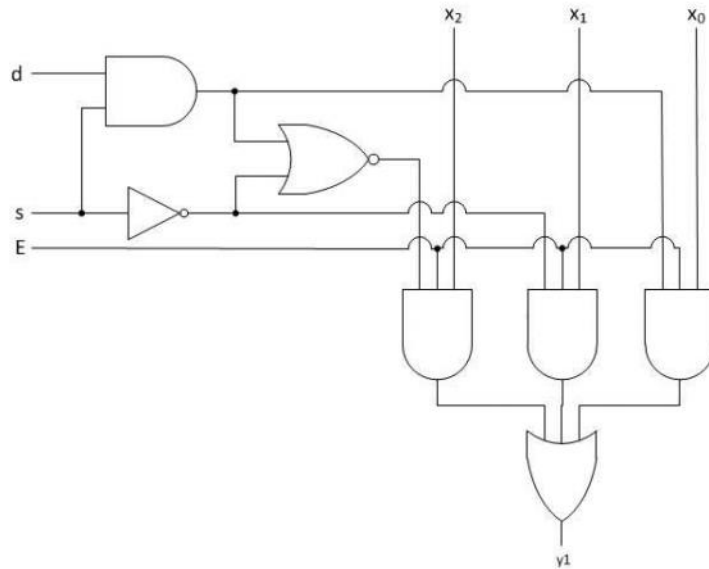


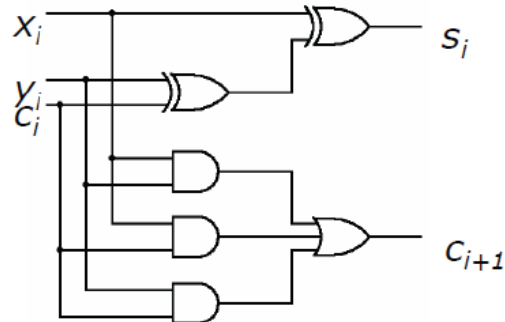
Bài tập chương 5 – Mạch tổ hợp

Mới: bài 8 – 12 (SANG review)

1. Cho sơ đồ sau:



- Viết hàm boolean cho sơ đồ
 - Lập bảng chân trị cho hàm y_1
2. Chuyển mạch FA dưới đây sang mạch với các cổng NAND



Gợi ý: viết lại hàm sau chuyển sang dạng NAND

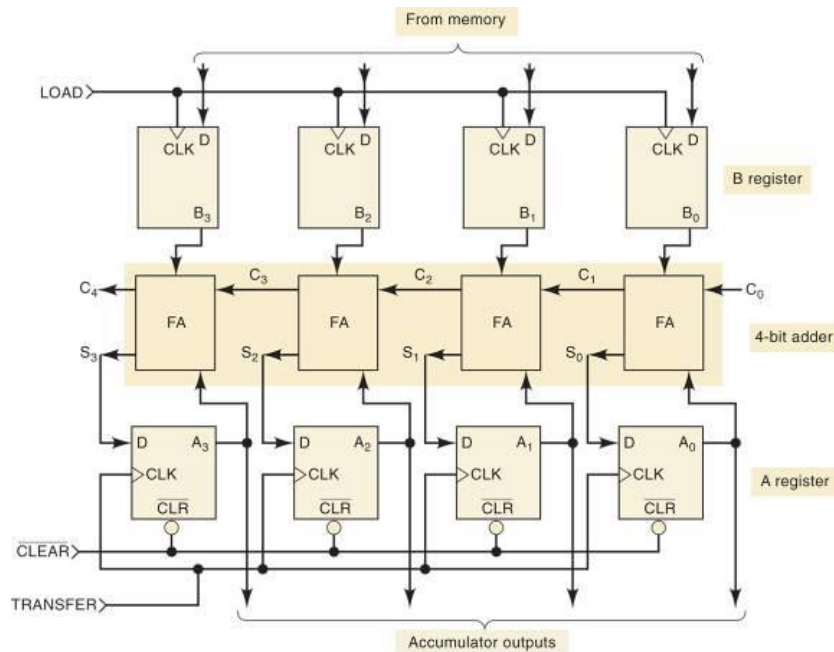
- Trong các máy vi tính hoặc các máy tính, thường có một mạch để xác định overflow. Có vài cách để làm việc này. Một trong những cách có thể được sử dụng cho mạch cộng số bù 2 được thực hiện theo các bước như sau:
 - Kiểm tra bit dấu của 2 số được cộng
 - Kiểm tra bit dấu kết quả
 - Có overflow khi các số được cộng cùng là số dương và bit dấu của kết quả là 1 hoặc khi các số được cộng là số âm và bit dấu của kết quả là 0.

