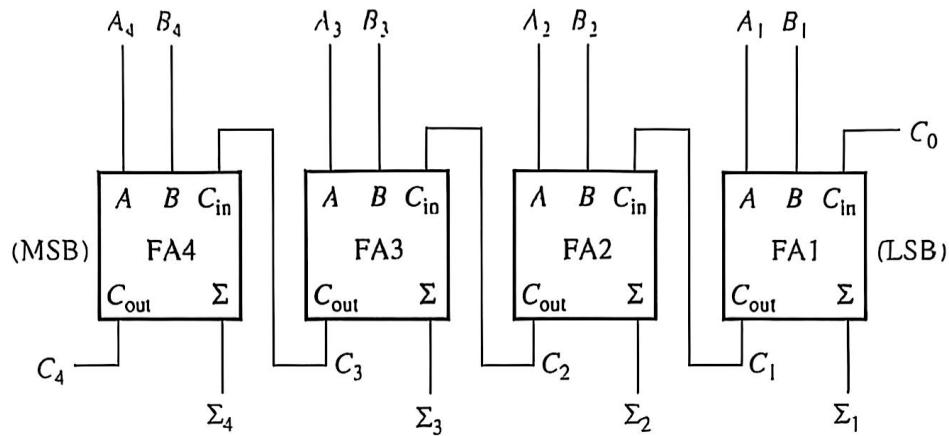


ECE2050 Homework 5

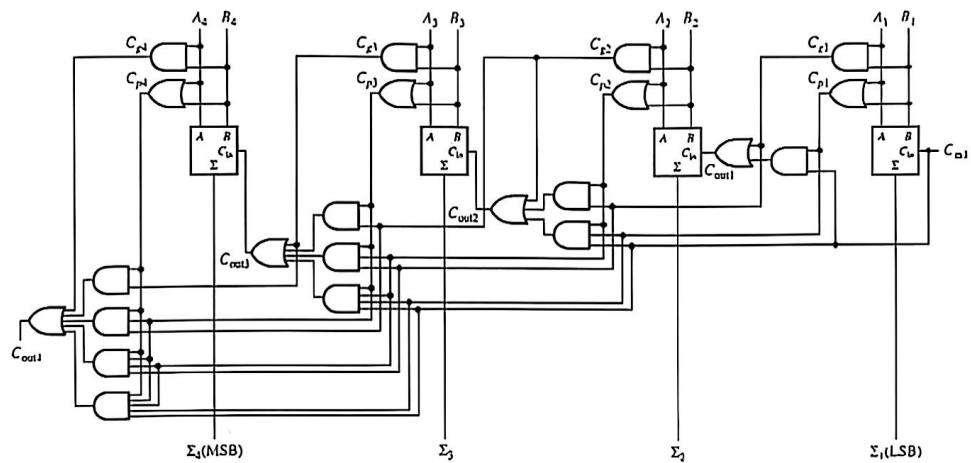
Due: March 20, 2025

Q1 What is the worst-case delay for the following types of 4-bit adders? Assume that each logic gate delay is 150 ps and that a full adder delay is 450 ps.

1. a 4-bit parallel adder.



2. a 4-stage look-ahead carry adder.



Q.

(1) As every full-adder delay is 450ps

$$t_{4\text{-bit parallel}} = 4 \times 450 \text{ ps} = 1800 \text{ ps}$$

(2) Here, we can treat the full-adder as a whole (450ps)

$$\Rightarrow t = 3 \times 150 + 450 \text{ ps} = 900 \text{ ps} .$$



扫描全能王 创建

Q2 Multiplication of unsigned binary numbers is similar to decimal multiplication but involves only 1's and 0's. The figure below compares multiplication in decimals and binary. In both cases, partial products are formed by multiplying a single digit of the multiplier with the entire multiplicand. The shifted partial products are summed to form the result.

