*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: 12**

**Name: _____ ID: _____ Section: _____**

**Notes:**

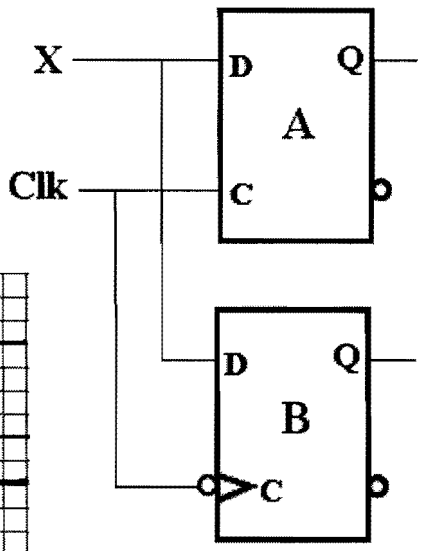
- Do not open the exam book until instructed
- Calculators are not allowed (basic, advanced, cell phones, etc.)
- Answer all questions
- All steps must be shown
- Any assumptions made must be clearly stated

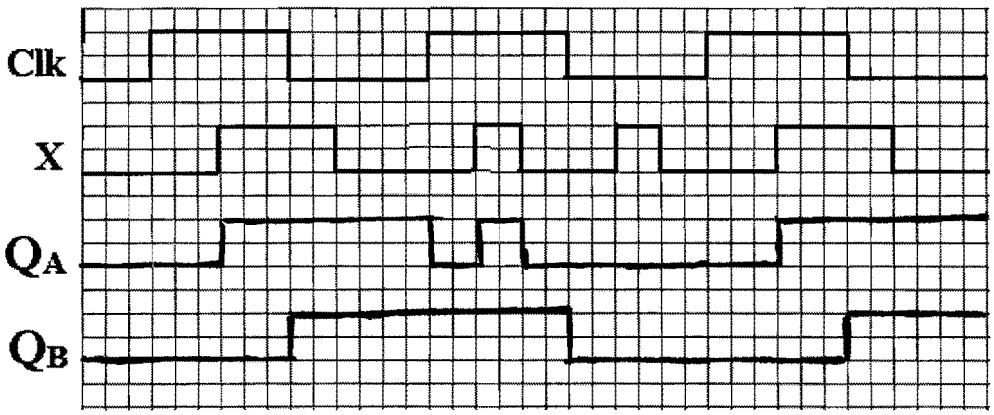
| Question | Maximum Points | Your Points |
|---|---|---|
| 1 | 13 | |
| 2 | 12 | |
| 3 | 10 | |
| 4 | 10 | |
| 5 | 10 | |
| 6 | 10 | |
| 7 | 15 | |
| Total | 80 | |

**Q1** **(13 Points)**

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

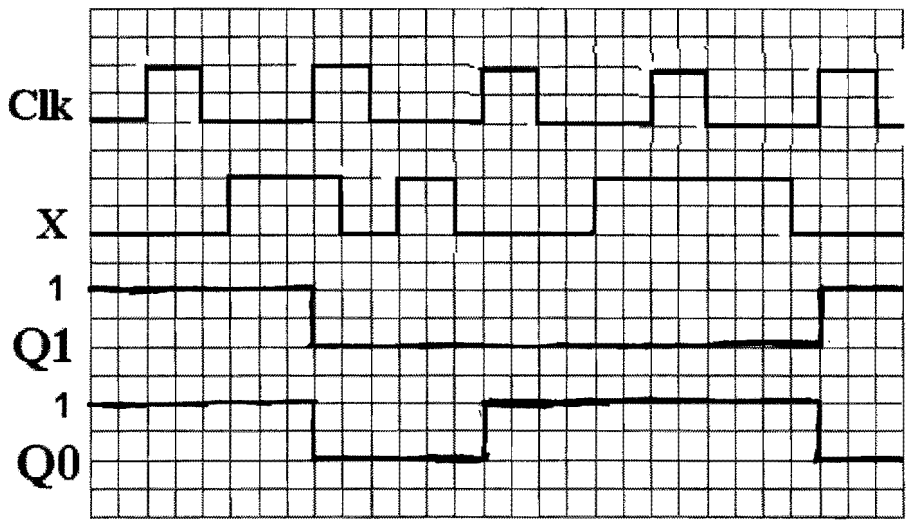
Assume that both $Q_A$ and $Q_B$ are initially at 0.

