

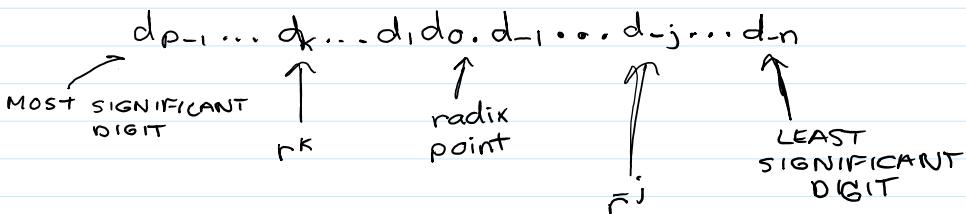
INTEGER REPRESENTATIONS

$$\text{INTEGER } N = a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b^1 + a_0$$

 $b = \text{BASE}$ $a = \text{INTEGERS} < b$

NUMBER SYSTEMS

A POSITIONAL NUMBER SYSTEM HAS A RADIX (OR BASE OF THE NUMBER) ANY INTEGER $r \geq 2$

 $i = \text{INTEGER OR COUNTER}$ $d = \text{DIGIT OR CHARACTER}$ $p = \text{NUMBER OF DIGITS TO LEFT OR DECIMAL POINT}$ $n = \text{NUMBER OF DIGITS TO RIGHT}$

$$D = \sum_{i=-n}^{p-1} d_i \cdot r^i \quad 0 \leq d_i \leq (r-1)$$

Ex) 5183.68_{10}

$p = 4 \quad n = 2$

 $\leftarrow \text{WEIGHT}$

$d_0 r^0 = 3 \times 10^0 = 3$

$d_{-1} r^{-1} = 6 \times 10^{-1} = 0.6$

$+ d_1 r^1 = 8 \times 10^1 = 80$

$+ d_2 r^2 = 1 \times 10^2 = 100$

$+ d_3 r^3 = 5 \times 10^3 = 5000$

$+ d_4 r^4 = 8 \times 10^4 = 80000$

$5000 + 100 + 80 + 3 + 0.6 + 0.08 = 5183.68$

$\nearrow \text{MSD}$ $\nwarrow \text{LSD}$

EX) 25.375 radix 10

$$2 \cdot 10^1 + 5 \cdot 10^0 + 3 \cdot 10^{-1} + 7 \cdot 10^{-2} + 5 \cdot 10^{-3}$$

EX) DECIMAL NUMBERS

$$5374_{10} = 5 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

BINARY NUMBERS

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

POWERS OF 2

$$2^0 = 1$$

$$2^7 = 128$$

$$2^1 = 2$$

$$2^8 = 256$$

NOTE: HANDY TO MEMORIZE
UP TO 2^8

$$2^2 = 4$$

$$2^9 = 512$$

$$2^3 = 8$$

$$2^{10} = 1024$$

$$2^4 = 16$$

$$2^{11} = 2048$$

$$2^5 = 32$$

$$2^{12} = 4096$$

$$2^6 = 64$$

$$2^{13} = 8192$$

EX) CONVERT 10011_2 TO DECIMAL

$$16 \times 1 + 8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 19_{10}$$

EX) CONVERT 47_{10} TO BINARY

$$32 \times 1 + 16 \times 0 + 8 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 101111_2$$

DIGITAL LOGIC MAPS THE INFINITE SET OF REAL VALUES FOR A PHYSICAL QUANTITY INTO TWO SUBSETS CORRESPONDING TO JUST TWO POSSIBLE NUMBERS "LOGIC VALUES" OF 0 AND 1

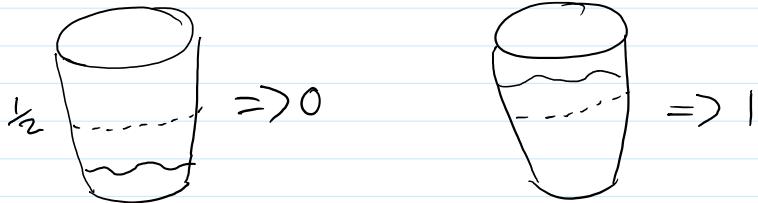
1-BIT = LOGIC VALUES 0 OR 1

DIGITAL DESIGNERS TEND TO REPLACE 0 AND 1 w/ LOW AND HIGH

LOW - SIGNAL IN RANGE OF ALGEBRAICALLY LOWER
VOLTAGES (LOGIC 0)

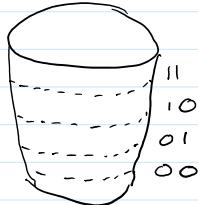
HIGH - SIGNAL IN RANGES OF ALGEBRAICALLY HIGHER
VOLTAGES (LOGIC 1)

EX) SUPPOSE WE HAVE A CKT TO DETERMINE IF WE HAVE
ENOUGH WATER



WIDE RANGE OF PHYSICAL VALUES MAPPED TO THE SAME
BINARY VALUE, DIGITAL LOGIC IS HIGHLY IMMUNE TO COMPONENT
AND POWER SUPPLY VARIATIONS AND NOISE

WE CAN BREAK INTO FINER RESOLUTION (2 OR MORE BITS)



BINARY VALUES AND RANGE

FOR AN N-DIGIT DECIMAL NUMBER THERE ARE 10^N VALUES
WITH A RANGE OF $[0, 10^N - 1]$

EX) $10^3 = 1000$ POSSIBLE VALUES w/ RANGE $[0, 999]$

FOR AN N-DIGIT BINARY NUMBER THERE ARE 2^N VALUES w/
RANGE $[0, 2^N - 1]$

EX) 3-DIGIT BINARY NUMBER

$2^3 = 8$ POSSIBLE VALUES w/ RANGE $[0, 7] = [000_2 \text{ TO } 111_2]$

$r = 2 \Rightarrow \text{BINARY NUMBER} = B$

WHERE $B = \sum_{i=-m}^{p-1} b_i \underbrace{2^i}_r$ $b = \text{BIT}$
DIGITS ARE CALLED BITS

$b_{p-1}, b_{p-2} \dots b_1, b_0, b_{-1}, b_{-2} \dots b_{-N}$

EX)

$$\begin{array}{rcl} \text{BINARY TO DECIMAL: } 10011_2 & = & 1 \cdot 2^0 = 1 \quad (\text{LSB}) \\ & + & 1 \cdot 2^1 = 2 \\ & + & 0 \cdot 2^2 = 0 \\ & + & 0 \cdot 2^3 = 0 \\ & + & \underline{1 \cdot 2^4 = 16} \quad (\text{MSB}) \\ & & 19 \quad (\text{DECIMAL}) \end{array}$$

EX) $\leftarrow \cdot \rightarrow$ START AT POINT GO LEFT THEN RIGHT
 101.011_2

$$\begin{array}{rcl} 1 \cdot 2^0 & = & 1 \\ + 0 \cdot 2^1 & = & 0 \\ + 1 \cdot 2^2 & = & 4 \\ \hline + 0 \cdot 2^{-1} & = & 0 \cdot 1_2 = 0 \\ + 1 \cdot 2^{-2} & = & 1 \cdot 1/4 = 0.25 \\ + 1 \cdot 2^{-3} & = & 1 \cdot 1/8 = 0.125 \end{array}$$

5.375 DECIMAL

HEXADECIMAL NUMBERS

- BASE 16
- SHORTHAND FOR BINARY

HEX DIGIT	DECIMAL	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

5	5	0101
6	6	0110
7	7	0111
8	8	1000
4	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

OCTAL + HEXADECIMAL
 $r = 8$ $r = 16$

BINARY TO OCTAL: BREAK BINARY BITS INTO GROUPS OF 3 THEN CONVERT

EX) $\begin{array}{cccc} 100 & , 011 & , 001 & , 110_2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 4 & 3 & 1 & 6 \end{array} = 4316_8$

BIN TO HEX: BREAK BINARY TO GROUPS OF FOUR THEN CONVERT

1,1101,1011,1010,1001

$\begin{array}{ccccc} 0001 & , 1101 & , 1011 & , 1010 & , 1001 \\ | & | & | & | & | \\ 1 & D & B & A & 9_{16} \end{array}$

BITS

 MOST SIGNIFICANT BIT (MSB)
 LEAST SIGNIFICANT BIT (LSB)

BYTES + NIBBLES

 BYTE
 NIBBLE

BYTES

 CEBF
 9AD7

BYTES

CEBF9AD7

↑
MOST SIGNIFICANT
BYTE

↑
LEAST SIGNIFICANT
BYTE

ADDITION

DECIMAL

$$\begin{array}{r}
 & 11 \leftarrow \text{CARRIES} \\
 3734 & \\
 + 5168 & \\
 \hline
 8902
 \end{array}$$

BINARY

$$\begin{array}{r}
 & 11 \leftarrow \text{CARRIES} \\
 1011 & \\
 + 0011 & \\
 \hline
 1110
 \end{array}$$

CONVERSION FROM BINARY TO OCTAL + HEXADECIMAL w/ BINARY POINT

EX) 10.1011001 (PAD w/ ZEROS)

← • →

BIN TO OCTAL: $\underbrace{010}_2, \underbrace{101}_5, \underbrace{100}_4, \underbrace{100}_4 = 2.544_8$

BIN TO HEX: $\underbrace{0010}_2, \underbrace{1011}_B, \underbrace{0010}_2 = 2.B2_{16}$

CONVERTING FROM OCTAL OR HEXADECIMAL TO DECIMAL

JUST REPLACE EACH DIGIT w/ BINARY EQUIVALENT

EX) OCTAL START AT POINT

$2046.17_8 = 010, 000, 100, 110.001, 111_2$

← . →

HEX: $9F.46C_{16} = 1001, 1111.0100, 0110, 1100_2$

(DECIMAL)
RADIX 10 CONVERSIONS MORE INVOLVED SINCE NOT POWERS OF 2

$$D = \sum_{i=0}^{p-1} d_i r^i$$

$$D = \sum_{i=-N}^n d_i r^i$$

TO GO FROM OCTAL TO HEXADECIMAL, MULTIPLY EACH DIGIT BY THE DECIMAL EQUIVALENT

EX) HEXADECIMAL TO DECIMAL

$$\begin{aligned} F1A3_{16} &= 15 \cdot 16^3 + 1 \cdot 16^2 + 10 \cdot 16^1 + 3 \cdot 16^0 \\ &= 15 \cdot 4096 + 1 \cdot 256 + 10 \cdot 16 + 3 \cdot 1 \\ &= 61,440 + 256 + 160 + 3 = 61,859_{10} \end{aligned}$$

EX) OCTAL TO DECIMAL

$$\begin{aligned} 436.5_8 &= 4 \cdot 8^2 + 3 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} \\ 256 + 24 + 6 + 5/8 &= 286.625_{10} \end{aligned}$$

NESTED METHOD FROM HEXADECIMAL TO DECIMAL

$$F1AC_{16} \Rightarrow 15 \cdot 16^3 + 1 \cdot 16^2 + 10 \cdot 16^1 + 12 \cdot 16^0$$

TO SOLVE FIRST DO FACTORS

FACTOR OUT A 16

$$(15 \cdot 16^2 + 1 \cdot 16 + 10) 16 + 12$$

FACTOR OUT A 16

$$((15 \cdot 16 + 1) 16 + 10) 16 + 12$$

GENERALIZED FORM FOR RADIX (2, 8, 16) TO DECIMAL CAN BE WRITTEN AS:

$$D = ((\dots ((d_{p-1}) \cdot r + d_{p-2}) \cdot r + \dots) \cdot r + d_1) \cdot r + d_0$$

$\underbrace{\quad}_{\text{DEC VALUE}}$

WHERE $d_i = \text{DECIMAL VALUE OF DIGIT } i$

FOR DECIMAL TO BINARY, OCTAL, HEXADECIMAL

WE DIVIDE D BY r WE GET

WE DIVIDE D BY r WE GET

↑
DECIMAL
IF

$$Q = \frac{D}{r} \quad \text{THE QUOTIENT}$$

FROM PREVIOUS GENERALIZED FORM

$$= (\dots ((d_{p-1}) \cdot r + d_{p-2}) \cdot r + \dots) \cdot r + d_1 + \frac{d_0}{r}$$

WHERE d_0/r WOULD BE THE REMAINDER OF Q AND LEAST SIGNIFICANT CHARACTER CONTINUE PROCEDURE TO GET NEXT CHARACTER

DECIMAL TO BINARY

EX) 179_{10} TO BINARY

$$179 \div 2 = 89 + 1 \quad (\text{REMAINDER}) \quad (\text{LSB})$$

$$\div 2 = 44 + 1$$

$$\div 2 = 22 + 0$$

$$\div 2 = 11 + 0$$

$$\div 2 = 5 + 1$$

$$\div 2 = 2 + 1$$

$$\div 2 = 1 + 0$$

$$\div 2 = 0 + 1 \quad (\text{MSB})$$

$$\therefore 179_{10} = 10110011_2$$

FRACTIONAL PART : DECIMAL TO BINARY

MULTIPLY w/ RADIX = 2 (SAME AS DIVIDE BY $1/2$)

EX) $0.375_{10} \Rightarrow \text{BINARY}$

$$0.375 \times 2 = 0.750 < 1 \quad d_{-1} = 0 \quad (\text{MSB})$$

$$0.750 \times 2 = 1.500 \geq 1 \quad d_{-2} = 1$$

$$0.500 \times 2 = 1.0 \geq 1 \quad d_{-3} = 1 \quad \begin{matrix} \nearrow \\ (\text{LSB}) \end{matrix}$$
$$0 \times 2 = 0 \quad d_{-4} = 0 \quad \begin{matrix} \nwarrow \\ \end{matrix}$$

$$0.375_{10} = 0.0110_2$$

$$0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4}$$

$$0/2 + 1/4 + 1/8 + 0/16$$

$$0.25 + 0.125 = 0.375$$

EX) DECIMAL TO OCTAL

467_{10} TO OCTAL

$$\begin{aligned} 467 \div 8 &= 58 + 3 \quad (\text{REMAINDER}) \quad (\text{LSC}) \\ \div 8 &= 7 + 2 \\ \div 8 &= 0 + 7 \quad (\text{MSC}) \\ &\quad \uparrow \\ &\quad \text{STOP PROCEDURE} \end{aligned}$$

$$= 723_8$$

EX) DECIMAL TO HEX

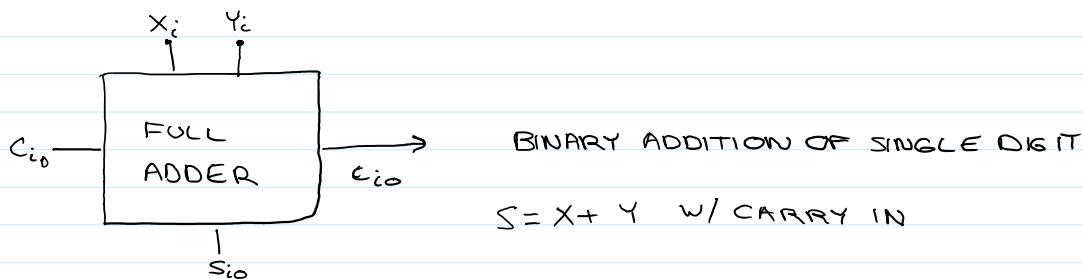
3417_{10} TO HEXADECIMAL

$$3417 \div 16 = 213 + 9 \quad (\text{REMAINDER}) \quad (\text{LSC})$$

$$\div 16 = 13 + 5$$

$$\begin{aligned} \div 16 &= 0 + 13 \quad "0" \quad (\text{MSC}) \\ &\quad \uparrow \\ &\quad \text{STOP} \end{aligned}$$

$$3417_{10} \leftrightarrow D59_{16}$$



3 INS
3 OUTS

8 POSSIBLE
COMBOS

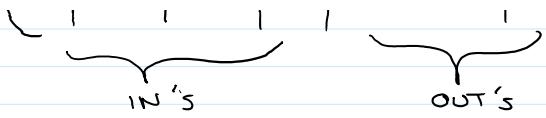
IN ANY
ORDER

BUT COUNT
000-111

IN'S

OUT'S

	X	Y	C _i	C ₀	S
0	0	0	0	0	0
0	0	0	1	0	1
0	1	0	0	0	1
0	1	0	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	1	0	0	1	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1



NOTE: $S = 1$ FOR ODD # OF 1'S
 $C_0 = 1$ FOR TWO OR MORE 1'S

EX

G_i	1 0 1 1 1 0 0 0	K START w/ $C_i(b_0) = 0$
X	1 9 0	1 0 1 1 1 1 0
Y	1 4 1	1 0 0 0 1 1 0 1
$X+Y = 331$	1 0 1 0 0 1 0 1 1	

↑
CARRY OUT OF BIT N

BINARY SUBTRACTION OF $X-Y$ w/ CARRY BORROW

$$D = (X - B_i) - Y \quad D = \text{DIFFERENCE}$$

X	B_i	Y	B_o	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

NOTE: D IS THE RESULT OF AN ODD # OF 1'S FROM X, Y, + B_i

$$B_o = 1 \text{ IF } \underbrace{B_i + Y}_{\text{OUT}} > X \quad \underbrace{\text{IN}}$$

EX

B_i	0 1 1 0 1 1 0 1 0	← START w/ '0' AS $B_i(b_0)$
X	2 1 0	1 1 0 1 0 0 1 0
Y	-1 0 9	0 1 1 0 1 1 0 1
$X-Y = 1 0 1$	0 1 1 0 0 1 0 1	

NOTE: THIS ONLY WORKS WHEN $X > Y$

YOU MUST DETERMINE WHICH NUMBER IS LARGER THEN
 SUBTRACT SMALLER FROM LARGER

EX) $X=45 \quad Y=101$ THEN DO $Y-X$

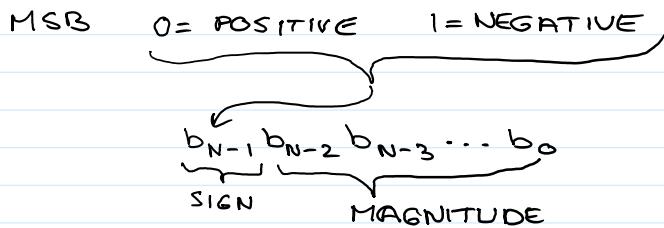
NOTE: THIS METHOD OF SUBTRACTION IS HARD TO DO IN
 HARDWARE

NOTE: $R_o = 1$ IF $B_i + Y > X$

ANOTHER WAY TO SUBTRACT IS TO DO ADDITION OF A NEGATIVE NUMBER

EXAMINATION OF NEGATIVE NUMBERS

SIGNED MAGNITUDE REPRESENTATION OF A NUMBER



(EX) N=8

The diagram shows two binary numbers. The first has a circled '0' sign bit above it, followed by 7 magnitude bits: 101 0101. To its right is the equation $= +85$. The second has a circled '1' sign bit above it, followed by 7 magnitude bits: 101 0101. To its right is the equation $= -85$. An arrow points from the 'SIGN BIT' label to the circled sign bit.

PROBLEM!

SUPPOSE WE HAVE A DIGITAL CKT THAT ADDS OR SUBS THE SIGN #'S OF X AND Y

CKT MUST:

- EXAMINE b_{N-1} OF X AND Y

ADDITION { - IF $b_{N-1}(X) = b_{N-1}(Y)$ THEN $S = X + Y$
THEN $b_{N-1}(S) = b_{N-1}(X) = b_{N-1}(Y)$

- NOW IF $b_{N-1}(X) \neq b_{N-1}(Y)$ AND IS $X > Y$
THEN $D = X - Y$ $b_{N-1}(D) = b_{N-1}(X)$ D=DIFFERENCE
IF $X < Y$ THEN $D = Y - X$ $b_{N-1}(D) = b_{N-1}(Y)$

LOTS OF HARDWARE IS REQUIRED TO DO THIS

BETTER METHOD: SUBTRACTION BY ADDING COMPLEMENT #'S

TO UNDERSTAND CONCEPT LOOK AT ONE'S COMPLIMENT OF A BINARY # N=8

MSB=SIGN
 $0000\ 1010_2 = +10_{10}$ JUST INVERT TO GET ONE'S COMP
 $1111\ 0101_2 = -10_{10}$ MSB IS SIGN BIT

EX) N=8 0111 0111, $= +19_{10}$

EX] $N=8$

$$\begin{array}{r} 0111\ 0111_2 = +119_{10} \\ 1000\ 1000_2 = -119_{10} \Rightarrow 18) \end{array}$$

NOTE: $| -119 | + | 8 | = 127 = 2^{N-1} - 1$

SO THAT $| [B] | = 2^{N-1} - 1 - | B |$

↗
BRACKETS MEAN
COMPLEMENT

$$8 = (128 - 1) - 119$$

NOTE: $0111\ 1111 = +127_{10}$

$$\begin{array}{r} \Downarrow \\ 1000\ 0000 = -127_{10} \end{array}$$

TWO'S COMPLEMENT WORKS MUCH BETTER FOR SUBTRACTION
FEWER RULES = LESS HARDWARE

$$\begin{aligned} [B]_2 &= 2^{N-1} - 1 - B + 1 = \text{TWO'S COMPLEMENT} \\ &= 2^{N-1} - B \quad \uparrow \\ &\quad \text{ADD ONE, SET } C_{i0} = 1 \end{aligned}$$

EX] $0111\ 0111 = 119_{10}$

$$\begin{array}{r} \Downarrow \\ 1000\ 1000 \quad (\text{FLIP BITS}) \\ + 1 \quad (\text{ADD ONE}) \\ \hline 1000\ 1001 = -119_{10} \text{ IN TWO'S COMP} \end{array}$$

EX] $0111\ 1111_2 = 127_{10}$

$$\begin{array}{r} \Downarrow \\ 1000\ 0000_2 \\ + 1 \\ \hline 1000\ 0001 = -127_{10} \end{array}$$

TWO'S COMP ADDITION + SUBTRACTION

MSB IS NEEDED FOR SIGN

\therefore RANGE OF ALL VALUES OF X, Y, S, D MUST BE $< 2^{N-1} - 1$

EX] $N=5$ THEN MAX VALUE FOR THE RESULTING SUM
OR DIFFERENCE = ± 15

..... -

GR DIFFERENCE = ± 15

$$\begin{array}{r} 01111_2 = +15_{10} \\ 11111_2 = -1 \quad (0001 \Rightarrow 1110+1) \end{array}$$

EX)

$$\begin{array}{r} 3_{10} \\ + 4_{10} \\ \hline 7_{10} \end{array} \quad \begin{array}{r} 0011_2 \\ 0100_2 \\ \hline 0111_2 \end{array}$$

$$\begin{array}{r} 6_{10} \\ - 3_{10} \\ \hline 3_{10} \end{array} \quad \begin{array}{r} 0110_2 \\ 1101_2 \\ \hline 0011_2 \end{array}$$

DUMPED
SIGN BIT

EX)

$$\begin{array}{r} 4_{10} \\ - 7_{10} \\ \hline -3_{10} \end{array} \quad \begin{array}{r} 0100_2 \\ 1001_2 \\ \hline 1101_2 \end{array}$$

$$\begin{array}{r} 0111_2 = 7 \\ 1000_2 \\ + 1 \\ \hline 1001_2 = 7 \end{array}$$

REVERSE 2'S COMP TO OBTAIN MAGNITUDE

$$\begin{array}{r} -7 \\ + 4 \\ \hline -3 \end{array} \quad \begin{array}{r} 1001_2 \\ 0100_2 \\ \hline 1101_2 \end{array}$$

ORDERING OF X & Y NOT IMPORTANT

MAIN PROBLEM IS OVERFLOW (EXCEED MAX RANGE FOR 2'S COMP)

EX)

$$\begin{array}{r} -3 \\ -6 \\ \hline -9 \end{array} \quad \begin{array}{r} 1101_2 \\ 1010_2 \\ \hline 0111_2 \end{array}$$

DUMPED

NEED TWO ADDITIONAL BITS
ONE FOR SIGN
ONE FOR ACCOMODATING RANGE

ERROR OCCURS WHEN: BOTH X & Y HAVE SAME SIGNS
AND RESULT HAS DIFFERENT SIGN

ADDITION OF TWO NUMBERS WITH DIFFERENT SIGNS HAS NO
OVERFLOW

EX)

SAME SIGNS

$$\begin{array}{r} +5 \\ +6 \\ \hline +11 \end{array} \quad \begin{array}{r} 0101_2 \\ 0110_2 \\ \hline 1011_2 \end{array}$$

$\begin{array}{r} 011 \\ -1 \\ \hline 010 \\ \downarrow \\ 101 = 5 \end{array}$

IF MSB
 $b_{N-1}(x) = b_{N-1}(y) \neq b_{N-1}(s)$

EX) N=5

SUBTRACTION

OVERFLOW : N = 5

MAX RANGE = 1111 = 15₁₀

$$-14 - (+7) = -21$$

$$\begin{array}{r} -14 \\ -+7 \\ \hline -21 \end{array}$$

(COMP 7)

$\begin{array}{r} 10010 \\ 11001 \\ \hline 1011 \end{array} = +11$

\checkmark OUT ERROR FLAG

OVERFLOW OCCURS WHEN : BOTH X AND Y HAVE SAME SIGN
AND RESULT HAS DIFFERENT SIGN

USING THE CARRY IN, C_{IN}, TO DO COMPLEMENT SUBTRACTION

EX) $\begin{array}{r} +4 & 0100 \\ -+3 & -0011 \end{array} \Rightarrow$

1's COMP 2's COMP

$\begin{array}{r} 0100 \\ +1100 \\ \hline 10001 \end{array} = +1$

↑ NOT USED FOR OVERFLOW

∴ WHEN WE WANT TO DO SUBTRACTION WE JUST
COMPLEMENT THE SUBTRAHEND THEN ADD 1 BY
USING THE C_{IN} (FIRST BIT CARRY-IN)

MULTIPLICATION

JUST SHIFT + ADD

$$3 \times 5$$

$$\begin{array}{r} 011 & 3 \\ \times 101 & \\ \hline 011 & \\ 000 - & \\ \hline 011 -- & \\ 1111 = 15_{10} & \end{array}$$

$$\begin{array}{r} 1011 & 11 \\ \times 1101 & \\ \hline 1011 & \\ 0000 - & \\ 1011 -- & \\ \hline 10001111 = 143_{10} & \end{array}$$

DIVISION

SHIFT + SUBTRACT

$$5 \overline{)26} \quad 1R$$

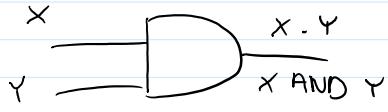
$$\begin{array}{r} 101 \overline{)11010} \\ -101 \\ \hline 001 \end{array}$$

3126

$$\begin{array}{r} 111010 \\ -101 \\ \hline 001 \\ 0011 \\ -000 \\ \hline 011 \\ 110 \\ -101 \\ \hline 001 \end{array}$$

$$\therefore \text{ANS} = 101 + 1 \text{ REMAINDER}$$

AND GATE



X	Y	$X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

AND GATE \Rightarrow 1 IFF ALL INPUTS ARE 1

OR / XOR / XNOR GATE



X	Y	$X + Y$	$X \oplus Y$	$X \ominus Y$
0	0	0	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	0	1



OR GATE \Rightarrow 1 IFF ONE OR MORE INPUTS ARE 1

XOR GATE \Rightarrow 1 IFF $X \neq Y$

XNOR GATE \Rightarrow 1 IFF $X = Y$

NOT GATE

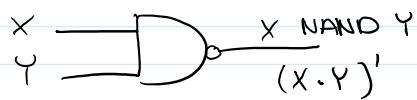


X	NOT X
0	1
1	0

NOT GATE (INVERTER) OUTPUT OPPOSITE FROM INPUT

NAND GATE

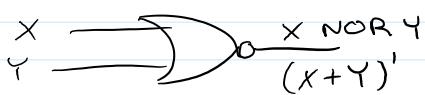
NAND GATE



X	Y	$(X \cdot Y)'$
0	0	1
0	1	1
1	0	1
1	1	0

NAND GATE \Rightarrow 1 IFF BOTH INPUTS AREN'T HIGH

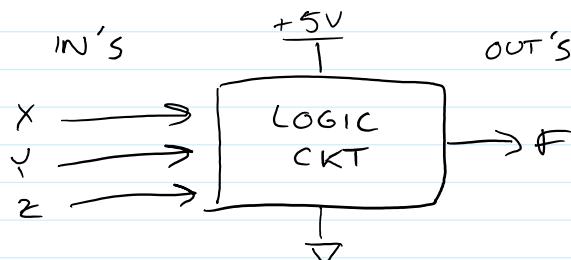
NOR GATE



X	Y	$(X + Y)'$
0	0	1
0	1	0
1	0	0
1	1	0

NOR GATE \Rightarrow 1 IFF BOTH INPUTS ARE LOW

LOGIC CKTS REPRESENTED AS BLACK BOX WITH IN'S AND OUT'S



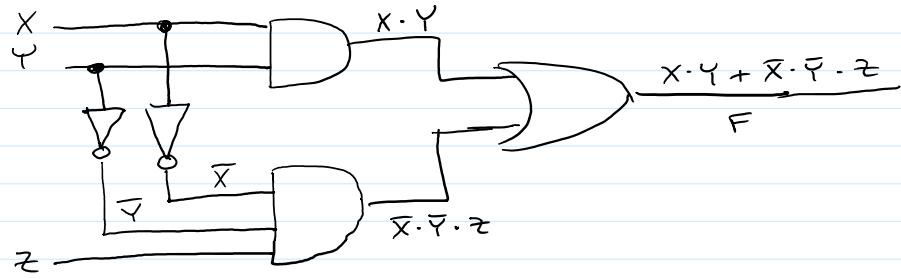
X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$F = \bar{X}\bar{Y}Z + X\bar{Y}\bar{Z} + XY\bar{Z}$$

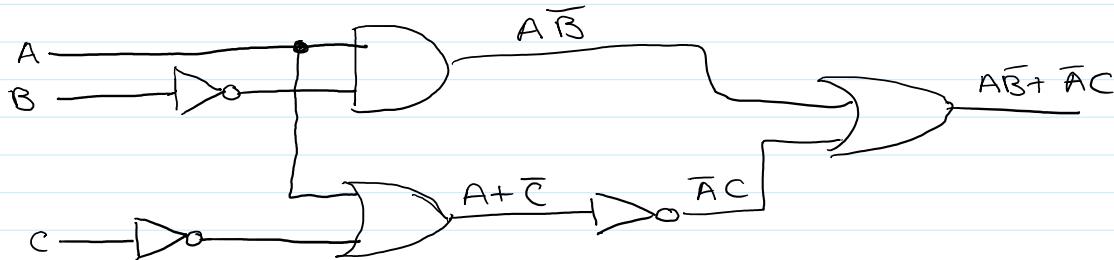
$$= \bar{X}\bar{Y}Z + XY(\bar{Z} + Z)$$

ALWAYS 1

$$\therefore F = \bar{X}\bar{Y}Z + XY = F$$



EX] FILL TRUTH TABLE FOR GIVEN CKT



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	1
1	1	0	0
1	1	1	0

LOGIC FAMILIES

LOGIC FAMILY \Rightarrow COLLECTION OF DIFFERENT IC - CHIPS THAT HAVE SIMILAR INPUT, OUTPUT, AND INTERNAL CHARACTERISTICS BUT PERFORM DIFFERENT LOGIC FUNCTIONS

TRANSISTOR - TRANSISTOR LOGIC (TTL)

- MOST SUCCESSFUL OF BIPOLEAR LOGIC FAMILY

- INTRODUCED IN 1960 S

COMPLIMENTARY METAL-OXIDE SEMICONDUCTOR LOGIC (CMOS)

- POPULARITY STARTED IN 1980 S

- PRIOR MOS FIELD EFFECT TRANSISTORS (MOSFETS) WERE
(1) DIFFICULT TO FABRICATE
(2) LAGGED CONSIDERABLY IN SPEED TO BIPOLEAR CKTS

- NEW ADVANCES IN 1980 S :

- (1) PERFORMED EQUIVALENT FUNCTIONALITY OR GREATER THAN TTL
- (2) HIGHER SPEEDS
- (3) LOWER POWER CONSUMPTION
- (4) EASIEST TO UNDERSTAND
- (5) MOST CAPABLE

LOGIC VOLTAGE LEVELS

INPUT

V_{IL} → HIGHEST INPUT VOLTAGE AND STILL
IV RECOGNIZED AS A "0"

V_{IH} → LOWEST INPUT VOLTAGE AND STILL "1"
3V

OUTPUT

V_{OL} → ACTUAL OUTPUT VOLTAGE WHEN GATE
0.5V WHEN GATE OUTPUT IS AT "0"

V_{OH} → ACTUAL OUTPUT VOLTAGE WHEN GATE OUTPUT
4V IS AT "1"

WE WANT

$$V_{OL} < V_{IL}$$

$$V_{OH} > V_{IH}$$

LOGICAL OPERATIONS OF DIGITAL NUMBERS ARE CARRIED OUT
ON BIT LEVEL

LET US HAVE 2 BINARY NUMBERS X , Y S.T.

$$X \Rightarrow \dots x_3, x_2, x_1, x_0$$

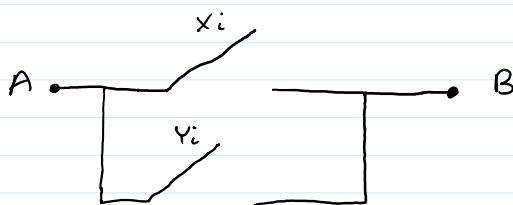
$$Y \Rightarrow \dots y_3, y_2, y_1, y_0$$

$1 \Rightarrow$ SWITCH CLOSED

$0 \Rightarrow$ SWITCH OPEN



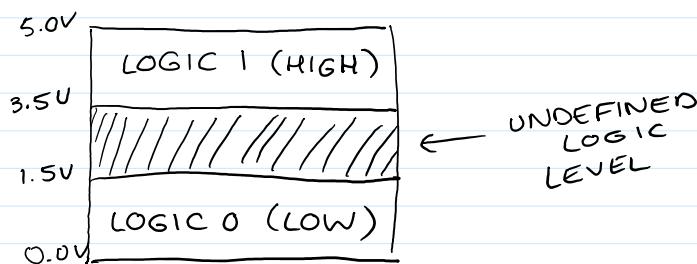
WHICH GATE DOES THIS REPRESENT? (AND GATE)



WHICH GATE DOES THIS REPRESENT? (OR GATE)

CMOS LOGIC LEVELS

CMOS TYPICALLY OPERATES FROM 5V POWER SUPPLY



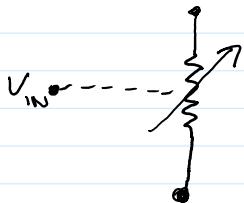
NOTE: CMOS NOT LIMITED TO 5V

CMOS w/ LOWER/HIGHER POWER SUPPLIES WILL ADJUST THE LEVELS ACCORDINGLY, THOUGH NOT NECESSARILY PROPORTION

MOS TRANSISTORS

CAN BE MODELED AS A 3-TERMINAL DEVICE

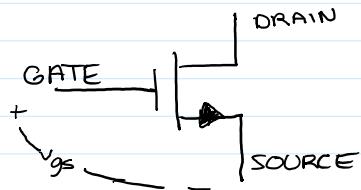
MOS TRANSISTOR AS A VOLTAGE CONTROLLED RESISTANCE



- RESISTANCE CONTROLLED BY INPUT VOLTAGE
- HIGH RESISTANCE (TRANSISTOR IS "OFF")
- LOW RESISTANCE (TRANSISTOR IS "ON")

TYPES OF MOS TRANSISTORS

n-CHANNEL MOS (NMOS)



VOLTAGE CONTROLLED RESISTANCE

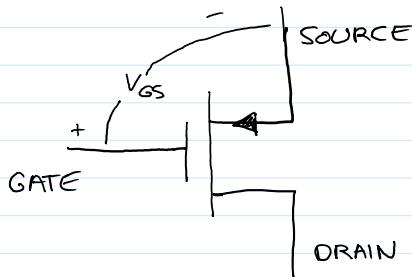
$$\uparrow V_{GS} \Rightarrow \downarrow R_{DS}$$

$$\text{IF } V_{GS} = 0 \text{ THEN } R_{DS} \geq 10^6 \Omega$$

$$V_{GS} \geq 0 \quad \text{AS } V_{GS} \uparrow \quad R_{DS} \downarrow \leq 10 \Omega \text{ or less}$$

DRAIN IS TYPICALLY AT A HIGHER VOLTAGE THAN THE SOURCE

p-CHANNEL MOS (μ MOS)



VOLTAGE CONTROLLED RESISTANCE

$$\downarrow V_{GS} \Rightarrow \downarrow R_{DS} \quad \text{IF } V_{GS} = 0 \quad R_{DS} \geq 10^6 \Omega$$

$$V_{GS} \leq 0 \quad \text{AS } V_{GS} \downarrow \quad R_{DS} \downarrow$$

SOURCE IS AT A HIGHER VOLTAGE THAN DRAIN

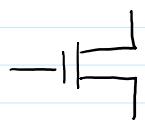
GATE IS SEPARATED FROM SOURCE AND DRAIN BY INSULATING MATERIAL WITH VERY HIGH RESISTANCE

GATE VOLTAGE CREATES AN ELECTRIC FIELD WHICH ENHANCES OR REDUCES FLOW OF CURRENT FROM GATE TO SOURCE

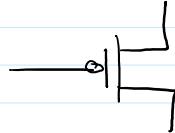
DUE TO HIGH RESISTANCE BETWEEN GATE AND OTHER TERMINALS, ONLY SMALL AMOUNTS OF CURRENT FLOWS FROM THE GATE ($< \mu\text{A}$)



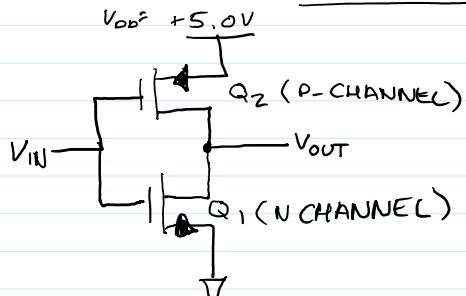
n - CHANNEL



p - CHANNEL



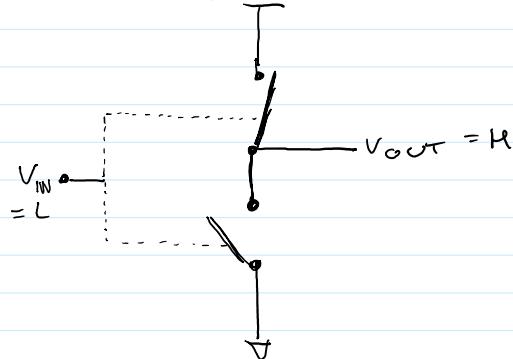
CMOS INVERTER



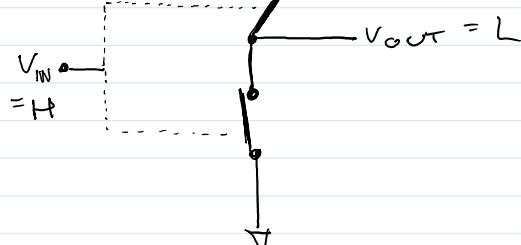
V_{IN}	Q1	Q2	V_{OUT}
0.0V(L)	OFF	ON	5.0V(H)
5.0V(H)	ON	OFF	0.0V(L)



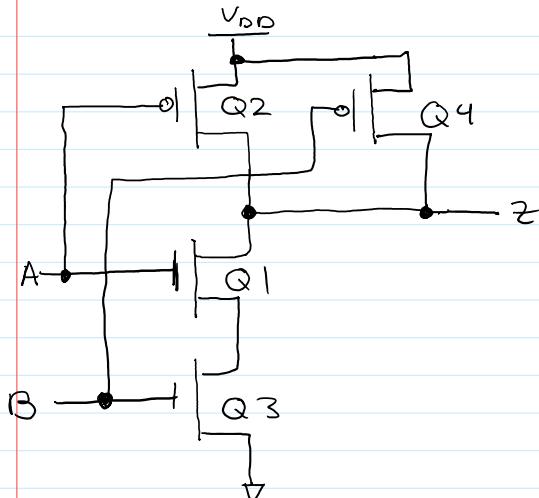
$V_{DD} = +5.0V$



$V_{DD} = +5.0V$



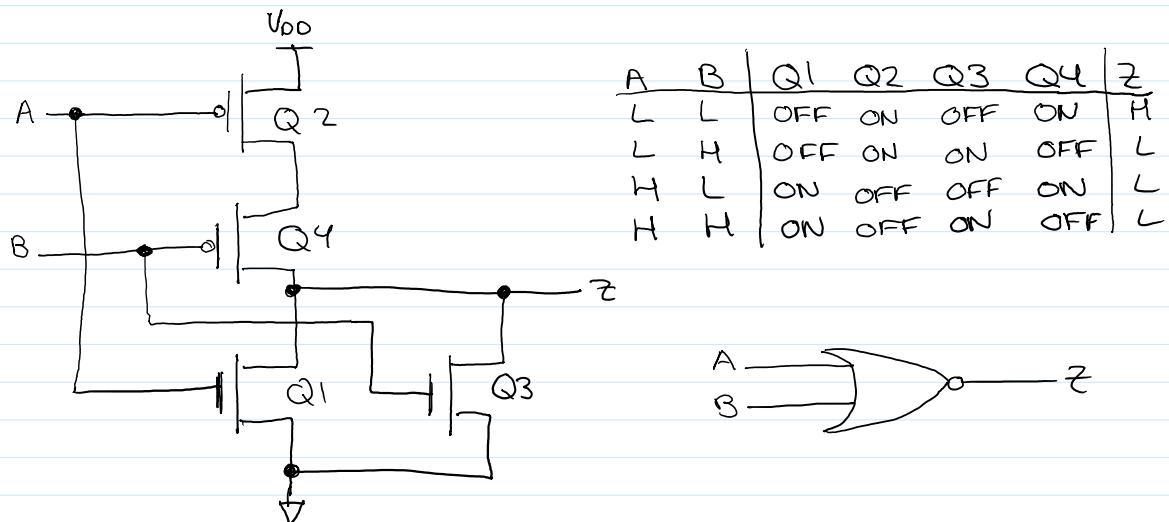
CMOS NAND



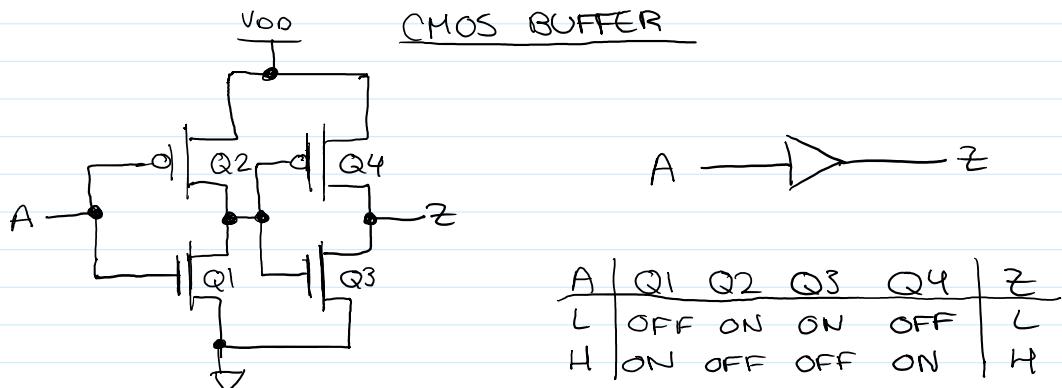
A	B	Q1	Q2	Q3	Q4	Z
L	L	OFF	ON	OFF	ON	H
L	H	OFF	ON	ON	OFF	H
H	L	ON	OFF	OFF	ON	H
H	H	ON	OFF	ON	OFF	L



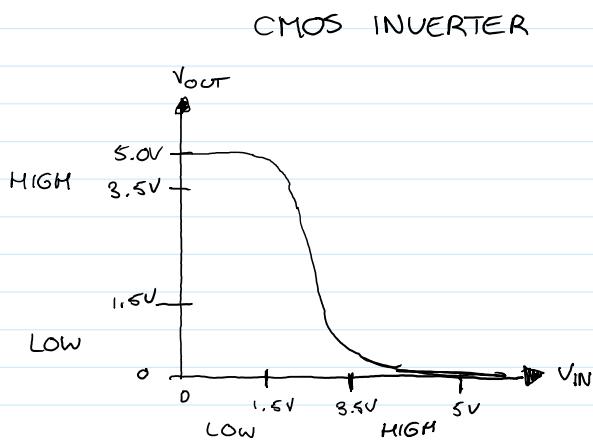
CMOS NOR



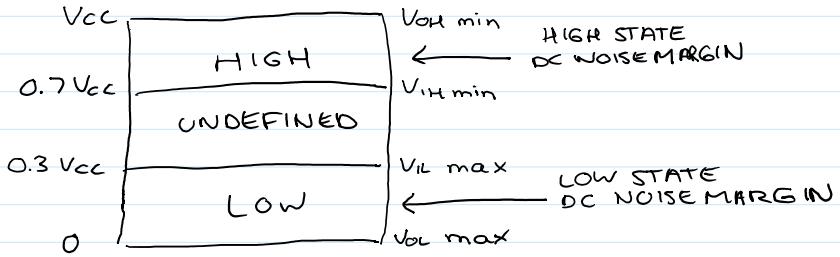
IN CMOS, AND OTHER LOGIC FAMILIES, INVERTING GATES ARE THE SIMPLEST
CMOS NON INVERTING GATES ARE CREATED BY ATTACHING AN INVERTER TO THE
OUTPUT TO THE CORRESPONDING INVERTING GATE



