University of California, Santa Barbara

Department of Electrical and Computer Engineering

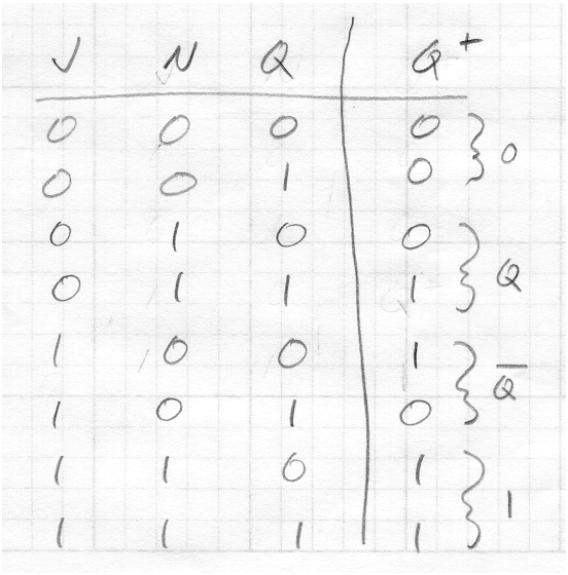
ECE 152A – Digital Design Principles

Homework #4 – Solution

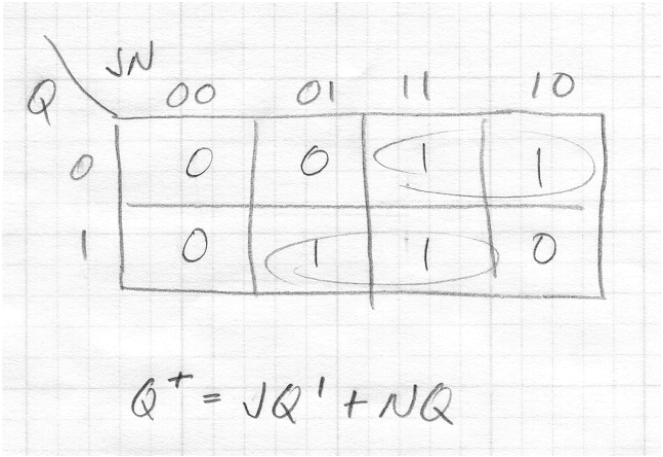
Problem #1.

A JN flip flop is constructed by (internally) complementing the K input to an otherwise normal JK flip flop. For the JN flip flop, derive the following:

1. The characteristic table



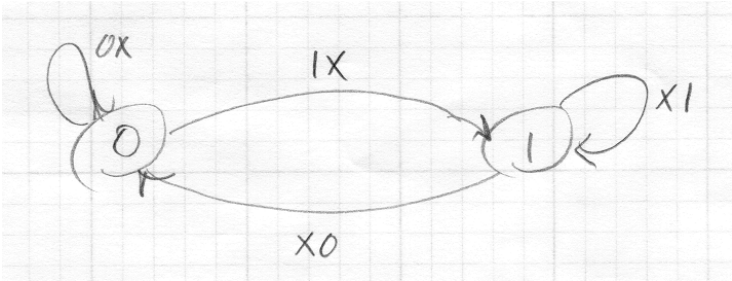
2. The characteristic equation



3. The state table

	NS			
PS	JN= 00	01	11	10
0	0	0	1	1
1	0	1	1	0

4. The state diagram



5. The excitation table

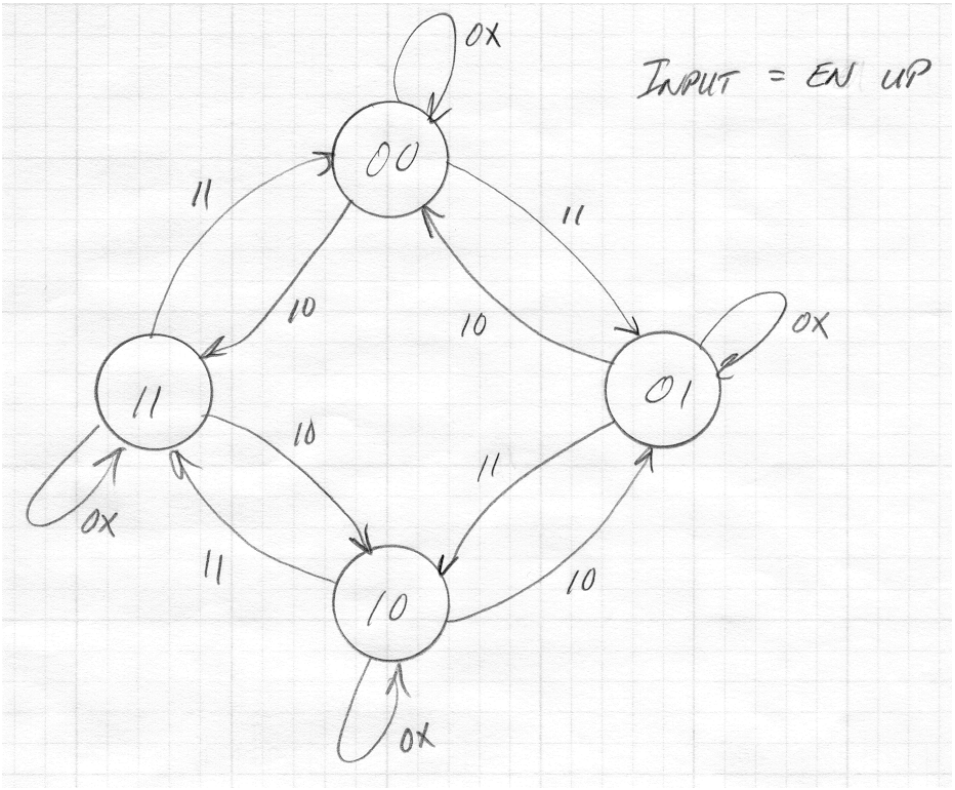
Q	Q+	J	N
0	0	0	X
0	1	1	X
1	0	X	0
1	1	X	1

Problem #2.

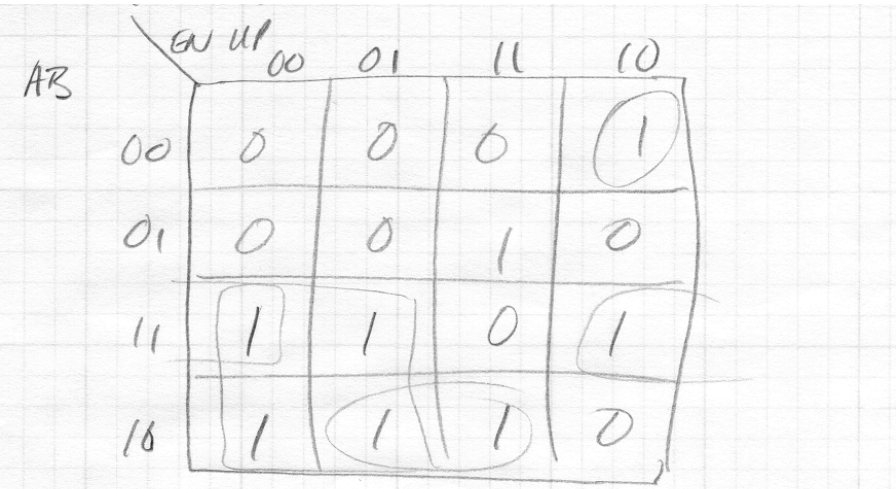
Design a 2-bit, binary up/down counter. The counter has 2 inputs, up_down and enable. The truth table below defines the operation of the counter:

enable	up down	operation
0	0	hold count
0	1	hold count
1	0	decrement count
1	1	increment count

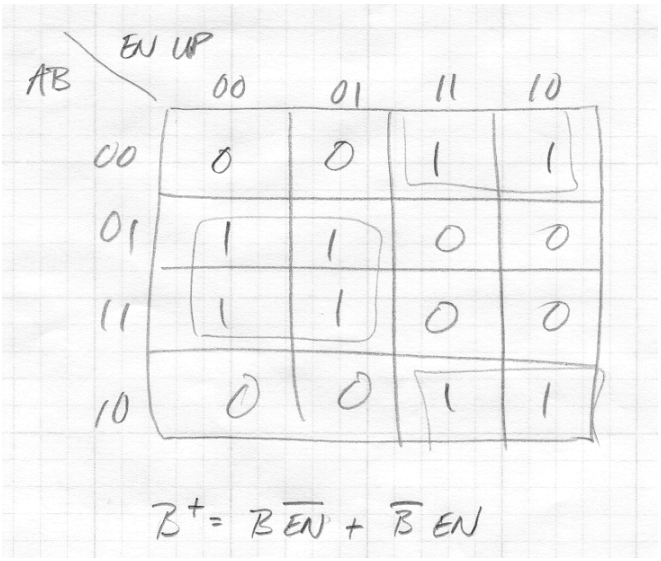
Use positive edge triggered JK flip flops. In your answer, include (1) a state diagram, (2) a state table, (3) a next state map and (4) all Kmaps used in determining flip flop inputs