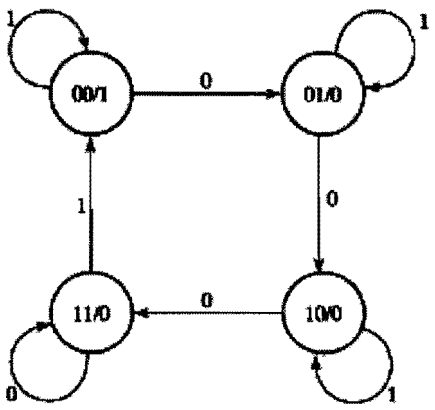



b.
The state diagram shown is for a sequential state that has an input X, and output Y, and state Q1Q0. The circuit uses positive edge triggered D-type flip flops and operates from a 2 kHz clock.

i. Starting with the circuit in state Q1Q0=11, complete the missing waveforms in the timing diagram below.




ii. Let the circuit be in state 00 with input X held permanently at 0. The circuit will end up being stuck at state 11. This state transition requires a minimum time duration of 1.5 ms.

3 X T = 3 X 0.5 ms

**Q2** **(12 Points)**

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

a. Is the circuit Mealy or Moore?

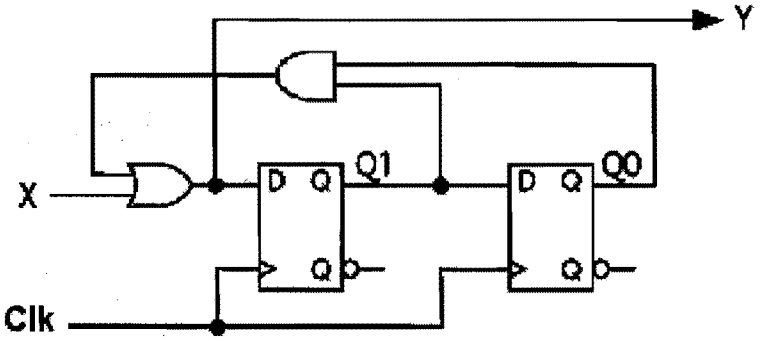
Mealy

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

$D_{Q_0} = Q_1$

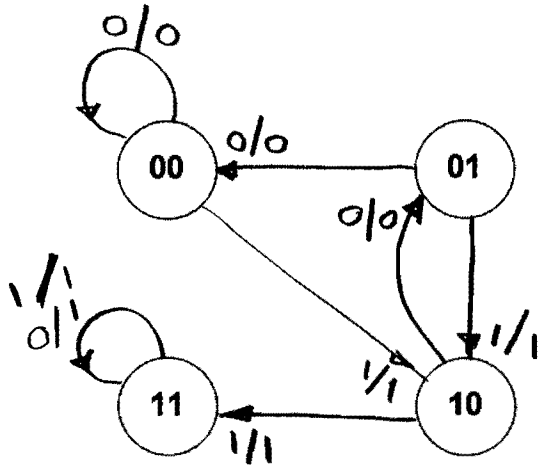
$D_{Q_1} = Q_0 Q_1 + X$

$Y = Q_0 Q_1 + X$




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.

| $Q_1$ | $Q_0$ | X | $Q_1^+$ | $Q_0^+$ | Y |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |



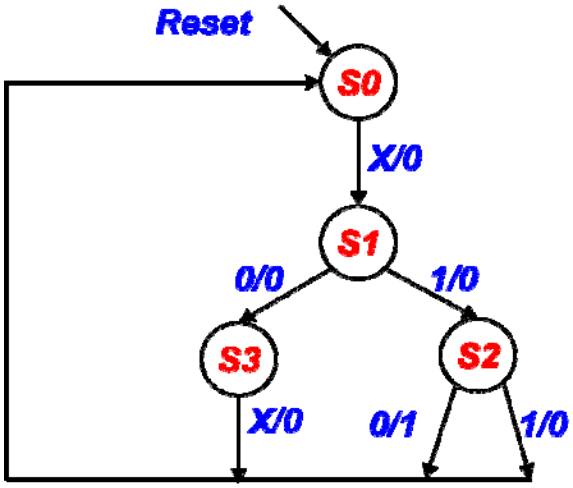
**Question 3.** **(10 Points)**

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:

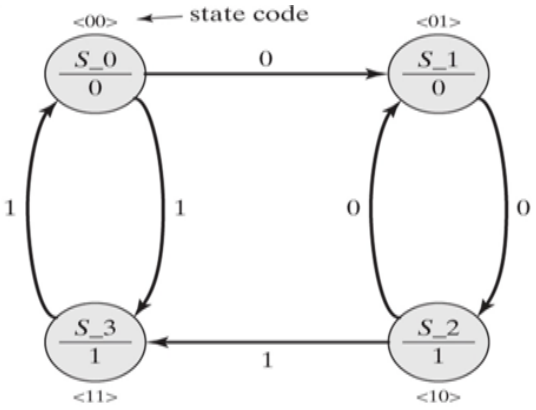
**Time** ⟹

| | | |
|---|---|---|
| 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.




**(i)** Draw the state transition table for the sequential circuit.

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

**(iii)** Draw the resulting circuit.

**(i)** State Transition Table:

| Current State | | Input | Next State | | Output |
|---|---|---|---|---|---|
| F1 | F0 | X | F1+ | F0+ | Y |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | X | X | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | X | X | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

**(ii)** FF and Output Equations:

| Q1 \ Q0X | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | X | 1 |
| 1 | 0 | 1 | 0 | X |

$Q1+ = Q0' X + Q0 X' = Q0 \oplus X$

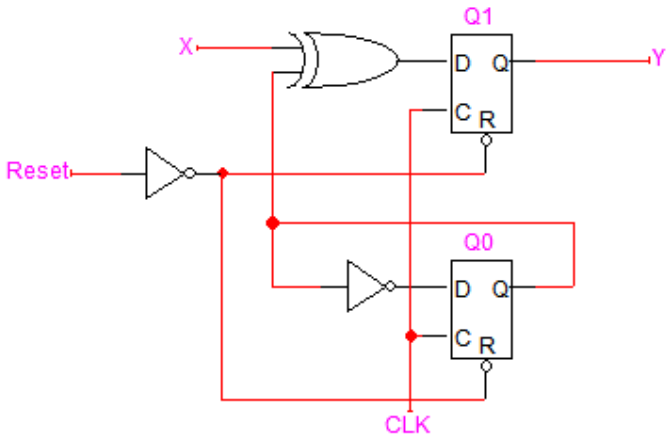
| Q1 \ Q0X | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | X | 0 |
| 1 | 1 | 1 | 0 | X |

Q0+ = Q0’

| Q1 \ Q0 | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |

Y = Q1

**(iii)** Circuit:

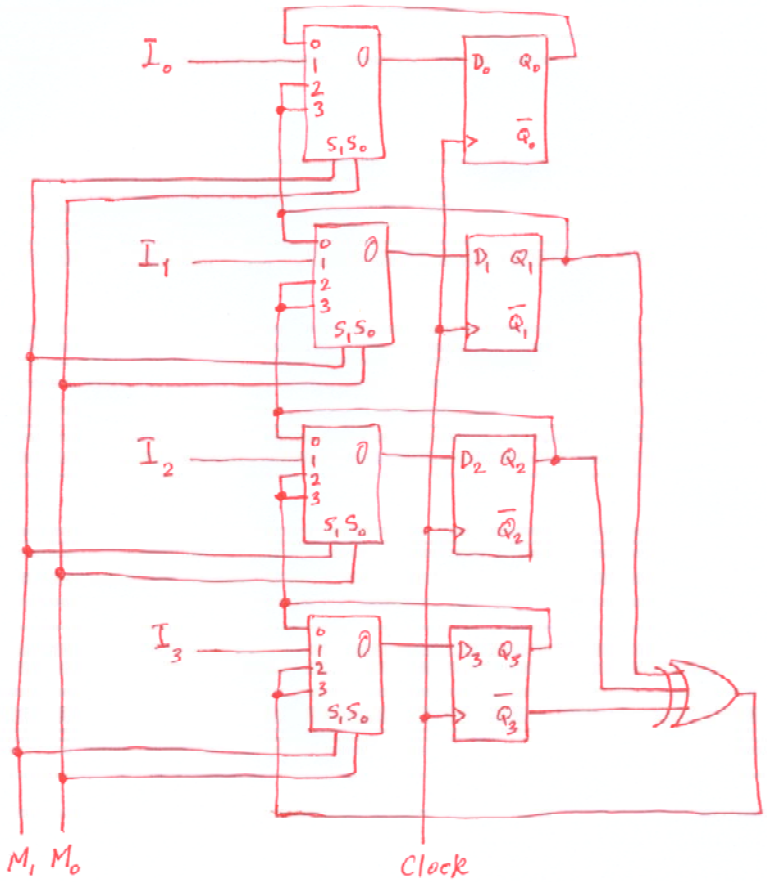


**Question 5.** **(10 Points)**

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 $M_1M_0$ and 4 load inputs $I_3I_2I_1I_0$. Note that D flip-flop outputs include both the state and its complement (i.e., $Q$ and $\bar{Q}$) available for use. The register should operate according to the following table:

| $M_1M_0$ | 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.<br><br>(**Examples**: **1.** register content **before** shifting = 0110,<br>register content **after** shifting = 1011<br><br>**2**. register content **before** shifting = 1001,<br>register content **after** shifting = 0100) |

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




**Note:** It is possible to have any of the following connected to inputs 2 and 3 of the MUX that is connected to $D_3$:

$$\overline{Q_3} \oplus \overline{Q_2} \oplus \overline{Q_1}$$

$$\overline{Q_3} \oplus Q_2 \oplus Q_1$$

$$Q_3 \oplus \overline{Q_2} \oplus Q_1$$

$$Q_3 \oplus Q_2 \oplus \overline{Q_1}$$

**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 $\mathbf{Y_0}$, $\mathbf{Y_1}$, and $\mathbf{Y_2}$. The states are encoded using binary codes **001**, **010**, **011**, **100**, and **101.**

| **PS** | **$(Y_2\ Y1\ Y_0)^{t+1}$** | | **Z** | |
|---|---|---|---|---|
| **$(Y_2\ Y_1\ Y_0)^t$** | **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**.

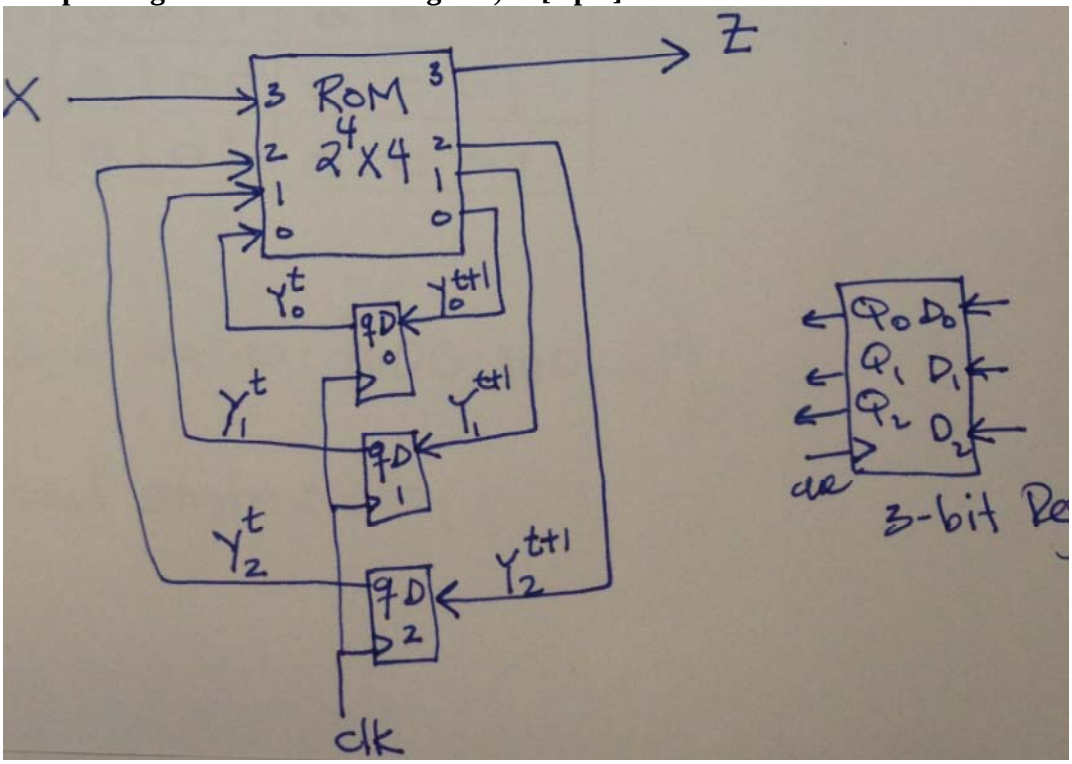
a. What is the minimum size of the register (i.e., number of D flip-flops)? **[1 pt]**