INPUT-OUTPUT TRANSFER CHARACTERISTIC



LOGIC LEVELS OF CMOS



CH 4

SWITCHING ALGEBRA

AXIOMS

$$(A1) \quad x = 0 \text{ IF } x \neq 1 \quad (A1') \quad x = 1 \text{ IF } x \neq 0$$

$$(A2) \quad \text{IF } x = 0, \text{ THEN } \bar{x} = 1 \quad (A2') \quad \text{IF } x = 1, \text{ THEN } \bar{x} = 0$$

$$x \rightarrowtail o \quad Y = \bar{x}$$

$$\begin{matrix} x \\ y \end{matrix} \rightarrowtail D \quad z = x \cdot y$$

$$\begin{matrix} x \\ y \end{matrix} \rightarrowtail D \quad z = x + y$$

$$(A3) \quad 0 \cdot 0 = 0$$

$$(A3') \quad 1 + 1 = 1$$

$$(A4) \quad 1 \cdot 1 = 1$$

$$(A4') \quad 0 + 0 = 0$$

$$(A5) \quad 0 \cdot 1 = 1 \cdot 0 = 0$$

$$(A5') \quad 1 + 0 = 0 + 1 = 1$$

SWITCHING ALGEBRA (ONE VARIABLE)

IDENTITY

$$(T1) \quad x + 0 = x \quad (T1') \quad x \cdot 1 = x$$

NULL ELEMENTS

$$(T2) \quad x + 1 = 1 \quad (T2') \quad x \cdot 0 = 0$$

IDEOMPOTENCY

$$(T3) \quad x + x = x \quad (T3') \quad x \cdot x = x$$

INVOLUTION

$$(T4) \quad (x')' = x$$

COMPLEMENT

$$(T4) \quad (X) = X$$

COMPLEMENT

$$(T5) \quad X + X' = 1 \quad (T5') \quad X \cdot X' = 0$$

COMMUTATIVITY

$$(T6) \quad X + Y = Y + X \quad (T6') \quad X \cdot Y = Y \cdot X$$

ASSOCIATIVITY

$$(T7) \quad (X + Y) + Z = X + (Y + Z) \quad (T7') \quad (X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$$

DISTRIBUTIVITY

$$(T8) \quad X \cdot Y + X \cdot Z = X \cdot (Y + Z) \quad (T8') \quad (X + Y) \cdot (X + Z) = X + Y \cdot Z$$

COVERING

$$(T9) \quad X + X \cdot Y = X \quad (T9') \quad X \cdot (X + Y) = X$$

\Downarrow

$$X \cdot (1 + Y) = X$$



COMBINING

$$(T10) \quad X \cdot Y + X \cdot Y' = X \quad (T10') \quad (X + Y) \cdot (X + Y') = X$$

CONSENSUS

$$(T11) \quad X \cdot Y + X' \cdot Z + Y \cdot Z = X \cdot Y + X' \cdot Z$$

$$(T11') \quad (X + Y) \cdot (X' + Z) \cdot (Y + Z) = (X + Y) \cdot (X' + Z)$$

X	Y	Z	X · Y	X' · Z	Y · Z	F
0	0	0				
0	0	1				
0	1	0				
0	1	1		1		1
1	0	0				
1	0	1				
1	1	0	1			1
1	1	1	1	1	1	1

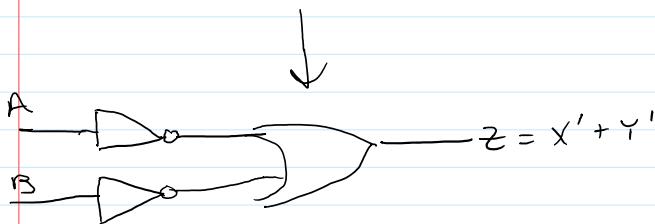
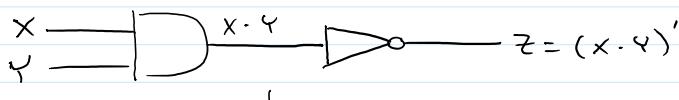
NOTE: THE Y · Z TERM WAS COVERED BY THE OTHER TERMS

AND HENCE REDUNDANT

DE MORGAN THEOREM

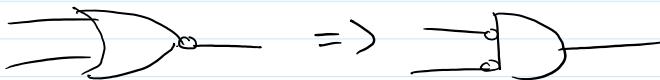
$$(x_1 \cdot x_2 \cdots x_n)' = x_1' + x_2' + \dots + x_n'$$

NAND = OR WITH INVERTED INPUTS



$$(x_1 + x_2 + \dots + x_n)' = (x_1' \cdot x_2' \cdots x_n')$$

NOR = AND W/ INVERTED INPUTS



$$(A+B)' = A' \cdot B'$$

THIS COUNTS AS NOR

DUALITY ANY THEOREM REMAINS TRUE IF 0 & 1 ARE SWAPPED
AND · & + ARE SWAPPED

EX $A + 0 = A \longrightarrow A \cdot 1 = A$

$$A + A' = 1 \longrightarrow A \cdot A' = 0$$

EX) REDUCE THE FUNCTION

$$F = \underbrace{A \cdot B + A \cdot B \cdot C' \cdot D}_{(A \cdot B) \cdot (1 + C'D)} + \underbrace{A \cdot B \cdot D \cdot E'}_{A' \cdot C' \cdot E} + \underbrace{A' \cdot B \cdot C' \cdot E}_{A' \cdot C' \cdot E} + \underbrace{A' \cdot B \cdot C \cdot E'}_{(B \cdot B')'}$$

$$F = A \cdot B + A' \cdot C' \cdot E$$

EX) BUILD $F = A \cdot C' + B \cdot C'$ w/ 4 NAND GATES

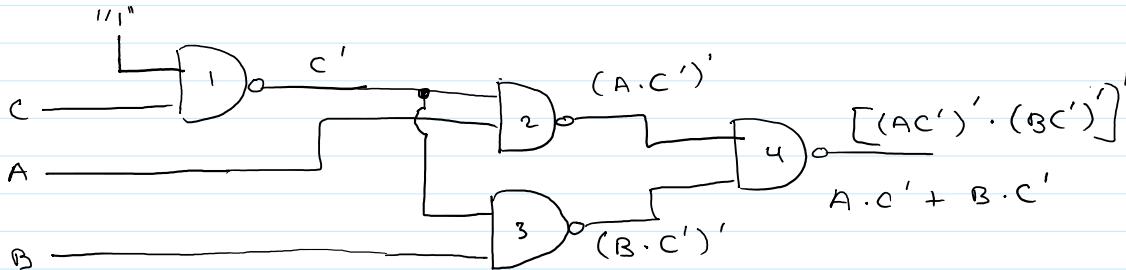
$$\text{LET } X = A \cdot C' \quad Y = B \cdot C'$$

$$\text{THEN } A \cdot C' + B \cdot C' = X + Y$$

$$\text{DE MORGAN: } X + Y = (X' \cdot Y')'$$

$$\text{PROOF: IF } X' + Y' = (X \cdot Y)' \text{ THEN } \underbrace{X'' + Y''}_{X \quad Y} = (X' \cdot Y')'$$

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



$$\begin{aligned} \text{EX}) \quad (\bar{A} + B)(B + A) &= \bar{A} \cdot A + \bar{A} \cdot B + A \cdot B + B \cdot B \\ &= 0 + \bar{A} \cdot B + A \cdot B + B \\ &= B(\bar{A} + A + 1) = B(1) = B \end{aligned}$$

$$\text{EX}) \quad A \cdot \bar{C} + A \cdot B \cdot \bar{C} = A \cdot \bar{C} \cdot (1 + B) = A \cdot \bar{C} \cdot (1) = A \cdot \bar{C}$$

OPERATOR PRECEDENCE

(1) NOT (HIGHEST)

(2) AND

(3) OR (LOWEST)

TERMINOLOGY

LITERALS - x, y, \bar{x}, \bar{y}

SUM TERM - $\bar{z}, w+x+y, \bar{x}+y+z$

PRODUCT TERM - $\bar{z}, w \cdot x \cdot y, \bar{x} \cdot y \cdot z$

SUM OF PRODUCTS (SOP) - $\bar{z} + w \cdot x \cdot y + \bar{x} \cdot y \cdot z + \bar{w} \cdot \bar{y} \cdot z$

PRODUCT OF SUMS (POS) - $\bar{z} \cdot (w \cdot x \cdot y) \cdot (\bar{x} + y + z)$

NORMAL TERM - NO VARIABLE APPEARS MORE THAN ONCE

EX) NON-NORMAL

$$w \cdot x \cdot x \cdot \bar{y}, w + w + \bar{x} + y, x \cdot \bar{x} \cdot y$$

NORMAL

$$w \cdot x \cdot \bar{y}, w + \bar{x} + y$$

MINTERM - NORMAL PRODUCT W/ N-LITERALS

$$\bar{w} \cdot \bar{x} \cdot \bar{y} \cdot \bar{z}, w \cdot x \cdot \bar{y} \cdot z$$

MAXTERM - NORMAL SUM W/ N-LITERALS

$$\bar{w} + \bar{x} + \bar{y} + \bar{z}, w + x + \bar{y} + z$$

ROW	X	Y	Z	F	MINTERM NAME	MINTERM	MAXTERM NAME	MAXTERM
0	0	0	0	1	m_0	$\bar{x} \cdot \bar{y} \cdot \bar{z}$	M_0	$x + y + z$
1	0	0	1	0	m_1	$\bar{x} \cdot \bar{y} \cdot z$	M_1	$x + y + \bar{z}$
2	0	1	0	0	m_2	$\bar{x} \cdot y \cdot \bar{z}$	M_2	$x + \bar{y} + \bar{z}$
3	0	1	1	1	m_3	$\bar{x} \cdot y \cdot z$	M_3	$x + \bar{y} + z$
4	1	0	0	1	m_4	$x \cdot \bar{y} \cdot \bar{z}$	M_4	$\bar{x} + y + \bar{z}$
5	1	0	1	0	m_5	$x \cdot \bar{y} \cdot z$	M_5	$\bar{x} + y + z$
6	1	1	0	1	m_6	$x \cdot y \cdot \bar{z}$	M_6	$\bar{x} + \bar{y} + \bar{z}$
7	1	1	1	1	m_7	$x \cdot y \cdot z$	M_7	$\bar{x} + \bar{y} + z$

CANONICAL SUM - SUM OF MINTERMS WHERE OUTPUT IS 1

$$F = \sum_{x,y,z} (0, 3, 4, 6, 7) = \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot \bar{z} + x \cdot y \cdot \bar{z} + x \cdot y \cdot z$$

CANONICAL PRODUCT - PRODUCT OF MAXTERMS WHERE OUT = 0

$$F = \prod_{x,y,z} (1, 2, 5) = (x + y + \bar{z}) \cdot (x + \bar{y} + z) \cdot (\bar{x} + y + \bar{z})$$

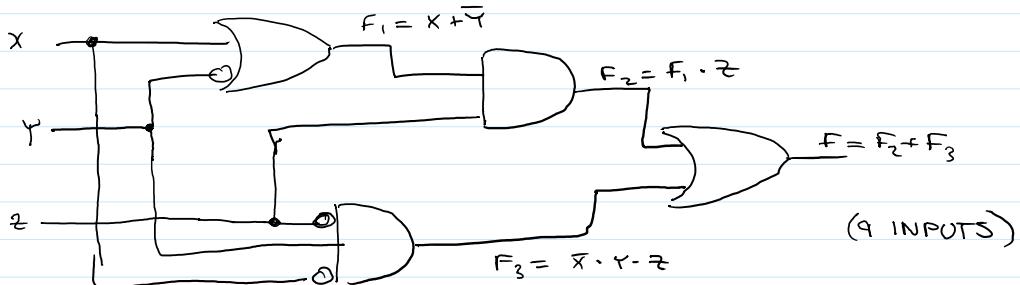
FOR A FUNCTION OF VARIABLES THERE ARE $2^n - 1$ ROWS W/ THE MINTERM AND MAXTERM CONTAINING A SUBSET OF ROWS

$$\sum_{ABC}(0, 1, 2, 3) = \prod_{ABC}(4, 5, 6, 7)$$

$$\sum_{XYZ}(1) = \prod_{XYZ}(0, 2, 3)$$

$$\sum_{WXYZ}(0, 1, 2, 3, 5, 7, 11, 13) = \prod_{WXYZ}(4, 6, 8, 9, 10, 12, 14, 15)$$

ANALYSIS EXAMPLE



COMBINATIONAL LOGIC CKT FIND TRUTH TABLE AND WRITE THE CANONICAL SUM AND PRODUCT

ROW	X	Y	Z	F_1	F_2	F_3	F
0	0	0	0	0	0	0	0
1	0	0	1	1	1	0	1
2	0	1	0	0	0	1	1
3	0	1	1	0	0	0	0
4	1	0	0	1	0	0	0
5	1	0	1	1	1	0	1
6	1	1	0	1	0	0	0
7	1	1	1	1	1	0	1

$F = \sum_{XYZ}(1, 2, 5, 7)$
 $= \prod_{XYZ}(0, 3, 4, 6)$

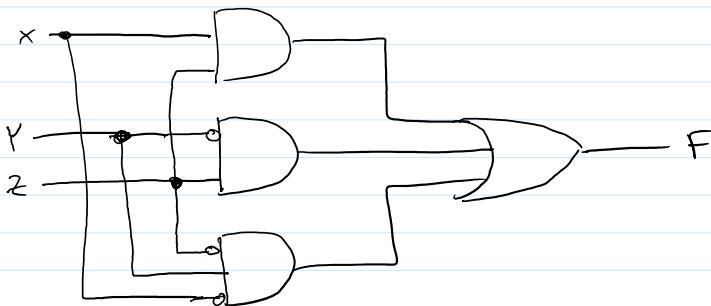
$$F = F_2 + F_3 = ((X + Y) \cdot Z) + (\bar{X} \cdot Y \cdot \bar{Z})$$

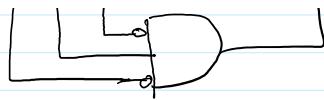
USE SWITCHING ALGEBRA TO MAKE NEW CKT

DISTRIBUTE Z

$$F_2 = X \cdot Z + \bar{Y} \cdot Z \quad \text{WE CAN NOW MAKE CKT}$$

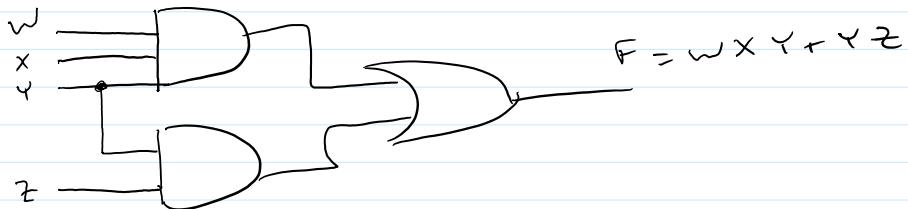
$$F = X \cdot Z + \bar{Y} \cdot Z + \bar{X} \cdot Y \cdot \bar{Z}$$



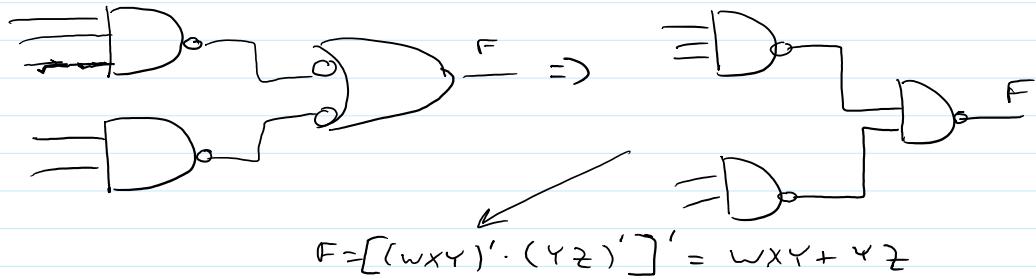


WHY IS BETTER? ONLY TWO LEVELS BUT REQUIRES 10 INPUTS

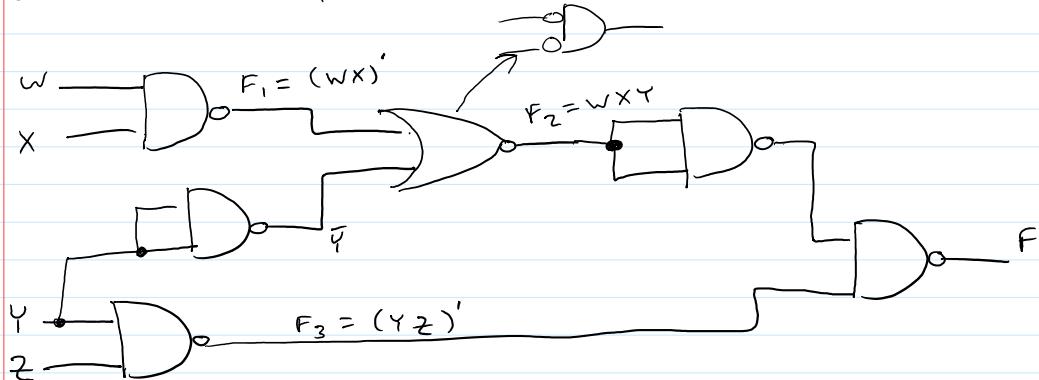
EX] DESIGN A CKT TO DO $F = W \cdot X \cdot Y + Y \cdot Z$
ONLY AND OR GATES



NAND-NAND



2-INPUT NAND/NOR

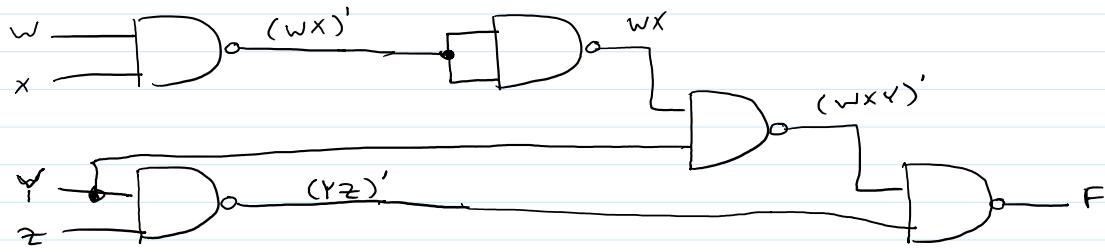


$$F_2 = [(W \cdot X)' + Y']' = (W \cdot X) \cdot Y = WXY$$

$$F_4 = (F_2)' = (WXY)' \quad F = (F_3 \cdot F_4)'$$

$$F = F_3' + F_4' = YZ + WXY$$

2 INPUT NANDS ONLY



$$F = [(wx'y) \cdot (yz)']' = wx'y + yz$$

COMBINATIONAL CIRCUIT SYNTHESIS

DESIGN A LOGIC CKT TO DO A TASK

LETS DESIGN AN ALARM CKT FOR A HOUSE

WE ARE GIVEN SENSORS ON THE WINDOW, GARAGE, AND DOOR
READING 0 FOR UNLOCKED AND 1 FOR LOCKED

THE HOUSE IS CONSIDERED SECURE IF THE WINDOW, DOOR, AND
GARAGE ARE 1

$$\text{SECURE} = \text{WINDOW} \cdot \text{GARAGE} \cdot \text{DOOR}$$

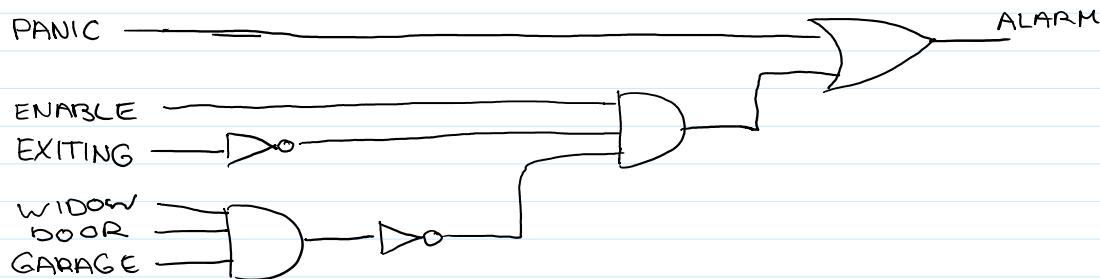
THE ALARM HAS A PANIC BUTTON THAT WHEN TRIGGERED
WILL SET OFF THE ALARM

$$\text{ALARM} = \text{PANIC}$$

IF ALARM IS ENABLED AND WE ARE NOT EXITING THE HOUSE
AND THE HOUSE IS NOT SECURE THE ALARM SHOULD GO
OFF

$$\text{ALARM} = \text{ENABLE} \cdot \overline{\text{EXITING}} \cdot \overline{\text{SECURE}}$$

$$\therefore \text{ALARM} = \text{PANIC} + \text{ENABLE} \cdot \overline{\text{EXITING}} \cdot (\text{WINDOW} \cdot \text{GARAGE} \cdot \text{DOOR})$$



EX DETECT THE FIRST FOUR BINARY PRIME NUMBERS

2, 3, 5, 7

3 VARIABLES REQUIRED TO MODEL
0-7

	X	Y	Z	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

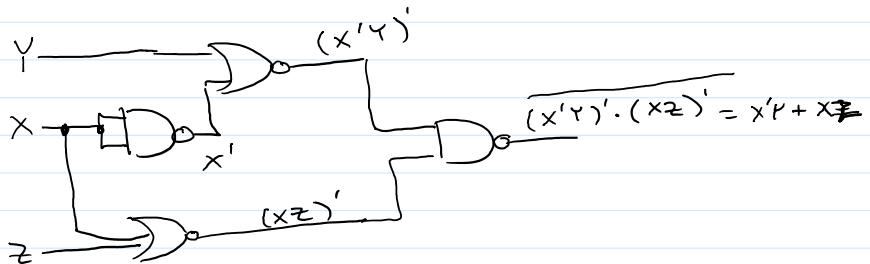
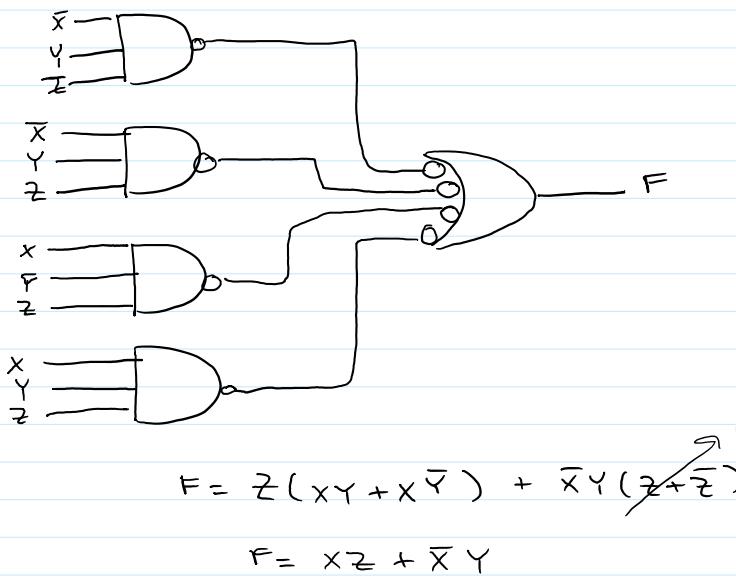
CANONICAL SUM

$$F = \sum_{xyz} (2, 3, 5, 7)$$

CANONICAL PRODUCT

$$F = \prod_{xyz} (0, 1, 4, 6)$$

$$F = \sum_{xyz} (2, 3, 5, 7) = \bar{x}yz + \bar{x}yz + x\bar{y}z + xy\bar{z}$$



IF ANY OF THE FOUR MINTERMS ARE PRESENT F=1

DESIGN A CKT USING THE CANONICAL PRODUCT FACTORS

WE WANT F=0 FOR 0, 1, 4, 6 AND F=1 FOR ALL OTHERS

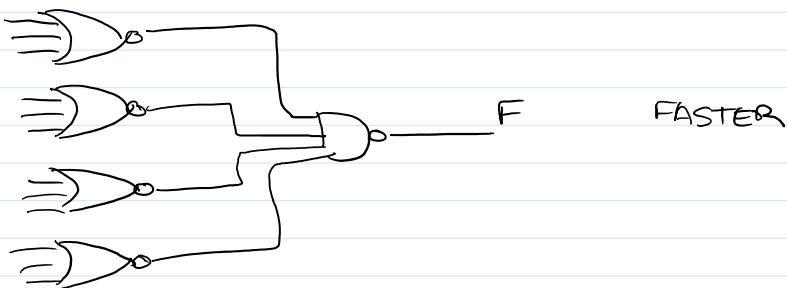
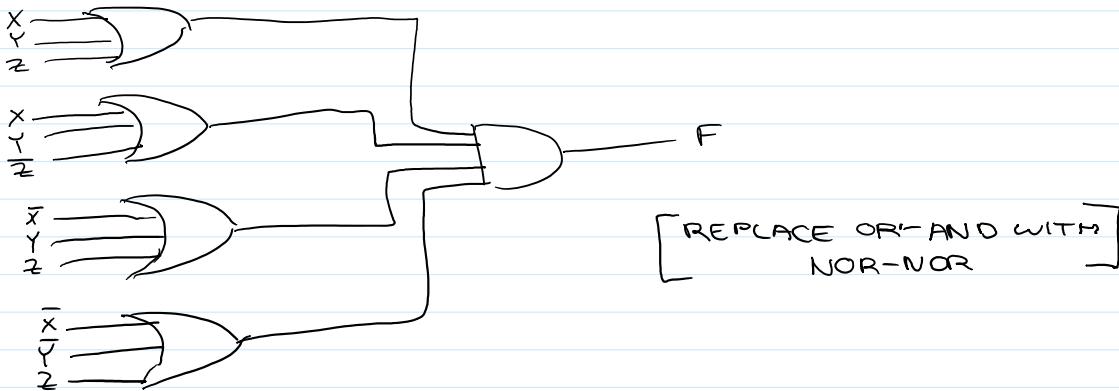
EACH PRODUCT TERM REQUIRES AN OR GATE W/ A ZERO OUTPUT

	X	Y	Z	F	$(x+y+z)=0$ ONLY WHEN $x=0 y=0 z=0$
0	0	0	0	0	$(x+y+\bar{z})=0$ WHEN $x=0 y=0 z=1$
1	0	0	1	0	$(\bar{x}+y+z)=0$ WHEN $x=1 y=0 z=0$
4	1	0	0	0	$(\bar{x}+\bar{y}+z)=0$ WHEN $x=1 y=1 z=0$
6	1	1	0	0	$(\bar{x}+\bar{y}+\bar{z})=0$ WHEN $x=1 y=1 z=1$

CANONICAL PRODUCT

$$F = \prod_{x,y,z} (0,1,4,6) = (x+y+z) \cdot (x+y+\bar{z}) \cdot (\bar{x}+y+z) \cdot (\bar{x}+\bar{y}+z)$$

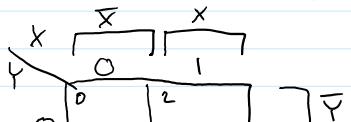
4 3 INPUT OR + 1 4 INPUT AND



IF ANY OF THE FOUR MAXTERMS 0, 1, 4, 6 WAS PRESENT THEN
THE OUTPUT OF THAT OR GATE = 0 $\Rightarrow F=0$

KARNAUGH MAPS

ALTERNATIVE FUNCTION TRUTH TABLE REPRESENTATION



ARRAY w/ 2 ROW OR 2 COLUMNS

X	0	1	
Y	0	2	Y
1	1	3	Y

ARRAY WI 2 ROW OR 2 COLUMNS
FOR EACH VARIABLE · ONE FOR THE
VARIABLE & ONE MORE FOR ITS COMPLIMENT

2 VARIABLE XY

XY		$X = 1$				
00 01 11 10						
0	0	2	4			
1	1	3	7	5		
						$Y = 1$

NOTE THE GRAY CODE SCHEME
FOR XY

ONLY ONE VARIABLE CHANGES
FOR EACH INCREMENT

3 VARIABLE XYZ

GRAY CODE FOR 4 BITS

BINARY WEIGHTED DECIMAL	W	X	Y	Z		
0	0	0	0	0	↑	
1	0	0	0	1	2 BITS YZ	
3	0	0	1	1	↓	
2	0	0	1	0		
6	0	1	1	0		3 BITS XYZ
7	0	1	1	1		
5	0	1	0	1		
4	0	1	0	0		
12	1	1	0	0		
13	1	1	0	1		
15	1	1	1	1		
14	1	1	1	0		
10	1	0	1	0		
11	1	0	1	1		
9	1	0	0	1		
8	1	0	0	0		

4 VARIABLE K-MAPS

WX		$w = 1$			
YZ					
00 01 11 10					
00	0	0	12	0	0
01	1	0	13	1	0
11	1	1	15	1	1
10	1	0	11	0	1

$z = 1$

7 MAXTERMS

7 MINTERMS

2 DONT CARES

01	1	5	13	9	0
11	8	7	15	11	1
10	2	6	14	10	d

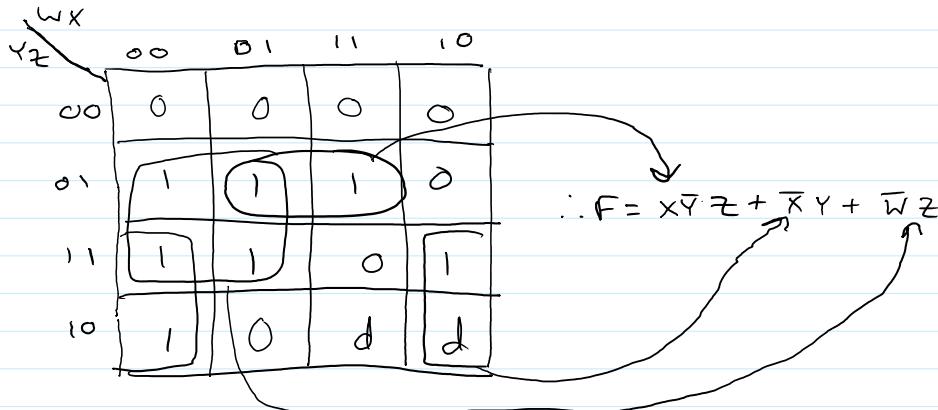
$x = 1$

2 DONT CARES

TO REPRESENT A FCN JUST
PUT IN ONES IN THE DECIMAL
NUMBERED CELLS

FILLED IN K-MAP

EX $F = \sum_{wxz} (1, 2, 3, 5, 7, 11, 13) + d(10, 14)$



MUST FASTER FOR REDUCING

K-MAP REVIEW

MAP FOR N-INPUT LOGIC FUNCTION IS AN ARRAY W/ 2^n CELLS

SMALL NUMBER IN CELL CORRESPONDS TO MINTERM # IN T.T.

EX MAP FOLLOWING LOGIC FUNCTIONS

(a) $F = \sum_{xy} (3)$

0	0
0	1

(b) $F = \sum_{xyz} (0, 3, 4, 6, 7)$

0	1	0	1	1
1	0	1	1	0

(c) $F = \sum_{wxz} (1, 2, 3, 5, 7, 11, 13)$

00	00	01	11	10
01	0	0	0	0
11	1	5	1	9
10	3	7	1	11

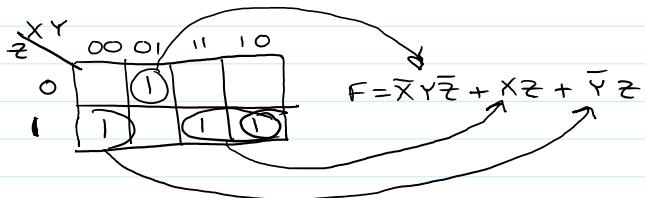
11	3	1	7	1	15	0	11	-
10	2	1	6	0	14	0	10	

EX)

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$F = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}Z + XY\bar{Z}$$

$$= \sum_{XYZ} (1, 2, 5, 7)$$



EACH CELL CORRESPONDS TO AN INPUT COMBINATION THAT DIFFERS FROM EACH OF ITS IMMEDIATELY ADJACENT NEIGHBORS IN ONLY ONE VARIABLE

∴ ADJACENT "1" CELLS CAN BE COMBINED USING GENERALIZATION OF THEOREM 10

$$\text{TERM} \cdot Y + \text{TERM} \cdot \bar{Y} = \text{TERM}$$

- (1) IF CIRCLE COVERS ONLY WHERE VARIABLE IS "0", THEN THE VARIABLE IS COMPLEMENTED IN PRODUCT TERM
- (2) IF CIRCLE COVERS ONLY WHERE VARIABLE IS "1", THEN THE VARIABLE IS UNCOMPLEMENTED IN PRODUCT TERM
- (3) IF CIRCLE COVERS WHERE VARIABLE IS "1" AND "0", THEN THE VARIABLE DOESN'T APPEAR IN PRODUCT TERM

DEFINITIONS

- MINIMAL SUM OF A LOGIC FUNCTION $F(x_1, \dots, x_n)$ IS SOP EXPRESSION FOR F S.T. NO SOP EXPRESSION FOR F HAS FEWER PRODUCT TERMS, AND ANY SOP EXPRESSION W/ THE SAME # OF PRODUCT TERMS HAS AT LEAST AS MANY LITERALS
- A LOGIC FUNCTION $P(x_1, \dots, x_n)$ IMPLIES A LOGIC FUNCTION $F(x_1, \dots, x_n)$ IF FOR EVERY INPUT COMBINATION SUCH THAT $P = 1$, THEN $F = 1$

$$\therefore P \rightarrow F \stackrel{\Delta}{=} F = 1 \wedge P = 1$$

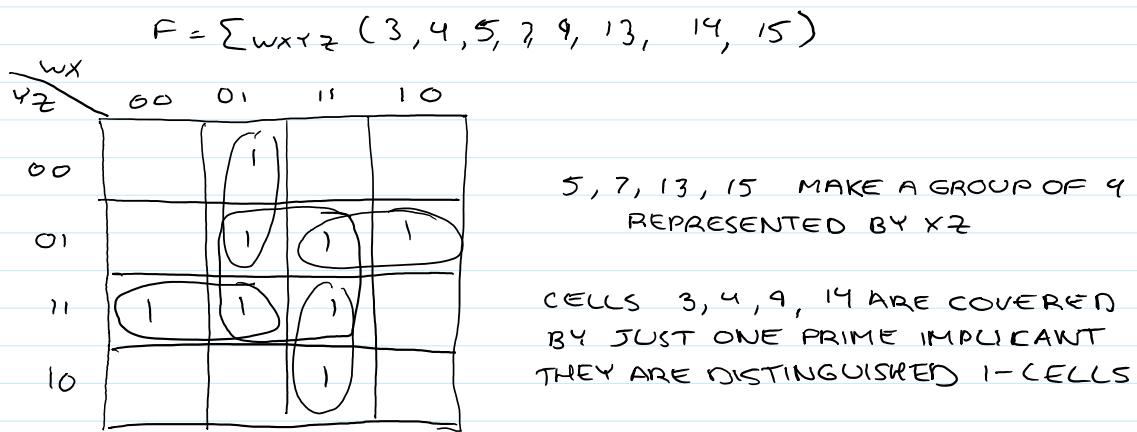
$\therefore "F INCLUDES P" \text{ OR } "F COVERS P"$

- PRIME IMPLICANT OF A LOGIC FUNCTION $F(x_1, \dots, x_n)$ IS A NORMAL PRODUCT TERM $P(x_1, \dots, x_n)$ THAT IMPLIES F , S.T. ANY VARIABLE IS REMOVED FROM P , THEN THE RESULTING PRODUCT TERM DOES NOT IMPLY F

- COMPLETE SUM - SUM OF ALL PRIME IMPLICANTS OF A LOGIC FUNCN

- DISTINGUISHED 1-CELL - INPUT COMBO THAT IS COVERED BY ONLY 1 PRIME IMPLICANT

4 VARIABLE MAP FOR



GROUP REMAINING 4 CELLS

GROUP : $\begin{Bmatrix} 3 \\ 7 \end{Bmatrix} \bar{w}yz$

$\begin{Bmatrix} 9 \\ 13 \end{Bmatrix} w\bar{y}z$

$\begin{Bmatrix} 9 \\ 5 \end{Bmatrix} \bar{w}x\bar{y}$

$\begin{Bmatrix} 14 \\ 15 \end{Bmatrix} wxy$

SINCE THE 4 GROUPS ARE
ESSENTIAL TO COVER 3, 4, 9, 14
THESE ARE ESSENTIAL PRIME
IMPLICANTS

STEPS LEADING TO A MINIMAL EXPRESSION

(1) LOOK FOR DISTINGUISHED 1-CELLS
CIRCLE & ACCEPT ANY BOX THAT CAN'T BE COMBINED
W/ ANY OTHER

(2) CIRCLE AND ACCEPT ANY BOX THAT CAN BE COMBINED W/
JUST ONE OTHER BOX IN ONLY ONE WAY

SKIP THOSE THAT CAN BE GROUPED IN TWO OR MORE SETS

OF TWO

(3) CIRCLE + ACCEPT THOSE THAT CAN BE COMBINED w/ ONLY ONE GROUP OF 4

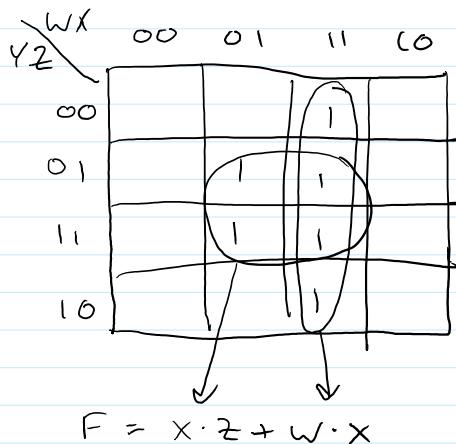
SKIP THOSE THAT CAN BE COMBINED IN TWO OR MORE GROUPS OF FOUR

(4) BOXES OF EIGHT

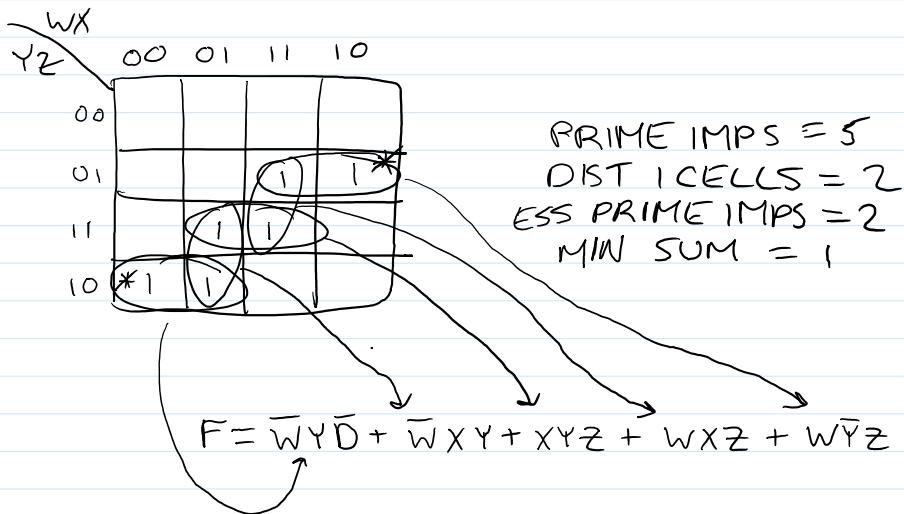
(5) FINALLY, GROUP ANY REMAINING UNCOVERED BOXES IN ANY MINIMAL WAY

BOXES ARE NOT DISTINGUISHED

EX] $F = \sum_{wxxyz} (5, 7, 12, 13, 14, 15)$



ESSENTIAL PRIME IMPlicants COVERS ONE OR MORE DISTINGUISHED CELLS



AR

	AB	CD	00	01	11	10
00			1		1	1
01			1	1		
11				1	1	
10			1		1	1

PRIME IMPLICANTS = 7

DISTINGUISHED 1 CELL = 2

ESSENTIAL PRIME IMPLICANTS = 2

MINIMAL SUMS = 1

$$F = \overline{B}\overline{D} + \overline{A}\overline{B}\overline{C} + \overline{A}\overline{C}\overline{D} + A\overline{B}D + B\overline{C}D + A\overline{B}C + A\overline{D}$$

	AB	CD	00	01	11	10
00						1
01					1	1
11			1	1	1	
10			1	1		

PRIME IMPLICANTS = 6

DISTINGUISHED 1 CELLS = 2

ESSENTIAL PRIME IMPLICANTS = 2

MINIMAL SUMS = 3

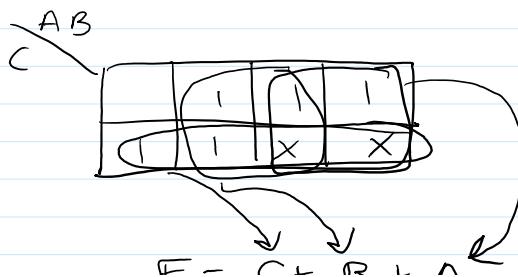
$$F = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{C}\overline{D} + A\overline{C}\overline{D} + B\overline{C}D$$

$$F = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{C}\overline{D} + \overline{A}\overline{B}C + A\overline{B}D$$

$$F = \overline{A}\overline{C}\overline{D} + B\overline{C}D + A\overline{B}D + A\overline{B}\overline{C}$$

DONT CARES

$$F = \sum_{ABC} (1, 2, 3, 4, 6) + d(7, 5)$$



$$F = C + B + A'$$

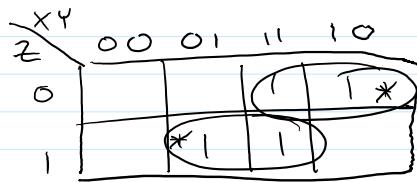
DONT CARES COULD BE EITHER 0 OR 1 AS WE DONT CARE ABOUT THE OUTPUT FOR AN INPUT CONDITION WE EXPECT TO NEVER SEE

STATIC "1" HAZARD

SET OF TWO PRIME IMPLICANTS THAT

- (1) DIFFER IN ONE INPUT VARIABLE
- (2) BOTH GIVE A "1" OUTPUT

LOCATE STATIC "1" HAZARD FOR $F = \sum_{XYZ} (3, 4, 6, 7)$



ESS PRIME IMPLICANTS

$$\begin{aligned} 6 - 4 &\Rightarrow X\bar{Z} \\ 3 - 7 &\Rightarrow YZ \end{aligned}$$