230	multiplicand	0101
\times	multiplier	\times
42		0111
		0101
460	partial	0101
+ 920	products	0101
9660		0101
	result	+ 0000
		0100011

$$230 \times 42 = 9660$$

$$5 \times 7 = 35$$

An $N \times N$ multiplier multiplies two N -bit numbers and produces a $2N$ -bit result. The partial products in binary multiplication are either the multiplicand or all 0's. Multiplication of 1-bit binary numbers is equivalent to the AND operation, so AND gates are used to form the partial products.

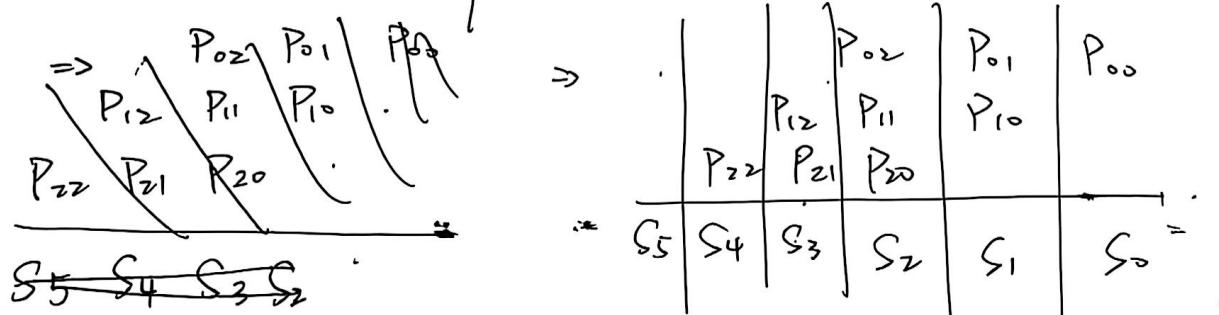
Please use 1-bit full adders and AND gates to design a 3-bit \times 3-bit multiplier.



Q2 - Assume the 2 numbers are $B = (B_2 B_1 B_0)$ $A = (A_2 A_1 A_0)$
 The product is $P = (P_2 P_1 P_0)$

$$\Rightarrow P_0 = (B_0 \times A) + P_{00} = A_0 \cdot B_0 \quad P_{01} = A_1 \cdot B_0 \quad P_{02} = A_2 \cdot B_0$$

P_1, P_2 are quite similar



$$S_0 = P_{00} \quad S_1 = \text{SUM}(P_{01}, P_{10}) \quad \text{because the } C_{in} = 0 \text{ initially}$$

$$S_2 = \text{Adding } P_{02}, P_{11}, P_{20} \quad C_{in} = C_1$$

$$FA_2 \text{ (Full adder)} \Rightarrow \text{calculate } P_{02} + P_{11} + C_1 \quad \text{Sum} = S_2' \quad \text{Cout} = C_2$$

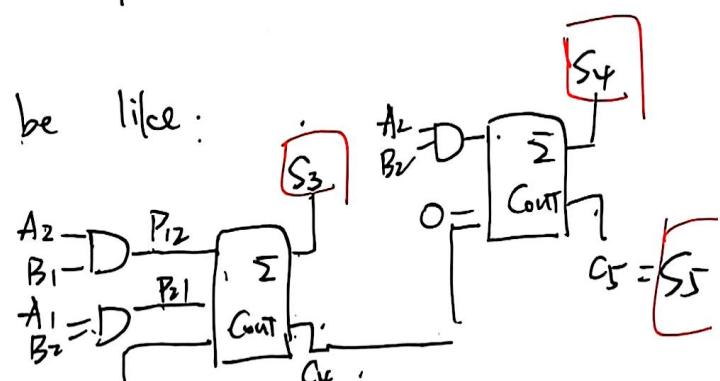
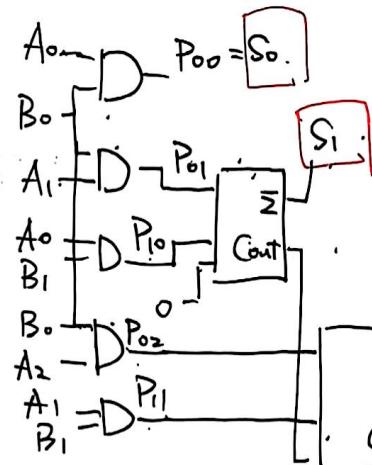
$$FA_3 \text{ calculate } S_2' + P_{20} \quad S_2 = \text{SUM} \quad C_3 = \text{Cout}$$

