Indian Institute of Information Technology Allahabad

Review Test - First Component (C1) (February 2019)

Second semester B.Tech (IT): Section A

Course Name	**Course Code**	**Date of Exam**	**MM**	**Time**
Computer organization and Architecture	ICOA230C	Feb. 27, 2019	30	1 Hr

Important Instructions: *All questions are compulsory. Answer all questions in strict order as is given in the question paper otherwise a penalty of 3 marks will be applied.*

1. **(5 marks) (*Logic Design*)**:

   A IIIT student staying in one of the institute hostels has to make up his/her mind about his/her dinner. There can be three situations which may arise :
   Situation 1 - If he/she has enough money (M) and at least three of his/her friends (F) also agree to go out for dinner, and it is not raining (R), he/she will have dinner with his/her friends in a restaurant at Civil Lines.
   Situation 2 - If he/she is not able to go out, but at least three of his/her friends agree to join him/her (J), if the kind of food he/she wanted is available on home delivery (K) he/she will order home delivery of food from online food delivery services, *Swiggy* maybe.
   Situation 3 - But, if the general feeling is that the food in the hostel mess is good on that day (G), he/she will have his/her dinner in the hostel mess.

   Let his/her decision be denoted by a 2-bit output D_1 D_0:
   D_1 D_0 = 00 : He/She eats in the hostel mess,
   D_1 D_0 = 01 : He/She goes out to have dinner in a restaurant,
   and D_1 D_0 = 10 : He/She orders food through home delivery app.

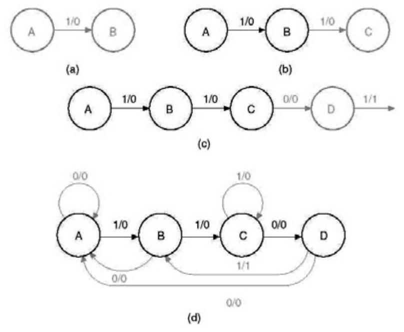
   Assign binary variables to represent the three situations using the letters indicated in parentheses above, and obtain both POS expressions for D_1 and D_0 in terms of these variables, simplifying the expressions as far as possible.

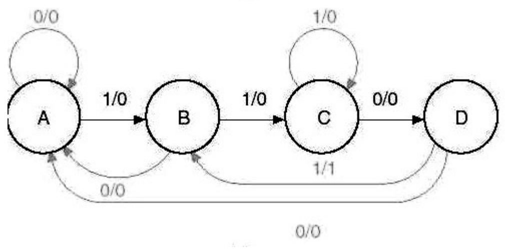
   Solution: (If the third condiion is interpreted as as a condition applicable when he is not able to go out) Expression for D0: D0 is zero only when the student does not go out for a dinner in a restaurant i.e when either of M, F or R' is zero. This gives us D0 = MFR' (choosing R as 1 when it rains)
   Expression for D1: D1 is zero when he eats in the hostel mess or goes out to have dinner in a restaurant. The first condition can be written as JKG' (Any one of J or K being zero he would eat in the mess and also when the food in the mess is good i.e. G' is zero). The second condition can be written as (M' + F' + R), since he would go out for a dinner in a restaurant only if M=F=R'=1 (if MFR' = 1) This gives D1 = (M' + F' + R)JKG'
   Solution: (If the third condiion is interpreted as as a condition applicable under all conditions) Expression for D0: D0 is zero only when the student does not go out for a dinner in a restaurant i.e when either of M, F, R' or G' is zero. This gives us D0 = MFR'G' (choosing R as 1 when it rains)
   Expression for D1: D1 is zero when he eats in the hostel mess or goes out to have dinner in a restaurant. The first condition can be written as JKG' (Any one of J or K being zero he would eat in the mess and also when the food in the mess is good i.e. G' is zero). The second condition can be written as (M' + F' + R + G'), since he would go out for a dinner in a restaurant only if M=F=R'=G'=1 (if MFR'G' = 1) This gives D1 = (M' + F' + R + G)JKG' = (M' + F' + R)JKG'

2. **(5 marks) (*FSM Design*)**:

   Implement a circuit that recognizes the occurrence of the sequence of bits 1101 on input X by making output Z equal to 1 when the previous three inputs to the circuit were 110 and current input is a 1.

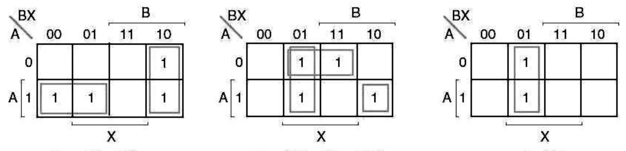
Solution:





Present State	Next State X = 0	Next State X = 1	Output Z X = 0	Output Z X = 1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

Present State A	Present State B	Input X	Next State A	Next State B	Output Y
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	0	0



$D_A = A\overline{B} + B\overline{X}$ $D_B = \overline{A}X + \overline{B}X + AB\overline{X}$ $Y = \overline{B}X$

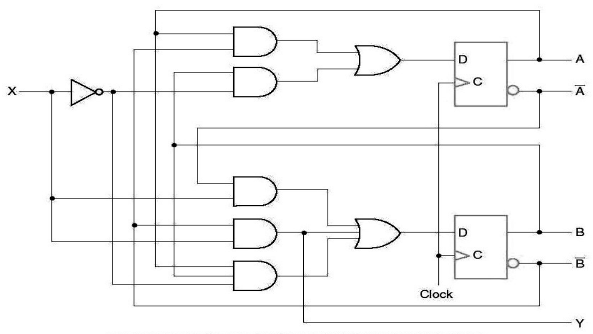


Fig. 4-25 Logic Diagram for Sequential Circuit with *D* Flip-Flops

3. **(5 marks) (*Representing Numbers*)**:

   (a) Represent -4 as a 4-bit and 8-bit two's complement number.

