EEN202 Test **28 April 2021**

Total: 65 Marks **Indicative total time: 120 minutes**

Student Name	**Tristan Basinang**
Student Number	**300524583**

Instructions:

Attempt all questions.

You have until Thursday 29 April at 4 pm NZ time to complete and submit the test.

Your answers can be neatly hand written and any figures or schematics can be hand drawn or you can type your answers and/or submit electronic drawings.

Turn your answers sheets into a single pdf document and submit on the wiki. Ensure that your name is student number is on the test.

The test should be completed individually, but you can use resources such as your class notes, recorded lectures or even the internet to help you. However, you should not discuss the test or collaborate with fellow students in completion of the test.

I will have a Zoom meeting on Wednesday 27 April at 4 pm NZ time to discuss any test related issues.

Estimated time for completion: The test is roughly equivalent to a conventional 90 minute closed book test and you should thus budget no more than two hours for completion of the test and creation of a pdf document to submit.

The test must be submitted on the Assessment System on the ECS wiki by 4 pm NZ time on Thursday 29 April. Please submit as a pdf and use of filename of the format:

"Last name_First name_EEEN202Test".

Question 1 **[10]**

(a) Convert between the following number systems (show your working):

(i) 578 decimal to binary (2)

Binary Conversion through successive long division; remainders are binary conversion

$$2\overline{)578} = 289\ r\ 0 \rightarrow LSB$$
$$2\overline{)289} = \ 144\ r\ 1$$
$$2\overline{)144} = 72\ r\ 0$$
$$2\overline{)72} = 36\ r\ 0$$
$$2\overline{)36} = 18\ r\ 0$$
$$2\overline{)18} = 9\ r\ 0$$
$$2\overline{)9} = 4\ r\ 1$$
$$2\overline{)4} = 2\ r\ 0$$
$$2\overline{)2} = 1\ r\ 0$$
$$2\overline{)1} = 0\ r\ 1 \rightarrow MSB$$

Binary conversion = 1001000010
Checking:

$$1 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$
$$512 + 0 + 0 + 64 + 0 + 0 + 0 + 0 + 2 + 0$$

(ii) 101101 binary to decimal (2)

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$
$$32 + 0 + 8 + 4 + 0 + 1$$
$$45$$

(iii) 10010011 BCD to decimal (2)

$$(1001)_{tens}\ (0011)_{ones}$$

$$1001 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 1 = 9$$

$$0011 = 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 2 + 1 = 3$$

Decimal: 93

(iv) 691 decimal to BCD (2)

$2\overline{)6} = 3\ r\ 0$	$2\overline{)9} = 4\ r\ 1$	$2\overline{)1} = 0\ r\ 1$
$2\overline{)3} = 1\ r\ 1$	$2\overline{)4} = 2\ r\ 0$	Binary: 1
$2\overline{)1} = 0\ r\ 1$	$2\overline{)2} = 1\ r\ 0$	
Binary: 110	$2\overline{)1} = 0\ r\ 1$	
	Binary: 1001	

BCD: 0110 1001 0001

(v) 796 decimal to hexadecimal (2)

$$2\overline{)796} = 398\ r\ 0$$
$$2\overline{)398} = 199\ r\ 0$$
$$2\overline{)199} = 99\ r\ 1$$
$$2\overline{)99} = 49\ r\ 1$$
$$2\overline{)49} = 24\ r\ 1$$
$$2\overline{)24} = 12\ r\ 0$$
$$2\overline{)12} = 6\ r\ 0$$
$$2\overline{)6} = 3\ r\ 0$$
$$2\overline{)3} = 1\ r\ 1$$
$$2\overline{)1} = 0\ r\ 1$$

Binary: 1100011100
To hex: (0011) (0001) (1100)

$$1100 = 12 = C$$
$$0001 = 1_{dec} = 1_{hex}$$
$$0011 = 3_{dec} = 3_{hex}$$

Hexadecimal: 31C

Question 2 **[15]**

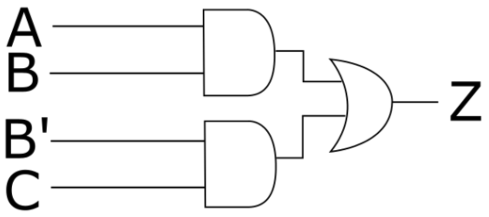
A truth table indicating the state of an output, Z, as it depends on three inputs A, B and C is given below:

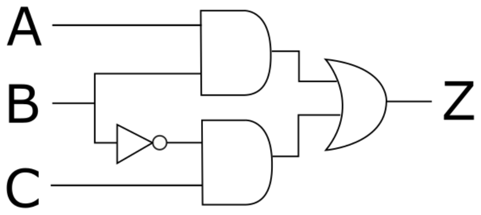
A	**B**	**C**	**Z**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(i) Write down the sum of products expression as represented by the truth table and then simplify this expression if possible. (2)

$$Z = \bar{A}\bar{B}C + A\bar{B}C + AB\bar{C} + ABC$$
$$Z = (\bar{A}\bar{B}C + A\bar{B}C) + (AB\bar{C} + ABC)$$
$$Z = \bar{B}C(\bar{A} + A) + AB(\bar{C} + C)$$
$$Z = \bar{B}C + AB$$

(ii) Sketch the logic circuit that represent your simplified expression in (i) (2)





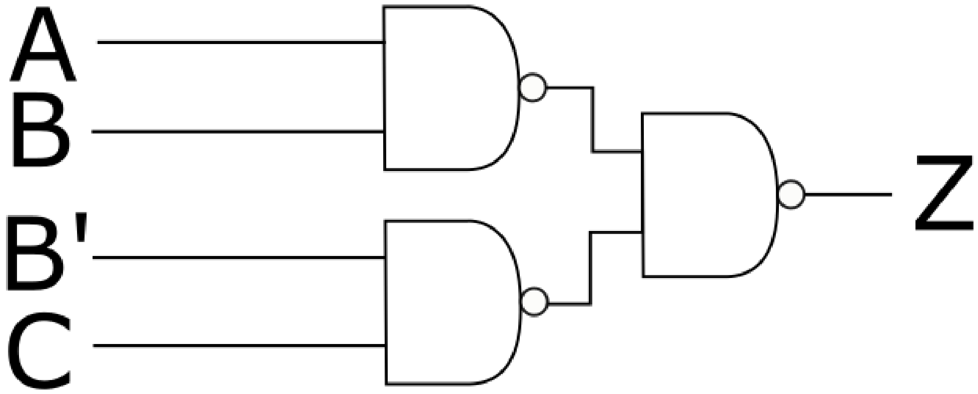
(iii) Now sketch the same logic circuit as for (ii) above but only use NAND gates to implement the logic. (4)

$$Z = \bar{B}C + AB$$

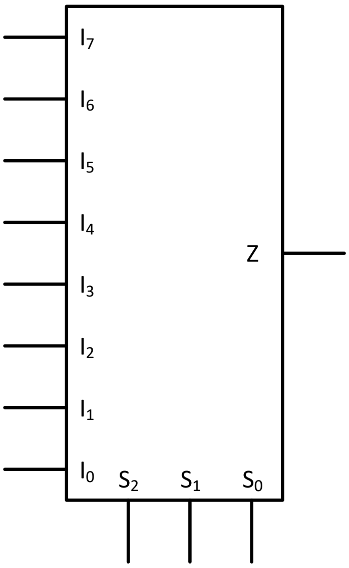
$$\overline{x+y} = \bar{x}.\bar{y} \rightarrow \overline{\overline{x+y}} = \overline{\bar{x}.\bar{y}} \rightarrow x+y = \overline{\bar{x}.\bar{y}} \quad \therefore$$

$$Z = \bar{B}C + AB \rightarrow Z = \overline{\overline{\bar{B}C}.\overline{AB}}$$

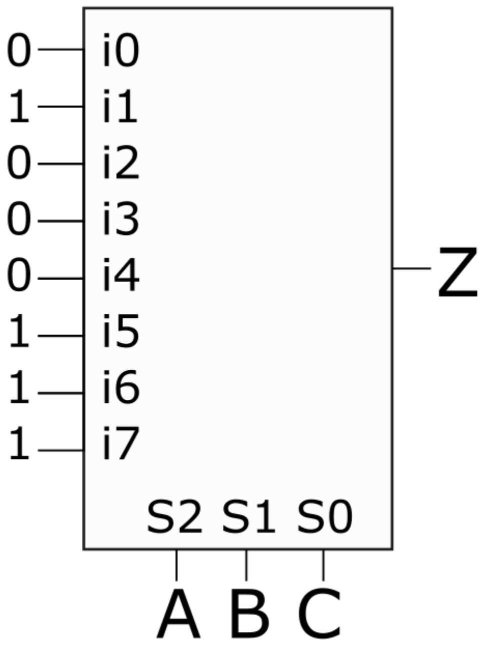
$$Z = \overline{\overline{\bar{B}C}.\overline{AB}}$$