Phương pháp này có thể được kiểm chứng bằng các ví dụ.

- $5 + 4$
- $-4 + (-6)$
- $3 + 2$

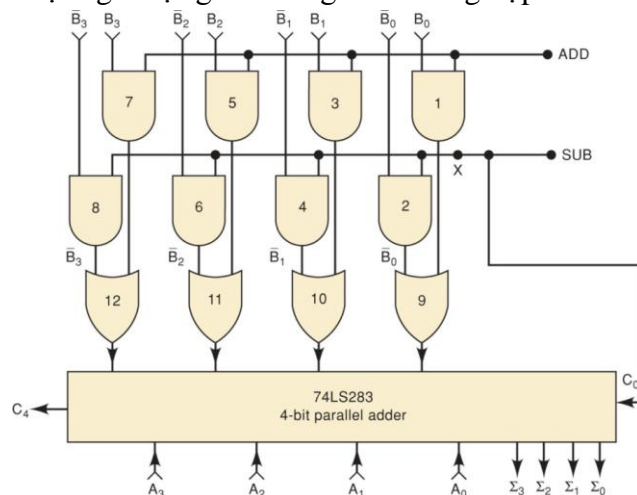
Trường hợp a, b sẽ có overflow còn trường hợp c thì không.

Vì vậy, bằng cách kiểm tra bit dấu, một mạch logic có thể được thiết kế để cho ra 1 khi nào overflow xuất hiện. Hãy thiết kế mạch kiểm tra overflow cho mạch cộng bên dưới.



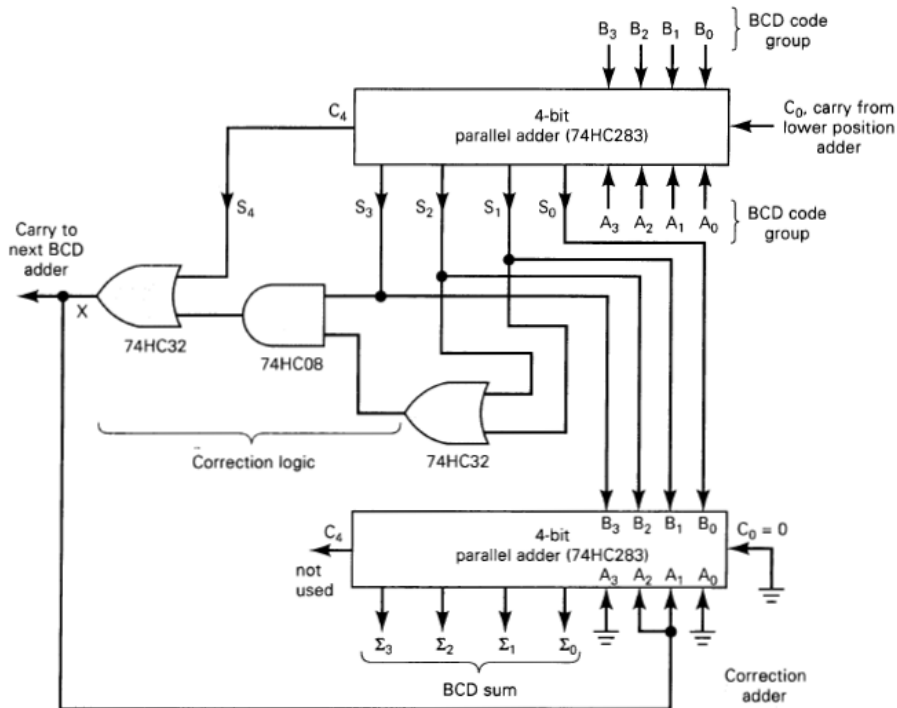
Gợi ý: Mạch trên là $A=A+B$, lập bảng chân trị cho mạch có đầu vào là $A_3B_3S_3$, đầu ra là tín hiệu Overflow O