**Three state variables → 3 FFs**

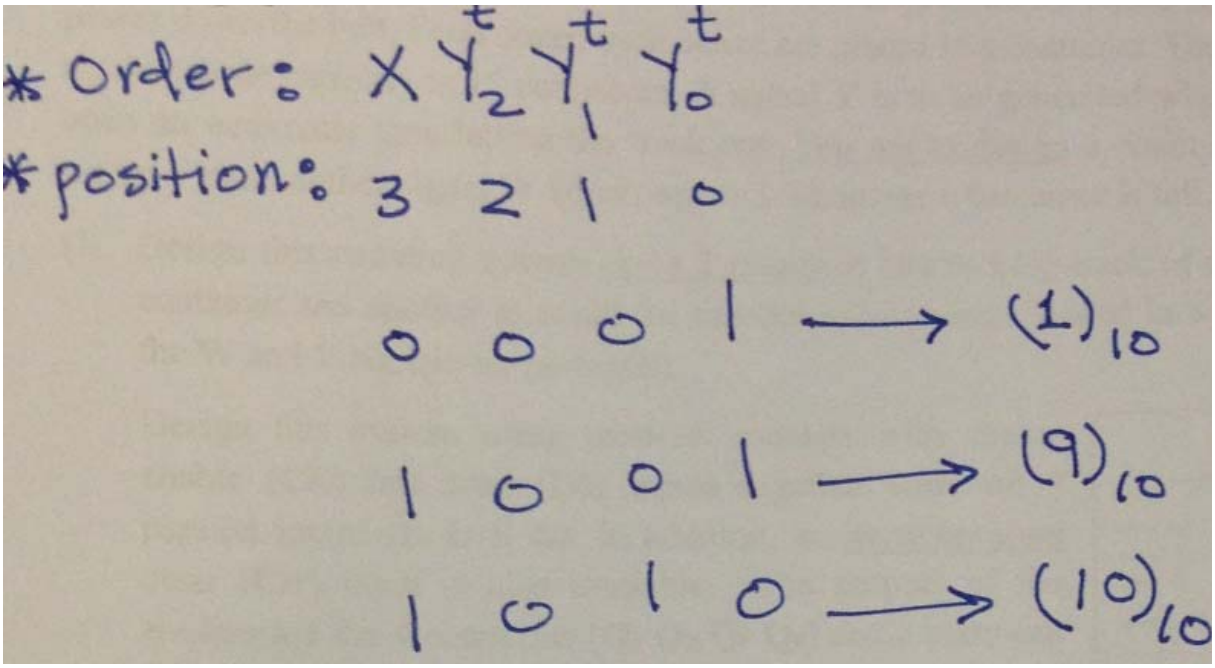
b. What is the minimum size of the ROM (number of memory locations × number of memory bits per location)? **[2 pt]**

# of Locations = $2^4$ =16 word
# of Bits = 4

c. Draw the block diagram for such an implementation. (**Label all components inputs and outputs together with various signals**) **[3 pts]**



d. 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]**




e. 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** | | | |
|---|---|---|---|---|---|---|---|
| **X** | $Y_2^t$ | $Y_1^t$ | $Y_0^t$ | **Z** | $Y_2^{t+1}$ | $Y_1^{t+1}$ | $Y_0^{t+1}$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

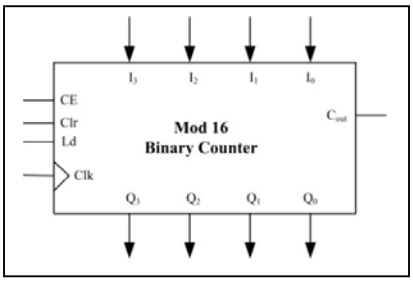
**Note: Unused states are states {000, 110, 111}**

