Digital Systems Design (4743/6743)

Prerequisite Example Problems:

- Convert the following **1's complement** number to decimal.
  01101011 = _______107_________
- Convert the following decimal number to **signed magnitude** representation using an **8-bit** format.
  -26 = _____10011010_____
- Convert the following **unsigned number** to decimal.
  $A9 = ____169_____
- Convert the following to fixed 8-bit number representations.

Decimal	Unsigned	Signed Magnitude	1's Complement	2's Complement
53	%00110101	%00110101	%00110101	%00110101
53	--------	--------	$35	--------
-53	--------	%10110101	%11001010	%11001011
-53	--------	$B5	--------	$CB
-54	--------	10110110	-------	-------
22	--------	-------	-------	00010110
56	--------	-------	$38	--------
179	$B3	-------	-------	--------

- Convert the following to fixed **8-bit** number representations. For cells that have -----, they are to be left blank. Make certain to pad your answers to the correct number of bits. You may give your answer in hex or binary - just indicate using the correct prefix symbol.

Decimal	Unsigned	Signed Magnitude	1's Complement	2's Complement
41	$29 %00101001	$29 %00101001	$29 %00101001	$29 %00101001
-41	-----------	$A9 %10101001	$D6 %11010110	$D7 %11010111
	------------	$41	-----------	--------
	------------	----------	------------	$92

- Convert the following values into 12 bit fixed-precision binary or Hex values.

Decimal	Unsigned	Signed Magnitude	1's Complement	2's Complement
138	%0000 1000 1010	%0000 1000 1010	%0000 1000 1010	%0000 1000 1010
138	$08A	$08A	$08A	$08A
-87	% out of range	%1000 0101 0111	%1111 1010 1000	%1111 1010 1001
-87	$ out of range	$857	$FA8	$FA9

- Convert the following 3-digit HEX values into Decimal values.

Decimal	Unsigned	Signed Magnitude	1's Complement	2's Complement
3383	$D37	-------	-------	-------
-1335	-------	$D37	-------	-------
-712	-------	-------	$D37	-------
-713	-------	-------	-------	$D37
312	$138	-------	-------	-------
312	-------	$138	-------	-------
312	-------	-------	$138	-------
312	-------	-------	-------	$138

- Convert the following 8-bit binary values into Decimal values.

Decimal	Unsigned	Signed Magnitude	1's Complement	2's Complement
107	%01101011	-------	-------	-------
107	-------	%01101011	-------	-------
107	-------	-------	%01101011	-------
107	-------	-------	-------	%01101011
222	%11011110	-------	-------	-------
-94	-------	%11011110	-------	-------
-33	-------	-------	%11011110	-------
-34	-------	-------	-------	%11011110

- Convert the following decimal numbers to binary using **two's complement** to represent negative numbers. Use a work length of **5 bits** including sign. Add the two numbers in binary and indicate if an overflow occurs. Show all work in binary.

-12_{10} = %_____10100_____
-11_{10} = %_____10101_____

Final Answer = %____01001____ = $___09___
Overflow (yes or no) ___yes____

- Perform the following operations involving 8-bit **2's complement** numbers. Indicate whether arithmetic overflow occurs.

a. 00110110
+01100101
10011011 Overflow? (Y/N) _Y__

c. 11010101
+11011110
10110011 Overflow? (Y/N) _N__

b. 00110110
- 00100101
00010001 Overflow? (Y/N) _N__

d. 01110101
-11011110
10010111 Overflow? (Y/N) _Y_

- Perform the following operations involving 8-bit **UNSIGNED** numbers. Indicate whether arithmetic overflow occurs. Box the final answer for credit.

a. 00110110
+01100101
10011011 Overflow? (Y/N) _N__

c. 11010101
+11011110
10110011 Overflow? (Y/N) _Y_

b. 00110110
- 00100101
00010001 Overflow? (Y/N) _N___

d. 01110101
-11011110
10010111 Overflow? (Y/N) _Y__

- Perform the following operations involving 4-digit HEX numbers that are in **2's complement** representation. Indicate whether arithmetic overflow occurs. Box the final answer for credit.

35DC
+ 4757
7D33 Overflow? (Y/N) __N___

D738
+3A7F
11B7 Overflow? (Y/N) _N__

200A
- 07B7
1853 Overflow? (Y/N) _N__

65D4
-A325
C2AE Overflow? (Y/N) _Y__

- How can you tell if a hex number is negative?
  Most significant digit is greater than 7

- What is the range of numbers that can be represented using 6 bits?

