Written Exam
IE1205 Digital Design
2019-04-15, 08:00-12:00

Solution

Part 1 (4p)

1. (3p)
Eight parties qualified for Swedish parliament in the last election with varying number of seats in the parliament as follows:

Party	**Seats in Parliament**
(S) Social Democrats	100
(M) Moderates	70
(SD) Sweden Democrats	62
(C) Centre Party	31
(V) Left Party	28
(KD) Christian Democrats	22
(L) Liberal	20
(MP) Environment Party	16
Total	**349**

To form a government, 175 seats are required. Formulate this as a Boolean algebra problem. Write a truth table, where each row represents a valid constellation that has a minimum of 175 votes. Factor in the following positions of the parties:

a) All parties except M and KD will not form a constellation in which SD is present
b) All parties except S, V and MP will not form a constellation in which V is present

Hints:

a) There will be eight columns corresponding to the eight parties and one output
b) Organize the columns in the order of number of seats in the parliament from left to right. Each row will represent a valid constellation.
c) In each row, 1 in a column represents inclusion of that particular party. `0` represents not inclusion and "–" represents do not care. Don't care is used as soon as the bigger parties to the left have added up to 175.
d) All constellations will require, presence of at least one of the top three parties (S, M or SD)

Solution:

S	M	SD	C	V	KD	L	MP	Out
1	1	0	1	0	-	-	-	1
1	1	0	0	0	1	-	-	1
1	1	0	0	0	0	1	-	1
1	1	0	0	0	0	0	1	1
1	0	0	1	0	1	1	1	1

2. (1p)
Simplify the following expression using Boolean identities and theorems.

A B (A' C D + A E + E G)

Show all the intermediate steps.

Solution:

ABE+ABEG = ABE

Part 2 (8p)

3. (3p)

Three multiple choice questions on CMOS

(1) The switching power is most effectively reduced by – select one

A. Decreasing Voltage (VDD)
B. Increasing VDD
C. Reducing the amount of capacitance that toggles (charged to 1 or discharged to 0) per unit time
D. Increasing the amount of capacitance that toggles (charged to 1 or discharged to 0) per unit time

(2) Which of the following statement is true for crowbar current

A. A Slow transition between Logic 1 and Logic 0 will increase the crowbar current
B. A Slow transition between Logic 1 and Logic 0 will decrease the crowbar current
C. A Slow transition between Logic 1 and Logic 0 will not have an effect on the crowbar current

(3) When NMOS transmits one, the maximum voltage that it can transmit is

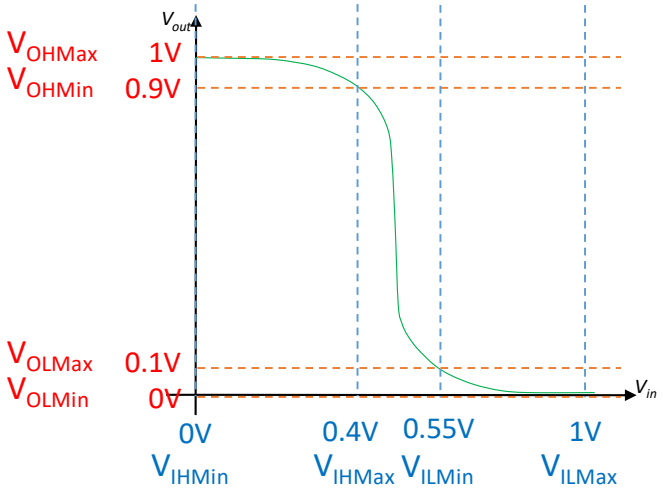
A. VDD
B. VDD – Vt
C. 0V
D. 0V + Vt

4. (2p)

A CMOS inverter has the following specifications:

V_{OHMax} = 1 V
V_{OHMin} = 0.9 V
V_{OLMin} = 0 V
V_{OLMax} = 0.1 V
V_{IHMax} = 0.4 V
V_{IHMin} = 0 V
V_{ILMin} = 0.55 V
V_{ILMax} = 1 V

Label these values on the CMOS inverter transfer curve (Vin-Vout) as shown below. Please use the attached answer sheet to write your answer.