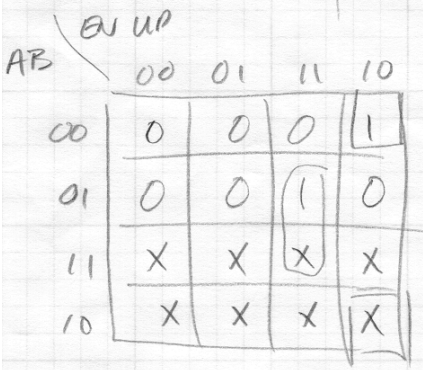
A	B	EN	UP	A^+	B^+	
0	0	0	0	0	0	HOLD
0	0	0	1	0	0	
0	0	1	0	1	1	+1
0	0	1	1	0	1	-1
0	1	0	0	0	1	HOLD
0	1	0	1	0	1	
0	1	1	0	0	0	-1
0	1	1	1	1	0	+1
1	0	0	0	1	0	HOLD
1	0	0	1	1	0	
1	0	1	0	1	1	-1
1	0	1	1	1	1	+1
1	1	0	0	1	1	HOLD
1	1	0	1	1	1	
1	1	1	0	1	0	-1
1	1	1	1	0	0	+1



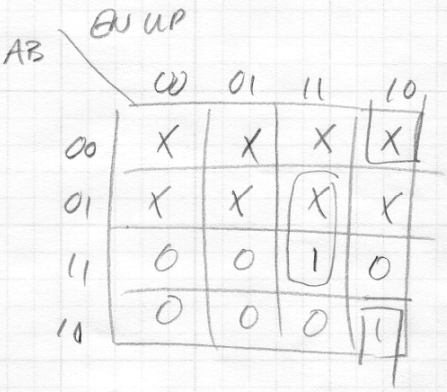
$$A^+ = A\overline{EN} + A\overline{B}\,UP + AB\,\overline{UP} + \overline{A}\,\overline{B}\,EN\,\overline{UP}$$



Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

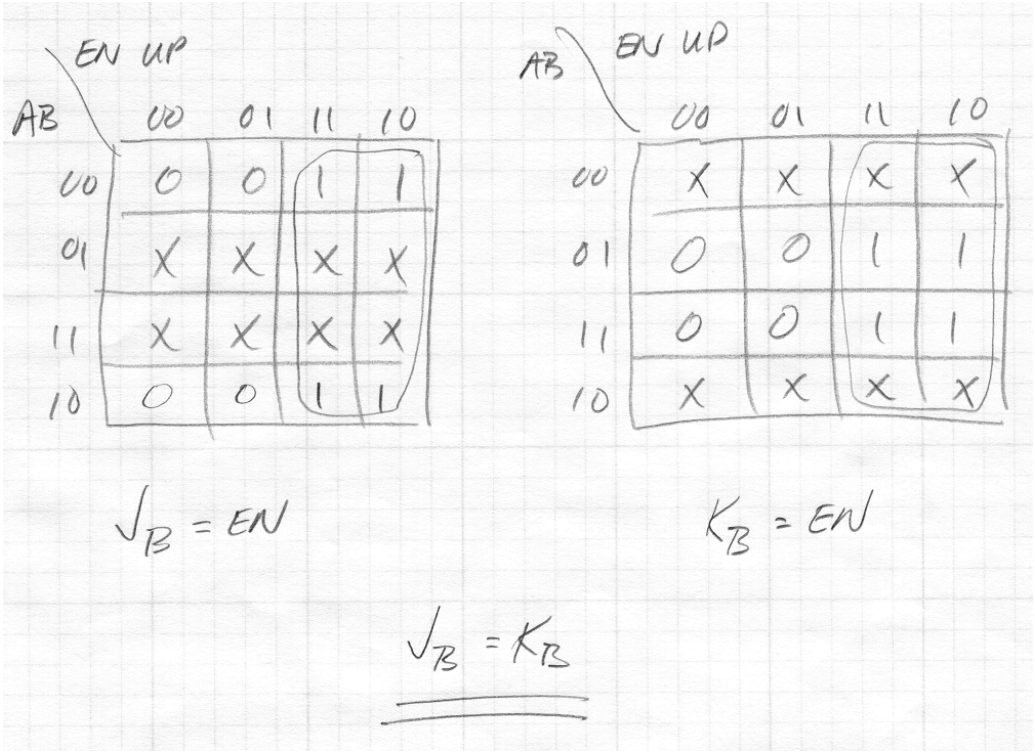


$J_A = EN \cdot UP \cdot B + \overline{B} \cdot EN \cdot \overline{UP}$



$K_A = B \cdot EN \cdot UP + \overline{B} \cdot EN \cdot \overline{UP}$

$J_A = K_A$



Problem #3.

Design a three bit counter with a single input called ***mode***. The counter counts in binary if the mode bit is zero, and counts in gray code if the mode bit is one (recall the three bit gray code is 000, 001, 011, 010, 110, 111, 101, 100). The mode bit can change at any time during the count sequence and your counter should begin counting in the new mode on the next clock input.

Design the counter using JK flip flops. You don't have to draw the logic diagram, specifying the J and K inputs to each flip flop is sufficient. Use A, B and C as the state variable names.

Include (1) a state table, (2) next state maps and (3) K-maps for all flip flop inputs.

			MODE = 0			MODE = 1		
A	B	C	A	B	C	A	B	C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	1	1	0
0	1	1	1	0	0	0	1	0
1	0	0	1	0	1	0	0	0
1	0	1	1	1	0	1	0	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	1	0	1

