

CSE140 - HW #4 - Solution

Due Monday May 28, 11:59PM

We practice the standard interconnect module designs and applications.

1. (Decoders) Given four four-input Boolean functions **(35 Points)**

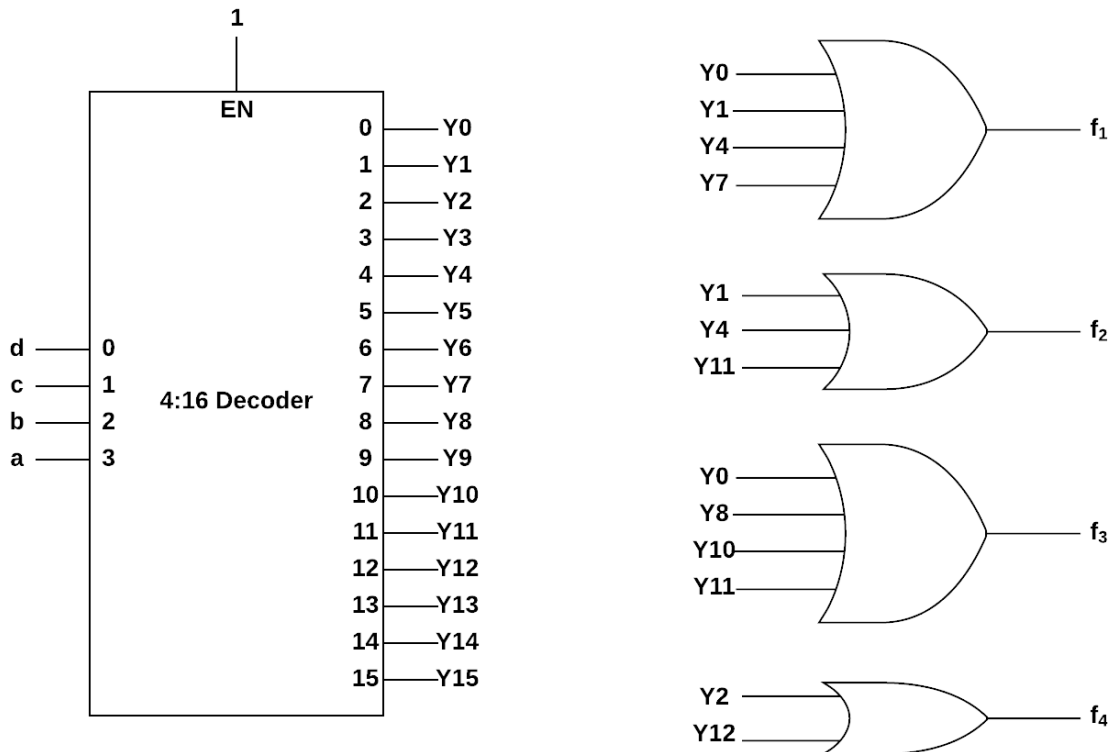
$$f_1(a, b, c, d) = \sum m(0, 1, 4, 7) + \sum d(2, 3, 5),$$