5. (3p)

A four-variable logic function that is equal to 1 if any three or all four of its variables are equal to 1 is called a majority function. Design a SOP circuit that implements this majority function. Assuming the inputs are "a, b, c and d", output is "f". You need to:

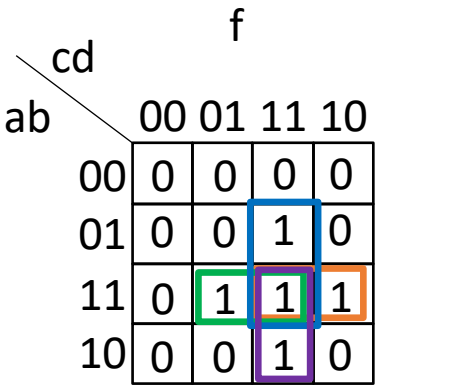
a) (1p) Write the truth table.
b) (1p) Find the minimum SOP form of output "f" using Karnaugh map. Expression is sufficient, no need to draw the circuit.
c) (1p) Implement the majority function using only NAND gate. You can use 2-input, 3-input and 4-input NAND gate if needed.

Solution:

a)

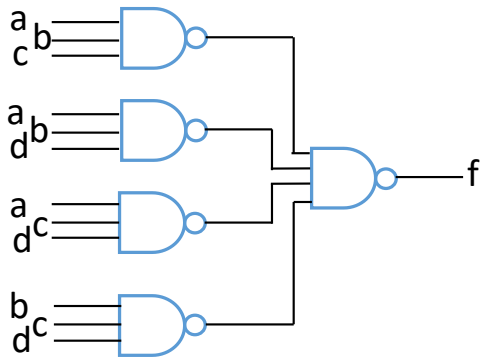
a	b	c	d	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

b) The expression for f is:



f=abd+abc+acd+bcd

c)



Part 3 (12p)

6. (3p)

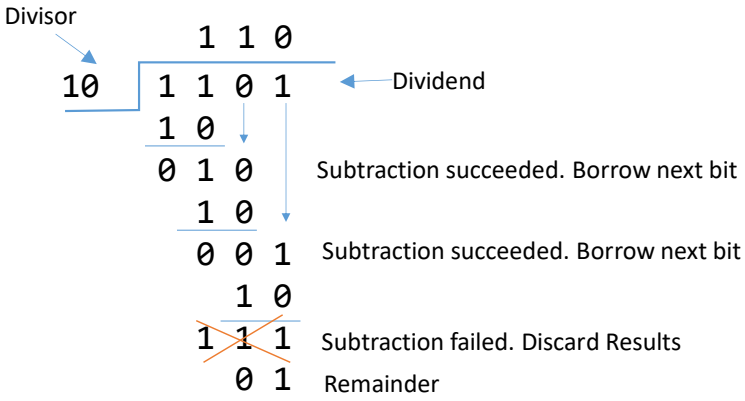
a. (1p) Multiply 4-bit two's complement number 7 and -6. Express the result in 8-bit 2's complement number.

Solution:

```
          0 1 1 1      7
        * 1 0 1 0     -6
-------------------------
0 0 0 0 0 0 0 0      0*7 = 0
0 0 0 0 1 1 1 x      2*7 = 14
0 0 0 0 0 0 x x      0*7 = 0
1 1 0 0 1 0 0 0     -8*7 = -56
-------------------------
1 1 0 1 0 1 1 0          = -42
```

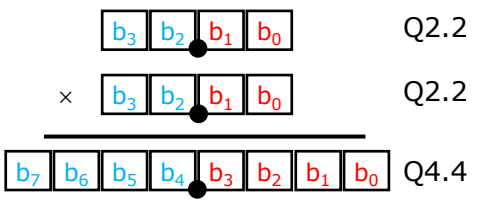
b. (1p) Divide 4-bit unsigned number "13" by 2-bit unsigned number "2". Show each step of the division.

