***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 142 (Spring 2015)

## Final Exam

**Tuesday May 19, 2015**

**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
- Mobile phones must be switched off

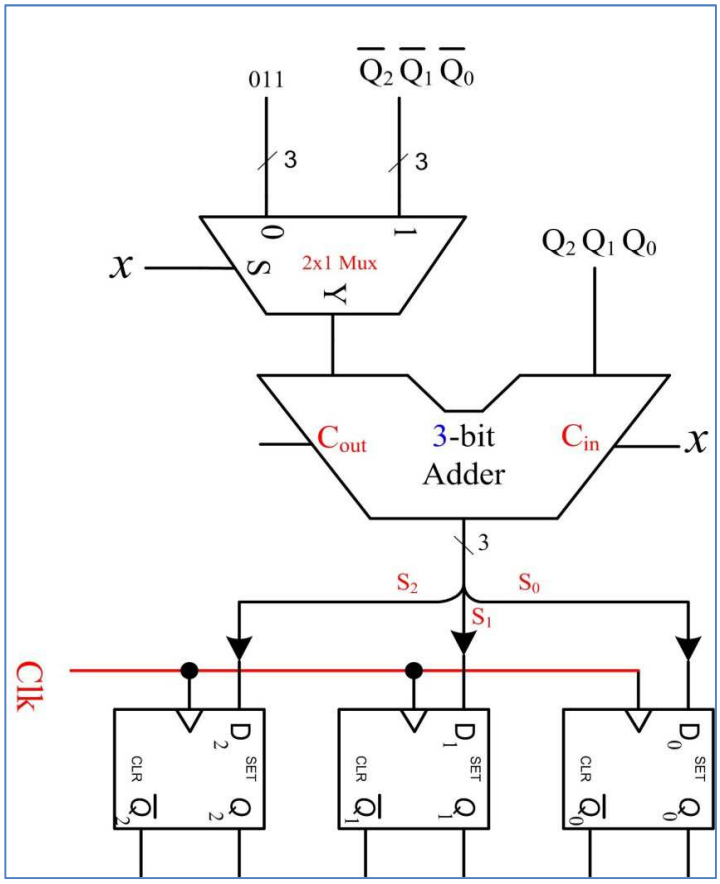
| Question | Maximum Points | Your Points |
|---|---|---|
| 1 | 13 | |
| 2 | 18 | |
| 3 | 10 | |
| 4 | 6 | |
| 5 | 8 | |
| 6 | 12 | |
| Total | 67 | |

**Question 1** **(13 Points)**

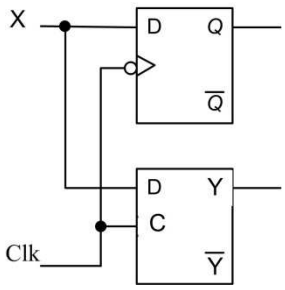
I. The shown sequential circuit has a single input ***X*** and three outputs $\mathbf{Q_2\ Q_1\ Q_0}$.

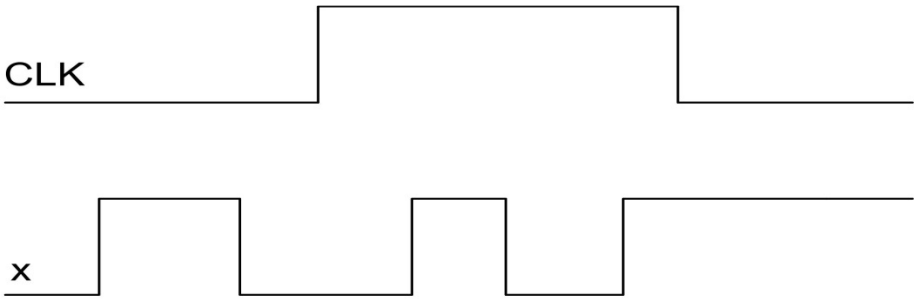
a. Analyze the circuit and derive its state table. (**6 points**)
b. The circuit should have an asynchronous **reset** input, which puts the circuit in initial state **010.** Show how this can be done? (**1 point**)
c. Is the circuit Moore or Mealy? (**1 point**)
d. Briefly describe the function of this circuit. (**1 point**)

*{Note: As shown in the figure, all D-FFs are equipped with asynchronous set (SET) and clear (CLR) inputs}*



II. For the shown D-latch and D-FF, plot the outputs Q and Y for the shown applied signals X and CLK. Assume that the initial state of **both** memory elements to be **1,** and that all delays =0. (**4 points**)




