Written Exam
IE1205 Digital Design
2022-04-21, 08:00-12:00

General Information

Examiner: Ahmed Hemani

Responsible teacher for the exam: Ahmed Hemani

Exam text must be turned in when you hand in your solutions. No aids allowed except ruler and calculator.

Start each problem on a new piece of paper. Do not write on the backside of the papers.

Write your name and personal number on every paper that you hand in.

Organization

The exam consists of three parts and a total of 40 points. The exercises are NOT in order of difficulty.

Part 1 (10p): Digital Design Fundamentals

Part 2 (14p): Binary Arithmetic and Combinational Building Blocks

Part 3 (16p): Sequential Circuit and State Machines

Grade

To pass the exam requires **at least 20 points in total** AND **a minimum of 2 points of each part reached.**

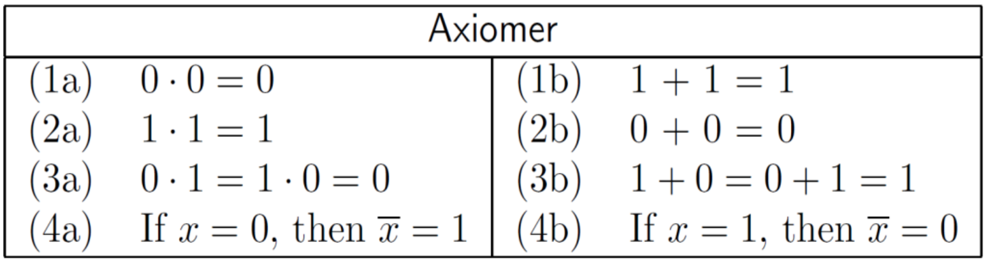
Grades are given as follows:

0 – 19	20 – 23	24 – 27	28 – 31	32 – 34	35 – 40
F	E	D	C	B	A

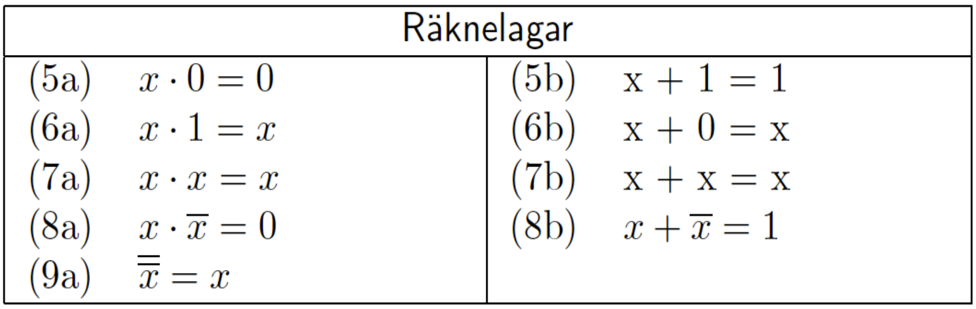
Fx if (19 points AND minimal required points reached for all parts) OR (20 points or more AND minimal required points reached except for 1 part)