   (b) Subtract 4 from 3 using 2's complement arithmetic. Consider both the numbers of size 4bits.Does the arithmetic operation operation generate overfow?

Solution:

(a)
```
+4 as 4 bit number  = 0100
-4 as 4 bit number  = two's complement of +4
                    = 1011+1 = 1100

+4 as 8 bit number  = 00000100
-4 as 8 bit number  = two's complement of +4
                    = 11111011+1 = 11111100
```

(b)
```
3 - 4 = 00000011 + 11111100 = 11111111
No overflow which means the number is a eight bit number
The MSB bit being 1 represents a negative number whose value is
two's complement of (11111111) = 00000001 = 1
Therefore the result is -1
```

4. **(3+2+5 = 10 marks) (*Floating Point Number Representation*)**:

   (a) Write -6.125 in IEEE 754 single precision. Label all bits properly (sign bit, exponent, fraction).

   (b) How can you tell if a 32 bit number is NaN. That is, describe what bits you look at to determine that it's NaN.

   (c) Around 250 B.C., the Greek mathematician Archimedes proved that 223/71 < pi < 22/7. Had he had access to computer and the standard library <math.h>, he would have been able to determine that the single-precision floating-point approximation of pi has the hexadecimal representation *0x40490FDB*. Of course, all of these are just approximations, since pi is not rational. What is the fractional binary number denoted by this floating-point value?

Solution:

(a) To answer this, you should initially forget about the negative sign.
You'll run into a problem if you deal with it now (in particular,
you may end up convering -6 to 2C and .125 to UB, and then put them
together, which wouldn't make sense.

So, we follow the usual steps:

Convert the integer part to binary: 610 = 1102
Convert the fraction part to binary: 0.12510 = .0012

To get the fraction part, it's useful to know that 2-1 = .5, 2-2 = .25,
2-3 = .125.

The algorithm runs as follows. To convert x (a number less than 1 written
in base 10) to binary, do:
If x > bi, then let bi=1 and let x >= x - 2i.
If x < bi, then let bi=0.
If x == bi, then let bi=1, and stop, otherwise decrement i
and go back to step 1.
In other words, attempt to subtract .5, then .25, then .125.
Repeat until you have converted the number. The bad news is that
some numbers infinitely repeat, so you'd have to stop the
process at some point.

Add the two parts together: 110.0012
Get ready to convert to scientific notation: 110.0012 x 20
Normalize the representation: 1.100012 x 22
This involves moving the radix point so that the integer part is 1.

Convert exponent from base 10 to excess 127 (in binary):
210 = 1000 0001
A convenient way to do this for positive numbers is
to subtract 1, then convert to binary, then flip the MSb.

Finally, put it all together.(This is where you deal with the sign bit).

1 | 1000 0001 | 1000 1000 0000 0000 0000 000

You can either write out all the bits, separated by a bar,
or write them on separate lines, as in:

(Sign Bit) b31 = 1
(Exponent) b30-23 = 1000 0001
(Fraction) b22-0 = 1000 1 --0--
where --0-- would indicate the remaining bits are 0
(and --1-- would indicate the remaining bits are 1).

It's useful to indicate the number of bits or the bit
subscripts to show you know how many bits there are.

(b) The exponent (b30-23) is all 1's.
The fraction (b22-0) is NOT all 0's.
Note: you should indicate which bits (or how many bits)
the exponent and fraction contain.

(c) $\pi = 11.00100100001111110110 11$

5. **(5 marks)** (***Arithmetic Circuits***): A 4-bit carry lookahead adder, which adds two 4-bit numbers, is designed using AND, OR, NOT, NAND, NOR gates only. Assuming that all the inputs are available in both complemented and uncomplemented forms and the delay of each gate is one time unit, what is the overall propagation delay of the adder? Assume that the carry network has been implemented using two-level AND-OR logic.

Solution:

```
 It would take 6 time units.

We know that: Gi=Ai and Bi, Pi=Ai xor Bi

and Si=Pi xor Ci

Also,

C1=G0+P0C0

C2=G1+P1G0+P1P0C0

C3=G2+P2G1+P2P1G0+P2P1P0C0

C4=G3+P3G2+P3P2G1+P3P2P1G0+P3P2P1P0C0

XOR can be implemented in 2 levels; level-1 ANDs and Level-2 OR.
Hence it would take 2 time units to calculate Pi
and Si

The 4-bit addition will be calculated in 3 stages

1. (2 time units) In 2 time units we can compute Gi and Pi in parallel.
2 time units for  Pi since its an XOR operation and 1 time unit for Gi
since its an AND operation.

2. (2 time units) Once Giand Pi are available, we can calculate
the caries, Ci, in 2 time units.

Level-1 we compute all the conjunctions (AND).
Example P3G2,P3P2G1,P3P2P1G0 and P3P2P1P0C0 which
are required for C4

Level-2 we get the carries by computing the disjunction (OR).

3. (2 time units) Finally we compute the Sum in 2 time units,
as its an XOR operation.

Hence, the total is 2 + 2 + 2 = 6 time units.
```