$$S_3 = \text{SUM}(P_{12}, P_{21}, C_3) \quad C_4 = \text{Cout}$$

$$S_4 = \text{SUM}(P_{20}, C_4, 0) \quad C_5 = \text{Cout}$$

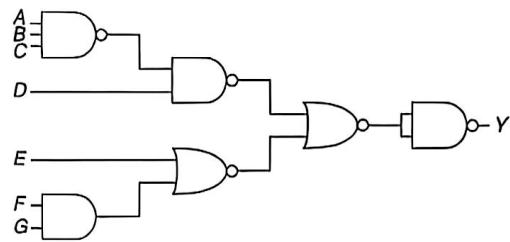
$$S_5 = C_5$$

\Rightarrow The whole picture will be like:



扫描全能王 创建

Q3 Determine the propagation delay and contamination delay of the circuit below.



Use the gate delays given below.

Gate	t_{pd} (ps)	t_{cd} (ps)
NOT	15	10
2-input NAND	20	15
3-input NAND	30	25
2-input NOR	30	25
3-input NOR	45	35
2-input AND	30	25
3-input AND	40	30
2-input OR	40	30
3-input OR	55	45
2-input XOR	60	40



Q3:

Here, to find what is required, we should consider the longest and shortest path:

tpd is the maximum among all tpd's

tcd is the minimum among all tcd's

For A, B, C to Y: $30 + 20 + 30 + 20 = 100 \text{ ps}$

D to Y: $20 + 30 + 20 = 70 \text{ ps} \Rightarrow \text{tpd}$

E to Y: $30 + 30 + 20 = 80 \text{ ps}$

F, G to Y: $30 + 30 + 30 + 20 = 110 \text{ ps}$

$\Rightarrow 110 \text{ ps}$ for tpd

\Rightarrow For A, B, C to Y: $25 + 15 + 25 + 15 = 80 \text{ ps}$

D to Y: $15 + 25 + 15 = 55 \text{ ps}$

E to Y: $25 + 25 + 15 = 65 \text{ ps} \Rightarrow \text{tcd}$

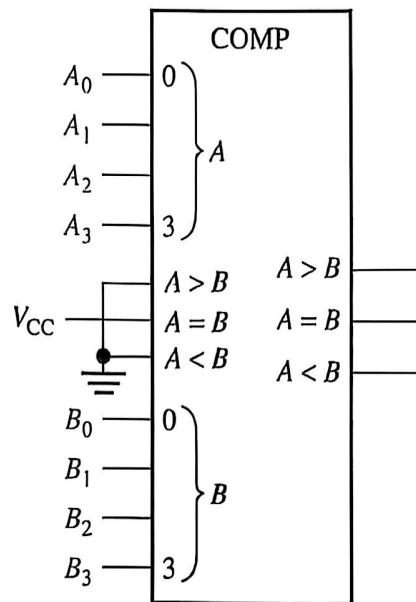
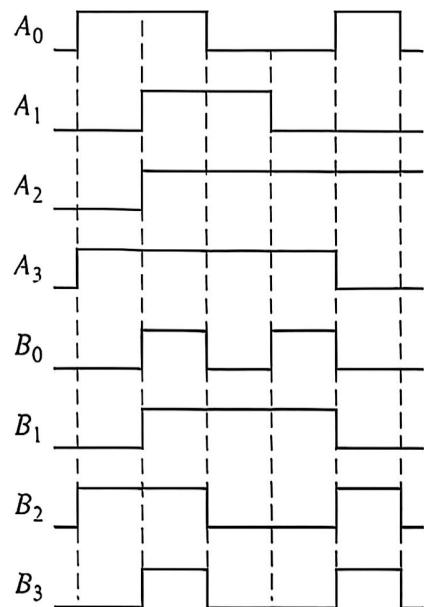
F, G to Y: $25 + 25 + 25 + 15 = 80 \text{ ps}$