Cheat Sheet

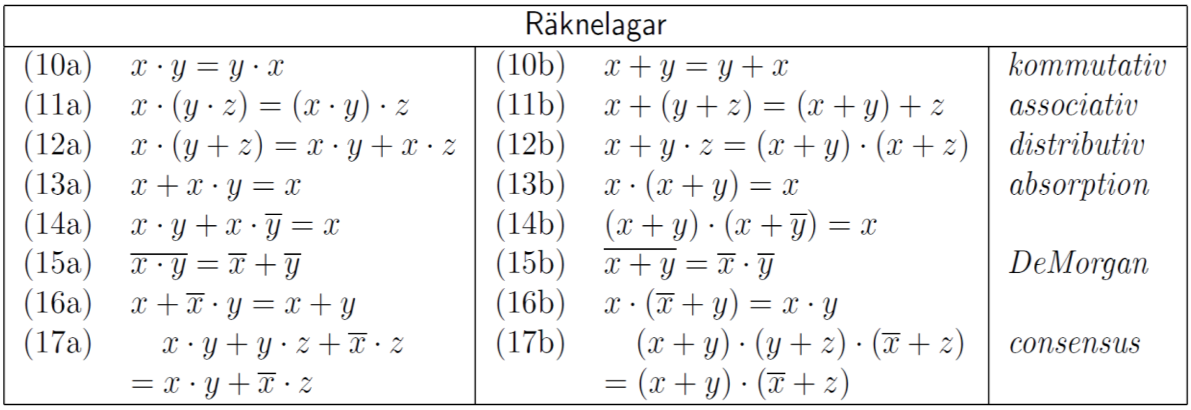
Boolean Identities



Axiomer			
(1a)	$0 \cdot 0 = 0$	(1b)	$1 + 1 = 1$
(2a)	$1 \cdot 1 = 1$	(2b)	$0 + 0 = 0$
(3a)	$0 \cdot 1 = 1 \cdot 0 = 0$	(3b)	$1 + 0 = 0 + 1 = 1$
(4a)	If $x = 0$, then $\overline{x} = 1$	(4b)	If $x = 1$, then $\overline{x} = 0$



Räknelagar			
(5a)	$x \cdot 0 = 0$	(5b)	$\mathrm{x} + 1 = 1$
(6a)	$x \cdot 1 = x$	(6b)	$\mathrm{x} + 0 = \mathrm{x}$
(7a)	$x \cdot x = x$	(7b)	$\mathrm{x} + \mathrm{x} = \mathrm{x}$
(8a)	$x \cdot \overline{x} = 0$	(8b)	$x + \overline{x} = 1$
(9a)	$\overline{\overline{x}} = x$		



Räknelagar				
(10a)	$x \cdot y = y \cdot x$	(10b)	$x + y = y + x$	*kommutativ*
(11a)	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	(11b)	$x + (y + z) = (x + y) + z$	*associativ*
(12a)	$x \cdot (y + z) = x \cdot y + x \cdot z$	(12b)	$x + y \cdot z = (x + y) \cdot (x + z)$	*distributiv*
(13a)	$x + x \cdot y = x$	(13b)	$x \cdot (x + y) = x$	*absorption*
(14a)	$x \cdot y + x \cdot \overline{y} = x$	(14b)	$(x + y) \cdot (x + \overline{y}) = x$	
(15a)	$\overline{x \cdot y} = \overline{x} + \overline{y}$	(15b)	$\overline{x + y} = \overline{x} \cdot \overline{y}$	*DeMorgan*
(16a)	$x + \overline{x} \cdot y = x + y$	(16b)	$x \cdot (\overline{x} + y) = x \cdot y$	
(17a)	$x \cdot y + y \cdot z + \overline{x} \cdot z$ $= x \cdot y + \overline{x} \cdot z$	(17b)	$(x + y) \cdot (y + z) \cdot (\overline{x} + z)$ $= (x + y) \cdot (\overline{x} + z)$	*consensus*

Area and Delay Costs of 2, 3 and 4 input NAND Gates

1. It is allowed to use 2, 3, 4-input NAND/NOR gate unless stated otherwise in the question.
2. Properties of NAND gates:
   a. 2-input NAND: delay=1.00T, area=1.0A
   b. 3-input NAND: delay=1.25T, area=1.5A
   c. 4-input NAND: delay=1.50T, area=2.0A
3. Delay of N input NAND gates in terms of
   a. 2 input NAND gates: $2.00\ T \cdot \lceil \log_2 (N) \rceil - T$
   b. 3 input NAND gates: $2.25\ T \cdot \lceil \log_3 (N) \rceil - T$
   c. 4 input NAND gates: $2.50\ T \cdot \lceil \log_4 (N) \rceil - T$
   d. $\lceil\ \rceil$ is the round-up function e.g. $\lceil 2.3 \rceil = 3$ and $\lceil 2.7 \rceil = 3$

e. To compute “log” to the base 3 or 4 in terms of the natural logarithm, you can use these formulas: i) $\log_3(N) = \log(N)/\log(3)$ ii) $\log_4(N) = \log(N)/\log(4)$

