***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 141 (Fall 2014)**

**Final Exam**

**Wednesday Dec. 31, 2014**

**7:00 p.m. – 10:00 p.m.**

**Time: 180 minutes, Total Pages: 11**

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

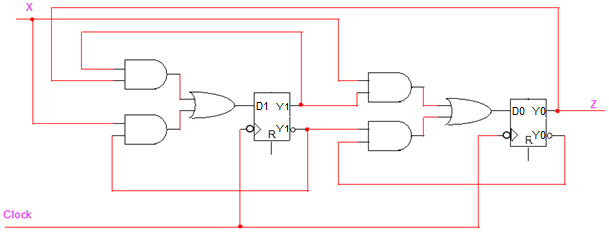
**Notes:**

* 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** | **15** |  |
| **3** | **10** |  |
| **4** | **6** |  |
| **5** | **8** |  |
| **6** | **15** |  |
| **Total** | **67** |  |

**Question 1. (13 Points)**

1. Given the sequential circuit below with a single input X, a single output Z and two D flip-flops:



1. Is this a rising-edge or falling-edge triggered circuit? **(1 Point)**
2. Is this a Mealy or Moore circuit? **(1 Point)**
3. Obtain the state table of the circuit. **(5 Points)**
4. Given the state table below for a sequential circuit with a single input x and a single output z,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Current State  (q1q0)t | Next State  (q1q0)t+1 | | Output  z | |
| x=0 | x=1 | x=0 | x=1 |
| 00 | 00 | 01 | 0 | 0 |
| 01 | 00 | 10 | 0 | 1 |
| 10 | 00 | 10 | 0 | 1 |
| 11 | 00 | 01 | 0 | 0 |

1. Draw the state diagram of the circuit. **(2 Points)**
2. Complete the timing diagram below for the values of the two flip flops q1q0 and the output z assuming falling edge triggered flip flops are used and starting from initial state q1q0=00. **(4 Points)**

Clock

x

q1

q0

z

**Question 2. (15 Points)**

1. Draw a circuit implementing the following state table minimizing the number of used gates. **(7 Points)**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Current State (AB)*** | ***Input (x)*** | ***Next State (DADB)*** | ***Output (z)*** |
| 00 | 0 | 00 | 0 |
| 00 | 1 | 01 | 0 |
| 01 | 0 | 00 | 1 |
| 01 | 1 | 11 | 0 |
| 10 | 0 | 00 | 1 |
| 10 | 1 | 10 | 0 |
| 11 | 0 | 10 | 1 |
| 11 | 1 | 11 | 0 |

1. Choose the correct answer for each of the following: **(2 Points)**
   1. A PLA is made of:
      1. Fixed AND array, fixed OR array
      2. Fixed AND array, programmable OR array
      3. Programmable AND array, Fixed OR array
      4. Programmable AND array, programmable OR array
   2. A PAL is made of:
      1. Fixed AND array, fixed OR array
      2. Fixed AND array, programmable OR array
      3. Programmable AND array, Fixed OR array
      4. Programmable AND array, programmable OR array
2. Optimize a solution to program the following programmable logic to implement the following functions: **(6 Points)**





**Question 3. (10 Points)**

A **Moore**  *Transition Detector* synchronous sequential circuit has a single input *x* and a single output z*.* The input data is applied serially at the input x and the circuit produces a 1 in the output z whenever a transition from 0 to 1 or from 1 to 0 are detected at the applied input data. Draw the state diagram of this circuit. Assume the existence 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**)

t = 0

time

Example:

|  |  |  |
| --- | --- | --- |
| Input | ***x*** | 0 1 1 0 0 1 0 0 0 1 1 1 1 1 |
| Output | ***z*** | 0 0 1 0 1 0 1 1 0 0 1 0 0 0 |

**Question 4. (6 Points)**

It is required to design a synchronous sequential circuit that receives two serial inputs **x** and **y** and produces a serial output **z** that computes the equation **z=2x-y**. Draw the state diagram of this circuit assuming a **Mealy** model. Assume the existence of an asynchronous reset input to reset the machine to a reset state. Two samples of input/output data are given below.