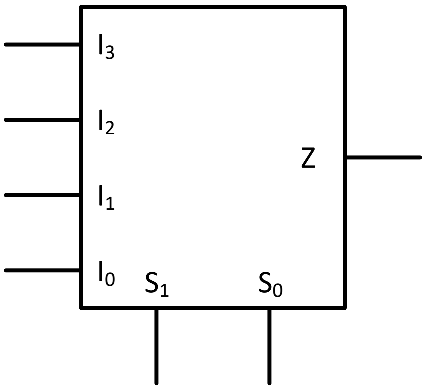
(iv) Show how you can also implement this logic using a generic 8:1 MUX instead of logic gates. (2)



MUX inputs are taken directly from truth table

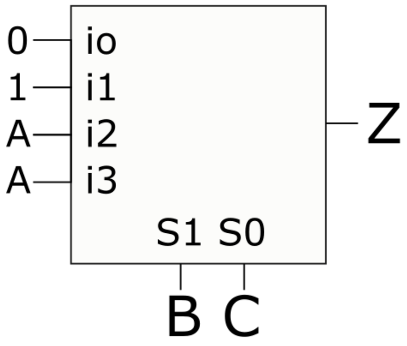


(v) Show how you can implement this logic using a generic 4:1 MUX (2)

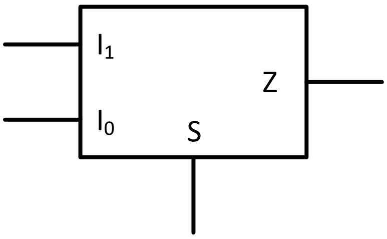


Setting A as an input to the MUX:

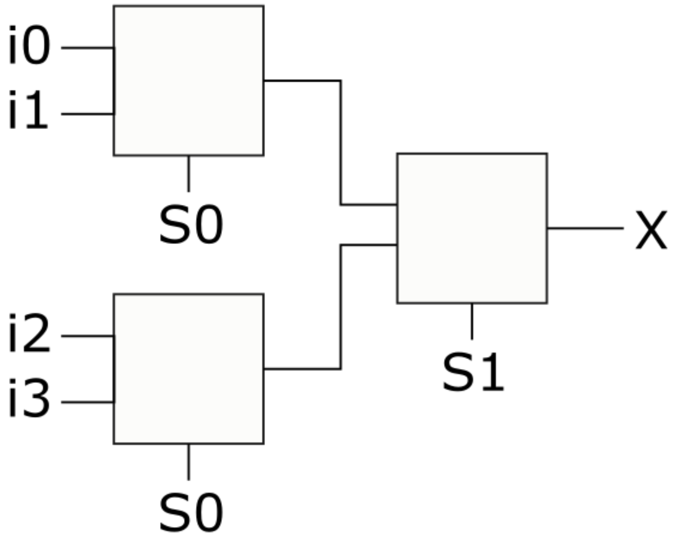
A					A					A=0	A=1	
	B	C	Z			B	C	Z		Z	Z	MUX IN
0	0	0	0		1	0	0	0		0	0	0
0	0	1	1		1	0	1	1		1	1	1
0	1	0	0		1	1	0	1		0	1	A
0	1	1	0		1	1	1	1		0	1	A



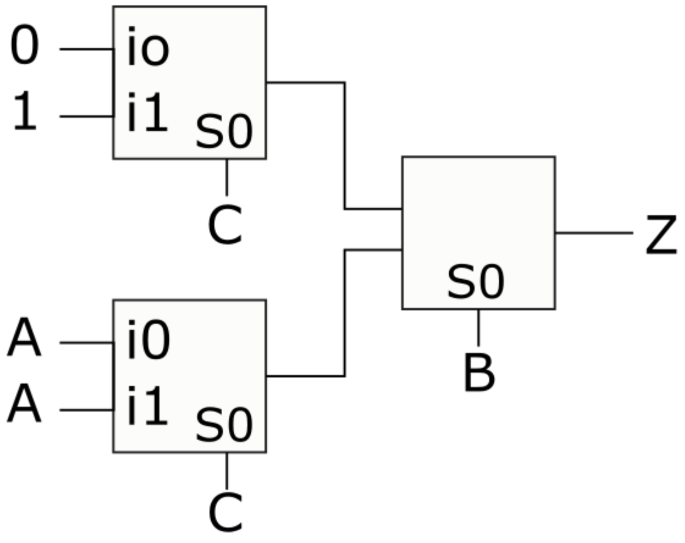
(vi) Show how you can implement this logic using **three** generic 2:1 MUXes as shown below. (i.e. show how you can turn three 2:1 MUXes into a 4:1 MUX) (3)