**Question 7.** **(15 Points)**

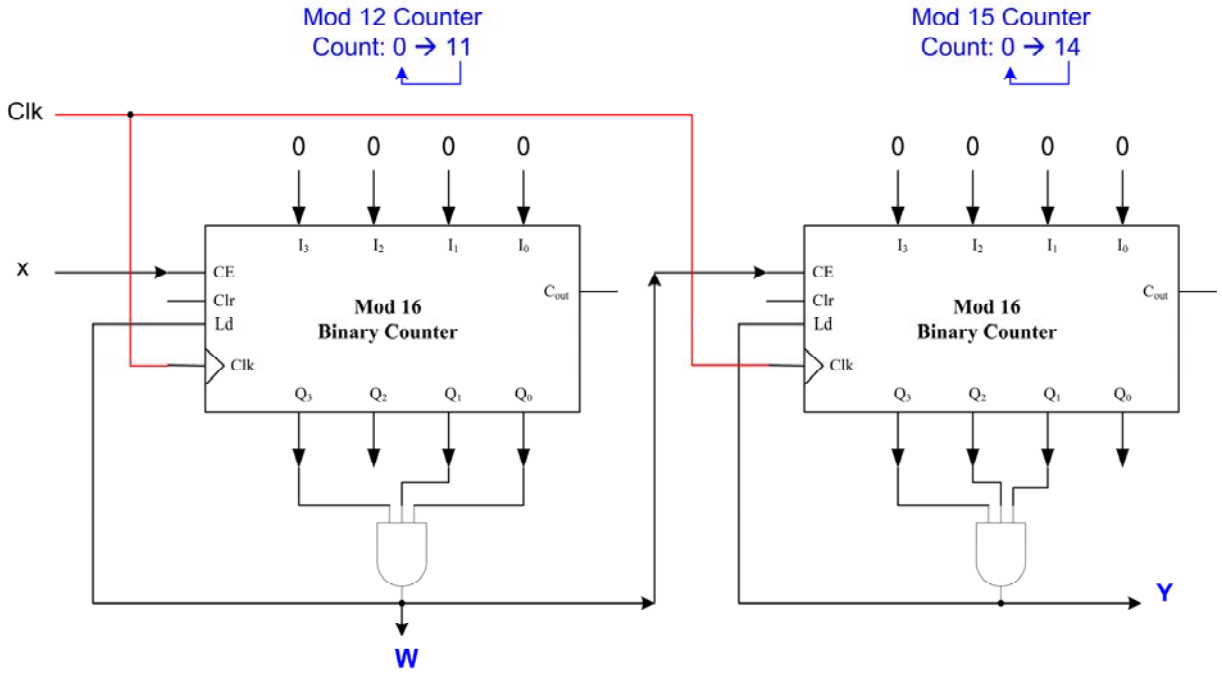
In an assembly line a conveyer belt has a sensor that generates a signal **X** whenever a cartoon box passes down the belt. Each dozen such boxes are placed in a container. The containers are loaded in trucks whose capacity is 15 containers. A signal **Y** is to be generated whenever the truck is full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

(I) 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 (**$I_3$ $I_2$ $I_1$ $I_0$**). In addition, an ***asynchronous*** clear (**Clr**) input is also available. The outputs of the counter are the 4 count bits (**$Q_3$ $Q_2$ $Q_1$ $Q_0$**) and a carry-out signal **$C_{out}$** which equals **1** when $Q_3Q_2Q_1Q_0 = 1111$.