$$f_2(a, b, c, d) = \sum m(1, 4, 11) + \sum d(3, 14),$$

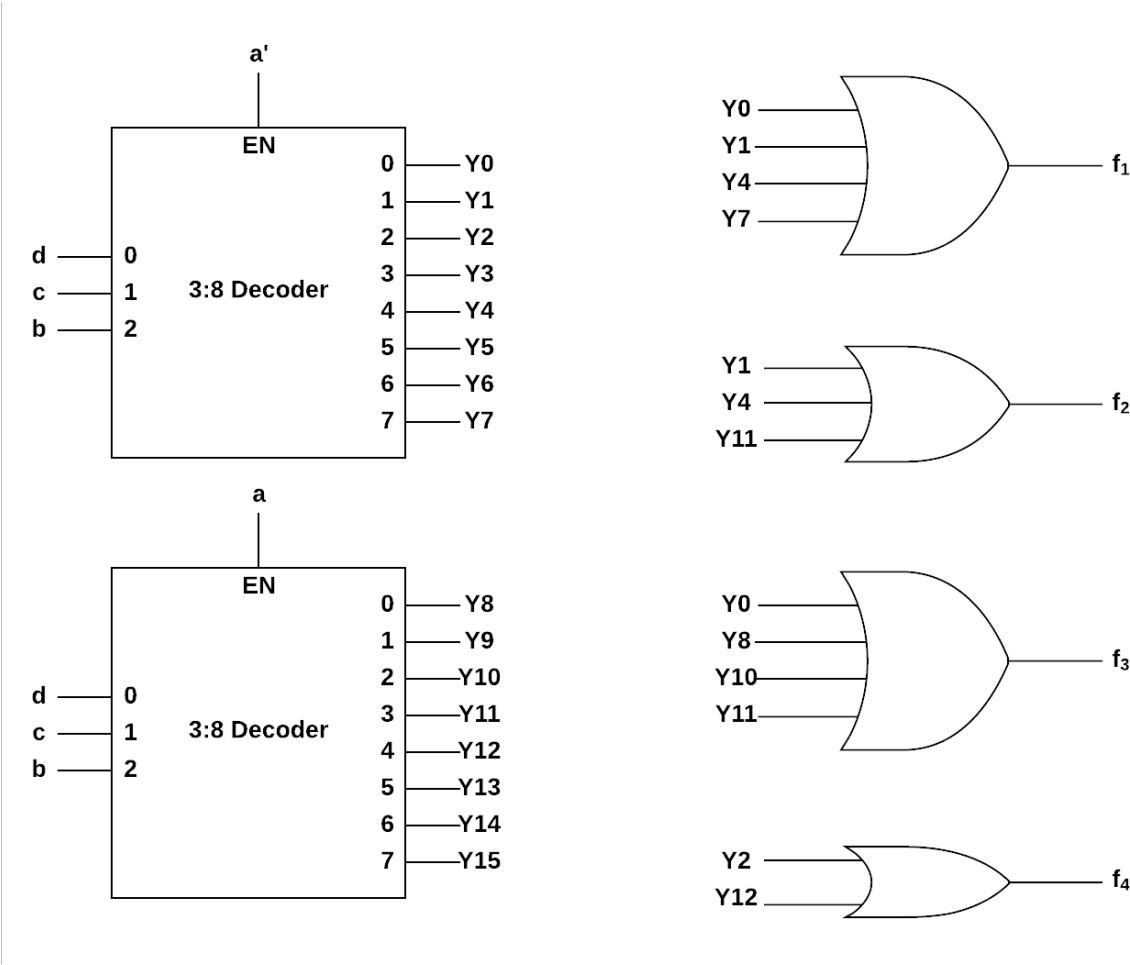
$$f_3(a, b, c, d) = \sum m(0, 8, 10, 11) + \sum d(9, 12, 15).$$