  Assuming unsigned numbers ___0-63____
  Assuming sign and magnitude representation ___-31 - +31____
  Assuming 2's complement representation ___-32 - +31___

- What is the range of numbers represented with 12 bits.

Unsigned	Sign + Magnitude	1's complement	2's Complement
0 - 4095	(-2047) thru +2047	(-2047) thru +2047	(-2048) thru +2047

- To represent 27 distinct items via digital codes, what is the minimum number of binary digits that I can use?
  5

- Is the equation F(A,B,C) = (A'+B+C')(A'+B') in SOP form (Yes/No) ___no___

- Is B'C' the complement of BC? (Yes/No) ____no____

- Give an example of a function in POS form. __F(A,B,C) = (A'+B+C')(A'+B')__

- Use the consensus theorem to eliminate redundant terms.
  ABC + A'B'C' + ABD + B'C'D + AC'D

  Consensus term for ABC and AC'D is ABD
  Consensus term for A'B'C' and AC'D is B'C'D
  Consensus term for ABD and B'C'D is AC'D
  Eliminate firt two consensus terms – final equation is
  ABC + A'B'C' + AC'D

- Find the complement of the following expression.

  F = A(B'+C) + B + C'D' + C'(D+E)
  F' = __(A' + BC')B'(C + D)(C +D'E')____

- Simplify the following equation to minimum SOP form
  B + C + AD'(B + C)'
  B + C + AD'

- Put F = (A+B+C)(A+B'+C')(A'+B+C') in SOP form.
  F = ΠM(0,3,5) maxterm notation
  Therefore, the minterms must be all of the other terms
  F = Σm(1,2,4,6,7)
  F = A'B'C + A'BC' + AB'C' + ABC + ABC

- Fill in the blank with SOP, POS, Neither
  Z = AB'(C + D')(E'+F'G)
  Z' = A' + B + C'D + EF + EG'

  In the equation above, Z is in ____Neither______ form and Z' is in ____SOP______ form.

- Find an equivalent Sum of Product form for Z

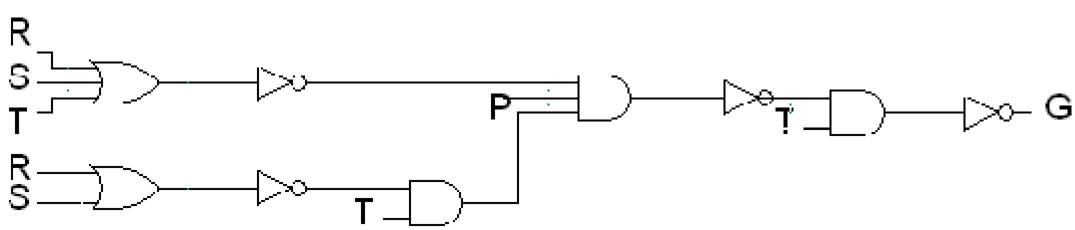
  Z = (A+B')(A+C'+D')(A+B+D)
  Z = [A+B'(C'+D')][A+B+D]
  Z = A+B'(C'+D')(B+D)
  Z = A + B'(B+D)(C'+D')
  Z = A + (B'B+B'D)(C'+D')
  Z = A + (0+B'D)(C'+D')
  Z = A + (B'D)(C'+D')
  Z = A + (B'C'D + B'DD')
  Z = A + (B'C'D + 0)
  Z = A + (B'C'D)

  (X+Y)(X+Z) = X+YZ X=A, Y=B', Z=(C'+D)
  (X+Y)(X+Z)=X+YZ X=A,Y=B'(C'+D'),Z=(B+D)
  Commutative property

- An equivalent POS form of AB + C'DA + C'AE is

  Z = A[B + C'D + C'E] Distributive property
  Z = A[B+C'(D+E)] XY + XZ = X(Y+Z) X=C', Y=D, Z=E
  Z = A[(B+C')(B + D + E)] X + YZ = (X+Y)(X+Z) X=B, Y=C', Z=D+E

- Find the equation for G and simplify. Put a box around your answer.



  G = [[(R+S+T)'•P•[(R+S)'•T]]'•T]'
  G = [(R+S+T)'•P•[(R+S)'•T]] + T'
  G = R'S'T' • P •R'S'• T + T'
  G = PR'R'S'S'T'T + T
  G = 0 + T
  G = T

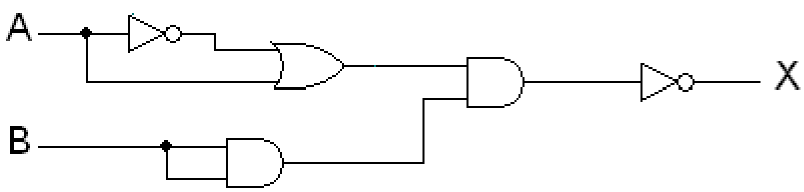
- Write the algebraic equation for Z if Z equals (AC) xor (BC).