Q

Y

**Question 2** **(18 Points)**

I. The synchronous sequential circuit whose state table is shown has **three inputs**; **reset**, $\mathbf{x_1}$, and $\mathbf{x_0}$ together with a single output **Z**. Assuming state **01** to be the *reset state*, implement this circuit using D-FFs that have asynchronous set and reset inputs and *minimum* logic. Draw the detailed logic diagram of your implementation. (**10 points**)

| **PS** $y_1y_0$ | **NS** $(y_1y_0)^+$ $x_1x_0$ 00 | 01 | 11 | 10 | **Z** $x_1x_0$ 00 | 01 | 11 | 10 |
|---|---|---|---|---|---|---|---|---|
| **01** | 10 | 01 | 01 | 10 | 1 | 1 | 0 | 1 |
| **11** | 11 | 10 | 01 | 11 | 0 | 0 | 1 | 0 |
| **10** | 01 | 10 | 11 | 01 | 0 | 0 | 1 | 1 |

II. If, upon powering the circuit on, the initial state was **00** and assuming that the **reset** signal is **not applied**, what would the next state be in case $x_1x_0$=01 or in case $x_1x_0$=10? (**2 points**)

{*Note:* **PS** = *Present State*, and **NS** = *Next State*}

III. Write a behavioral Verilog model of a 4-bit register which has the function table given below:

(**6 points**)

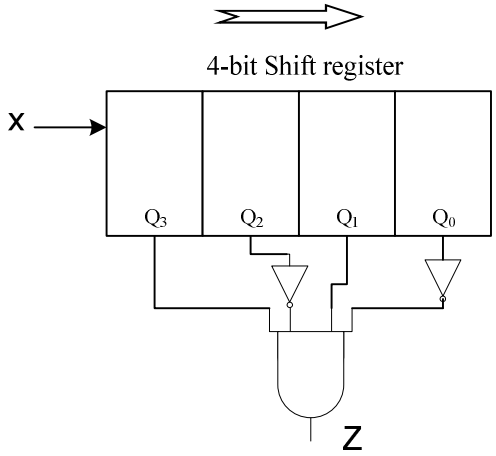
| **Clr_b** | **$S_1$** | **$S_0$** | **Clk** | **Function** |
|---|---|---|---|---|
| 0 | X | X | X | Register Asynchronously cleared |
| 1 | 0 | 0 | ↑ | No Change |
| 1 | 0 | 1 | ↑ | Parallel Load (Q ← x) |
| 1 | 1 | 0 | ↑ | Shift left ( MSB ← LSB) , (Q[0] ← Sin) |
| 1 | 1 | 1 | ↑ | Count Up by 1 |

**Use the following module declaration**:

**module** Q2_Reg **(output reg** [3:0] Q, **input** [3:0] x , **input** Clr_b, Clk, S1, S0, Sin**) ;**

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

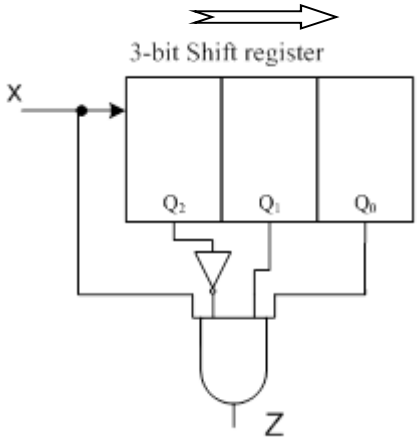
I. A sequence detector has a single input X and a single output Z. The circuit below gives a possible implementation of this detector. Assume that the circuit has a reset input that initializes it to the value $Q_3Q_2Q_1Q_0$=1111.




Answer the following questions:

a. The sequence detected by this circuit is _____
b. The circuit detects {overlapped / non-overlapped} occurrence of the sequence (circle the correct answer)
c. The circuit output is {Mealy / Moore}. (circle the correct answer)

II. The circuit below shows an implementation of another sequence detector. Assume that the circuit has a reset input that initializes it to the value $Q_2Q_1Q_0$=000.




Answer the following questions:

a. The sequence detected by this circuit is _____
b. The circuit detects {overlapped / non-overlapped} occurrence of the sequence (circle the correct answer)
c. The circuit output is {Mealy / Moore}. (circle the correct answer)

III. Suggest a Moore circuit that can detect the two sequences 0011 (two 1's followed by two 0's) and 0001 (1 followed by three 0's). What should be the initial state for your circuit to work properly?

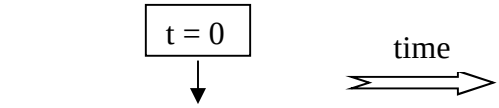
**Question 4.** **(6 Points)**

Draw the state diagram of a **MOORE** sequential circuit that receives two serial input numbers (X and Y) starting with the least significant bits (i.e. $x_0$ $y_0$ then $x_1y_1$ then $x_2y_2$ …$x_ny_n$).
The circuit has two outputs bits $Z_1Z_0$ as shown in the table.

| **Bits of X & Y received thus far** | **$Z_1Z_0$** |
|---|---|
| X = Y | 0 0 |
| X > Y | 1 0 |
| X < Y | 0 1 |

Assume the availability of an asynchronous reset input to reset the machine to a reset state. A sample input/output data is given below.
(**NOTE**: You are *only* required to draw the state diagram, **Nothing MORE**)