To turn 3 2:1 MUXs into a 4:1 MUX, the inputs of one MUX are tied to the outputs of the other two, e.g:

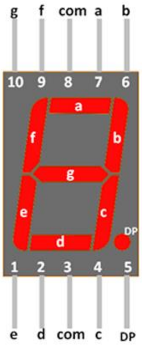


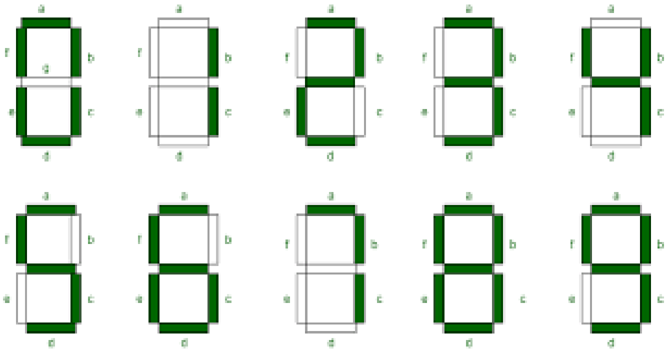
So the final circuit is:



Question 3 **[10]**

A seven segment display can be used to display a decimal 0 – 9 digit by activating the correct segments as shown in the figure below. Note particularly when the b-segment (the top right hand vertical segment) will turn ON or OFF.





(i) Complete the truth table that **<u>will indicate the status of the b segment</u>** when a four-bit BCD code is put on the display. Assume that a 1 output will mean that the segment is turned ON and a 0 output implies that the segment remains OFF. (4)

Following states after 1001 or 9 are different if BCD code can go past 1001 and therefore have undesired states

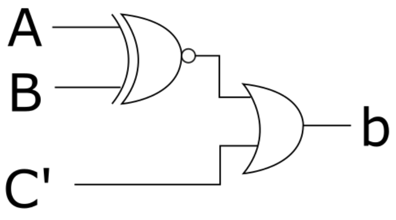
- If no, then output for B is don't care (since those states are never reached)
- If yes, then output for B is 0 (back to the start)

D	C	B	A	b Segment
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X or 0
1	0	1	1	X or 0
1	1	0	0	X or 0
1	1	0	1	X or 0
1	1	1	0	X or 0
1	1	1	1	X or 0

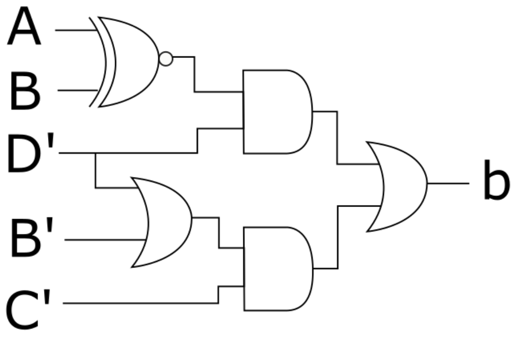
(ii) Use your truth table to construct a K-map and simplify the logic needed to **produce the b-segment.**

If no undesired states						If undesired states present				
	/B/A	/BA	BA	B/A			/B/A	/BA	BA	B/A
/D/C	1	1	1	1		/D/C	1	1	1	1
/DC	1	0	1	0		/DC	1	0	1	0
DC	X(1)	X(0)	X(1)	X(0)		DC	0	0	0	0
D/C	1	1	X(1)	X(1)		D/C	1	1	0	0
$b = \bar{A}\bar{B} + AB + \bar{C}$ $b = \overline{A \oplus B} + \bar{C}$						$b = \bar{C}\bar{D} + \bar{B}\bar{C} + \bar{A}\bar{B}\bar{D} + AB\bar{D}$ $b = \bar{C}(\bar{B} + \bar{D}) + \bar{D}(\overline{A \oplus B})$				