  Z = (AC)'(BC) + (AC)(BC)'
  = (A' + C')(BC) + AC(B'+C')
  = A'BC + 0 + AB'C + 0
  = A'BC + AB'C

- Find the inverse of the function Z. (hint: Use DeMorgan's law)

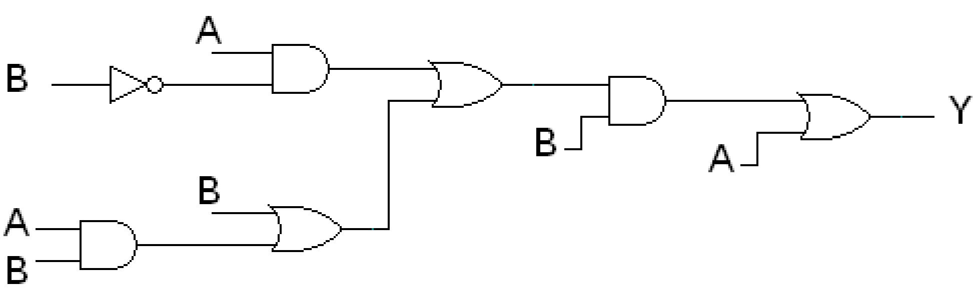
  $Z = AB'(C + D')(E'+F'G)$
  $Z = [AB'(C + D')(E'+F'G)]'$
  $Z' = A' + B + (C+D')' + (E'+F'G)'$
  $Z' = A' + B + C'D + E(F'G)')$
  $Z' = A' + B + C'D + E(F + G')$
  $Z' = A' + B + C'D + EF + EG'$

- Simplify the following circuit:



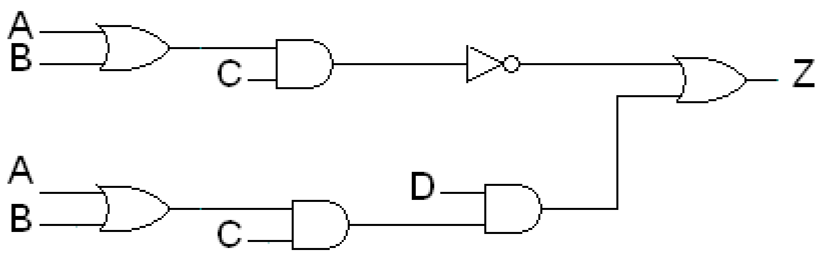
$X = [(A' + A) \cdot (B \cdot B)]'$ use $A' + A = 1$; $B \cdot B = B$
$= [\;1 \cdot B\;]'$ use $1 \cdot B = B$
$= B'$

- Simplify the following circuit:



$Y = [[(A \cdot B') + [(A \cdot B) + B]] \cdot B] + A$
$= (AB' + AB + B)B + A$ writing $A \cdot B'$ as AB'; $A \cdot B$ as AB
$= (AB'B + ABB + BB) + A$ distributing B through first parentheses
$= (\;0 + AB + B) + A$ use $AB'B=A(B'B)=A(0)=0$; $ABB=AB$
$= (\;B\;) + A$ use $AB+B = B$; $0 + B = B$;
$= A + B$ commutative property

- Simplify the following circuit:



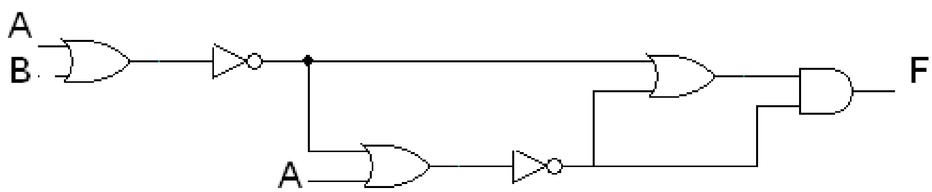
$Z = [(A + B) \cdot C]' + [[(A + B) \cdot C] \cdot D]$

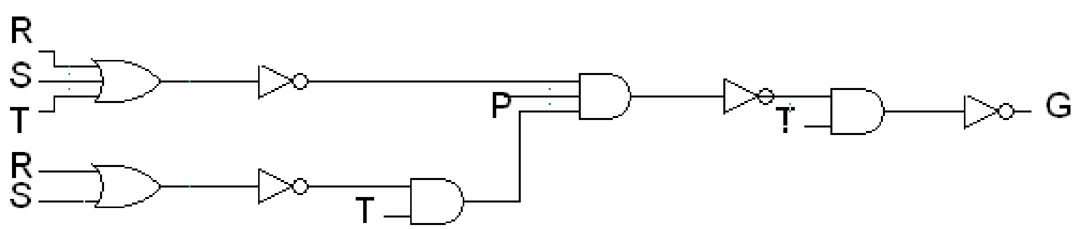
$= [(A + B) \cdot C]' + D$ — use theorem $x + x'y = x + y$

