***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 112 (Spring 2012)**

**Final Exam**

**Monday May 28, 2012**

**7:00 p.m. – 9:30 p.m.**

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

**Name: \_KEY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ID: \_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_**

**Notes:**

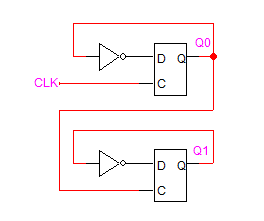
* Do not open the exam book until instructed
* Calculators are not allowed (basic, advanced, cell phones, etc.)
* Answer all questions
* All steps must be shown
* Any assumptions made must be clearly stated

|  |  |  |
| --- | --- | --- |
| **Question** | **Maximum Points** | **Your Points** |
| **1** | **21** |  |
| **2** | **8** |  |
| **3** | **10** |  |
| **4** | **15** |  |
| **5** | **29** |  |
| **6** | **17** |  |
| **Total** | **100** |  |

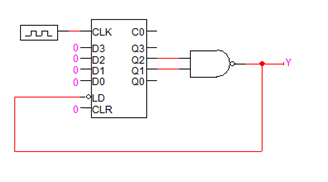
**Question 1. (21 Points)**

Answer the following questions by either **filling** the required spaces or **underlining** the correct answers:

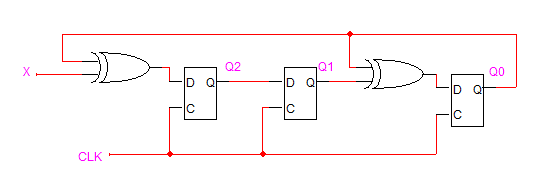
1. Given a synchronous sequential circuit with 20 states, the minimum number of flip-flops required to implement the circuit is 5 flip flops and the number of unused states is 12 states. **(3 points)**
2. Given a synchronous sequential circuit with 3 inputs, 3 flip-flops and 2 outputs, to implement the output and next state equations using a ROM requires a ROM with 64 locations storing 5 bits each. **(3 points)**
3. The following circuits implements a 2-bit (synchronous, asynchronous) down counter. **(1 point)**



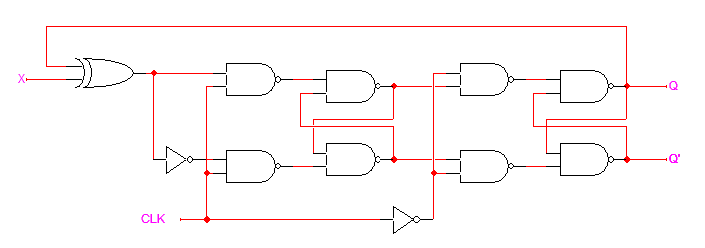
1. Given a 3-bit synchronous counter with outputs Q2, Q1 and Q0, assuming that the counter clock has a frequency of 16 MHZ, then the frequency of Q0 is 8 MHZ and the frequency of Q2 is 2 MHZ. **(3 points)**
2. For the circuit given below, assuming that its clock has a frequency of 14 MHZ, then the frequency of Y is 2 MHZ. **(2 points)**



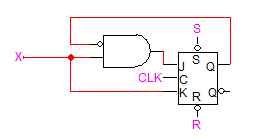
1. For the 3-bit register given below, given that Q2=0, Q1=1, Q0=1, and X=0, the content of the register after one clock cycle will be Q2=1, Q1=0, Q0=0. (**3 points)**



1. Given that F1=A’B’+AC, F1’=A’B+AC’ and F2=A’B+BC’+AB’, F2’=A’B’+ABC, to implement F1 and F2 using a PLA, the minimum number of AND gates needed is 3 AND gates (implementing F1 and F2’). **(2 points)**
2. The following circuit implements a (rising, falling) -edge triggered (D flip flop, T flip flop). **(3 points)**



1. Given the circuit below having a JK flip flop and assuming that the current state Q=0 and X=1, the next state will be 1. **(1 point)**

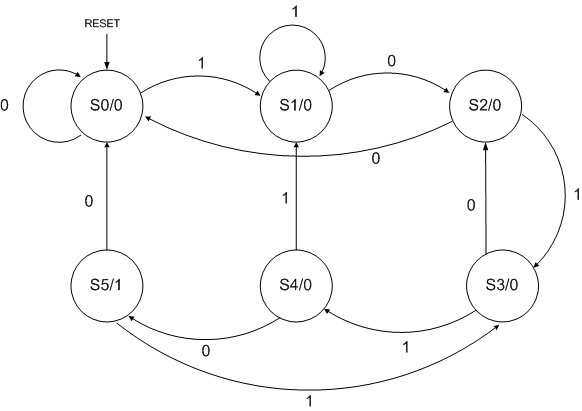


**Question 2. (8 Points)**

# It is required to design a sequence detector that detects overlapped occurrences of the sequence **10110**. The circuit receives a serial input **X** and produces a serial output **Z**. The output Z will be 1 when the circuit detects the sequence **10110**. Assume the existence of a reset input to reset the machine to a reset state. Draw the state diagram of the circuit assuming a **Moore** model. *You are not required to derive the equations and the circuit*. The following is an example of some input and output streams:

Example:

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



**Question 3. (10 Points)**

It is required to design a sequential circuit that has two inputs X and Y and one output Z. The output equals “1” when it receives **two consecutive bits where x=y, followed by one bit where x≠y, followed by two consecutive bits where x=y**. After receiving the required sequence, the circuit starts detecting these occurrences over again without overlap. The circuit has an additional reset input R which resets the circuit into the initial state. Draw the state diagram of the circuit assuming a **Mealy** model. *You are not required to derive the equations and the circuit*. The following is an example of some input and output streams:

# 

Example:

x≠y

x=y

x=y

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



**Question 4. (15 Points)**

Consider the following sequential circuit:



1. Derive simplified Boolean expressions for the ***DA*** and ***DB*** inputs of the flip flops and the external output ***Z***. **(6 points)**

Considering the decoder and using K-maps we get the following:

1. Is the circuit type *Mealy* or *Moore*? Justify your answer. **(1 point)**

Since ⇒ ***Mealy***

1. Provide a state table showing {present state and external inputs} and {next state and external output}. **(8 points)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *QA* | *QB* | *X* | *Y* | *QA+* | *QB+* | *Z* |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

**Question 5. (29 Points)**

Consider the following state table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *QA* | *QB* | *X* | *QA+* | *QB+* | *Z* |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |

1. Draw the state diagram corresponding to the given state table. **(4 points)**



1. If ***falling edge-triggered*** D-FF(s) and simple gates (i.e., AND, OR, NOT gates) are to be used to implement the given state table, provide the following:
   1. The simplified Boolean expressions of all FF inputs. **(4 points)**

Using K-maps we get the following:

* 1. The simplified Boolean expression of the output. **(2 points)**
  2. The logic diagram of the circuit. ***Label your circuit properly***. **(2 points)**



1. If the same previous state table is implemented using a single ROM and a single register, provide the following:
   1. The size of the required ROM, and its total capacity in bits. **(3 points)**

**8×3 = 24 bits**

* 1. The complete ROM table. **(2 points)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *A*2 | *A*1 | *A*0 | *D2* | *D1* | *D0* |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |

* 1. The block diagram of this implementation (**You must CLEARLY LABEL each signal and each component inputs and outputs**). **(3 points)**



1. Consider the same previous state table which is repeated here for your convenience. Assume that the initial state of the circuit implementation of the given state table is (*QAQB* = 01), draw the waveforms of *QA*, *QB*, and *Z* in response to the shown applied input *X*. ***Assume that the circuit uses falling edge-triggered D-FF(s)***. **(9 points)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *QA* | *QB* | *X* | *QA+* | *QB+* | *Z* |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |



**Question 6. (17 Points)**

A mod-16 *binary* counter consists of 4 D-Flip Flops with inputs (D0, D1, D2, D3) and *corresponding* outputs (Q0, Q1, Q2, Q3). The input equations of the counter are given by:

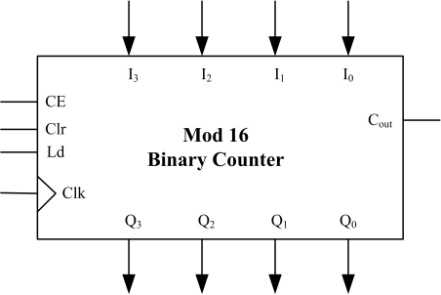
**, , ,** *and*

The counter also has a carry-out signal ***Cout*** =

1. Show, *through drawing a logic diagram*, how can the above design be modified to include:
2. A count enable input “*CE*” such that if *CE* = 0 counting is disabled (i.e. count does not change with incoming clock pulses) while if *CE* = 1 counting is enabled. **(2 points)**
3. Show how can the design in (a) be modified to include a***synchronous***clear input “CLR”, which clears the counter on the next active clock edge when CLR=1.  **(2 points)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Counter Function Table** | | | | |
| **LD** | **CE** | **CLR** | **Clk** | **Function** |
| 0 | 0 | 0 | X | Counting Disabled  (No change in current count) |
| 0 | 1 | 0 | ↑ | Counting enabled |
| 0 | X | 1 | ↑ | Current count 🡨 0000 |
| 1 | X | X | ↑ | Current count 🡨 I3 I2 I1 I0 |

1. Show how can the design in (b) be modified to include a load input “LD” with additional 4 Data-In inputs I3, I2, I1 and I0 such that the counter operation is in accordance with the shown function table. **(2 points)**
2. Using the mod-16 counter in (c), show how to turn this counter into a **mod-10 counter** with the same input control signals (CE, CLR, and LD) and a *proper* carry-out signal **Cout**. You may show your logic modifications using the symbol shown below. **(3 points)**



1. Using the mod-16 counter in (c), show how to turn this counter into a **mod-6 counter** with the same input control signals (CE, CLR, and LD) and a *proper* carry-out signal **Cout**. You may show your logic modifications using the symbol shown above. **(3 points)**

1. Given a *1 Hz clock*, show how the counters designed in (II) and (III) can be used to build a stop watch that counts seconds (0 to 59). The counter should have a **Cout** signal that may be used for a minutes counter. The outputs of the mod-10 and mod 6 counters will drive two 7-segment displays to be able to read the number of passed seconds in decimal (00 to 59).  **(5 points)**