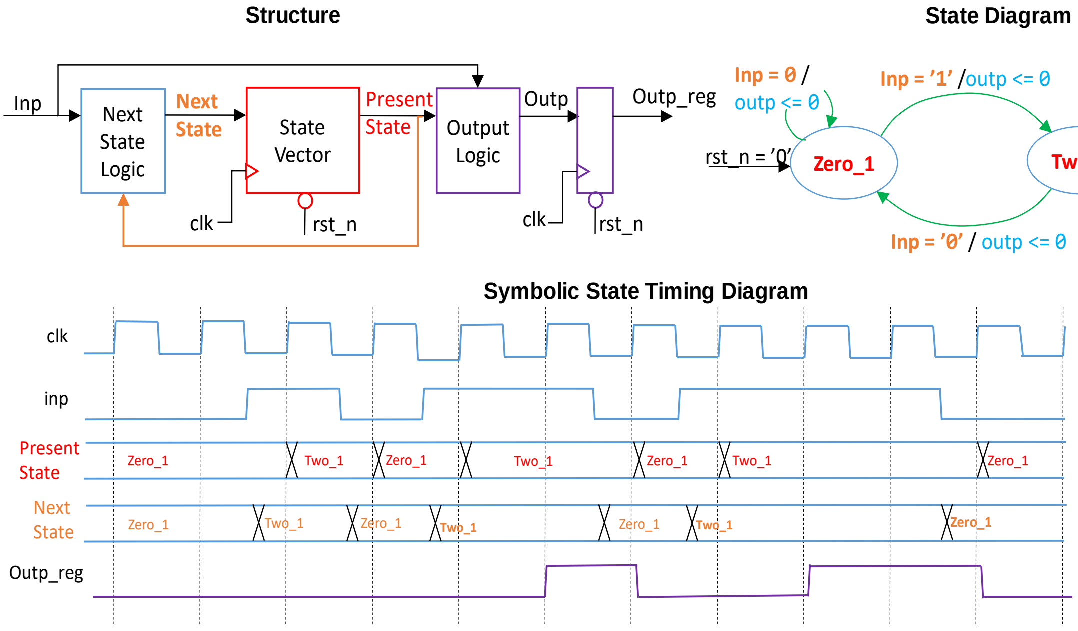
Excitation Tables of Latches and Flops

Inputs			Present State		Next State	
Clocked S R Latch (Positive Level Sensitive)						
Clk = 1	S = 0	R = 1	-	-	$Q^+ = 0$	$QB^+ = 1$
Clk = 1	S = 1	R = 0	-	-	$Q^+ = 1$	$QB^+ = 0$
Clk = 1	S = 0	R = 0	Q	QB	$Q^+ = Q$	$QB^+ = QB$
Clk = 1	S = 1	R = 1	-	-	$Q^+ = 1$	$QB^+ = 1$
Clk = 0	S = -	R = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
Clocked D Latch (Positive Clock Level Sensitive)						
Clk = 1		D = 1	-	-	$Q^+ = 1$	$QB^+ = 0$
Clk = 1		D = 0	-	-	$Q^+ = 0$	$QB^+ = 1$
Clk = 0		D = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
D Flip-Flop (Rising Edge Sensitive;)						
Rising edge of the Clk		D = 1	-	-	$Q^+ = 1$	$QB^+ = 0$
		D = 0	-	-	$Q^+ = 0$	$QB^+ = 1$
Rest of the Clk Period		D = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
S R Flip-Flop (Rising Edge Sensitive)						
Rising Edge of the Clk	S = 0	R = 1	-	-	$Q^+ = 0$	$QB^+ = 1$
	S = 1	R = 0	-	-	$Q^+ = 1$	$QB^+ = 0$
	S = 0	R = 0	-	-	$Q^+ = Q$	$QB^+ = QB$
	S = 1	R = 1	Q	QB	$Q^+ = 1$	$QB^+ = 1$
Rest of the Clk Period	S = -	R = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
J K Flip-Flop (Rising Edge Sensitive)						
Rising Edge of the Clk	J = 0	K = 1	-	-	$Q^+ = 0$	$QB^+ = 1$
	J = 1	K = 0	-	-	$Q^+ = 1$	$QB^+ = 0$
	J = 0	K = 0	Q	QB	$Q^+ = Q$	$QB^+ = QB$
	J = 1	K = 1	Q	QB	$Q^+ = QB$	$QB^+ = Q$
Rest of the Clk Period	J = -	K = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$
T Flip-Flop (Rising Edge Sensitive)						
Rising edge of the Clk		T = 1	Q	QB	$Q^+ = QB$	$QB^+ = Q$
		T = 0	Q	QB	$Q^+ = Q$	$QB^+ = QB$
Rest of the Clk Period		T = -	Q	QB	$Q^+ = Q$	$QB^+ = QB$