$$f_4(a, b, c, d) = \sum m(2, 12) + \sum d(8, 11, 14).$$

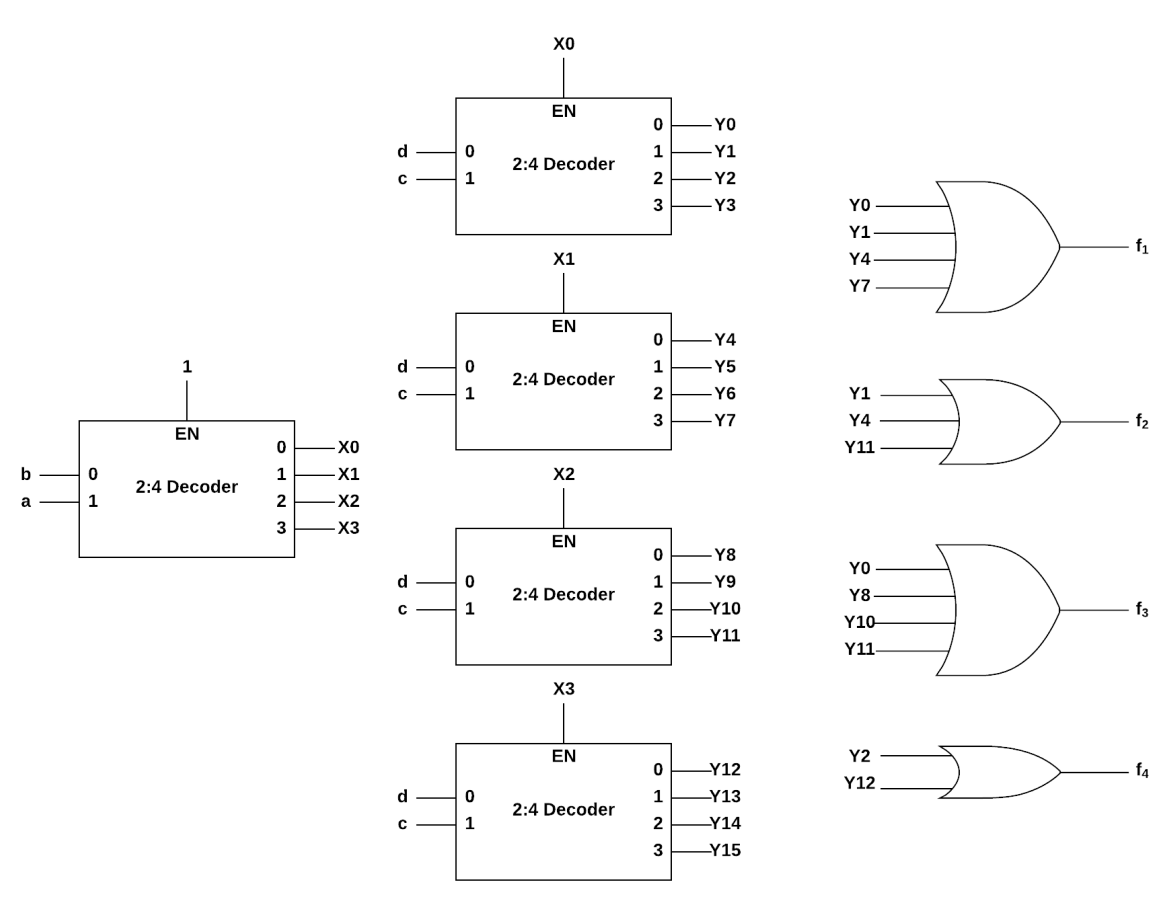
1.1. Implement the functions using a minimal network of 4:16 decoders and OR gates. **(5 Points: Completion)**



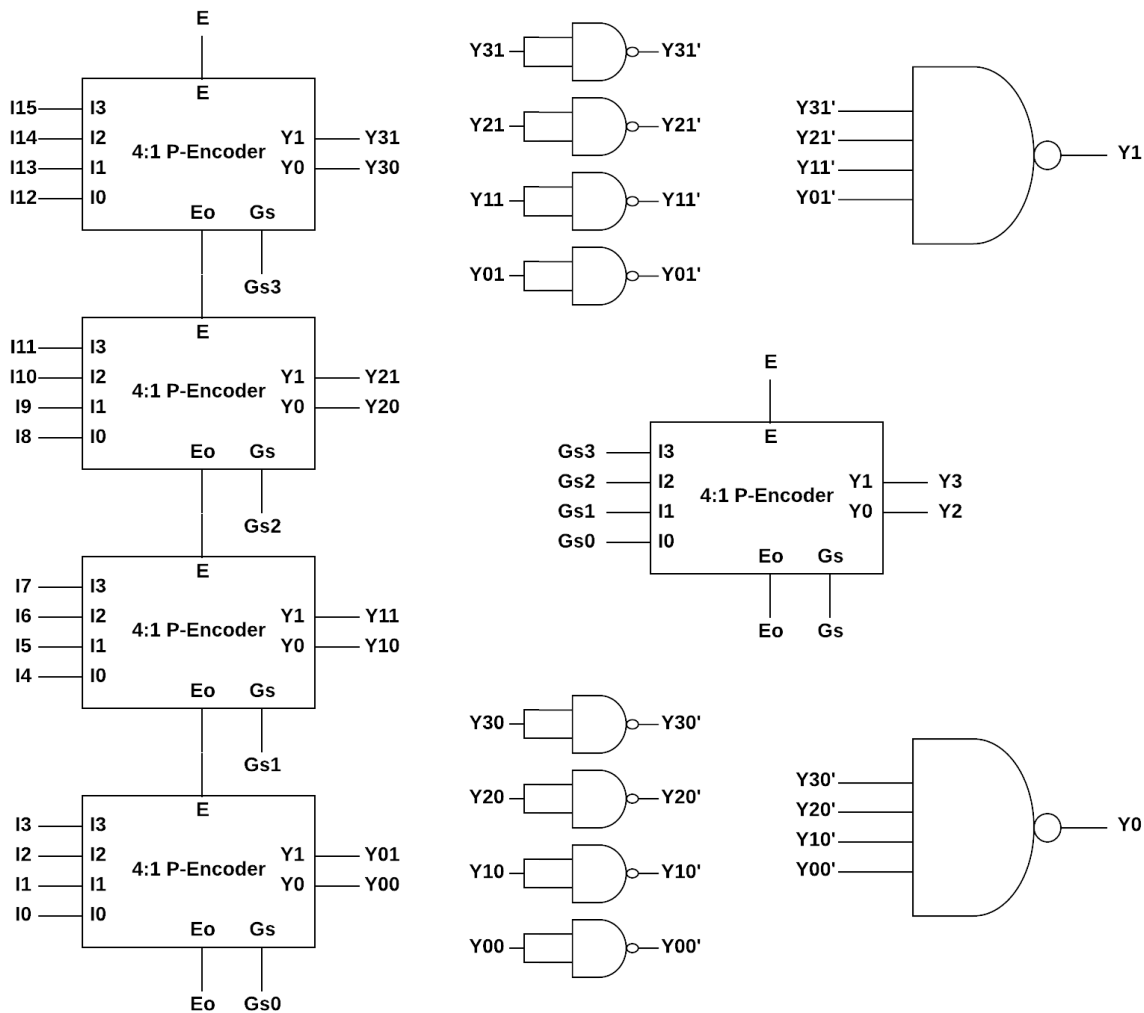
1.2. Implement the functions using a minimal network of 3:8 decoders and OR gates. (25 Points: Correctness)



