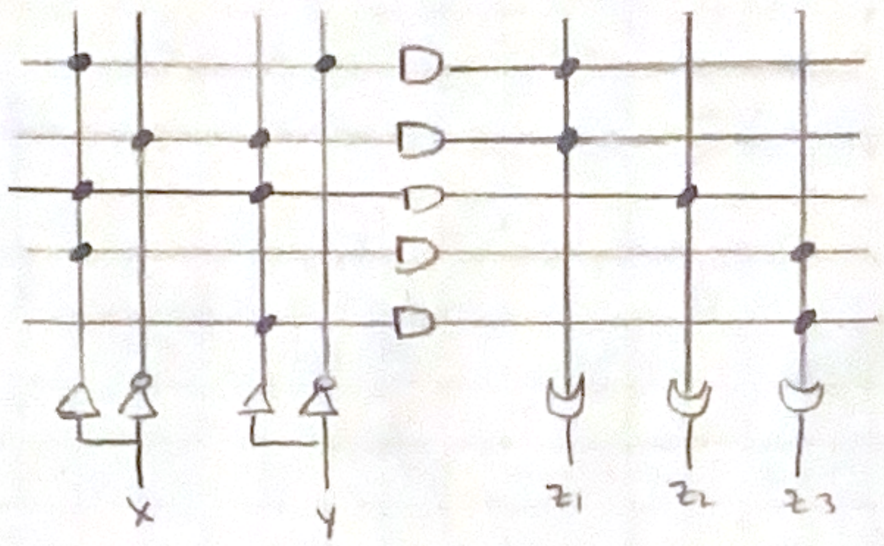
① PAL DEVICE PROGRAM THE DEVICE SO $z_1 = X \oplus Y$, $z_2 = XY$, $z_3 = X + Y$.

↳ $z_1 = x\bar{y} + \bar{x}y$



z_1

TO CONSERVE SPACE, IT WOULD BE NICE TO SHARE TERMS WITH EITHER z_2 OR z_3. LET'S SEE IF WE CAN GENERATE z_1 WITH XY OR $(X+Y)$. KNOWING WE CAN OR IN $x\bar{x}$ OR $y\bar{y}$ AS FILLER TERMS.

* $x\bar{x} = y\bar{y} = 0$

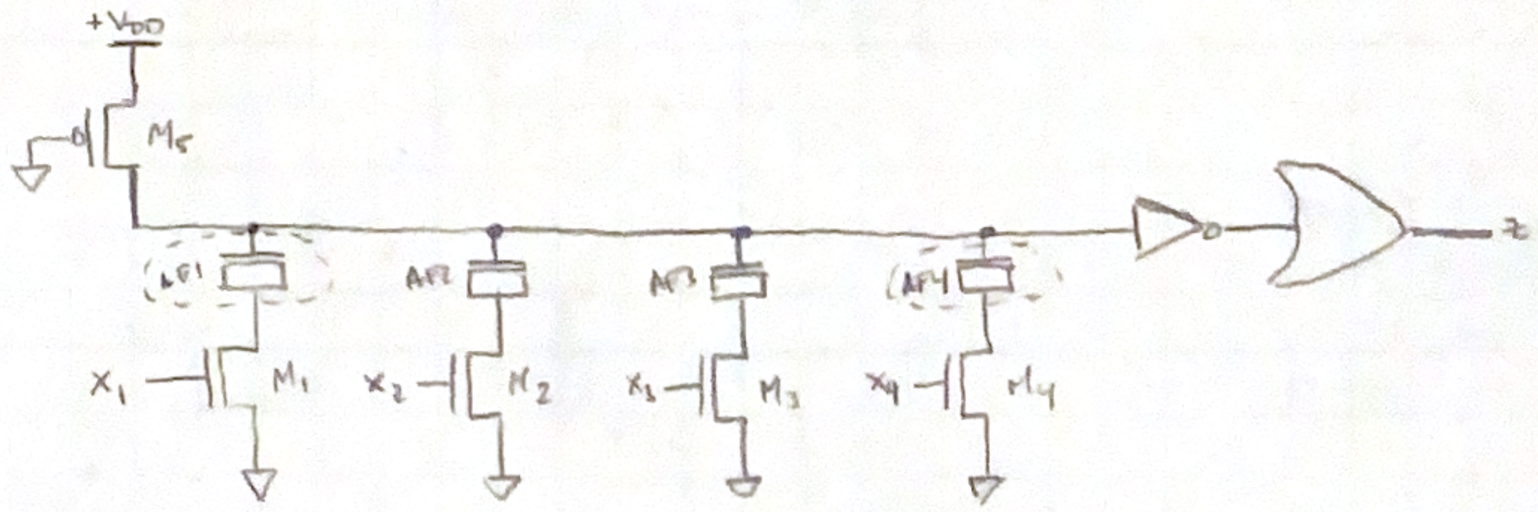
$$\Rightarrow z_1 = X \oplus Y = x\bar{y} + \bar{x}y$$
$$= x\bar{y} + \bar{x}y + x\bar{x} + y\bar{y} \quad (0+0=0)$$
$$= (x\bar{y} + x\bar{x}) + (\bar{x}y + y\bar{y})$$
$$= x(\bar{x}+\bar{y}) + y(\bar{x}+\bar{y})$$
$$= (x+y)(\bar{x}+\bar{y})$$
$$= \overline{(x+y)} + \overline{(\bar{x}+\bar{y})} \quad \text{DEMORGAN'S}$$
$$= \overline{\bar{x}\bar{y}} + \overline{x\bar{y}}$$
$$= xy + \overline{x\bar{y}}$$

