1. a. 00100010

b. 1101110

c. 10000000

d. Not enough bits! (8-bit two's complement range is from -128 to 127)

e. 00000000

2. a. 74 = 01001010
96 = 01100000

$$\begin{array}{r} 01001010 \\ +\ 01100000 \\ \hline 10101010 \end{array}$$ (Overflow)

b. 19 = 00010011
54 = 00110110
−54 = 11001010

$$\begin{array}{r} 00010011 \\ +\ 11001010 \\ \hline 11011101 \end{array}$$

c. 55 = 00110111
−55 = 11001001
73 = 01001001
−73 = 10110111

$$\begin{array}{r} 11001001 \\ +\ 10110111 \\ \hline 10000000 \end{array}$$ (Overflow)

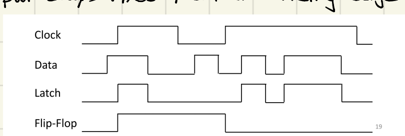
d. 55 = 00110111
73 = 01001001

$$\begin{array}{r} 00110111 \\ +\ 01001001 \\ \hline 10000000 \end{array}$$ (Overflow)

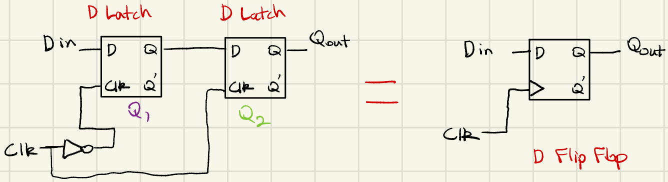
e. 32 = 00100000
44 = 00101100

$$\begin{array}{r} 00100000 \\ +\ 00101100 \\ \hline 01001100 \end{array}$$

3. a. A D Latch has an output that follows the data input whenever the clock is high.
A D Flip Flop captures the data input into its output when the clock goes high, and retains its value until the next rising edge of the clock.
The difference is illustrated by the diagram below, which shows that with a varying data input, the latch also varies throughout the duration of the clock being high, whereas the D Flip Flop's output stays fixed from one rising edge to the next.

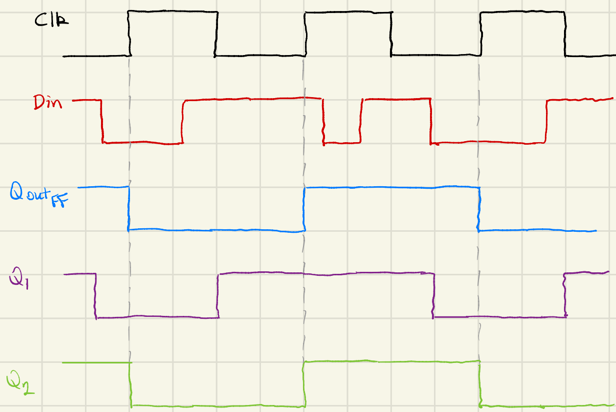


3b. We can make a D Flip Flop out of 2 D Latches by connecting them in series with opposite clock inputs



A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

We can verify this with a timing diagram:



As we can see, the outputs of Q_{DFF} and Q_2 match up.

4. $f(x_1,x_2,x_3) = \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_2\bar{x}_3 + x_1x_2x_3$

$= \bar{x}_1x_3 + x_1\bar{x}_3 + x_1x_2x_3$

$= x_3(\bar{x}_1 + x_1x_2) + x_1\bar{x}_3$

$= x_3(\bar{x}_1 + x_2) + x_1\bar{x}_3$

We will "not" the whole function and then "not" it again at the end:

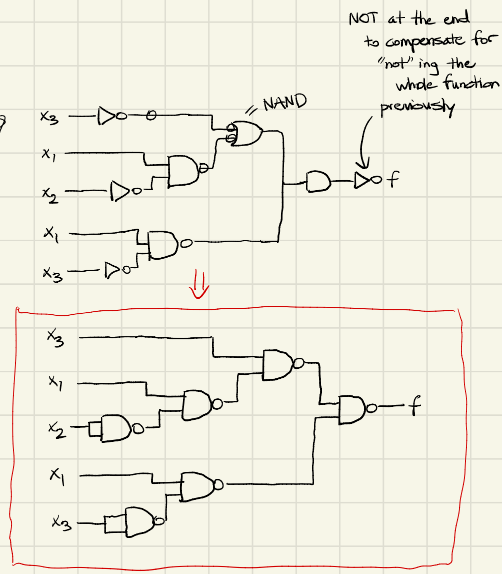
$\Rightarrow \overline{x_3(\bar{x}_1+x_2) + x_1\bar{x}_3}$

$= \overline{x_3(\bar{x}_1+x_2)} \cdot \overline{x_1\bar{x}_3}$

$= \left[\bar{x}_3 + \overline{(\bar{x}_1+x_2)}\right] \overline{x_1\bar{x}_3}$

$= \left[\bar{x}_3 + \bar{\bar{x}}_1\bar{x}_2\right] \overline{x_1\bar{x}_3}$

$= \left[\bar{x}_3 + x_1\bar{x}_2\right] \overline{x_1\bar{x}_3}$



5. $-17 = 11101111 =$ Multiplicand (00010001 = Negative Multiplicand)

$-10 = 11110110 =$ Multiplier

	Product before shift	Product after shift
Cycle 1	0000 0000 1111 0110 0	0000 0000 0111 1011 0
Cycle 2	0000 0000 0111 1011 0 +0001 0001 0001 0001 0111 1011 0	0000 1000 1011 1101 1
Cycle 3	0000 1000 1011 1101 1	0000 0100 0101 1110 1
Cycle 4	0000 0100 0101 1110 1 +1110 1111 1111 0011 0101 1110 1	1111 1001 1010 1111 0
Cycle 5	1111 1001 1010 1111 0 +0001 0001 0000 1010 1010 1111 0	0000 0101 0101 0111 1
Cycle 6	0000 0101 0101 0111 1	0000 0010 1010 1011 1
Cycle 7	0000 0010 1010 1011 1	0000 0001 0101 0101 1
Cycle 8	0000 0001 0101 0101 1	0000 0000 1010 1010 1

Final Product: 0000 0000 1010 1010

6. −17 = 11101111 = Multiplicand

−10 = 11110110 ⇒ flip sign, then flip product at the end ⇒ 00001010
= Multiplier

	Product before shift	Product after shift
Cycle 1	0000 0000 0000 1010	0000 0000 0000 0101
Cycle 2	1110 1111 0000 0101	1111 0111 1000 0010
Cycle 3	1111 0111 1000 0010	1111 1011 1100 0001
Cycle 4	1111 111 1111 1011 1100 0001 +1110 1111 1110 1010 1100 0001	1111 0101 0110 0000
Cycle 5	1111 0101 0110 0000	1111 1010 1011 0000
Cycle 6	1111 1010 1011 0000	1111 1101 0101 1000
Cycle 7	1111 1101 0101 1000	1111 1110 1010 1100
Cycle 8	1111 1110 1010 1100	1111 1111 0101 0110
	Negate Product:	0000 0000 1010 1010