1.3. Implement the functions using a minimal network of 2:4 decoders and OR gates. (5 Points: Completion)



2. (Encoders) A high bit priority encoder inputs 2^n bits from 2^n devices and outputs n bit as the index of the asserted input line with the highest priority (largest in binary code) as shown in page 25 of lecture 11. Implement a high bit 16:4 priority encoder using 4:2 high bit priority encoders and minimal networks of NAND gates. **(15 Points: Completion)**

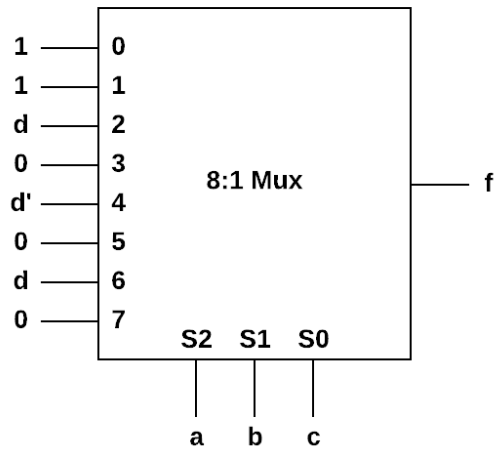


3. (Multiplexers) Assume a dual-railed system, where you have access to any variable and its complement. Implement the following four-input Boolean function as indicated in each of the following subproblems. **(15 Points)**

$$f(a, b, c, d) = \sum m(1, 2, 3, 5, 8, 13) + \sum d(0, 7, 10, 14).$$

3.1. Implement the function using a minimal network of 8:1 multiplexers. **(5 Points: Completion)**

a	b	c	$d = 0$	$d = 1$	$D(d)$
0	0	0	X	1	1
0	0	1	1	1	1
0	1	0	0	1	d
0	1	1	0	X	0
1	0	0	1	0	d'
1	0	1	X	0	0
1	1	0	0	1	d
1	1	1	X	0	0

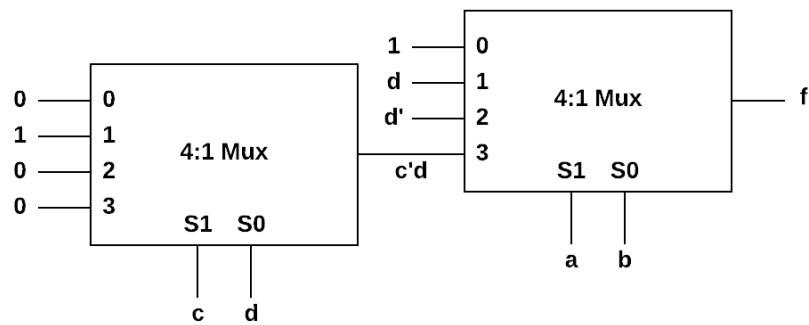


3.2. Implement the function using a minimal network of 4:1 multiplexers. **(5 Points: Completion)**

a	b	$cd = 00$	$cd = 01$	$cd = 10$	$cd = 11$	$D(c, d)$
0	0	X	1	1	1	1
0	1	0	1	0	X	d
1	0	1	0	X	0	d'
1	1	0	1	X	0	$c'd$