(iii) Implement this logic using logic gates (2)



No undesired states

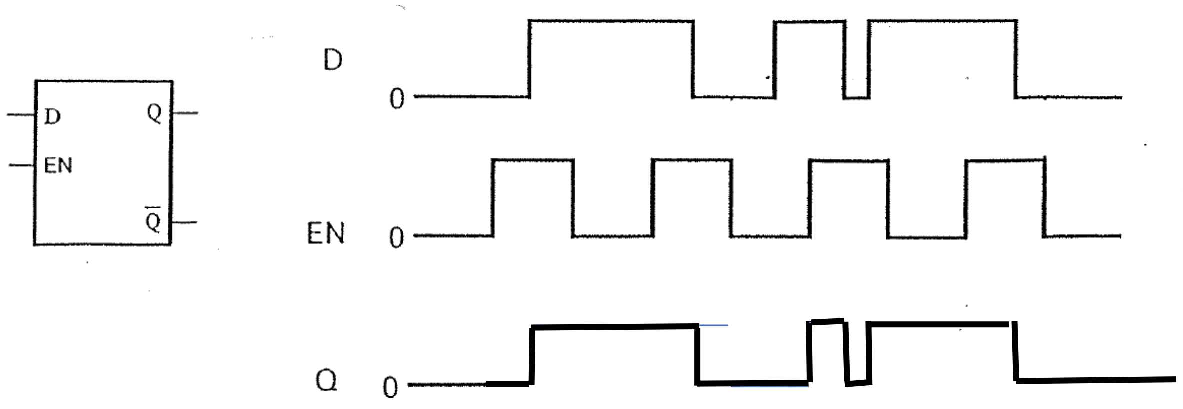


Undesired states

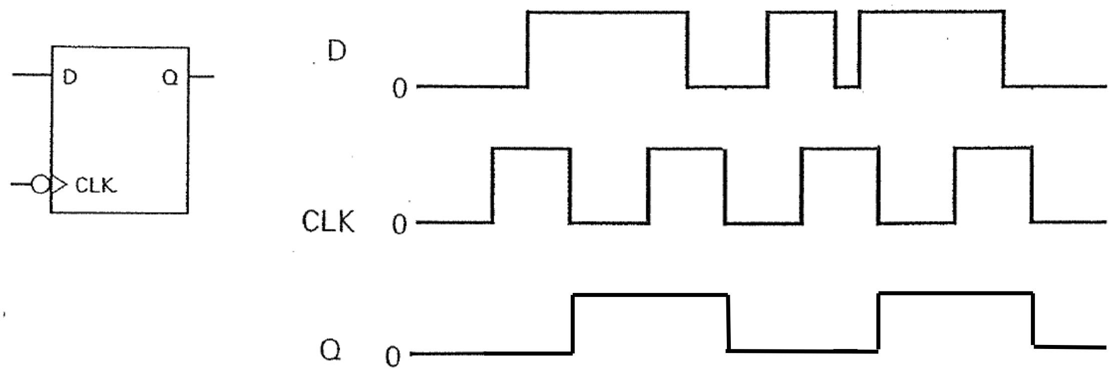
Question 4 **[15]**

(a) For each of the flip flops below complete the timing diagram by adding the timing for the Q output. In each case assume that Q is initially LO.

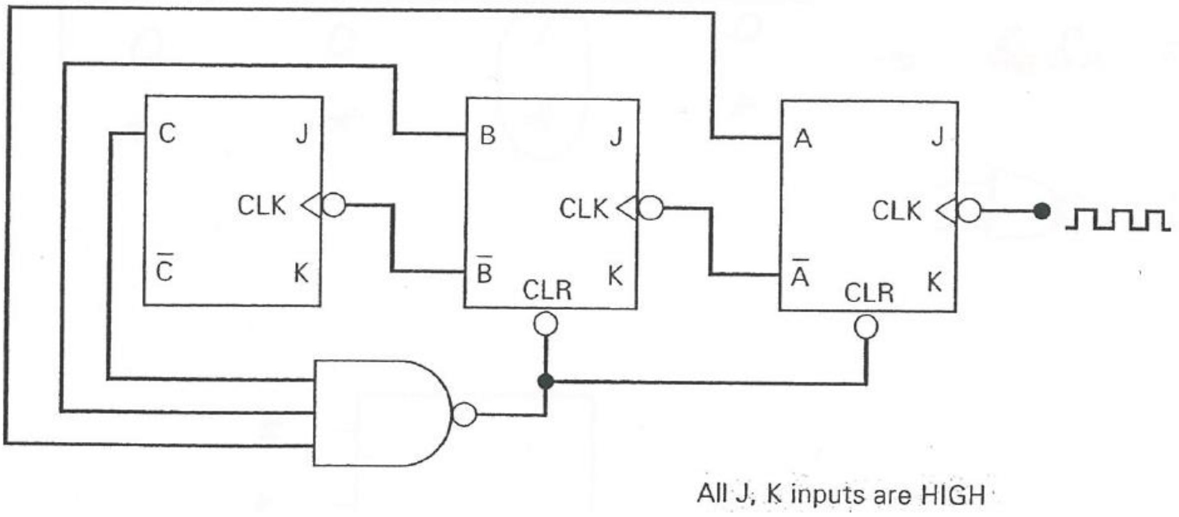
(i) (2)