... THIS DOESN'T WORK. NEED A 5TH AND GATE.

N. F. E.

(2) PAL PROGRAMMABLE OR PLANE

2.1) SHOW HOW TO IMPLEMENT CIRCUIT WITH ANTIFUSES. THEN SHOW HOW TO PROGRAM TO IMPLEMENT $Z = X_1 + X_4$.

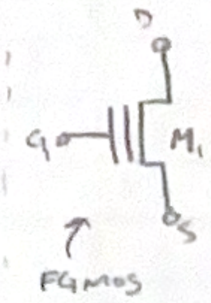


THIS CIRCUIT WORKS VIA A PULL-UP PMOS TYING ALL ANTIFUSES (AF1-AF4) TO +VDD; IF THE ANTIFUSES ARE INTACT, THEN THE DRAINS OF M1-4 ARE PULLED TO +VDD. WHEN THE "VISIBLE" NMOS FETS ARE BIASED AT THE GATE, THEN THE CHANNELS INVERT, SENDING THE OUTPUT TO GND. WE HAVE THE INVERTER PRESENT TO CORRECT FOR THAT INPUT STAGE INVERSION.

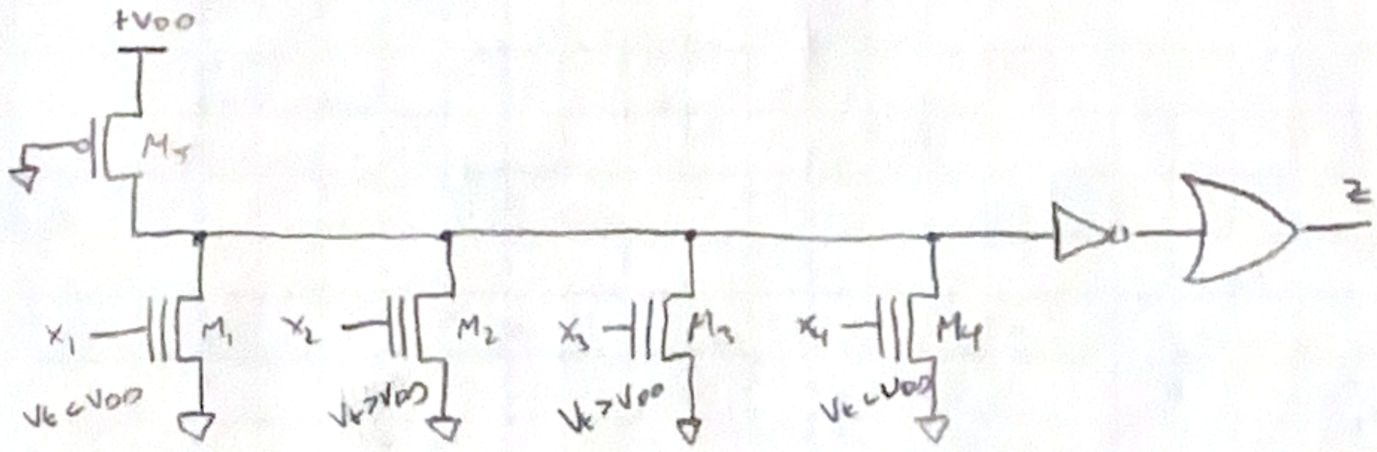
TO PROGRAM $Z = X_1 + X_4$, WE SIMPLY NEED TO BURN AF1 AND AF4 TO PROVIDE THE OUTPUT STAGE WITH THE SIGNALS FROM M_1 AND M_4. THE ANTIFUSES NEEDED TO BE BURNT ARE CIRCLED.