xA \ BC	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	0	1	1	1
10	0	0	0	1

A^+

xA \ BC	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	0	1
10	0	1	1	1

B^+

xA \ BC	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	0	0	1	1
10	1	1	0	0

C^+

Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

xA \ BC	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	0	0	0	1

$J_A = \bar{x}BC + xB\bar{C}$

xA \ BC	00	01	11	10
00	X	X	X	X
01	0	0	1	0
11	1	0	0	0
10	X	X	X	X

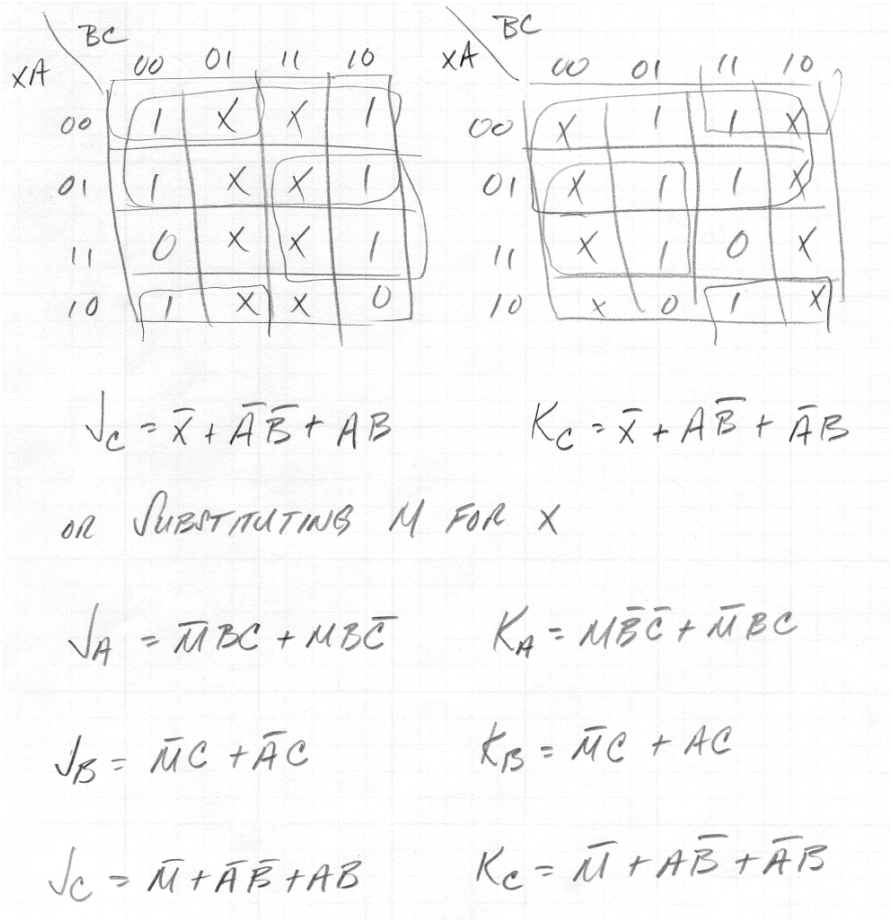
$K_A = x\bar{B}\bar{C} + \bar{x}BC$

xA \ BC	00	01	11	10
00	0	1	X	X
01	0	1	X	X
11	0	0	X	X
10	0	1	X	X

$J_B = \bar{x}C + \bar{A}C$

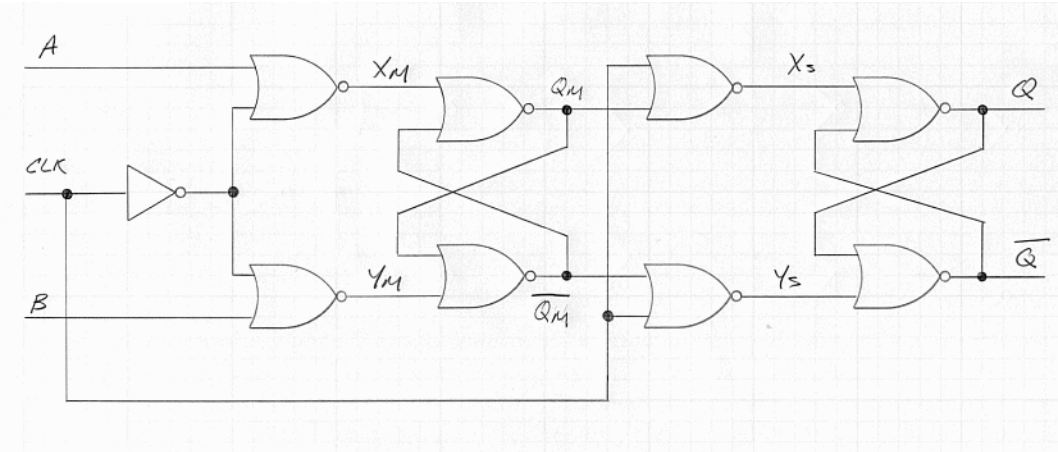
xA \ BC	00	01	11	10
00	X	X	1	0
01	X	X	1	0
11	X	X	1	0
10	X	X	0	0

$K_B = \bar{x}C + AC$



Problem #4.

Consider the flip flop illustrated below:



1. Would this flip flop function as a positive (rising) or negative (falling) edge

triggered flip flop (and why)?

NEGATIVE (OR FALLING) EDGE TRIGGERED BECAUSE SLAVE IS ACTIVE ON LOW PHASE OF CLOCK

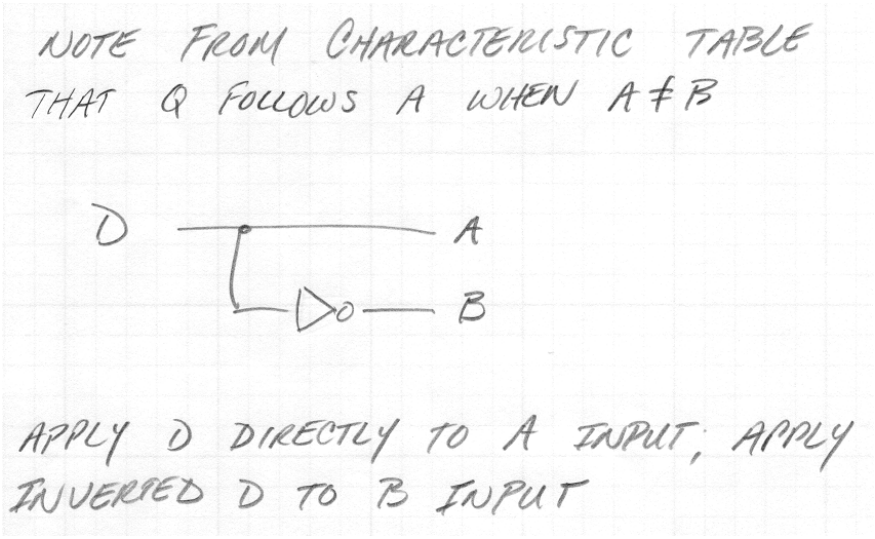
2. Construct the characteristic table for this flip flop.

CHARACTERISTIC TABLE

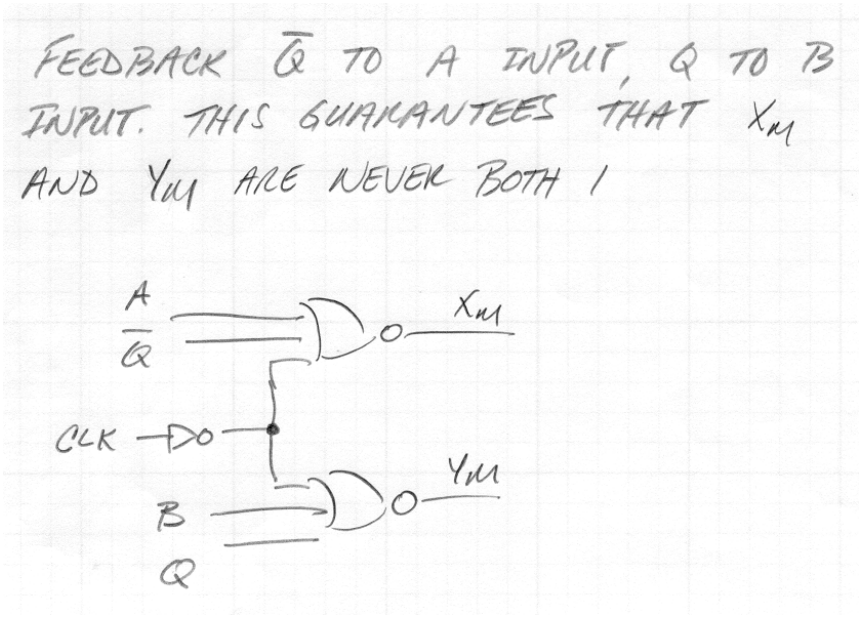
NOTE: MASTER IS ACTIVE ON HIGH PHASE OF CLOCK; A = B = 0 NOT ALLOWED BECAUSE THIS WOULD RESULT IN $X_M = Y_M = 1$.

A	B	Q	Q^+	
0	0	0		NOT ALLOWED
0	0	1		
0	1	0	0	RESET
0	1	1	0	
1	0	0	1	SET
1	0	1	1	
1	1	0	0	HOLD
1	1	1	1	

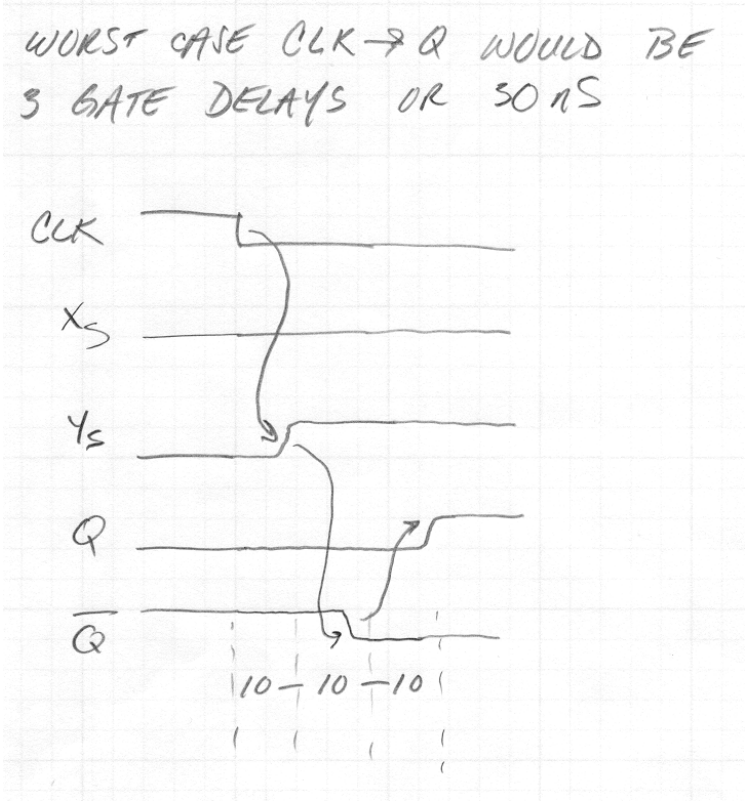
3. What modifications are necessary to transform this into a D flip flop (the output Q should take the value of the single input D after the active clock edge)?



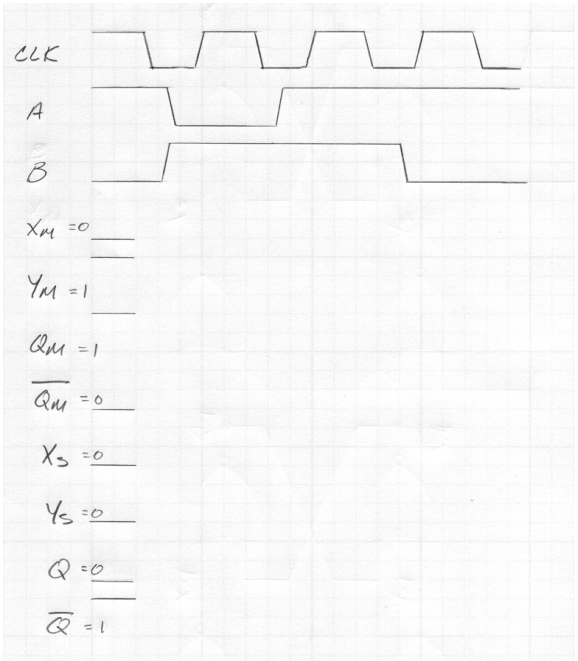
4. What modifications are necessary to transform this into a JK flip flop (the outputs Q and Q' should toggle after the active clock edge when A and B = 0)?