(**NOTE**: You are *only* required to draw the state diagram **Nothing MORE**)

t = 0

time

Examples:

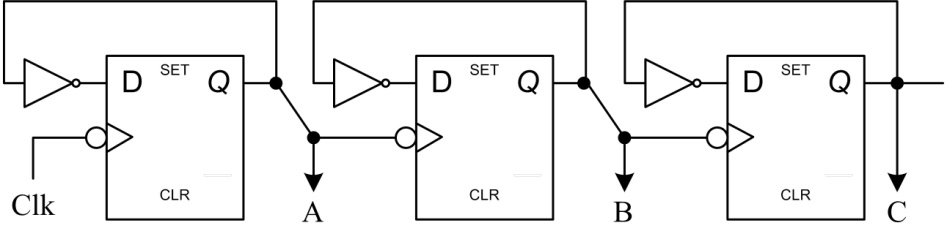
|  |  |  |
| --- | --- | --- |
| Input | ***x*** | 0 0 1 0 0 |
| ***y*** | 0 1 1 0 0 |
| Output | ***z*** | 0 1 0 0 0 |

t = 0

time

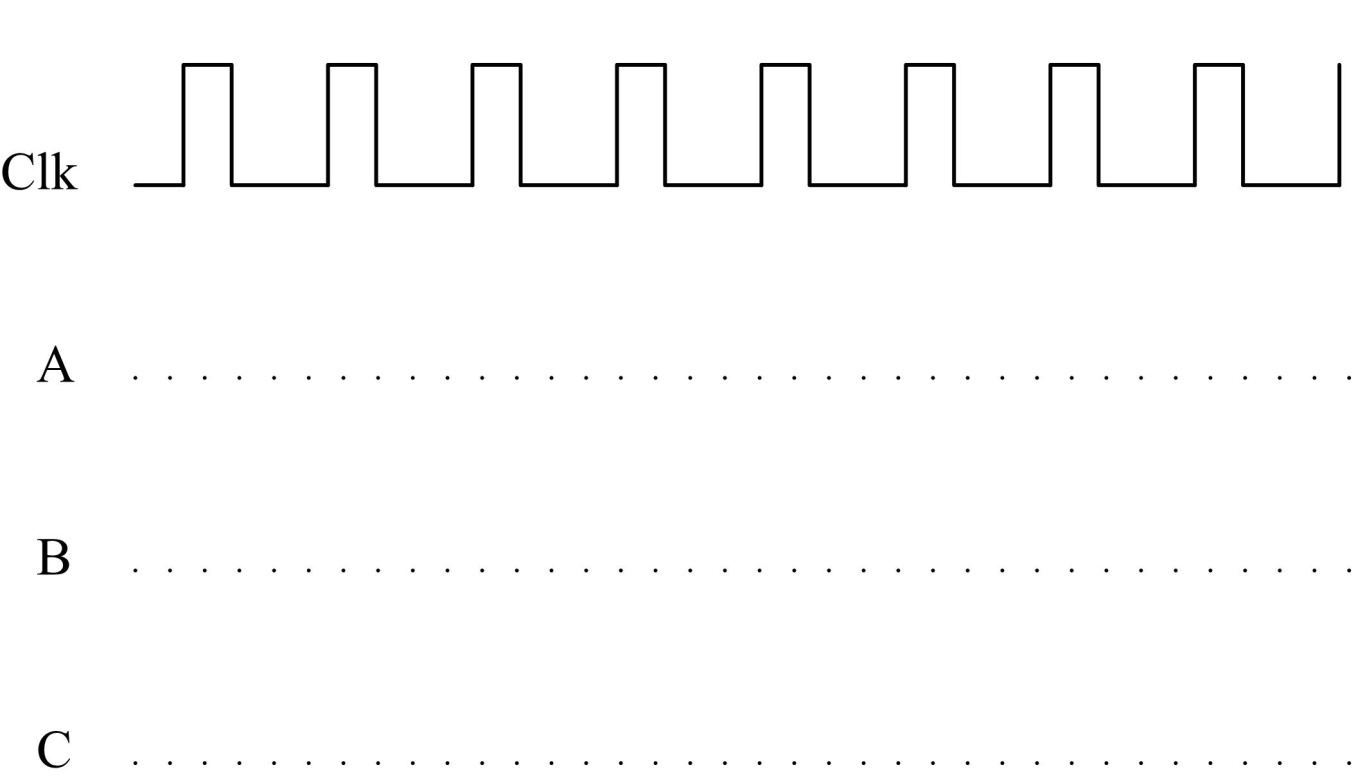
|  |  |  |
| --- | --- | --- |
| Input | ***x*** | 1 0 1 1 0 |
| ***y*** | 1 1 0 1 0 |
| Output | ***z*** | 1 1 1 1 0 |