$\Rightarrow 55 \text{ ps}$ for tcd



扫描全能王 创建

Q4 For the 4-bit comparator below, plot each output waveform for the inputs shown. The outputs are active-HIGH.

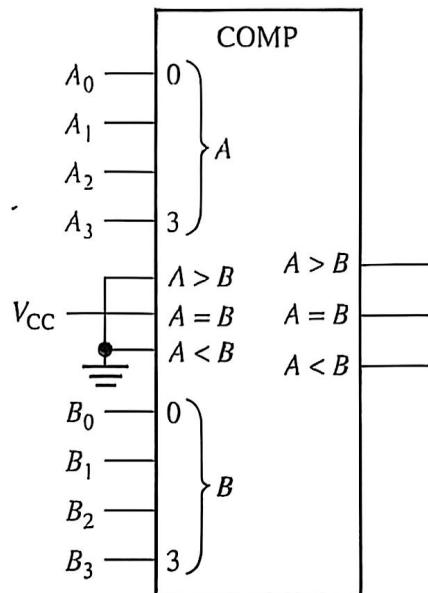


74HC85

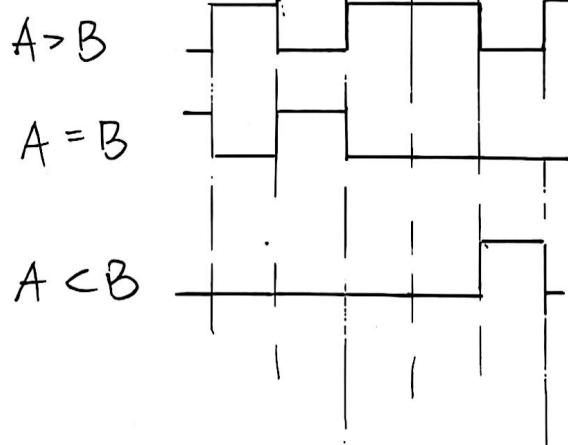


Q4 For the 4-bit comparator below, plot each output waveform for the inputs shown. The outputs are active-HIGH.

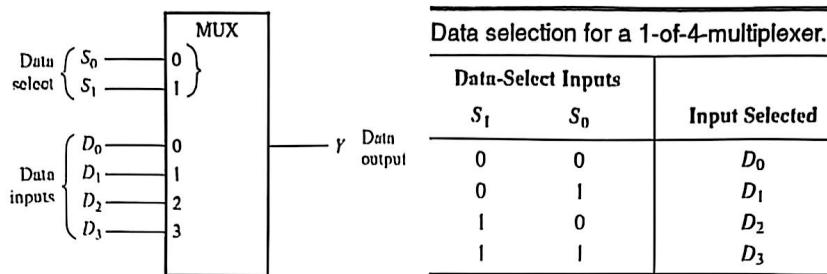
A_0	0	1	1	0	0	1	0
A_1	0	0	1	1	0	0	0
A_2	0	1	1	1	1	1	1
A_3	1	1	1	1	0	0	0
B_0	0	1	0	1	0	0	0
B_1	0	1	1	1	0	0	0
B_2	1	1	0	0	1	0	0
B_3	0	1	0	0	1	0	0



74HC85



- Q5** 1. Using basic gates (such as AND, OR, Inverter) to implement the following data selector/multiplexer. Show your logic diagram.