For the case of $ab = 11$:

c	d	D
0	0	0
0	1	1
1	0	0
1	1	0

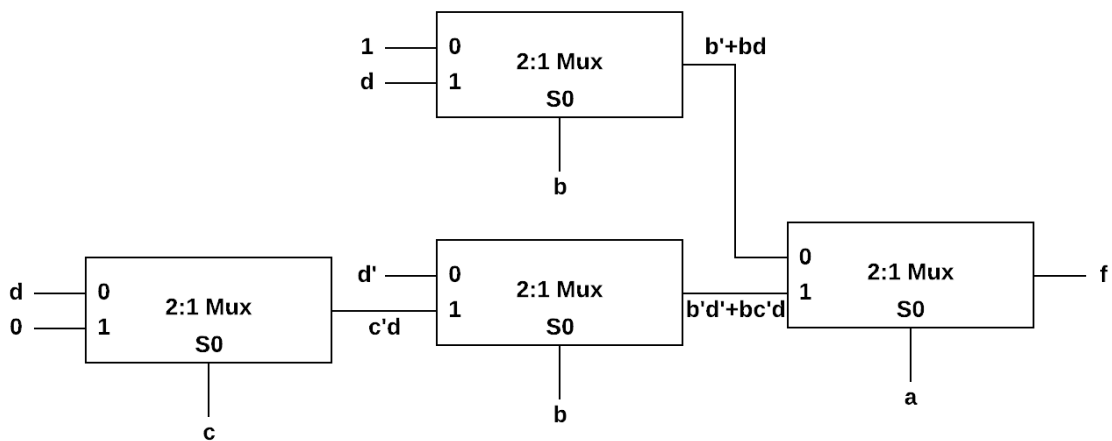


3.3. Implement the function using a minimal network of 2:1 multiplexers. **(5 Points: Completion)**

a	b	$cd = 00$	$cd = 01$	$cd = 10$	$cd = 11$	$D(c, d)$
0	0	X	1	1	1	1
0	1	0	1	0	X	d
1	0	1	0	X	0	d'
1	1	0	1	X	0	$c'd$

For the case of $ab = 11$:

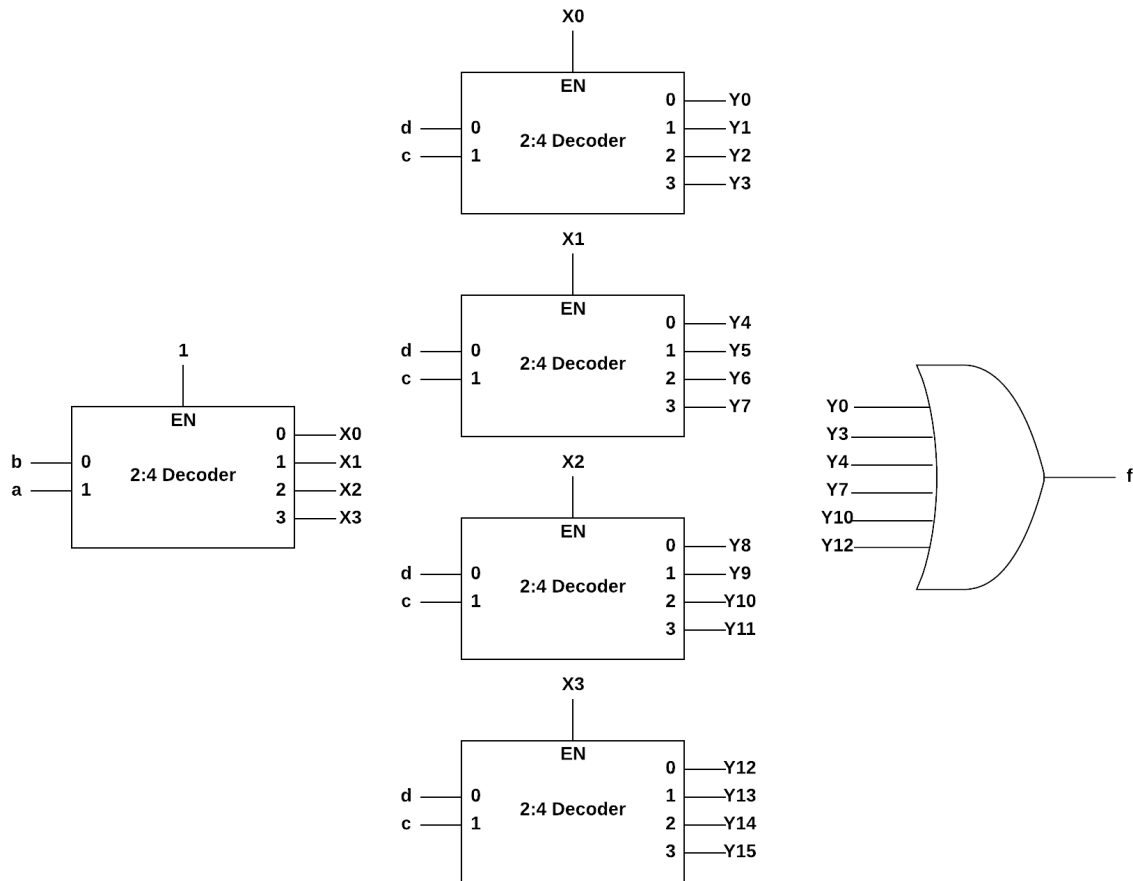
c	$d = 0$	$d = 1$	$D(d)$
0	0	1	d
1	X	0	0



