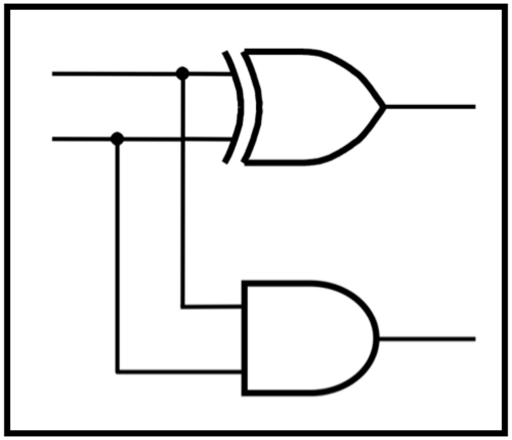


CprE 2810: Digital Logic

Instructor: Alexander Stoytchev

http://www.ece.iastate.edu/~alexs/classes/

Multiplexers

CprE 2810: Digital Logic
Iowa State University, Ames, IA

Administrative Stuff

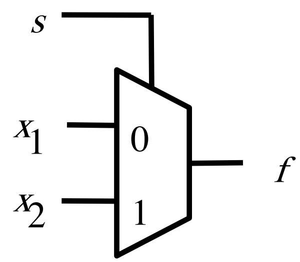
- HW 6 is due on Monday Oct 13 @ 10pm
- HW 7 is due on Monday Oct 20 @ 10pm
- Next week: Lab 6 + TTL2
- Midterm progress report grades are due next week

2-to-1 Multiplexer

2-to-1 Multiplexer (Definition)

- Has two inputs: x_1 and x_2
- Also has another input line s
- If s=0, then the output is equal to x_1
- If s=1, then the output is equal to x_2

Graphical Symbol for a 2-to-1 Multiplexer

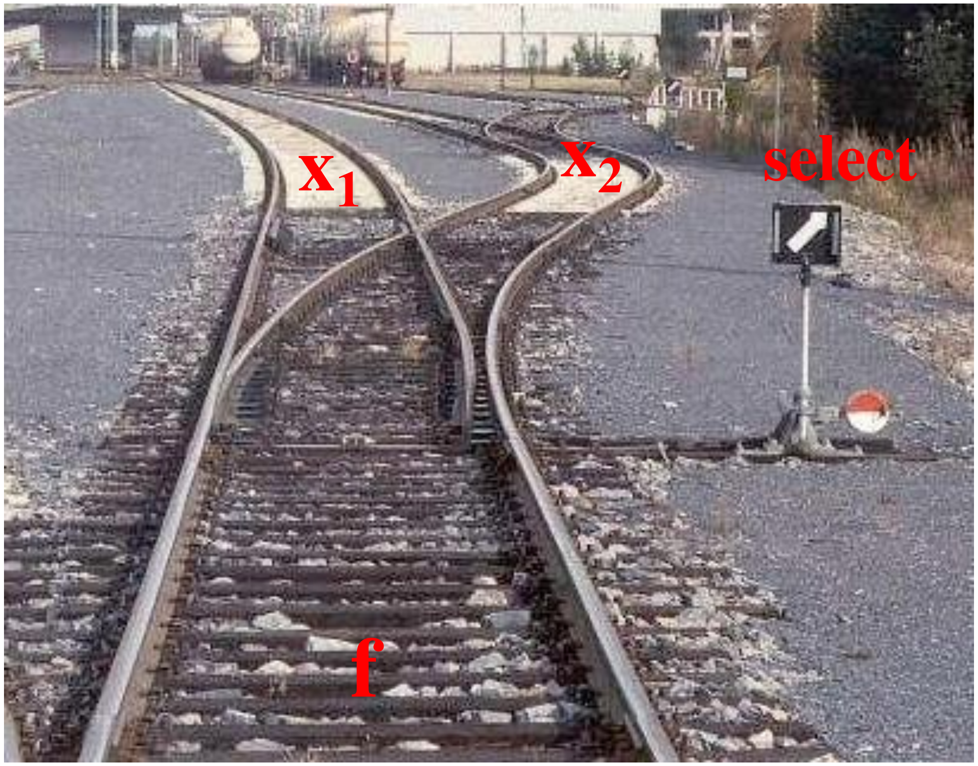


[Figure 2.33c from the textbook]