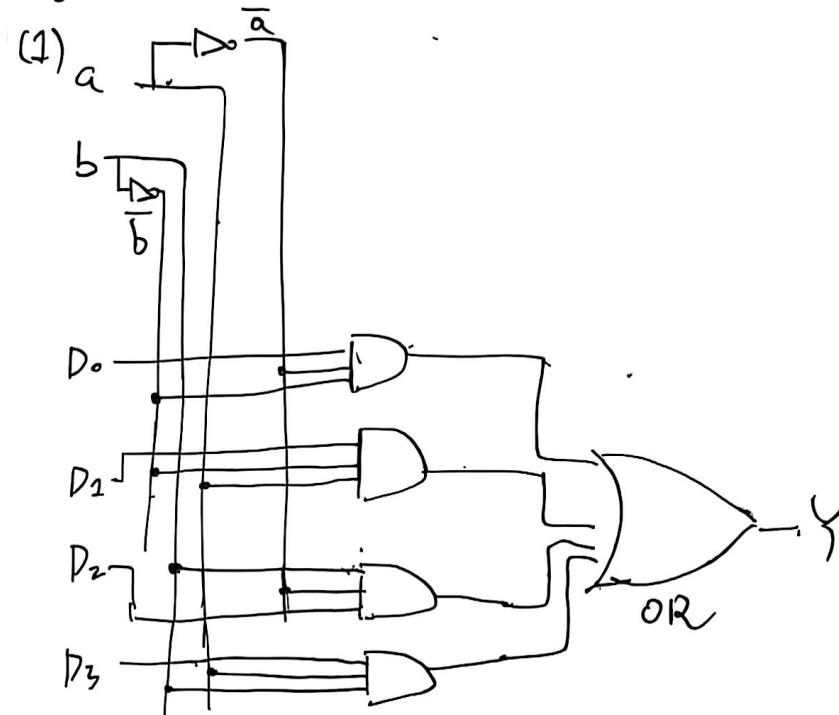
2. Use K-map to find out the Sum of Products (SOP) expression of Y ;

Inputs			Output
A_2	A_1	A_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

3. Using one 1-of-4 data selector/multiplexer designed in (1) and necessary basic logic gates to implement the truth table in (2). Show your steps and the resulting logic diagram.



Q5.



(2) We write it as

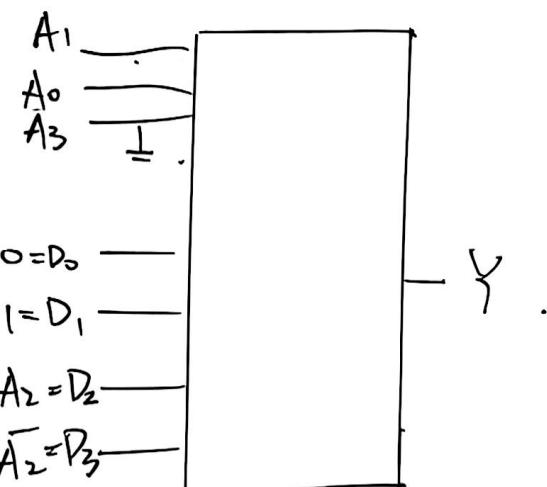
$A_2 \backslash A_1 A_0$	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$$\Rightarrow Y = \bar{A}_2 A_0 + \bar{A}_1 A_0 + A_2 A_1 \bar{A}_0$$

(3) We let $D_0 = 0$ $D_1 = 1$ $D_2 = A_2$ $D_3 = \bar{A}_2$

\Rightarrow It turns to be

A_3	A_2	A_1	A_0	Y
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	1	1	0	1
0	1	1	1	0

