

1) Number representation

a) Convert the following binary numbers to unsigned decimal numbers (e.g., “1000” = 8):

100	4
0000000000000001	1
111111	63
101010	42

b) Convert the following numbers to a 5-bit 2’s complement binary representation (e.g. 3 = “00011” and −3 = “11101”):

9	01001
15	01111
−16	10000
−5	11011

c) Convert the following hexadecimal numbers to binary and octal (e.g. AA = “1010 1010” in binary or “252” in octal):

hexadecimal	binary	octal
FF	1111 1111	377
F0A2	1111 0000 1010 0010	170242
0F100	0000 1111 0001 0000 0000	0170400
100	0001 0000 0000	0400

For part c), leading 0’s can be omitted for binary (e.g. “1111 0001 0000 0000” instead of ‘0000 1111 0001 0000 0000’ is fine) or octal (e.g. “170400” instead of “0170400” is fine).

2) Two-level logic minimization

$$F(A, B, C, D) = \sum m(0, 3, 6, 8, 9)$$

$$d(A, B, C, D) = \sum m(11, 12, 15)$$

	CD	00	01	11	10
AB					
00		1	0	1	0
01		0	0	0	1
11		x	0	x	0
10		1	1	x	0

All Primes

	CD	00	01	11	10
AB					
00		1	0	1	0
01		0	0	0	1
11		x	0	x	0
10		1	1	x	0

Minimal Solution 1

	CD	00	01	11	10
AB					
00		1	0	1	0
01		0	0	0	1
11		x	0	x	0
10		1	1	x	0

Minimal Solution 2

a) Identify all the prime and essential prime implicants.

	Prime	Essential
1-00	$A\bar{C}\bar{D}$	
-000	$\bar{B}\bar{C}\bar{D}$	$\bar{B}\bar{C}\bar{D}$
100-	$A\bar{B}\bar{C}$	
10-1	$A\bar{B}D$	
1-11	ACD	
-011	$\bar{B}CD$	$\bar{B}CD$
0110	$\bar{A}BC\bar{D}$	$\bar{A}BC\bar{D}$

b) Find the minimum two-level logic implementation.

$$\begin{aligned}
 F &= \bar{B}\bar{C}\bar{D} + \bar{B}CD + \bar{A}BC\bar{D} + A\bar{B}\bar{C} \text{ OR} \\
 &= \bar{B}\bar{C}\bar{D} + \bar{B}CD + \bar{A}BC\bar{D} + A\bar{B}D
 \end{aligned}$$

3) Multi-level logic minimization

Given following logic equations, minimize the number of literals, e.g. by using common subexpressions, Boolean rules, etc. You can introduce new immediate equations for common sub-expressions if it helps to reduce the number of literals. Put a box around your final answer, and indicate the number of literals in your final answer.

$$\begin{aligned}M &= A\bar{D}\bar{E}\bar{F} + B\bar{D}\bar{E}\bar{F} + C\bar{D}\bar{E}\bar{F} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}E + \bar{A}\bar{B}\bar{C}F \\N &= A + B + C + D + E + F + GHK + GH \\P &= AD + BD + CD + AE + BE + CE + AF + BF + CF + G + \bar{G} + H + K\end{aligned}$$

Answer:

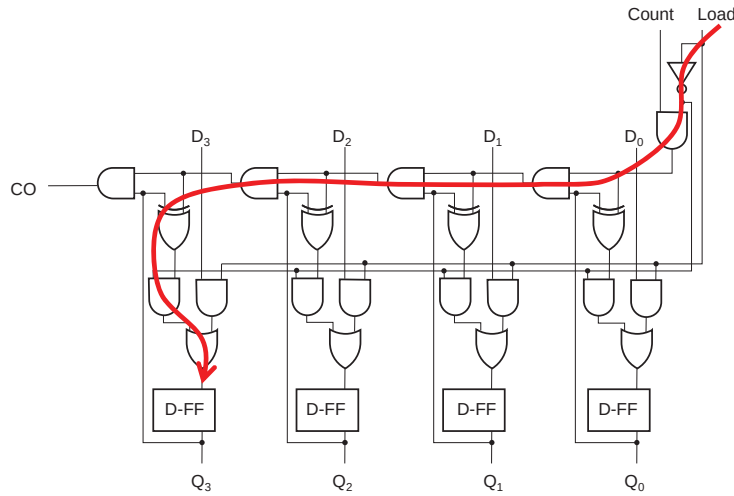
$$\begin{aligned}X &= A + B + C \\Y &= D + E + F \\M &= A\bar{D}\bar{E}\bar{F} + B\bar{D}\bar{E}\bar{F} + C\bar{D}\bar{E}\bar{F} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}E + \bar{A}\bar{B}\bar{C}F \\&= (A + B + C)\bar{D}\bar{E}\bar{F} + \bar{A}\bar{B}\bar{C}(D + E + F) \\&= X\bar{Y} + \bar{X}Y \\N &= A + B + C + D + E + F + GHK + GH \\&= X + Y + GH \\P &= AD + BD + CD + AE + BE + CE + AF + BF + CF + G + \bar{G} + H + K \\&= AD + BD + CD + AE + BE + CE + AF + BF + CF + \mathbf{1} + H + K \\&= 1\end{aligned}$$

Final answer: **14 literals**

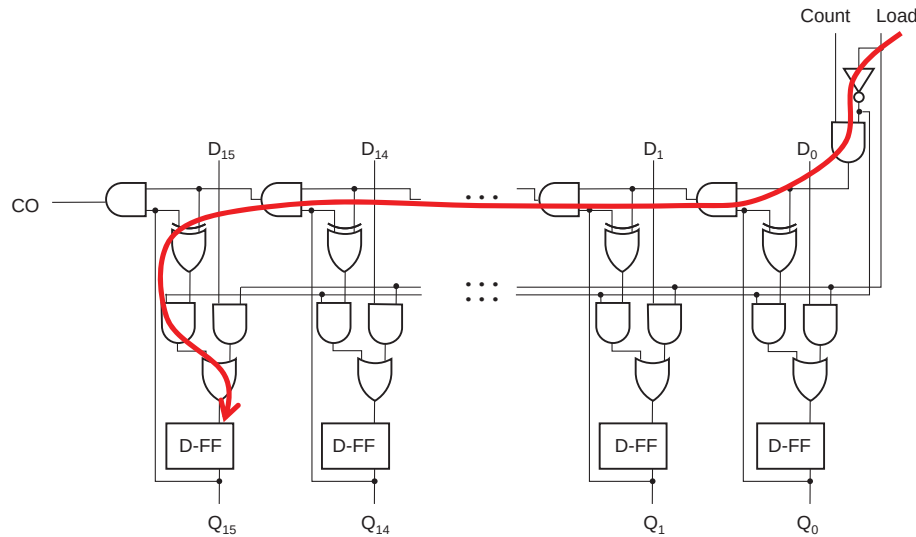
4) Clock period analysis

Consider the following 4-bit counter design, which loads the counter (Q_3, Q_2, Q_1, Q_0) with data (D_3, D_2, D_1, D_0) when “Load = 1”; otherwise, when “Count = 1”, the counter (Q_3, Q_2, Q_1, Q_0) gets incremented. Assume all gates delays are **1 ns** (e.g., AND, OR, XOR, INV). Assume the positive edge-triggered flip-flop delay is $T_{delay_{FF}} = 1 \text{ ns}$, and setup time is also $T_{setup_{FF}} = 1 \text{ ns}$.

a) What is the **minimum clock period** that would satisfy these delays? Clock period = **9 ns**



b) What is the **minimum clock period** for a 16-bit version of the same counter? Clock period = **21 ns**



Generally, 2 gate delays for INV→AND for Load/Count, plus $(N - 1)$ AND gate delays for carry chain, plus 3 gate delays for XOR→AND→OR at the last stage, and plus set-up time for last FF. Therefore, Delay = $2 + (N - 1) + 3 + 1$.