4. Cho mạch bên dưới, xác định giá trị ngõ ra trong các trường hợp sau:



- Số $A = 0101 (+5)$, Số $B = 1110 (-2)$; SUB = 1, ADD = 0
- Số $A = 1100 (-4)$, Số $B = 1110 (-2)$; SUB = 0, ADD = 1

5. Sửa lại mạch ở bài 3 với chỉ 1 ngõ vào X thay cho 2 ngõ vào ADD và SUB. Mạch mới này sẽ thực hiện nhiệm vụ mạch cộng khi $X = 0$, và mạch trừ khi $X = 1$. Sau đó rút gọn các cổng.
(Gợi ý: dùng thêm một cổng NOT)
Có thể dùng MUX 2-1 với ngõ điều khiển là X , 2 đầu vào là B và B' .
6. Đối với mạch cộng BCD bên dưới, có hay không có sự khác biệt gì nếu C_0 của mạch cộng phía trên được giữ ở mức thấp trong khi C_0 của mạch cộng dưới được sử dụng như là số nhớ đầu vào (carry in)? Giải thích.



Có. Bộ cộng BCD, hiệu chỉnh khi kết quả sai thì hiệu chỉnh bằng cách cộng với 0110.

27	0010 0111	
+ 36	0011 0110	
63	0101 1101	→ Ký số vượt quá => kết quả sai

0000 0110	→ Sửa sai kết quả
0110 0011	Kết quả = 63

28	0010 1000	
+ 59	0101 1001	
87	1000 0001	→ Có nhớ 1 => kết quả sai

0000 0110	→ Sửa sai kết quả
1000 0111	Kết quả = 87

7. Mỗi phát biểu dưới đây liên hệ đến 1 decoder hay 1 encoder?
- Có nhiều input hơn output
 - Được sử dụng để chuyển đổi 1 phím được bấm sang mã nhị phân
 - Chỉ 1 output được tích cực tại một thời điểm

d. Có thể được sử dụng để giao tiếp 1 input dạng BCD với 1 bộ hiển thị LED

8. Chứng minh theo Shannon biểu thức sau:

a) $AB + B'CD + ACD = AB + B'CD$

Gợi ý: khai triển theo B

b) $(A+B+C)(A'+D)(B+C+D) = (A+B+C)(A'+D)$

Gợi ý: khai triển theo A công thức Shannon dưới dạng tích các tổng

9. Một Full subtractor (x-y) có 3 bits vào: x, y, borrow-in b_{in} và 2 đầu ra: hiệu số d (difference), bit mượn borrow-out b_{out}

a) Xây dựng bảng sự thật

b) Viết hàm dạng SOP

c) Viết hàm dạng POS

d) Vẽ sơ đồ mạch của Full Sub cho câu b hoặc c

10. Cho các hàm sau:

– $F1(A,B,C,D) = \Sigma(1,2,4,7) + d(3)$

– $F2(A,B,C,D) = \Sigma(0,3,14) + d(15)$

– $F3(A,B,C,D) = \Sigma(12,15)$

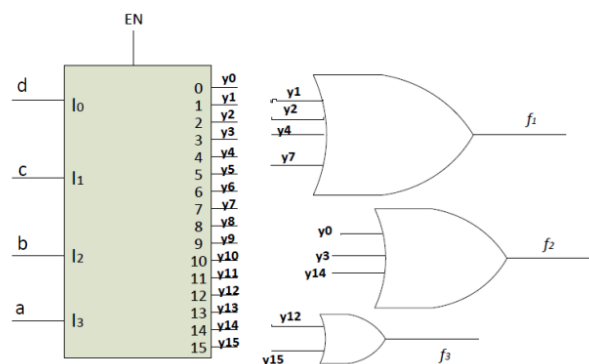
a. Hiện thực các hàm sử dụng 4-16 Decoders và cổng OR