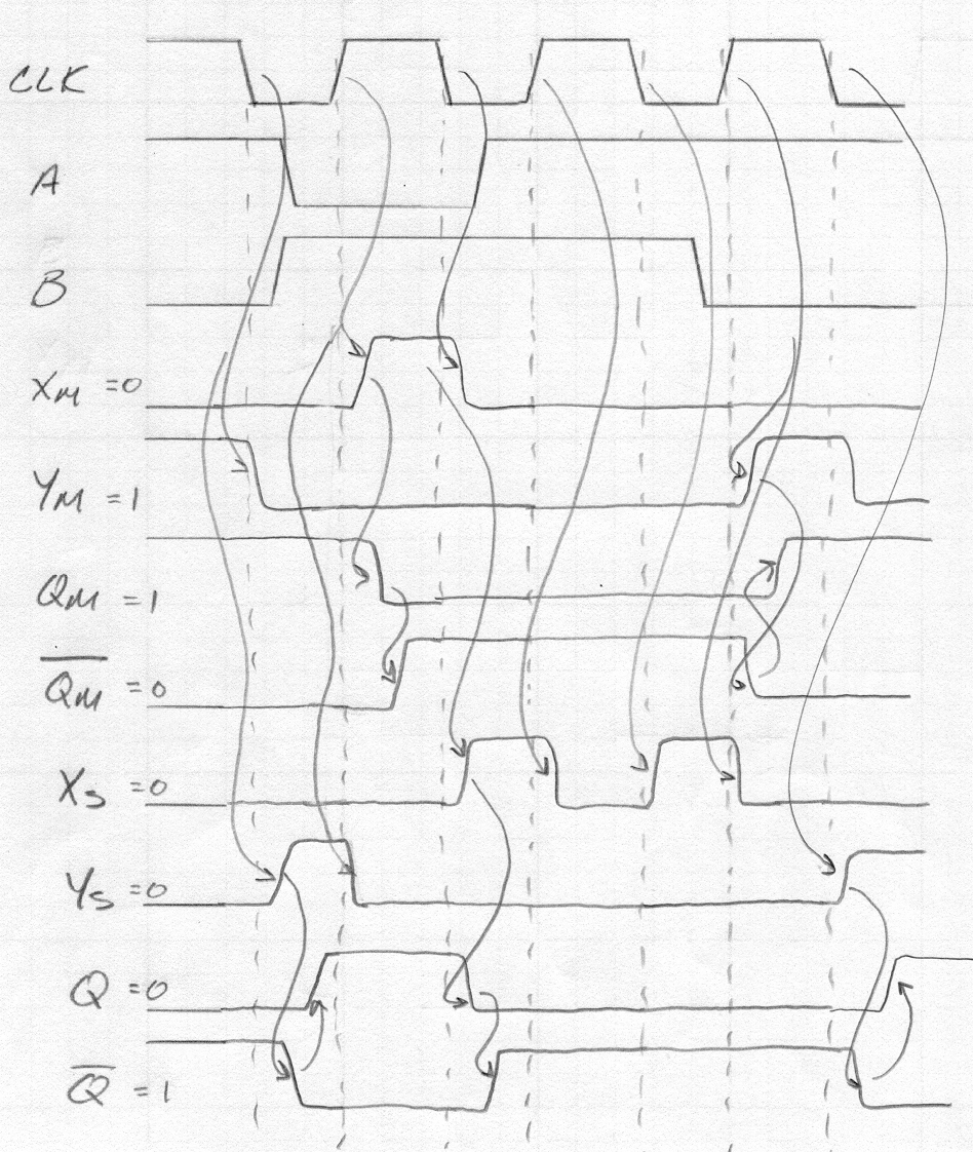


5. If all the gates have a propagation delay of 10 ns (both t_{PLH} and t_{PHL}), what would the worst case CLK to Q delay be?



6. Complete the timing diagram on the following page. Include arrows indicating the relationship and order of signal transitions. Assume the clock period is much greater than the gate delays. The initial conditions are noted on the timing diagram.





Problem #5.

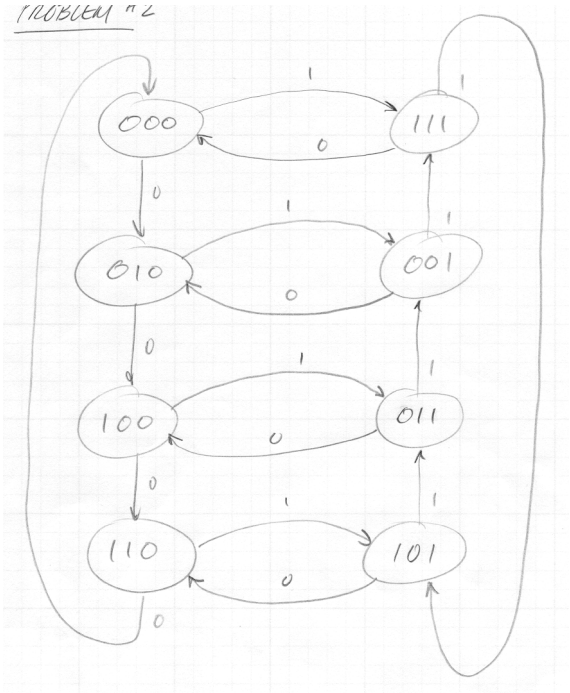
In this problem you are to design a 3-bit counter having a single control input: x. When x is 0, the counter counts up in even numbers (0,2,4,6,0,...). When x is 1, the counter counts down in odd numbers (7,5,3,1,7,...).

When the input changes, the count sequence should change appropriately from the current count. For example, if the current count is 4 and x changes from 0 to 1, the next count should be 3. If the current count is 5 and x changes from 1 to 0, the next count should be 6, etc.

Implement your design using T flip flops. Note that the state diagram is the most important part of the design. Any error in the state diagram will ripple throughout the design.

Include the following:

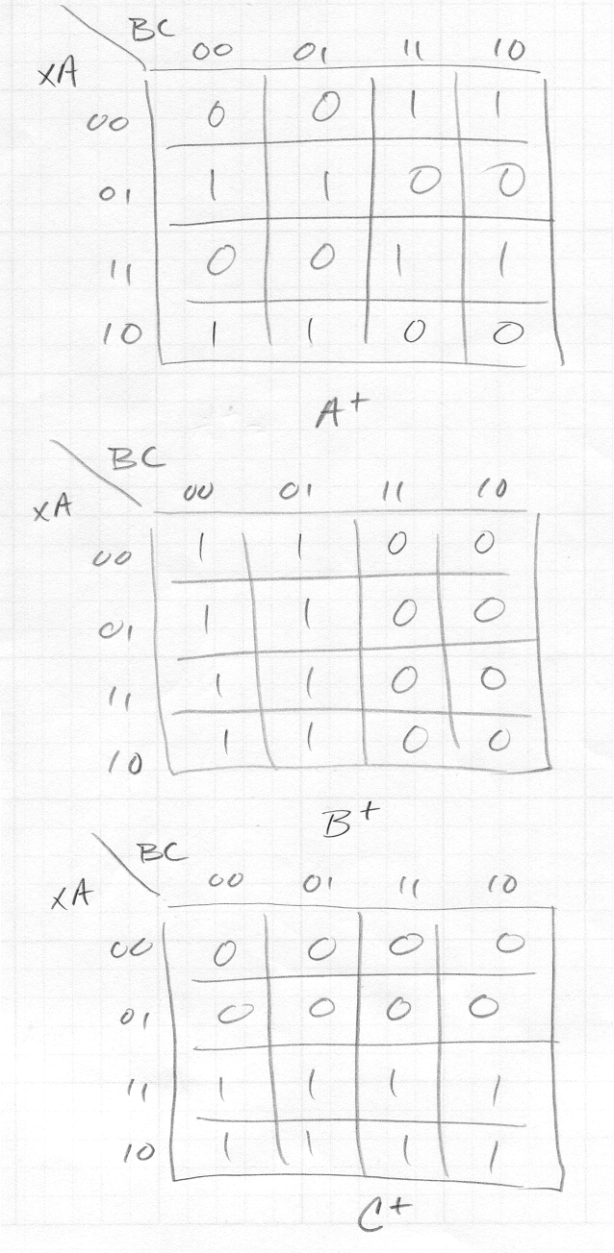
1. A state diagram



2. A state table

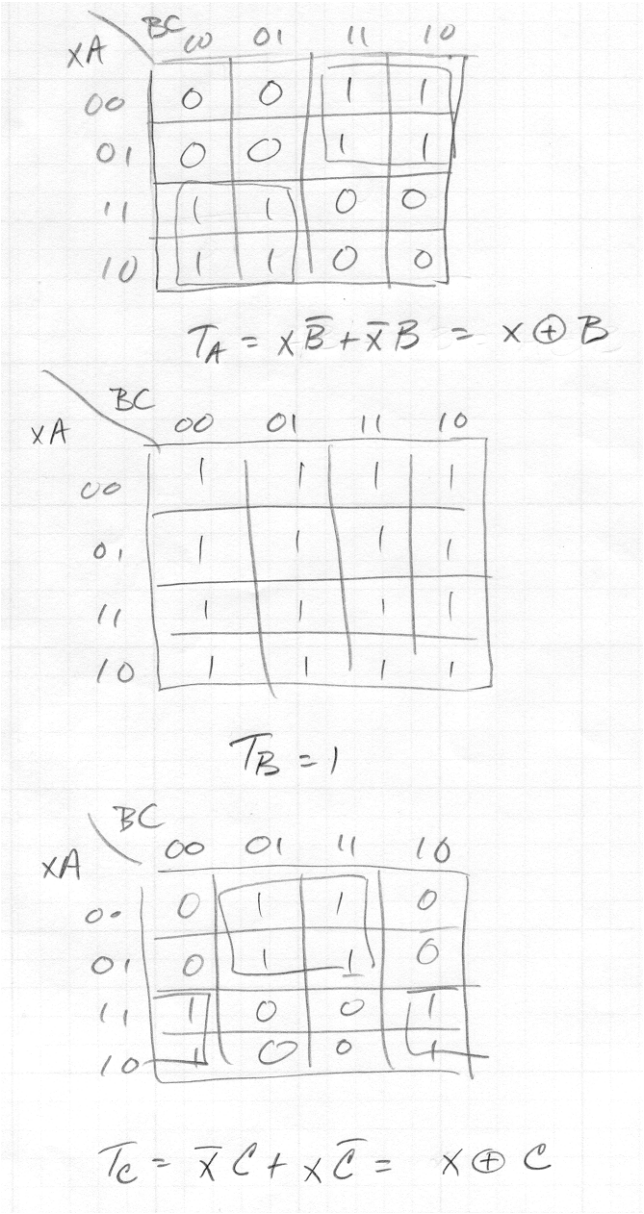
PS			NS x=0			NS x=1		
A	B	C	A^+	B^+	C^+	A^+	B^+	C^+
0	0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	1	1
0	1	0	1	0	0	0	0	1
0	1	1	1	0	0	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	1	0	0	1	1
1	1	0	0	0	0	1	0	1
1	1	1	0	0	0	1	0	1

3. Next state maps for the three flip flops



4. Karnaugh maps for the three T inputs

5. Simplified equations for the three T inputs


BC
xA
00 01 11 10
00 0 0 1 1
01 0 0 1 1
11 1 1 0 0
10 1 1 0 0
$T_A = x\bar{B} + \bar{x}B = x \oplus B$
BC
xA
00 01 11 10
00 1 1 1 1
01 1 1 1 1
11 1 1 1 1
10 1 1 1 1
$T_B = 1$
BC
xA
00 01 11 10
00 0 1 1 0
01 0 1 1 0
11 1 0 0 1
10 1 0 0 1
$T_C = \bar{x}C + x\bar{C} = x \oplus C$

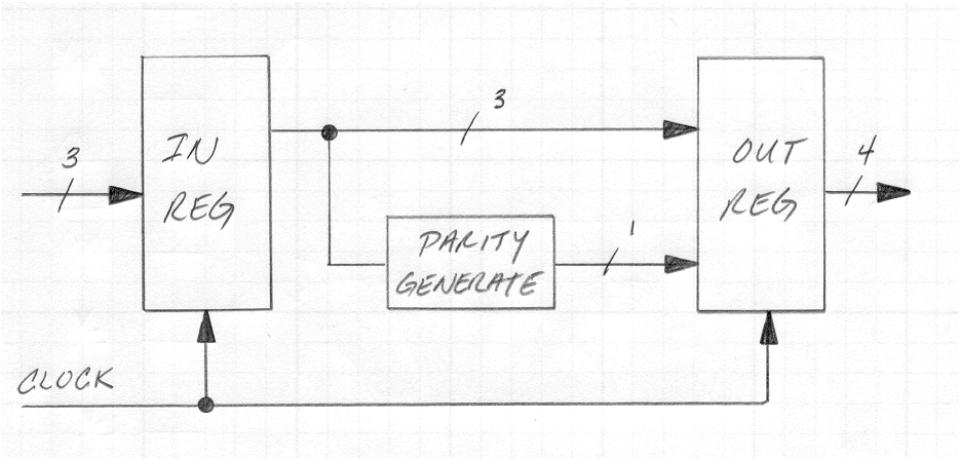
Problem #6.

This problem deals with the minimum clock period (or maximum frequency) of a digital system which generates a parity bit for a 3 bit data word. A parity bit is added to a data word to allow for single bit error detection and either even or odd parity can be implemented. In even parity, the total number of bits (including the parity bit) is an even number.

The truth table below illustrates the generation of an even parity bit for a 3-bit data word.

A	B	C	Even Parity Bit
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

The block diagram for the complete digital system is given below:



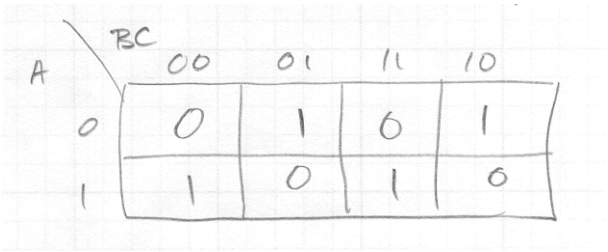
The parity generate block is the implementation of the combinational circuit defined in the truth table on the previous page. Based on that truth table, design the circuit using any combination of 7410's (3 input NAND gates), 7420's (4 input NAND gates) or 7486 (2 input XOR gates). Pinouts and switching characteristics for these devices are included on the following pages.

The input and output registers are to be implemented using 7474 D flip flops. The pinout and switching characteristics for this device are also included on the

following pages. The specific characteristics for the 7474 are circled (other devices are also described on this table).

Include the following in your answer:

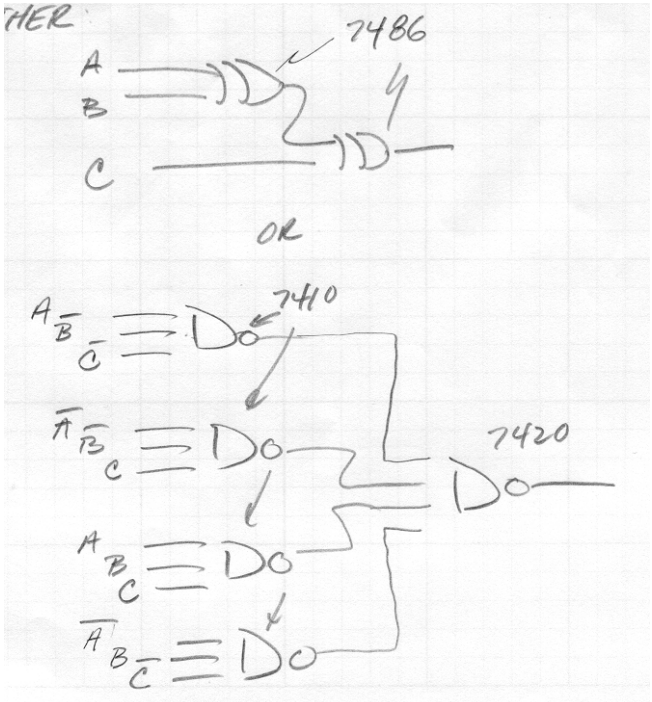
1. A Karnaugh map for the parity generate circuitry.