5) Sequential logic implementation

Consider the following Moore machine with initial state A. (“*” means don’t care.)

$S_2 S_1 S_0$	$S_2^+ S_1^+ S_0^+$		F
	$J = 0$	$J = 1$	
A	D	B	1
B	B	A	*
C	C	F	0
D	C	B	0
E	B	F	1
F	A	E	1

Assume the following state encoding.

State	$S_2 S_1 S_0$
A	000
B	010
C	100
D	001
E	011
F	101

Fill-out the following K-Maps.

		S_2^+			
		00	01	11	10
S_2	S_1				
00		0	0	0	1
01		0	0	1	0
11		x	x	x	x
10		1	1	0	0

		S_1^+			
		00	01	11	10
S_2	S_1				
00		0	1	1	0
01		1	0	0	1
11		x	x	x	x
10		0	0	1	0

		S_0^+			
		00	01	11	10
S_2	S_1				
00		1	0	0	0
01		0	0	1	0
11		x	x	x	x
10		0	1	1	0

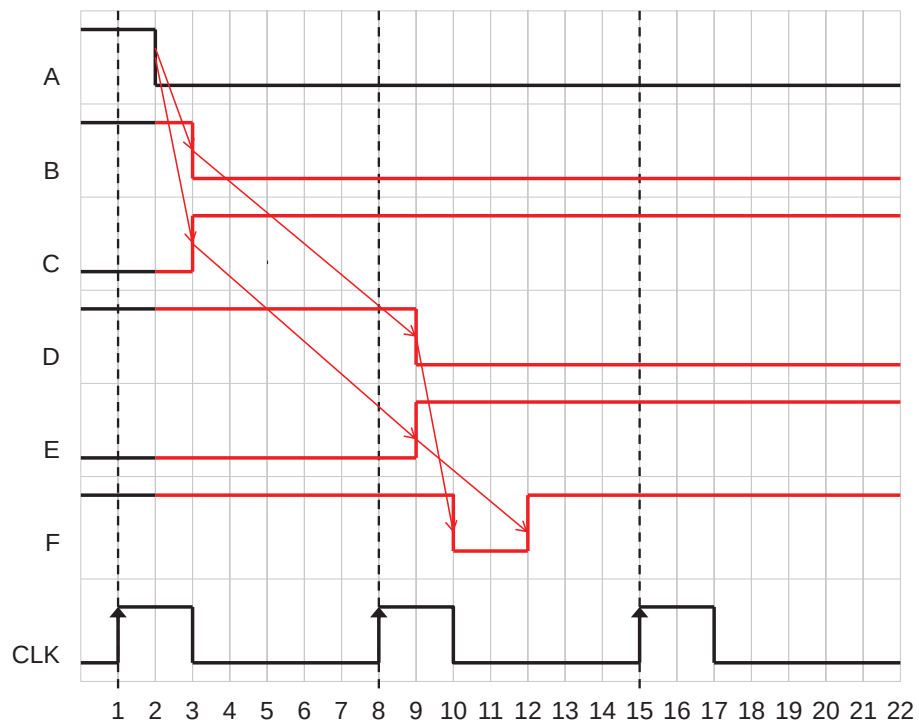
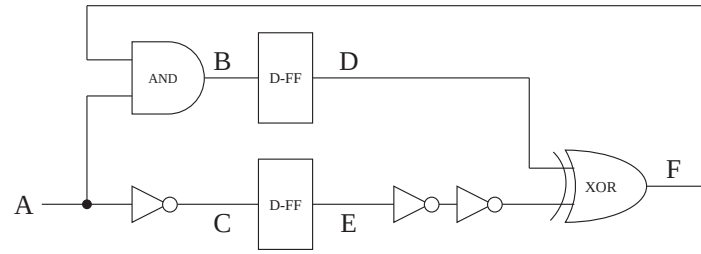
		F	
		0	1
S_2	S_1		
00		1	0
01		x	1
11		x	x
10		0	1