4. Assume a dual-railed system, where you have access to any variable and its complement. Given a four-input Boolean function **(35 Points)**

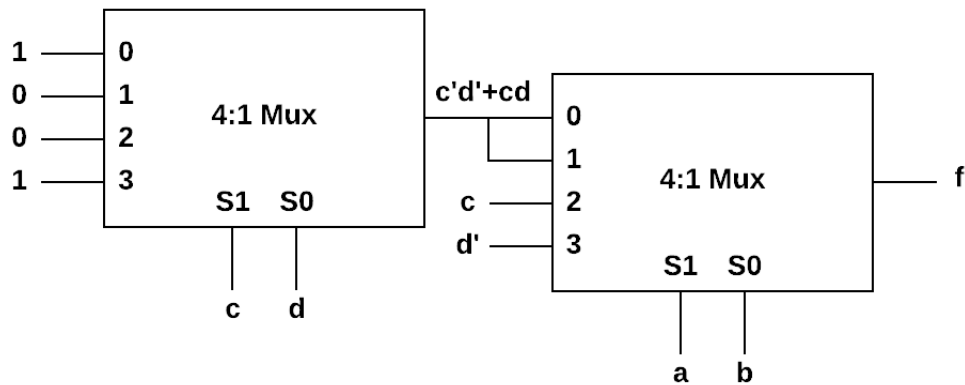
$$f(a, b, c, d) = \sum m(0, 3, 4, 7, 10, 12) + \sum d(5, 11, 14).$$