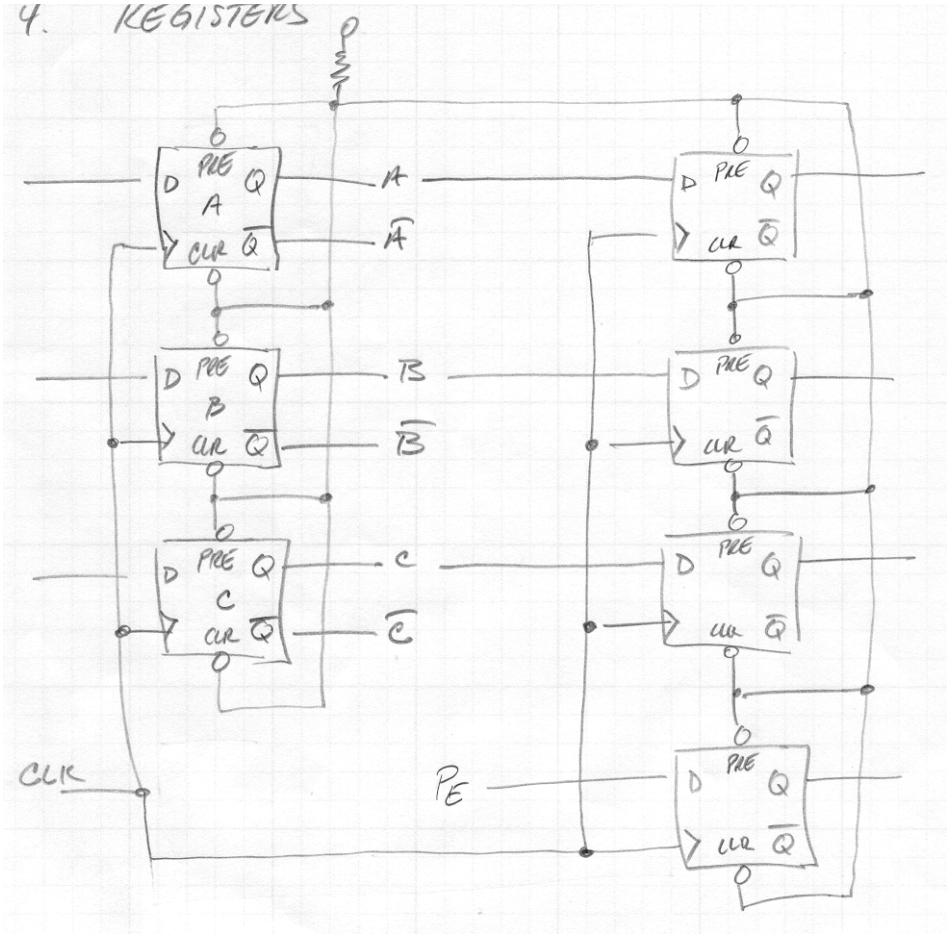
2. The minimized Boolean equation for the parity generate circuitry.

$$P_E = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$$
$$= A \oplus B \oplus C$$

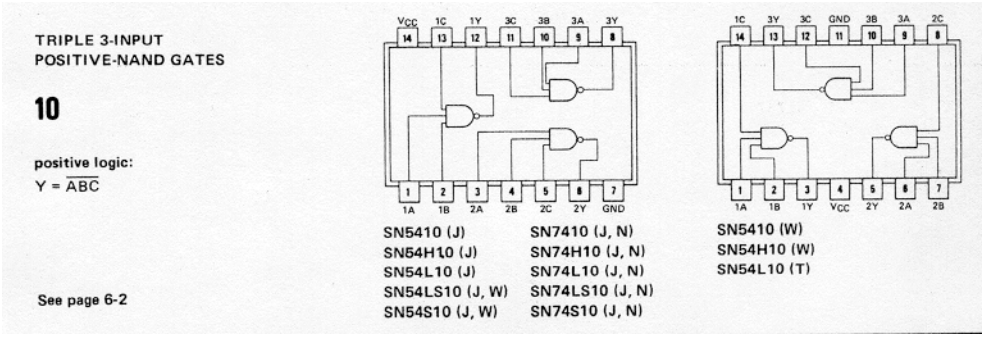
3. A logic diagram indicating which gates are used for the parity generate circuitry and how they are interconnected.



4. A logic diagram indicating how the 7474's are used to implement the input and output registers. Indicate what is done with all inputs and outputs (clock, preset, clear, Q, Q' and D).



5. A calculation of the minimum clock period necessary for your implementation to function properly. Include the equation you used to determine the minimum clock period and which numbers from the data sheet were selected and incorporated into this calculation.


TRIPLE 3-INPUT
POSITIVE-NAND GATES

10

positive logic:
$Y = \overline{ABC}$

See page 6-2

SN5410 (J)
SN54H10 (J)
SN54L10 (J)
SN54LS10 (J, W)
SN54S10 (J, W)

SN7410 (J, N)
SN74H10 (J, N)
SN74L10 (J, N)
SN74LS10 (J, N)
SN74S10 (J, N)

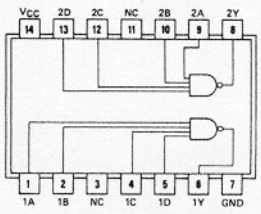
SN5410 (W)
SN54H10 (W)
SN54L10 (T)

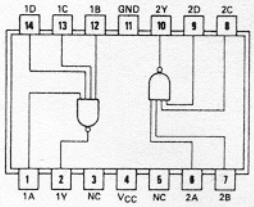
DUAL 4-INPUT POSITIVE-NAND GATES

20

positive logic:
$Y = \overline{ABCD}$

See page 6-2





SN5420 (J) SN7420 (J, N)
SN54H20 (J) SN74H20 (J, N)
SN54L20 (J) SN74L20 (J, N)
SN54LS20 (J, W) SN74LS20 (J, N)
SN54S20 (J, W) SN74S20 (J, N)

SN5420 (W)
SN54H20 (W)
SN54L20 (T)

NC—No internal connection

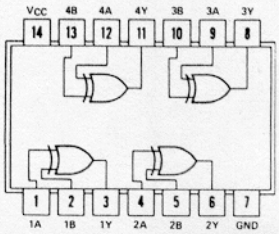
7410, 7420 Switching Characteristics

switching characteristics at V_{CC} = 5 V, T_A = 25°C

TYPE	TEST CONDITIONS#	t_{PLH} (ns) Propagation delay time, low-to-high-level output			t_{PHL} (ns) Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
'00, '10	C_L = 15 pF, R_L = 400 Ω		11	22		7	15
'04, '20			12	22		8	15
'30			13	22		8	15

QUADRUPLE 2-INPUT EXCLUSIVE-OR GATES

86 $Y = A \oplus B = \overline{A}B + A\overline{B}$

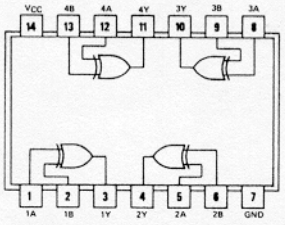


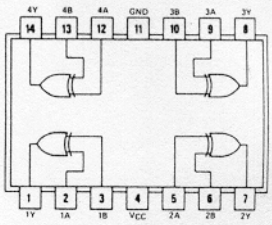
SN5486 (J, W) SN7486 (J, N)
SN54LS86 (J, W) SN74LS86 (J, N)
SN54S86 (J, W) SN74S86 (J, N)

FUNCTION TABLE

INPUTS A	INPUTS B	OUTPUT Y
L	L	L
L	H	H
H	L	H
H	H	L

H = high level, L = low level





See page 7-65

SN54L86 (J) SN74L86 (J, N)

SN54L86 (T)

7486 Switching Characteristics

switching characteristics, V_{CC} = 5 V, T_A = 25°C

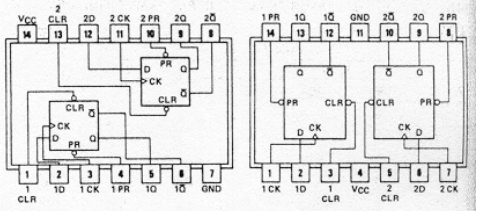
PARAMETER¶	FROM (INPUT)	TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{PLH}	A or B	Other input low	C_L = 15 pF, R_L = 400 Ω, See Note 3		15	23	ns
t_{PHL}					11	17	
t_{PLH}	A or B	Other input high			18	30	ns
t_{PHL}					13	22	

DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

74

FUNCTION TABLE

INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\bar{Q}_0$



SN5474 (J) SN7474 (J, N) SN5474 (W)
SN54H74 (J) SN74H74 (J, N) SN54H74 (W)
SN54L74 (J) SN74L74 (J, N) SN54L74 (T)
SN54LS74A (J, W) SN74LS74A (J, N)
SN54S74 (J, W) SN74S74 (J, N)

See pages 6-46, 6-50, 6-54, and 6-56

7474 Switching Characteristics

recommended operating conditions

	SERIES 54/74	'70			'72, '73, '76, '107			'74			'109			'110			'111			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	Series 54	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
	Series 74	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}				−400			−400			−400			−800			−800			−800	μA
Low-level output current, I_{OL}				16			16			16			16			16			16	mA
Pulse width, t_w	Clock high	20			20			30			20			25			25			ns
	Clock low	30			47			37			20			25			25			
	Preset or clear low	25			25			30			20			25			25			
Input setup time, t_{su}		20↑			0↑			20↑			10↑			20↑			0↑			ns
Input hold time, t_h		5↑			0↓			5↑			5↑			5↑			30↑			ns
Operating free-air temperature, T_A	Series 54	−55		125	−55		125	−55		125	−55		125	−55		125	−55		125	°C
	Series 74	0		70	0		70	0		70	0		70	0		70	0		70	

↑↓The arrow indicates the edge of the clock pulse used for reference: ↑ for the rising edge, ↓ for the falling pulse.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'70			'72, '73 '76, '107			'74			'109			'110			'111			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}			C_L = 15 pF, R_L = 400 Ω, See Note 2	20	35		15	20		15	25		25	33		20	25		20	25		MHz
t_{PLH}	Preset (as applicable)	Q				50		16	25			25		10	15		12	20		12	18	ns
t_{PHL}		$\bar{Q}$				50		25	40			40		23	35		18	25		21	30	
t_{PLH}	Clear (as applicable)	$\bar{Q}$				50		16	25			25		10	15		12	20		12	18	ns
t_{PHL}		Q				50		25	40			40		17	25		18	25		21	30	
t_{PLH}	Clock	Q or $\bar{Q}$			27	50		16	25		14	25		10	16		20	30		12	17	ns
t_{PHL}					18	50		25	40		20	40		18	28		13	20		20	30	

¶f_{max} ≡ maximum clock frequency; t_{PLH} ≡ propagation delay time, low-to-high-level output; t_{PHL} ≡ propagation delay time, high-to-low-level output.
NOTE 2: Load circuit and voltage waveforms are shown on page 3-10.

