Question 1

(a) (i) $-64_{10} + 43_{10}$.

In 7 bits, $64_{10} = 1000000_2$. Applying 2's-complement

$$\begin{array}{rr} 1000000 \rightarrow & 0111111 \\ + & 1 \\ \hline & 1000000 \end{array}$$

[Note this is the same as +64, thanks to the asymmetry of 2's-complement representation]

And $+43_{10} = 32 + 8 + 2 + 1 = 0101011_2$

Add:

$$\begin{array}{rr} & 1000000 \\ + & 0101011 \\ \hline & 1101011 \end{array}$$

$\rightarrow$ Negative number, so find magnitude

$$\begin{array}{rr} 1101011 \rightarrow & 0010100 \\ + & 1 \\ \hline & 0010101 \end{array} = 21$$

Therefore, answer $= -21_{10}$.

Overflow not possible.

(ii) $29_{10} + 43_{10}$

From above, $43_{10} = 0101011_2$

And $29_{10} = 16 + 8 + 4 + 1 = 0011101_2$

Add:

$$\begin{array}{rr} & \boxed{01}11111 \\ & 0101011 \\ + & 0011101 \\ \hline & 1001000 \end{array}$$

indicates overflow!

Answer negative, so determine magnitude

$$\begin{array}{rr} 1001000 \rightarrow & 0110111 \\ + & 1 \\ \hline & 0111000 \end{array} = 56_{10}$$

Therefore, answer $= -56_{10}$ (with overflow)

(b) Can't directly add numbers in sign-magnitude when either or both numbers are negative.

Must convert 110110_2 to 6-bit 2's complement

$\underbrace{1}_{\text{negative}}\underbrace{10110}_{22_{10}}$. Convert +22 :

$$010110 \rightarrow \begin{array}{r} 101001 \\ +\quad 1 \\ \hline 101010 \end{array}$$

Add:

$$\begin{array}{r} {\scriptstyle 0111} \\ 101010 \\ 001110 \\ \hline 111000 \end{array}$$

Answer is negative, so find magnitude

$$111000 \rightarrow \begin{array}{r} 000111 \\ +\quad 1 \\ \hline 001000 \end{array}$$

Convert back to sign-magnitude

$$-001000 \rightarrow 101000_2$$

(c) $89_{10} + 84_{10}$ in BCD.

$$\begin{array}{rr} 89_{10} = & 1000\ 1001 \\ 84_{10} = & 1000\ 0100 \\ \hline & 1\ 0000\ \boxed{1101} \\ & 0110 \\ \hline & 1\ \boxed{0001}\ 0011 \\ & 0110 \\ \hline & 1\ 0111\ 0011 \end{array}$$

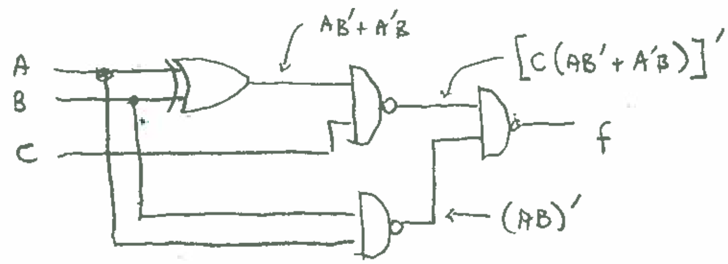
$1101 \rightarrow$ not valid BCD

Also invalid BCD $\rightarrow$ 0001

$1\ 0111\ 0011 \Rightarrow 173_{10}$

<u>Question (2)</u>

(a) For the circuit on the left,



We have $f = \left[[AB'C + A'BC]' \cdot (AB)' \right]'$

DeMorgan's theorem $\quad f = AB'C + A'BC + AB$

$AB(C + C')$

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

$f = AC + BC + AB$

For the circuit on the right,

$g = \left[(AB)' \cdot (AC)' \cdot (BC)' \right]'$

$\;\; = AB + AC + BC \qquad$ SAME!

<u>BONUS</u>: This is the carry bit for a 1-bit full adder.

(b) With $g = AB + AC + BC$

C \ AB	00	01	11	10
0			1	
1		1	1	1

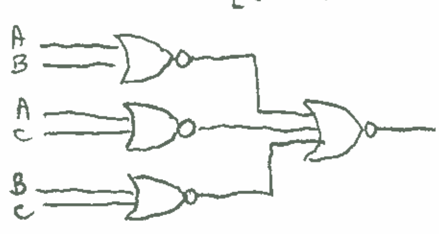
$\Rightarrow$

C \ AB	00	01	11	10
0	0	0		0
1	0			

$g' = A'B' + A'C' + B'C'$

$g = [A'B' + A'C' + B'C']'$

$\;\; = [(A+B)' + (A+C)' + (B+C)']'$



Question 3

(a)

cd \ ab	00	01	11	10
00	0	1	1	X
01	X	X	0	X
11	X	0	0	1
10	0	0	1	1

$f' = bc' + a'c + bd$

so $f = (b'+c)(a+c')(b'+d')$

cd \ ab	00	01	11	10
00	0			X
01	X	X	0	X
11	X	0	0	
10	0	0		

$f' = a'b' + a'c + bd$

so $f = (a+b)(a+c')(b'+d')$

(b)