2.2) SHOW HOW TO IMPLEMENT USING EEPROM TECHNOLOGY. THEN SHOW HOW TO PROGRAM $Z = X_1 + X_4$.

USING EEPROM TECHNOLOGY, WE WILL USE MOSFETS (n+-type) WITH AN EXTRA FLOATING GATE. THIS GATE ALLOWS FOR CONFIGURABILITY OF THE DEVICE'S THRESHOLD VOLTAGE.



IF WE BIAS THE FGMOS'S TO GAIN A THRESHOLD VOLTAGE V_t GREATER THAN OUR CIRCUIT'S LOGIC VOLTAGE V_{DD}, THEN WE CAN DISABLE THE DEVICE.

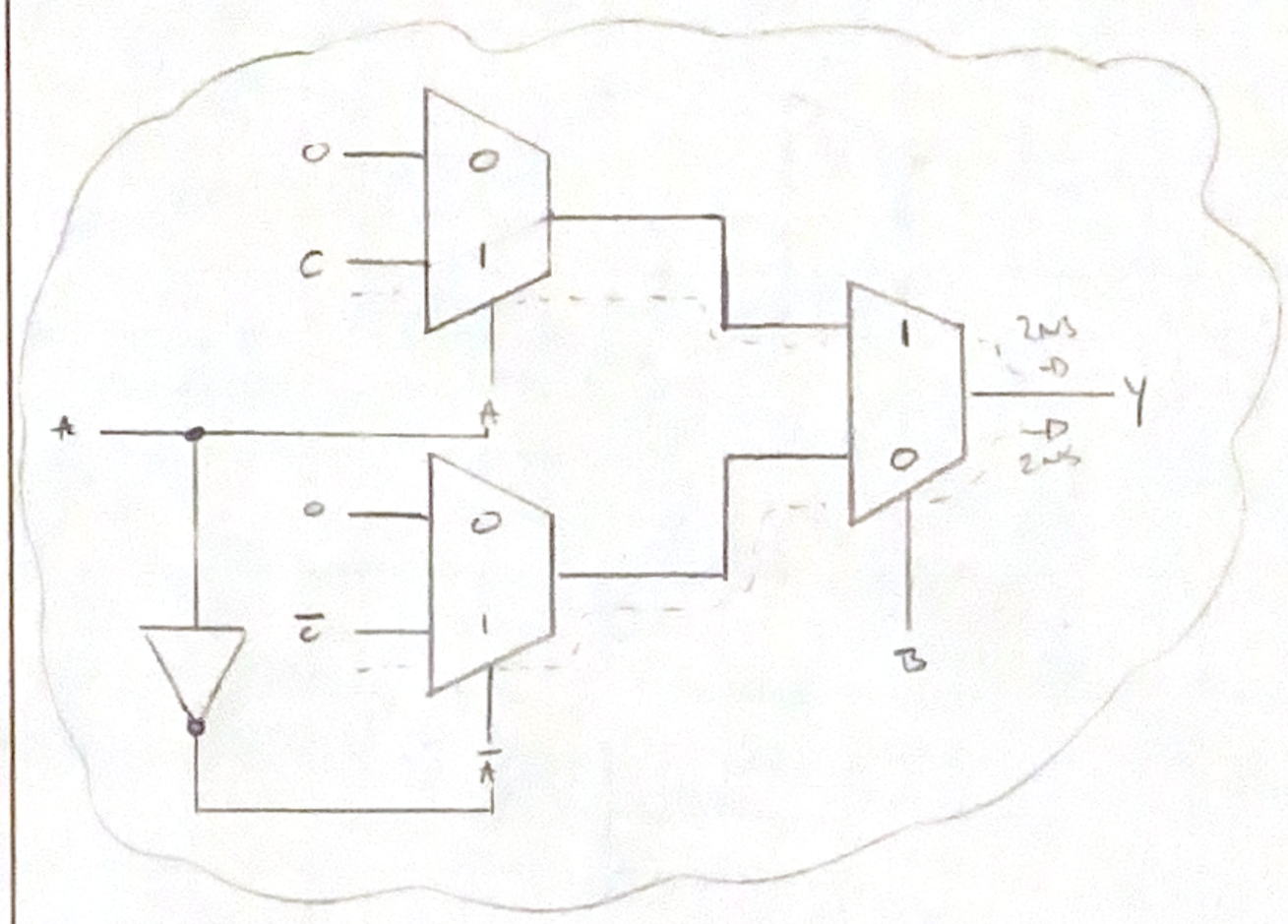


WHEN PROGRAMMING, WE WOULD DEFAULT HAVE $Z = X_1 + X_2 + X_3 + X_4$, SO WE NEED TO DISABLE M_2 & M_3. TO DO SO, WE WILL BIAS THEM TO INCREASE V_t SUCH THAT $V_t > V_{DD}$.

(3) IMPLEMENT $y = ABC + \bar{A}\bar{B}\bar{C}$ USING 2^{nd} 1 MUX. THE SIGNAL B ALWAYS ARRIVES LATE. MUX DELAY IS 1ns. WHAT IS THE LARGEST DELAY?

$$y = ABC + \bar{A}\bar{B}\bar{C}$$
$$= F(B=1)\,AC + F(B=0)\,\bar{A}\bar{C}$$

* CHOOSE $F(B=1,0)$ SINCE B IS LATE...