= (CLK → Q) + (COMBO GATE DELAY) + (SETUP TIME)

FROM 7474 DATA SHEET

MAX CLOCK → Q = 40 nS (t_{PHL})

SETUP TIME = 20 nS

COMBINATIONAL DELAY

FOR 7410, 7420 IMPLEMENTATION

- BECAUSE TRUE AND COMPLEMENTED VERSIONS AVAILABLE FROM FF's, NO INVERTERS NECESSARY
- TWO CASES:

OUTPUT HIGH: t_{PHL} (7410) + t_{PLH} (7420)

15 + 22 = 37

OUTPUT LOW: t_{PLH} (7410) + t_{PHL} (7420)

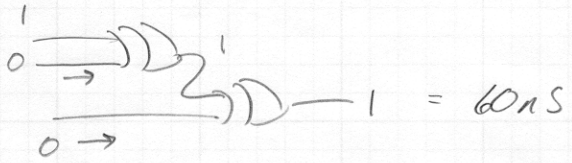
22 + 15 = 37

FOR 7486 (XOR) IMPLEMENTATION

- BECAUSE WORST CASE CONDITION

t_{PLH} (OTHER INPUT HIGH) = 30nS

CAN OCCUR FOR BOTH GATES, THIS COMBINATIONAL DELAY MUST BE USED



= 60nS

MINIMUM CLOCK PERIOD

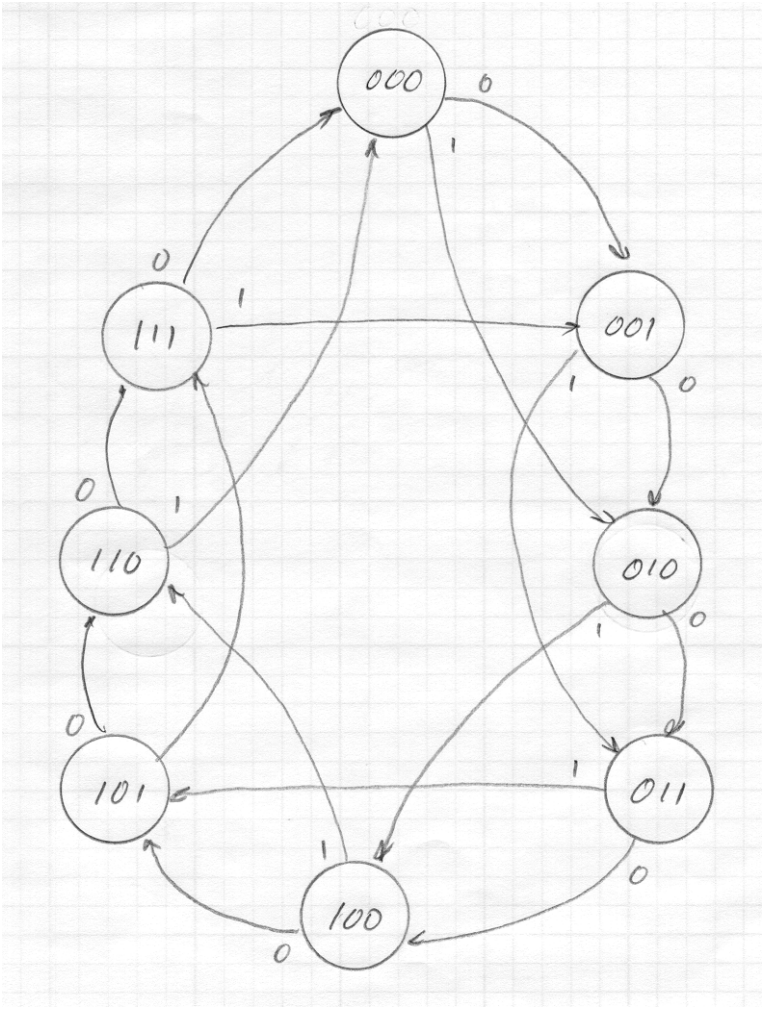
20 + 37 + 40 = 97nS FOR NANDS

20 + 60 + 40 = 120nS FOR XOR

Problem #7.

In this problem you are to design a 3-bit counter with a single input (x). When x is 0, the count should be incremented by 1 and when x is 1, the count should be incremented by 2. Label the state variables A, B and C where A is the most significant bit.

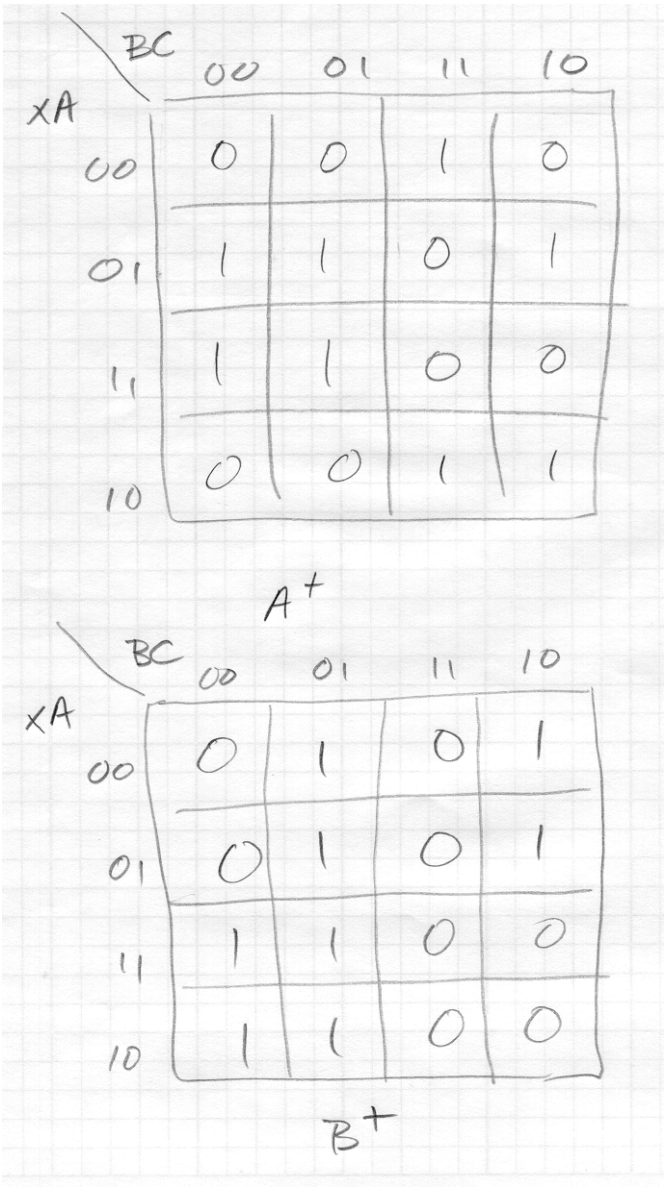
1. Generate the state diagram for this counter.