6) Timing analysis

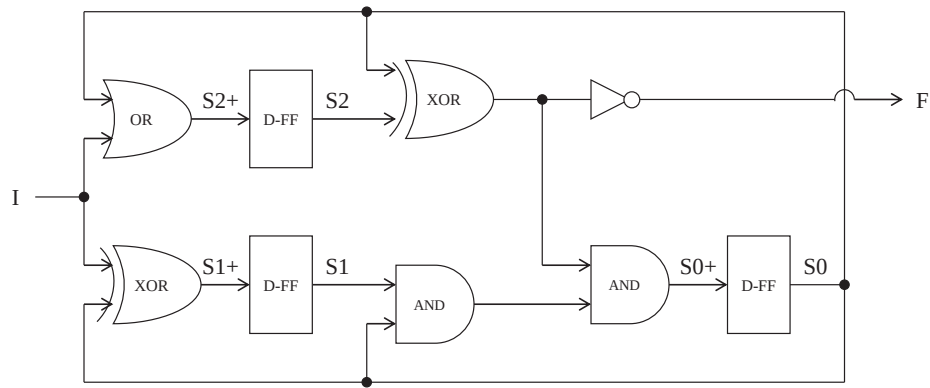
Consider the following sequential logic diagram.

Assume all gate delays are **1 ns** (AND, XOR, and INV). Assume the positive edge-triggered flip-flop delay is $T_{delay_{FF}} = \mathbf{1\ ns}$, and setup time is also $T_{setup_{FF}} = \mathbf{1\ ns}$.

Fill out the rest of the timing diagram for A , B , C , D , E and F .



7) Reverse engineer sequential circuit to finite state machine



a) Fill out the following functions.

$$\begin{aligned}
 F &= \overline{S_2 \oplus S_0} = \bar{S}_2 \oplus S_0 = S_2 \oplus \bar{S}_0 = S_2 \bar{S}_0 \\
 S_2+ &= I + S_0 \\
 S_1+ &= I \oplus S_0 \\
 S_0+ &= (S_2 \oplus S_0)S_1S_0
 \end{aligned}$$

b) Assume initial state is **000**, fill out the following state table. **Important: Only include reachable states in the state table.**

$S_2S_1S_0$	$S_2^+S_1^+S_0^+$		F
	$I = 0$	$I = 1$	
000	000	110	1
001	110	100	0
010	000	110	1
011	111	101	0
100	000	110	0
101	110	100	1
110	000	110	0
111	110	100	1