SINCE B IS ALWAYS LATE, I USE IT AS THE CONTROL SIGNAL FOR THE FINAL 2-TO-1 MUX.

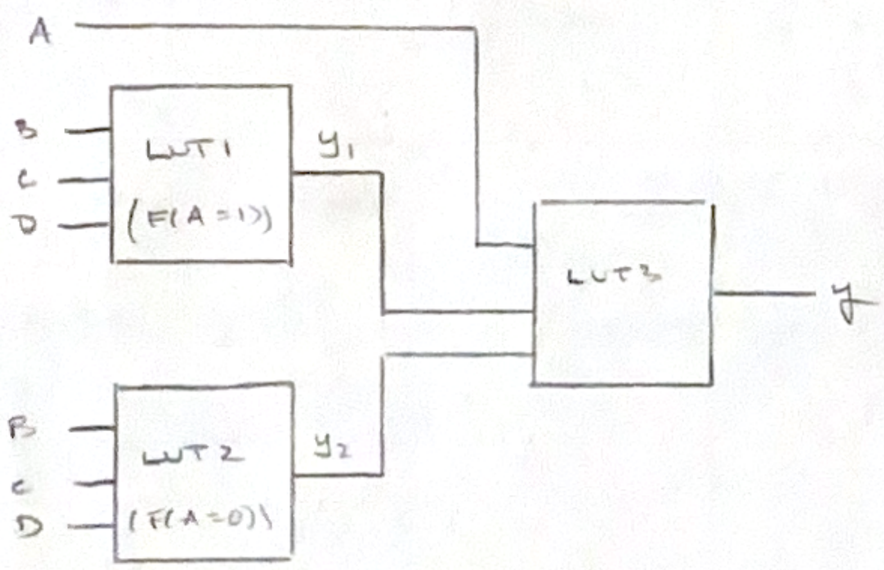
THE LONGEST DELAY IS THROUGH THE FIRST & SECOND LAYER OF THE MUXs, WHERE C & $\bar{C}$ PROPAGATE (→ NO DELAY ASSUMED).

THE LONGEST DELAY IS **2NS**

(4) IMPLEMENT $y = ABCD + \bar{A}\bar{B}\bar{C}\bar{D}$ USING 3-INPUT LOOK-UP TABLES (LUTs) WHICH EACH HAVE SIZE 8. BLOCK DIAGRAM & LIST LUT CONTENTS.

$$\Rightarrow y = A\underbrace{BCD}_{F(A=1)} + \bar{A}\underbrace{\bar{B}\bar{C}\bar{D}}_{F(A=0)} = A \cdot F(A=1) + \bar{A} \cdot F(A=0) = Ay_1 + \bar{A}y_2$$

* $y_1 = BCD$
* $y_2 = \bar{B}\bar{C}\bar{D}$



LUT1

B	C	D	y_1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

LUT2

B	C	D	y_2
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

LUT3

A	y_1	y_2	y
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	0