b. Hiện thực các hàm sử dụng 3-8 Decoders và cổng OR

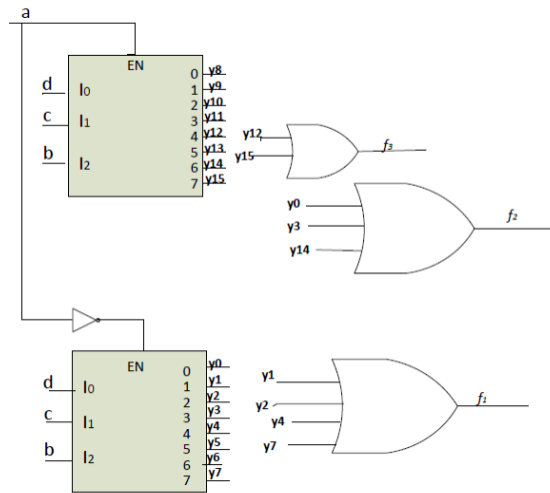
c. Hiện thực các hàm sử dụng 2-4 Decoders và cổng OR

Gợi ý:

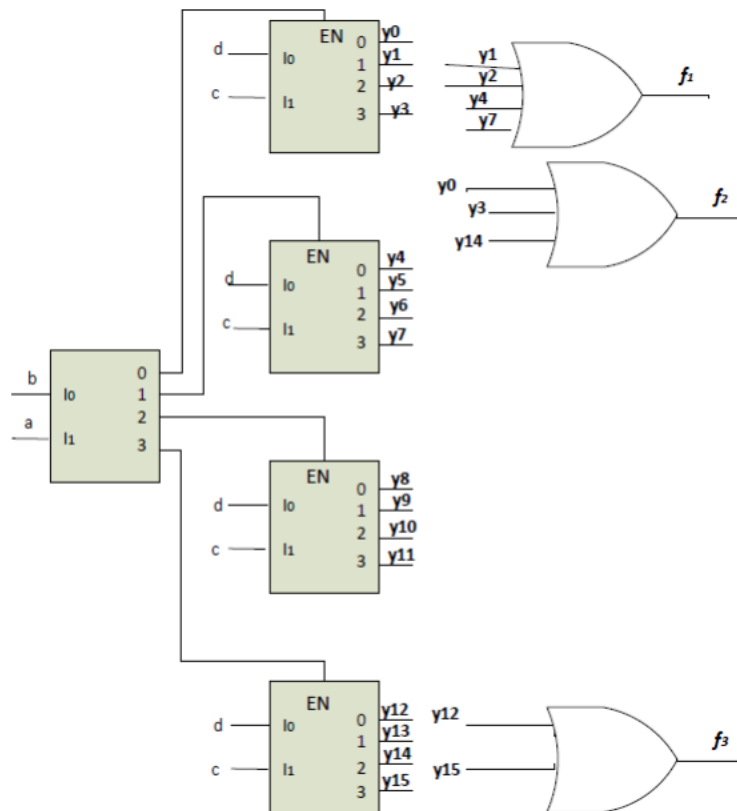
a)



b)



c)