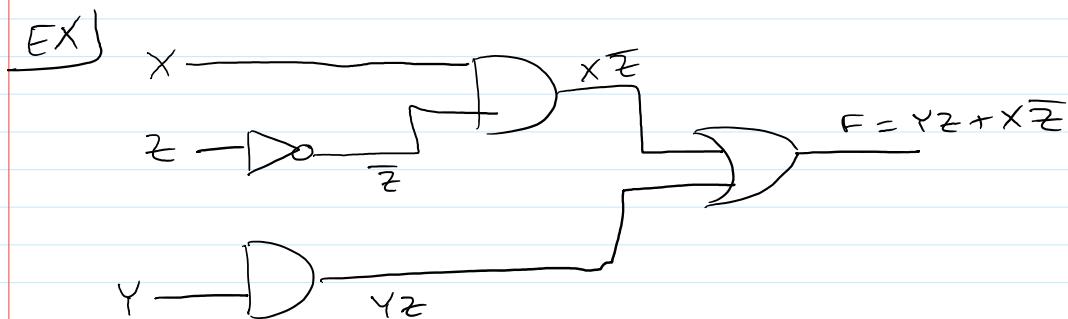
BUT 6 - 7 FORM A STATIC "1" HAZARD

NOTE: 3 AND 4 ARE DISTINGUISHED
6 AND 7 ARE NOT

$$F = X\bar{Z} + YZ$$

HAZARDS DUE TO DELAY IN THE NEGATION OF \bar{Z}

THE "1" INPUT PRODUCES A MOMENTARY "0" OUTPUT

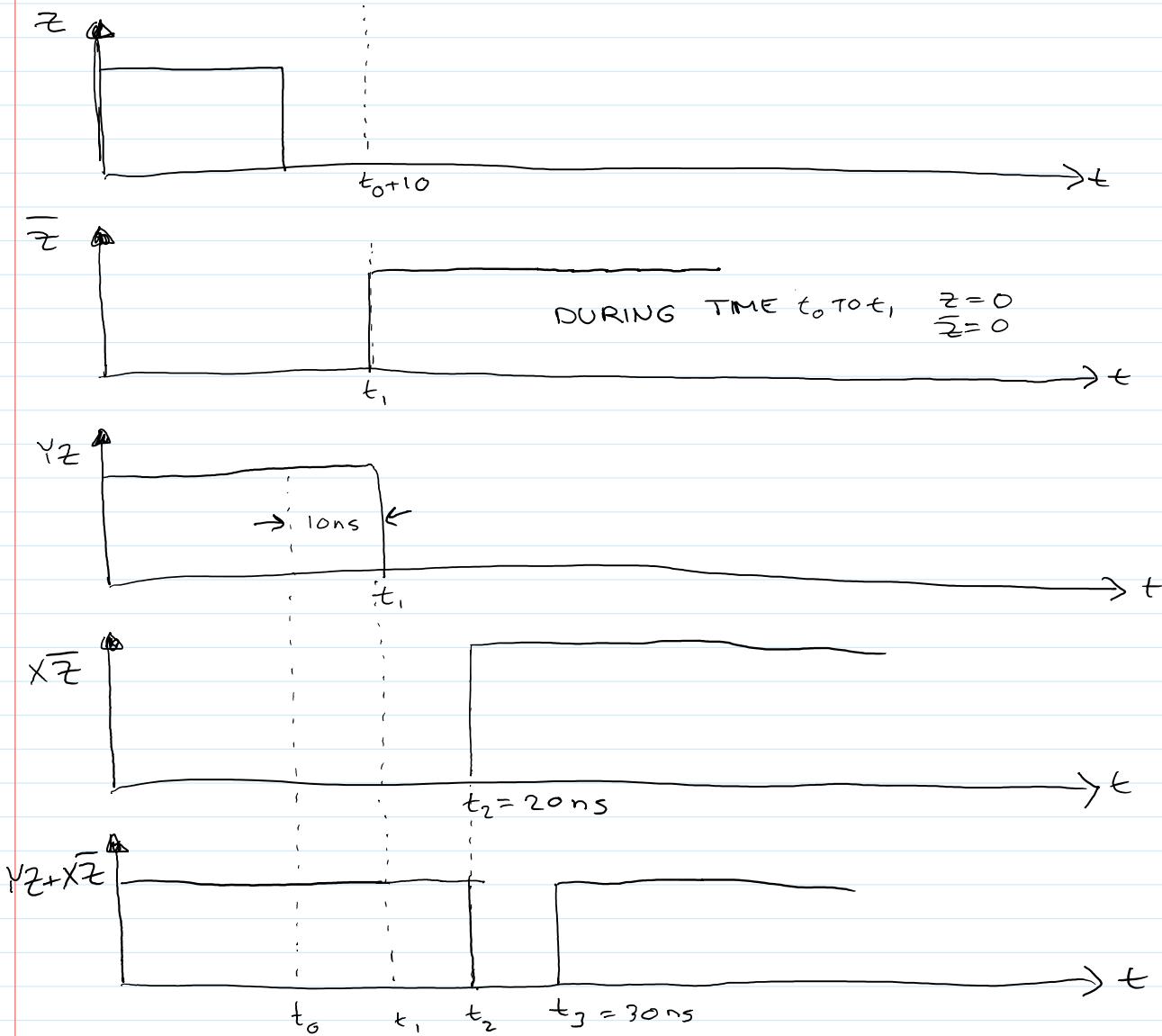


START W/ $X = 1 \quad Y = 1 \quad \bar{Z} = 1$

GATE 2 = 1 GATE 1 = 0 GATE 3 = 0 F = 1

THEN $\bar{z} = 0$ GATE 2 = 0 BUT GATE 1 = 0 STILL IONS
 GATE 3 = 0 F = 0

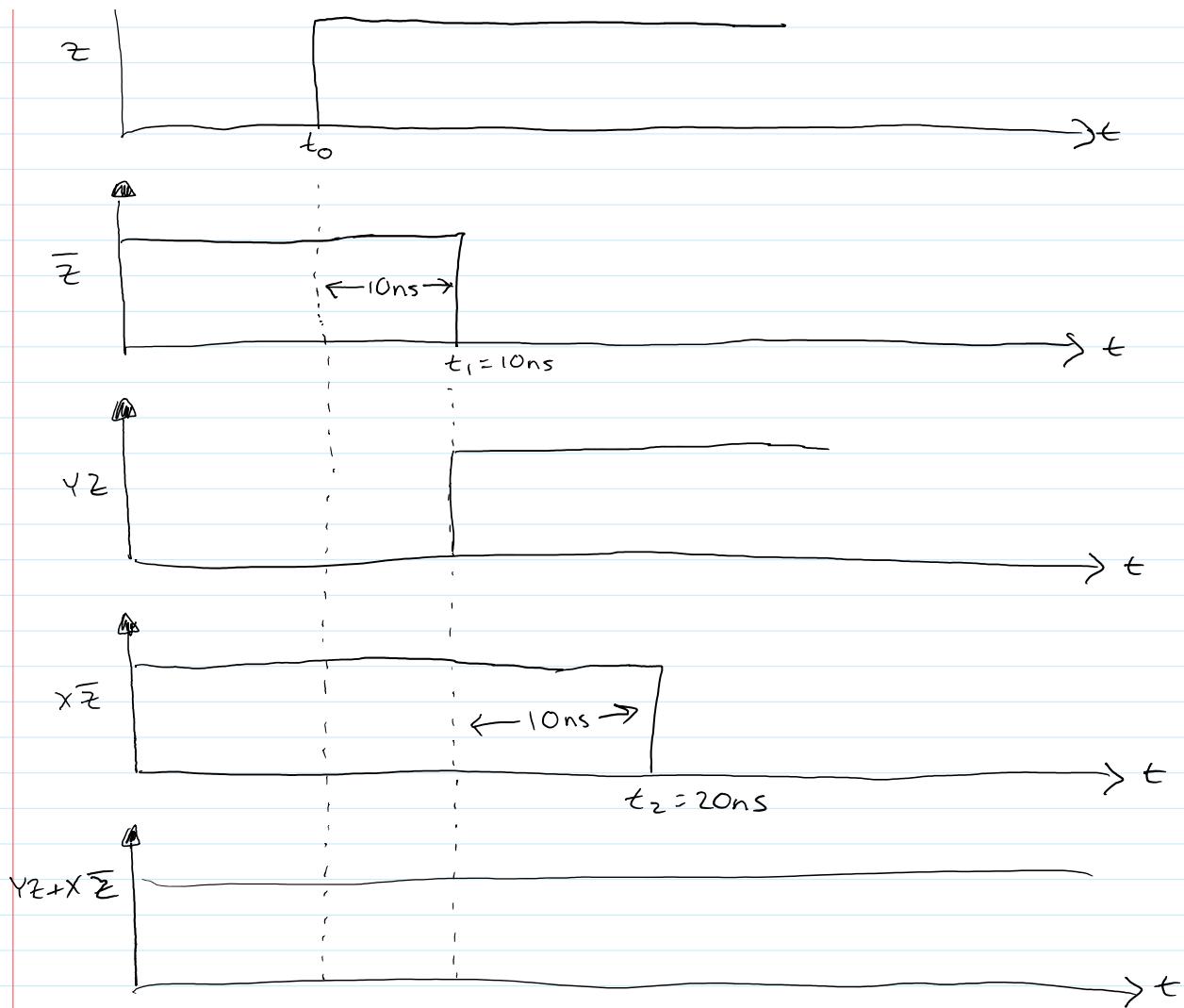
\bar{z} CHANGES FROM "1" TO "0" WHILE FIXED AT $x = 1$ $y = 1$
 GATE DELAY = 10 ns



DURING TIME $t_3 - t_2$ THE OR GATE RECEIVES 0's ON BOTH INPUTS
~~FOR 10 ns~~ \Rightarrow "0" OUT FOR 10 ns

TIMING DIAGRAM FOR $\bar{z} = 0$ TO 1 $x = y = 1$





NO HAZARD

ORIGINAL GROUPINGS CAME FROM RULE 2 THAT IS
 4^* HAD TO BE GROUPED W/ 6 AND 3^* HAD TO BE GROUPED
 W/ 7 THESE GROUPINGS ARE ALL MINTERMS

WE NOTE 6 & 7 DIFFER IN THAT z IS COMPLEMENTED
 IN 6 AND UNCOMPLEMENTED IN 7

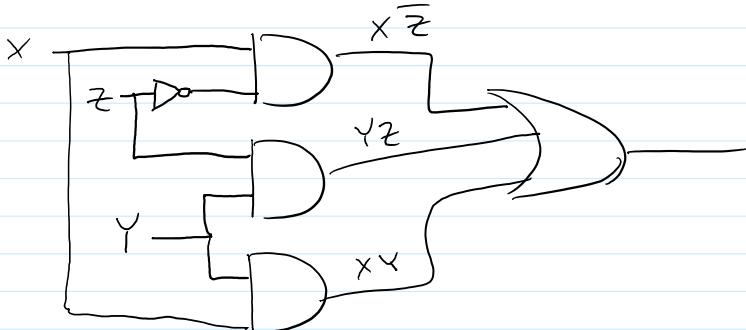
MEANING THAT WHEN z CHANGES LOGIC LEVELS,
 ONE "AND" GATE WILL GO TO "0" BEFORE OTHER "AND" GATE
 GOES TO "1" AND THE OR GATE WILL RECEIVE TWO 0's
 BRIEFLY

XY	00	01	11	10
z	0	1	1	1
0	1	1	1	1
1	1	1	1	1

THE FIX

NOTE: THAT AS LONG AS $X=1 + Y=1$ WE WANT $F=1$

TO FIX, WE CIRCLE BOXES 6 W/ 7 AS WELL AS
KEEPING THE ORIGINAL GROUPS OF 6, 4 + 3, 7

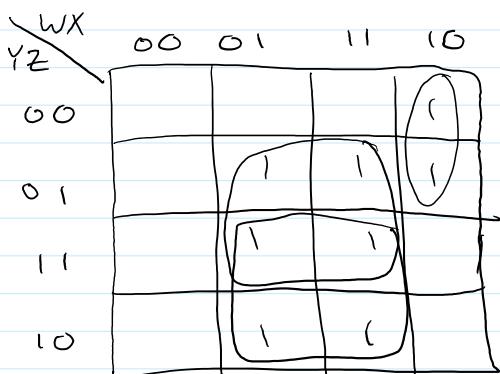


$$F = X\bar{Z} + YZ + XY$$

WE CAN SHOW USING A TRUTH TABLE, THAT THE $X \cdot Y$ TERM IS REDUNDANT BY CONSENSUS BUT IT REMOVES THE STATIC 1 HAZARD

DETERMINE IF THE FOLLOWING FCN HAS ANY STATIC HAZARDS

$$F = W\bar{X}\bar{Y} + X\bar{Y} + XY$$



$$F = \sum (5, 6, 7, 8, 9, 13, 14, 15)$$

CANONICAL SUM w/
MINTERMS

HAZARDS: 9 W/ 13 Δ X

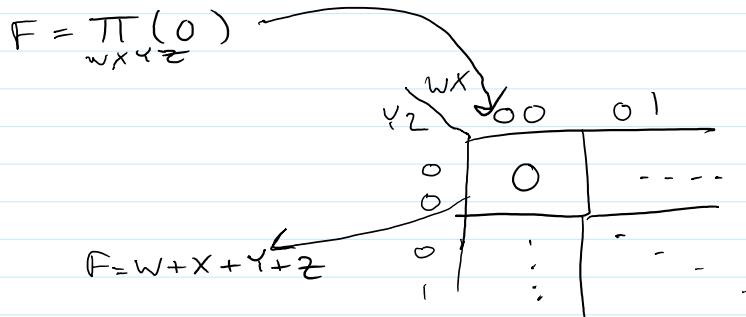
DIST 1-CELLS: 5, 8, 6, 14

FIX: GROUP 9 w/ 13 \Rightarrow $w\bar{y}z$

$$F = w\bar{x}\bar{y} + xz + xy + w\bar{y}z$$

STATIC 0 HAZARD

REVIEW OF PRODUCT TERM



TO REPRESENT THE MAX TERM THE FCN HAS TO EQUAL 0 AT LOCATION 0 AND 1 EVERYWHERE ELSE

THE OR GATE OF $(w+x+y+z) = F$

F WILL EQUAL 0 ONLY WHEN $w=0 x=0 y=0 z=0$

NOTE: IF THERE WOULD BEEN A 1 @ POSITION 0

MINTERM THEN WOULD BE $F = \bar{w}\bar{x}\bar{y}\bar{z} = 1$

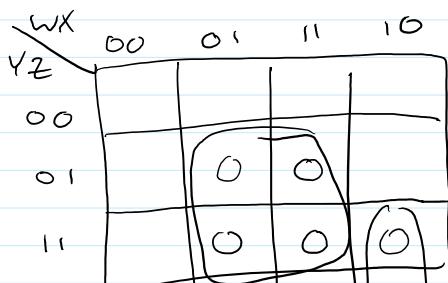
$$\begin{array}{l} w=0 \\ \text{IF } x=0 \\ y=0 \\ z=0 \end{array}$$

EXAMPLE OF DUALITY

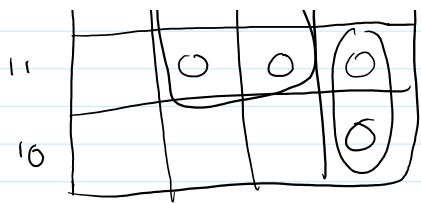
EX) $F = \prod(5, 7, 10, 11, 13, 15)$

RULE 1) NONE

RULE 2) 10 w/ 11
RULE 3) 5, 7, 13, 15



DIST 0 CELLS = 10, 5, 7, 13



WISI 0 CELLS = 10, 5, 1, 1

$$F = (\bar{W} + X + \bar{Y}) \cdot (\bar{X} + \bar{Z})$$

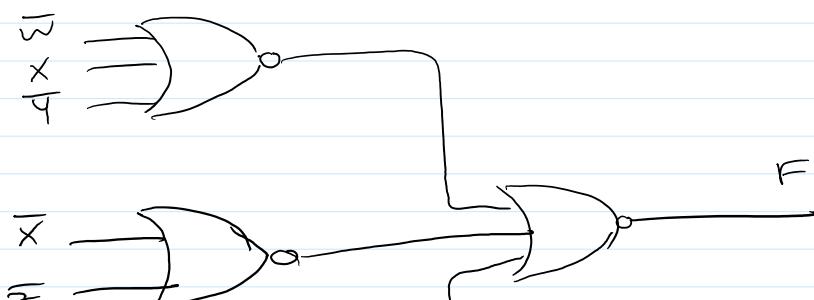
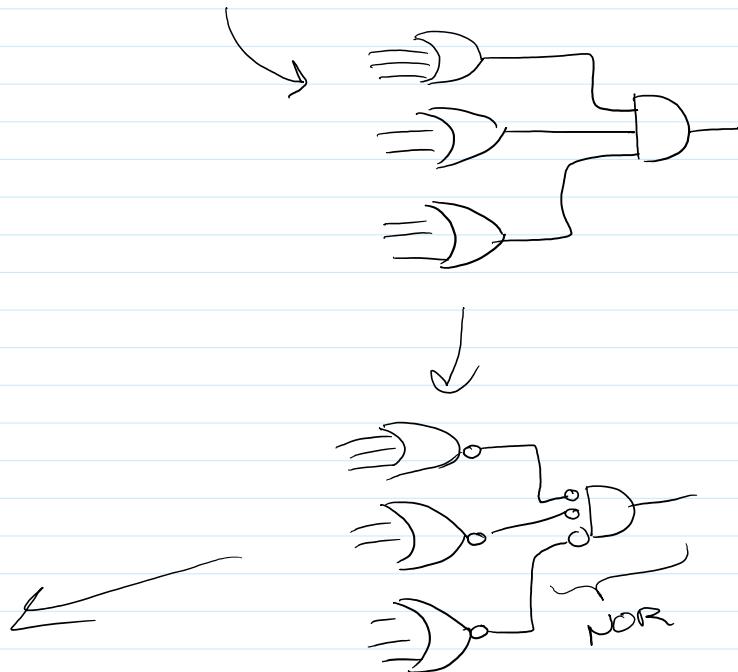
15 w/ 11 CAUSES STATIC "0" HAZARD WHEN X GOES FROM "0" TO "1"

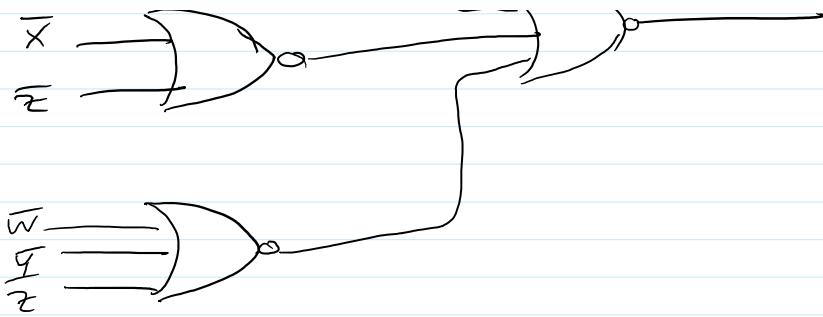
THE COMPLIMENT WILL BE DELAYED, HENCE, BOTH OR GATES WILL PUT OUT A "1" AND $F = 1 \Rightarrow$ ERROR

TO FIX: ADD ANOTHER OR GATE & ANOTHER INPUT TO THE AND GATE

$$\text{CIRCLING BOXES } 15 \& 1 \Rightarrow (\bar{W} + \bar{Y} + \bar{Z})$$

$$F = (\bar{W} + X + \bar{Y}) \cdot (\bar{X} + \bar{Z}) \cdot (\bar{W} + \bar{Y} + \bar{Z})$$





OR - AND REPRESENTED BY NOR - NOR

EX) $F = \sum_{wx,y,z} (0, 2, 3, 4, 5, 7, 8, 9, 13, 15)$

		WX	YZ	00	01	11	10
		00	00	1	1		1
		01	01		1	1	1
		11	11	1	1	1	
		10	10	.			

- 1) NO SINGLES THAT ARE ALONE
- 2) NO DOUBLES S.T. NEITHER BOX CAN BE COMBINED w/ ANOTHER
- 3) BOXES 5, 7, 13, 15 CAN COMBINED IN ONE UNIQUE WAY SUCH THAT NONE OF THE BOXES WILL FIT IN ANY OTHER GROUP OF 4 $\Rightarrow XZ$

6 REMAINING BOXES (0, 2, 3, 4, 8, 9) THAT CAN BE GROUPED IN ANY MINIMAL MANNER

GROUP 0 + 4 $\Rightarrow \bar{W}\bar{X}\bar{Y}$

8 + 9 $\Rightarrow W\bar{X}\bar{Y}$

2 + 3 $\Rightarrow \bar{W}\bar{X}Y$

NOTE: WE SKIPPED RULE 2 B.C. WE HAD TO FIND GROUP OF FOUR FIRST

$$\therefore F = XZ + \bar{W}\bar{X}\bar{Y} + W\bar{X}Y + \bar{W}\bar{X}Y$$

DIST 1 CELLS : 15

STATIC "1" HAZARDS

$$9 \text{ w/ } 13 \quad \Delta X \Rightarrow W \cdot \bar{Y} \cdot Z$$

$$3 \text{ w/ } 7 \quad \Delta X \Rightarrow \bar{W} \cdot Y \cdot Z$$

$$4 \text{ w/ } 5 \quad \Delta Z \Rightarrow \bar{W} \cdot X \cdot \bar{Y}$$

$$2 \text{ w/ } 0 \quad \Delta Y \Rightarrow \bar{W} \cdot \bar{X} \cdot \bar{Z}$$

$$0 \text{ w/ } 8 \quad \Delta W \Rightarrow \bar{X} \cdot \bar{Y} \cdot \bar{Z}$$

$$F = XZ + \bar{W}\bar{X}\bar{Y} + W\bar{X}\bar{Y} + \bar{W}\bar{X}Y + W\bar{Y}Z + \bar{W}YZ + \bar{W}X\bar{Y} + \bar{W}\bar{X}\bar{Z} + \bar{X}\bar{Y}\bar{Z}$$

EX $F = \prod_{wxyz} (0, 3, 4, 5, 6, 7, 11, 13, 15)$

FIND + FIX STATIC 0 HAZARD

		WX	YZ	00	01	11	10
		00	00	0	0		
		01	01	0	0		
		11	11	0	0	0	0
		10	10	0			

- 1) NO SINGLES
- 2) 0 w/ 4 ONLY
- 3) 13 \leftrightarrow 5, 7, 15
- 3 \leftrightarrow 7, 11, 15
- 11 \leftrightarrow 3, 7, 15
- 6 \leftrightarrow 4, 5, 7

DIST 0 CELLS: 0, 3, 6, 11, 13

GROUP: 5, 7, 13, 15 $\Rightarrow (\bar{X} + \bar{Z})$

3, 7, 11, 15 $\Rightarrow (\bar{Y} + \bar{Z})$

4, 5, 6, 7 $\Rightarrow (W + \bar{X})$

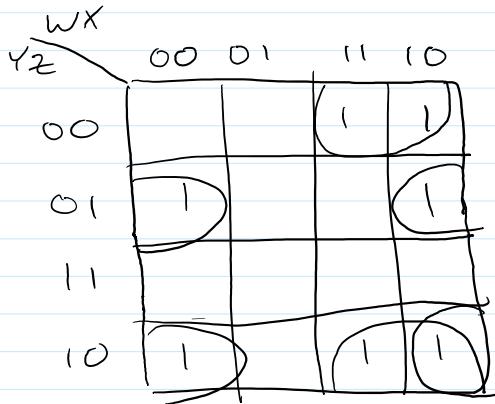
NO HAZARDS

$$0, 4 \Rightarrow (w + y + z)$$

$$F = (w + y + z)(\bar{x} + \bar{z})(\bar{y} + \bar{z})(w + \bar{x})$$

Ex $F = \sum(1, 2, 8, 9, 10, 12, 14)$

FIND STATIC "1" HAZARDS + DIST 1 CELLS



1) NONE

2) 1 w/ 9 $\Rightarrow \bar{x}\bar{y}z$

2 w/ 10 $\Rightarrow \bar{x}yz$

3) 8 w/ 10, 12, 14 $\Rightarrow w\bar{z}$

STATIC 1 HAZARD

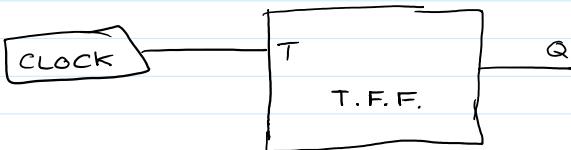
8 w/ 9 $\Delta z \Rightarrow w\bar{x}\bar{y}$

DIST 1-CELLS: 1, 2, 12, 14

BISTABLE MULTIVIBRATOR

WAVEFORMS

TOGGLE FLIP-FLOPS (DIVIDE BY 2 FREQUENCY)

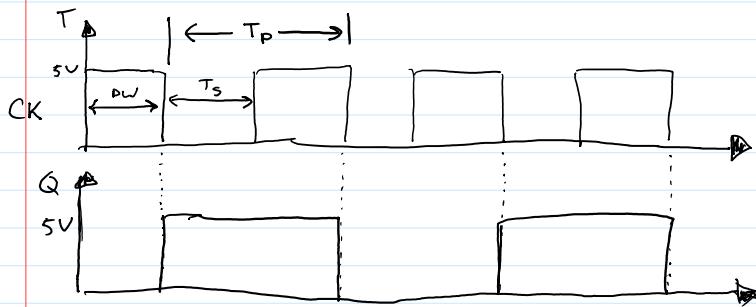


PULSE TRAIN

$$T_p = \text{PERIOD} = 1/f \quad T_s = \text{SPACE}$$

PW = PULSE WIDTH

$$\text{DUTY CYCLE} = \frac{\text{PW}}{T_p} \times 100\%$$



EVERY NEGATIVE EDGE TOGGLES FLIP-FLOP W/ NO DELAY

$$f(\tau) = \frac{1}{T_p(\tau)} \quad T_p(Q) = 2T_p(\tau)$$

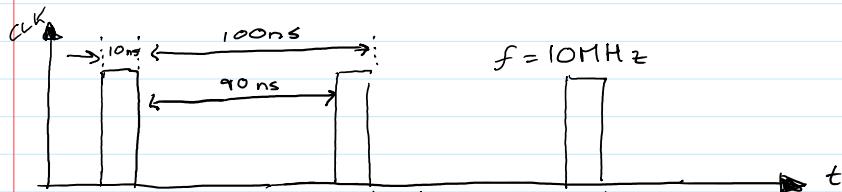
$$f(Q) = f(\tau)/2 \quad \therefore \text{DIVIDE BY 2}$$

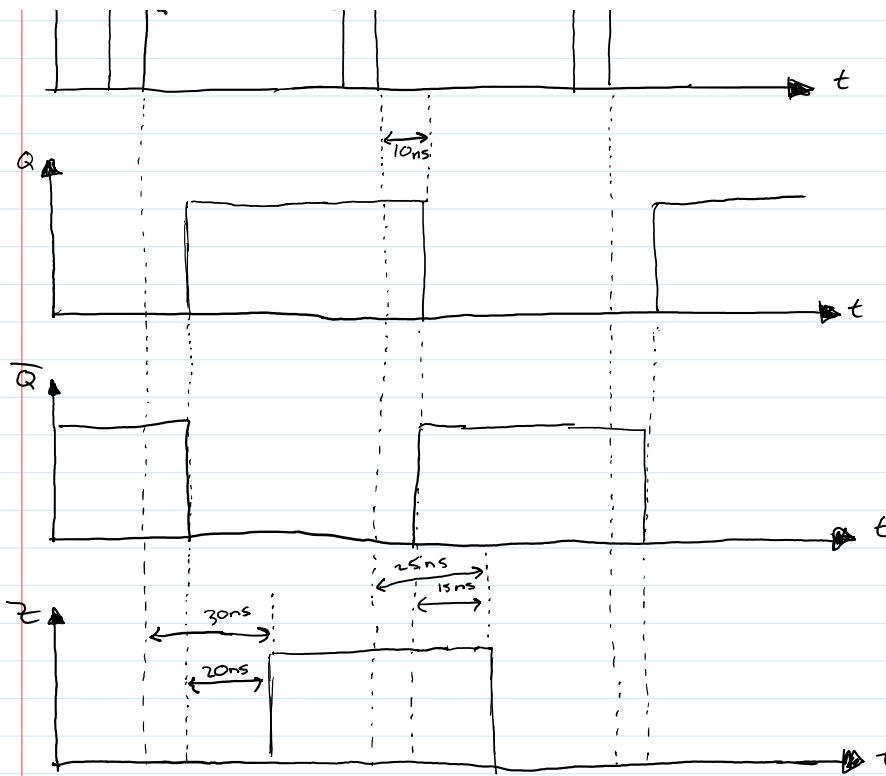
EX] NEGATIVE EDGE TOGGLE FLIP FLOP



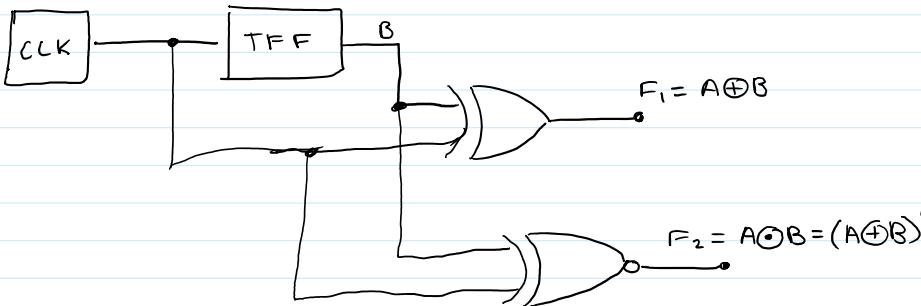
$$T_{PO}(\text{FF}) = 10 \text{ ns} \quad T_{POH(\text{INV})} = 15 \text{ ns} \quad T_{PDCK(\text{INV})} = 20 \text{ ns}$$

$$\text{D.C.} = 10\% \quad T_p = 100 \text{ ns} \quad \text{PW} = 10 \text{ ns} \quad T_s = 90 \text{ ns}$$





TIMING EXAMPLE

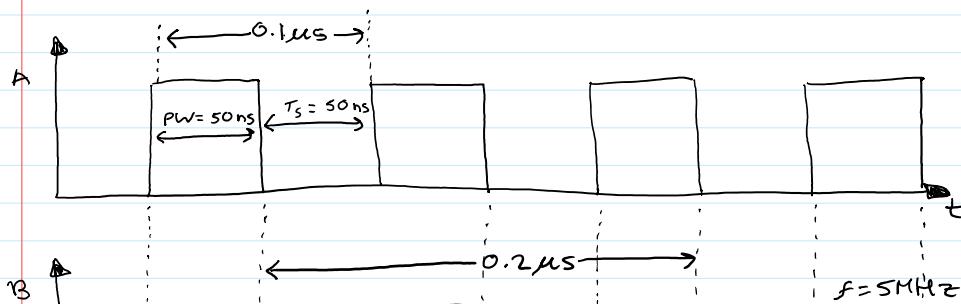


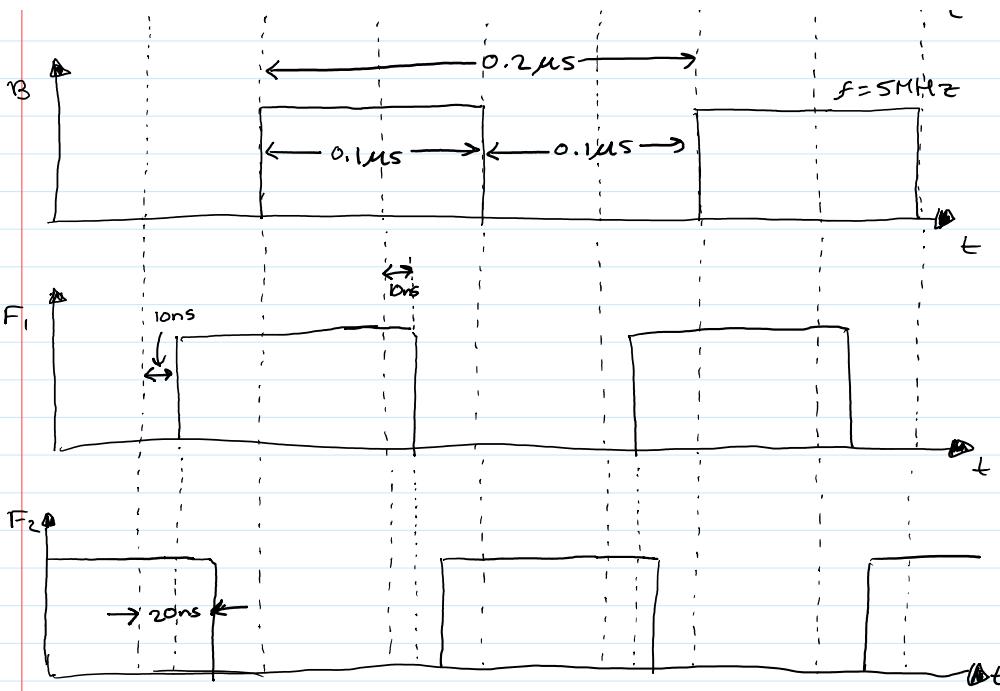
NO DELAY w/ F.F.

10 ns DELAY THROUGH XOR GATE

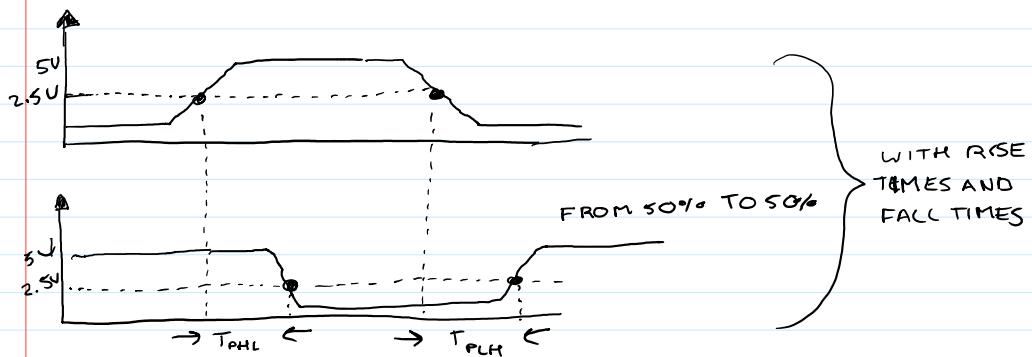
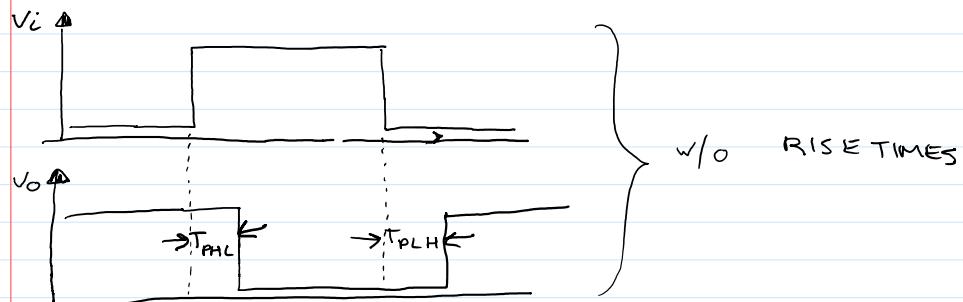
20 ns DELAY THROUGH XNOR GATE

$f=10\text{MHz}$ 50% DUTY CYCLE



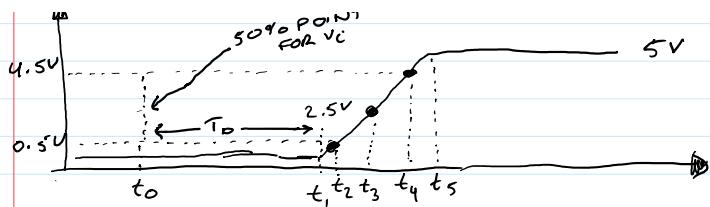


PROPAGATION DELAY INVERTOR $v_i \rightarrow v_o$



T_{PLH} CONSISTS OF TWO COMPONENTS

- (1) INTERNAL DELAY
- (2) RISE TIME TO 50%



$$T_{PDH} = t_3 - t_0 \quad \text{IF } t_f \text{ OF } V_i = 0$$

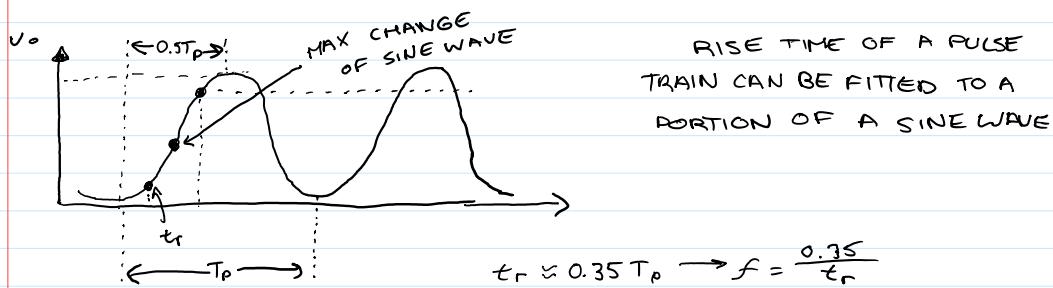
$T_d = t_1 - t_0 = \text{INTERNAL DELAY}$

$t_3 = 50\%$ POINT FOR T_{PLH}

$t_4 - t_2 = \text{RISE TIME OF SIGNAL 10\% TO 90\%}$

USEFULL FOR DETERMINING SYSTEM BANDWIDTH
TIME FOR OUTPUT Q'S TO DRIVE C_L

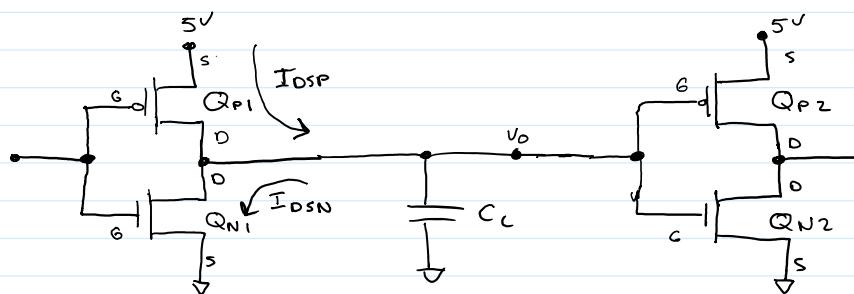
CLOSER LOOK AT RISE TIME TO BANDWIDTH



$$\text{EX)} \quad t_r = 1 \text{ ms} \rightarrow f \approx 300 \text{ MHz}$$

RISE TIME COMES FROM C_L OF DRIVER TRANSISTORS, RECEIVER TRANSISTORS, AND ROUTING WIRE

$$\text{NOTE: } \frac{\Delta V_o}{\Delta t} \text{ (OF DIGITAL SIGNAL)} = \left. \frac{\Delta V}{\Delta t} \right|_{\text{MAX}} \text{ (OF SINEWAVE)}$$



$$\text{IF } |V_{GS}| = 5 \text{ V} \Rightarrow \text{D.S.} = I_{DS} \quad V_{GS} = 0 \Rightarrow \text{D.S.} = \text{OPEN CIRCUIT}$$

$$C_L = C_o(Q_{p1}) + C_o(Q_{n1}) + C_i(Q_{p2}) + C_i(Q_{n2}) + C_w$$

C_w = CAPACITANCE DUE TO METAL ROUTE FROM G1 TO G2

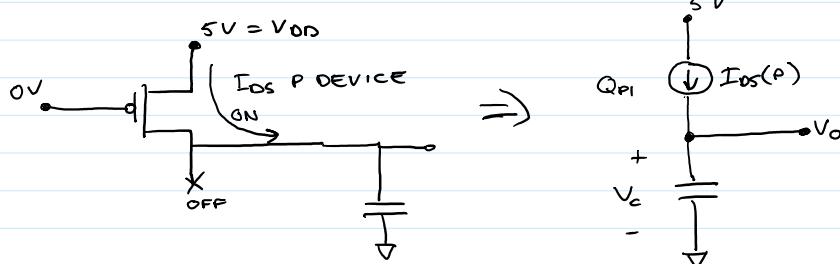
$$C_0 \approx 0.1 \text{ pF} - 1 \text{ pF}$$

NOTE: C_L IS NOT A PHYSICAL CAP
BUT A COMPOSITE OF
SEVERAL ITEMS

$$C_i \approx 0.01 \text{ pF} - 0.1 \text{ pF}$$

$$C_w \approx 1-2 \text{ pF/cm}$$

RISE TIME OCCURS WHEN OUTPUT VOLTAGE IS 0V AND RISES TO 5V.



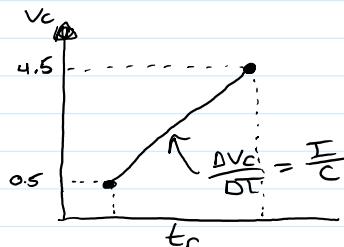
$$Q = CV_C \quad \text{CHARGE IN CAP} = C \cdot V_o$$

$$\text{AND } I = \dot{Q} = \frac{\Delta Q}{\Delta T} = C \frac{\Delta V_C}{\Delta T}$$

$$t_r: \Delta V_C = 0.9 V_{DD} - 0.1 V_{DD}$$

$$\Delta T = t_r$$

$$t_r = \frac{\Delta V_C C}{I}$$



$$\underline{\text{EX}} \quad C_0(Q) = 1 \text{ pF} \quad C_i(Q) = 0.2 \text{ pF} \quad L_{\text{ROUTE}} = 3 C_W @ 2 \text{ pF/cm}$$

$$V_{DD} = 5V$$

$$I_{DS}(P) = 100 \text{ mA} \quad \text{FIND: } t_r + \text{BANDWIDTH OF SYSTEM}$$

$$2 \text{ pF/cm}$$

$$C_L = 2 \cdot C_0 + 2 \cdot C_i + 3 \cdot C_W$$

$$C_L = 2 + 0.4 + 6 = 8.4 \text{ pF}$$

$$\Delta V_o = 4.5V - 0.5V = 4V$$

$$t_r = \frac{4 \cdot 8.4 \text{ pF}}{100 \text{ mA}} = \frac{33.6}{100} \text{ ns} = 0.34 \text{ ns}$$

$$\text{BANDWIDTH} = 0.35/t_r = 0.35/0.34 \approx 1 \text{ GHz}$$

$$L = \frac{C}{f} = \frac{300 \text{ Km/s}}{1000 \text{ MHz}} = 0.3 \text{ meters } \mu\text{WAVE}$$

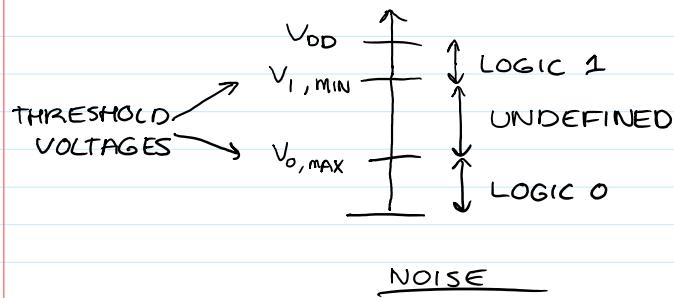
CONSTRAINTS IN LOGIC CKT DESIGN

WE HAVE FOCUSED ON THE DESIGN OF LOGIC CKTS IN TERMS OF THEIR LOGICAL BEHAVIOR ONLY

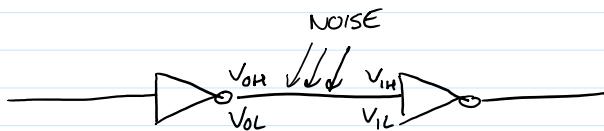
WHEN DESIGNING YOU MUST CONSIDER REAL-WORLD CONSTRAINTS SUCH AS

- NOISE
- FAN-OUT
- FAN-IN
- POWER CONSUMPTION
- TIME DELAY
- TRANSIENT BEHAVIOR

VOLTAGE RANGE SPECIFIED FOR EACH LOGIC LEVEL



EXTERNAL NOISE SOURCES CAN CAUSE THE LOGIC GATE OUTPUT VOLTAGES TO DEVIATE FROM THEIR EXPECTED VALUES



AS A RESULT, VOLTAGES MAY BE MISINTERPRETED

- EX)
- OUTPUT LOW VOLTAGE NOT INTERPRETED AS A LOGIC 0
 - OUTPUT HIGH " " " " " " " "

MUST SELECT LOGIC GATES TO ALLOW THE LOGIC CKT TO FUNCTION PROPERLY IN THE PRESENCE OF NOISE

NOISE MARGIN - AMOUNT OF NOISE THAT THE LOGIC CKT CAN WITHSTAND WHILE STILL FUNCTIONING PROPERLY



MEASURE OF NOISE IMMUNITY PROVIDED BY CKT

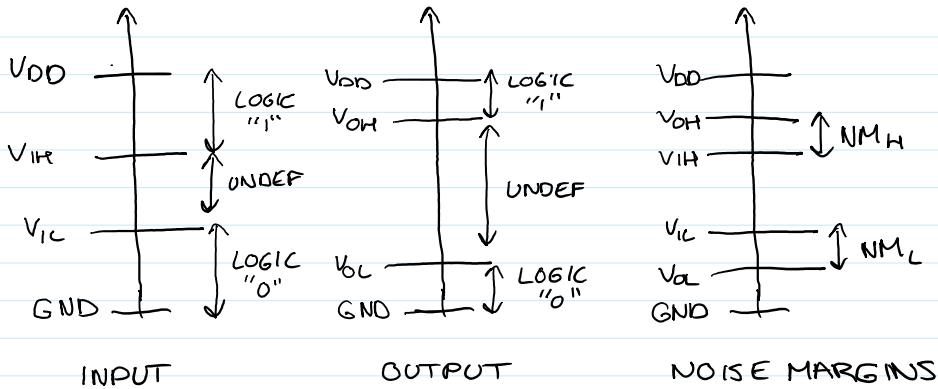
THE NOISE MARGIN IS DEFINED FOR BOTH LOGIC "1" AND LOGIC "0"

$$NM_H = V_{OH} - V_{IH}$$

HIGH NOISE MARGIN

$$NM_L = V_{IL} - V_{OL}$$

LOW NOISE MARGIN

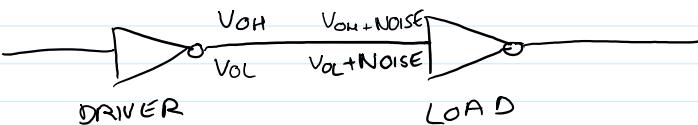


THE NOISE MARGIN MUST BE POSITIVE, FOR BOTH LOGIC 1 AND LOGIC 0, FOR THE CKT TO FUNCTION

$$V_{OH}(\text{DRIVER}) > V_{IH}(\text{LOAD})$$

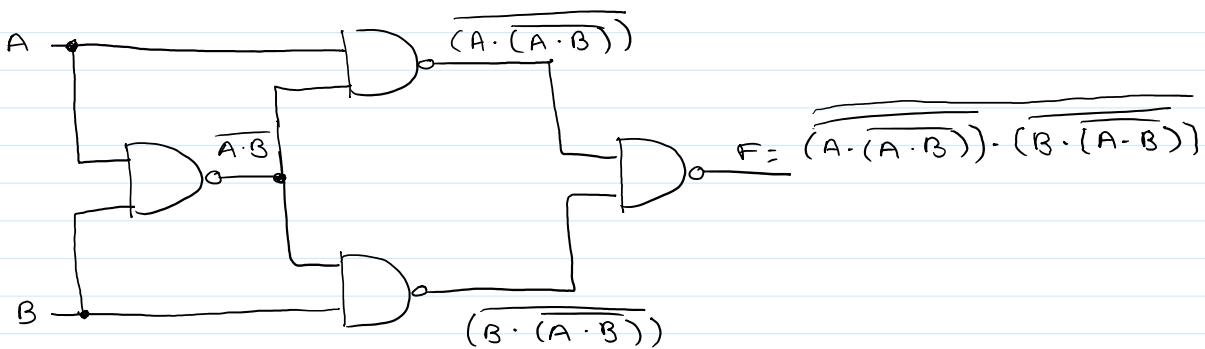
$$V_{OL}(\text{DRIVER}) < V_{IL}(\text{LOAD})$$

NEGATIVE NOISE MARGIN IMPLIES THAT THE VOLTAGE OUTPUT BY THE DRIVING GATE WILL NOT BE INTERPRETED PROPERLY BY THE LOAD GATES



EXCLUSIVE OR

4 NAND REPRESENTATION XOR



$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{(A+B)} = \overline{A} \cdot \overline{B}$$