Solution:



c. (1p) Multiply -1.5 with 0.75 as two two's complement fixed point binary numbers in Q2.2 format. Result format should be Q4.4

Solution:




```
          1 0 1 0      −1.50
        × 0 0 1 1    × 0.75
-------------------------------------------------
1 1 1 1 1 0 1 0      0.25 X -1.50 = -0.3750 +
1 1 1 1 0 1 0 X      0.5  X -1.50 = -0.7500 +
0 0 0 0 0 0 X X      0    X -1.50 =  0.0000 +
0 0 0 0 0 X X X      0    X -1.50 =  0.0000
-------------------------------------------------
1 1 1 0 1 1 1 0                   = -1.1250
```

7. (3p)

In a ternary number system, there are three digits: 0, 1, and 2. Table below defines a ternary half-adder. Design a circuit that implements this half-adder using binary-encoded signals, such that two bits are used for each ternary digit. Let A=a_1a_0, B=b_1b_0, and Sum = s_1s_0; note that Carry is just a binary signal. Use the following encoding: 00=$(0)_3$, 01=$(1)_3$, and 10=$(2)_3$. (Hint: You should use don't-car for outputs if any of the input (A or B) is not a valid ternary number.)

AB	Carry	Sum
00	0	0
01	0	1

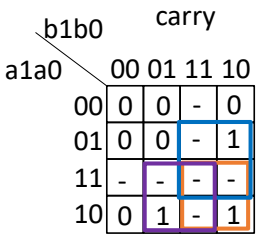
02	0	2
10	0	1
11	0	2
12	1	0
20	0	2
21	1	0
22	1	1

You need to:

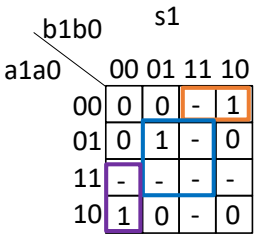
a. (1p) Write the truth table.
b. (2p) Use Karnaugh map to simplify carry and sum.

Solution:

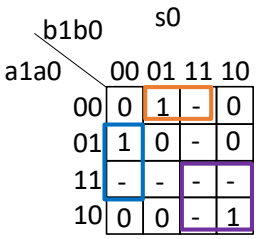
a1a0	b1b0	carry	s1	s0
00	00	0	0	0
00	01	0	0	1
00	10	0	1	0
00	11	-	-	-
01	00	0	0	1
01	01	0	1	0
01	10	1	0	0
01	11	-	-	-
10	00	0	1	0
10	01	1	0	0
10	10	1	0	1
10	11	-	-	-
11	00	-	-	-
11	01	-	-	-
11	10	-	-	-
11	11	-	-	-



$carry = a_1b_0 + a_1b_1 + a_0b_1$



$s_1 = a_1b_1'b_0' + a_0b_0 + a_1'a_0'b_1$



$s_0 = a_0b_1'b_0' + a_1b_1 + a_1'a_0'b_0$

8. (4p)
Implement two-bit **2's complement** comparator using 2-to-1 MUXes. This comparator produces

$G = 1, L = 0$ if $(a_1\ a_0) > (b_1\ b_0)$
$G = 0, L = 1$ if $(a_1\ a_0) < (b_1\ b_0)$
$G = 0, L = 0$ if $(a_1\ a_0) = (b_1\ b_0)$

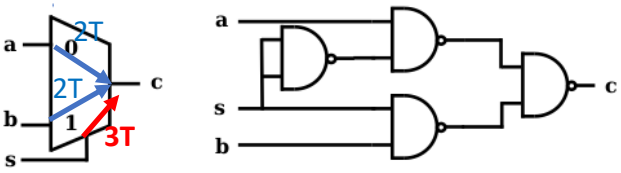
You need to:

a. (1p) Write the truth table with a_1, a_0, b_1, b_0 as inputs and G, L as outputs.
b. (1p) Implement 2-to-1 MUX using only 2-input NAND gate. Find out the critical path for your implementation.
c. (1p) Using Shannon Decomposition to find the simplified expression **only for G**.

d. (1p) Draw your design for **output G** using 2-to-1 MUXes and find out the critical path.