cd \ ab	00	01	11	10
00		1	1	X
01	X	X		X
11	X			1
10			1	1

$f = ab' + ad' + bc'd'$

cd \ ab	00	01	11	10
00		1	1	X
01	X	X		X
11	X			1
10			1	1

$f = b'd + ad' + bc'd'$

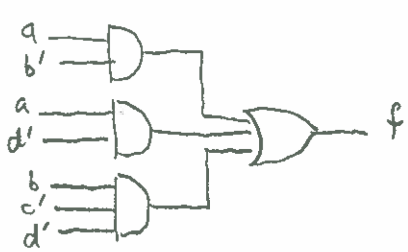
cd \ ab	00	01	11	10
00		1	1	X
01	X	X		X
11	X			1
10			1	1

$f = ab' + ad' + a'bc'$

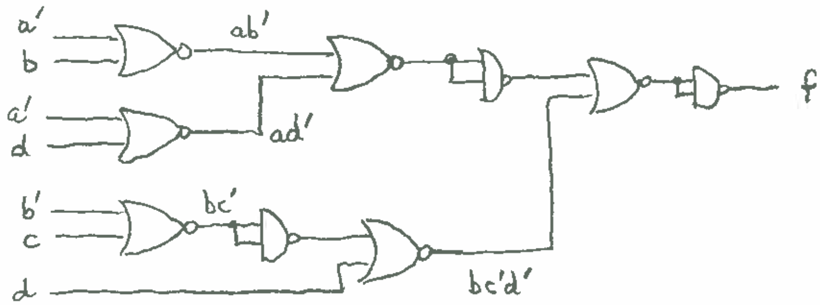
cd \ ab	00	01	11	10
00		1	1	X
01	X	X		X
11	X			1
10			1	1

$f = b'd + ad' + a'bc'$

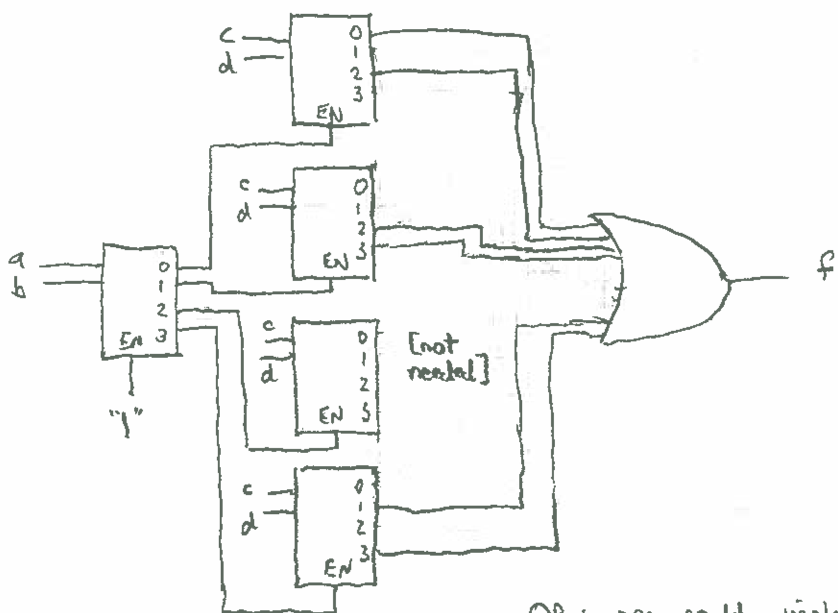
(c) Let's choose $f = ab' + ad' + bc'd'$



Using 2-input NOR gates

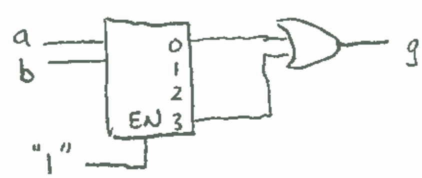


(d) Decoder implementation; need $f' = \sum m(0, 2, 6, 7, 13, 15)$

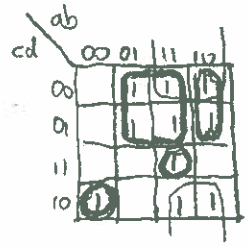


OR: one could implement f and use a NOR gate.

(e) Using the don't-cares,

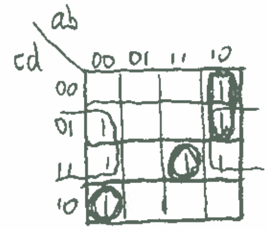


Question 4

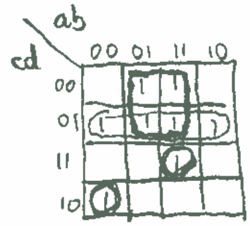


$F = ad' + \underline{bc'} + \underline{ab'c'} + \underline{a'b'cd'} + \underline{abcd}$

shared terms



$G = b'd + \underline{a'b'cd'} + \underline{abcd} + \underline{ab'c'}$



$H = c'd + \underline{bc'} + \underline{a'b'cd'} + \underline{abcd}$

Total number of shared products : 4
Total number of non-shared products : 3
Number of OR gates : 3

→ Total number of gates = 10 with 32 inputs