Complete Truth Table

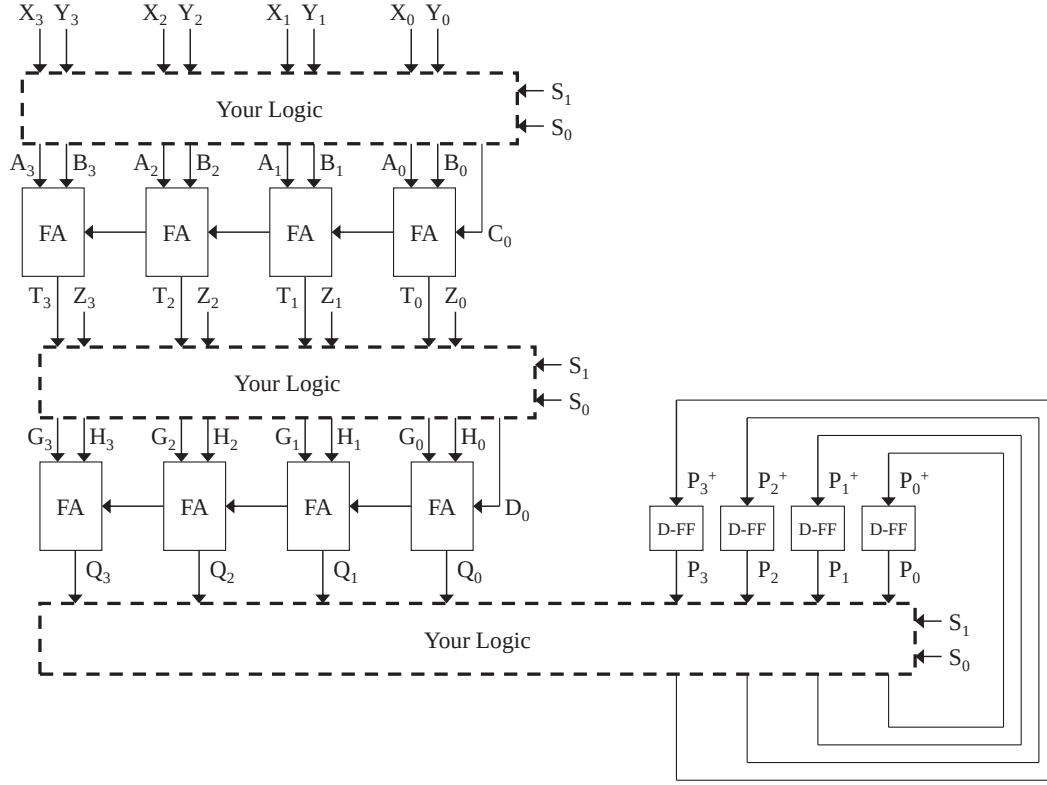
$S_2S_1S_0$	$S_2^+S_1^+S_0^+$		F
	$I = 0$	$I = 1$	
000	000	110	1
110	000	110	0

Just Reachable States

c) With initial state = **000**, what states, if any, are **not** reachable? 001, 010, 011, 100, 101, 111

8) Datapath/registers

Consider the following diagram.



The external data inputs are $X_3X_2X_1X_0$, $Y_3Y_2Y_1Y_0$, and $Z_3Z_2Z_1Z_0$, and the external control signals are S_1S_0 . We would like to implement the following operations:

S_1S_0	Operations
00	$P^+ = X + Y - (Z_2, Z_1, Z_0, Z_3)$
01	$P^+ = X - Y + (Z_0, Z_3, Z_2, Z_1)$
10	$P^+ = (P_2, P_1, P_0, P_3)$
11	$P^+ = (P_0, P_3, P_2, P_1)$

where “+” means “addition”, “−” means “subtraction”, X means $X_3X_2X_1X_0$, Y means $Y_3Y_2Y_1Y_0$, and P^+ means $P_3^+P_2^+P_1^+P_0^+$.

To implement the above computations, please describe:

a) the logic required to implement A_2 , B_2 , and C_0 as functions of S_1S_0 , $X_3X_2X_1X_0$, and $Y_3Y_2Y_1Y_0$,

$$A_2 =$$

$$B_2 =$$

$$C_0 =$$

b) the logic required to implement G_2 , H_2 , and D_0 as functions of S_1S_0 , $T_3T_2T_1T_0$, and $Z_3Z_2Z_1Z_0$,

$$G_2 =$$

$$H_2 =$$

$$D_0 =$$

c) and the logic required to implement P_2^+ as a function of S_1S_0 , $Q_3Q_2Q_1Q_0$, and $P_3P_2P_1P_0$.

$$P_2^+ =$$

You should minimize each logic equation. No need to worry about “overflow” problems or “overflow” logic.

Answers:

$$A_2 = X_2$$

$$B_2 = S_0 \oplus Y_2$$

$$C_0 = S_0$$

$$G_2 = T_2$$

$$H_2 = \bar{S}_0\bar{Z}_1 + S_0Z_3$$

$$D_0 = \bar{S}_0$$

$$P_2^+ = \bar{S}_1Q_2 + S_1(\bar{S}_0P_1 + S_0P_3)$$