Solution:

a1a0	b1b0	G	L
00	00	0	0
00	01	0	1
00	10	1	0
00	11	1	0
01	00	1	0
01	01	0	0
01	10	1	0
01	11	1	0
10	00	0	1
10	01	0	1
10	10	0	0
10	11	0	1
11	00	0	1
11	01	0	1
11	10	1	0
11	11	0	0



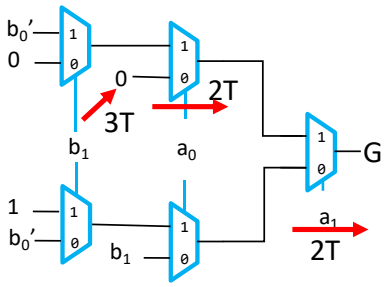
$G = a_1'a_0'b_1b_0' + a_1'a_0'b_1b_0 + a_1'a_0b_1'b_0' + a_1'a_0b_1b_0' + a_1'a_0b_1b_0 + a_1a_0b_1b_0'$

$= a_1'(a_0'b_1b_0' + a_0'b_1b_0 + a_0b_1'b_0' + a_0b_1b_0' + a_0b_1b_0) + a_1(a_0b_1b_0')$

$= a_1'(a_0'(b_1b_0' + b_1b_0) + a_0(b_1'b_0' + b_1b_0' + b_1b_0)) + a_1(a_0(b_1b_0'))$

$= a_1'(a_0'(b_1(b_0' + b_0)) + a_0(b_1'(b_0') + b_1(b_0' + b_0))) + a_1(a_0(b_1(b_0')))$

$= a_1'(a_0'(b_1) + a_0(b_1'(b_0') + b1(1))) + a_1(a_0(b_1'(0) + b_1(b_0')) + a_0(0))$

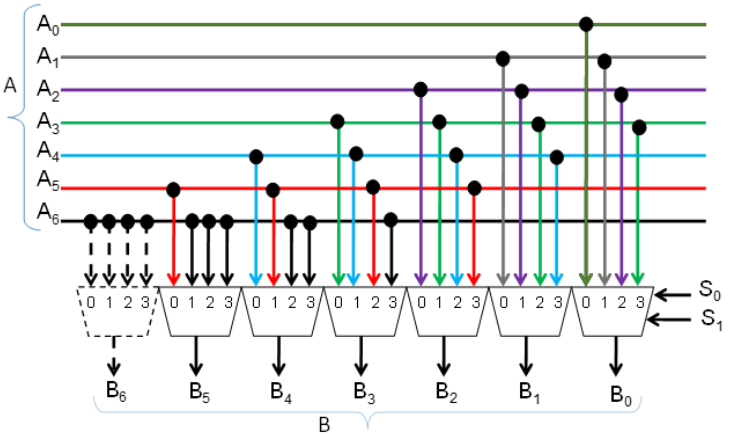


9. (2p)

Analyze the design shown in the figure below.

a. (1p) What is the function that is performed by the following design? You can choose from: no operation, arithmetic right shift, arithmetic left shift, logical right shift, logical left shift, rotate right, rotate left.

b. (1p) If A = $(6A)_{16}$ and $S_1S_0 = (10)_2$ then what would be the value of B in **hexadecimal** format.



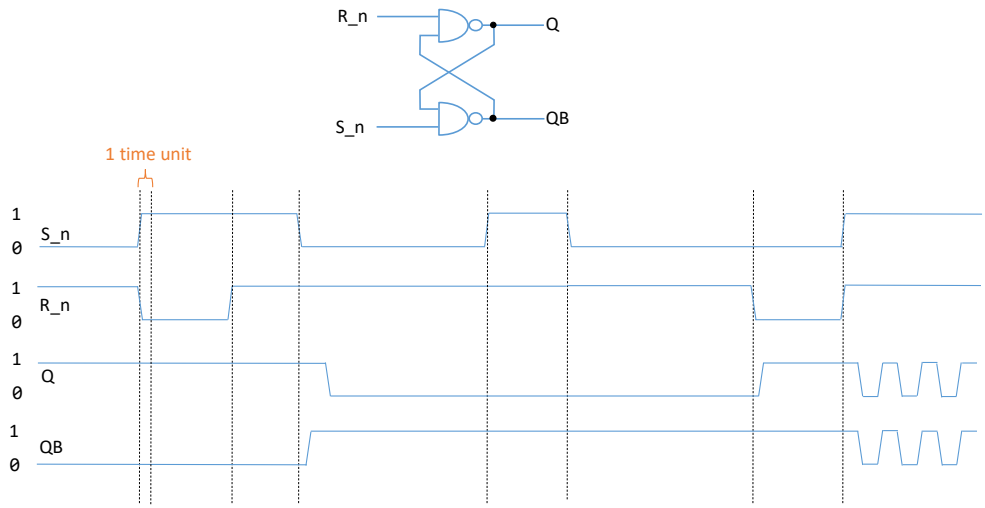
Solution:

a. Arithmetic right shift
b. $(7A)_{16}$

Part 4 (16p)