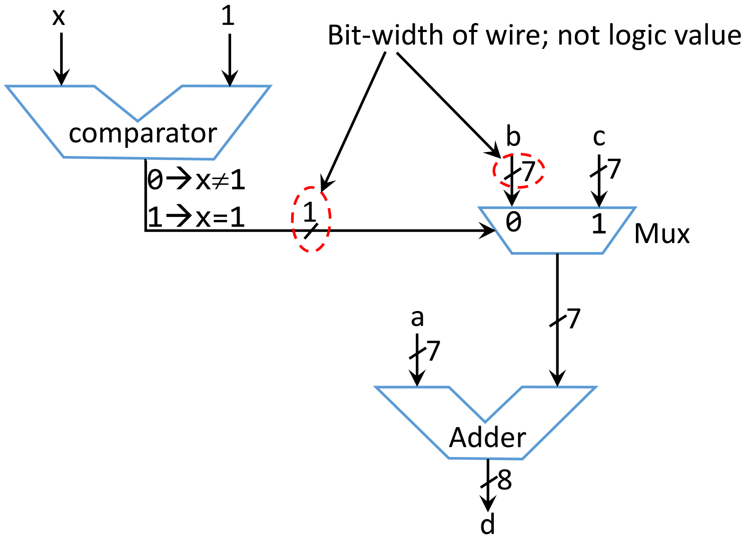
Style Guide of Symbolic Timing Diagram for FSM

As an example, we show how to draw a symbolic timing diagram for a registered Mealy FSM to detect two 1s in a row. The state diagram is shown in the figure below in which state *Zero_1* represents there is no 1 detected, *Two_1* represents there is at least one 1 detected, *Inp* is the input and *Outp* is the output.

When asked to draw a symbolic state diagram, you should draw *present states* and *next states* like the timing diagram shown below. Pay attention to the "present state" and "next state", they are both marked by symbolic states *Zero_0* or *Two_1*. Their transition points are marked by a cross-like divider.



```
If x = 1 Then
   d = a + c
Else
   d = a + b
End If
```



Part 1 (10p)

1. **(2p)**

Eight parties qualified for the Swedish parliament in the last election with a 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**

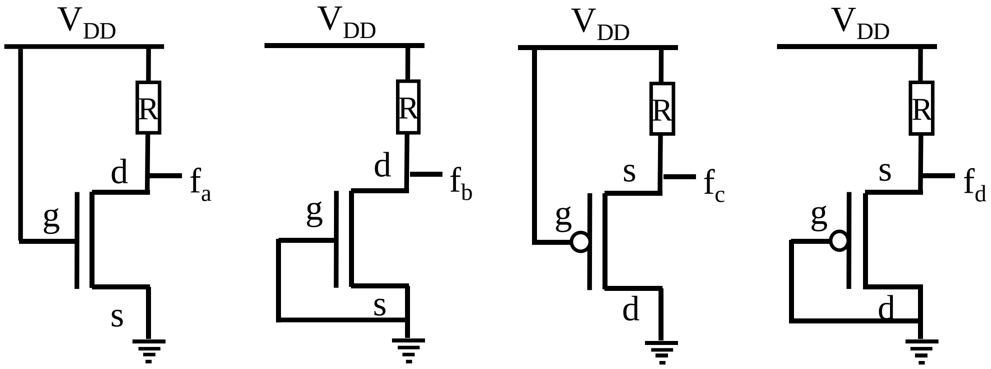
To form a government, 175 seats are required. Formulate this as a Boolean algebra problem. Write a truth table, where each row represents a valid constellation of eight parties that has a minimum of 175 votes. The truth table should only have minterms for the ON set. Factor in the following positions of the parties:
All parties, except, M and KD will not form a constellation in which SD is present
All parties, except, S and MP will not form a constellation in which V is present

Hints:
Organize the columns in the order of the number of seats in the parliament from left to right. Each row will represent a valid constellation.
In each row, 1 in a column represents the inclusion of that particular party. 0 represents not inclusion and "–" represents do not care.

2. **(2p)**

Answer the two multiple choice questions on CMOS:

(1) (1p) In the four configurations of NMOS and PMOS transistors, what should be the voltage value of f_a, f_b, f_c and f_d? Write an expression in terms of V_{DD}, 0 and V_t (threshold voltage)



f_a =
f_b =
f_c =
f_d =

(2) (1p) In an attempt to make the transition time (from Low to High and from High to Low) for PMOS and NMOS transistors equal, which one of the following is a correct measure:

A. Increase the size of PMOS because the majority charge carriers in PMOS are holes that have higher mobility.

B. Increase the size of NMOS because the majority charge carriers in NMOS are electrons that have higher mobility.

C. Increase the size of PMOS because the majority charge carriers in PMOS are holes that have lower mobility.

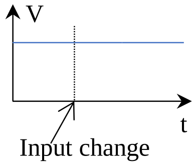
D. Increase the size of NMOS because the majority charge carriers in NMOS are electrons that have lower mobility.