2. Generate the state table for this counter.

PS	NS x=0	NS x=1
000	001	010
001	010	011
010	011	100
011	100	101
100	101	110
101	110	111
110	111	000
111	000	001

3. Generate the next state K-maps for this counter.

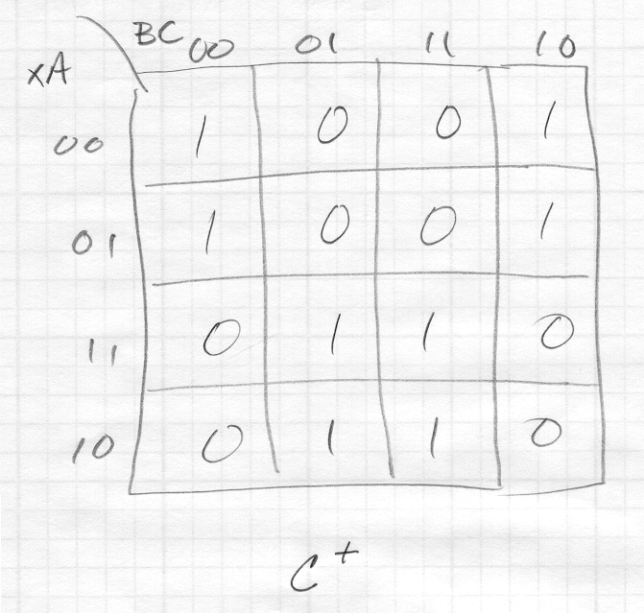


xA \ BC	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	0	0
10	0	0	1	1

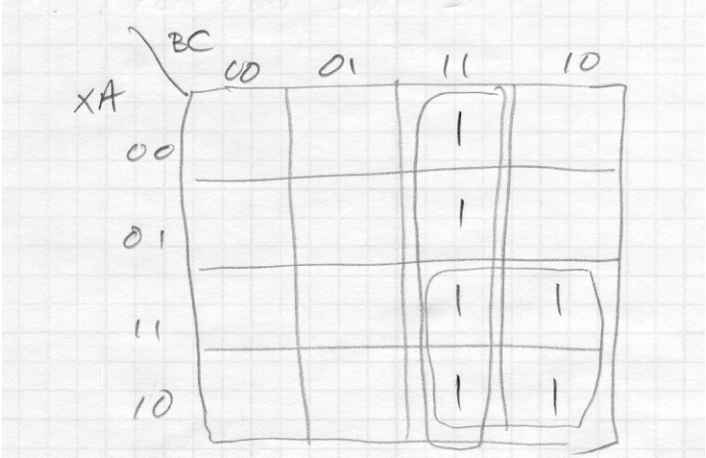
A^+

xA \ BC	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	0

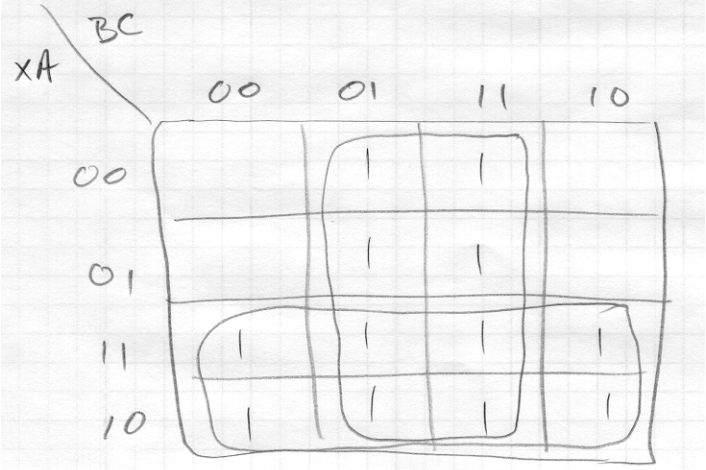
B^+



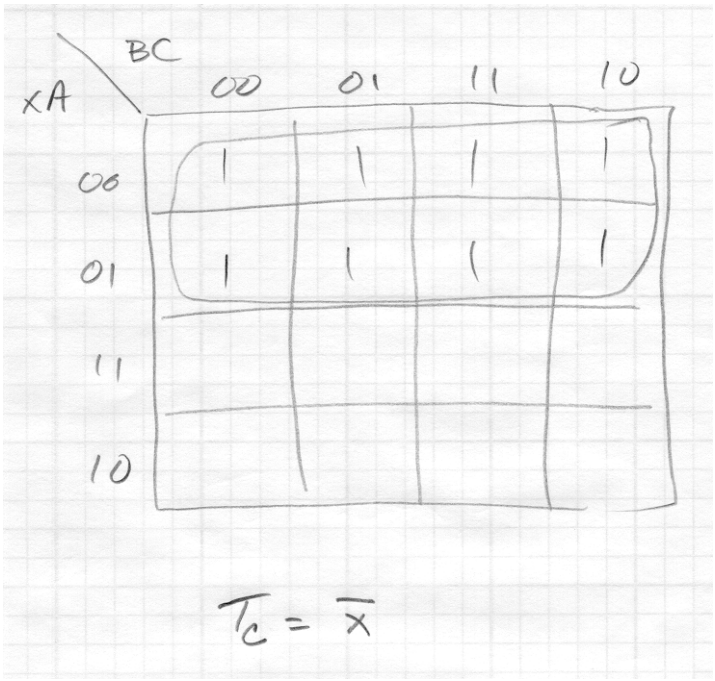
4. Implement the design using T flip-flops. Indicate the inputs to the three T flip-flops (the Boolean expression). You don't have to draw the circuit.



$$T_A = BC + xB$$



$$T_B = x + C$$



Problem #8.

This problem addresses Verilog.

1. There are two errors in the code below; identify them and indicate a fix that will allow the code to compile (make any changes necessary but be clear on the changes).

```
module problem4_1 (x1, x2, x3, f, g);
      input x1, x2, x3;
      output f,g;
      reg f,g;

      assign f = (~x1 & ~x2 & x3) | (x1 & ~x2 & ~x3) |
                 (x1 & ~x2 & x3) | (x1 & x2 & ~x3) ;

      g = (~x1 & ~x2 & x3) | (x1 & ~x2 & ~x3) |
                 (x1 & ~x2 & x3) | (x1 & x2 & ~x3) ;


endmodule
```

1. ASSIGNMENT SYNTAX ERRORS

(1) ASSIGN f = ...

f DECLARED REG, BUT ASSIGNED TO CONTINUOUSLY (WIRE)

FIX: ASSIGN TO PROCEDURALLY (ALWAYS) OR MAKE f A WIRE

(2) g = ...

g IS REG VARIABLE, SO MUST BE ASSIGNED TO PROCEDURALLY

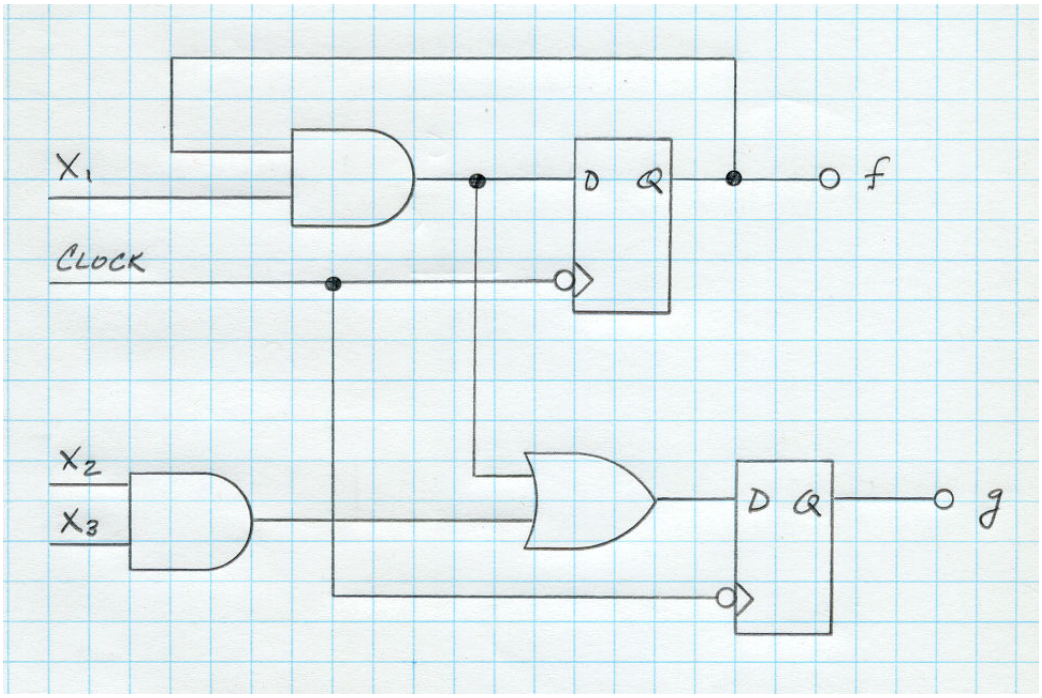
FIX: ADD ALWAYS BLOCK OR MAKE WIRE AND ASSIGN TO CONTINUOUSLY

2. For the module below, indicate the circuit that will be synthesized.

```
module problem4_2_A(x1, x2, x3, Clock, f, g);
      input x1, x2, x3, Clock;
      output f, g;
      reg f, g;

      always @(negedge Clock)
      begin
            f = x1 && f;
            g = f || x3 && x2;
      end

endmodule
```



3. Construct a state diagram for the Verilog code given below

```
module problem4_3 (Clock, A, B, Q);
      parameter n=3;
      input Clock, A, B;
      output [n-1:0] Q;
      reg [n-1:0] Q;

      always @(posedge Clock)
            begin

            if (A)
              if (B)
                Q <= Q + 1;
```

```
            else begin
              Q[1] <= ~Q[1]^Q[0];
              Q[0] <= ~Q[0];
            end
          end

endmodule
```