Analogy: Railroad Switch

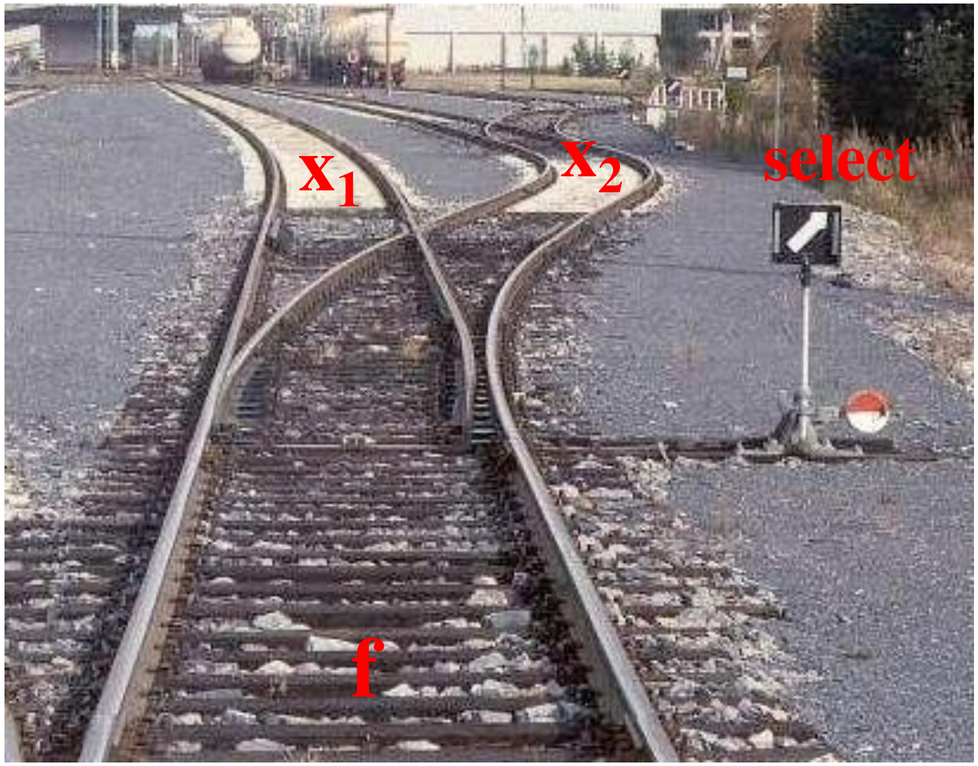
http://en.wikipedia.org/wiki/Railroad_switch]

Analogy: Railroad Switch



http://en.wikipedia.org/wiki/Railroad_switch]

Analogy: Railroad Switch



This is not a perfect analogy because the trains can go in either direction, while the multiplexer would only allow them to go from top to bottom.

http://en.wikipedia.org/wiki/Railroad_switch]

Truth Table for a 2-to-1 Multiplexer

$s\ x_1\ x_2$	$f(s, x_1, x_2)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

[Figure 2.33a from the textbook]

Let's Derive the SOP form

$s\, x_1\, x_2$	$f(s, x_1, x_2)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

Let's Derive the SOP form

$s\ x_1\ x_2$	$f(s, x_1, x_2)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

Let's Derive the SOP form

$s\ x_1\ x_2$	$f(s, x_1, x_2)$
0 0 0	0
0 0 1	0
0 1 0	$\boxed{1}$
0 1 1	$\boxed{1}$
1 0 0	0
1 0 1	$\boxed{1}$
1 1 0	0
1 1 1	$\boxed{1}$

Where should we
put the negation signs?

$s\ x_1\ x_2$

$s\ x_1\ x_2$

$s\ x_1\ x_2$

$s\ x_1\ x_2$

Let's Derive the SOP form

$s\ x_1\ x_2$	$f(s, x_1, x_2)$	
0 0 0	0	
0 0 1	0	
0 1 0	$\boxed{1}$	$\overline{s}\, x_1\, \overline{x_2}$
0 1 1	$\boxed{1}$	$\overline{s}\, x_1\, x_2$
1 0 0	0	
1 0 1	$\boxed{1}$	$s\, \overline{x_1}\, x_2$
1 1 0	0	
1 1 1	$\boxed{1}$	$s\, x_1\, x_2$