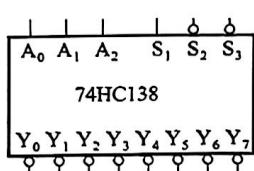


$$Y = D_0 \bar{A}_1 \bar{A}_0 + D_1 \bar{A}_1 A_0 + D_2 A_1 \bar{A}_0 + D_3 A_1 A_0$$



Q6 A lab in CUHK SZ uses two LEDs to display the fault status of three devices (A , B and C). The yellow LED (Z_1) turns on if only one device has faults, the red LED (Z_2) turns on if any two devices have faults. Both yellow and red LEDs turn on if all three devices have fault detected. Please design a logic circuit to realize the above functions. Note that “fault detected” and “LED on” are represented with logic high (‘1’), “no fault” and “LED off” are represented with logic low (‘0’);

1. Please develop the truth table for the scenario;
2. Please derive the logic expression in **standard Product of Sum (POS)** form from the truth table;
3. The chip 74HC138 is a 3-to-8 decoder, whose chip diagram and truth table are shown below. Please use one 74HC138 chip and the necessary logic gates to implement the expressions in (2).



S_1	Input			Output								
	$\bar{S}_2 + \bar{S}_3$	A_2	A_1	A_0	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7
0	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0



Q6 1. According to the description, the truth table should be like

A	B	C	Z_1 (Yellow)	Z_2 (Red)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2. For POS:

$$\text{when } Z_1 = 0 \Rightarrow 000 \quad 011 \quad 101 \quad 110$$

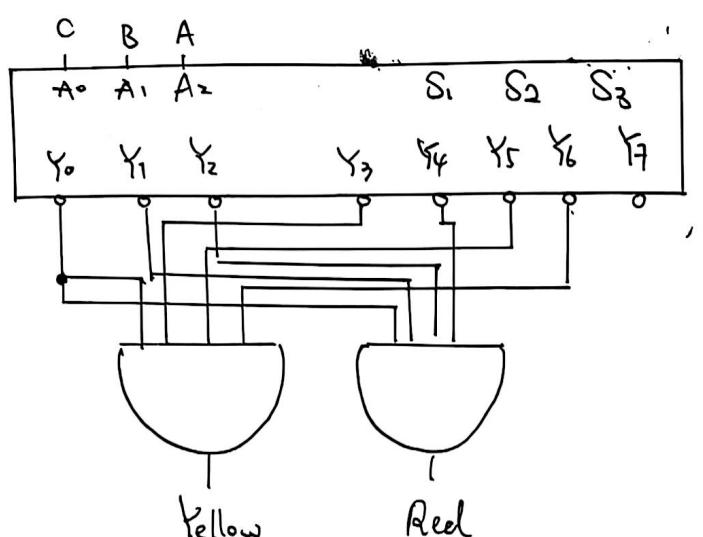
$$\Rightarrow Z_1 = (A+B+C)(A+\bar{B}+\bar{C})(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)$$

$$\text{when } Z_2 = 0 \Rightarrow 000 \quad 001 \quad 010 \quad 100$$

$$Z_2 = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+\bar{B}+C)$$

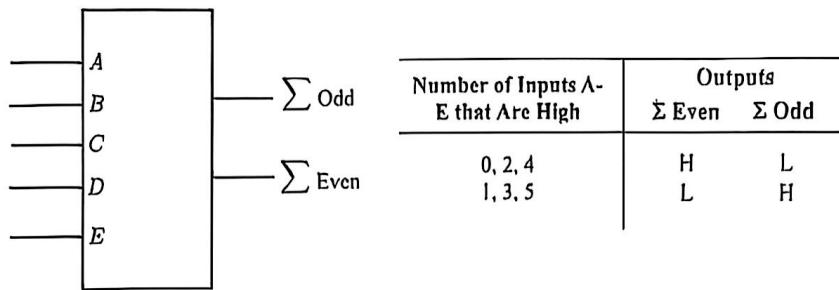
3. So, for the connection:

$$Z_1 = \overline{Y_0 + Y_3 + Y_5 + Y_6}, \quad Z_2 = \overline{Y_0 + Y_1 + Y_2 + Y_4}$$

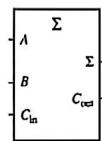


扫描全能王 创建

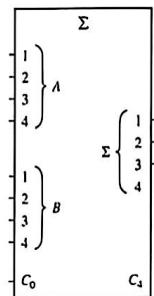
Q7 A Parity Generator/Checker and its function table are shown below.