$= (A + B)' + C' + D$ — use DeMorgan's law

$= A'B' + C' + D$ — put equation in SOP form

- Find F and G and simplify:





$F = [[(A + B)' + A]' + (A + B)'] \cdot [(A + B)' + A]'$

$= [(A + B)' + A]'$ — use theorem $(x + y) \cdot x = x$

$= [(A + B)'' \cdot A']$ — use DeMorgan's law

$= (A + B) \cdot A'$ — use indempotent rule $(x'' = x)$

$= AA' + A'B$ — use distributive property

$= 0 + A'B$ — use $x \cdot x' = 0$

$= A'B$

$G = [[(R + S + T)' \cdot P \cdot [(R + S)' \cdot T]]' \cdot T]'$

$= [R'S'T' \cdot P \cdot R'S' \cdot T]' \cdot T]'$

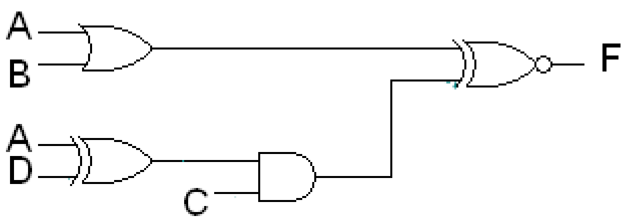
$= [PR'R'S'S'T'T]' \cdot T]'$

$= [[0]' \cdot T]'$

$= [1 \cdot T]'$

$= T'$

5)Write an expression for F and simplify:



The mathematical equation for $z = x$ (XOR) y is $z = xy' + x'y$
The mathematical equation for $z = x$ (XNOR) y is $z = xy + x'y'$

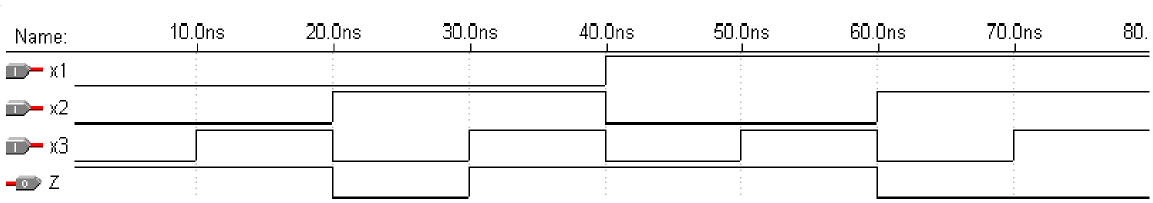
$F = [(A + B) \cdot [(AD' + A'D) \cdot C]] + [(A + B)' \cdot [(AD' + A'D) \cdot C]']$
$= [(A + B)(ACD' + A'CD)] + [(A'B')[(ACD' + A'CD)']]$
$= [AACD' + AA'CD + ABCD' + A'BCD] + \ldots$
$= [ACD' + 0 + ABCD' + A'BCD] + \ldots$
$= [ACD' + A'BCD] + [(A'B')[(ACD')' \cdot (A'CD)']$
$= [ACD' + ABCD' + A'BCD] + [(A'B')[(A'+C'+D)(A+C'+D')]]$
$= \ldots + [A'B'[(0 + A'C' + A'D' + AC' + C' + C'D' + AD + C'D + 0)]$
$= \ldots + [A'B'[(C' + A'D' + AD)]$
$= \ldots + [A'B'C' + A'B'D' + 0]$
$= ACD' + A'BCD + A'B'C' + A'B'D'$

- Implement the following equation in a 2 level NOR-NOR circuit. You can assume that variables and their complements are available as inputs.
  $A'B(C + D + E')(B + D' + E)$
  $= [A + B' + (C+D+E')' + (B+D'+E)']'$

- Give the truth table for a 2-input NOR gate.

A	B	(A + B)'
0	0	1
0	1	0
1	0	0
1	1	0

- For the following timing diagram, write the equation for the function z(x1,x2,x3) in POS form



$Z = (x1 + x2' + x3)(x1' + x2' + x3)(x1' + x2' + x3')$

- Show the truth table for F = (AB)' + C.