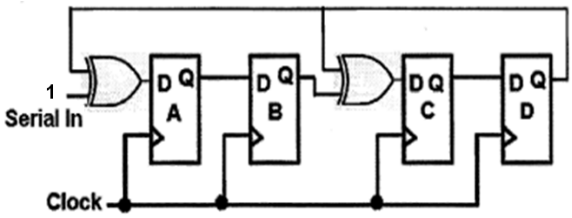
Example:




| XY | 10 | 11 | 00 | 01 | 10 | 11 |
|---|---|---|---|---|---|---|
| output | 00 | 10 | 10 | 10 | 01 | 10 |

**Question 5** **(8 Points)**

I) Consider the shown 4-bit register. If the register contents ABCD = 1001 and the serial input is kept at 1, the register contents after **<u>two</u>** clock pulses will be ABCD = _____. **(2 Points)**




II) Using ***<u>only</u>*** D flip-flop(s) and MUX(s) draw the logic diagram for a 4-bit register with 2 mode selection inputs $M_1M_0$ and 4 load inputs $I_3I_2I_1I_0$. The register should operate according to the following table:

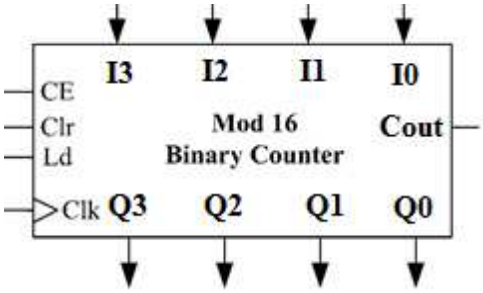
| $M_1M_0$ | **Register operation** |
|---|---|
| 00 | **No change**. |
| 01 | **Parallel Load**: $(Q_3Q_2Q_1Q_0)^+ = I_3I_2I_1I_0$ |
| 10 | **Rotate left**: $(Q_3Q_2Q_1Q_0)^+ = Q_2Q_1Q_0Q_3$<br><br>(**<u>Examples</u>**: **1.** register content **<u>before</u>** rotation = 0110,<br>register content **<u>after</u>** rotation = 1100<br><br>**2**. register content **<u>before</u>** rotation = 1110,<br>register content **<u>after</u>** rotation = 1101) |
| 11 | **Reverse the order of bits in the register**: $(Q_3Q_2Q_1Q_0)^+ = Q_0Q_1Q_2Q_3$<br><br>(**<u>Examples</u>**: **1.** register content **<u>before</u>** reversing = 1000,<br>register content **<u>after</u>** reversing = 0001<br><br>**2**. register content **<u>before</u>** reversing = 1101,<br>register content **<u>after</u>** reversing = 1011) |

**You must clearly label the D flip-flop(s) and MUX(s) inputs and outputs. (6 Points)**

**Question 6** **(12 Points)**

I) It is required to design a **mod 16** up counter that has the following control inputs :

- **Clr** (synchronous clear),
- **Ld** (parallel load), together with its associated inputs $I_3$, $I_2$, $I_1$, $I_0$.
- **CE** (Count_Enable)




Assume that Clr has the highest priority followed by Ld, as illustrated in the table below. The counter produces an output signal ($\mathbf{C_{out}}$) which equals **1** when its output equals 15. Design the counter using D-FFs and minimal logic gates. **(6 Points)**

| Clr | Ld | CE | **Counter Next Content after the clock pulse** $(Q_3Q_2Q_1Q_0)^+$ |
|---|---|---|---|
| 1 | X | X | 0 0 0 0 (clear) |
| 0 | 1 | X | I3 I2 I1 I0 (load) |
| 0 | 0 | 1 | $(Q_3Q_2Q_1Q_0)+1$ (increment up by 1) |
| 0 | 0 | 0 | $Q_3Q_2Q_1Q_0$ (no change) |

II) Given that the clock frequency of the mod 16 up counter is 1 MHZ, what is the clock frequency of the Q3 output of the counter? **(2 Points)**

III) Using any number of the above **mod 16** up counter and other needed logic gates design a **mod 4096**[1] up counter, which has the same control inputs: Clr, LD, and CE. **(4 Points)**

[1] $4096 = 2^{12}$

# Verilog Operators

| | |
|---|---|
| { } | concatenation |
| + - * / ** | arithmetic |
| % | modulus |
| > >= < <= | relational |
| ! | logical NOT |
| && | logical AND |
| \|\| | logical OR |
| == | logical equality |
| != | logical inequality |
| === | case equality |
| !== | case inequality |
| ? : | conditional |

| | |
|---|---|
| ~ | bit-wise NOT |
| & | bit-wise AND |
| \| | bit-wise OR |
| ^ | bit-wise XOR |
| ^~ ~^ | bit-wise XNOR |
| & | reduction AND |
| \| | reduction OR |
| ~& | reduction NAND |
| ~\| | reduction NOR |
| ^ | reduction XOR |
| ~^ ^~ | reduction XNOR |
| << | shift left |
| >> | shift right |