10. (2p)

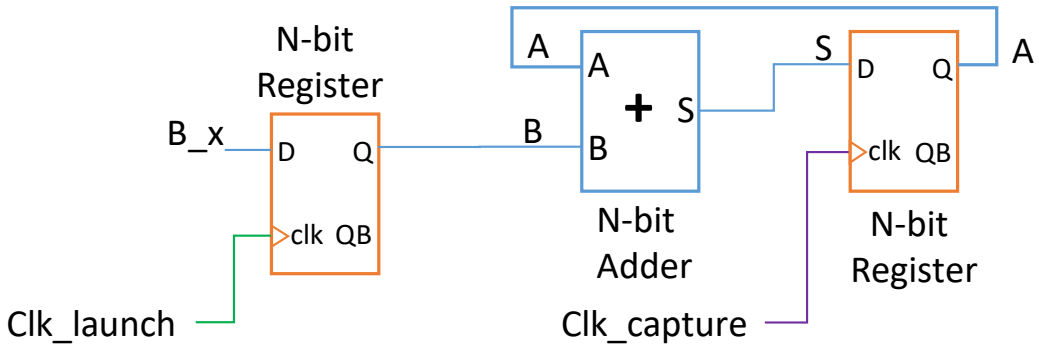
For the S-R Latch, complete the timing diagram for output Q and QB. Assume that both NAND gates have a small delay of 1 time unit. *Note: use the separate answer sheet provided to write your answer for this question.*



11. (4p)

Given:

- t_{Setup} is 1 ns,
- t_{hold} is 0.5 ns,
- $t_{Clock\text{-}2\text{-}Q}$ delay is 1.5 ns,
- T is the delay of one 2-input NAND gate: 1 ns
- Critical path delay formulas:
  - N-bit Ripple carry adder = (4+2N)T
  - N-bit Carry look ahead adder = $2T + (5T \cdot \lceil \text{Log}_4 (N+1) \rceil - 2T) + 6T$
- Shortest path delay formulas:
  - Ripple carry adder = Carry look ahead adder = 2 ns



a) Operating with frequency equal to 20MHz, zero skew between the two flops and 32-bit carry look ahead and ripple carry adder, will there be a setup or hold violation? Answer quantitatively, showing the calculation.

b) If there is a violation what is the amount of skew that can be introduced in the system to correct this violation

c) Operating at frequency of 50MHz. If the design has a *positive* skew of 1 ns, how many bits can such a design have with carry look ahead adder

d) Considering the maximum number of bits that you calculated in question 11.b) what is the maximum positive skew that design can tolerate before you get a hold violation?

Note: The design includes two different paths; the skew only affects one of the two paths

Solution:

a) 32 bit ripple carry adder critical path delay = 68 ns
32 bit carry look ahead critical path delay =26
Setup check:
Ripple carry adder: 68 + 1 + 1.5 =< 50 → setup violation, slack -20.5ns
Carry look ahead : 26 + 1 + 1.5 =<50 → setup ok, slack 21.5
Hold check for both:
2 + 1.5 >= 0.5 → no violation

b) To solve the setup violation a positive skew of 20.5ns is needed, it will cause hold violation though

c) Critical + 1 + 1.5 =< 20 + 1 ➔ critical =< 18.5 ➔ $\log_4(N+1)$ =< 2.5 ➔ N+ 1 =< 16 ➔ N_{max}= 15