(ii) (2)



(b) Study the circuit of the counter below and answer the questions below. <u>You can assume that all J=K=1.</u>



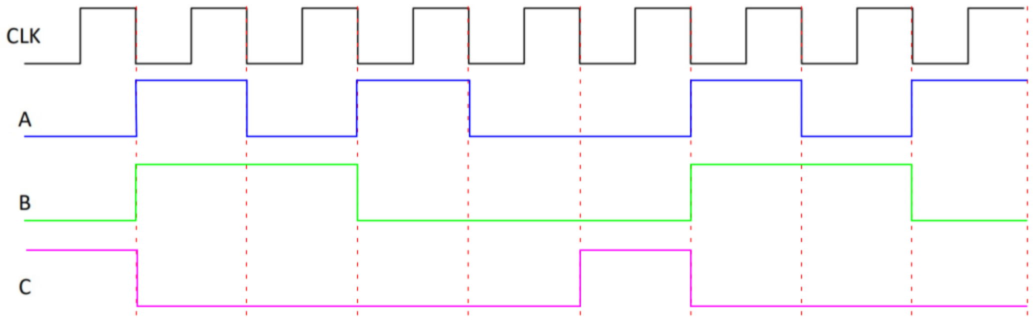
(i) On what output code (CBA) will the counter reset ? (1)

111

(ii) What code will the counter reset to ? (1)

100

(iii) Assume that the counter has just reset to the code in (ii). Sketch a timing diagram showing the CLK pulse as well as the outputs A, B and C for successive further CLK pulses until the counter repeats. (3)



(iv) Determine the counting sequence that the counter will cycle through. (2)

100 011 010 001 000

(iv) What is the MOD number of this counter ? (1)

MOD 5

(c) Explain the problem that will be encountered when using an asynchronous counter with moderately high speed input signals. Base your answer on the use of an 8-bit asynchronous counter and assume that each flip flop element has a propagation delay of approximately 20 ns. (3)

Asynchronous counters naturally have a problem with propagation delay due to the small amount of time required for FFs to process a state change, which gets multiplied by the number of FFs you use and increases if there are any combinatorial logic elements needed. This means there is an upper limit for the range of clock frequencies you can use; higher frequencies can result in states skipping a clock cycle due to propagation delay of the FFs and any other logic.

