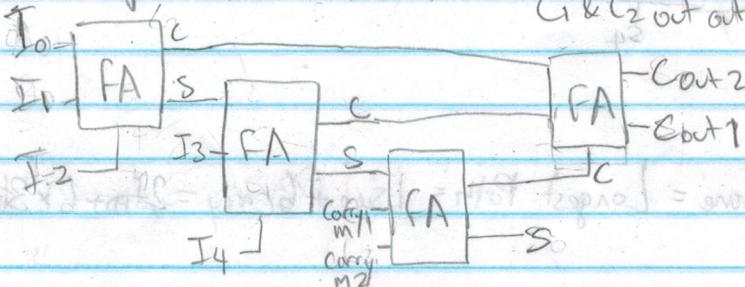


Collaborators:

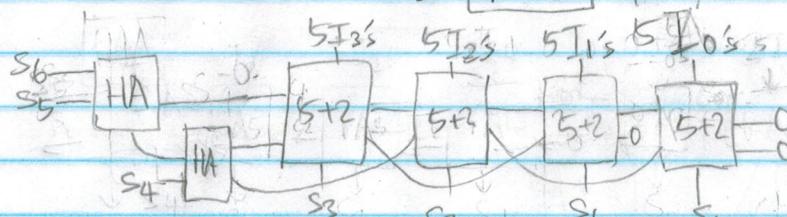
Haoyu Yu, Kyle Wong.

Online Resources:

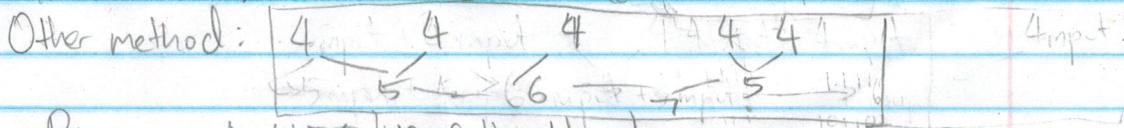
## 1. Five Input Adder

(Assumes  $C_1$  &  $C_2$  are one's place) $C_1$  &  $C_2$  out out carry and two's place

$$\text{Propagation Delay} = 4 \times \text{FA} = 4 \times 3 + \text{pd} = 12 + \text{pd}$$

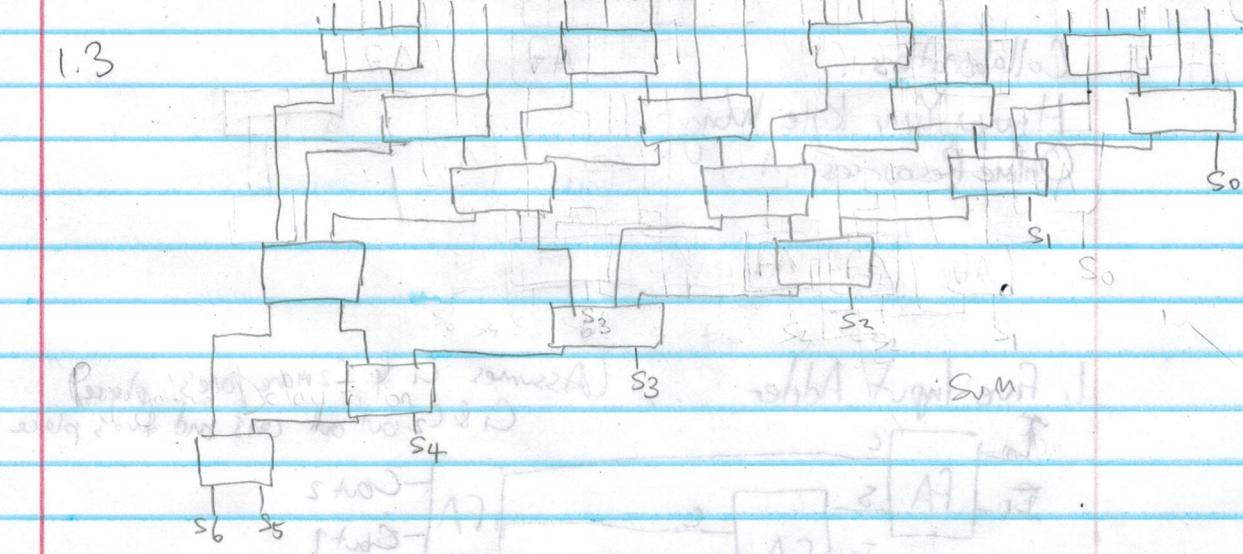
1.2 The sum should have  $4+3=7$  bits.