← DE MORGANS

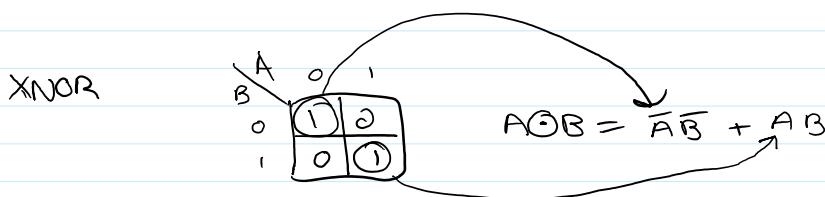
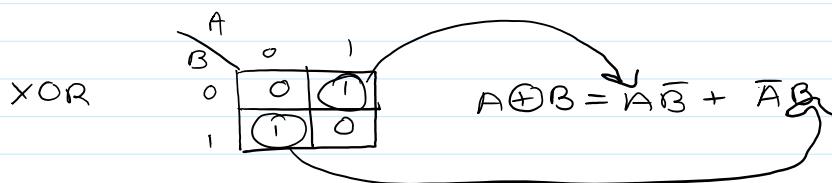
$$F = \overline{(A \cdot (\overline{A} \cdot B))} \cdot \overline{(B \cdot (\overline{A} \cdot B))}$$

$$= (A \cdot \overline{(A \cdot B)}) + (B \cdot \overline{(A \cdot B)})$$

$$= [A \cdot (\bar{A} + \bar{B})] + [B \cdot (\bar{A} + \bar{B})]$$

$$= A\bar{A} + A\bar{B} + B\bar{A} + B\bar{B}$$

K-MAP REPRESENTATION



$$A \oplus B = \overline{(A \odot B)}$$

$$A \odot B = \overline{(A \oplus B)}$$

3 VARIABLE XOR

$A \oplus B \oplus C$ IS DEFINED AS

$$A = 1 \quad B = C = 0$$

OR

$$B = 1 \quad A = C = 0$$

OR

$$C = 1 \quad A = B = 0$$

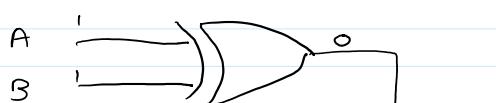
OR

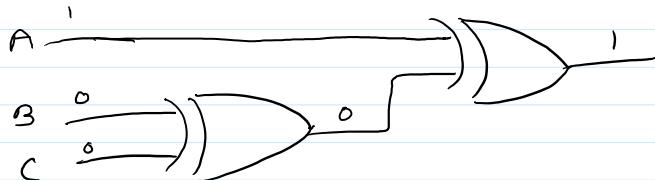
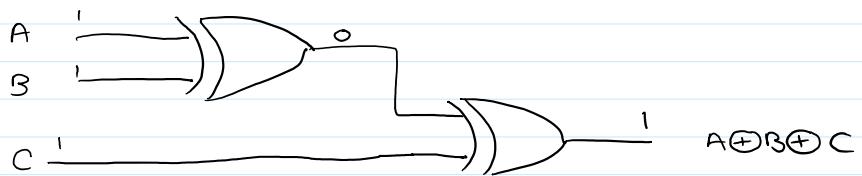
$$A = B = C = 1$$

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

NOTE : 000 ≠ 1

CKT

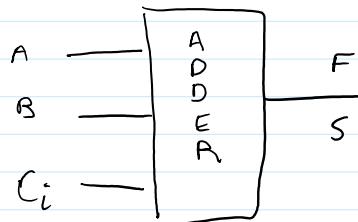




Ex] 3 VARIABLE ODD BIT DETECTOR

$$F = A \oplus B \oplus C$$

	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1



$$\Rightarrow F = \sum(1, 2, 4, 7)$$

$F = 1$ IF OOD # OF BITS

		A	B	C	00	01	11	10
0					0	1	1	1
1					1	1	1	1

$$1 + 2 \Rightarrow \overline{A} \overline{B} C + \overline{A} B \overline{C} = \overline{A} \cdot (B \oplus C)$$

$$4 + 7 \Rightarrow A \overline{B} \overline{C} + A B C = A \cdot (B \odot C) = A \cdot (\overline{B \oplus C})$$

$$F = \overline{A} \cdot (B \oplus C) + A \cdot (\overline{B \oplus C})$$

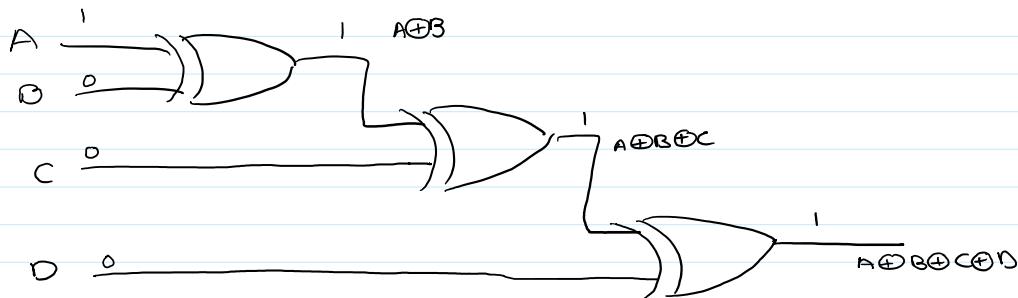
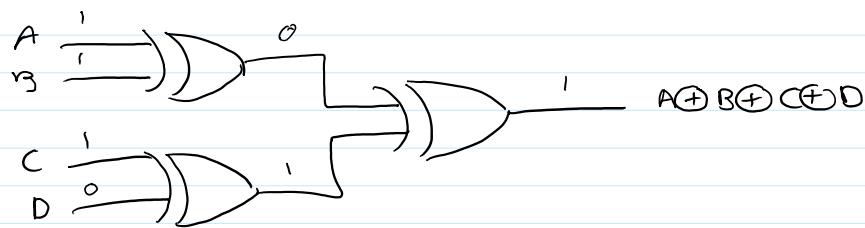
$$\text{LET } X = B \oplus C$$

$$F = \overline{A} X + A \overline{X}$$

$$F = A \oplus X = A \oplus B \oplus C$$

4 VARIABLE XOR CKT





ABOVE CKT WILL DETECT ODD # BITS

$$4 \text{ VARIABLES } F = A \oplus B \oplus C \oplus D$$

$$F = 1 \text{ IF } A \text{ OR } B \text{ OR } C \text{ OR } D = 1 \text{ OTHER } 3=0$$

OR

$$\left\{ \begin{array}{ll} A = B = C = 1 & D = 0 \\ A = C = D = 1 & B = 0 \\ B = C = D = 1 & A = 0 \\ A = B = D = 1 & C = 0 \end{array} \right. \text{ OTHER 8 COMBOS } = 0$$

NOTE: AGAIN ODD # OF ONES = 1

4 CASES OF 1 VARIABLE = 1

AND

4 CASES OF 3 VARIABLE = 1

AB	CD	00	01	11	10
00		1		1	
01		1	1		
11		1	1	1	
10		1	1	1	

8 "1"

$$F = \sum_{ABCD} (1, 2, 4, 7, 8, 11, 13, 14)$$

APPLICATIONS FOR XOR

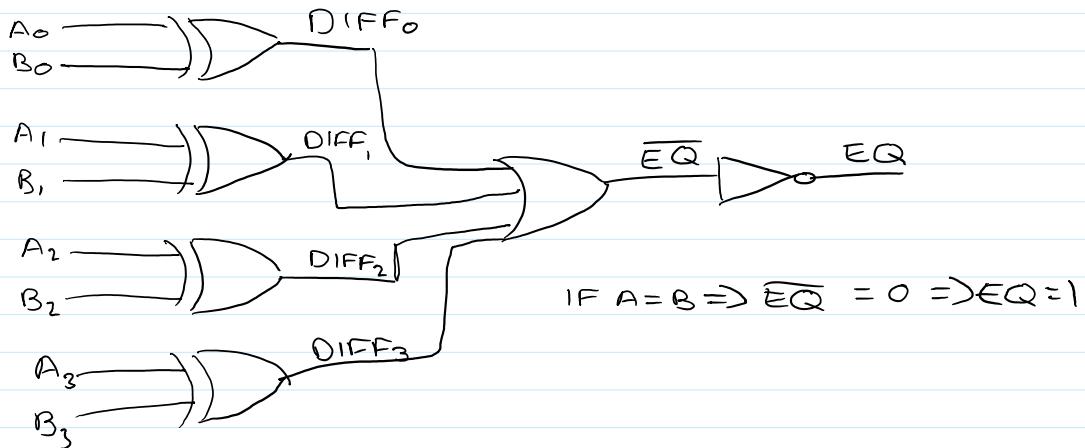
APPLICATIONS FOR XOR

ARITHMETIC CKTS USING AN XOR GATE AS AN EQUALITY COMPARATOR



IF $A = B \Rightarrow \text{DIFF} = 0$ IF NOT THEN $\text{DIFF} = 1$

4 BIT COMPARATOR - GIVEN TWO NIBBLES $A = A_3 A_2 A_1 A_0$
 $B = B_3 B_2 B_1 B_0$

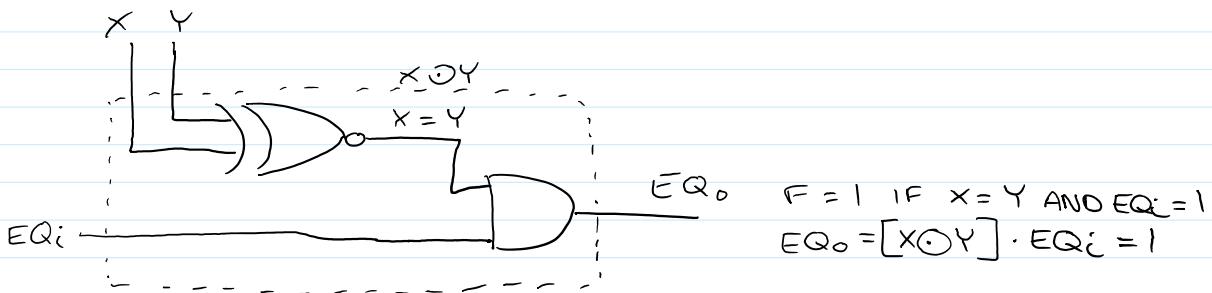


IF ANY SET OF BITS ARE NOT EQUAL THAN A "1" IS SENT TO THE OR GATE

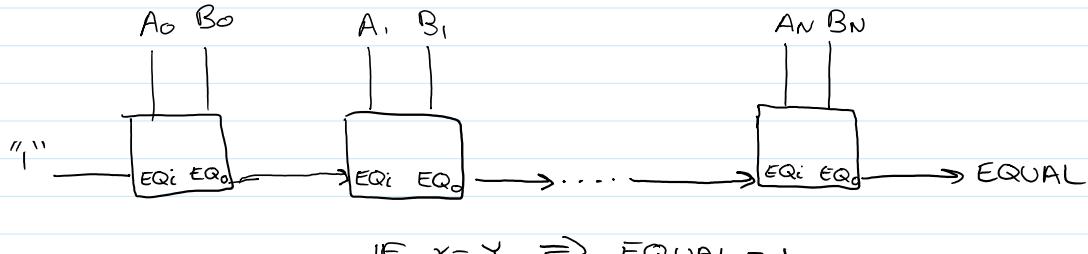
ITERATIVE APPROACH EQUAL COMPARATOR CKT

DOES WORD 1 = WORD 2

FIRST BLOCK CKT w/ 3 IN AND 1 OUT



RIPPLE EQUAL COMPARATOR



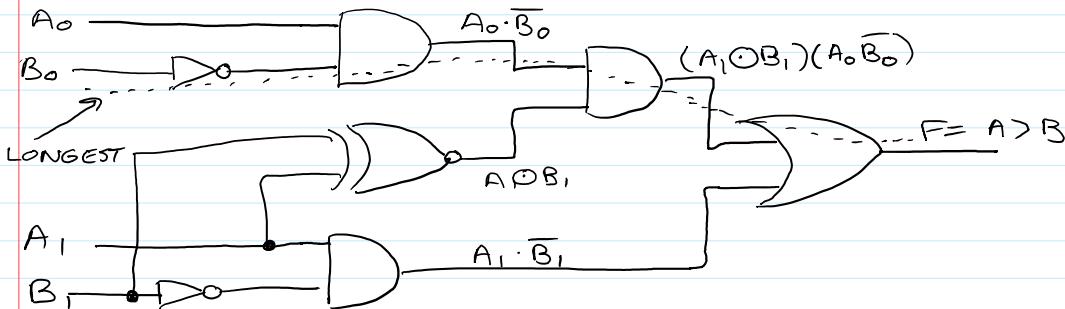
A > B CKT USED FOR SUBTRACTION

$$1\text{-BIT} \quad A > B = A \cdot \bar{B}$$

$$2\text{-BIT} \quad A_1 A_0 > B_1 B_0$$

$$F = A_1 > B_1 + (A_1 = B_1) \cdot (A_0 > B_0)$$

$$= A_1 \cdot \bar{B}_1 + (A_1 \oplus B_1) \cdot (A_0 \cdot \bar{B}_0)$$



TIMING: THIS CKT REQUIRES 7 LEVELS OF GATES

INV, NAND, NOR = 1 LEVEL

AND, OR = 2 LEVELS

HOW CAN WE DO BETTER?

FOR A FASTER VERSION WE RECOGNIZE THAT

$$A_1 \oplus B_1 = A_1 B_1 + \bar{A}_1 \bar{B}_1$$

$$A > B = A_1 \bar{B}_1 + (A_1 B_1 + \bar{A}_1 \bar{B}_1) \cdot (A_0 \bar{B}_0)$$

DISTRIBUTE $(A_0 \bar{B}_0)$

DISTRIBUTE $(A_0 \bar{B}_0)$

$$A > B = A_1 \bar{B}_1 + A_1 B_1 A_0 \bar{B}_0 + \bar{A}_1 \bar{B}_1 A_0 \bar{B}_0$$

(1) (2) (3)

TWO WAYS PROCEEDING

FIRST MATHEMATICALLY

$$\text{RECALL CONSENSUS: } A + \bar{A}B = A + B$$

$$\text{SO THAT: } AB + A\bar{B}C = A(B + \bar{B}C) = AB + AC$$

$$\text{LIKEWISE: } A\bar{B} + ABC = A\bar{B} + AC$$

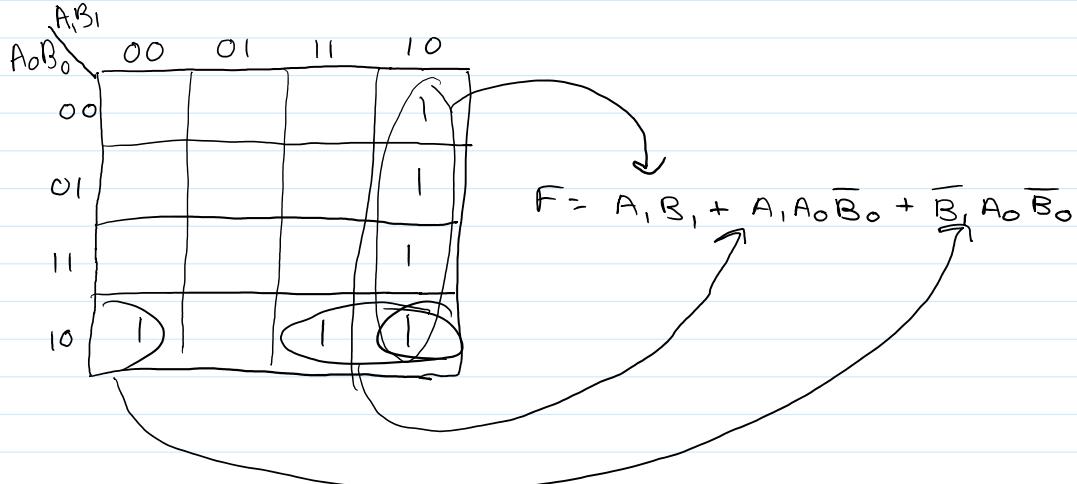
COMBINE TERMS 1+2 AND 1+3 AND GET

$$A_1 \bar{B}_1 + A_1 A_0 \bar{B}_0 \text{ AND } A_1 \bar{B}_1 + \bar{B}_1 A_0 \bar{B}_0$$

$$\therefore A > B = A_1 \bar{B}_1 + A_1 A_0 \bar{B}_0 + \bar{B}_1 A_0 \bar{B}_0$$

3 LEVELS TOTAL USING INV-NAND-NAND

SECOND KMAP



MUXES

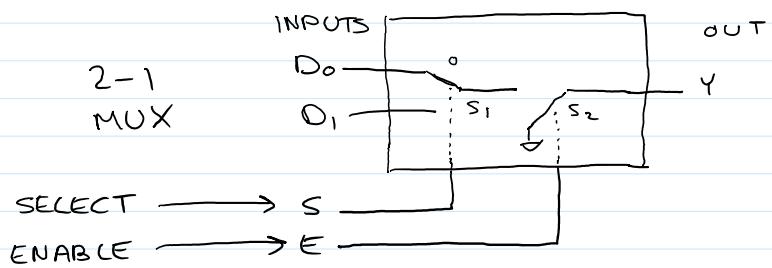
MULTIPLEXER ATTRIBUTES

- 2^n DATA INPUTS

- N CONTROL INPUTS

- 1 OUTPUT

ROUTES (OR CONNECTS) THE SELECTED DATA INPUT TO THE OUTPUT



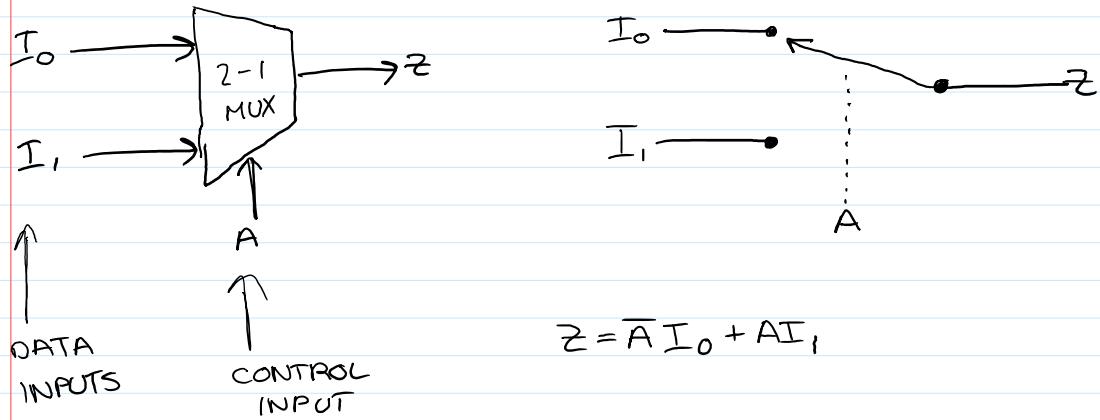
S	E	Y
0	0	O.C. OR GND
0	1	D ₀
1	0	O.C. OR GND
1	1	D ₁

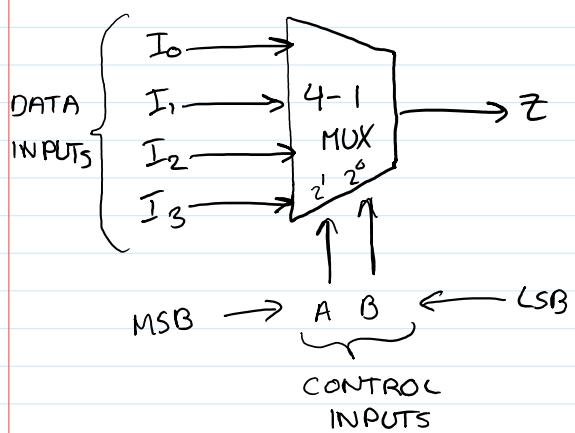
SELECT DETERMINES WHICH POSITION S₁ IS IN D₀ OR D₁

ENABLE "1" CONNECTS OUTPUT Y TO D₀ AND D₁

ENABLE "0" Y = GND "0"

SOME DEVICES WILL HAVE THE OUTPUT TRI-STATED OR OPEN CKT INSTEAD OF GND
Hi Z





A	B	Z
0	0	I ₀
0	1	I ₁
1	0	I ₂
1	1	I ₃

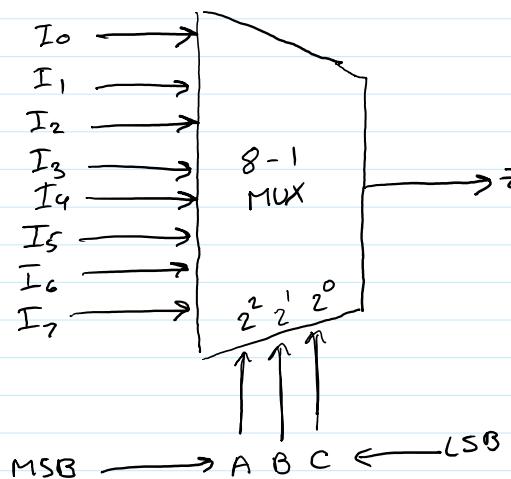
$$m_0 = \overline{A} \overline{B}$$

$$m_1 = \overline{A} B$$

$$m_2 = A \overline{B}$$

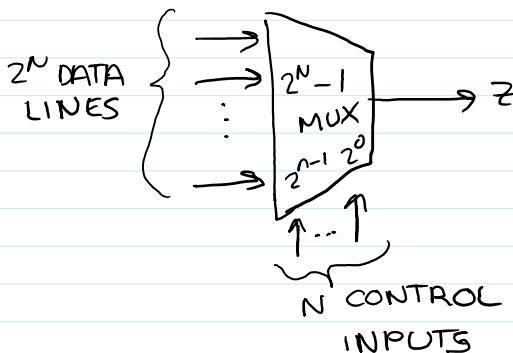
$$m_3 = A B$$

$$Z = \overline{A} \overline{B} I_0 + \overline{A} B I_1 + A \overline{B} I_2 + A B I_3$$



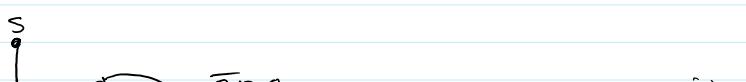
A	B	C	Z
0	0	0	I ₀
0	0	1	I ₁
0	1	0	I ₂
0	1	1	I ₃
1	0	0	I ₄
1	0	1	I ₅
1	1	0	I ₆
1	1	1	I ₇

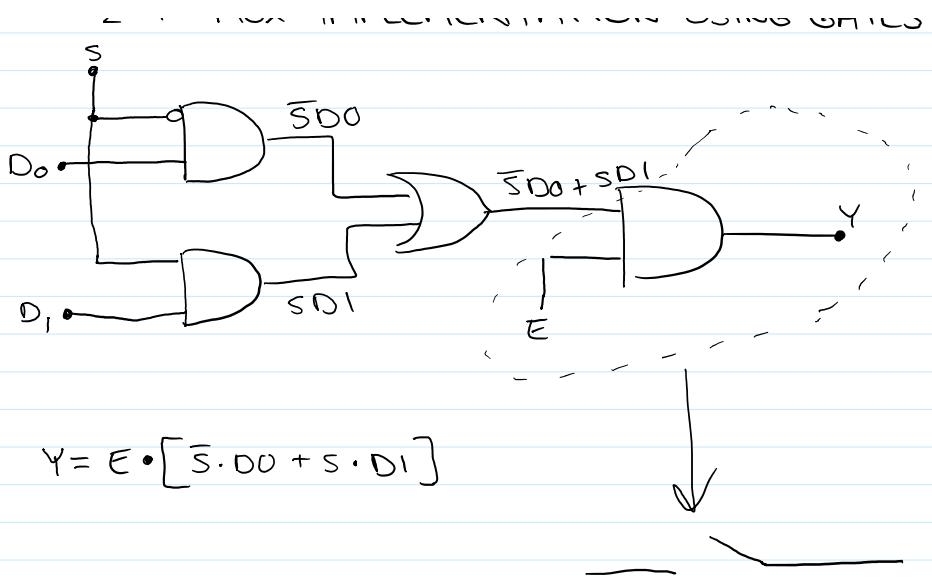
$$Z = \overline{A} \overline{B} \overline{C} I_0 + \overline{A} \overline{B} C I_1 + \overline{A} B \overline{C} I_2 + \overline{A} B C I_3 + A \overline{B} \overline{C} I_4 + A \overline{B} C I_5 + A B \overline{C} I_6 + A B C I_7$$



$$Z = \sum m_i \cdot I_i$$

2-1 MUX IMPLEMENTATION USING GATES





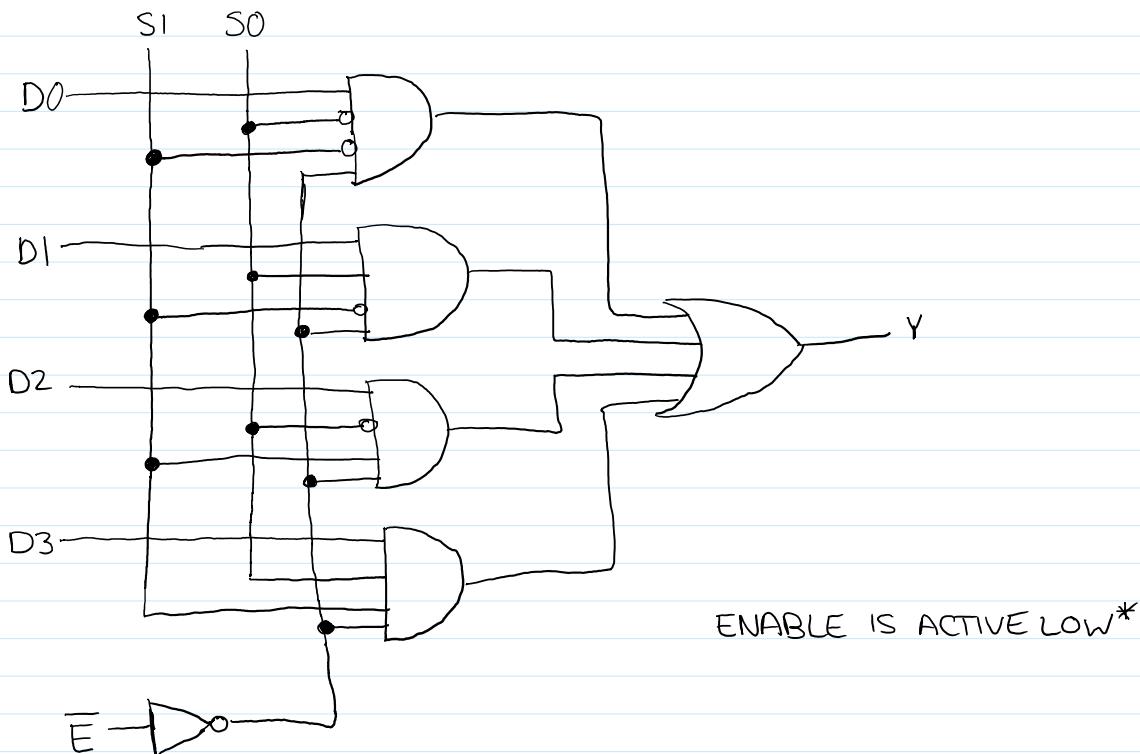
TRI - STATE BUFFER

$E=1 \Rightarrow \text{SWITCH CLOSED}$

4 LINE TO 1 LINE MUX

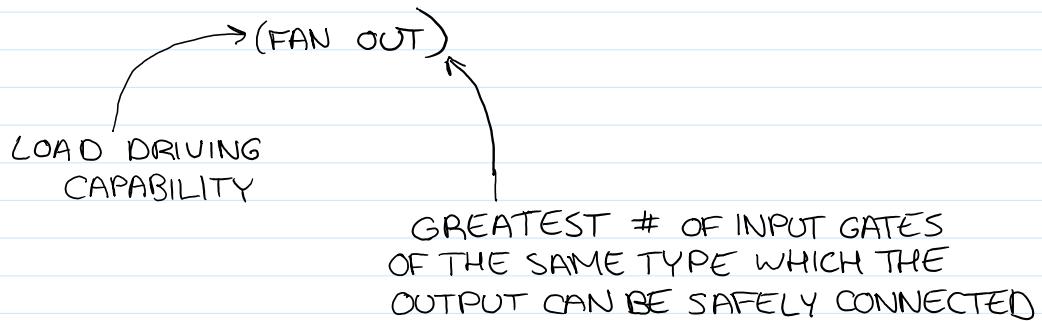
2 SELECT BITS

4 DATA INPUT LINES

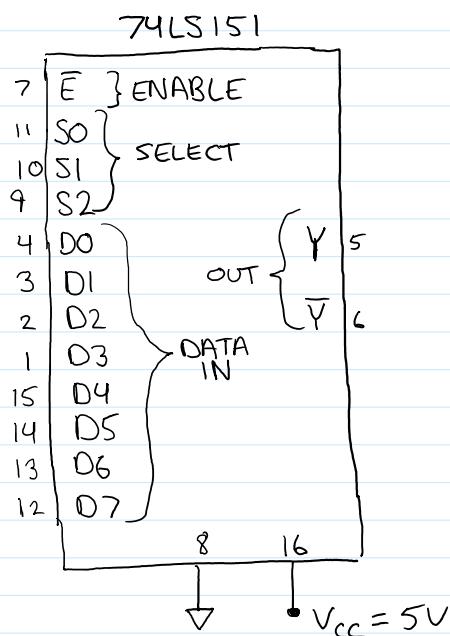


$$Y = \bar{E} \cdot [\bar{S_0} \cdot \bar{S_1} \cdot D_0 + S_0 \cdot \bar{S_1} \cdot D_1 + \bar{S_0} \cdot S_1 \cdot D_2 + S_0 \cdot S_1 \cdot D_3]$$

* A GATE CAN DRIVE MORE GATES TO LOGIC LOW THAN LOGIC HIGH



8 INPUT 1 BIT MULTIPLEXER



TRUTH TABLE

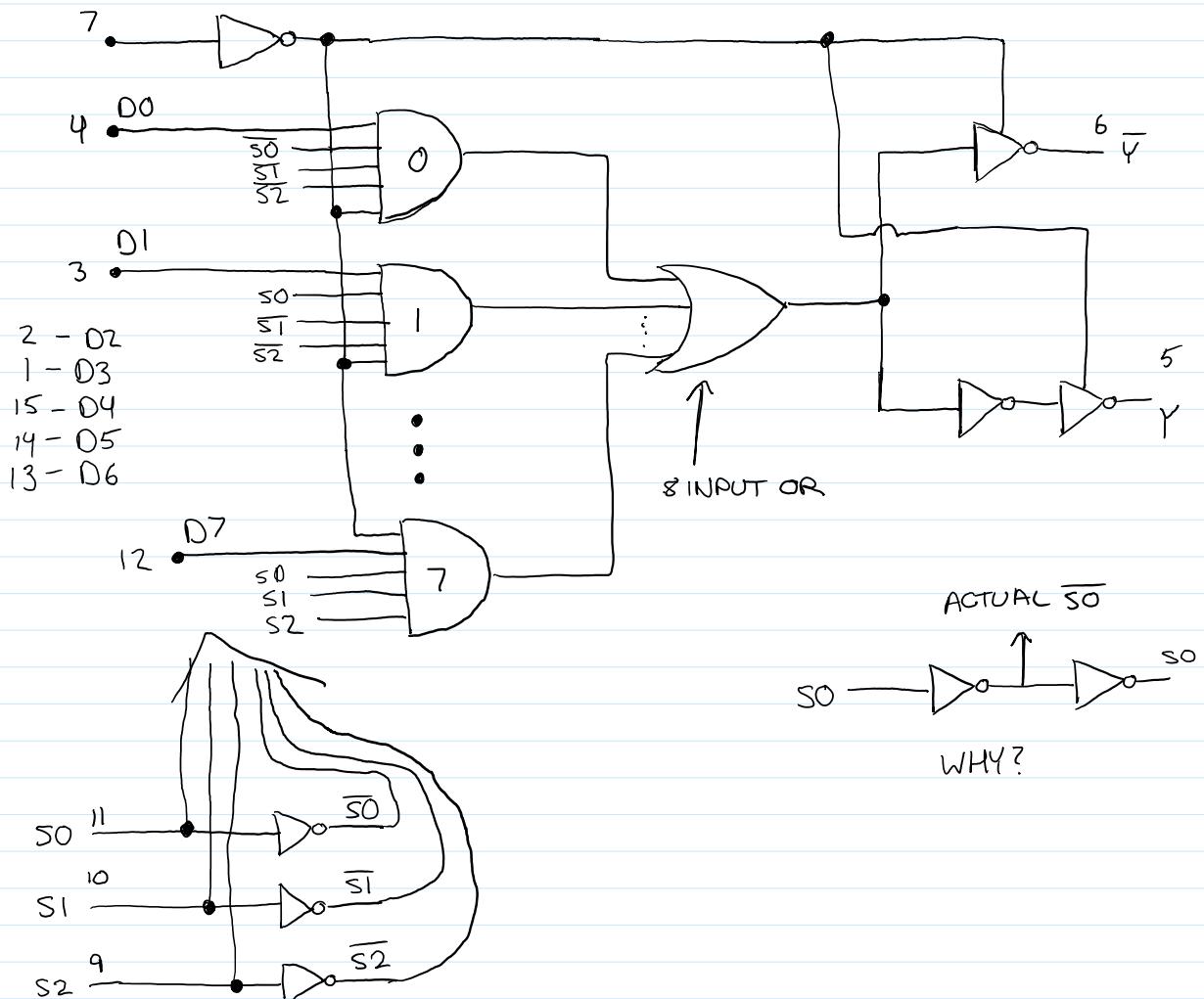
E	S_2	S_1	S_0	Y	\bar{Y}
1	-	-	-	O.C.	O.C.
0	0	0	0	D0	
0	0	0	1	D1	
0	0	1	0	D2	
0	0	1	1	D3	
0	1	0	0	D4	
0	1	0	1	D5	
0	1	1	0	D6	
0	1	1	1	D7	

ON ↑
↓ OFF

THREE SELECT INPUTS DETERMINE WHICH OF 8 DATA LINES (INPUT) IS CONNECTED TO THE OUTPUT Y .

THE DEVICE ALSO HAS A COMPLIMENTED OUTPUT. THE TWO OUTPUTS CAN BE TRI-STATED THROUGH E

74151 LOGIC DIAGRAM



ENCODERS

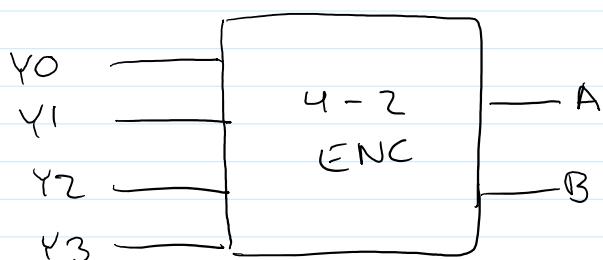
ATTRIBUTES

- 2^N INPUTS

OUTPUTS THE BINARY VALUE OF SELECTED OR ACTIVE INPUT

- N OUTPUTS

PERFORMS INVERSE OPERATION OF DECODER



<u>Y_0</u>	<u>Y_1</u>	<u>Y_2</u>	<u>Y_3</u>	<u>A</u>	<u>B</u>
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

PRIORITY ENCODER

AN ENCODER THAT SELECTS OUTPUT BASED ON ONLY ONE INPUT LINE AT A TIME

A PRIORITY ENCODER WILL OUTPUT DATA FOR THE HIGHEST PRIORITY INPUT

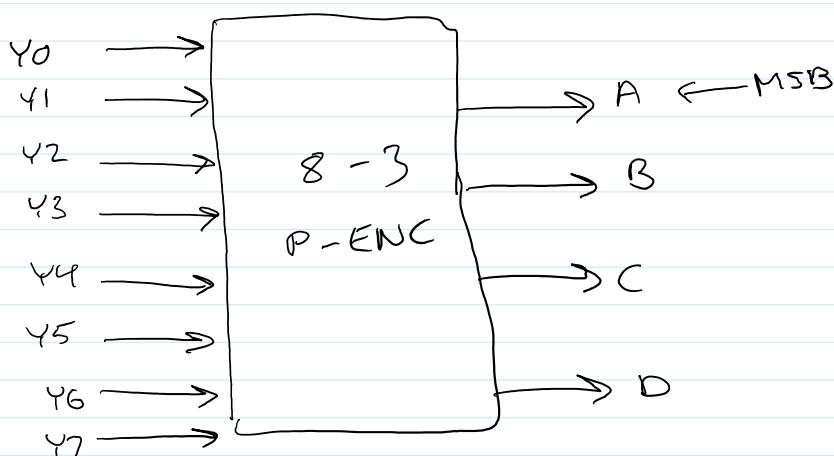
IF MORE THAN ONE INPUT IS ACTIVE, THE HIGHER-ORDER INPUT HAS PRIORITY OVER THE LOWER ORDER INPUT

HIGHER VALUE WILL BE ENCODED ON OUTPUT

A VALID INDICATOR D IS INCLUDED TO INDICATE WHETHER OR NOT OUTPUT IS VALID

D=0 OUTPUT INVALID WHEN NO INPUTS ARE ACTIVE

D=1 OUTPUT VALID WHEN AT LEAST 1 INPUT ACTIVE



<u>Y_0</u>	<u>Y_1</u>	<u>Y_2</u>	<u>Y_3</u>	<u>Y_4</u>	<u>Y_5</u>	<u>Y_6</u>	<u>Y_7</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1

Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
X	-	0	0	0	0	0	0	0	0	0	0	0	0	1	1
X	X	-	0	0	0	0	0	0	0	0	0	0	0	0	1
X	X	X	-	0	0	0	0	0	0	0	0	0	0	0	1
X	X	X	X	-	0	0	0	0	0	0	0	0	0	0	1
X	X	X	X	X	-	0	0	0	0	0	0	0	0	0	1
X	X	X	X	X	X	-	0	0	0	0	0	0	0	0	1
X	X	X	X	X	X	X	-	0	0	0	0	0	0	0	1

$$A = \sum(4, 5, 6, 7) = Y_4 + Y_5 + Y_6 + Y_7$$

$$\begin{aligned} B &= \sum(2, 3, 6, 7) = \bar{Y}_7 \bar{Y}_6 \bar{Y}_5 \bar{Y}_4 \bar{Y}_3 Y_2 + \bar{Y}_7 \bar{Y}_6 \bar{Y}_5 \bar{Y}_4 Y_3 + \bar{Y}_7 Y_6 + Y_7 \\ &= \bar{Y}_5 \cdot \bar{Y}_4 Y_2 + Y_5 Y_4 Y_3 + Y_6 + Y_7 \\ &= \bar{Y}_5 \bar{Y}_4 (Y_2 + Y_3) + Y_6 + Y_7 \end{aligned}$$

$$\begin{aligned} C &= \sum(1, 3, 5, 7) = \bar{Y}_7 \bar{Y}_6 \bar{Y}_5 \bar{Y}_4 \bar{Y}_3 \bar{Y}_2 Y_1 + \bar{Y}_7 \bar{Y}_6 \bar{Y}_5 \bar{Y}_4 Y_3 + \bar{Y}_7 \bar{Y}_6 Y_5 + Y_7 \\ &= \bar{Y}_6 \bar{Y}_4 \bar{Y}_2 Y_1 + \bar{Y}_6 \bar{Y}_4 Y_3 + \bar{Y}_6 Y_5 + Y_7 \\ &= \bar{Y}_6 (\bar{Y}_4 \bar{Y}_2 Y_1 + \bar{Y}_4 Y_3 + Y_5) + Y_7 \end{aligned}$$

DECODER ATTRIBUTES

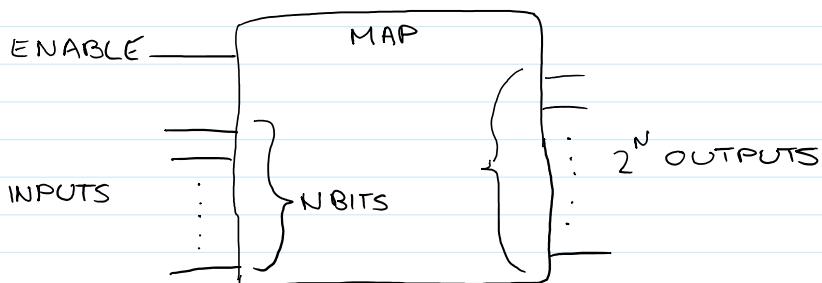
- N INPUTS

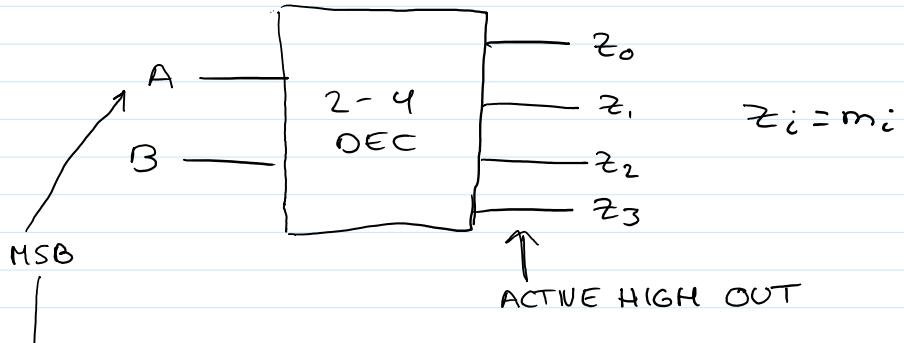
- 2^N OUTPUTS

SELECTS ONE OF 2^N OUTPUTS BY DECODING THE BINARY VALUE OF N INPUTS

DECODER GENERATES ALL OF THE MINTERMS OF THE N INPUT VARIABLES

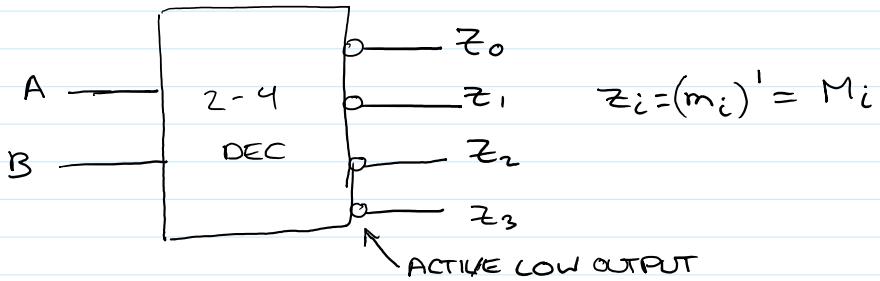
EXACTLY ONE OUT WILL BE ACTIVE FOR EACH COMBINATION OF THE INPUTS



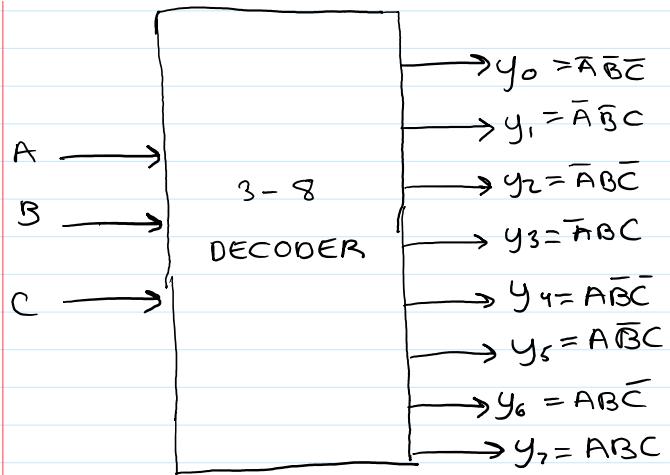


2 INPUT BITS SELECT ONE OF FOUR LINES TO HAVE A
"1" OUT OTHER LINES = 0

A	B	Z_0	Z_1	Z_2	Z_3	
0	0	1	0	0	0	m_0
0	1	0	1	0	0	m_1
1	0	0	0	1	0	m_2
1	1	0	0	0	1	m_3

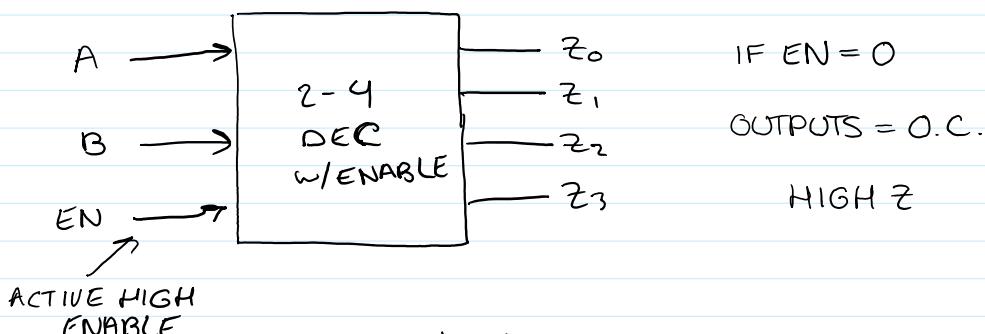


A	B	Z_0	Z_1	Z_2	Z_3	
0	0	0	1	1	1	M_0
0	1	1	0	1	1	M_1
1	0	1	1	0	1	M_2
1	1	1	1	1	0	M_3



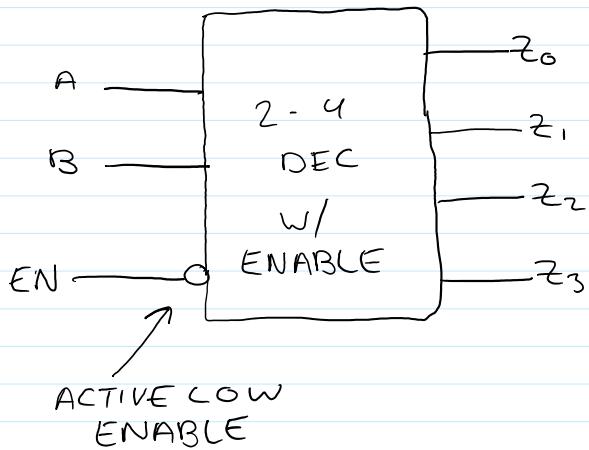
A	B	C	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