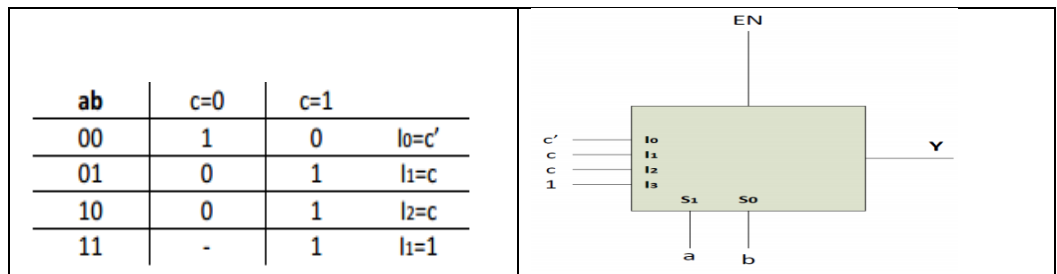


11. Cho hàm $F(A,B,C) = \Sigma(0,3,5,7) + d(6)$

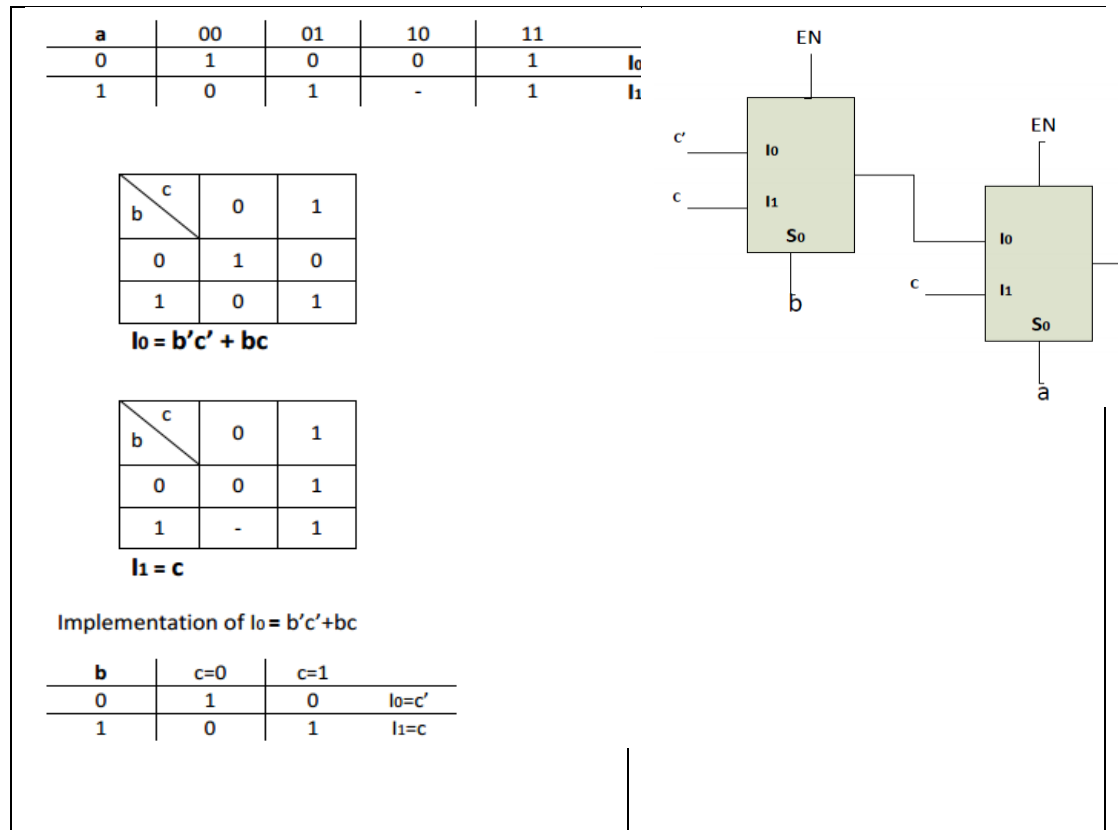
- Hiện thực các hàm sử dụng 8-1 MUX
- Hiện thực các hàm sử dụng 4-1 MUX
- Hiện thực các hàm sử dụng 2-1 MUX

Gợi ý:

- A,B,C là 3 biến điều khiển
-



c)



12. Cho hàm sau: $F(A,B,C) = \Sigma(0,2,4,6,7) + d(1)$

- Hiện thực hàm F sử dụng ít nhất các mạch 2-4 decoders và cổng OR
- Hiện thực hàm F sử dụng ít nhất các mạch 4-1 MUX
- Hiện thực hàm F sử dụng ít nhất các mạch 2-1 MUX