$$\text{Propagation Delay} = 2\text{HA} + 4 \times \text{FA} = 6\text{pd} + 4 \times 12\text{pd} = 54 + \text{pd}$$

Full Adders:  $1 + 4 \times 4 = 18$  full adders totalRequires  $4+4+5+6=19$  full addersTime is longest path:  $4 \times 3 + \text{pd} + 5 \times 3 + \text{pd} + 6 \times 3 + \text{pd} = 45 + \text{pd}$

1.3

$A_3\ B_3\ C_3\ D_3\ E_3\ A_2\ B_2\ C_2\ D_2\ E_2\ A_1\ B_1\ C_1\ D_1\ E_1\ A_0\ B_0\ C_0\ D_0\ E_0$



$$\text{Propagation Time} = \text{Largest Path} = 1 \text{ sum} + 6 \text{ carry} = 2t_{PD} + 6 \times 3t_{PD} = 20t_{PD}$$

$$[out\ S1] = [out\ S0] \oplus [in\ P1] = AA \times P = \text{full adder}$$

$$[out\ S2] = S1 + P1 \quad \text{and block uses out\ S1}$$



$$[out\ S3] = [out\ S2] + [in\ P2] = BA7 \times H \quad \text{full adder}$$

$$[out\ S4] = [out\ S3] + [in\ P3] = A \times P + P \quad \text{full adder}$$

Full adder:  $P = A \oplus B \oplus C$ ,  $S = A \times B + A \times C + B \times C$

$$[out\ S1] = [out\ P1] = A \times B + A \times C + B \times C$$

$$[out\ S2] = [out\ P2] = A \times B \times C + A \times B \times D + A \times C \times D + B \times C \times D$$

2.1 E=0

$A_1 A_0$	00	01	11	10
$B_1 B_0$	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	0	1	0	0
10	1	0	0	0

E=1

$A_1 A_0$	00	01	11	10
$B_1 B_0$	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	1	1	1	0
10	1	0	1	1

(a) S.S

Note: These  
are on both layers

Essential: Cover area that no other combination of  
implicants cover. Everything is essential here.

Boolean Expression:

$$A_1 B_1 + B_0 \bar{A}_1 \bar{A}_0 + \bar{A}_0 B_0 B_1 + \bar{A}_1 \bar{A}_0 E + B_0 \bar{A}_1 E + B_1 \bar{B}_0 E + \\ B_1 \bar{A}_0 E = \bar{A}_1 (B_1 + B_0 \bar{A}_0) + \bar{A}_0 B_0 B_1 + E (\bar{A}_1 (\bar{A}_0 + B_0) + B_1 (B_0 + \bar{A}_0))$$

Lenient:

Yes. Since the prime implicants are all essential, this  
minimal implementation is also lenient. It has no glitches.

$A_1 A_0$	00	01	11	10
$B_1 B_0$	00	01	11	10
00	X	0	0	0
01	1	X	0	0
11		1	X	1
10	1	1	0	X

Expression:

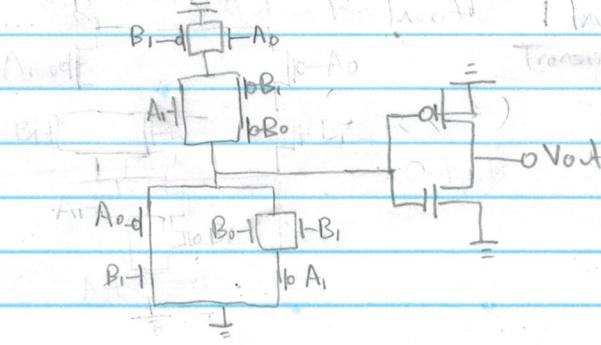
$$\bar{A}_0 B_1 + \bar{A}_1 B_0 + \bar{A}_0 B_0 = \bar{A}_1 (B_1 + B_0) + \bar{A}_0 (B_1) = 1$$

$$\text{or}, (A_1 | (B_1 \& B_0) \& A_0 | \bar{B}_1) = 0$$

Not lenient

or...

Circuit:



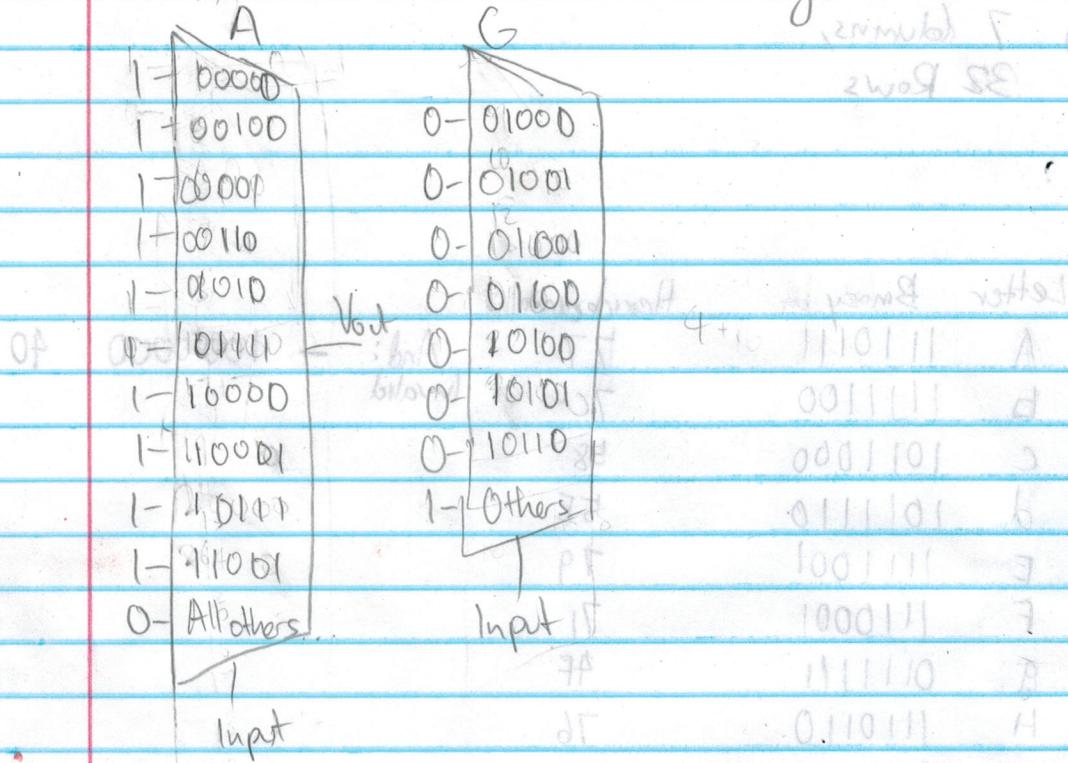


3.10 ROMS,  $\alpha = A_{\text{min}}/2A_{\text{max}}$ ,  $S^2 = 0.5/2$  for  $\sin F_0$  for  $S^2$

a) 7 columns,  
32 Rows

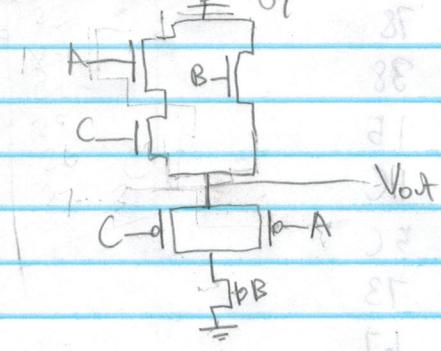
Letter	Binary	Hexadecimal	And:	
A	1110111	77	-	1000000
B	1111100	7C	invalid	
C	1011000	58		
D	1011110	5E		
E	1111001	79		
F	1110001	71	first	2000000
G	0111111	4F		
H	1110110	76		first
I	0000110	6		
J	0011110	3E		not next
K	1111000	78		
L	0111000	38		
M	0010101	15		
N	0011100	1C		
O	1011100	5C		
P	1100111	73		
Q	1100111	67		
R	1010000	50		
S	1101101	6D		
T	1000110	4		
U	0111110	3E		
V	00111100	1C		
W	0101010	2A		
X	1001001	49		
Y	1101110	6E		
Z	1011011	5B		

3.2 a) 7 muxes of size 32. Assuming A=0, B=10.2 and so on.



## 4. CMOS Technology

4.1



Q: Is the above a valid CMOS Gate? If so, write a Boolean expression for A.

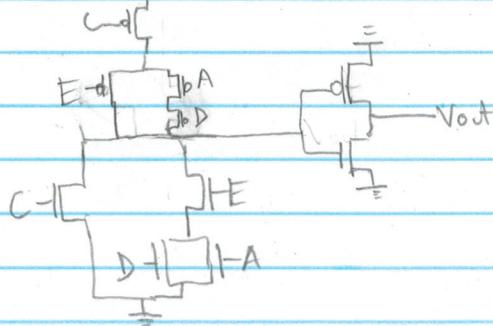
A: Yes, it's valid.