DECODER w/ ENABLE



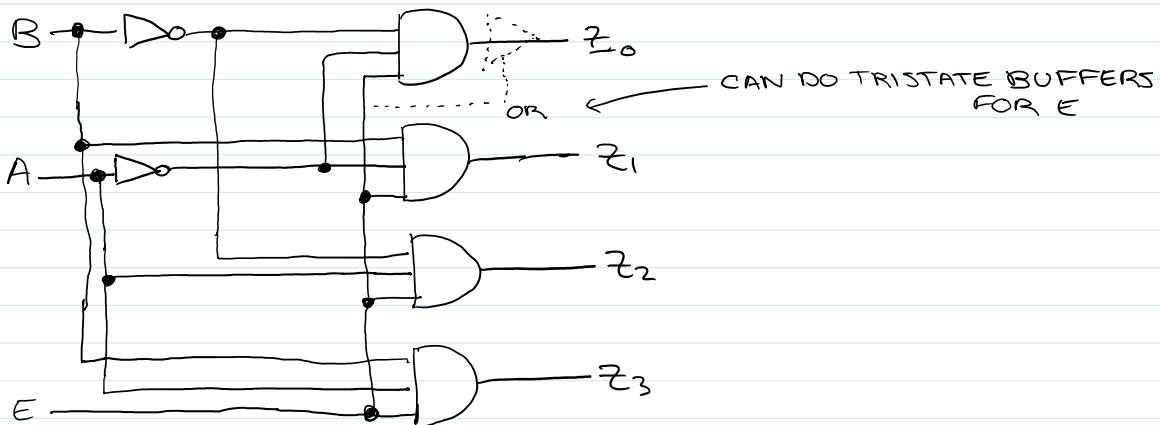
IF EN=1: 2 INPUT BITS SELECT ONE OF 4 LINES TO HAVE A "1" OUT OTHER LINES = "0"

EN	A	B	Z_0	Z_1	Z_2	Z_3
ENABLED {	1	0	0	1	0	0
	1	0	1	0	1	0
	1	1	0	0	1	0
	1	1	1	0	0	1
DISABLED { 0	x	x	0	0	0	0



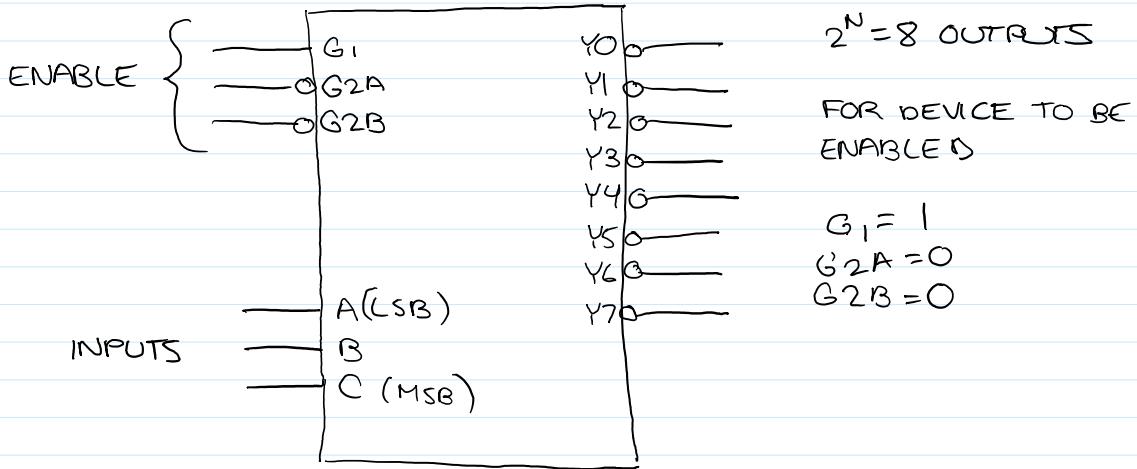
EN	A	B	Z ₀	Z ₁	Z ₂	Z ₃
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1
1	X	X	0	0	0	0

LOGIC CKT DIAGRAM OF 2-4 DECODER w/ ACTIVE HIGH
ENABLE



3 TO 8 DECODER 74 LS 138

N = 3



ENABLE CKT:

NOR GATE



G_1	G_{2A}	G_{2B}	E
1	0	0	1
⋮	⋮	⋮	⋮
ALL OTHERS			
⋮	⋮	⋮	⋮
			0

G_1 IS DOUBLE INVERTED SO THAT $G_1 = 1$ BECOMES THE NEEDED INPUT FOR ENABLE

$\{G_{2A}, G_{2B}\}$ IS NEGATED BY THE NOR GATE

\therefore BOTH HAVE TO BE "0" FOR THE NEEDED INPUT

OUTPUTS ARE NEGATED SINCE NAND GATES ARE FASTER

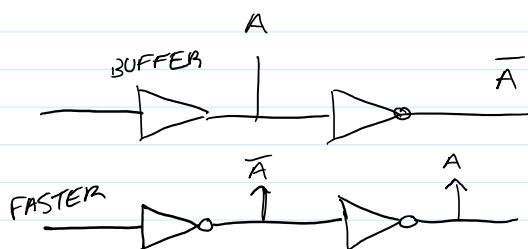
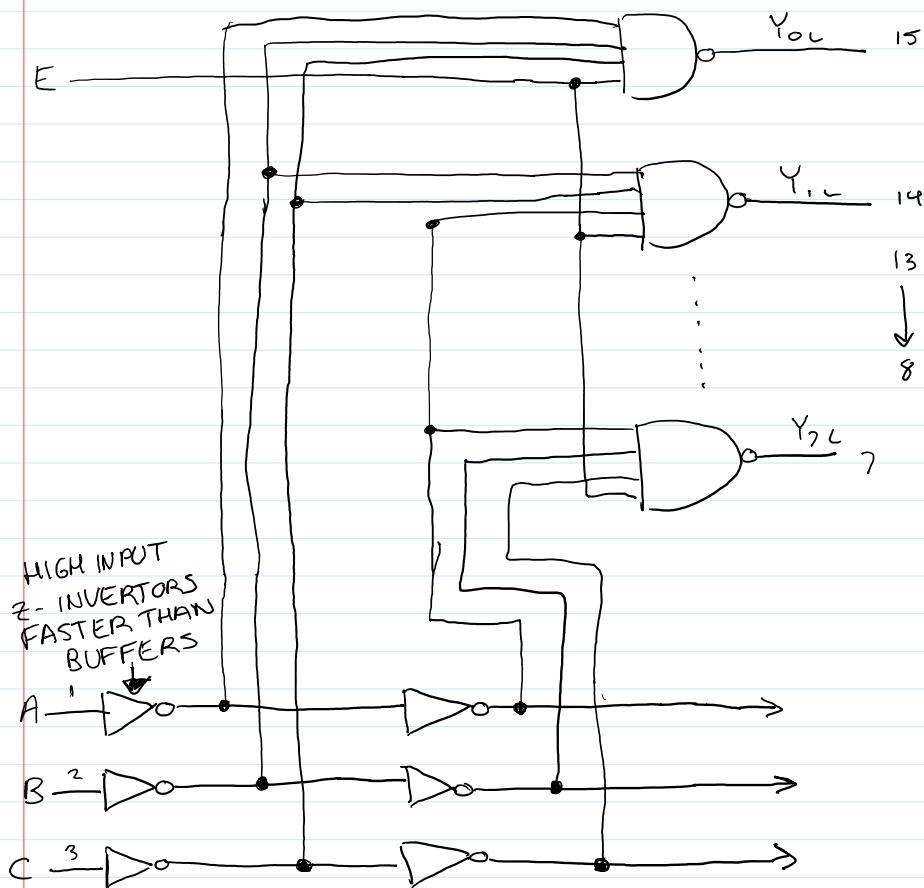
E	C	B	A	Y_7	\dots	Y_3	Y_2	Y_1	Y_0
1	0	0	0	1		1	1	1	0
.	0	0	1	1		1	1	0	1
.	0	1	0	1		1	0	1	1
.	0	1	1	1		0	1	1	1

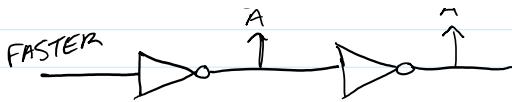
PARTIAL TRUTH TABLE

EX) $Y_6 = 0$ WHEN

$$Y_6 = \underbrace{[G_1 \cdot \overline{G_2}A \cdot \overline{G_2}B \cdot C \cdot B \cdot \overline{A}]}_E'$$

PARTIAL CKT 74LS138





EX) ENCODER USING 3 MUX'S TO MAKE A BINARY TO GRAY CODE CONVERTOR

BINARY . \rightarrow MUX \rightarrow GRAY

DATA	B ₂	B ₁	B ₀	G ₂	G ₁	G ₀	DECIMAL OF GRAY
D0	0	0	0	0	0	0	0
D1	0	0	1	0	0	1	1
D2	0	1	0	0	1	1	3
D3	0	1	1	0	1	0	2
D4	1	0	0	1	1	0	6
D5	1	0	1	1	1	1	7
D6	1	1	0	1	0	1	5
D7	1	1	1	1	0	0	4

TASK: CONVERT BINARY TO GRAY CODE USING MUX'S AND BACK TO BINARY USING XOR

B's \Rightarrow INPUTS (SELECT)
G's \Rightarrow OUTPUTS

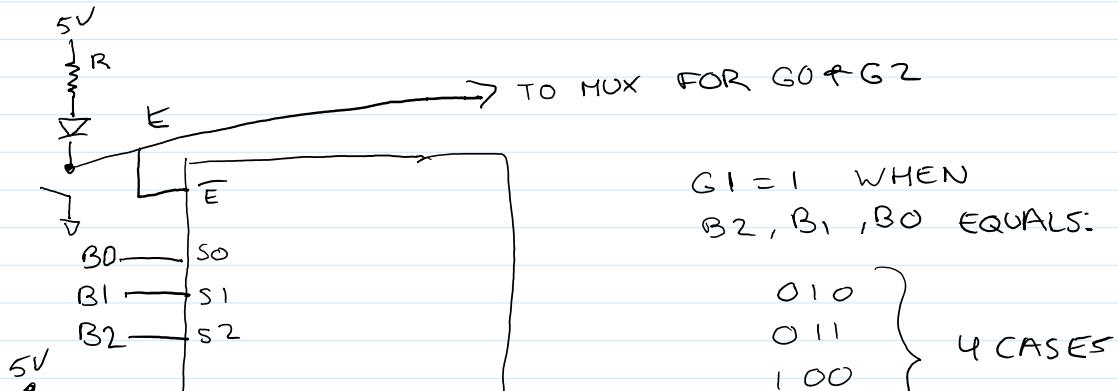
ENCODERS MAPS B₂, B₁, B₀ TO THE G'S

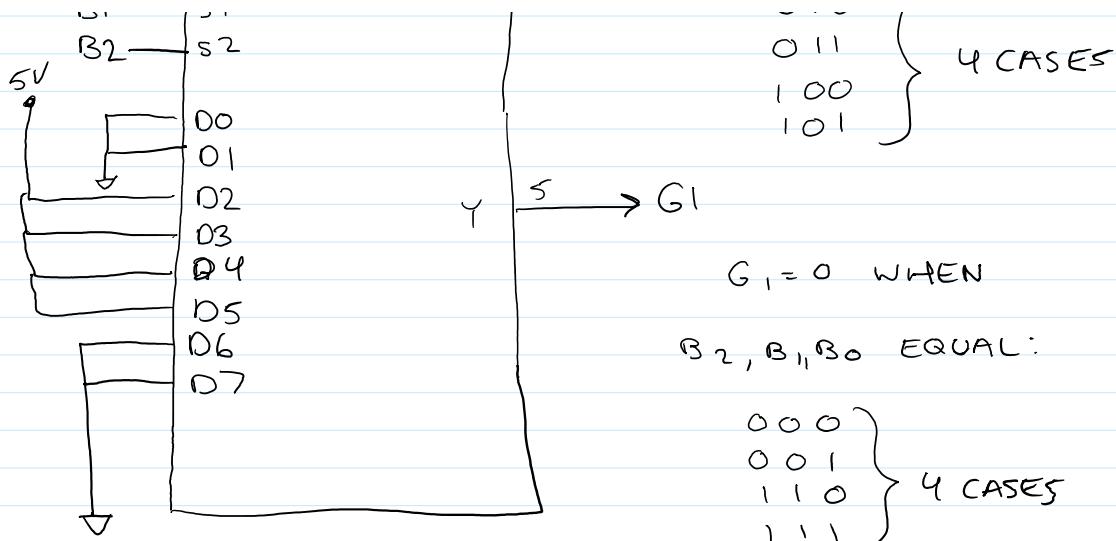
G₂ = 1 : WHENEVER D₄, D₅, D₆, D₇ IS SELECTED AND D₄-D₇=1

G₁ = 1 D₂, D₃, D₄, D₅

G₀ = 1 D₁, D₂, D₅, D₆

G1 USING THE 74LS151





DATA INPUTS D_2, D_3, D_4, D_5 GO TO $5V$

DATA INPUTS D_0, D_1, D_6, D_7 GO TO GND

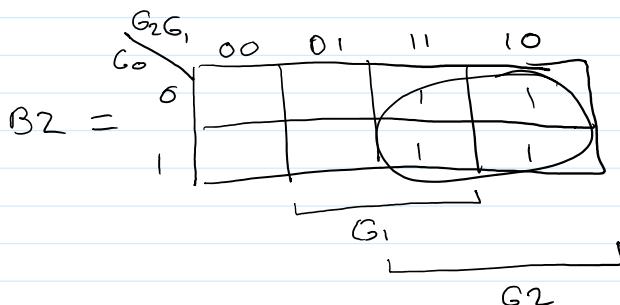
$G_2 \Rightarrow D_4, D_5, D_6, D_7 = 5V$ OTHER = GND

$G_0 \Rightarrow D_1, D_2, D_5, D_6 = 5V$ OTHER = GND

DECODER USING XOR GATES

OBTAIN B 'S FROM G 'S G = INPUT
 B = OUTPUT

$$B_2 = \sum_{G_2 G_1 G_0} (4, 5, 6, 7)$$



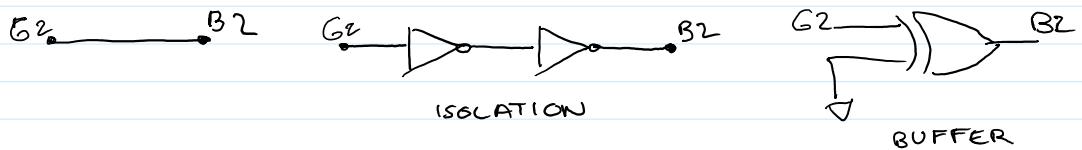
RECALL:

$$B_2 = 1$$

110	111	4 CASES
101	100	

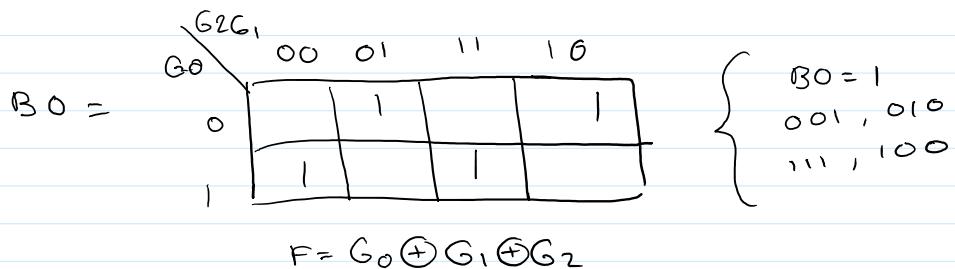
$$B_2 = G_2 \text{ (NO GATES)}$$





$$B_0 = \sum(1, 2, 4, 7)$$

K MAP FROM T.T. FOR B_0



PROOF

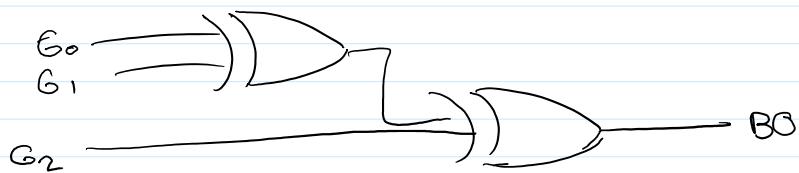
$$F = \overline{G_2} \overline{G_1} G_0 + \overline{G_2} G_1 \overline{G_0} + G_2 G_1 G_0 + G_2 \overline{G_1} \overline{G_0}$$

$$= \overline{G_2} (\overline{G_1} G_0 + G_1 \overline{G_0}) + G_2 (G_1 G_0 + \overline{G_1} \overline{G_0})$$

$$= \overline{G_2} (G_1 \oplus G_0) + G_2 (G_1 \odot G_0)$$

$$\text{LET } X = G_1 \oplus G_0 \quad \overline{X} = G_1 \odot G_0$$

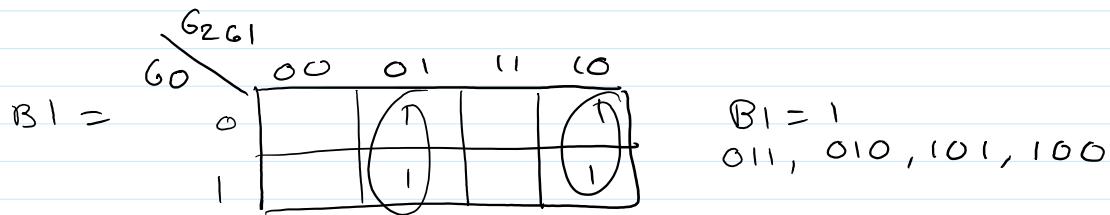
$$F = \overline{G_2} X + G_2 \overline{X} = G_2 \oplus X = G_2 \oplus G_1 \oplus G_0$$



B_0 REQUIRES 2 XOR GATES

NOTE: B_1 WILL REQUIRE JUST ONE XOR

$$B_1 = \sum(2, 3, 4, 5)$$



$$B_1 = 1 \\ 011, 010, 101, 100$$

$$B_1 = \overline{G_2} G_1 + G_2 \overline{G_1} = G_2 \oplus G_1$$

CKT DESIGN USING MUX

N = # OF CONTROL = # OF VARS IN $f(x)$

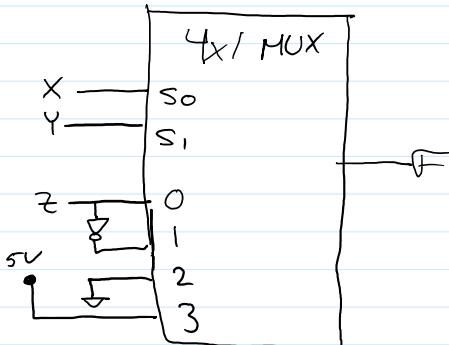
EACH MINTERM MAPPED TO A DATA INPUT OF MUX

EACH ROW OF TRUTH TABLE WHERE OUTPUT = 1 SET CORRESPONDING DATA INPUT OF MUX TO 1

SET REMAINING TO 0

EX $F = \sum_{xyz} (1, 2, 6, 7)$

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

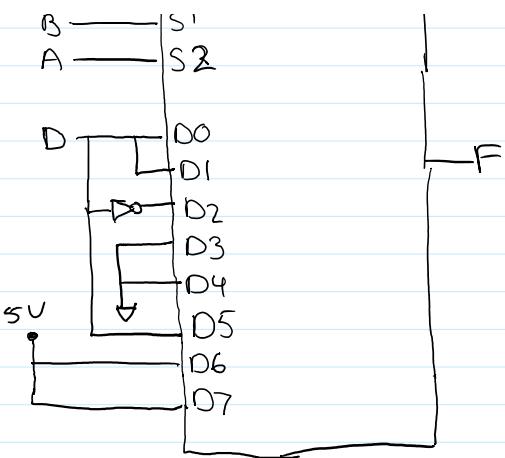


EX $F = \sum_{ABCD} (1, 3, 4, 11, 12 - 15)$

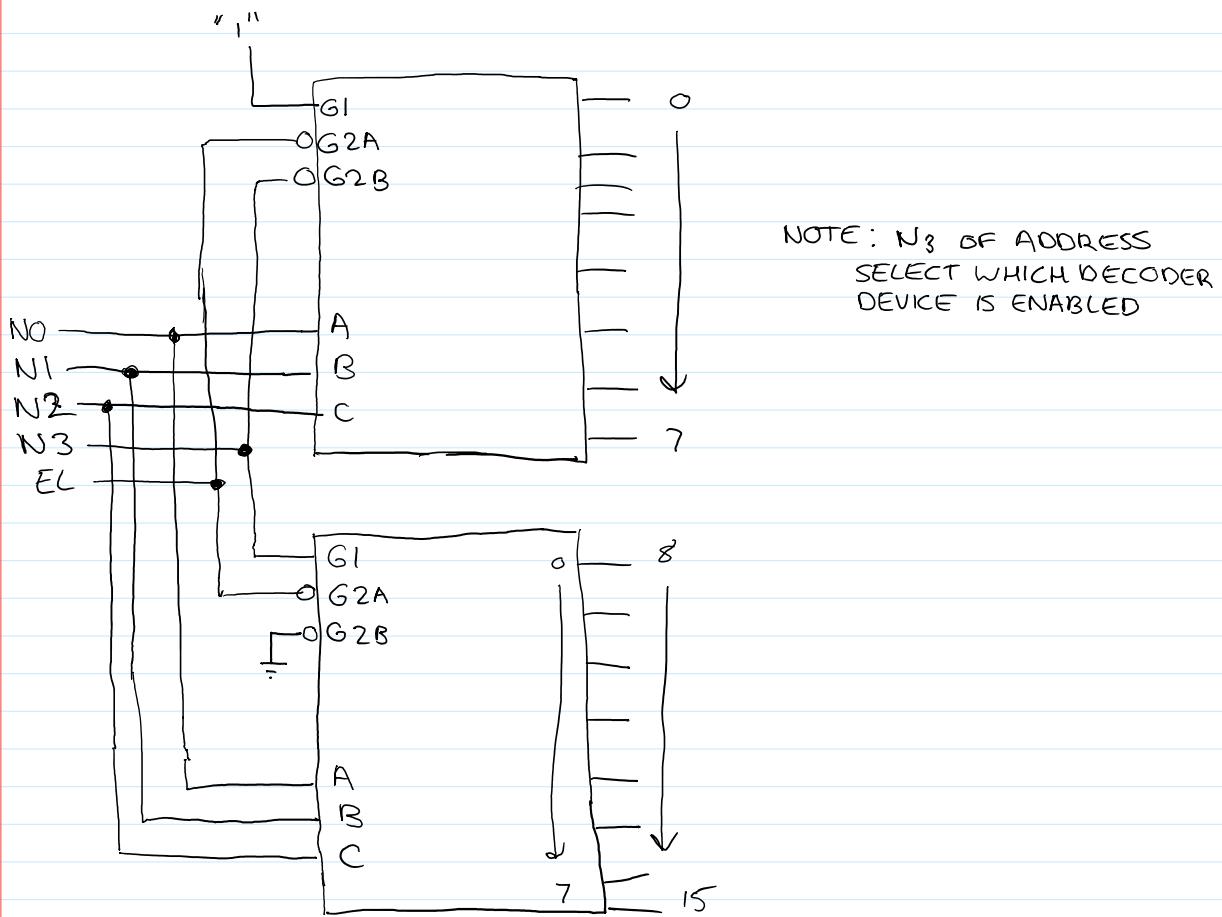
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1



0	0	1	1	1	
0	1	0	0	1	$F = \bar{0}$
0	1	0	1	0	$F = 0$
0	1	1	0	0	$F = 0$
0	1	1	1	0	$F = 0$
1	0	0	0	0	$F = 0$
1	0	0	1	0	$F = 0$
1	0	1	0	0	$F = 0$
1	0	1	1	1	$F = 1$
1	1	0	0	1	$F = 1$
1	1	0	1	1	$F = 1$
1	1	1	0	1	$F = 1$
1	1	1	1	1	

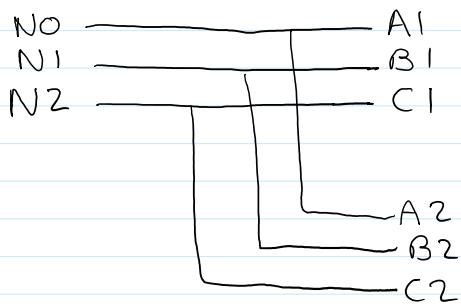


4 TO 16 DECODER USING 2 138's



EL WILL ENABLE BOTH DEVICES TO OUTPUT EXCEPT N3
WILL NOT PUT ONE DOWN

INPUTS N₀, N₁, N₂ ARE CONNECTED TO A, B, C



THESE THREE INPUTS WILL SELECT THE SAME OUTPUT ON BOTH 138's TOGETHER

HOWEVER, N3 IS CONNECTED TO G2B[1] (NEGATED ENABLE)
AND G1[2]

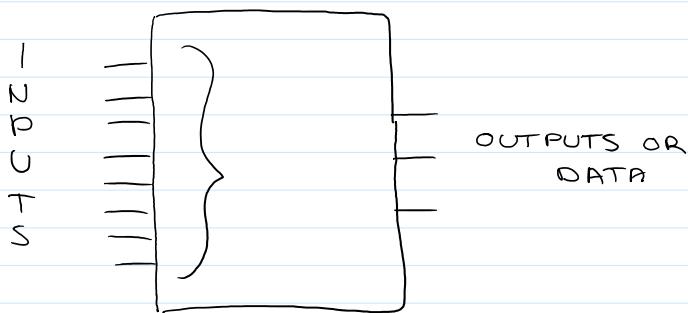
∴ N3 WILL SELECT WHETHER GROUP 0-7 OR 8-15 WILL BE ACTIVE

ENCODERS ARE "INVERSE" FCN OF DECODER

N INPUTS \Rightarrow M OUTPUTS

EACH INPUT LINE WILL SELECT A SPECIFIC ENCODED WORD WILL APPEAR ON THE OUTPUTS

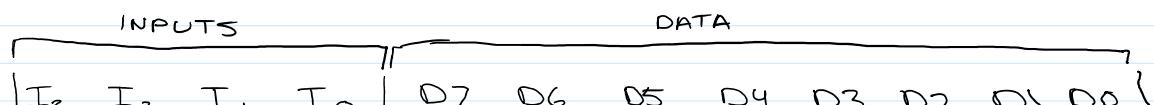
ROM'S AND RAM'S ARE ENCODERS
 ↘ READ ONLY ↘ RANDOM ACCESS



EX ROM

THAT HAS 4 LINES IN
[ADDRESS LOCATIONS]

8 LINES OUT
(DATA)



INPUTS				DATA							
I ₃	I ₂	I ₁	I ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	1	1	0	0	1	1
0	1	0	0	1	1	0	1	0	1	0	1
0	0	1	0	0	1	1	1	1	0	1	0
0	0	0	1	1	1	0	0	1	1	0	1

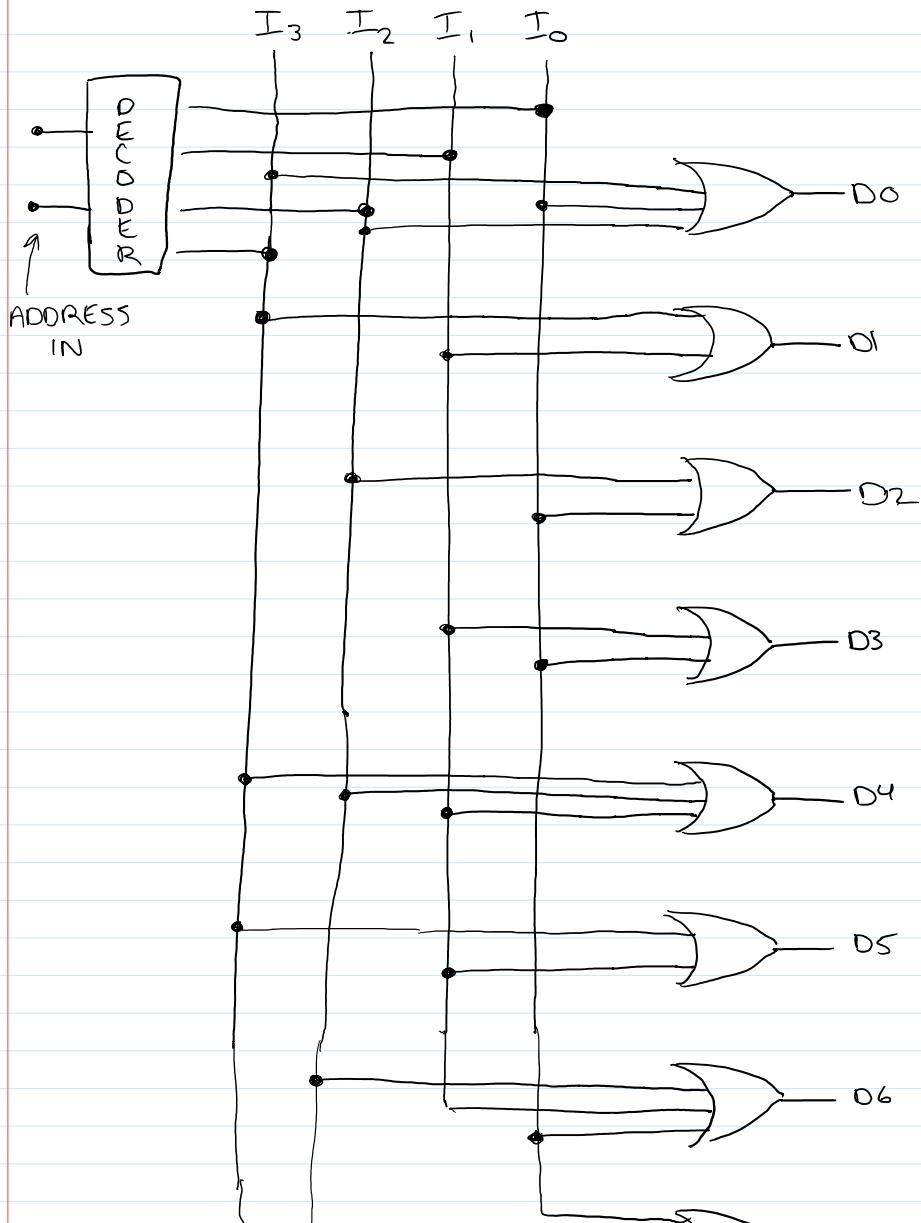
NOTE: ONLY ONE INPUT LINES CAN BE "1" AT A TIME

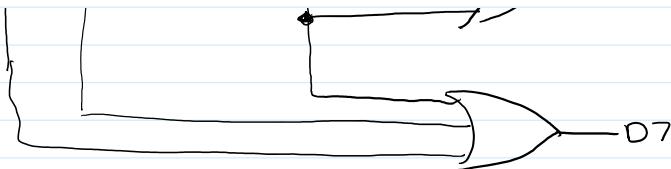
SELECT INPUT I₁ (LINE 1) OUTPUT INFO = 01111010

SELECT INPUT I₂ (LINE 2) OUTPUT INFO = 11010101

4 INPUT LINES CAN BE SELECTED USING A 2 BIT COLUMN DECODER 2 BITS IN / 4 LINES OUT

IMPLEMENTATION OF ROMS USING GATES



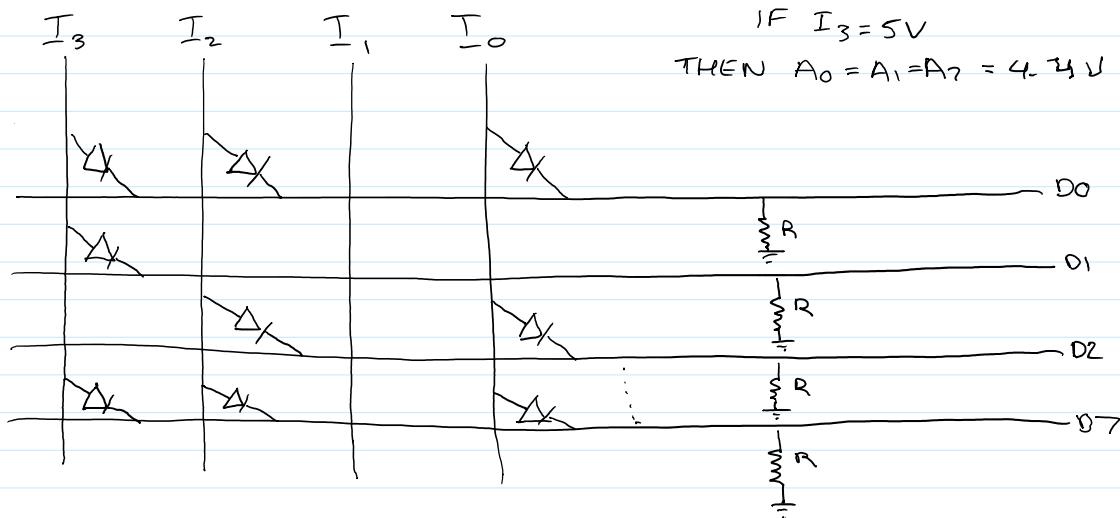


1 OF 4 BITS CAN BE SELECTED USING A 2-BIT DECODER

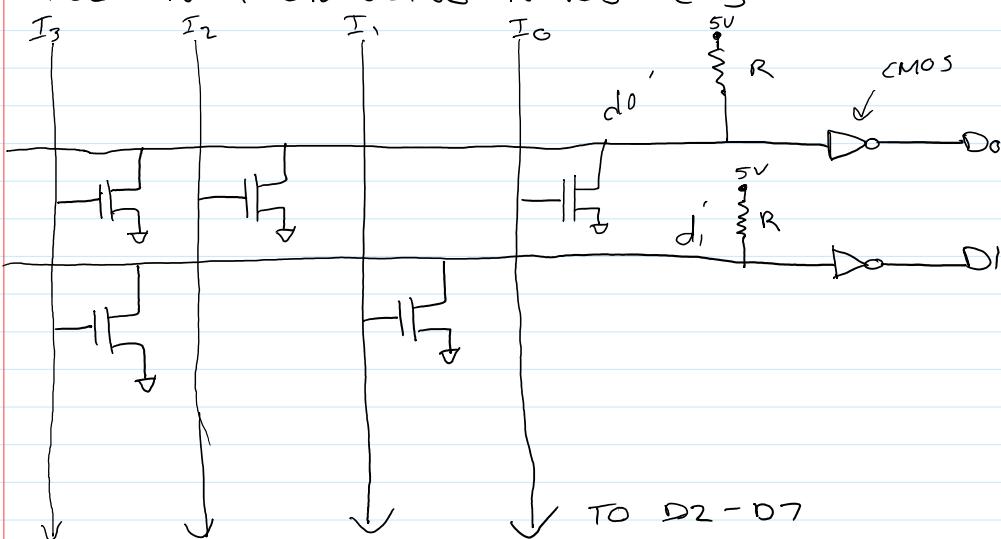
ONE GATE FOR EACH OUTPUT BUT EACH GATE COULD HAVE UP TO 4 INPUTS

IF WE HAD 32 BITS w/ 4 GIG LOCATIONS THEN WE WOULD NEED 32 GATES w/ 4 BILLION INPUTS

ROM IMPLEMENTATION w/ DIODES $V_D(F) = 0.6$



IMPLEMENTATION USING NMOS FETS

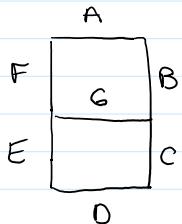


IF $I_3 = 5V \quad Q = ON \quad d_0 = d_1 = 0V \quad D0 = D1 = 5V$

$$I_3 = 0V \quad Q = OFF \quad d_o = d_1 = 5V \quad D0 = D1 = 0V$$

EXAMPLE OF ENCODER/DECODER

SEVEN SEGMENT DECODER FROM BINARY



ENCODER HAS 1 OF 10 LINES IN FOR 7 LINES OUT

INPUT LINES {

DEC	A	B	C	D	E	F	G
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

} OUTPUT LINES

SINCE THE DECIMAL DIGIT IS IN BINARY THEN WE NEED 4 BIT TO 10 LINE DECODER

INPUTS BINARY {

B ₃	B ₂	B ₁	B ₀	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7

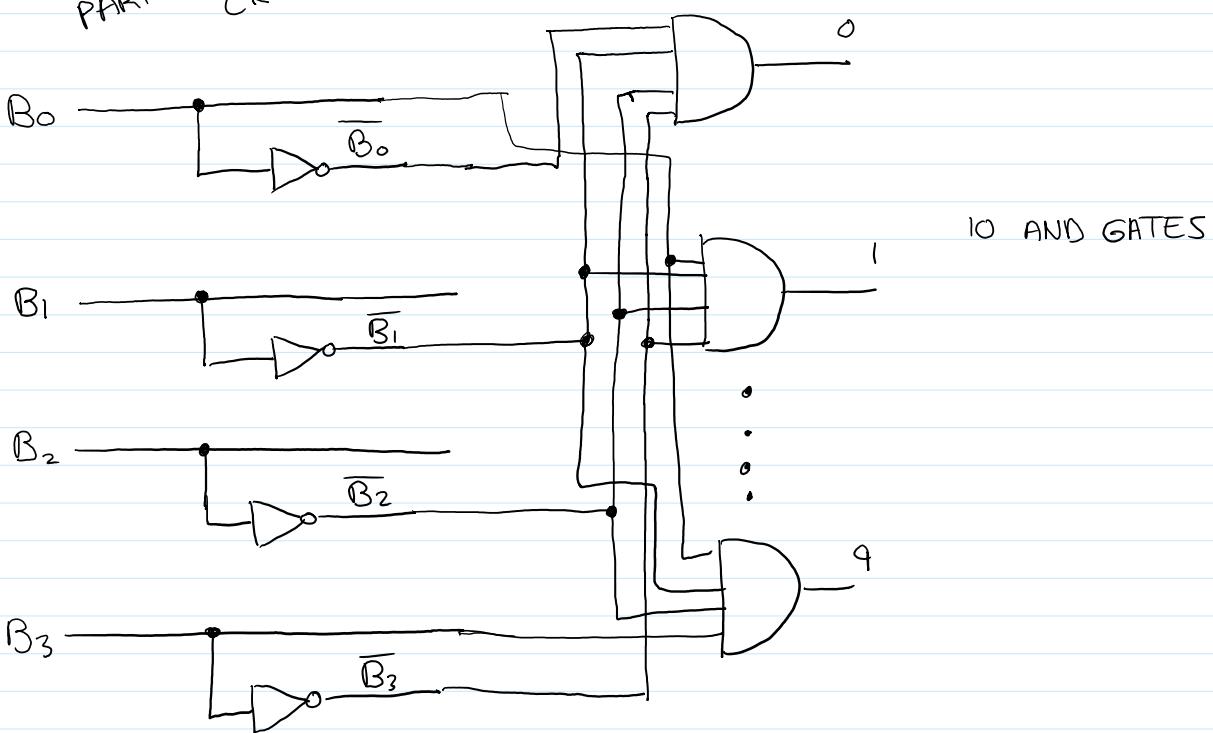
4 BINARY BITS IN
TO 1 OF 10 LINES OUT

} OUTPUTS

0	1	1	0	6
0	1	1	0	7
1	0	0	0	8
1	0	0	1	9

PARTIAL CKT

DECODER



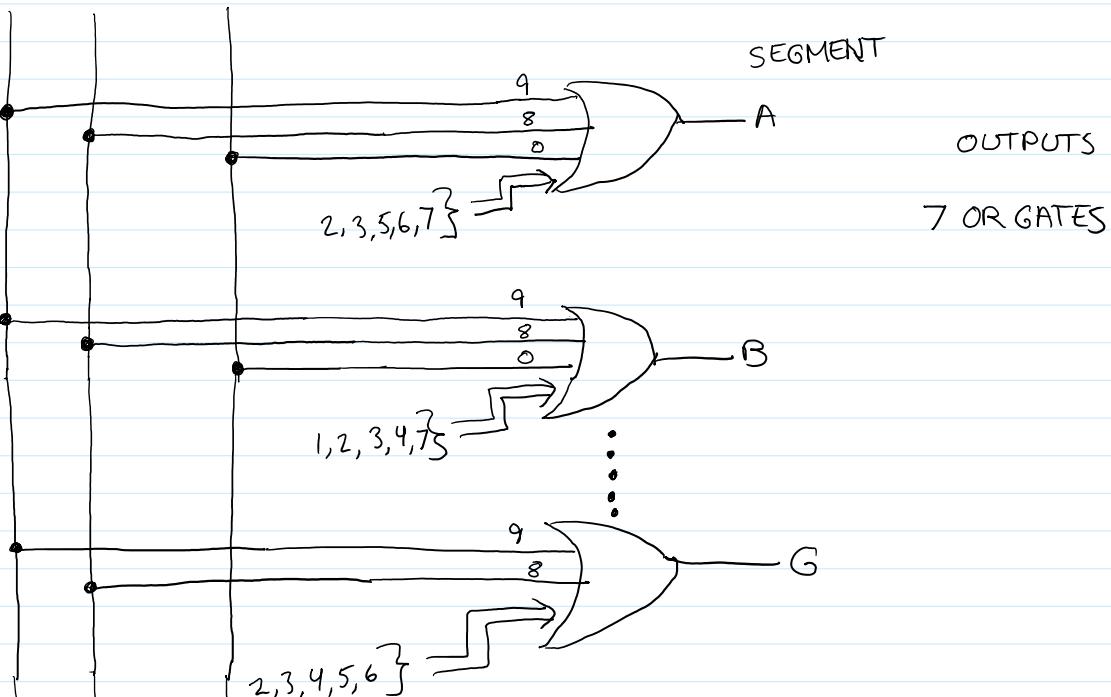
PARTIAL LOGIC DIAGRAM FOR THE 10 LINES TO 7 SEGMENT ENCODER

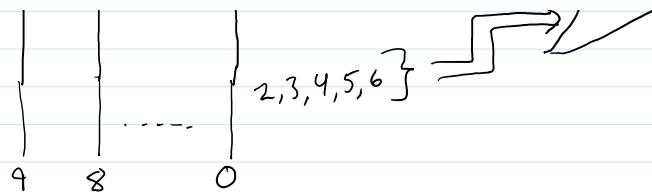
I₉ I₈ ... I₀

SEGMENT

OUTPUTS

7 OR GATES





IS PRIORITY NEEDED? NO

WHY NOT? 4 BIT TO 10 LINE USES ANDS ONLY ONE CAN BE ON

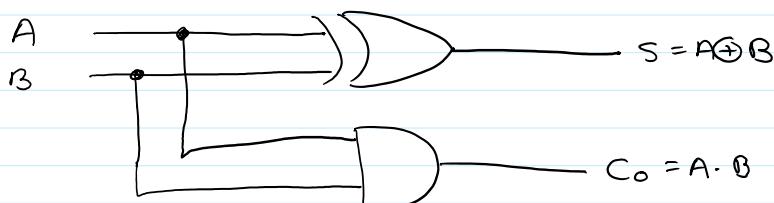
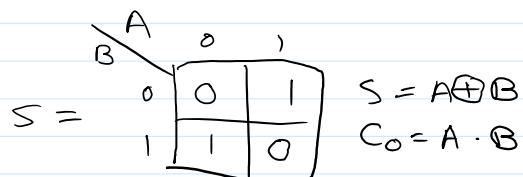
ADDER CKTS OF ARITHMETIC CKTS

HALF ADDER



A	B	S	C ₀
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

INPUTS $A + B$
 $S = \text{SUM}$ $C_0 = \text{CARRY OUT}$

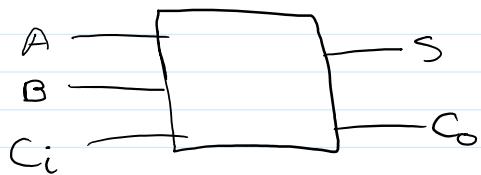


USEFUL FOR ADDING 2 BITS

BUT WHAT IS MISSING FOR FULL ADDS?