Let's Derive the SOP form

$s\ x_1\ x_2$	$f(s, x_1, x_2)$	
0 0 0	0	
0 0 1	0	
0 1 0	$\boxed{1}$	$\overline{s}\, x_1\, \overline{x_2}$
0 1 1	$\boxed{1}$	$\overline{s}\, x_1\, x_2$
1 0 0	0	
1 0 1	$\boxed{1}$	$s\, \overline{x_1}\, x_2$
1 1 0	0	
1 1 1	$\boxed{1}$	$s\, x_1\, x_2$

$$f(s, x_1, x_2) = \overline{s}\, x_1\, \overline{x_2} + \overline{s}\, x_1\, x_2 + s\, \overline{x_1}\, x_2 + s\, x_1\, x_2$$

Let’s simplify this expression

$$f(s, x_1, x_2) = \overline{s}\, x_1 \overline{x_2} + \overline{s}\, x_1 x_2 + s\, \overline{x_1}\, x_2 + s\, x_1 x_2$$

Let's simplify this expression

$$f(s, x_1, x_2) = \overline{s}\, x_1\, \overline{x_2} + \overline{s}\, x_1\, x_2 + s\, \overline{x_1}\, x_2 + s\, x_1\, x_2$$

$$f(s, x_1, x_2) = \overline{s}\, x_1\, (\overline{x_2} + x_2) + s\, (\overline{x_1} + x_1)\, x_2$$

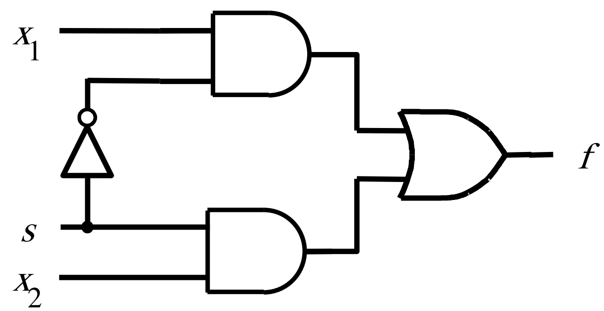
Let's simplify this expression

$$f(s, x_1, x_2) = \overline{s}\, x_1\, \overline{x_2} + \overline{s}\, x_1\, x_2 + s\, \overline{x_1}\, x_2 + s\, x_1\, x_2$$

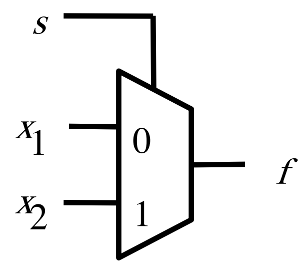
$$f(s, x_1, x_2) = \overline{s}\, x_1\, (\overline{x_2} + x_2) + s\, (\overline{x_1} + x_1)\, x_2$$

$$f(s, x_1, x_2) = \overline{s}\, x_1 + s\, x_2$$

Circuit for 2-1 Multiplexer



(b) Circuit

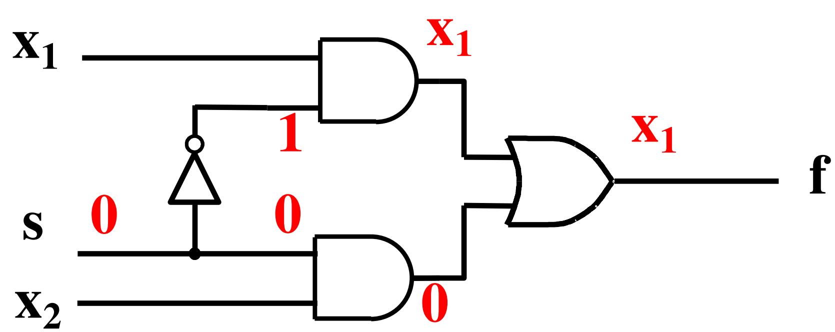


(c) Graphical symbol

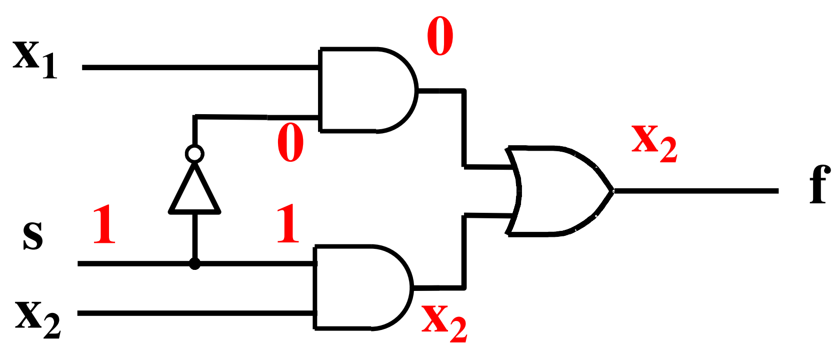
$$f(s, x_1, x_2) = \overline{s}\, x_1 + s\, x_2$$