3. (2p)

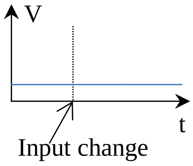
(1) (1p) Implement the function $z = c' + a'b'$ using only NMOS. Draw the schematic of the NMOS implementation, assuming that the inverted versions of signals a, b, and c are also available. Other electrical components except for PMOS (such as wires, resistors, capacitors and power sources) are also available.

(2) (1p) If you have a load capacitor attached to the output of the above circuit, how will the voltage of the load capacitor change when the inputs change from a=0, b=0, c=0 to a=1, b=1, c=1? Choose from the following options:

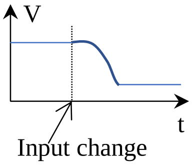
(A)



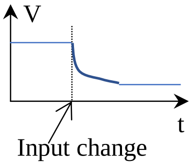
(B)



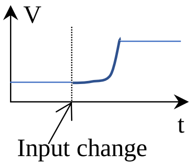
(C)



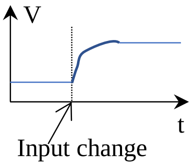
(D)



(E)



(F)



4. (4p)

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:

(1) (1p) Write the truth table.

(2) (1p) Find the minimum SOP form of output "f" using Karnaugh map. Expression is sufficient, no need to draw the logic schematic.

(3) (1p) Find the minimum POS form of output "f" using Karnaugh map. Expression is sufficient, no need to draw the logic schematic.

(4) (1p) Implement the majority function using only NAND gates based on the SOP form. You can use 2-input, 3-input and 4-input NAND gates if needed.

Part 2 (14p)

5. (2p)

(1) (1p) Multiply 3-bit two's complement numbers. The multiplicand is 2 and the multiplier is -3. Express the result as 6-bit 2's complement number.

(2) (1p) Multiply multiplicand 0.75 with multiplier -0.75 as two 2's complement fixed-point binary numbers in Q1.2 format. The result format should be Q2.4.

Note: In both problems above, you are not allowed to convert the negative number to a positive number. Nor can you change the role of multiplicand and multiplier.

6. (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=$(2)_3$.

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

7. (3p)

Implement a Full Adder using Shannon decomposition.

(1) (1p) Write the equation of the outputs of Full Adder – carry-out (co) and sum (s) in terms of its inputs a, b and ci. You can derive the equations from the truth table.

(2) (2p) Use Shannon decomposition to create a custom MUX-based structure for carry-out and sum. Draw the schematics of custom MUX implementation in terms of 2-to-1 Muxes. Custom MUX structure implies that you stop Shannon decomposition as soon as you have reached a literal in complemented or un-complemented form or a constant (0 or 1). You do not need to draw the gate-level schematic of 2-to-1 MUX

8. (3p)

Draw the schematic of digital design using comparator (to check if two inputs are equal or not), adder, multiplexors and wires.
Assume A, B, C and D are 8-bit unsigned integers
Clearly mark the width of the wires and draw the schematic shown in the style guide

```
if A = B Then // “=” means “equal to”
  if A = C Then
    D <= A – C // “<=” means assigned to, i.e. A-C is assigned to D
  else
    D <= A + C
  end if
else
  D <= B
end if
```

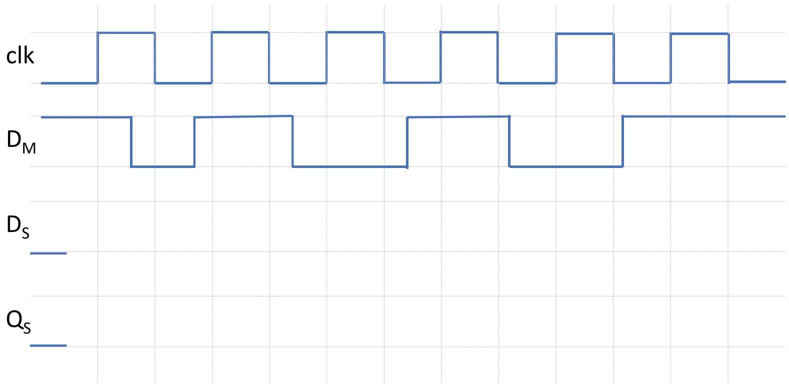
9. (3p)