FULL ADDER ACCEPTS A CARRY AS AN INPUT

∴ 3 IN 2 OUT



C_i = CARRY INPUT

A	B	C_i	S	C_o
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
<hr/>				
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

NOTE: SUM = 1 FOR ODD # OF "1"

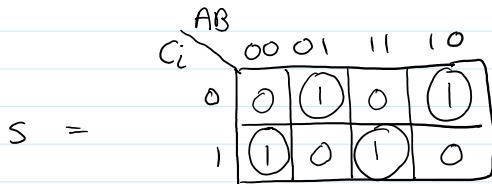
$C_o = 1$ FOR 2 OR MORE "1"

FROM TRUTH TABLE

$$S = A \oplus B \oplus C_i$$

$$C_o = A \cdot B + C_i (A \oplus B)$$

PROOF w/ KMAPS



$$S = \overline{A} \overline{B} C_i + \overline{A} B \overline{C}_i + A B C_i + A \overline{B} \overline{C}_i$$

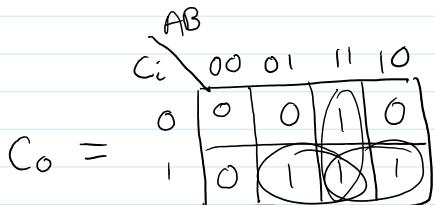
$$= C_i (\overline{A} \overline{B} + A B) + \overline{C}_i (\overline{A} B + A \overline{B})$$

$$= C_i (A \oplus B) + \overline{C}_i (A \oplus B)$$

$$\text{LET } X = A \oplus B \quad A \oplus B = \overline{(A \oplus B)}$$

$$= C_i \overline{X} + \overline{C}_i X$$

$$= C_i \oplus X = A \oplus B \oplus C_i$$



$$C_o = A \cdot B + A C_i + B C_i$$

FROM T1' WE ESTABLISHED $X \cdot 1 = X$

FROM T5 WE SEE $A + \overline{A} = 1 \quad B + \overline{B} = 1$

$$\therefore A C_i (\overline{B} + B) = A C_i \quad \text{AND} \quad B C_i (A + \overline{A}) = B C_i$$

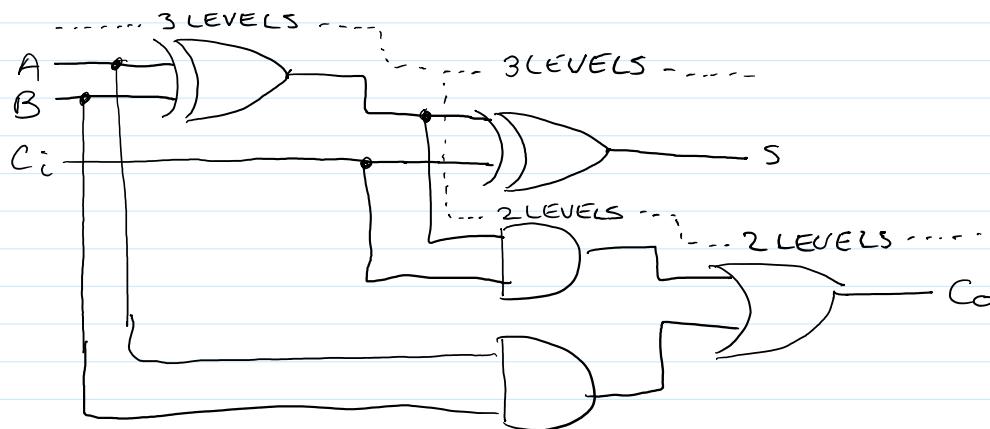
$$\therefore C_o = A \cdot B + A C_i (\overline{B} + B) + B C_i (A + \overline{A})$$

$$= A \cdot B + A B C_i + A \overline{B} C_i + A B C_i + A \overline{B} C_i$$

$$= A \cdot B + \underbrace{ABC_i + ABC_i}_{T3 = ABC_i} + C_i (\underbrace{A\bar{B} + \bar{A}B}_{A \oplus B})$$

$$= \underbrace{A \cdot B + A\bar{B}C_i}_{\text{FROM T9 } (x+x \cdot y=x)} + C_i (A \oplus B)$$

$$= A \cdot B + C_i (A \oplus B)$$



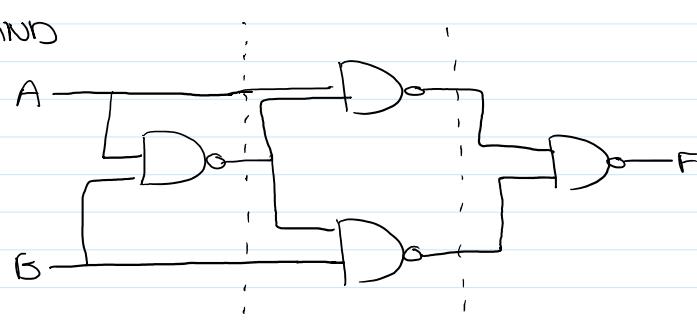
$S = 6 \text{ LEVELS}$

$C_0 = 7 \text{ LEVELS}$

REMEMBER

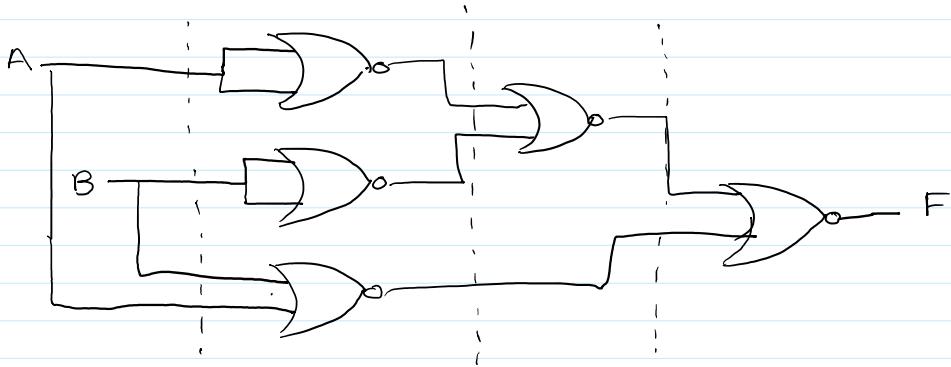
XOR

NAND / NAND



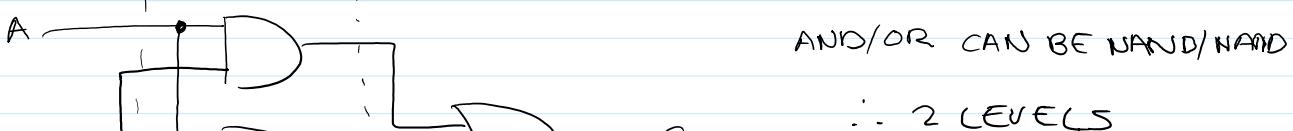
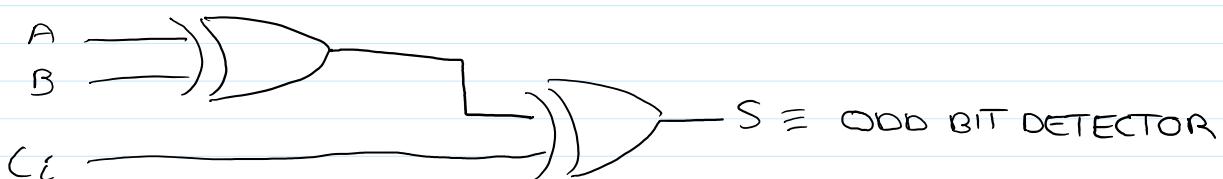
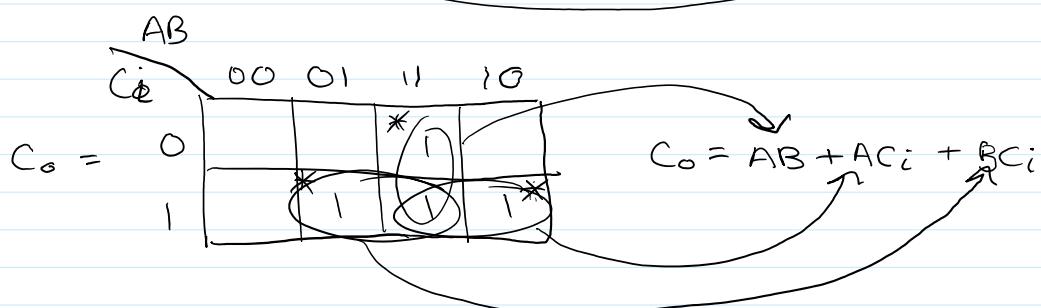
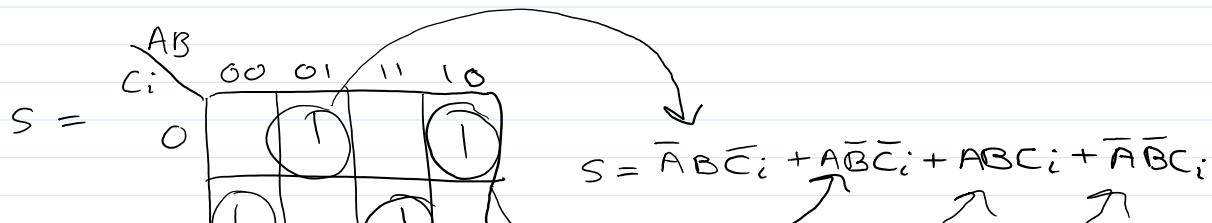
NOR / NOR





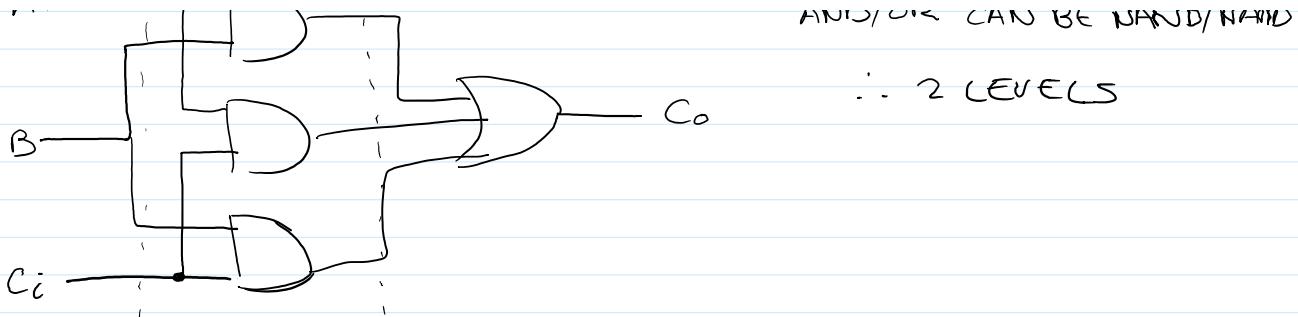
CAN WE DO BETTER?

FULL ADDER w/ KMAPS

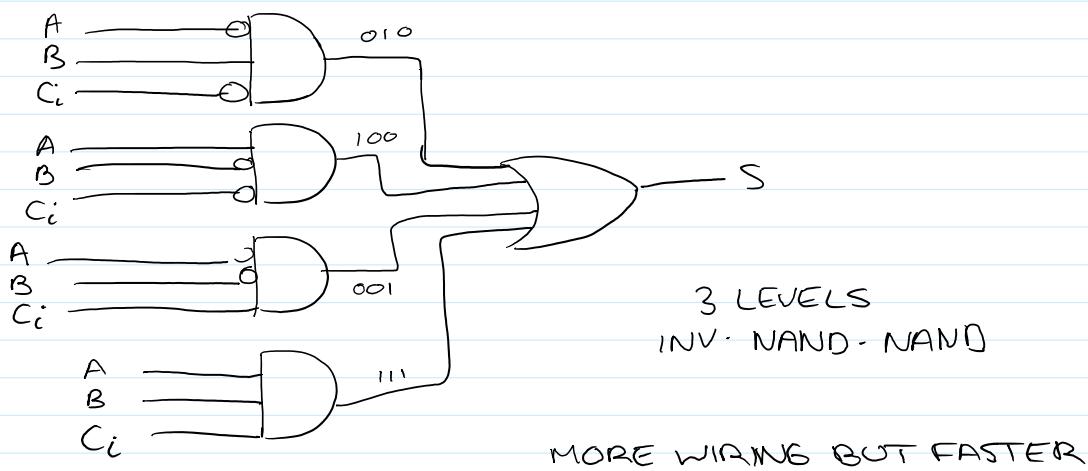


AND/OR CAN BE NAND/NAND

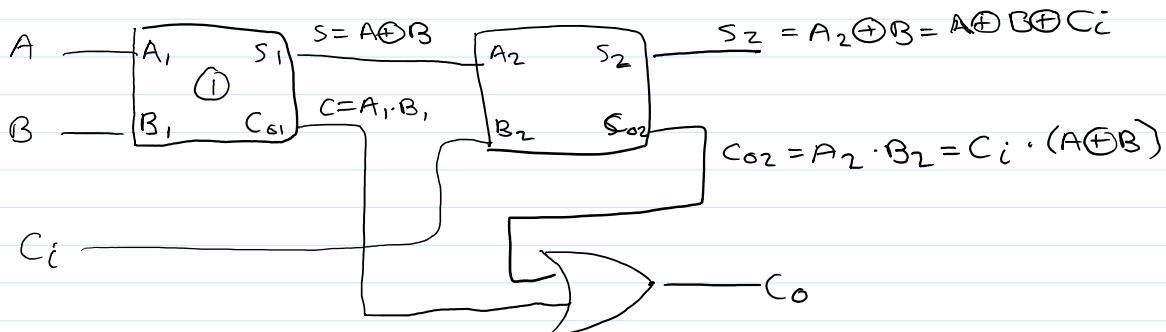
\therefore 2 LEVELS



SUM CAN BE GENERATED FASTER BY USING INV-AND-OR
INV - (NAND - NAND) RATHER THAN 2 XORS

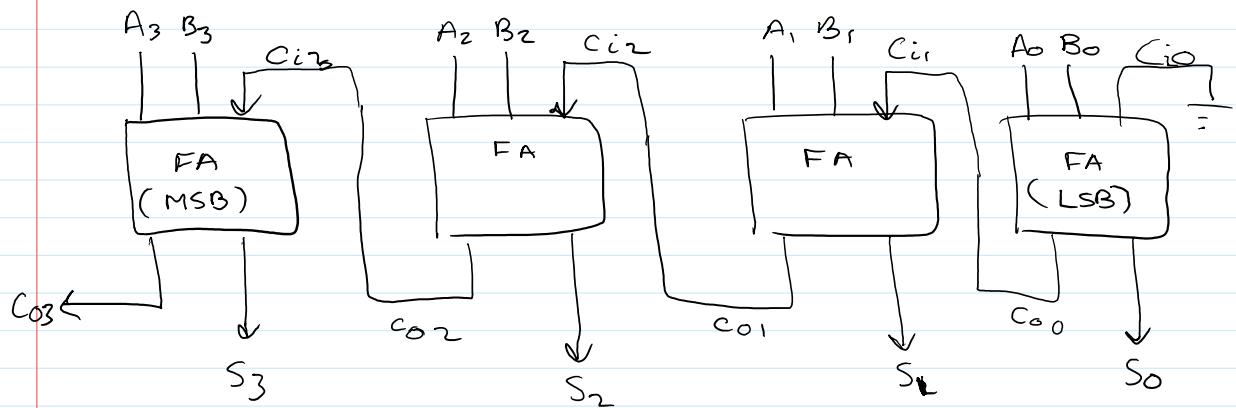


FULL ADDER FROM 2 HALF ADDER

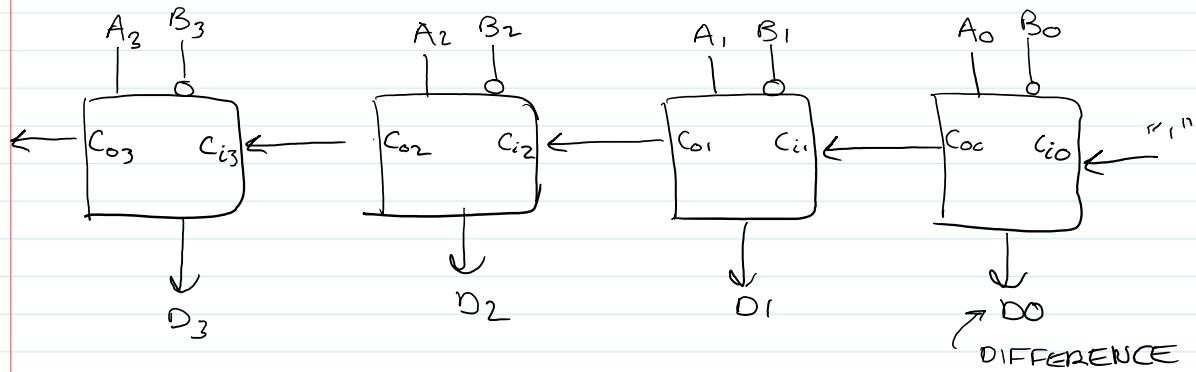


FOR ADDING WORDS WE USE :

RIPPLE FULL ADDER \Rightarrow ITERATIVE CKT

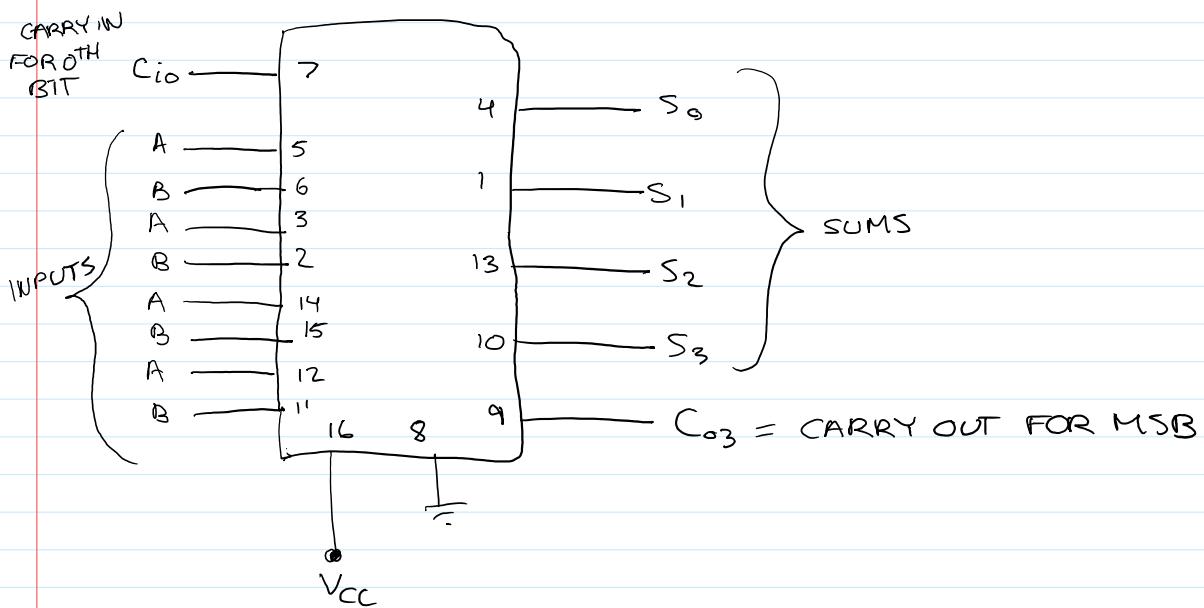


SUBTRACTION USING FULL ADDERS AND 2's COMP



INVERTING B AND THEN MAKING $C_{i0}=1$ PERFORMS 2's COMPLEMENT ON B THEN ADDING THE RESULT TO A GIVES A-B

IN LAB 4 74 LS 283 4-BIT FULL ADDER



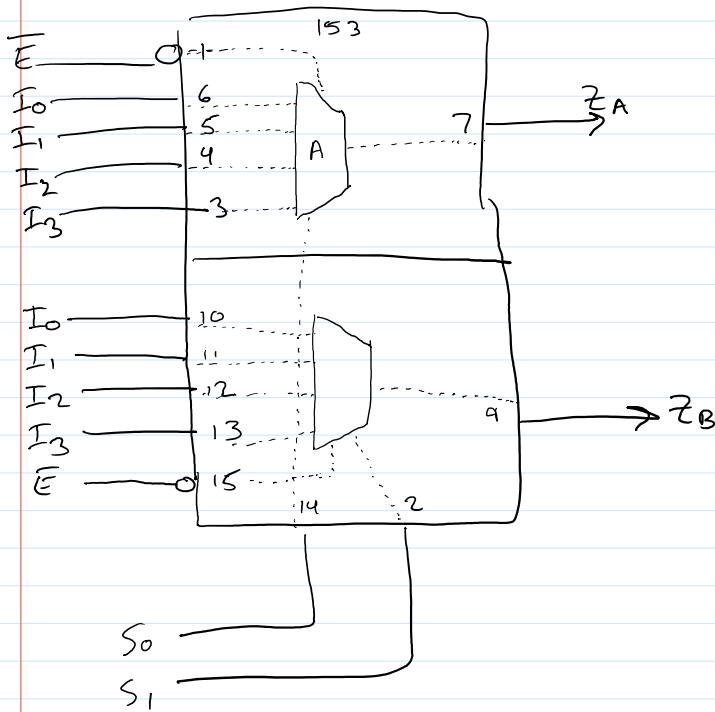
A INPUTS ARE ONE NUMBER

B INPUTS WILL COME FROM MUXS

C_{i0} INPUT COMES FROM CARRY IN GENERATOR

NEED 2 74LS153'S I.E. 4 MUX'S

74LS153 DUAL 4-1MUX w/ COMMON 2-IN SELECT w/ SEPARATE ENABLES



IF $E = "1"$ $Z = 0$

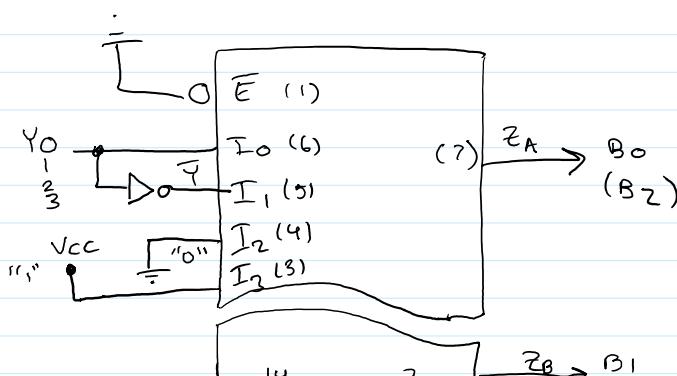
YOU TIE THE 4 E'S TO QND

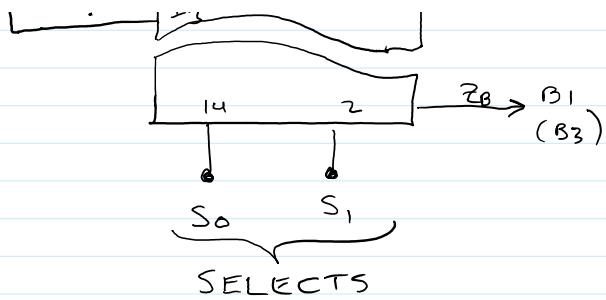
E	S_1	S_0	Z
1	—	—	0 GND
0	0	0	I_o
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3

WIRING 4 MUX'S TO ADDER

RECALL X NUMBERS WENT TO 283 ADDER

Y NUMBER GOES TO THE FOUR MUX'S FIRST





4 OPERATIONS

SELECT	INPUTS	
00	0	ADDING
01	1	SUBTRACT USING 2's COMPIMENT
10	2	ADD A "0" USED FOR INCREMENT w/ $C_{i0}=1$
11	3	ADD A "1" USED FOR DECREMENT

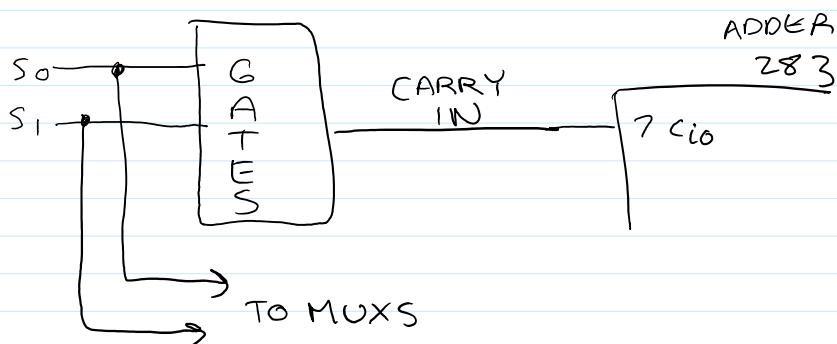
INCREMENT: SET ALL B BITS OF Z_B TO "0" BY SELECTING I_2 ON 153
THEN SET $C_{i0}(Z_B) \Rightarrow "1"$

DECREMENT: ADD -1 OR SUBTRACT A "1" USING 2's COMP
 -00001 IN 2's COMP = $1110 + 1 = 1111$ w/ $C_i=0$

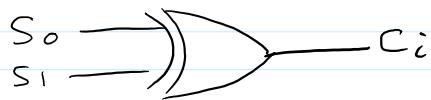
SELECT SWITCHES

S_F	S_O	
0	0	ADD
0	1	SUB
1	0	INC $Z = I_2 = "0"$
1	1	DEC $Z = I_3 = "1"$

CARRY IN GENERATOR FROM SWITCHES



	S_1	S_0	CARRY IN	
ADD	0	0	0	PLAIN ADD
SUB	0	1	1	2's COMP ADD
INC	1	0	1	INC X BY "1"
DEC	1	1	0	DEC w/ NO CARRY IN



SUBTRACTORS (FULL) $A - B$ WHERE $A > B$ OR $B > A$

$$C = \text{BORROW BIT}$$

$$D = \text{DIFFERENCE}$$

NOTE: D IS THE SAME AS S

$D = 1$ FOR ODD #1's

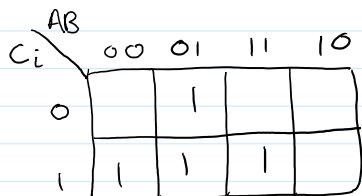
$$\therefore D = A \oplus B \oplus C_i$$

A	B	C_i	D	C_o
0	0	0	0	0
0	1	0	1	1
1	0	0	1	0
1	1	0	0	0
0	0	1	1	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	1

C_o (BORROW) IS DIFFERENT

$$\text{IF } B + C_i > A \Rightarrow C_o = 1$$

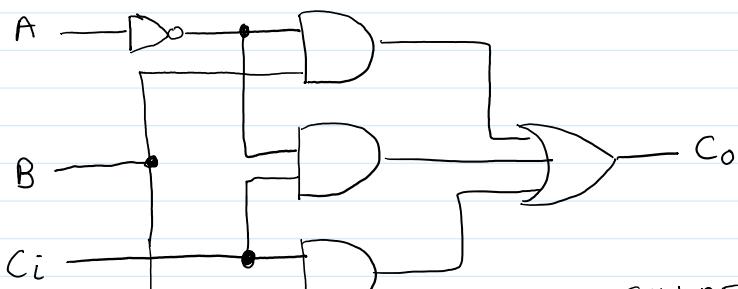
K MAP FOR C_o



$$C_o = \bar{A}B + \bar{A}C_i + BC_i$$

$$= \bar{A}B + C_i(\bar{A} + B)$$

SUBTRACTOR CKT FOR BORROW OUT

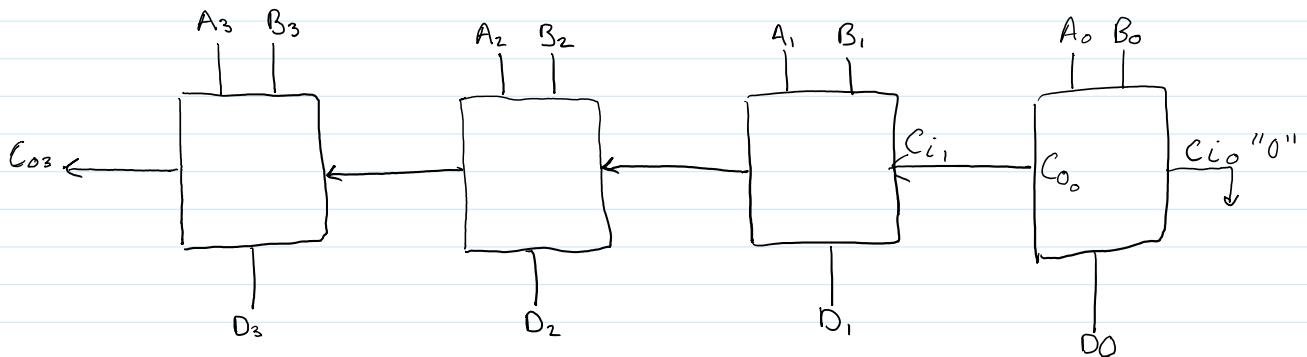


ANSWER: $C_o = \bar{A}B + C_i(\bar{A} + B)$



CAN BE NAND/NAND FOR INCREASED SPEED

4 BIT RIPPLE SUBTRACTOR USING FULL SUBTRACTORS



$$\begin{array}{ll} \text{IF } A > B & C_{03} = 0 \\ A < B & C_{03} = 1 \end{array}$$

1 - 7
SUBTRACT $A - B$ WHERE $A = 1011$ $B = 0111$ (REFER TO TRUTH TABLE)

BIT 0: $A = 1, B = 1, C_i = 0$

$$110 \Rightarrow D0 = 0 \quad C_0 = 0$$

BIT 1: $A = 1, B = 1, C_i = 0$

$$110 \Rightarrow D1 = 0 \quad C_0 = 0$$

BIT 2: $A = 0, B = 1, C_i = 0$

$$010 \Rightarrow D2 = 1, C_0 = 1$$

BIT 3: $A = 1, B = 0, C_i = 1$

$$101 \Rightarrow D3 = 0, C_0 = 0$$

$$\text{ANS: } 0, 0100 = 4$$

$C_0 = 0$ MEANS $A > B$

$C_0 = 1$ MEANS $B > A$

NO BORROW!

SUBTRACT $A - B$ WHERE $A = 0111$ $B = 1011$ USING FULL SUBTRACTOR

BIT 0: A=1 B=1 Ci=0

$$110 \Rightarrow D0 = 0 \quad C_0 = 0$$

BIT 1: A=1 B=1 Ci=0

$$110 \Rightarrow D1 = 0 \quad C_0 = 0$$

BIT 2: A=1 B=0 Ci=0

$$100 \Rightarrow D2 = 1 \quad C_0 = 0$$

BIT 3: A=0 B=1 Ci=0

$$010 \Rightarrow D3 = 1 \quad C_0 = 1$$

(NEED TO BORROW)

$$\text{ANS } 1, 1100 = 12 \times$$

$$C_0(3) = B > A$$

CAN GET CORRECT ANSWER BY DOING REVERSE TWO'S COMPLIMENT

$$1100 - 0001 = 1011 \text{ COMPLIMENT} = 0100 = 4$$

$$\text{SUBTRACT } A - B \quad A = 1011 - B = 0111$$

USING FULL ADDER SET FOR 2'S COMPLIMENT (B GETS COMPLIMENT)

$$B' = 1000 \quad C_{i_0} = 1$$

A	B	Ci	S	C0
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

BIT 0: A=1, B=0, Ci=1

$$101 \Rightarrow S0 = 0, \quad C_0 = 1$$

BIT 1: A=1, B=0, Ci=1

$$101 \Rightarrow S1 = 0, \quad C_0 = 1$$

BIT 2: A=0, B=0, Ci=1

$$001 \Rightarrow S2 = 1, \quad C_0 = 0$$

BIT 3: A=1, B=1, Ci=0

$$\text{ANS: } 1, 0100 = 4$$

$$110 \Rightarrow S3=0, C_0=1$$

NOTE: $C_0=1$ MEANS $A > B$ } OPPOSITE OF
 $C_0=0$ MEANS $B > A$ } FULL SUBTRACTOR

$$\begin{array}{r} 7 \\ \text{SUBTRACT } A - B \quad A = 0111 \quad B = 1011 \\ - \quad 4 \end{array}$$

$$2 \text{ COMP FULL ADDER } \bar{B} = 0100 \quad C_{i_0} = 1$$

$$\text{BIT 0: } A = 1 \quad B = 0 \quad C_i = 1$$

$$101 \Rightarrow S0 = 0 \quad C_0 = 1$$

$$\text{BIT 1: } A = 1 \quad B = 0 \quad C_i = 1$$

$$101 \Rightarrow S1 = 0 \quad C_0 = 1$$

$$\text{BIT 2: } A = 1 \quad B = 1 \quad C_i = 1$$

$$111 \Rightarrow S2 = 1 \quad C_0 = 1$$

$$\text{BIT 3: } A = 0 \quad B = 0 \quad C_i = 1$$

$$001 \Rightarrow S3 = 1 \quad C_0 = 0$$

$$\text{ANS } 0, \underbrace{1100}_{12} \quad B > A$$

MUST DO REVERSE 2's COMP ON 1100

$$\text{COMP} = 0011$$

$$\text{ADD 1} = 0100$$

$$\begin{aligned} \text{ANS} &= 4 \\ &= -4 \end{aligned}$$

ADDER PROPAGATION TIMES

5ms / GATE

$$S = A \oplus B \oplus C_i \leftarrow 2 \text{ XOR}'s = 30 \text{ ns}$$

FASTER WAY

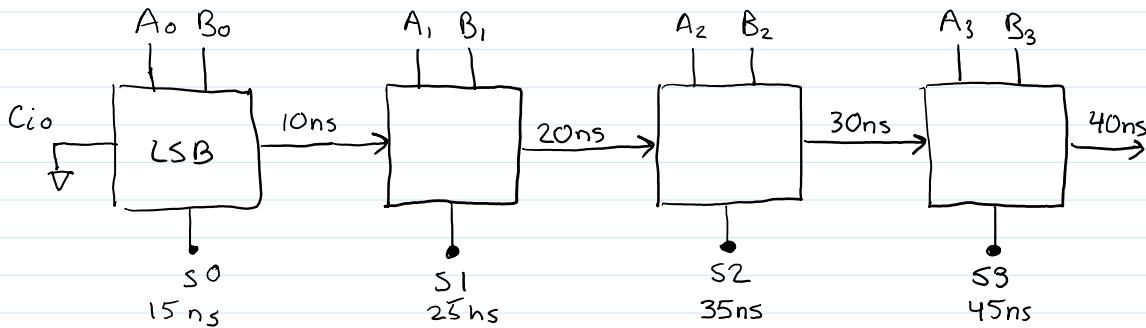
$$S = A\bar{B}\bar{C}_i + \bar{A}B\bar{C}_i + A\bar{B}C_i + ABC_i$$

INV - NAND-NAND $3 \times 5 \text{ ns} = 15 \text{ ns}$

$$C_o = AB + AC_i + BC_i$$

NAND/NAND $2 \times 5 \text{ ns} = 10 \text{ ns}$

RIPPLE ADD



NOTE: S_1, S_2, S_3, C_o VALUES WILL KEEP CHANGING FROM 15ns TO 45ns

ANSWER VALID $> 45 \text{ ns}$ SLOW!

LOOK AHEAD CARRY ADDER IS MUCH FASTER THAN RIPPLE SINCE CARRIES ARE NOT RIPPLED THROUGH THE ADDER

LET A_i, B_i BE INPUT BITS OF THE i^{TH} CELL OF THE ADDER

DEFINE 2 NEW VARIABLES FOR THE i^{TH} CELL

$G_i = A_i B_i$ AS GENERATE QUANTITY

$P_i = A_i + B_i$ AS PROPAGATE QUANTITY

C_i = A CARRY MADE FROM PREVIOUS CELLS

$P's, G's + C_o \Rightarrow C_{i0}$

IT IS USED ONLY BY THE i^{TH} CELL FOR S_i

WHERE $S_i = A_i \oplus B_i \oplus C_i$

E.G. FIRST CELL = 0^{TH} CELL

0^{TH} CELL USES A_0, B_0, C_0

$C_0 \Rightarrow$ CARRY IN, $C_0(\text{IN})$

$S_0 = A_0 \oplus B_0 \oplus C_0$

2^{nd} CELL = 1^{st} CELL USES A, B, C , WHERE $C_1 = G_0 + P_0 C_0$ NAND/NAND

C_1 , THE CARRY USED IN THE ONE CELL COMES FROM A CARRY GENERATED IN 0^{th} CELL G_0 , OR A CARRY IN, C_0 AND P_0 , PROPAGATED BY THE 0^{th} CELL

$$\text{RECALL: } C = \underbrace{AB}_G + \underbrace{AC + BC}_{P.C}$$

$$\begin{aligned} S_1 &= A_1 \oplus B_1 \oplus C_1 \\ &= A_1 \oplus B_1 \oplus (G_0 + P_0 C_0) \end{aligned}$$

NOTE: C_1 DOES NOT EXIT THE 1 CELL

2^{nd} CELL (REALLY THE THIRD CELL IF 0^{th} IS COUNTED) USES A_2, B_2, C_2

$$\begin{aligned} C_2 &= G_1 + P_1 \cdot C_1 \\ &\quad \swarrow \qquad \text{GENERATED IN CELL 2 ONLY!} \\ C_1 &= G_0 + P_0 C_0 \end{aligned}$$

$$\begin{aligned} \therefore C_2 &= G_1 + P_1 (G_0 + P_0 C_0) \\ &= G_1 + P_1 G_0 + P_1 P_0 C_0 \qquad \text{NAND/NAND} \end{aligned}$$

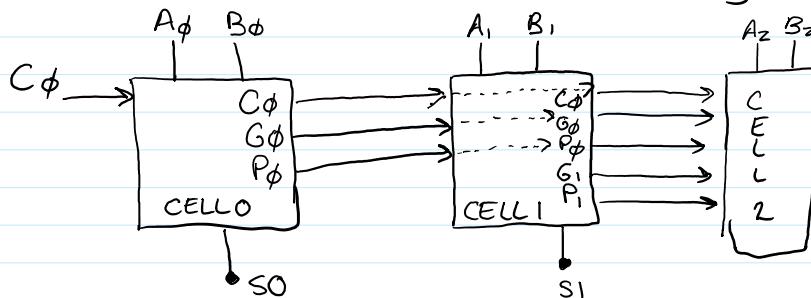
* NOTE: C_2 HAS SAME LEVEL OF GATES AS C_1

\therefore A CARRY USED IN 2^{nd} CELL FOR S_2 IS EITHER GENERATED IN CELL 1

OR GENERATED IN CELL 0 AND PROPAGATED THROUGH CELL 1

OR CARRIED INTO ADDER (C_0) AND PROPAGATED THROUGH CELL 0 + CELL 1

SCHEMATIC OF LOOK-AHEAD TWO CELLS 0+1



CELL 0 TO CELL 1 $\Rightarrow C_0, P_0 + C_0$

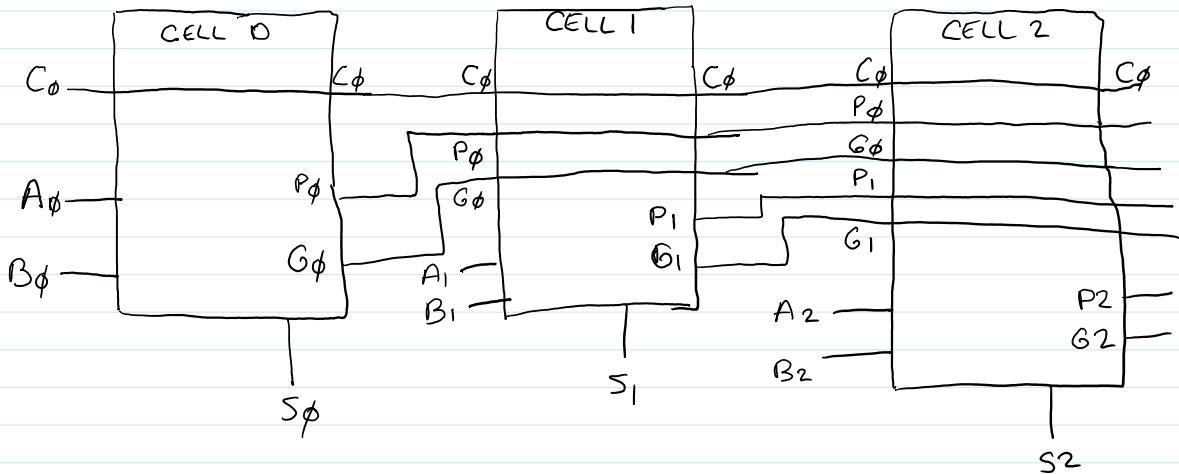
THE CARRY GENERATED IN THE i^{TH} CELL IS ONLY USED IN THE i^{TH} CELL AND NOT PASSED TO CELL $i+1$, SO THE CARRIES ARE NOT PASSED (RIPPLED) THROUGH THE CELLS ONLY THE G'S & P'S

EACH CELL PASSES ITS G+P ALONG W/ ALL G'S & P'S PASSED TO IT;

LOTS OF WIRES!