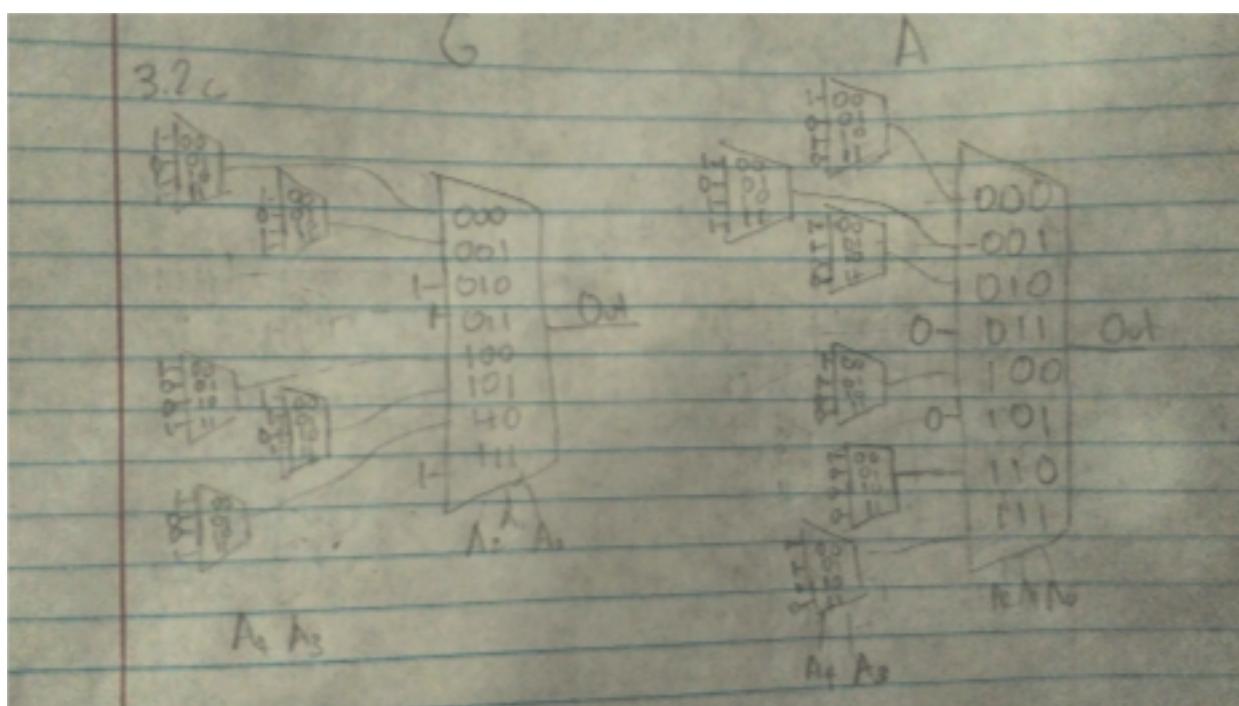
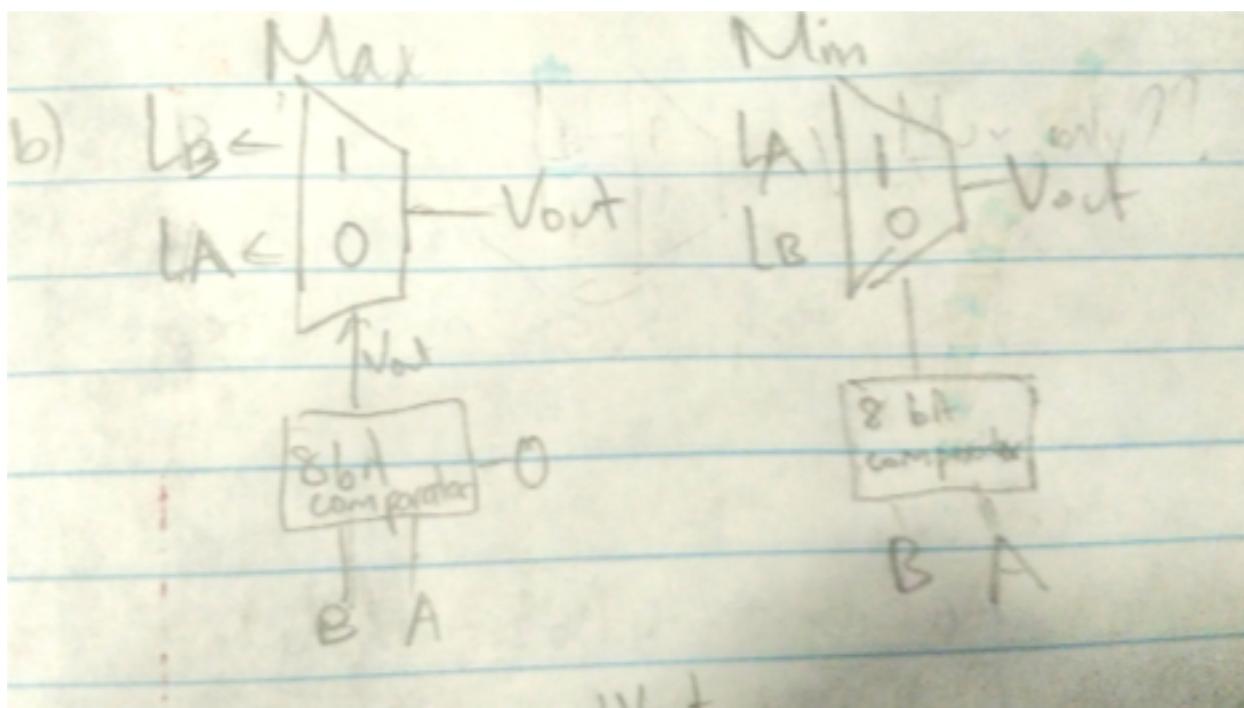
$$(A \wedge c) \mid B = P$$

4.2 Q Consider the expression  $F = (A + C + D) \& (E + C)$ ,  
 Simplify this expression and then implement it  
 using a CMOS gate and an inverter.

A: Simplified:  $C \& (E \& (A \mid D)) = F$

Inverted:  $\neg(C \& (\neg E \& (\neg A \& \neg D))) = F$





Mm Moves: (,

G-G, A-7