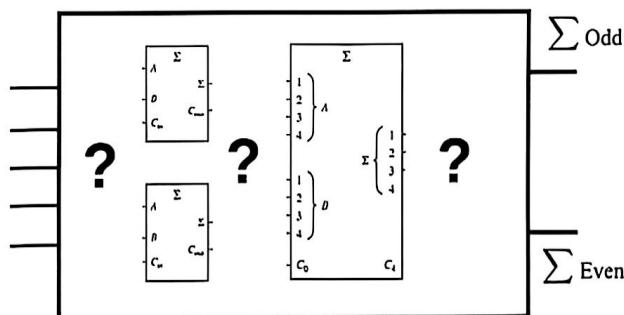
1. Use basic logic gates to design a 1-bit full adder as shown below where A and B are the input bits while C_{in} is the input carry. Finally, \sum is the sum and C_{out} is the output carry.

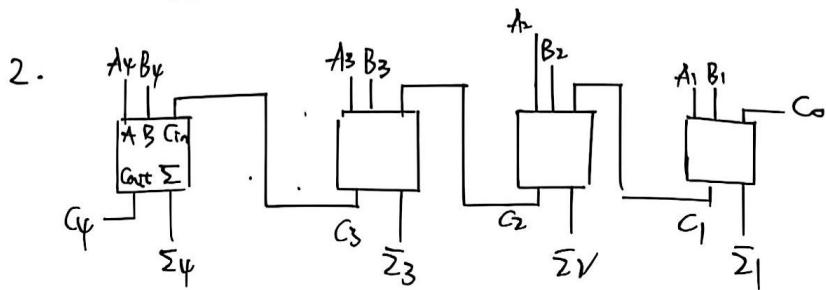
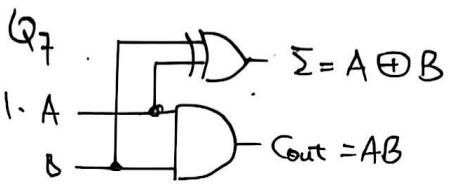


2. Use 4 1-bit full adders to implement a 4-bit adder as shown below. A and B are the 4-bit input number where \sum is the 4-bit output. C_0 is the input carry and C_4 is the output carry.



3. Finally, use two 1-bit full-adders and one 4-bit adder together with other basic logic gates to implement the Parity Generator/Checker.



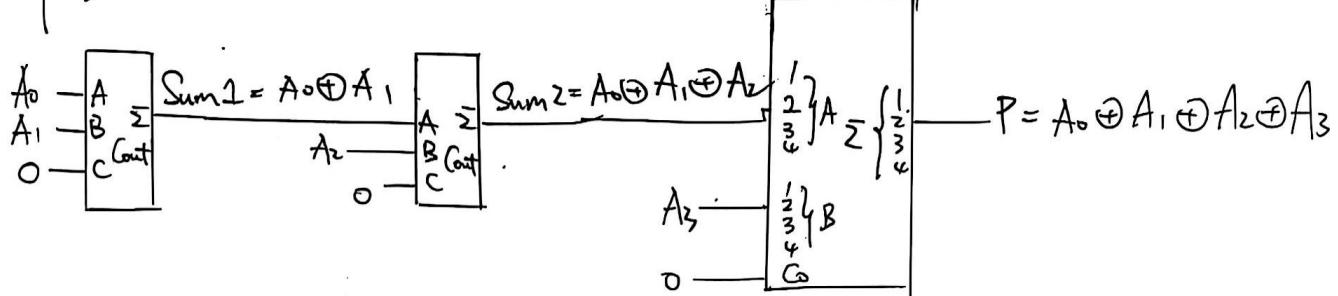


3. Step 1: If we want to design a parity checker, the checking bit should act like: $P = A_3 \oplus A_2 \oplus A_1 \oplus A_0$