* THE i^{TH} CARRY IS NOT MADE FROM THE i^{TH} P+G BUT FROM ALL PASSED G'S, P'S, & C_0

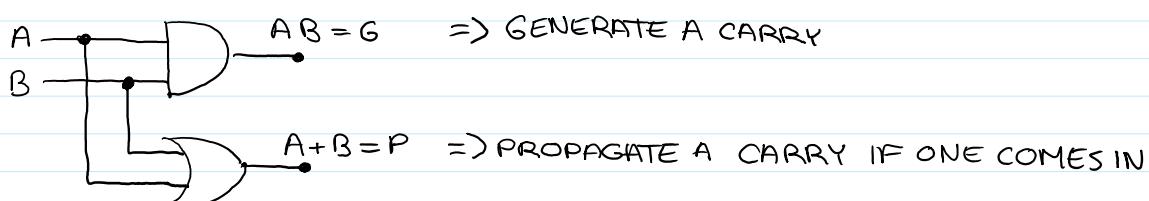
CARRY LOOK AHEAD ADD 3RD CELL 0, 1, 2



CELL	INPUTS	OUTPUTS
0	C_ϕ, A_ϕ, B_ϕ	$C_\phi, P_\phi, G_\phi, S_\phi$
1	$A_1, B_1, C_\phi, P_\phi, G_\phi$	$C_\phi, P_\phi, G_\phi, P_1, G_1, S_1$
2	$A_2, B_2, C_\phi, P_\phi, G_\phi, P_1, G_1$	$C_\phi, P_\phi, G_\phi, P_1, G_1, P_2, G_2, S_2$
N	A_N, B_N, C_ϕ	S_N, C_N
	ALL P+G'S $\xrightarrow{o \rightarrow N-1}$	

SPEED & TIMING

P+G CKT TAKES 2 GATES TIME

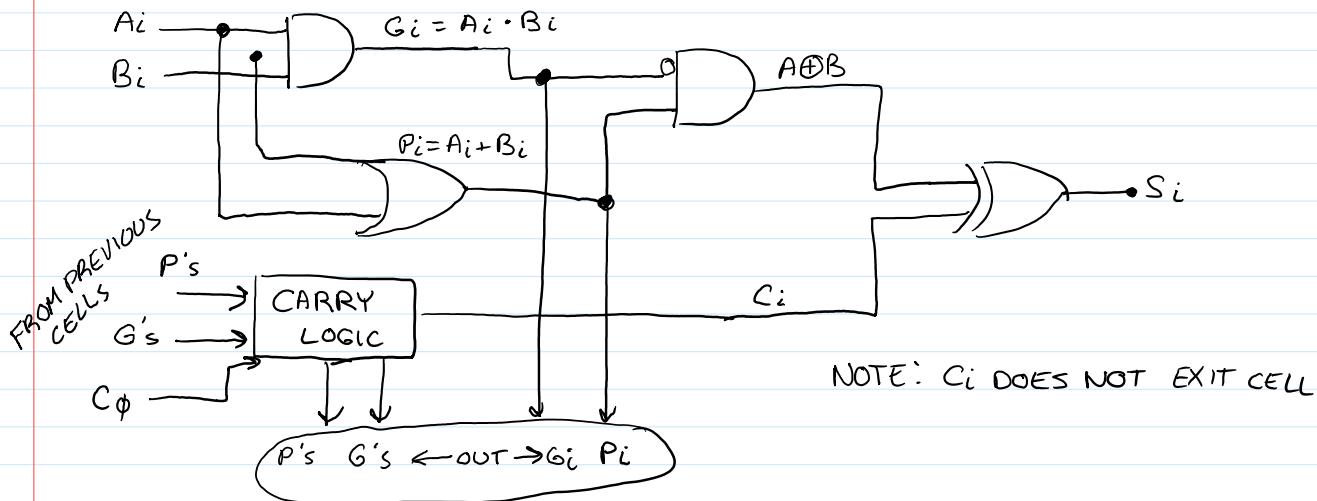


* NOTE: BOTH $G + P$ ARE INDEPENDANT OF CARRY IN

i^{TH} CKTS FOR S_i, G_i, P_i FROM $A_i, B_i, P's, G's, C_\phi$

$$S_i = A_i \oplus B_i \oplus C_i = (A_i \oplus B_i) \oplus C_i$$

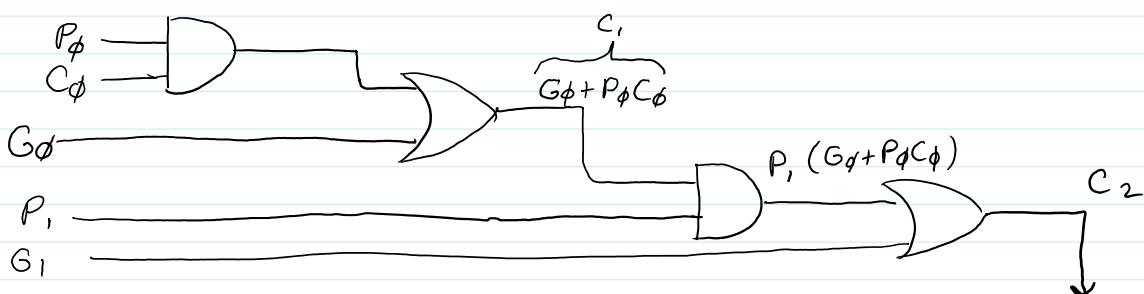
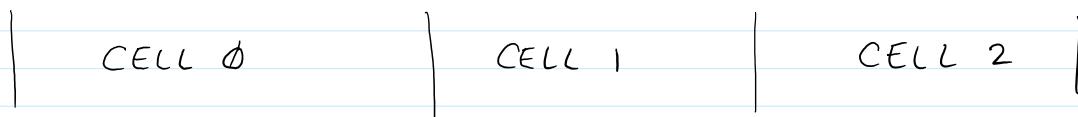
$$A \oplus B = (A+B) - A \cdot B = (A+B) \cdot (\overline{A} \cdot \overline{B})$$



CARRY LOGIC CELL 2

CKT IN DETAIL

$$C_2 = G_1 + P_1 \cdot (G_\phi + P_\phi C_\phi) \equiv 4 \text{ 2-LEVEL GATES}$$



* NEED 4 2-LEVEL + 1 2-LEVEL (FOR $P+G=10$)

NEED TO REDUCE THE LEVELS

USED TO
MAKE S_2
W/ A_2 AND B_2

DISTRIBUTE P_1 :

$$C_2 = G_1 + P_1 G_\phi + P_1 P_\phi G_\phi$$

ONLY REQUIRES 2 2-LEVELS + 1 2-LEVEL = 5
MUCH FASTER

IMPLEMENT THIS WITH GATES \rightarrow NAND/NAND FOR FURTHER IMPROVEMENTS

RECALL: A CARRY IS A "1" IN CELL 2 IF IT WAS GENERATED IN CELL 1 OR GENERATED IN CELL 0 AND PROPAGATED THROUGH CELL 1 OR CARRY INTO CELL 0 AND PROPAGATED THROUGH CELL 0 + 1

74 LS 283 \Rightarrow CARRY LOOKAHEAD ADDER

RECALL: $C_1 = G_\phi + P_\phi C_\phi$ IS THE CARRY NEEDED BY THE 1ST CELL TO MAKE S_1

283 USES A MODIFICATION OF THIS EQUATION WHICH IS STILL FAST BUT EASIER TO IMPLEMENT WITH NANDS WITH A PRODUCT OF SUMS EQUATION

WE WILL SEE LATER, MORE OBVIOUS WITH C_2

$$C_1 = P_\phi \cdot (G_\phi + C_\phi)$$

THE TWO EQUATIONS ARE EQUIVALENT

* CAN BE PROVEN WITH K MAPS

PROOF:

$$C_1 = P_\phi (G_\phi + C_\phi) = G_\phi + P_\phi C_\phi$$

$$(A+B)(AB+C_\phi) = AB + (A+B)C_\phi$$

$$AAB + AC_\phi + BAB + BC_\phi = AB + AC_\phi + BC_\phi$$

$$AB + AC_\phi + AB + BC_\phi = AB + AC_\phi + BC_\phi$$

$$AB + AC_\phi + BC_\phi = AB + AC_\phi + BC_\phi$$

$$C_2 = P_1 \cdot (G_1 + C_1)$$

$$C_2 = P_1 \cdot (G_1 + P_\phi \cdot (G_\phi + C_\phi))$$

WE DON'T WANT TO USE C_1 TO MAKE C_2

WE WANT TO USE THE STUFF THAT MADE C_1

$$C_2 = P_1 G_1 + P_1 P_\phi G_\phi + P_1 P_\phi C_\phi$$

REQUIRES 4 GATES
w/ 11 INPUTS

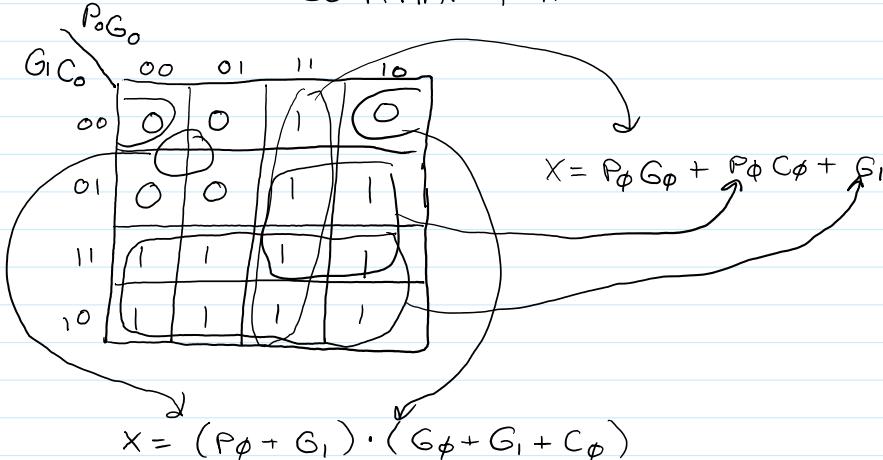
REDUCTION

$$C_2 = P_1 \cdot (G_1 + P_\phi G_\phi + P_\phi C_\phi) = P_1 \cdot X$$

REQUIRES THREE DOUBLE LEVELS OF GATES

REDUCE $X = G_1 + P_\phi G_\phi + P_\phi C_\phi$ FROM A SUM OF PRODUCTS (SOP)
TO A PRODUCT OF SUMS THEN ANDED WITH P_1

USE K MAP & PI OF MAXTERMS



$$\therefore \text{REDUCED VERSION } C_2 = (P_1) \cdot (G_1 + P_\phi) \cdot (G_1 + G_\phi + C_\phi)$$

3 GATES & 9 INPUTS

THIS IS AN ALTERNATIVE WAY OF DOING THE LOOKAHEAD
FROM ORIGINAL VERSION

$$C_2 = C_\phi \cdot P_\phi \cdot P_1 + G_\phi P_1 P_\phi + G_1 P_1$$

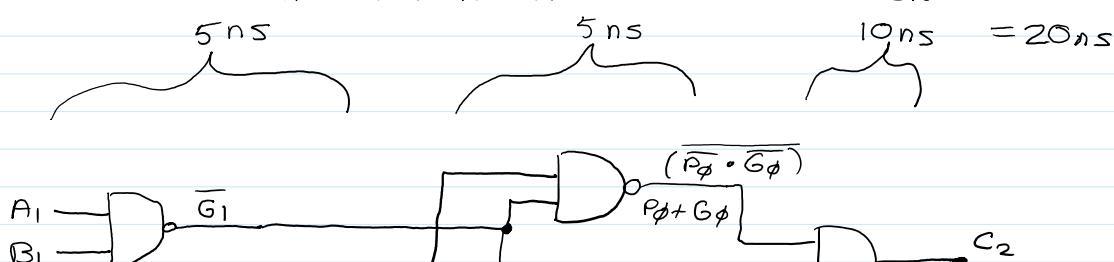
4 GATES & 11 INPUTS MORE HARDWARE

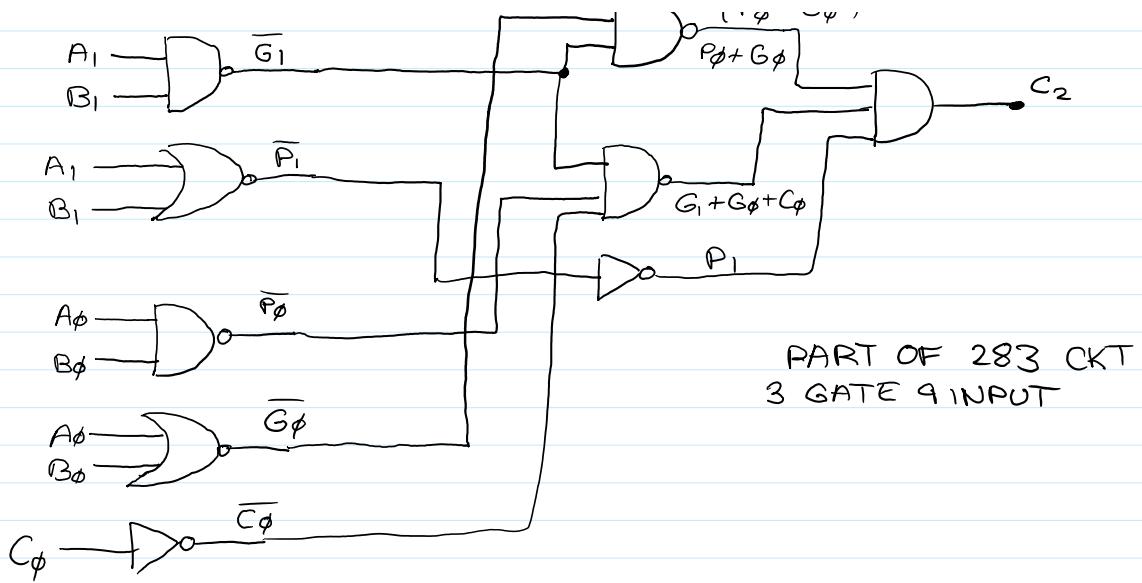
HOW MUCH FASTER IS THIS METHOD FOR C_2 NEEDED FOR S_2
COMPARED TO RIPPLE ADDER = 20 ns

THIS METHOD USING NAND GATES

NAND, NOR & INV = 5 ns AND, OR & BUFFER = 10 ns

IMPLEMENTATION USING REDUCED VERSION

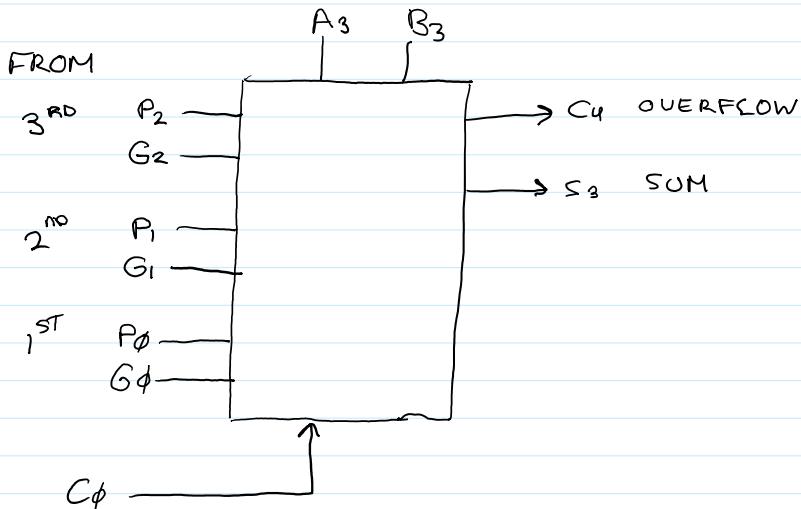




PART OF 283 CKT
3 GATE 9 INPUT

IT IS ATIE WITH RIPPLE FULL ADDER AT 20ns

LETS LOOK AT CARRY LOOK AHEAD FOURTH CELL, S_3 OF 283



THE FOURTH CELL WILL MAKE C_3 NEED ED FOR S_3
 $C_4 = C_{\text{OUT}}$

FROM $C_\phi, P_\phi, G_\phi, P_1, G_1, P_2, G_2$

$$\text{WHERE } C_3 = P_2 \cdot (G_2 + P_1) \cdot (G_2 + G_1 + P_\phi) \cdot (G_2 + G_1 + G_\phi + C_\phi) \\ = 20\text{ns}$$

$$S_3 = 20\text{ns} + 15\text{ns} = 35\text{ns} \leftarrow \text{BOTH 10ns FASTER THAN RIPPLE FULL ADDER}$$

LOOK AT C_4

$$C_4 = P_3 (G_3 + P_2) (G_3 + G_2 + P_1) (G_3 + G_2 + G_1 + P_\phi) (G_3 + G_2 + G_1 + G_\phi + C_\phi)$$

THIS CARRY BIT TAKES 2 LEVELS OF NAND/NOR/INV GATES AND 1 LEVEL OF AND

IT WOULD TAKE 20ns FOR C_4

THE RIPPLE ADDER TOOK 40ns FOR C_4

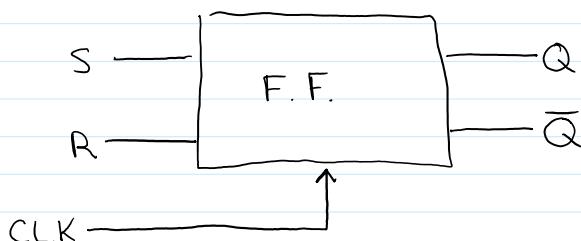
THIS IS TWICE AS FAST FOR 4 BITS

32 BIT ADDER RIPPLE \approx 500ns

LOOKAHEAD = 20ns

HOWEVER LOTS OF WIRES + GATES

BISTABLE MULTIVIBRATOR



TWO TYPES

(1) A SYNCHRONOUS \Rightarrow NO CLK

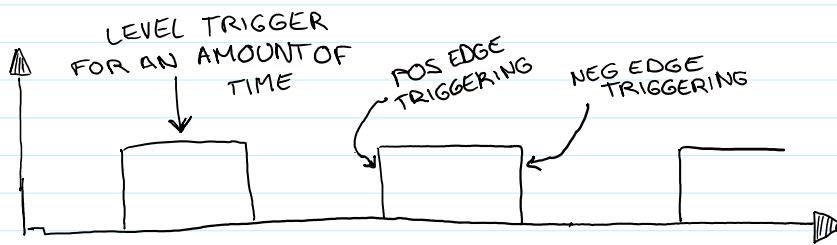
(2) SYNCHRONOUS \Rightarrow F.F WILL CHANGE STATES WHEN CLK=1 (LEVEL TRIGGERED ON A NEG/POS EDGE)

EXCITATION TABLE

S	R	Q_P	Q_N
0	0	0	0
0	0	1	1
<hr/>			
0	1	0	0
0	1	1	0
<hr/>			
1	0	0	1
1	0	1	1
<hr/>			
1	1	0	U
1	1	1	U

UNDEFINED OSCILLATOR

LEVEL TRIGGER
... AMOUNT OF TIME



BASIC MEMORY ELEMENTS

LATCH

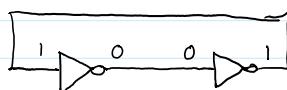
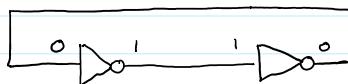
- CLK INPUT IS LEVEL SENSITIVE
- OUTPUT CAN CHANGE MULTIPLE TIMES DURING A CLK CYCLE
- OUTPUT CHANGES WHILE CLK IS ACTIVE

FLIP FLOPS

- CLK INPUT IS EDGE SENSITIVE
- OUTPUT CAN CHANGE ONLY ONCE DURING A CLK CYCLE
- OUTPUT CHANGES ON CLK TRANSITION

BOTH LATCHES AND FFs USE FEEDBACK TO ACHIEVE "MEMORY"

FEEDBACK CKTS w/ 2 STABLE STATES



SET-RESET (SR) LATCH (THEORY APPROACH)

S R LATCH HAS 2 INS

- SET (S) INPUT
- RESET (R) INPUT

CAN BE CONSTRUCTED FROM TWO CROSS-COUPLED NOR OR NAND GATES

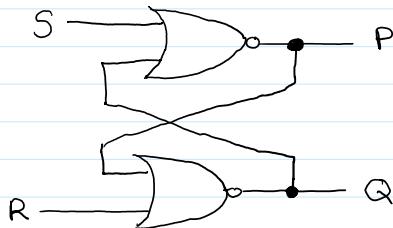
BEHAVIOR

SET: LATCH OUTPUT SET TO 1 ($Q^+ = 1$)

RESET: LATCH OUTPUT RESET TO 0 ($Q^+ = 0$)

STORE: LATCH OUTPUT DOES NOT CHANGE ($Q^+ = Q$)

NOR CONSTRUCTION



IF $S=1$ (SET) $Q^+ = 1$

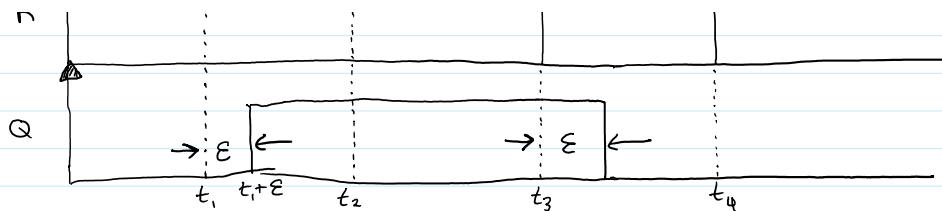
IF $R=1$ (RESET) $Q^+ = 0$

IF $S=R=0$ (STORE) $Q^+ = Q$ (NO CHANGE)

$S=R=1$ IS NOT ALLOWED

TIMING DIAGRAM





$E = \text{PROPAGATION DELAY OF LATCH}$

CHARACTERISTIC EQUATION

RQ	0	1
00	0	1
01	1	1
11	X	
10		X

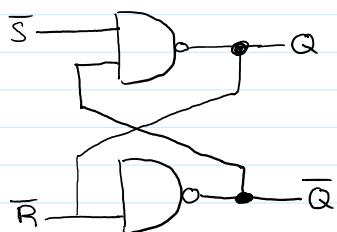
$$Q^+ = S + \bar{R}Q$$



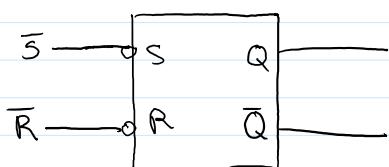
SR USING NAND GATES



A	B	F
0	X	1
1	X	X

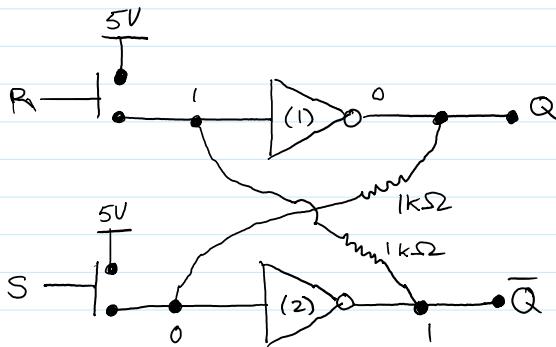


S	R	Q	\bar{Q}
1	1	0	0
1	1	1	1
1	0	0	0
1	0	1	0
0	1	0	1
0	1	1	1
0	0	0	0
0	0	1	0



SR LATCH (PHYSICAL APPROACH)

LET'S BUILD F.F. W/ GATES USING 2 INVERTERS FROM A 74LS04



TO CHANGE STATE, PUSH S BUTTON AND INV 2 INPUT GOES TO 5V AND $\bar{Q} \Rightarrow 0$

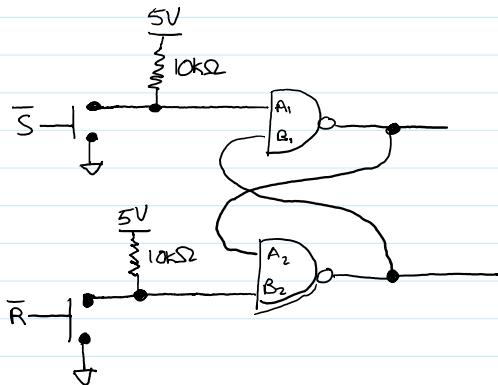
FORCES INPUT TO INV 1 = 0 $\Rightarrow Q = 5V$

$$I(\text{SWITCH}) = \frac{5 - 0}{1\text{k}\Omega} = 5\text{mA}$$

NOTE IF SWITCH MADE FROM TRANSISTORS, THEN 5mA IS BIG

S + R ARE ACTIVE HIGH

TO REDUCE SWITCH CURRENT AND PROVIDE ISOLATION BETWEEN SWITCHES AND OUTPUTS



NAND

A	B	V_o
0	0	1
0	1	1
1	0	1
1	1	0

PUSH \bar{R} (RESET) BUTTON $B_2 = 0$

THEN $A_2 = 0$ $B_2 = 0 \Rightarrow V_{o_2}(\bar{Q}) = 1$

NO CHANGE (FF ALREADY RESET)

NOTE: ACTIVE LOW

PUSH \bar{S} (SET) BUTTON $A_1 = 0$

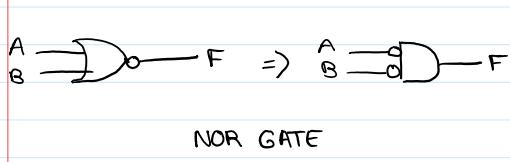
THEN $A_1 = 0 \quad B_1 = 1 \Rightarrow V_{O_1} = 1$

$A_2 = 1 \quad B_2 = 1 \Rightarrow V_{O_2} = 0$

$B_1 = 0 \Rightarrow V_{O_1} = 1 \Rightarrow \text{F.F. SET}$

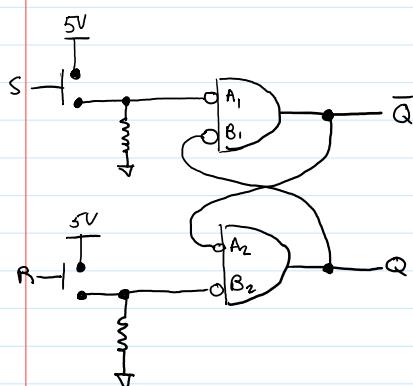
TO EMPLOY ACTIVE HIGH INPUTS

MOVE NEGATIONS AROUND TO INPUTS



A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

A	B	F
0	X	X
1	X	0



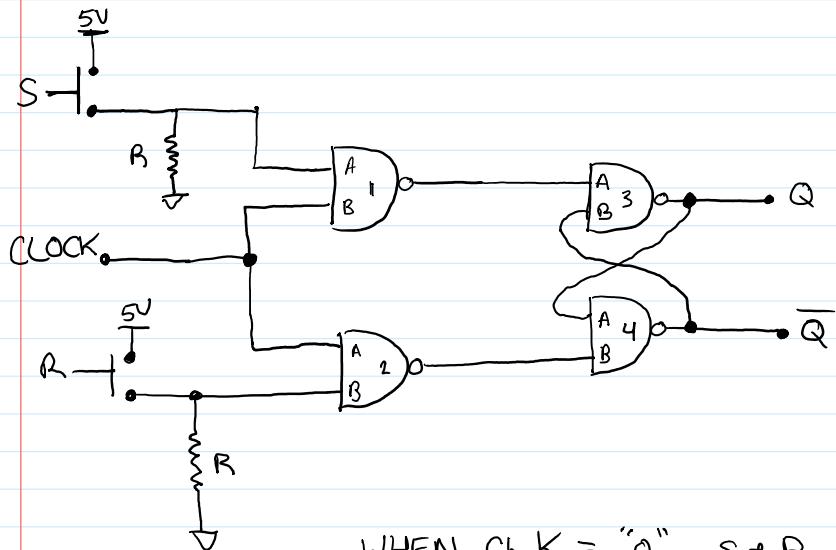
PUSH $R \Rightarrow B_2 = 1 \quad A_2 = 1 \quad \text{OUT} = 0$
NO CHANGE

PUSH $S \Rightarrow A_1 = 5V \quad B_1 = 0 \quad \bar{Q} = 0$
 $A_2 = 0 \quad B_2 = 0 \quad Q = 1$
 $B_1 = 1 \quad A_1 = 0 \quad \bar{Q} = 0$

F.F. IS SET

CLK'D SR FF

DATA IS SAMPLED ONLY DURING CLOCK HIGH (LEVEL TRIGGERED)



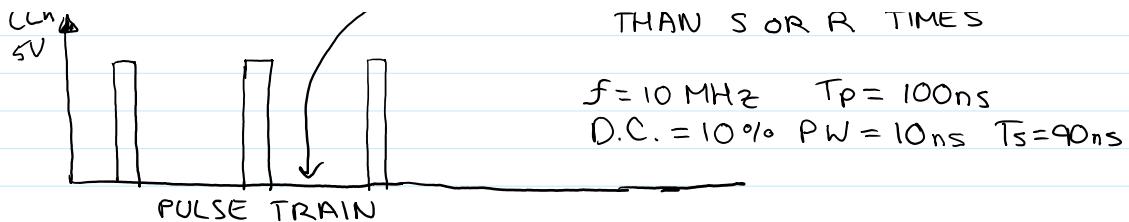
NAND		
A	B	V _O
0	0	1
0	1	1
1	0	1
1	1	0

A	B	V _O
0	X	1
1	X	X

WHEN $\text{CLK} = "0"$ S + R ARE DISABLED

CLK PULSE MUCH SMALLER
THAN S OR R TIMES

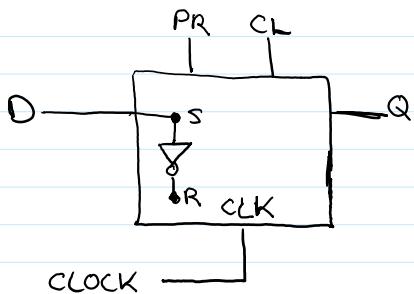
$$f = 10 \text{ MHz} \quad T_p = 100 \text{ ns}$$



WHEN $\text{CLK} = "1"$ + $S = 1$ $G_1 = 0 = A_3$

$$\left. \begin{array}{l} A_3 = 0 + B_3 = 1 \Rightarrow Q = 1 = A_4 \\ \text{THEN } A_4 = 1 + B_4 = 1 \Rightarrow \bar{Q} = 0 = B_3 \end{array} \right\} \text{F.F. IS SET}$$

D F.F. DATA OR DELAY



DELAY F.F. HAS AN INTERNAL
INVERTER FROM S TO R

WHATEVER APPEARS ON D WILL BE
CLOCKED TO OUTPUT Q (ONE CLOCK
CYCLE LATER)

D	CLK	Q_p	Q_n
0	0	Q_p	Q_p
0	1	—	0
1	0	Q_p	Q_p
1	1	—	1

D F.F. CAN ALSO HAVE

$PR = \text{PRESET}$ WILL SET $Q = 1$

$CL = \text{CLEAR}$ $Q = 0$

- = DON'T CARE

REVIEW OF F.F.

S.R. F.F.

3 INS

- CLK
- SET (S)
- RESET (R)

SIMILAR TO SR LATCH

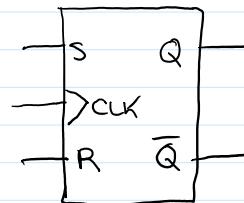
$S = 1$ SETS THE F.F. ($Q^+ = 1$)
 $R = 1$ RESETS THE F.F. ($Q^+ = 0$)

LIKE D F.F., Q OUTPUT OF AN SR FF ONLY CHANGES IN RESPONSE TO ACTIVE CLK EDGE

- POSITIVE EDGE TRIGGERED

- NEGATIVE EDGE TRIGGERED

S	R	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X



JK FF HAS 3 INS

- CLK
- J
- K

SIMILAR TO SR FF

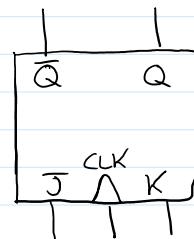
J CORRESPONDS TO S $J=1 \rightarrow Q^+=1$

K CORRESPONDS TO R $K=1 \rightarrow Q^+=0$

DIFFERENCE FROM SR FF INPUT COMBO $J=1, K=1$ IS ALLOWED

$J=K=1$ CAUSES THE Q OUTPUT TO TOGGLE AFTER AN ACTIVE CLK EDGE

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



CHARACTERISTIC EQ
 $Q^+ = J\bar{Q} + \bar{K}Q$

TOGGLE FF

2 INS

TOGGLE \leftrightarrow

2 INS

- CLK
- TOGGLE (T)

T CONTROLS STATE CHANGE

$T=0$, STATE DOES NOT CHANGE ($Q^+ = Q$)

$T=1$, STATE CHANGES FOLLOWING AN ACTIVE CLK EDGE ($Q^+ = \bar{Q}$)

TFF ARE OFTEN USED TO DESIGN COUNTERS

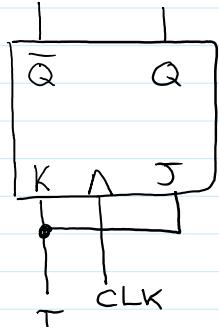
T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0



CHARACTERISTIC EQ

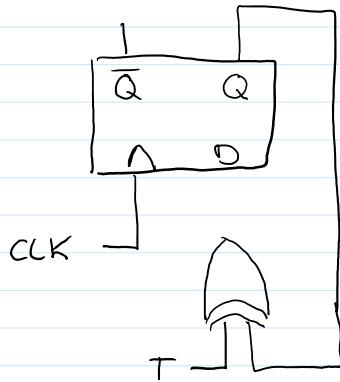
$$Q^+ = T\bar{Q} + \bar{T}Q = T \oplus Q$$

CONVERSION OF A JK TO T

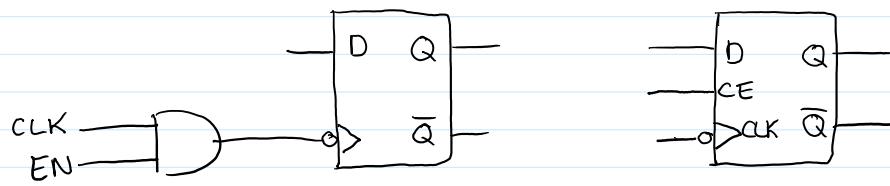


(REFER TO TABLE)

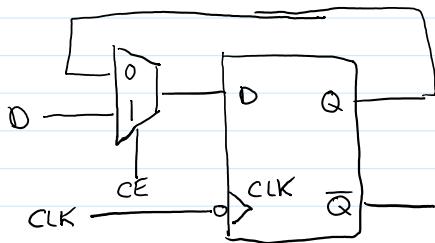
CONVERSION OF D TO T



D FF w/ CLK ENABLE



GATING THE CLK



CHARACTERISTICS EQ OF FFS

$$T: Q^+ = \bar{Q}T + Q\bar{T}$$

$$SR: Q^+ = S + \bar{R}Q \quad \text{NEXT STATE} = 1 \quad \text{IF SET} = 1 \quad \text{OR IF PRESENT STATE} = 1 \\ \text{AND NO RESET}$$

$$D: Q^+ = D \quad \text{NEXT STATE} = D \quad \text{ON NEXT CLK}$$

$$D \text{ w/ EN: } Q^+ = D \cdot EN + \bar{EN} \cdot Q \quad \text{NEXT STATE} = D \text{ IF } EN = 1 \text{ OTHERWISE } EN = 0 \Rightarrow Q^+ = Q$$

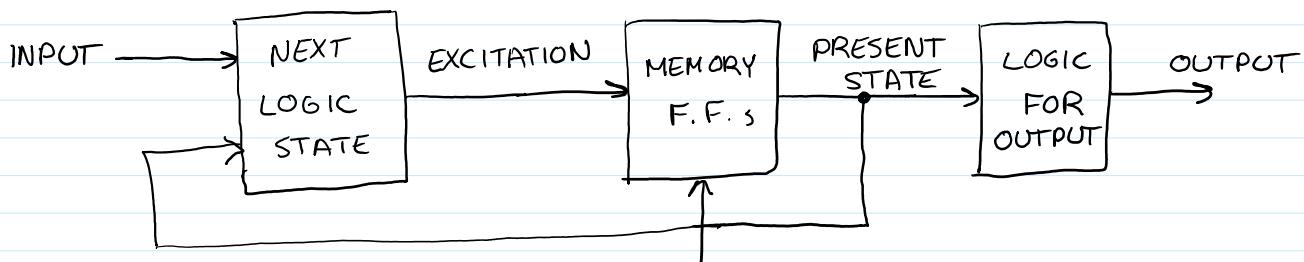
$$JK: Q^+ = J\bar{Q} + \bar{K}Q \quad \text{NEXT STATE} = 1 \quad \text{IF } J = 1 \text{ AND PRESENT}$$

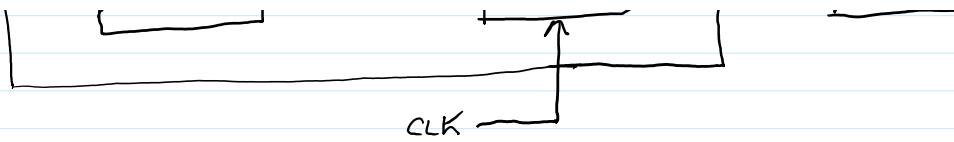
$$\text{STATE} = 0 \quad \text{OR } Q = 1 + K = 0$$

$$\text{NEXT STATE} = 0 \quad \text{IF } K = 1 \text{ AND } Q = 1 \text{ OR } Q = 0 + J = 0$$

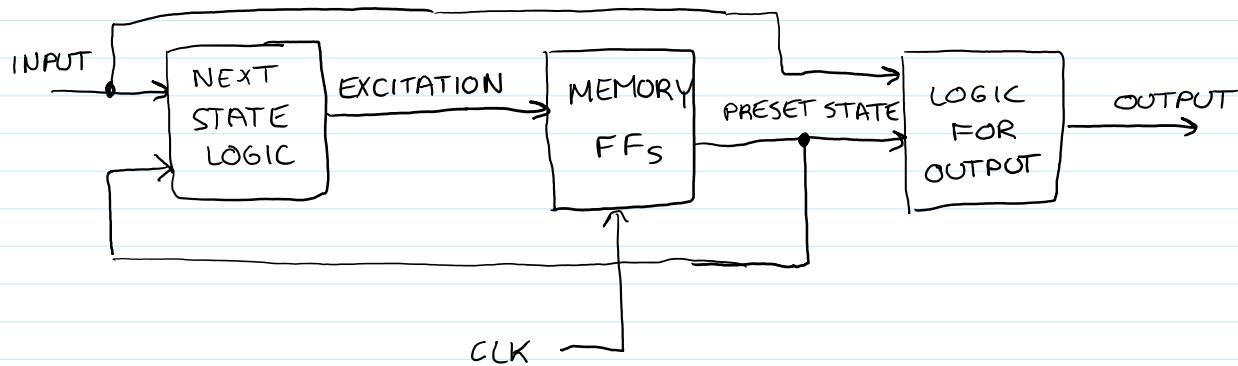
STATE MACHINES ARE SEQUENTIAL MACHINES THAT USE HISTORY AND INPUTS TO DETERMINE THE NEXT STATE OF THE MACHINE AND THE OUTPUT

MOORE MACHINE





MEALY MACHINE



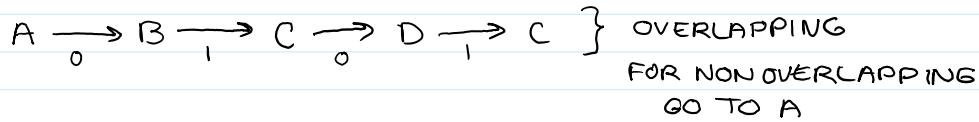
NOTE: IF THERE ARE NO INPUTS THEN MACHINE IS A COUNTER

STATE DIAGRAM THEN STATE TABLE AS ONE WAY OF DESIGNING A MACHINE

EX: DESIGN A SEQUENCE DETECTOR THAT DETECTS A SEQUENCE

"0101" FROM AN INPUT BIT STREAM

CORRECT STATE SEQUENCE: FOR A "1" OUTPUT FROM START



REVIEW OF STATE NAMES

A = START STATE

B = STATE WHERE A "0" HAS BEEN STORED

C = STATE WHERE "01" HAS BEEN STORED

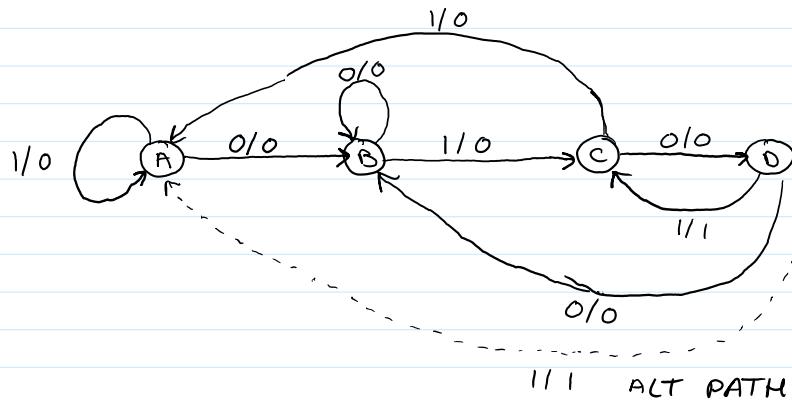
D = STATE WHERE "010" HAS BEEN STORED

OUTPUT: IF IN STATE D, A "1" COMES IN THEN THE CORRECT SEQUENCE HAS BEEN RECEIVED AND A "1" IS OUTPUTTED AND WE GO TO STATE C

OR NON OVERLAPPING GO TO A => START OVER

1/0

OR NON OVERLAPPING GO TO A => START OVER



WE CAN SHOW THIS WITH A TABLE

P.S.	N.S., Z	
	X=0	X=1
A	B, 0	A, 0
B	B, 0	C, 0
C	D, 0	A, 0
D	B, 0	C, 1

X = INPUT
 NS = NEXT STATE
 PS = PRESENT STATE
 Z = OUTPUT

4 STATES ARE NEEDED TO REPRESENT ALL POSSIBILITIES

TWO STATE VARIABLES ARE NEEDED TO REPRESENT THE 4 STATES

$$2^N = 4 \quad N = \# \text{ OF STATE VARIABLES}$$

STATE DESIGNATION

$$A = \overline{Q}_1, \overline{Q}_0 \Rightarrow 00$$

$$B = \overline{Q}_1, Q_0 \Rightarrow 01$$

$$C = Q_1, Q_0 \Rightarrow 11$$

$$D = Q_1, \overline{Q}_0 \Rightarrow 10$$

OFF EXCITATION TABLE

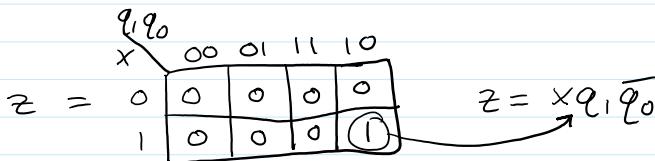
Q _P	Q _N	D
0	0	0
0	1	1
1	0	0
1	1	1

TRANSITION/OUTPUT TABLE

P.S.	NEXT STATE		OUTPUT	
	X=0	X=1	X=0	X=1
q ₁ q ₀	Q ₁ , Q ₀	Q ₁ , Q ₀	Z	Z
0 0	0 1	0 0	0	0

Y_1	Y_0	Q_1	Q_0	Q_1	Q_0	Z	T
0	0	0	1	0	0	0	0
0	1	0	1	1	1	0	0
1	1	1	0	0	0	0	0
1	0	0	1	1	1	0	1

EXCITATION TABLES AND OUTPUT K MAPS w/ D FF's



$Q_1 = D_1 = \sum_{q_1 q_0} \{00, 01, 11\}$

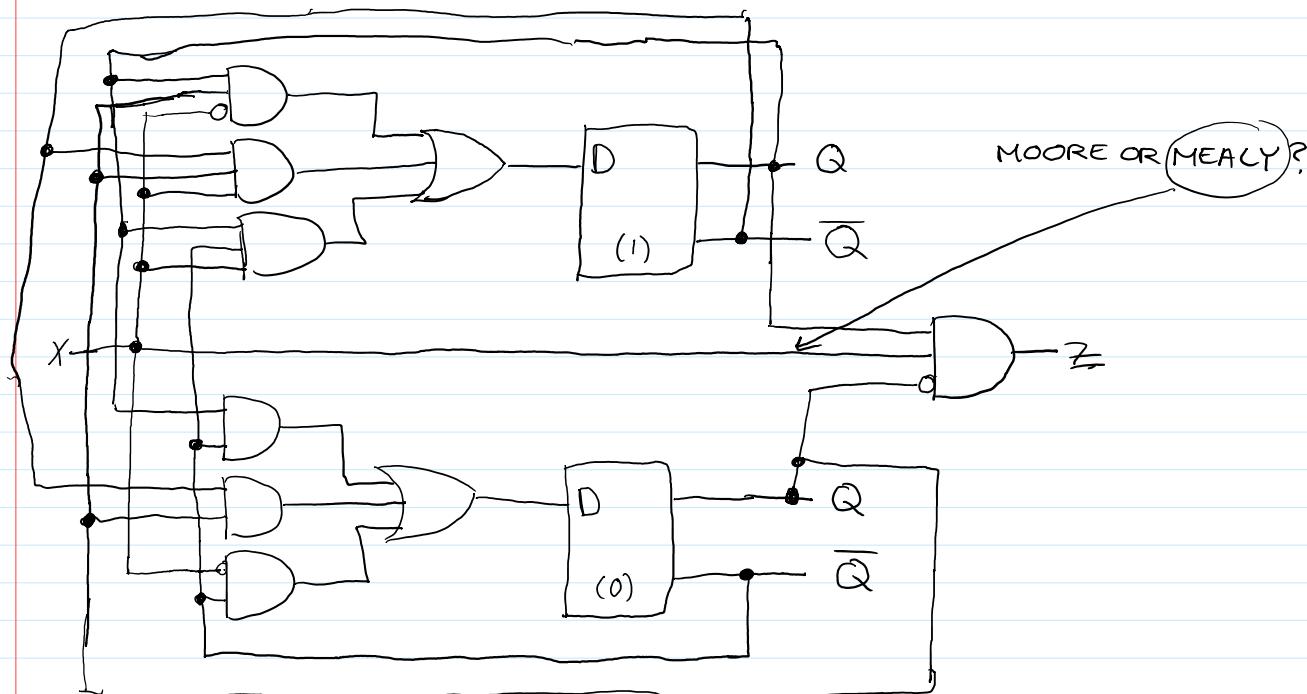
x	$q_1 q_0$	00 01 11 10
0	0	0 0 1 0
1	1	0 1 0 1

$D_1 = \overline{x} q_1 q_0 + \overline{x} \overline{q}_1 q_0 + x q_1 \overline{q}_0$
 $= \overline{x} q_1 q_0 + x \cdot (q_1 \oplus q_0)$

$Q_0 = D_0 = \sum_{q_1 q_0} \{00, 11\}$

x	$q_1 q_0$	00 01 11 10
0	0	1 1 0 0
1	1	1 0 0 1

$D_0 = q_1 \overline{q}_0 + \overline{q}_1 q_0 + \overline{x} \overline{q}_0$



0101 DETECTOR w/ S.R. F.F.s

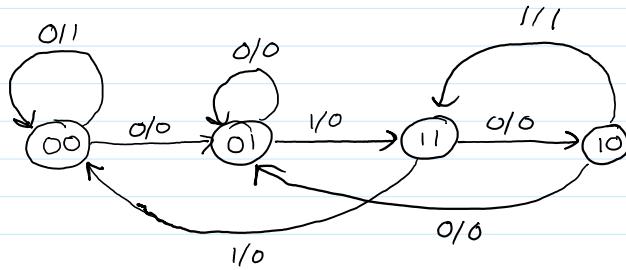
EXCITATION TABLE FOR SR

Q	Q^+	S	R
0	0	0	-
0	1	1	0

- = don't care for d

Q	Q^+	S	R
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

- = DON'T CARE OR d



0101 STATE TABLE W/ COLUMNS FOR SR

PRESENT STATE	NEXT STATE						NEXT STATE					
	X = 0			X = 1			X = 0			X = 1		
$q_1\ q_0$	Q_1	Q_0	S_1	R_1	S_0	R_0	Q_1	Q_0	S_1	R_1	S_0	R_0
0 0	0	1	0	-	1	0	0	0	0	-	0	-
0 1	0	1	0	-	-	0	1	1	1	-	-	0
1 1	1	0	-	0	0	1	0	0	0	1	0	1
1 0	0	1	0	1	1	0	1	1	-	0	1	0

NEED 4 K-MAPS FOR THE TWO SETS & TWO RESETS

$$S_1 = \begin{array}{c} \begin{array}{c} q_1 \\ q_0 \end{array} \\ \times \\ \begin{array}{ccccc} 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 0 & - & 0 \\ 1 & 0 & 1 & 0 & - \end{array} \end{array}$$

$$S_1 = x\bar{q}_1 q_0$$

$$R_1 = \begin{array}{c} \begin{array}{c} q_1 \\ q_0 \end{array} \\ \times \\ \begin{array}{ccccc} 00 & 01 & 11 & 10 \\ \hline 0 & - & - & 0 & 1 \\ 1 & - & - & 1 & 0 \end{array} \end{array}$$

$$R_1 = x \oplus q_1 \oplus q_0$$

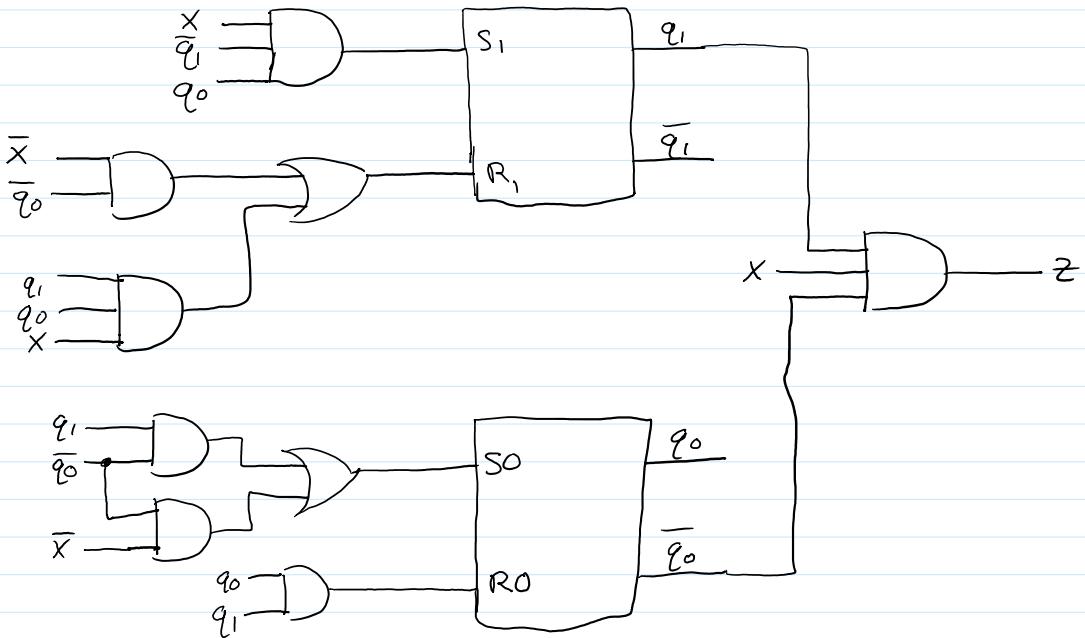
$$S_0 = \begin{array}{c} \begin{array}{c} q_1 \\ q_0 \end{array} \\ \times \\ \begin{array}{ccccc} 00 & 01 & 11 & 10 \\ \hline 0 & 1 & - & 0 & 1 \\ 1 & 0 & - & 0 & 1 \end{array} \end{array}$$

$$S_0 = q_1 \bar{q}_0 + \bar{x} \bar{q}_0$$

$$R_0 = \begin{array}{c} \begin{array}{c} q_1 \\ q_0 \end{array} \\ \times \\ \begin{array}{ccccc} 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 0 & - & 0 \\ 1 & - & 0 & 1 & 0 \end{array} \end{array}$$

$$R_0 = q_1 q_0$$

0101 MACHINE CKT WITH SR FF



D FF DESIGN FROM PREVIOUS EXAMPLE USED 8 GATES W/ 21 INPUTS

SR FF DESIGN USED 8 GATES WITH 18 INPUTS

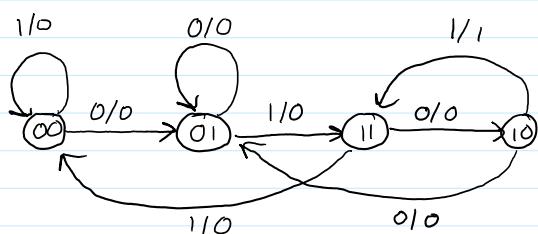
0101 DETECTOR USING JKFF

EXCITATION TABLE FOR JKFF

Q	Q^+	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

- DONT CARE OR d

NOTE: JKFF HAS MORE DON'T CARE
LESS "0"