A	B	C	(AB)'+C
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

- Design a circuit with output F and inputs x_1, x_0, y_1, and y_0. Let $X = x_1x_0$ be a number, where the four possible values of X, namely 00,01,10, and 11, represent the four numbers 0,1,2,3, respectively. Similarly, let $Y=y_1y_0$ represent another number with the same values. The output F should be 1 if X >= Y.

a) Show the truth table for F.

X1	X0	Y1	Y0	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

b) Write F in minterm notation.

$F(X_1, X_0, Y_1, Y_0) = \Sigma m(0,4,5,8,9,10,12,13,14,15)$

c) Show the minimum SOP equation for F.

$F(X_1, X_0, Y_1, Y_0) = Y_1'Y_0' + X_0Y_1' + X_1Y_1' + X_1X_0 + X_1Y_0'$

d) List all of the prime implicants.

$Y_1'Y_0'$, X_0Y_1', X_1Y_1', X_1X_0, X_1Y_0'

e) List all of the essential prime implicants.

$Y_1'Y_0'$, X_0Y_1', X_1Y_1', X_1X_0, X_1Y_0'

- A switching network has 3 inputs (A,B,C) and one output Z. The output is 1 if the sum of the three inputs is greater than or equal to 2.

a) Find the truth table for Z.

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

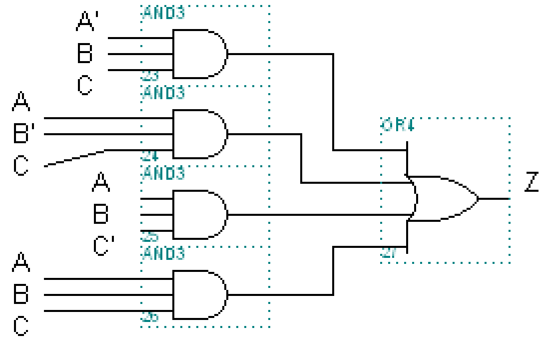
b) Write Z in minterm notation.

$Z = \Sigma m(3,5,6,7)$

c) Write the equation for Z in SOP form.

$Z = A'BC + AB'C + ABC' + ABC$

d) Draw the logic diagram for Z using an 2-level AND-OR network.



- Explain propagation delays (tphl and tplh).

  Propagation delay is the time between a change in an input and a change in the output. It is measured from a 50% change in input signal level to a 50% change in the output signal level.
  Tphl - the propagation delay time from when the input signal changes until the output changes (a high to low Transition on the output)
  Tplh - the propagation delay time from when the input signal changes until the output changes (a low to high transition on the output)

- Complete a truth table for the following function:

  $F(A,B,C,D) = \Sigma m\,(2,4,7,11,12,15) + \Sigma d(3,9)$

A,B,C,D	F
0 0 0 0	0
0 0 0 1	0
0 0 1 0	1
0 0 1 1	X
0 1 0 0	1
0 1 0 1	0
0 1 1 0	0
0 1 1 1	1
1 0 0 0	0
1 0 0 1	X
1 0 1 0	0
1 0 1 1	1
1 1 0 0	1
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1

- Write the above equation in maxterm notation:

$F(A,B,C,D) = \prod M(0,1,5,6,8,10,13,14) + \sum d(2,9,12)$

- What size memory would be used to implement this? Use KxN terminology.
  16 x 1

- Draw the K-map for this function:

  $F(A,B,C,D) = \Sigma m\,(0,1,4,5,6,10,14) + \Sigma d(7,13)$

\AB				
CD\	00	01	11	10
00	1	1	0	0
01	1	1	X	0
11	0	X	0	0
10	0	1	1	1

- Use the above graph to find the minimum SOP equation (circle all terms in the above graph):

  $F(A,B,C,D) = A'B + A'C' + ACD'$

- Draw the same K-map function below and use it to find the minimum POS equation for the compliment of the above function:

\AB				
CD\	00	01	11	10
00	1	1	0	0
01	1	1	X	0
11	0	X	0	0
10	0	1	1	1

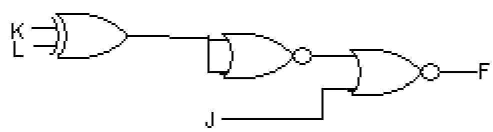
$F'(A,B,C,D) = (A + B')(A + C)(A' + C' + D)$

- Assume you have three inputs (J,K,L) but not their complements. Use only the chips 7486, 7402 (four 2-input XORs, four 2-input NORs) to implement the equation.
  F(J,K,L) = J'KL' + J'K'L