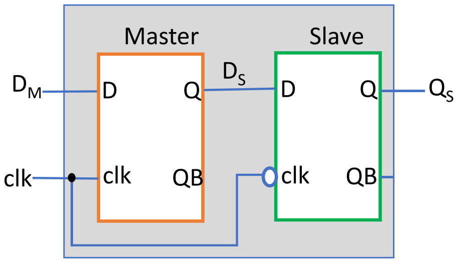
Implement 4-to-16 Decoder using 2-to-4 Decoders and MUXes.
You need to:
(1p) Implement the 2-to-4 Decoder using basic boolean gates: AND gate, OR gate, and Inverter.
(2p) Implement the 4-to-16 Decoder in terms of 2-to-4 Decoders.

Part 3 (16p)

10. (2p)

Complete the timing diagram for the negative edge triggered flip flop. (Orange and Green boxes are both latches). **Note:** You should answer this question on the "Answer Sheet".

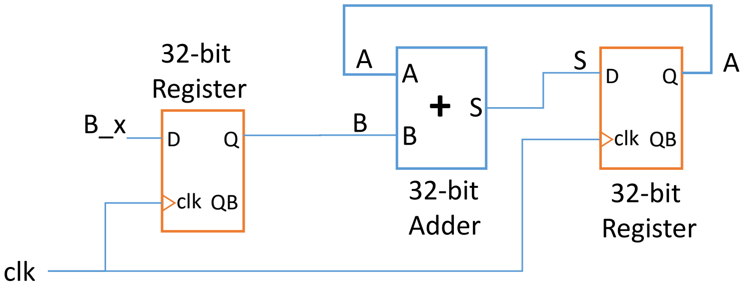




11. (4p)

For the following Flop based circuit, the clocks (clk_launch and clk_capture) are 100 MHz (10 ns clock period).

T_{SU} (setup time) is 1 ns,
$T_{clk\text{-}2\text{-}Q}$ (clock to Q Delay) is 1 ns
T_{hold} (hold time) is 0.1 ns
T_{MIN} (shortest path of the adder) is 2 ns
T_{MAX} (the longest - critical path) of the adder is 10 ns



Questions:

(1) (1p) What should be the minimum skew between clk_capture and clk_launch to avoid setup violation?

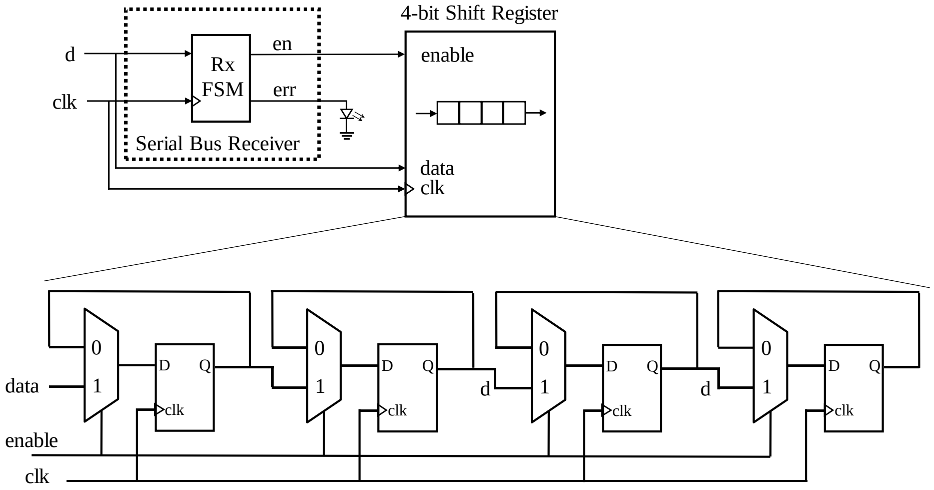
(2) (2p) Will this skew induce hold violation? By how much. Show the calculation

(3) (1p) How will you avoid the hold violation?

Note: You need to answer question (1) and (2) quantitatively and show the calculation that justifies your answers.