[Figure 2.33b-c from the textbook]

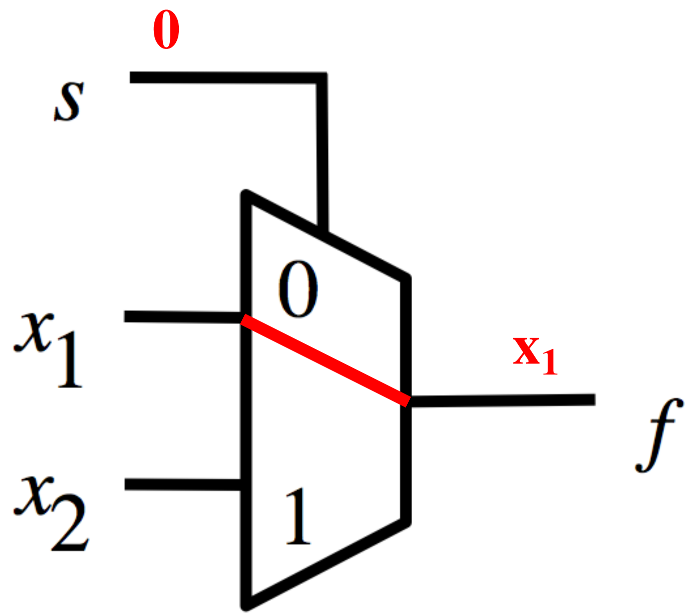
Analysis of the 2-to-1 Multiplexer (when the input s=0)



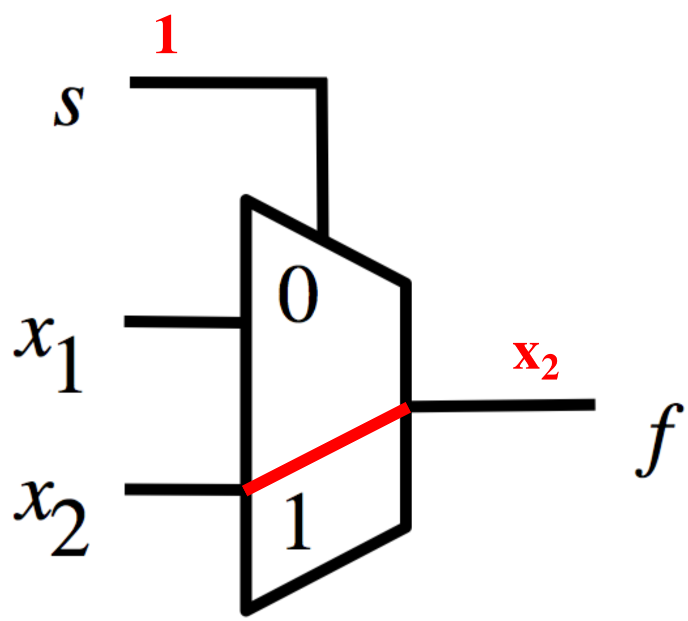
Analysis of the 2-to-1 Multiplexer (when the input s=1)



Analysis of the 2-to-1 Multiplexer (when the input s=0)



Analysis of the 2-to-1 Multiplexer (when the input s=1)



More Compact Truth-Table Representation

$s\ x_1\ x_2$	$f(s, x_1, x_2)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

(a)Truth table

s	$f(s, x_1, x_2)$
0	x_1
1	x_2

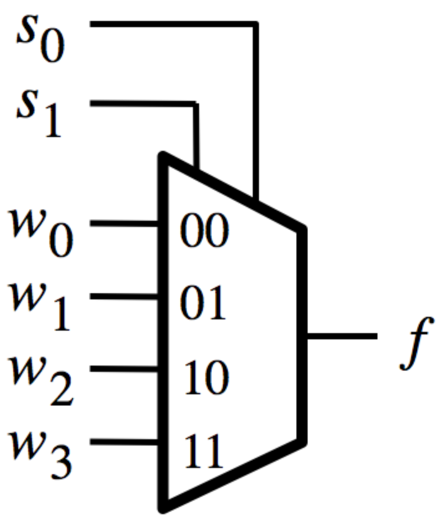
[Figure 2.33 from the textbook]