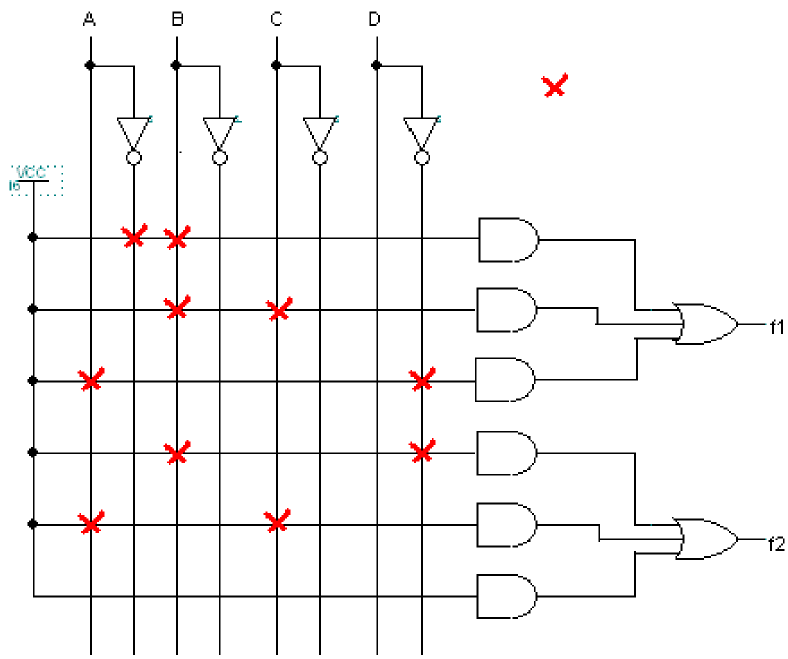
  - Rewrite the equation so it can be easily implemented.

    F(J,K,L) = (KL'+K'L)J'

  - Draw the circuit:



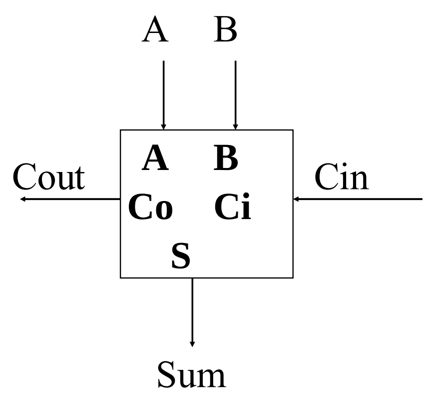
- Fill in the chart below with these equations:
  F1 = A'B + BC + AD'   F2 = BD' + AC



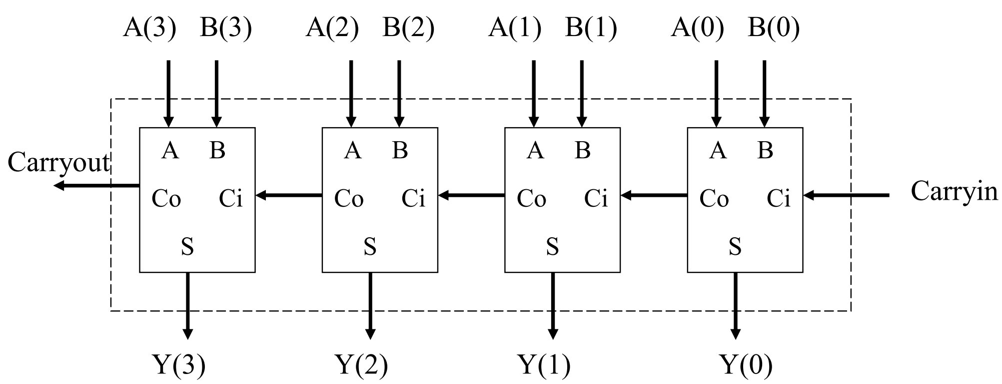
- What does PLD stand for?
  Programmable Logic Device

- What does PAL stand for?
  Programmable Array Logic

- Use the follow Full Adder block to create a 4-bit ripple carry adder using inputs A[3:0], B[3:0], CarryIn and the outputs Y[3:0] and CarryOut. Label all inputs



4 Bit Ripple Carry Adder



- Implement the following equation using VHDL. Complete the entity and architecture in the code below. Do not simplify the equation. Use correct syntax.

