition table for the sequential circuit.

## Using D-FFs and **minimal** combinational logic determine the equations for the D-FFs and output Y for this sequential circuit.

## Draw the resulting circuit.

**Question 5. (10 Points)**

Using ***only*** D flip-flop(s), MUX(s), and **XOR gate(s),** draw the logic diagram for a 4-bit register with 2 mode selection inputs *M*1*M*0 and 4 load inputs *I*3*I*2*I*1*I*0. Note that D flip-flop outputs include both the state and its complement (i.e., and ) available for use. The register should operate according to the following table:

|  |  |
| --- | --- |
| ***M*1*M*0** | **Register operation** |
| 00 | No change. |
| 01 | Parallel Load. |
| 1x | Shift *right* while feeding in an **ODD** parity bit for the 3 bits that remain in the register **after** shifting.  (**Examples**: **1.** register content **before** shifting = 0110,  register content **after** shifting = 1011  **2**. register content **before** shifting = 1001,  register content **after** shifting = 0100) |

**You must clearly label the D flip-flop(s) and MUX(s) inputs and outputs.**

**Question 6. (10 Points)**

Consider the following state transition table for a synchronous sequential circuit that detects five consecutive 1’s. The circuit has a single input **X**, a single output **Z**, and three state variables **Y0**, **Y1**, and **Y2**. The states are encoded using binary codes **001**, **010**, **011**, **100**, and **101.**

|  |  |  |
| --- | --- | --- |
| **PS** | **(Y2 Y1 Y0)t+1** | **Z** |
| (**Y2 Y1 Y0**)t | **X = 0 X = 1** | **X = 0 X = 1** |
| 0 0 1 | 0 0 1 0 1 0 | 0 0 |
| 0 1 0 | 0 0 1 0 1 1 | 0 0 |
| 0 1 1 | 0 0 1 1 0 0 | 0 0 |
| 1 0 0 | 0 0 1 1 0 1 | 0 0 |
| 1 0 1 | 0 0 1 1 0 1 | 0 1 |

You are required to implement the above circuit using a **ROM** device and a **register**. The circuit should be designed such that **any unused state should go to the initial state 001**.

1. What is the minimum size of the register (i.e., number of D flip-flops)? **[1 pt]**
2. What is the minimum size of the ROM (number of memory locations × number of memory bits per location)? **[2 pt]**
3. Draw the block diagram for such an implementation. (**Label all components inputs and outputs together with various signals**) **[3 pts]**
4. Starting in the initial state **001**, what is the sequence of ROM locations addresses that will be accessed as a result of applying an input sequence **X = 011** where **0** is applied first. **[2 pts]**
5. Starting from address **0,** complete the following table to show the data stored in the first six memory locations in the ROM device **[2 pts]**

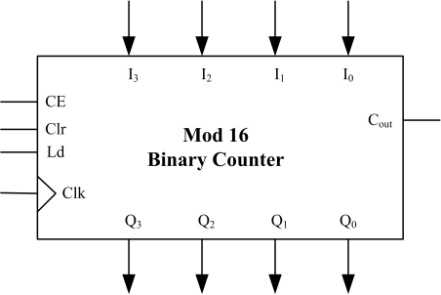
|  |  |
| --- | --- |
| **Binary Address** | **Binary Stored Data** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Question 7. (15 Points)**

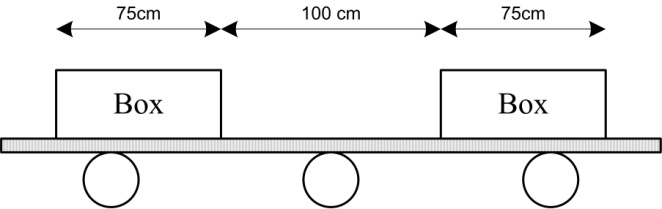
In an assembly line a conveyer belt has a sensor that generates a signal **X** whenever a cartoon box passes down the belt. Each dozen such boxes are placed in a container. The containers are loaded in trucks whose capacity is 15 containers. A signal **Y** is to be generated whenever the truck is full to open an automatic gate letting the truck out. You are to design a counting system that generates signal **Y** andanother signal **W** which equals **1** whenever a container is full**.**

1. Design this counting system *using* **2** *counters*; one to keep track of the number of boxes in a container and another to count the number of containers loaded in a truck. Clearly show how the **W** and **Y** signals are generated.

Design this system using **mod-16** counters with count- enable (**CE**) and load (**Ld**) inputs together with the 4 parallel inputs (**I3 I2 I1 I0**). In addition, an ***asynchronous*** clear (**Clr**) input is also available. The outputs of the counter are the 4 count bits (**Q3 Q2 Q1 Q0**) and a carry-out signal **Cout** which equals **1** when Q3Q2Q1Q0 = 1111.



Draw the complete block diagram of the counting system showing all logic components needed / used by the system **(6 Pts [3+3])**

1. Modify the counting system you designed in part (I) to make it self-resetting, i.e. whenever the system falls in any unused state, the count is automatically reset to zero. (**3 Pts**)
2. The carton boxes are 75 cm long, and are placed 100 cm apart on the conveyer belt. Signal **X** maintains a value of **1** until the full length of the box passes beyond the sensor. Given that the belt moves at a speed of *5 meters/sec*;
   1. Plot the *waveform* of signal **X** (value of X versus time). **(2 Pts)**
   2. Neglecting propagation, setup and hold delays, determine the maximum possible clock frequency. What happens if a higher frequency is used? **(2 Pts [2+1])**
   3. Neglecting propagation, setup and hold delays, determine the minimum possible clock frequency. What happens if a lower frequency is used? **(2 Pts [2+1])**