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



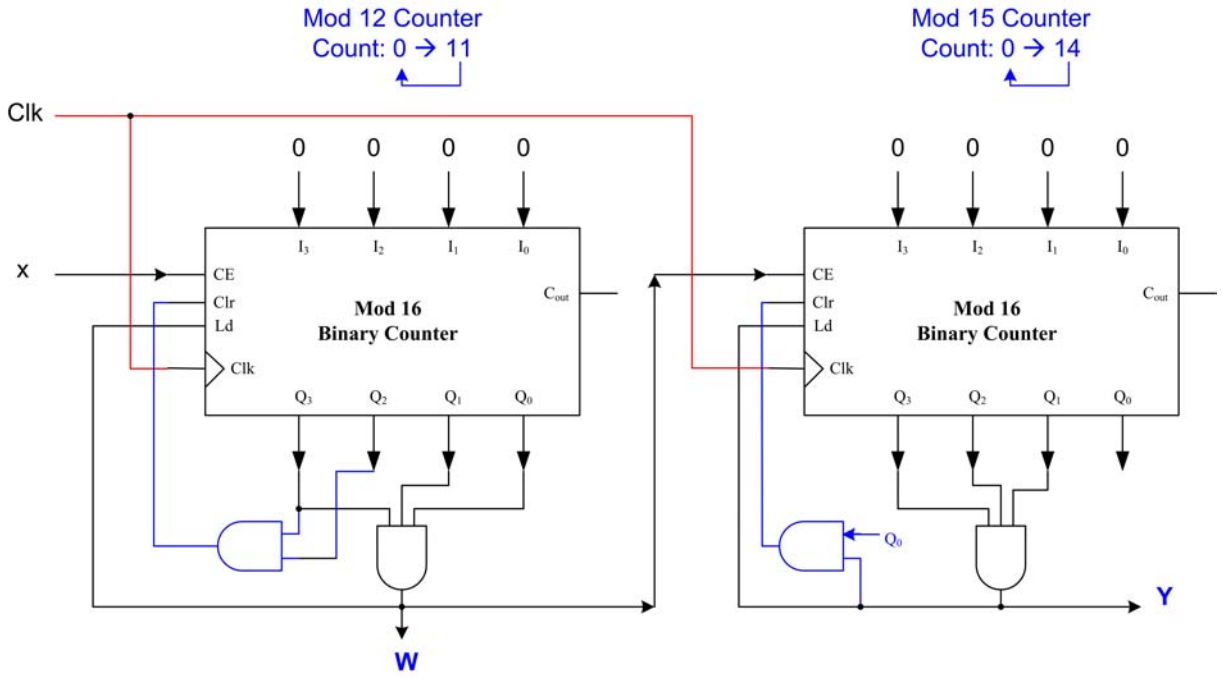
(II) Modify the counting system you designed in part (I) to make it self-resetting, i.e. whenever the system falls in any unused state, the count is automatically reset to zero. (**3 Pts**)

Unused states for the mod 12 counter correspond to counts: $Q_3Q_2Q_1Q_0$ =1100 → 1111 which correspond to a Boolean expression of unused-1 = $Q_3Q_2$

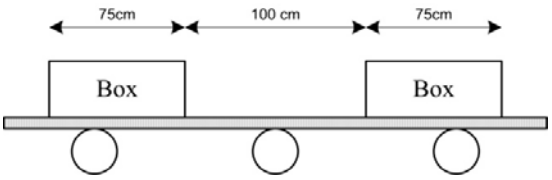
Unused states for the mod 15 counter correspond to the single count: $Q_3Q_2Q_1Q_0$ = 1111 which correspond to a Boolean expression of unused-1 = $Q_3Q_2Q_1Q_0$

Once the unused state is detected, the generated signal can be used to asynchronously clear the counter.

The self-resetting logic is shown in blue.



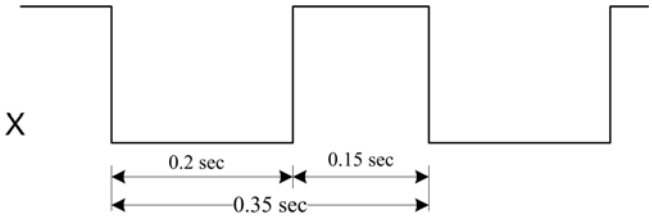
(III) The carton boxes are 75 cm long, and are placed 100 cm apart on the conveyer belt. Signal **X** maintains a value of **1** until the full length of the box passes beyond the sensor. Given that the belt moves at a speed of *5 meters/sec*;




a. Plot the *waveform* of signal **X** (value of X versus time). **(2 Pts)**

Signal X = 1 for a period of $= \frac{0.75\ meters}{5\ meters/sec} = 0.15\ seconds$

Signal X = 0 for a period of $= \frac{1\ meters}{5\ meters/sec} = 0.2\ seconds$




b. Neglecting propagation, setup and hold delays, determine the maximum possible clock frequency. What happens if a higher frequency is used? **(2 Pts )**

No more than one clock pulse may be received during the period when X=1, otherwise one box may count more than once. Thus, the minimum clock period = Period of X=1

$T_{Clk}$(min) = 0.15 sec → Maximum frequency $f_{max}$ = 1/0.15 = 6.66 Hz

c. Neglecting propagation, setup and hold delays, determine the minimum possible clock frequency. What happens if a lower frequency is used? **(2 Pts )**

At least one clock pulse should be received for each X pulse, otherwise a box may pass without being counted at all.

$T_{Clk}$(max) = 0.35 sec → minimum frequency $f_{min}$ = 1/0.35 = 2.86 Hz

**Thus the clock frequency *f* should satisfy:** **2.86. Hz ≤ *f* ≤ 6.66 Hz**