$$F(A,B,C) = (A+B')C + C'B$$

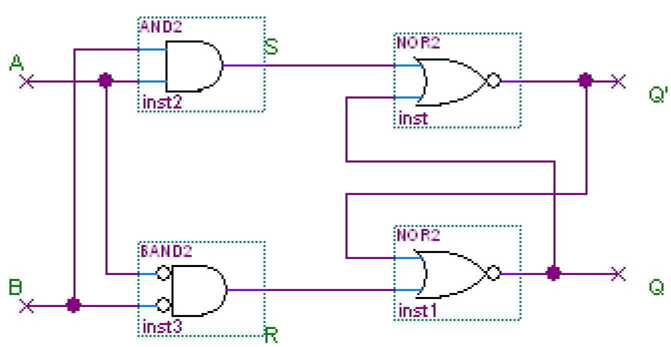
```
library ieee;
use ieee.std_logic_1164.all;

entity test2 is
   port (    A,B,C        : IN std_logic;     -- two dashes is a COMMENT in VHDL
             F            : OUT std_logic
        );
end test2;
architecture boolean of test2 is
begin
        F <= ((A or not B) and C) or (not C and B);
end boolean;
```

- Design a latch with the operation table below using an SR latch (NOR-based). Complete the truth table and draw the circuit.

A	B	Operation	S	R	Q	Q+
0	0	Reset	0	1	X	0
0	1	Hold	0	0	Q	Q
1	0	Hold	0	0	Q	Q
1	1	Set	1	0	X	1

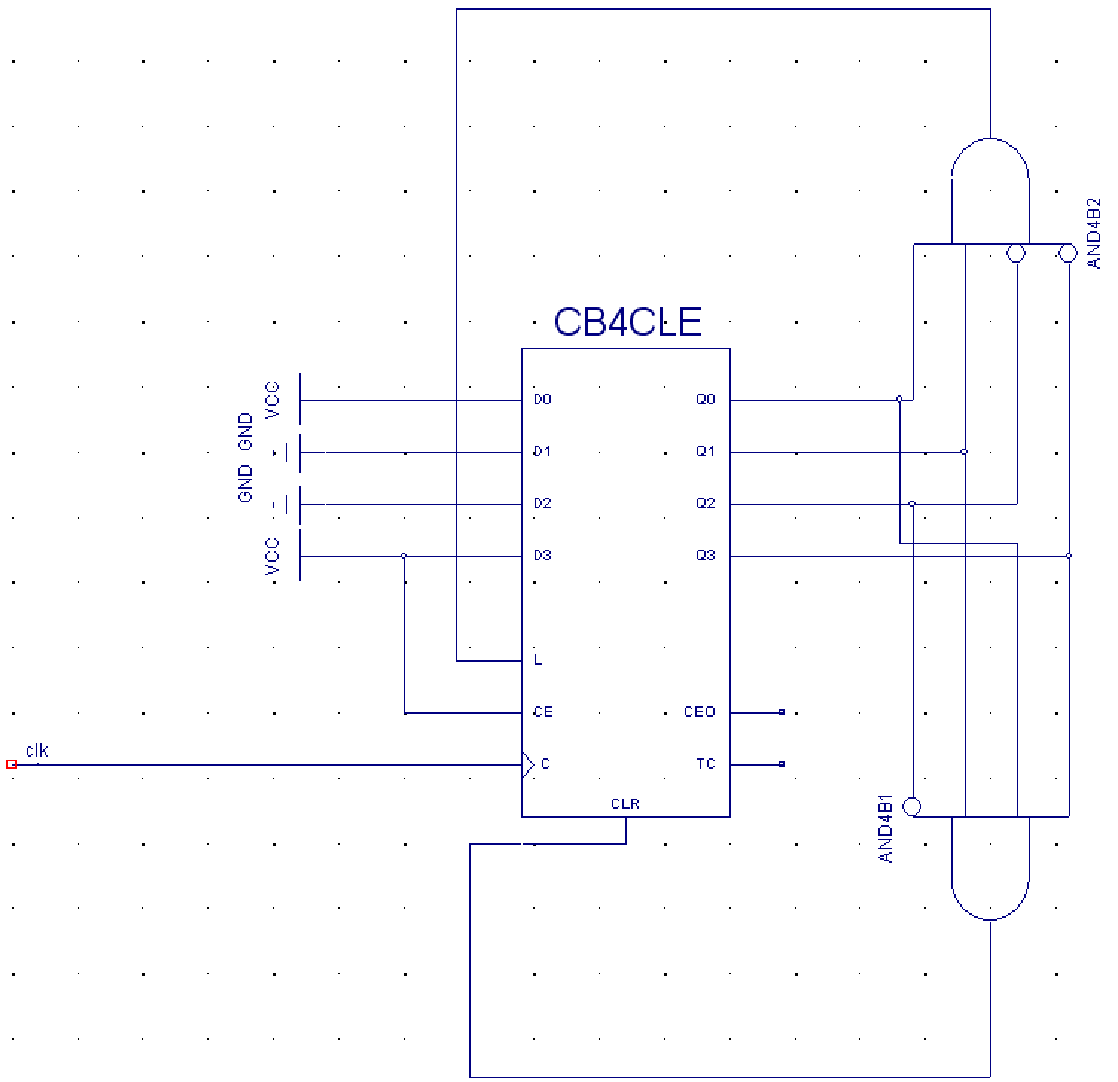
R = A'B'   S=AB



- Design a counter which repeatedly counts in this sequence:

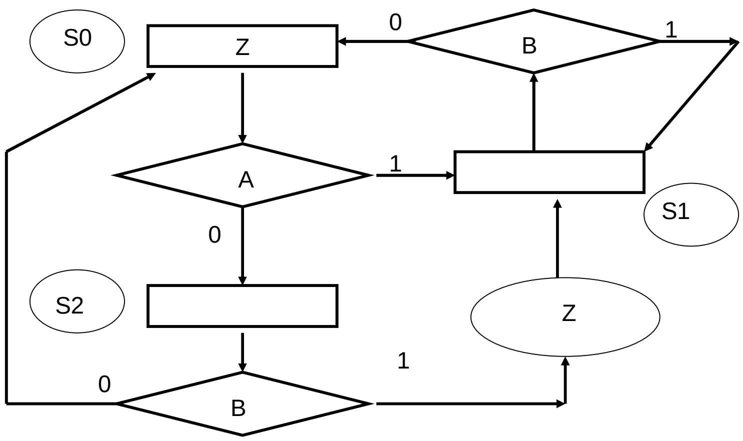
0,1,2,3,9,10,11

Assume you have a clock input. Start with the binary counter below, which has a synchronous reset.



- Use the State Table below to construct an ASM chart. A,B are inputs. Z is the output.

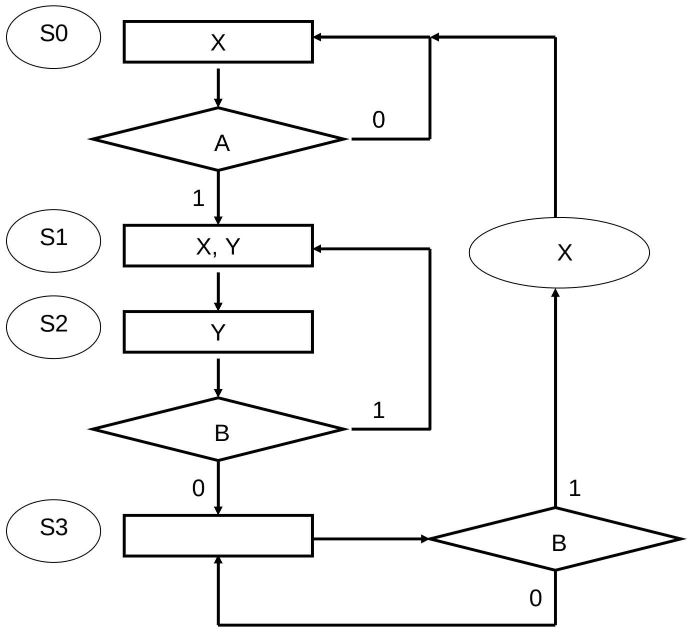
A	B	State	Next State	Z
0	X	S0	S2	1
1	X	S0	S1	1
X	0	S1	S1	0
X	1	S1	S0	0
X	0	S2	S0	0
X	1	S2	S1	1



- Use the state encoding: **S0 = 11, S1 = 10, S2 = 00** to complete the state table from above.

A	B	State	Next State	X
0	X	11	00	1
1	X	11	10	1
X	0	10	10	0
X	1	10	11	0
X	0	00	11	0
X	1	00	10	1

- Use the following ASM chart and state to create equations for a one-hot finite state machine. The state encoding is: **S0 = 0001, S1 = 0010, S2 = 0100, S3 = 1000**.



A	B	Q3Q2Q1Q0	D3D2D1D0	X	Y
0	X	S0	S0	1	0
1	X	S0	S1	1	0
X	X	S1	S2	1	1
X	1	S2	S1	0	1
X	0	S2	S3	0	1
X	0	S3	S3	0	0
X	1	S3	S0	1	0

D3 = Q3B’ + Q2B’

D2 = Q1

D1 = Q0A + Q2B

D0 = Q0A’ + Q3B

X = Q0 + Q1 + Q3B

Y = Q1 + Q2

- Given the ASM below, complete the timing diagram by adding **State** and **Sh** outputs. Assume the state machine starts in S0 and there is no propagation delay.