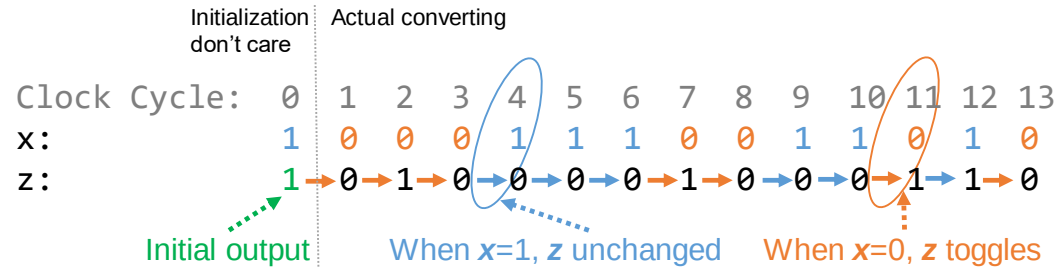
d) Bits do not affect the shortest path → do not affect hold check → max positive skew that does not give a hold violation → 2 + 1.5 >= 0.5 + skew ➔ skew =< 3

12. (3p)

Design a message converting system with the conversion rule below:

For every cycle, if the input message bit is '0', toggle the current output bit; if the input message bit is '1', keep the output bit unchanged.

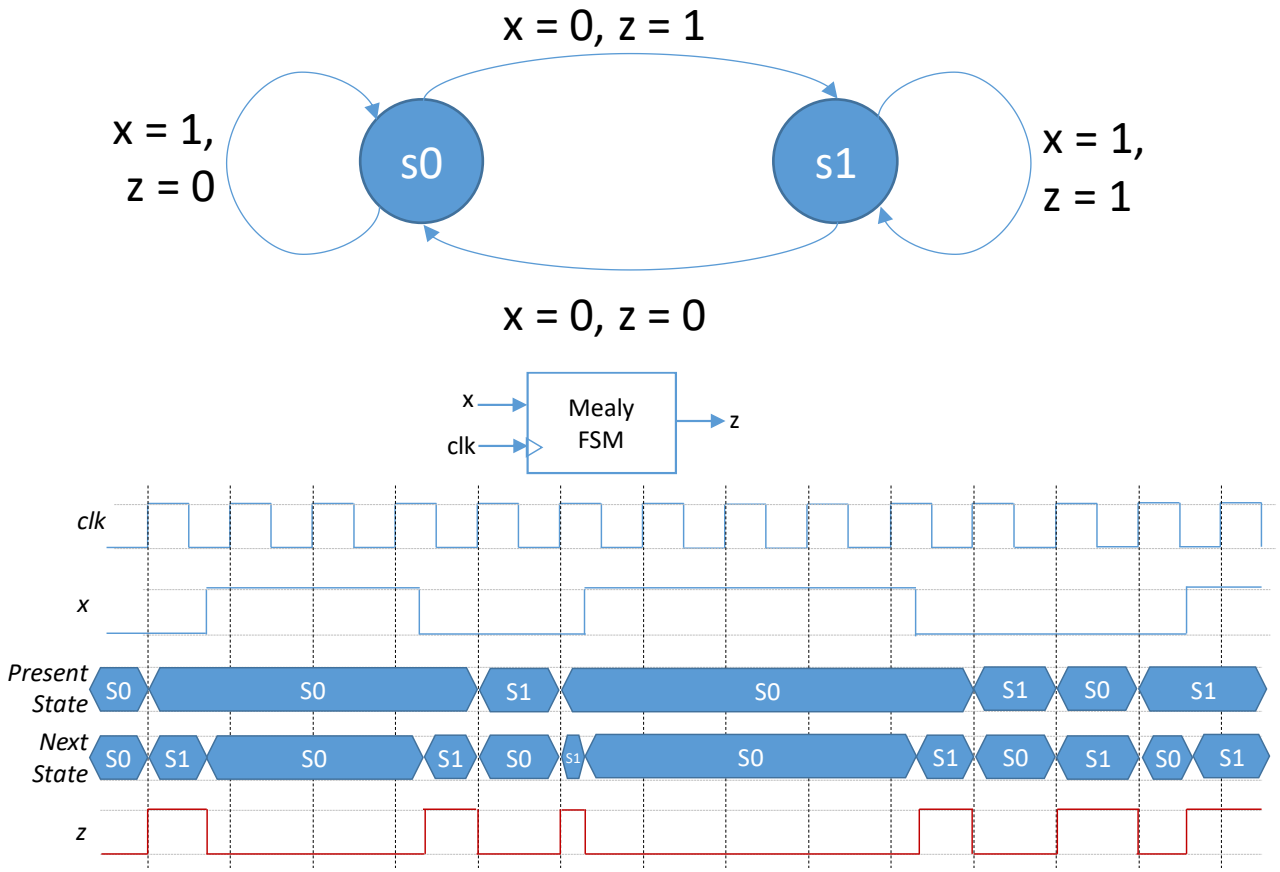
For example, let input message to be ***x***, and output message to be ***z***. Assuming ***z*** initially is '1'. Giving the sequence of ***x*** = 10001110011010, the output sequence of ***z*** should be: 10100001000110. See figure below.