0101 STATE TABLE W/ COLUMNS FOR JK

PRESENT STATE	NEXT STATE						NEXT STATE						
	$x = 0$			$x = 1$									
q_1	q_0	Q_1	Q_0	J_1	K_1	J_0	K_0	Q_1	Q_0	J_1	K_1	J_0	K_0

STATE	$x = 0$							$x = 1$						
	Q_1	Q_0	J_1	K_1	J_0	K_0	Q_1	Q_0	J_1	K_1	J_0	K_0		
0 0	0	1	0	-	1	-	0	0	0	-	0	-		
0 1	0	1	0	-	-	0	1	1	1	-	-	0		
1 1	1	0	-	0	-	1	0	0	-	1	-	1		
1 0	0	1	-	1	1	-	1	1	-	0	1	-		

$$J_1 = \begin{array}{|c|c|c|c|} \hline & 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 0 & - & - \\ \hline 1 & 0 & 1 & - & - \\ \hline \end{array} \quad J_1 = x q_0$$

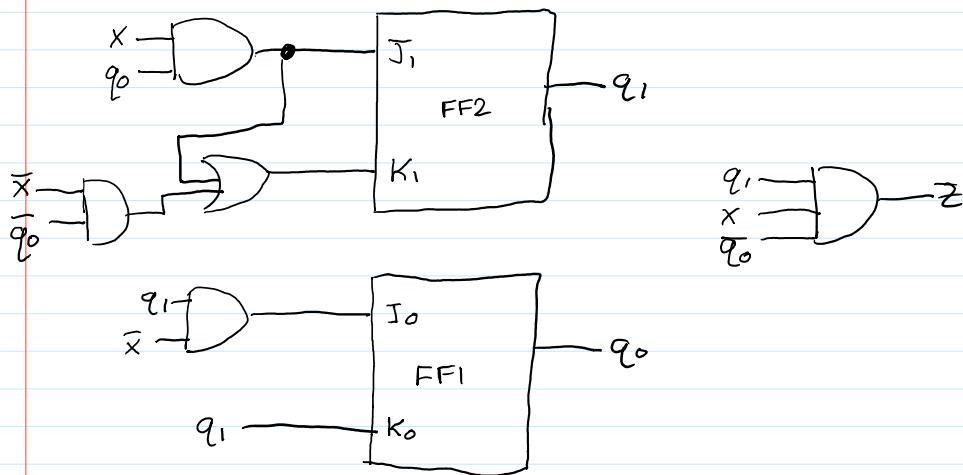
$$K_1 = \begin{array}{|c|c|c|c|} \hline & 00 & 01 & 11 & 10 \\ \hline 0 & - & - & 0 & 1 \\ \hline 1 & - & - & 1 & 0 \\ \hline \end{array} \quad K_1 = \bar{x} \bar{q}_0 + x q_0$$

$$J_0 = \begin{array}{|c|c|c|c|} \hline & 00 & 01 & 11 & 10 \\ \hline 0 & 1 & - & - & 1 \\ \hline 1 & 0 & - & - & 1 \\ \hline \end{array} \quad J_0 = q_1 + \bar{x}$$

$$K_0 = \begin{array}{|c|c|c|c|} \hline & 00 & 01 & 11 & 10 \\ \hline 0 & - & 0 & 1 & - \\ \hline 1 & - & 0 & 1 & - \\ \hline \end{array} \quad K_0 = q_1$$

NOTE: THE OUTPUT Z IS STILL THE SAME AS DFF

0101 MACHINE CKT w/ JKFF



NOTE: MUCH LESS HARDWARE IS NEEDED FOR JK THAN SR OR DFF

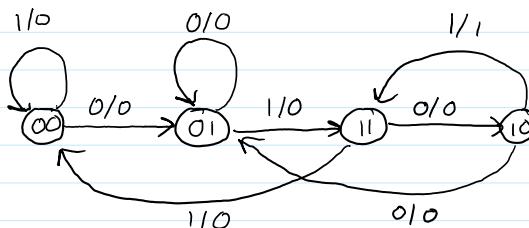
4 GATES & 8 INPUTS

0101 DETECTOR USING TFF

EXCITATION TABLE FOR TFF

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

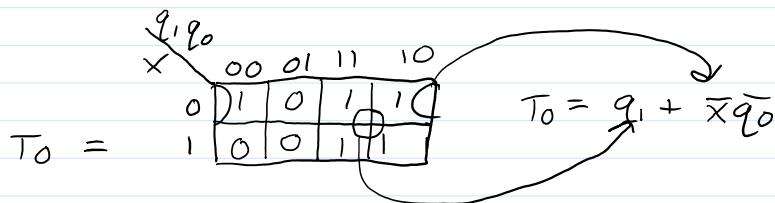
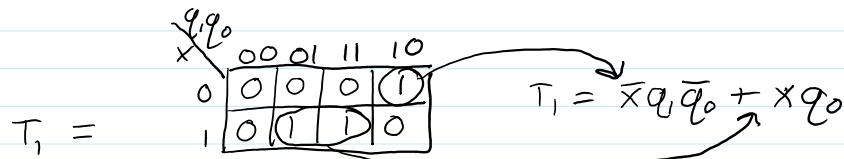
NOTE : THERE ARE NO DON'T CARES

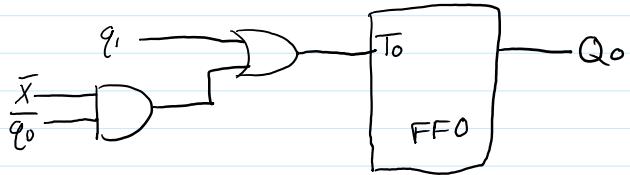
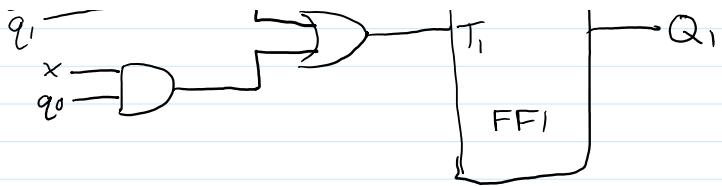


0101 STATE TABLE W/ COLUMN FOR T

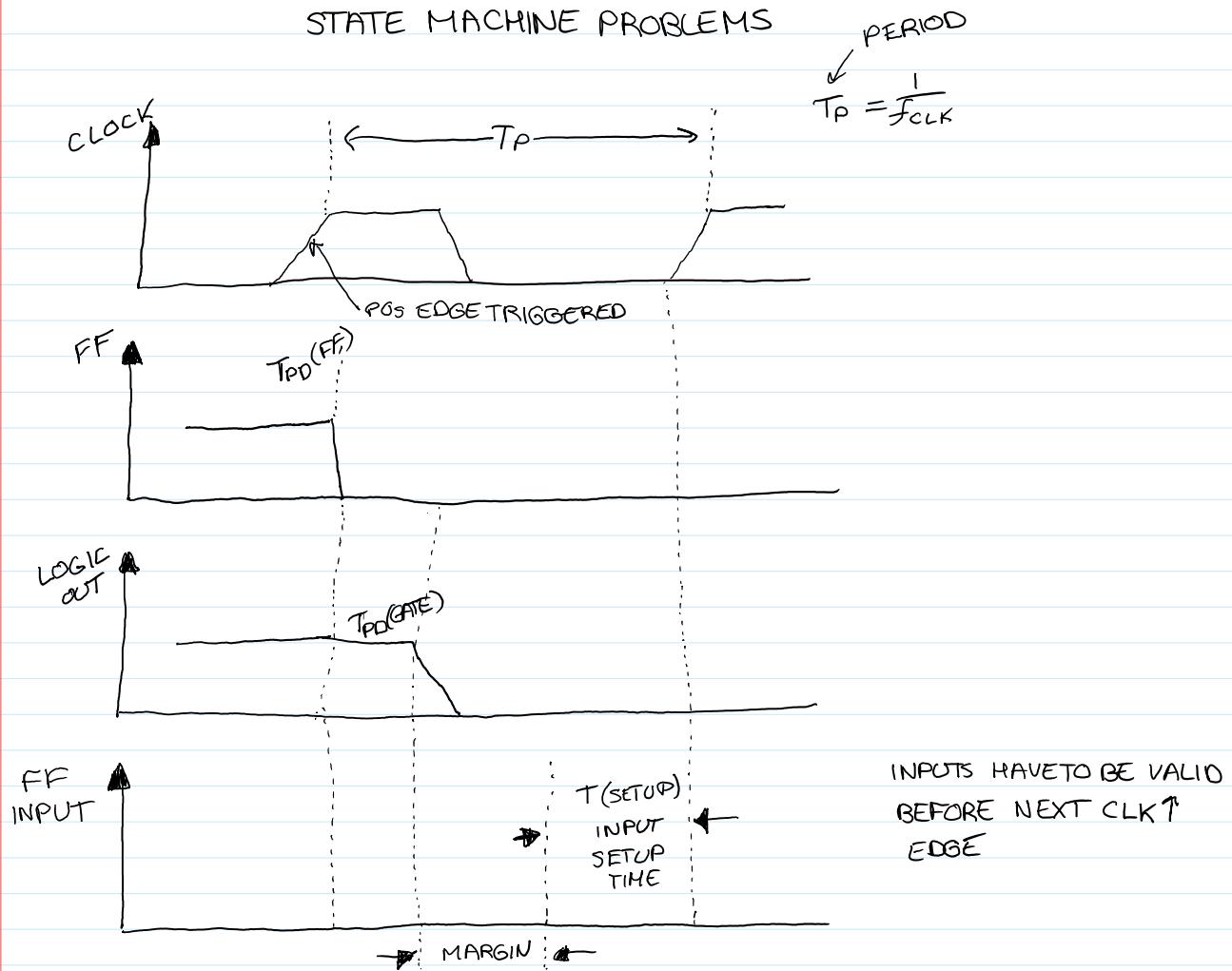
PRESENT STATE		NEXT STATE $X=0$				NEXT STATE $X=1$			
q_1	q_0	Q_1	Q_0	T_1	T_0	Q_1	Q_0	T_1	T_0
0	0	0	1	0	1	0	0	0	0
0	1	0	1	0	0	1	1	1	0
1	1	1	0	0	1	0	0	1	1
1	0	0	1	1	1	1	1	0	1

NEED 2 KMAPS FOR THE TWO T INPUTS





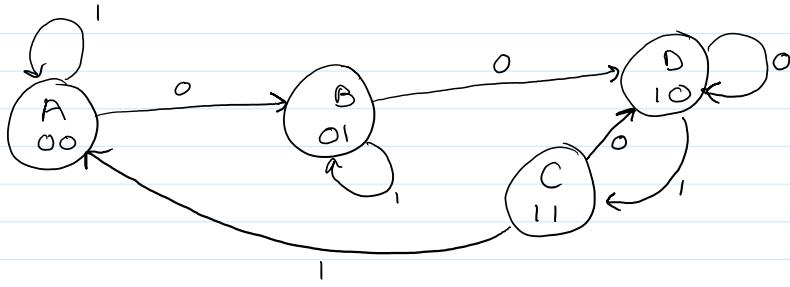
STATE MACHINE PROBLEMS



$$T_{MARGIN} = T_p - T_{PD(FF)} - T_{PD(GATE)} - T_{SETUP}$$

TIMING PROBLEMS "RACE" $T_{MARGIN} = \text{NEGATIVE}$

RACE OCCURS IN A SEQUENTIAL CKT WHEN MULTIPLE INTERNAL Q VARIABLES CHANGE WHEN GOING TO THE NEXT STATE

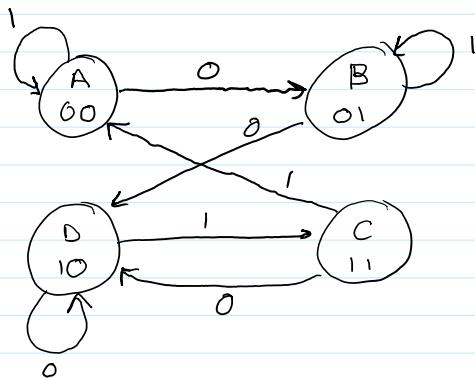


WHEN IN $C=11$ GOING TO $A=00$ ALSO $B \rightarrow D$

BOTH $Q_1 + Q_0$ CHANGE FROM 1 TO 0

IF Q_1 CHANGES BEFORE Q_0 THE MACHINE COULD GO TO B AND THEN
DEPENDING ON INPUT $X=1 \rightarrow B$ AND STAY THERE OR $X=0 \rightarrow D$ AND STAY
THERE

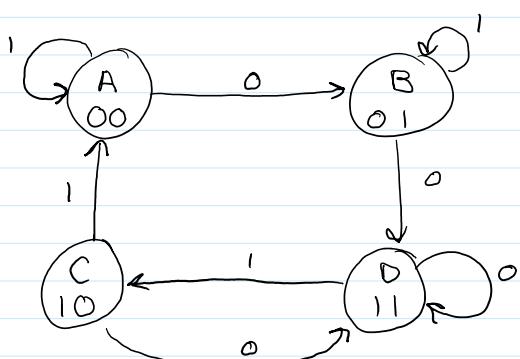
LETS REDRAW STATE DIAGRAM TO SHOW PROBLEM



NOTICE THAT C TO A AND B TO D PATHS CROSS

IF ONE FLIP FLOP IS SLOWER THAN OTHER BECAUSE ITS MARGIN WAS
NEGATIVE THE STATE MACHINE WOULD GOTO WRONG STATE AND
MAY FREEZE OR PERFORM A WRONG ACTION

THE FIX - REASSIGN STATE VARIABLES SO THAT PATHS DON'T CROSS



THERE ARE SEVERAL OTHER
ASSIGNMENTS THAT WILL
ALSO WORK

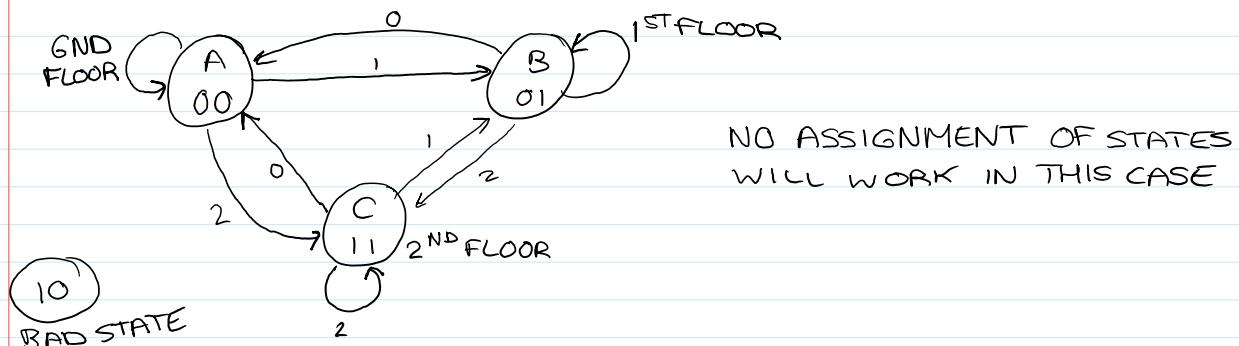
NOW, ONLY ONE F.F. WILL CHANGE AT A TIME

$$\begin{array}{l} C = 10 \\ D = 11 \end{array} \left\} \text{NEW ASSIGNMENTS} \right.$$

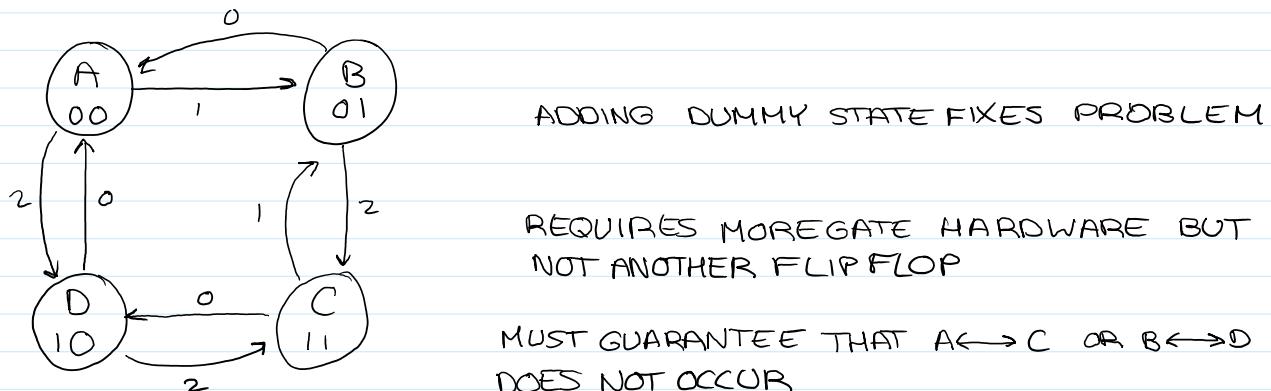
NOTE: GRAY CODE SEQUENCE $A \rightarrow B \rightarrow D \rightarrow C \rightarrow A$

A MACHINE MAY NOT BE ABLE TO HAVE THE CAPABILITY OF DOING THIS

TAKE THE 3 STATE CASE WITH A 3 FLOOR ELEVATOR

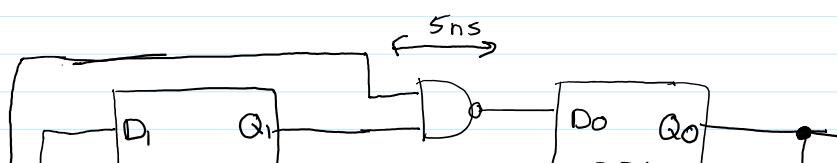


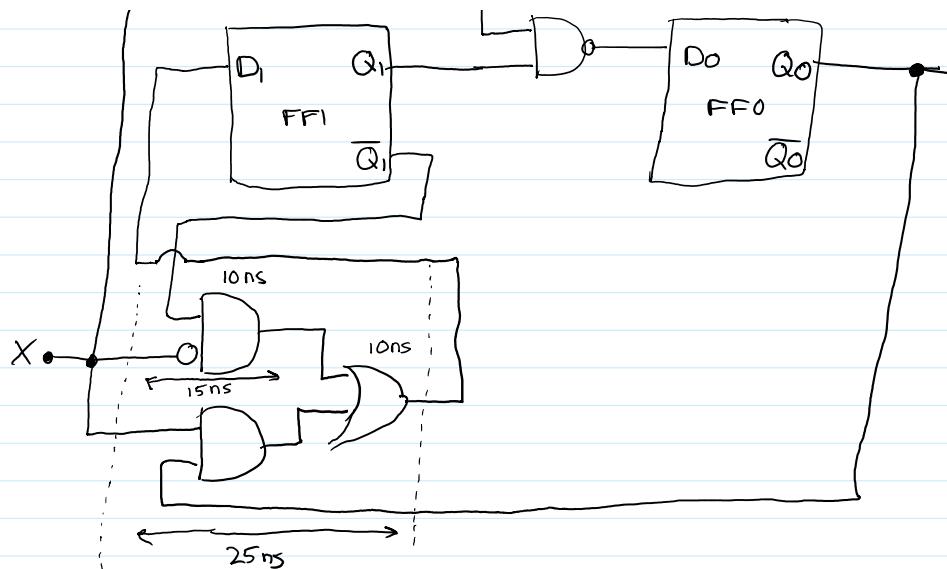
NEED TO ADD A STATE



WHAT CAUSES TIMING "RACE" PROBLEMS THE REASON WHY ONE FLIP FLOP FLIPS SOONER THAN ANOTHER

EXCITATION HARDWARE LOGIC EXAMPLE





D0 WILL RECEIVE INPUT IN 5ns; D1 IN 25ns

D0 20ns BEFORE D1. IF THIS WAS AN ASYNCHRONOUS MACHINE THERE COULD BE PROBLEMS. NEED TO FIX WITH A CLOCK BUT CLOCKS ARE COSTLY + POWER EXPENSIVE

ALSO RACE PROBLEMS CAN OCCUR IF ANY MACHINE HAS MULTIPLE INPUTS AND THE INPUTS DO NOT ARRIVE SIMULTANEOUSLY

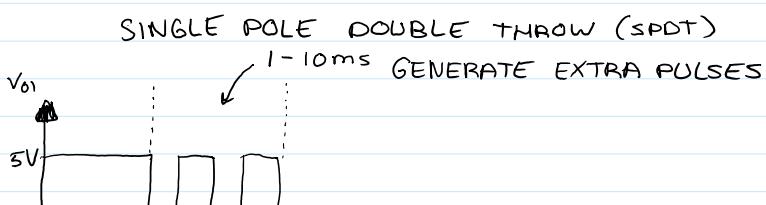
PRESENT STATE	NEXT STATE			
	X Y 0 0	X Y 0 1	X Y 1 1	X Y 1 0
A 00	00	00	10	01
B 01	01	00	11	01
C 11	11	11	10	00
D 10	10	11	00	01

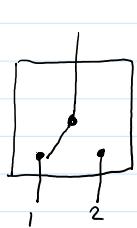
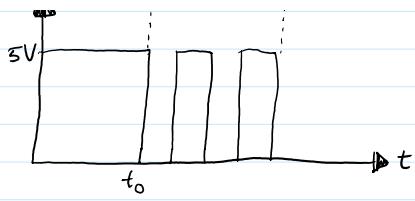
EX) IF WE ARE IN STATE D AND XY=11 SHOULD GO TO 00
 BUT IF X COMES IN FIRST THEN XY=10 AND THEN THE
 MACHINE WILL GO TO STATE B, 01, WHEN Y CATCHES
 UP THEN XY=11 AND MACHINE ENDS UP IN C. WRONG STATE

SWITCH DEBOUNCE FOR CLK

WHEN FLIPPING A SWITCH OR PUSHING A BUTTON IT FEELS INSTANTANEOUS HOWEVER THERE ARE MOVING PARTS

WHEN SWITCH IS CLOSED, TWO CONTACTS SEPARATE AND CONNECT TYPICALLY 10-100 TIMES OVER A PERIOD OF 1ms

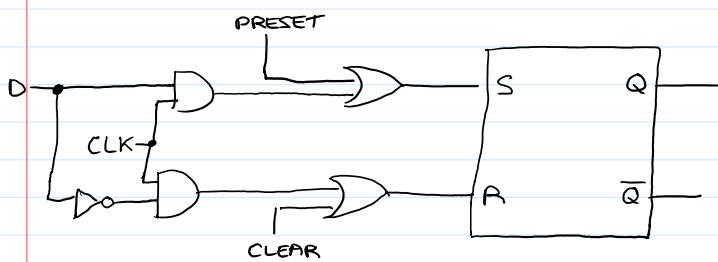




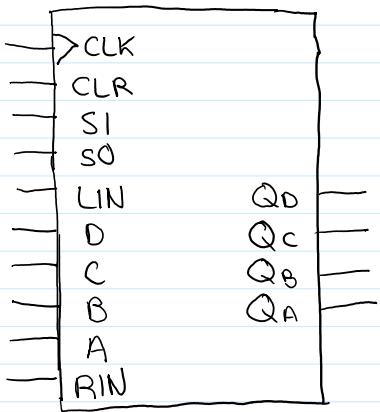
CALL $V_{O1} = Q$

D FLIP FLOP w/ PRESET AND CLEAR

\overline{PR}	\overline{CLR}	CLK	D	Q	\overline{Q}
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	X	X
1	1	↑	1	1	0
1	1	↑	0	0	1
1	1	0	X	Q_0	\overline{Q}_0



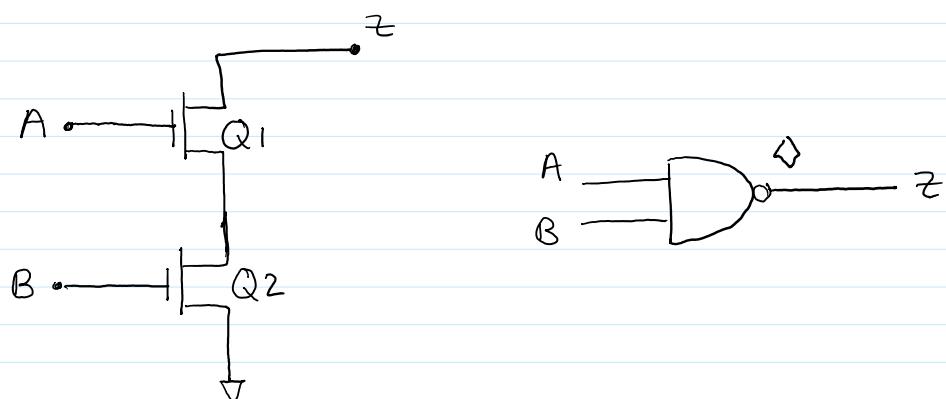
74x194 SHIFT REGISTER



FUNCTION	INPUTS			NEXT STATE			
	CLR	S1	S0	QA	QB	QC	QD
CLEAR	1	X	X	0	0	0	0
HOLD	0	0	0	QA	QB	QC	QD
SHIFT RIGHT	0	0	1				
SHIFT LEFT	0	1	0				
LOAD	0	1	1	A	B	C	D

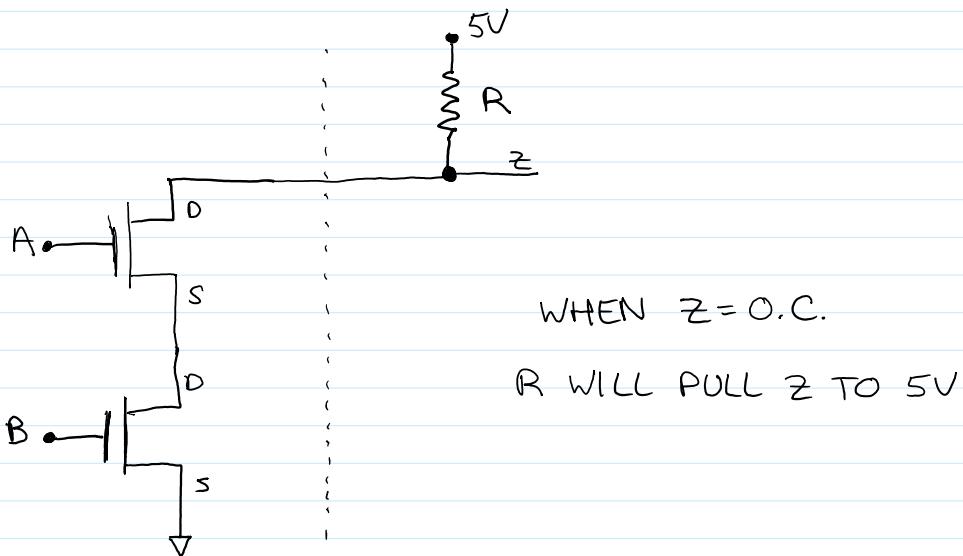
GATES WITH SPECIAL CHARACTERISTICS

OPEN DRAIN (COLLECTOR) NAND GATE



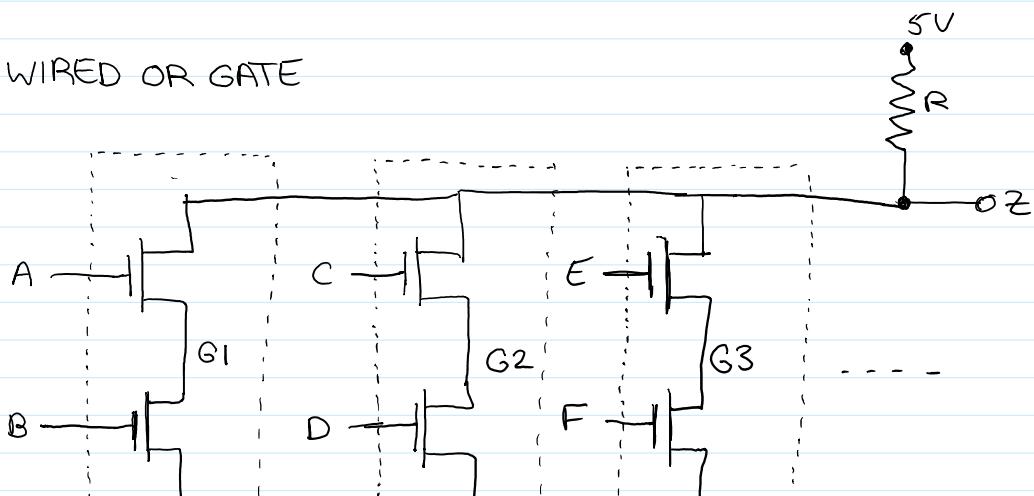
A	B	Q ₁	Q ₂	Z
0	0	OFF	OFF	O.C.
0	1	OFF	ON	O.C.
1	0	ON	OFF	O.C.
1	1	ON	ON	GND

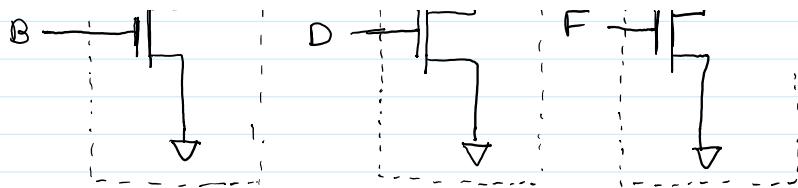
USED WITH A PULL UP RESISTOR



ONLY WHEN $A = B = 5V$ $Q_1 = Q_2 = \text{ON} \Rightarrow Z = 0V$

OPEN DRAIN GATE WIRED TOGETHER

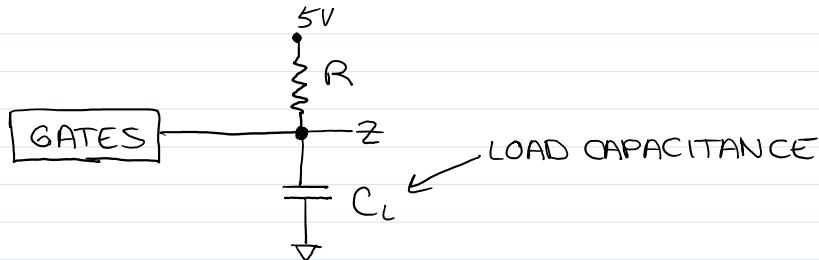




$$Z = AB + CD + EF + \dots$$

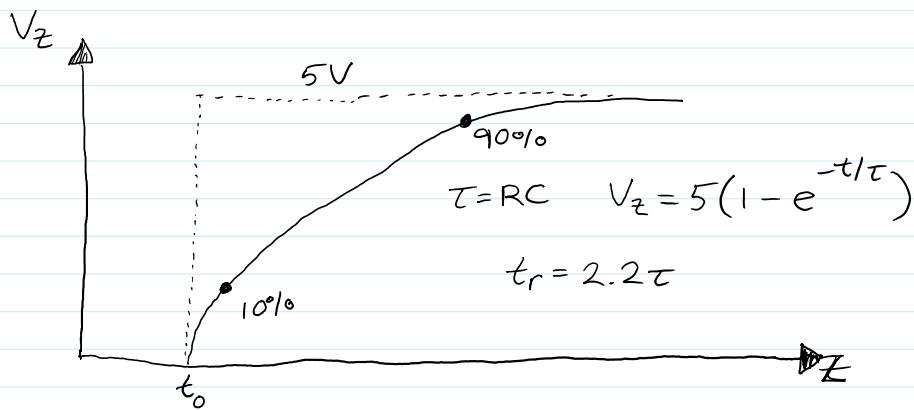
DISADVANTAGE SLOW RISE TIMES DUE TO R PULL UP

R IS LARGER THAN P CHANNEL DEVICE



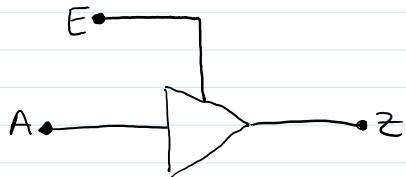
WHEN GATES GO OPEN CKT @ t_0

THE RESISTOR WILL CHARGE CAPACITOR EXPONENTIALLY



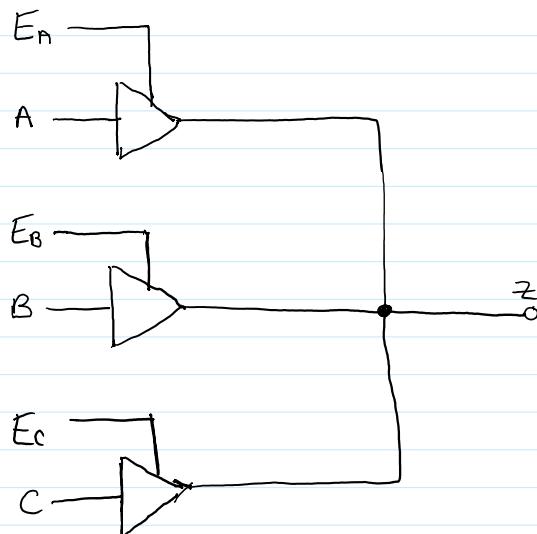
DASHED LINE PULL UP EMPLOYING P-CHANNEL FET

TRI STATE BUFFER



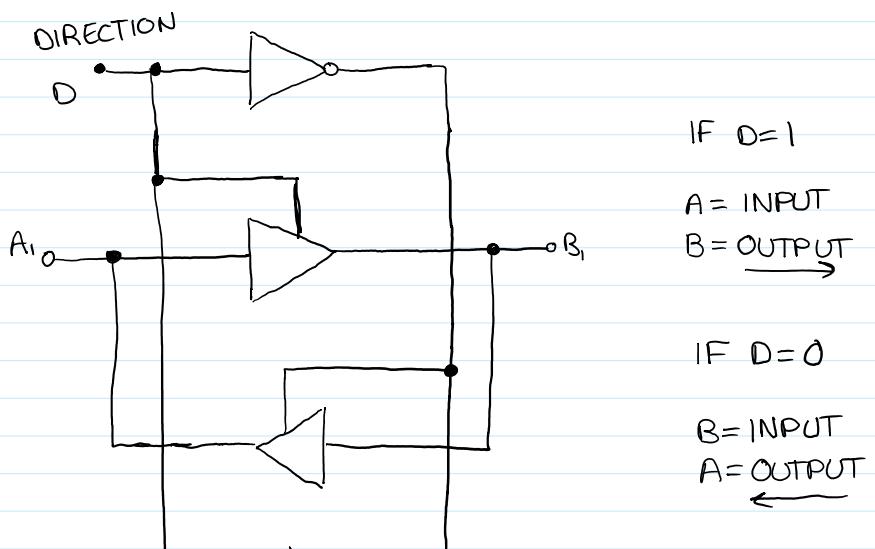
E	A	Z
0	0	O.C.
0	1	O.C.
1	0	0
1	1	1

ADVANTAGE: TRI STATE BUFFERS CAN BE BUSSSED TOGETHER
IF AND ONLY IF THE SYSTEM CAN GUARANTEE
 ONLY ONE GATE WILL BE ACTIVE AT A TIME



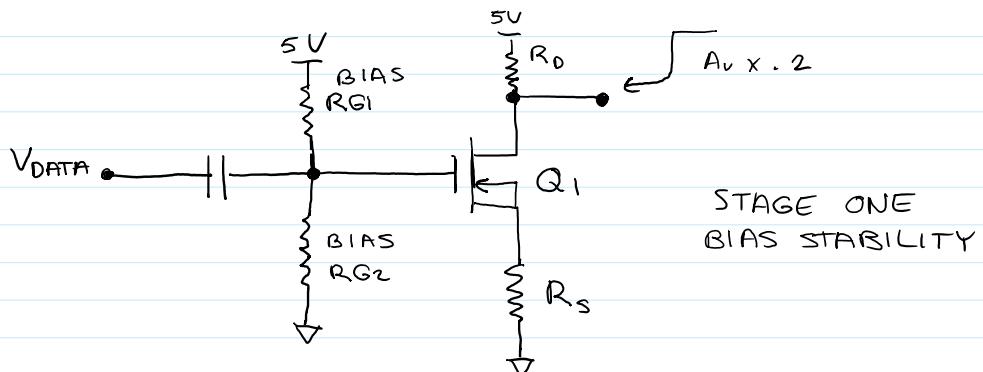
ONLY ONE BUFFER IS ENABLED AT A TIME

TRANSCIVER BI DIRECTIONAL





SMALL AMPLIFIER COMMON SOURCE



$$A_v \approx \frac{R_L}{R_s}$$

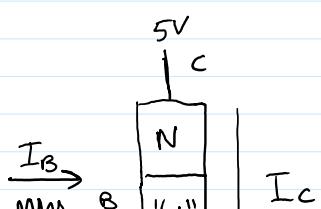
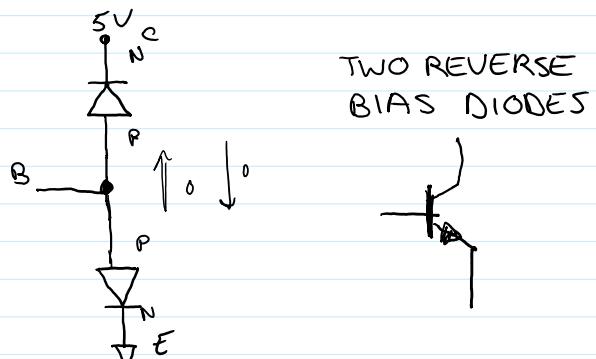
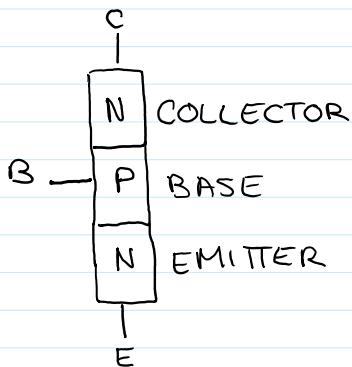
$$R_L = R_{D1} \parallel R_D$$

$$R_D = R_{D1} = R_{DS(\text{ON})}$$

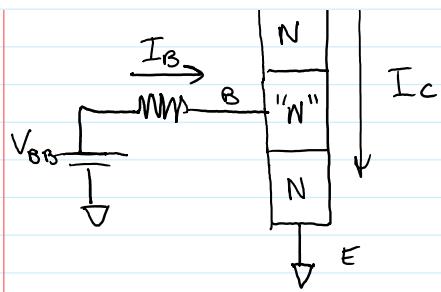
ADD SECOND STAGE FOR ADDITIONAL

$$A_{V_{\text{TOTAL}}} = A_{V_1} \times A_{V_2} \approx 20$$

BIPOLAR JUNCTION TRANSISTOR NPN



THE LARGER THE I_B



THE LARGER THE I_B

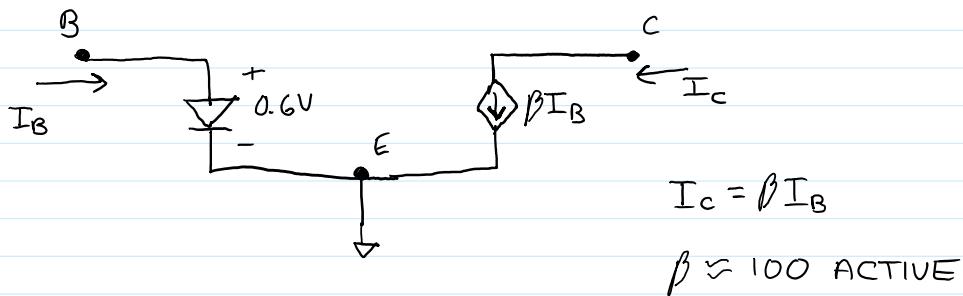
THE MORE CONDUCTIVE THE MIDDLE REGION

THE LARGER I_C

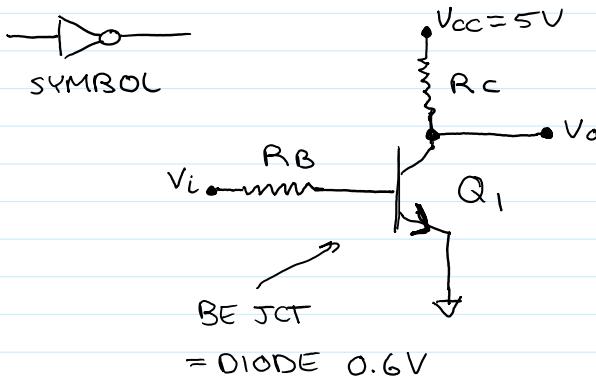
$$I_C = \beta I_B$$

$\beta \approx 100$
MULTIPLIER

MODEL OF ON BJT



TRANSISTOR (BJT) INVERTOR



WHEN $V_i < 0.6 \text{ V}$ $I_B = 0$ $Q_1 = \text{OFF}$

$I_C = 0$ $V_o = 5 - 0 \text{ V} \cdot R_C = 5 \text{ V}$ "1"

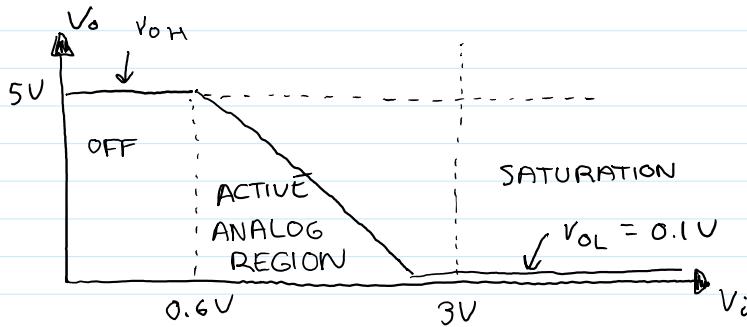
WHEN $V_i > 0.6 \text{ V}$ SAY $V_i \approx 3 \text{ V}$

$$I_B = \frac{V_i - V_{BE(\text{ON})}}{R_B} = \frac{3 - 0.6 \text{ V}}{R_B}$$

$$I_c = \beta I_B \quad \text{UNTIL LIMIT}$$

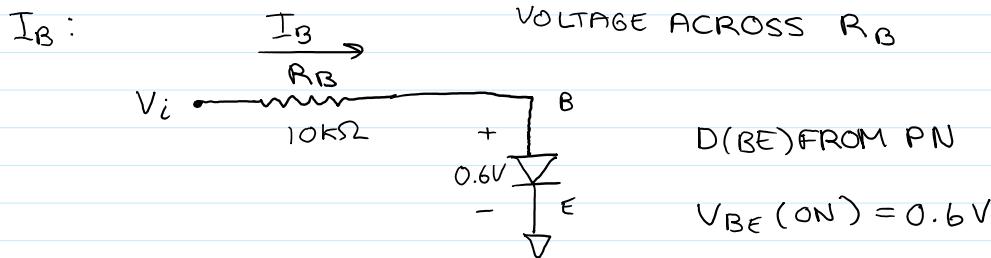
$$I_c (\text{SATURATION}) = \frac{5 - V_{CE}(\text{GATE})}{R_C} = \frac{5 - 0.1}{R_C}$$

TRANSFER CURVE

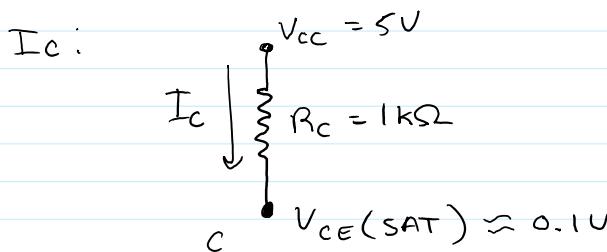


EX FIND CURRENTS IN INVERTOR WHEN Q=SAT

$$V_{CE} = 0.1V \quad V_i = 4V$$



$$I_B = \frac{V_{RB}}{R_B} = \frac{4V - 0.6V}{10k\Omega} = 0.34mA$$



$$I_c (\text{SAT}) = \frac{5 - 0.1}{1000} = 4.9mA \quad [\text{NOT } 0.34mA]$$

$$\beta_{\text{FORCED}} = 4.9 / 0.34 = 15$$

AS LONG AS $\beta > 15$ Q WILL BE SATURATED

DETERMINE V_{iH} FOR $\beta = 100$

FOR $V_O = V_{OL} = 0.1V$

$$I_C = 4.9 \text{ mA}$$

$$I_B = \frac{4.9 \text{ mA}}{100} = 49 \mu\text{A}$$

$$V_i = 0.6V + (49 \mu\text{A})(10k\Omega) = 1.09V = V_{iH}$$

RECALL $\beta = \frac{I_C}{I_B}$ ACTIVE $V_{CE} > 0.1V$

$$\beta_{FORCED} = \frac{I_C (\text{SAT})}{I_B} \quad V_{CE} \approx 0.1V - 0.2V$$