4.1. Implement the function using a minimal network of 2:4 decoders and OR gates. **(5 Points: Completion)**

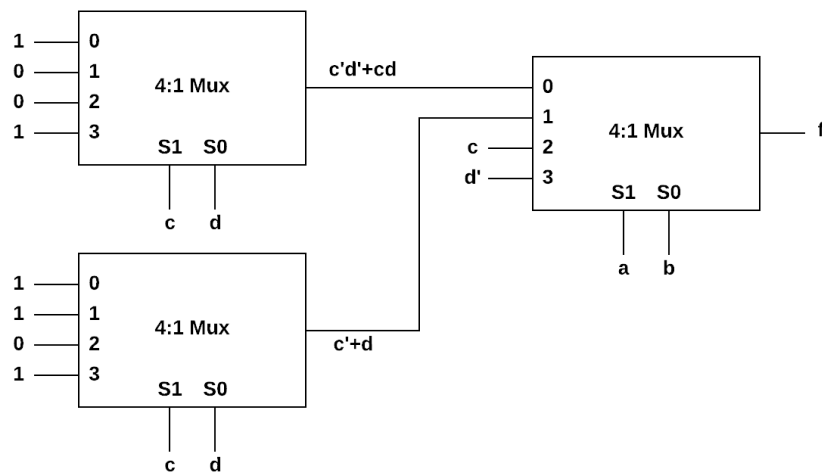


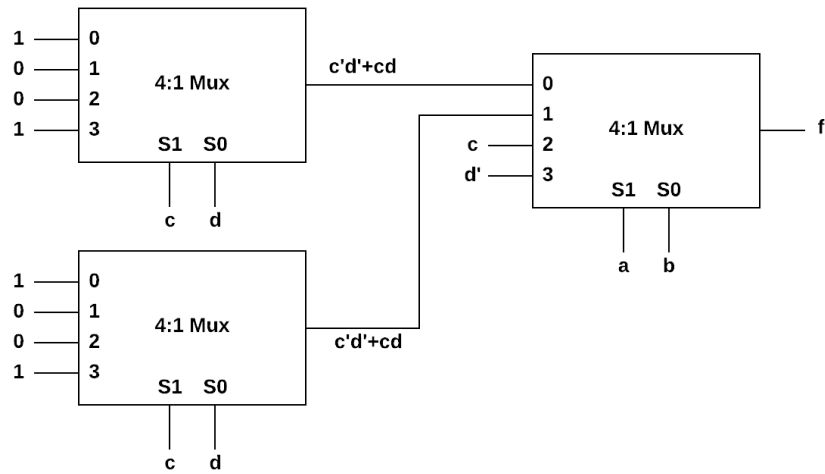
4.2. Implement the function using a minimal network of 4:1 multiplexers. **(25 Points: Correctness)**

a	b	$cd = 00$	$cd = 01$	$cd = 10$	$cd = 11$	$D(c, d)$
0	0	1	0	0	1	$c'd' + cd$
0	1	1	X	0	1	$c'd' + cd$
1	0	0	0	1	X	c
1	1	1	0	X	0	d'



Note that for the case of $ab = 01$, we could use the don't-care for $cd = 01$ (by letting $X = 1$) to simplify $D(c, d)$ to $c' + d$, but letting $X = 0$ makes the overall network minimal because we can reuse the multiplexer of the $ab = 00$ case. That said, either of the following answers would be acceptable as well:





4.3. Implement the function using a minimal network of 2:1 multiplexers. **(5 Points: Completion)**

a	bcd								$D(b, c, d)$
	000	001	010	011	100	101	110	111	
0	1	0	0	1	1	X	0	1	D_0
1	0	0	1	X	1	0	X	0	D_1

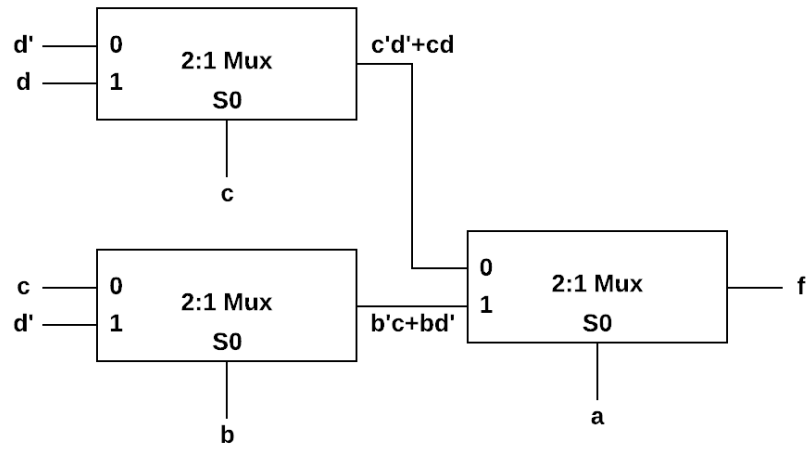
Here is the K-map for D_0 , which gives $D_0 = c'd' + cd$:

		cd			
		00	01	11	10
b	0	1	0	1	0
	1	1	X	1	0

Here is the K-map for D_1 , which gives $D_1 = bd' + cd'$ or $D_1 = bd' + b'c$:

		cd			
		00	01	11	10
b	0	0	0	X	1
	1	1	0	0	X

To get the minimal circuit, we choose $D_1 = bd' + b'c$:



Choosing $D_1 = bd' + cd'$ results in an extra multiplexer.

b	$cd = 00$	$cd = 01$	$cd = 10$	$cd = 11$	$D(c, d)$
0	0	0	1	0	cd'
1	1	0	1	0	d'