a) (1p) Draw the state diagram of a **Mealy FSM** with the same behavior as described above. No state transition table needed. Please use state symbol "S0, S1, S2 …" to represent your states.

b) (2p) Draw the timing diagram in terms of symbolic present and next states and output.
Note: To draw the diagram properly, you may follow the "style guide" section at the beginning of the exam paper.
Note: use answer sheet to write your answer for this question.

Solution:



13. (4p)
Design a modulo 10 counter that starts at 0 counts up by 2 and loops around, i.e. when the counter reaches 8 should go to 0. The counter counts up in every clock cycle.

a) (1p) Derive the symbolic state transition table and then do state assignment such that state code represents the desired output. *Reminder*: State code is the combination of bits that identifies a specific state. State assignment is the process of assigning such code to the states.
b) (1p) Implement the next state logic using T flops. Only equations needed. No need for logic schematics.
c) (2p) Identify illegal states and what would be the next states from illegal states. *Hint: you can use next state equations or K-map to find the next states from the illegal present states.*

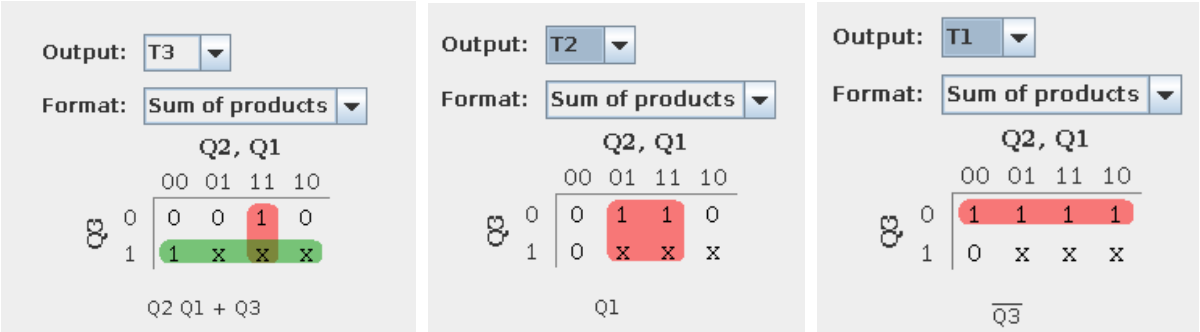
Solution:

Since we only need even numbers, the last bit Q0 is always 0. We only need 3 bits to represent number from 0 to 8.

Current state [Q3, Q2, Q1]	Next State	T flop input [T3, T2, T1]
0 [000]	2 [001]	001
2 [001]	4 [010]	011
4 [010]	6 [011]	001

6 [011]	8 [100]	111
8 [100]	0 [000]	100
others	Don't care [---]	---

We use K-map to calculate T3, T2 and T1.



Illegal states are:

Current state [Q3, Q2, Q1]	Next State	T flop input [T3, T2, T1]
10 [101]	6 [011]	110
12 [110]	4 [010]	100
14 [111]	2 [001]	110

14. (3p)

We want to create an asynchronous FSM with 2 inputs in1 and in2, and one output Z that can implement the following behavior.

a) Output toggles when in1 becomes '1' before in2 becomes '0'
b) Output becomes '0' when in1 becomes '1' before in2 becomes '1'
c) In all other conditions the output remains unchanged
d) Assume that initially both inputs are zero and the output is also zero

Draw the state diagram and use state names as S0, S1, S2 …