**Question 5 (8 Points)**

****

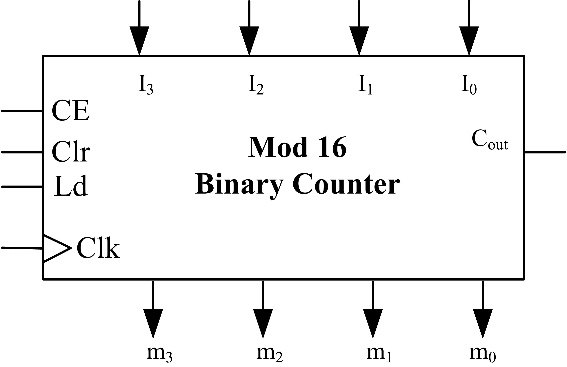
a. The sequential circuit above is clocked \_\_\_\_\_\_\_\_\_ (synchronously / asynchronously) **(1 Pt)**

b. Draw the waveforms of signals A, B, C in response to the shown Clk signal assuming initial ABC value of **000**. **(4 Points)**



c. Assuming a negligible setup and hold times, an *inverter* delay of **1 ns** and a delay from the *clock active edge till the new flip flop output* appearing of **4 ns**, what is the maximum clock frequency at which the above circuit can operate? **(3 Points)**

**Question 6**  **(15 Points)**

It is required to design a digital calendar that counts days and months of the year. Given a clock signal with frequency **1 pulse/24 hours,** you are required to design the following as part of this system:

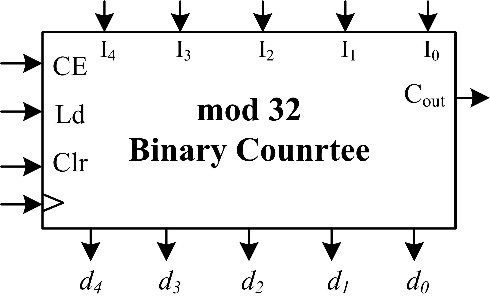
1. Design a **mod 12** *months* counter to count the months of the year (count 0 🡪 January[[1]](#footnote-1) up to Count 11 🡪 December). Use a **mod 16** counter to build this *months* counter. Assume the mod 16 counter to have the following control inputs :

* **CE** “Count\_Enable,
* **Clr** (synchronous clear), and
* **Ld** (parallel load), together with its associated inputs I0, I1, I2, I3.

The counter should produce an output signal (**Cout**) which equals **1** *only* during the last month of the year. **(3 Points)**

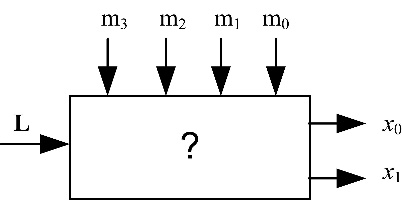
1. Design a *days* counter to count the days of the month. The counter has two input signals *x1x0* which indicate the number of days in the current month as shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| ***x1 x0*** | **# of Days** | **Months** | **Month Count** |
| 0 0 | 31 | January, March, May, July, August, October, December | 0, 2, 4, 6, 7, 9, 11 |
| 0 1 | 30 | April, June, September, November | 3, 5, 8, 10 |
| 1 0 | 29 | February in leap years | 1 |
| 1 1 | 28 | February in ordinary years | 1 |



The counter should produce an output signal (**Cmonth**) which equals **1** *only* during the last day of the month. Design this *days* counter using a **mod 32** binary counter having the same control inputs as the counter in part (a).

**(5 Points)**

**c.** Given the output (*m3 m2 m1 m0*) of the *months’* counter of part (a) and an input signal **L** that equals **1** only throughout leap years, derive the logic circuits which generate the signals ***x1*** and ***x0***used by the *days* counter. **(3 Points)**

**(*Hint:*** *Use a 4*x*16 decoder and any other needed parts***)**

**d.** Given the clock signal of frequency **1 pulse/24 hours,** show how to assemble the parts designed in (a), (b) and (c) to build a *synchronous* counter whichgives the current month of the year and the day of that month. Assume that the counter is reset to 0 at the beginning of each year. *This assembled design should produce an output signal* (***Cyear***) *which equals* ***1*** *only during the last day of the year.* **(4 Points)**

Note: Use black boxes for the parts designed in (a), (b) and (c) showing only the interface input and output signals, various signal connections and the logic to generate ***Cyear***

1. The Gregorian year months are: January, February, March, April, May, June, July, August, September, October, November and December respectively. [↑](#footnote-ref-1)