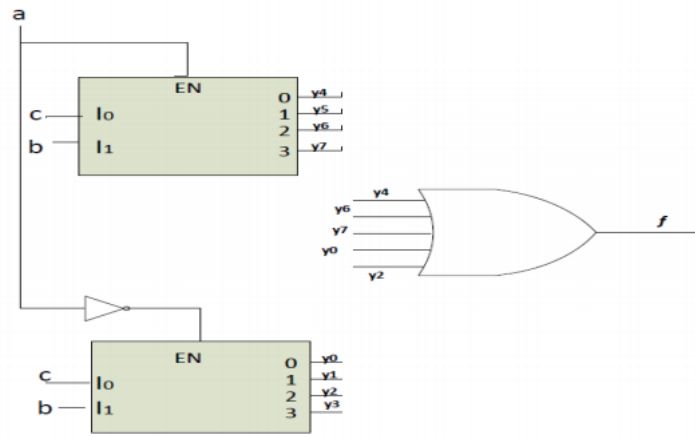
Gợi ý:

a)

The truth table for the given function looks like the following:

a	b	c	f
0	0	0	1
0	0	1	-
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

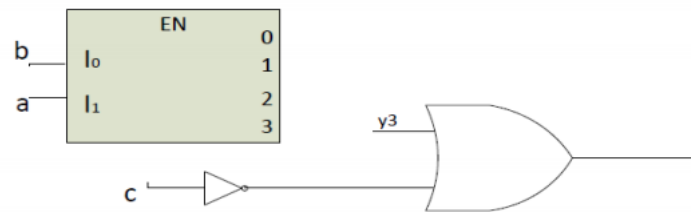
1. Implementation using 2:4 decoder is given below.



Alternative Solution.

$f(a,b,c) = ab + c'$ (using Kmap)

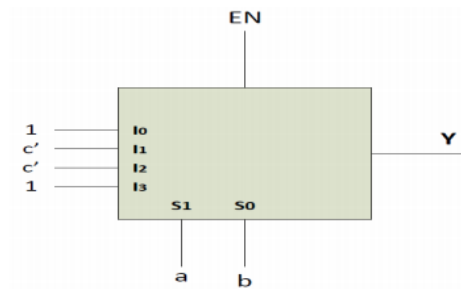
The implementation will therefore be



b)

Implementation using 4:1 Mux

ab	c=0	c=1	
00	1	-	$I_0=1$
01	1	0	$I_1=c'$
10	1	0	$I_2=c'$
11	1	1	$I_3=1$



c)

a	Bc			
	00	01	10	11
0	1	-	1	0
1	1	0	1	1

c \ b	0	1
0	1	-
1	1	0

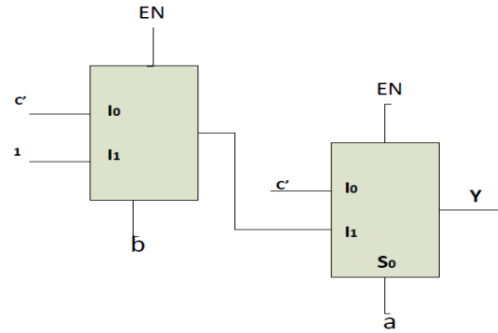
$$l_0 = c'$$

c \ b	0	1
0	1	0
1	1	1

$$l_1 = b + c'$$

Implementation of $l_1 = b + c'$

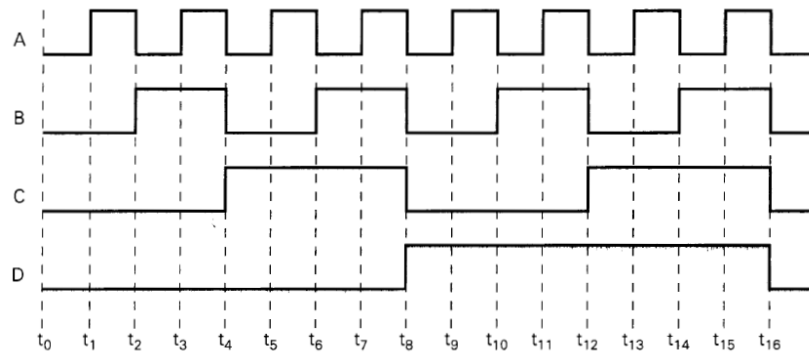
B	c=0	c=1	
0	1	0	$l_0 = c'$
1	1	1	$l_1 = 1$