For an 8 bit counter with a propagation delay of $t_{pd} = 20 \times 10^{-9} s$

$$F_{max} = \frac{1}{N \times t_{pd}}$$

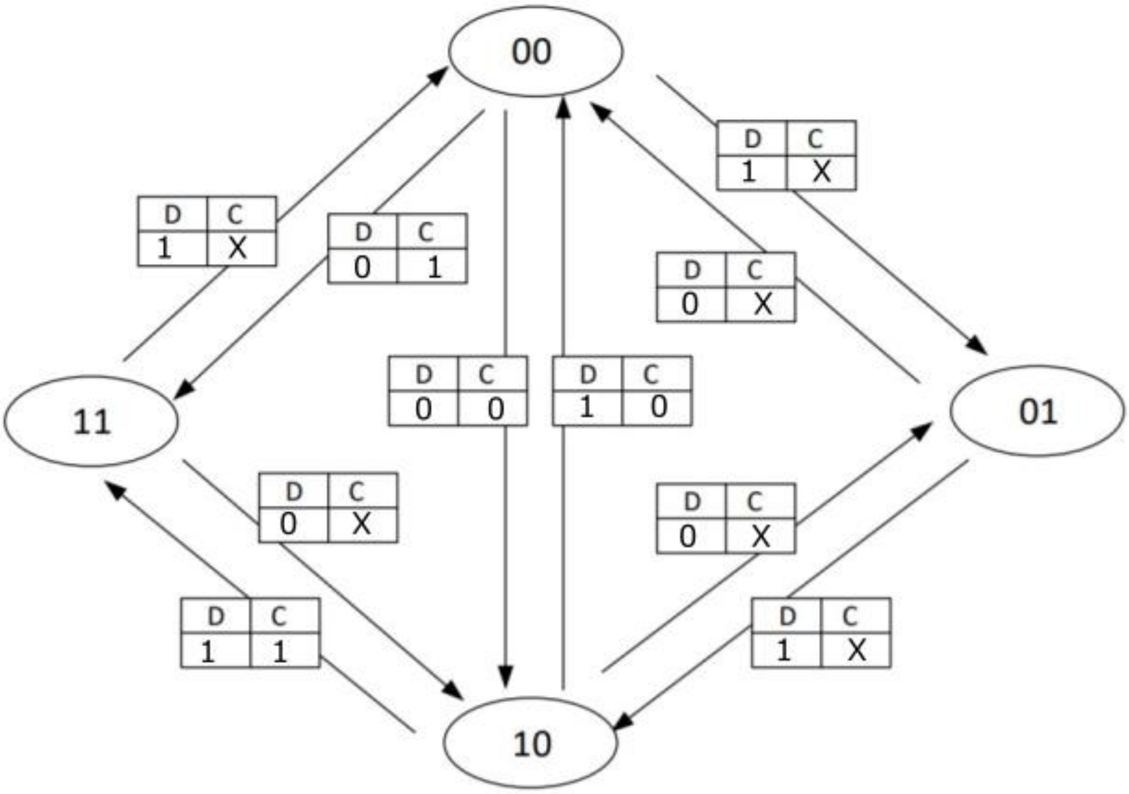
$$F_{max} = \frac{1}{8 \times 20 \times 10^{-9}} = 6{,}250{,}000 = 6.25\ MHz$$

Therefore, for the counter to operate properly, the clock frequency must be below 6.25 MHz, otherwise the propagation delay can mess with the state transition.

Question 5 **(15)**

You must design a counter that can operate either as a MOD 3 counter (states BA counting through 00-01-10) or as a MOD 4 counter (state 11 the additional state). It is driven by an external clock signal (CLK) and the counter should switch between the two types of counters depending on the state of a control input C. With C=0 the counter must be in the MOD 3 counter mode and with C=1 in the MOD 4 counter mode. Both counters should count UP when a direction bit D = 1 and DOWN when D = 0. If the counter should be in the 11 state (MOD 4) and the control bit C changes to 0 (MOD 3 mode) the counter should then go to the correct MOD 3 state as determined by the direction control bit.

(i) The state transition diagram for this design is shown below. Complete this diagram by inserting the values of D and C required for each transition. (5)


00
11
01
10
D C 1 X
D C 0 1
D C 1 X
D C 0 X
D C 0 0
D C 1 0
D C 0 X
D C 0 X
D C 1 1
D C 1 X

(ii) Now complete the excitation table below, filling in the values of the next state $B^{n+=1}A^{n+1}$ for each present state B^nA^n. (5)

(iii) Assume that you are required to design this counter based on J-K flip flops. Complete the excitation table for required values for J_A and K_A, the inputs into the flip flop controlling the LSB. (5)

D	C	B^n	A^n	B^{n+1}	A^{n+1}	J_A	K_A
0	0	0	0	0	1	1	X
0	0	0	1	1	0	X	1
0	0	1	0	0	0	0	X
0	0	1	1	0	0	X	1
0	1	0	0	0	1	1	X
0	1	0	1	1	0	X	1
0	1	1	0	1	1	1	X
0	1	1	1	0	0	X	1
1	0	0	0	1	0	0	X
1	0	0	1	0	0	X	1
1	0	1	0	0	1	1	X
1	0	1	1	0	0	X	1
1	1	0	0	1	1	1	X
1	1	0	1	1	0	X	1
1	1	1	0	0	1	1	X
1	1	1	1	0	0	X	1

*****