Step 2: The logic for full-adder is: $\text{Sum} = A \oplus B \oplus \text{Cin}$
when $\text{Cin} = 0$, $\text{Sum} = A \oplus B$

⇒ It can be considered as a two-in-XOR gate.

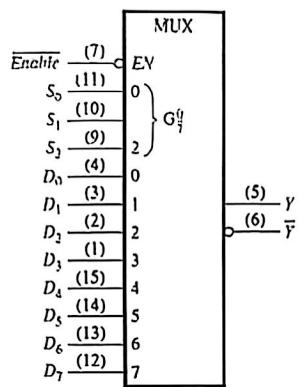
Step 3:



扫描全能王 创建

Q8 Implement the below logic expression by using a 74HC151 8-input data selector/multiplexer and logic gates.

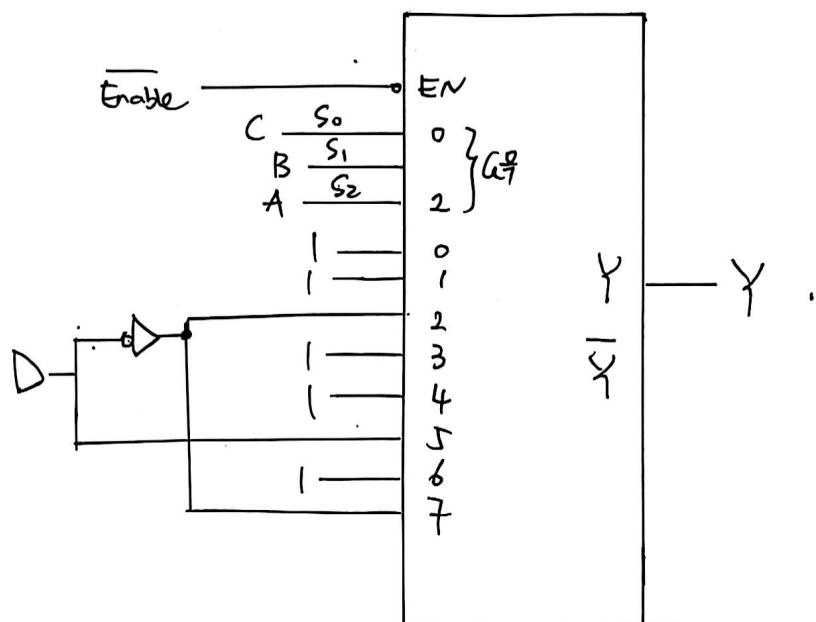
$$(A + \bar{B} + C + \bar{D})(\bar{A} + B + \bar{C} + D)(\bar{A} + \bar{B} + \bar{C} + \bar{D})$$



Q8:

Let's choose A, B, C to S_2, S_1, S_0 , D to be the controller for data input
Let's put A B C in and check D:

$ABC = \cancel{0} 0 00$	(D0) $\rightarrow Y=1$	$D_0=1$
$ABC = 0 0 1$	$Y=1$	$D_1=1$
$ABC = 0 1 0$	$Y=\bar{D}$	$D_2=\bar{D}$
$ABC = 0 1 1$	$Y=1$	$D_3=1$
$ABC = 1 0 0$	$Y=1$	$D_4=1$
$ABC = 1 0 1$	$Y=D$	$D_5=D$
$ABC = 1 1 0$	$Y=1$	$D_6=1$
$ABC = 1 1 1$	$Y=\bar{D}$	$D_7=\bar{D}$



Y is what we need



扫描全能王 创建