13. Xem xét dạng sóng dưới đây. Đưa tín hiệu này vào IC 74LS138 như sau: **Đề nghị sửa lại vì các ký hiệu tương ứng không có trong 74LS138**

$A \Rightarrow A_0$; $B \Rightarrow A_1$; $C \Rightarrow A_2$; $D \Rightarrow E_3$

Giả sử $\overline{E_1}$ và $\overline{E_2}$ được giữ ở mức thấp. Hãy vẽ dạng sóng cho ngõ ra $\overline{O_0}$, $\overline{O_3}$, $\overline{O_6}$, và $\overline{O_7}$



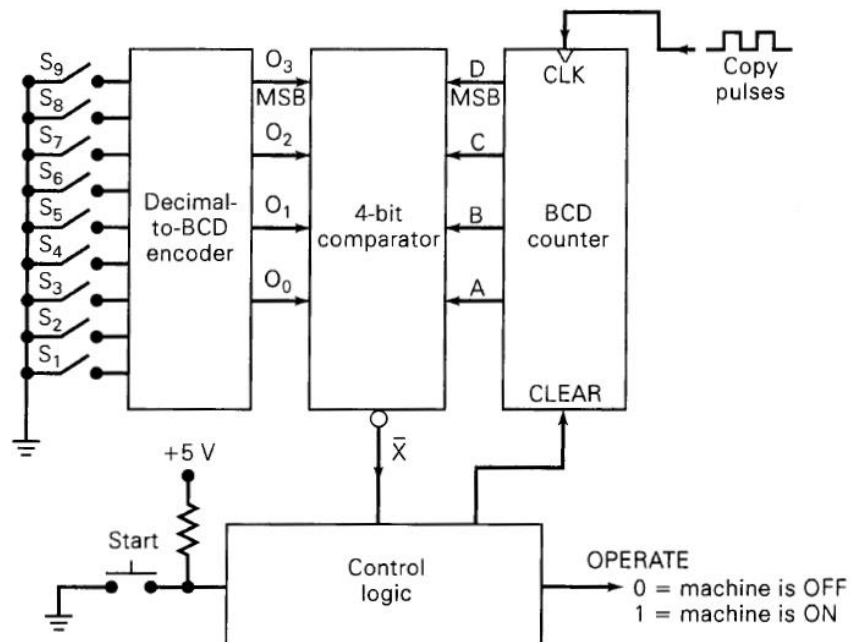
Gợi ý:

Inputs						Outputs							
G1	G2A_L	G2B_L	C	B	A	Y7_L	Y6_L	Y5_L	Y4_L	Y3_L	Y2_L	Y1_L	Y0_L
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	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	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	0	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

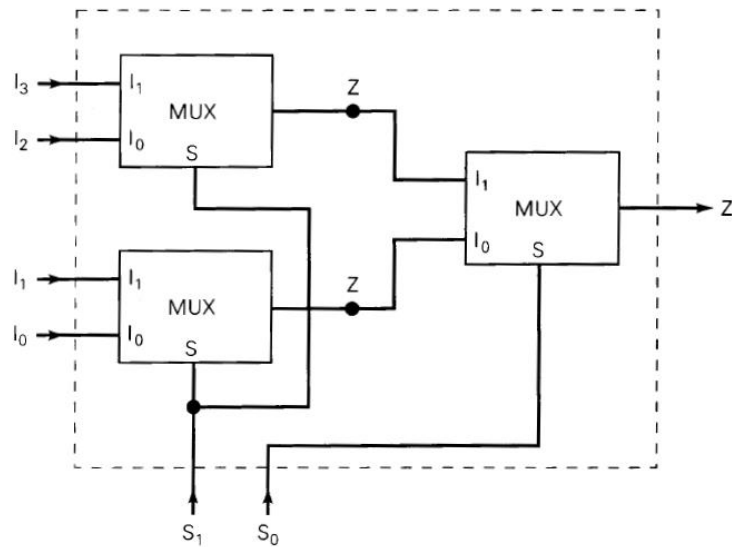
14. Bên dưới là sơ đồ khối của mạch điều khiển số lượng bản copy của 1 máy photocopy. Người thao tác trên máy sẽ thiết lập một số lượng bản copy mong muốn bằng cách đóng một trong các switch S1 đến S9. Số này được mã hóa BCD bởi mạch mã hóa và được đưa vào mạch comparator. Người thao tác máy sẽ nhấn nút START, để xóa counter và khởi tạo ngõ ra OPERATE ở mức cao để làm tín hiệu cho máy thực hiện việc copy.