12. (4p)

A custom serial bus receiver Rx (inside the dashed box) has the input/output interface as shown in the following diagram:



N.B: In the diagram above, Rx FSM, is an FSM and not a D Flop.

It works under the following protocol:

When the system is in *idle* mode, *d* is always high and no data is pushed into the shift register because *enable* is set to low by Rx FSM, i.e., all four flops retain the values they have.

The receiver Rx FSM starts receiving data when *d* changes from high to low, i.e. in the clock cycle *i*, *d* is high and then in the next clock cycle *i+1, d* is low.

Once Rx detects that it is in the receive mode, the value of *d* for the next four consecutive cycles is assumed to be the new four bits that are serially shifted into the shift register by setting enable to high.

After 4 clock cycles, the receive mode should end by *d* being high for at least two consecutive cycles for Rx FSM to go back into the idle mode. If *d* is not asserted high for two consecutive cycles, Rx goes into the error mode and sets the error signal “err” to high to light up the red LED.

You need to design a **Moore FSM** to implement the Rx FSM (Only the part inside the dashed box).

You need to:

(1) (2p) Draw the state diagram of a Moore FSM that models the behaviour described above. No need to create a state transition table. **The idle state is S0, the error state is S1.** You should use state symbols “S2, S3, S4 …” to represent your other states.

(2) (2p) Draw the timing diagram in terms of symbolic present and next states and outputs.

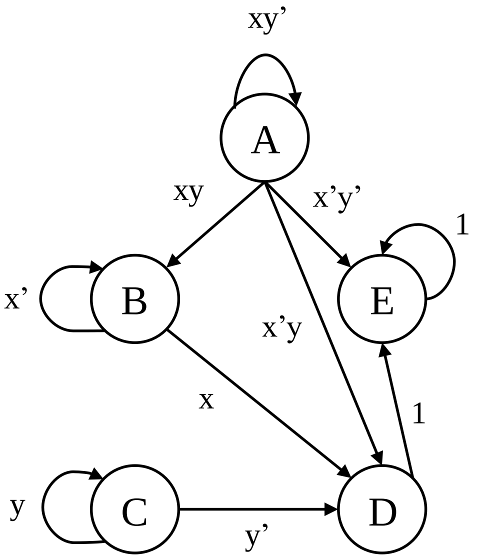
Note:

To draw the diagram properly, you may follow the “style guide” section at the beginning of the exam paper.

Use the answer sheet to write your answer to this question.

13. (4p)

An FSM is described by the following diagram:



The inputs to the FSM are x, and y.

The outputs in these states are the same as the state code, just like what you have in LAB3.

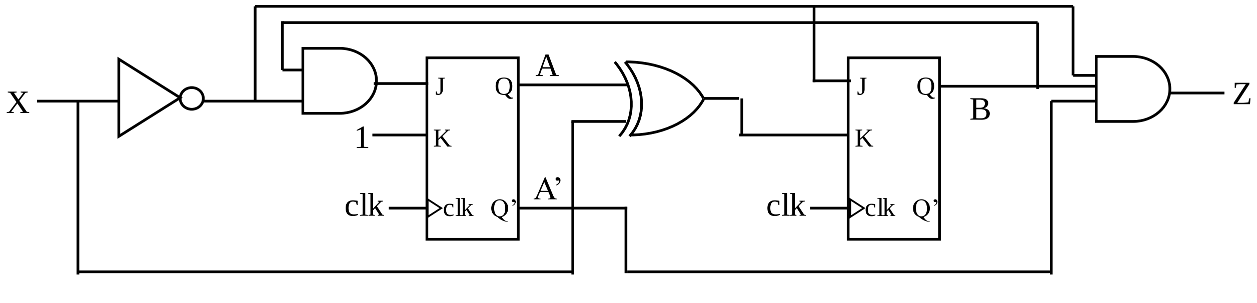
(1) (1p) Derive the symbolic state transition table and then do state assignment. Assign the state in alphabetic order and use binary code, i.e., A=000, B=001 and so on.

(2) (2p) Implement the next state logic using **T flops**. Only the next state equations are needed. No need for logic schematics.

(3) (1p) Identify illegal states and what would be the next states from the illegal states.

14. (2p)

An FSM is shown below.



Answer the following questions:

(1) (1p) What is the next state logic expression and output expression? You can call the J/K signal JA/KA for the JK flop that produces signal A, and JB/KB for the flop that produces signal B.

(2) (1p) Write the state transition table of the FSM.