Khi máy hoạt động, một xung copy được tạo ra và đưa vào BCD counter. Ngõ ra của counter liên tục được so sánh (bởi mạch comparator) với mã hóa của các switch. Khi 2 số BCD này giống nhau, cho biết đã copy đủ số lượng, ngõ ra \bar{X} của comparator sẽ xuống mức THẤP, làm cho tín hiệu OPERATE trở về mức THẤP và ngưng máy photocopy không cho copy thêm bản nào nữa. Kích hoạt nút START sẽ lặp lại quá trình này.

Hãy thiết kế một mạch comparator hoàn chỉnh và mạch điều khiển của hệ thống này.



15. Mạch ở dưới sử dụng 3 multiplexer 2-1. Xác định hàm được thực hiện bởi mạch này.

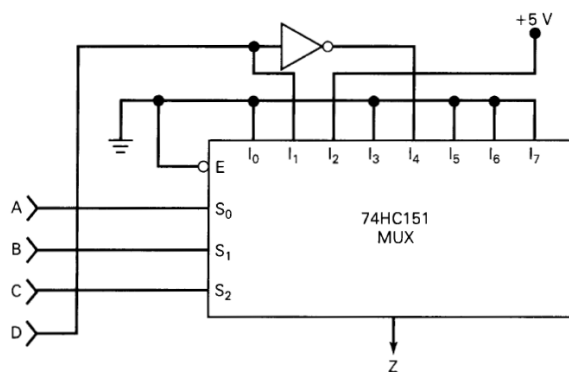


16. Mô tả cách sử dụng IC 74x151 (multiplexer 8 inputs) để tạo ra hàm logic $z = AB + BC + AC$.

17. Mạch bên dưới biểu diễn cách 1 multiplexer 8 inputs được sử dụng để tạo ra một hàm logic 4 biến ngay cả khi MUX chỉ có 3 tín hiệu SELECT đầu vào. Ba biến A, B, và C được nối vào SELECT. Biến thứ tư D và đảo của nó \bar{D} được nối vào một vài chân dữ liệu của MUX có chọn lựa như yêu cầu của hàm logic mong muốn. Những chân dữ liệu khác của MUX được giữ ở mức THẤP hoặc CAO qui định bởi hàm logic.

- Lập bảng sự thật cho mạch
- Viết biểu thức của Z dưới dạng SOP và tối ưu hóa Z để kiểm tra rằng

$$Z = \bar{C}B\bar{A} + D\bar{C}\bar{B}A + \bar{D}C\bar{B}\bar{A}$$



18. Mỗi phát biểu dưới đây liên hệ đến 1 decoder, 1 encoder, 1 MUX, hay 1 DEMUX?

- Có nhiều inputs hơn outputs
- Sử dụng chân input SELECT
- Có thể được sử dụng để chuyển từ song song sang tuần tự (parallel-to-serial conversion)
- Tạo ra 1 mã nhị phân ở outputs

- e. Chỉ một trong những outputs được tích cực tại một thời điểm
- f. Có thể được sử dụng để hướng 1 tín hiệu output tới 1 hoặc một vài outputs
- g. Có thể được sử dụng để sinh ra hàm logic bất kì