4-to-1 Multiplexer

4-to-1 Multiplexer (Definition)

- Has four inputs: w_0 , w_1, w_2, w_3
- Also has two select lines: s_1 and s_0
- If s_1=0 and s_0=0, then the output f is equal to w_0
- If s_1=0 and s_0=1, then the output f is equal to w_1
- If s_1=1 and s_0=0, then the output f is equal to w_2
- If s_1=1 and s_0=1, then the output f is equal to w_3

Graphical Symbol and Truth Table



(a) Graphic symbol

s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

(b) Truth table

[Figure 4.2a-b from the textbook]

The long-form truth table

The long-form truth table

S_1	S_0	I_3	I_2	I_1	I_0	F
0	0	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	0
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	1
		1	0	1	0	0
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	1
		1	1	1	0	0
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
0	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	1
		0	0	1	1	1
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	0
		1	1	1	0	1
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
1	0	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	1
		0	1	0	1	1
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	0
		1	0	1	1	0
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
1	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	0
		0	1	1	1	0
		1	0	0	0	1
		1	0	0	1	1
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

[http://www.absoluteastronomy.com/topics/Multiplexer]

The long-form truth table

S_1	S_0	I_3	I_2	I_1	I_0	F
0	0	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	0
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	1
		1	0	1	0	0
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	1
		1	1	1	0	0
		1	1	1	1	1

identical

S_1	S_0	I_3	I_2	I_1	I_0	F
0	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	1
		0	0	1	1	1
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	0
		1	1	1	0	1
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
1	0	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	1
		0	1	0	1	1
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	0
		1	0	1	1	0
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
1	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	0
		0	1	1	1	0
		1	0	0	0	1
		1	0	0	1	1
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

[http://www.absoluteastronomy.com/topics/Multiplexer]

The long-form truth table

S_1	S_0	I_3	I_2	I_1	I_0	F
0	0	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	0
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	1
		1	0	1	0	0
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	1
		1	1	1	0	0
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
0	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	1
		0	0	1	1	1
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	0
		1	1	1	0	1
		1	1	1	1	1

identical

S_1	S_0	I_3	I_2	I_1	I_0	F
1	0	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	1
		0	1	0	1	1
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	0
		1	0	1	1	0
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
1	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	0
		0	1	1	1	0
		1	0	0	0	1
		1	0	0	1	1
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

[http://www.absoluteastronomy.com/topics/Multiplexer]

The long-form truth table

S_1	S_0	I_3	I_2	I_1	I_0	F
0	0	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	0
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	1
		1	0	1	0	0
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	1
		1	1	1	0	0
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
0	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	1
		0	0	1	1	1
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	0
		1	1	1	0	1
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
1	0	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	1
		0	1	0	1	1
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	0
		1	0	1	1	0
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
1	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	0
		0	1	1	1	0
		1	0	0	0	1
		1	0	0	1	1
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

identical

[http://www.absoluteastronomy.com/topics/Multiplexer]

The long-form truth table

S_1	S_0	I_3	I_2	I_1	I_0	F
0	0	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	0
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	1
		1	0	1	0	0
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	1
		1	1	1	0	0
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
0	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	1
		0	0	1	1	1
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	0
		1	1	1	0	1
		1	1	1	1	1

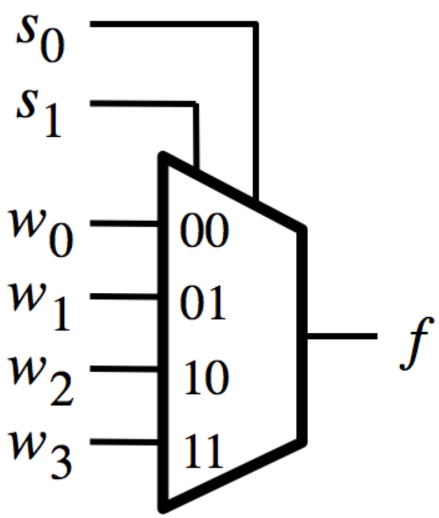
S_1	S_0	I_3	I_2	I_1	I_0	F
1	0	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	1
		0	1	0	1	1
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	0
		1	0	1	1	0
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

S_1	S_0	I_3	I_2	I_1	I_0	F
1	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	0
		0	0	1	1	0
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	0
		0	1	1	1	0
		1	0	0	0	1
		1	0	0	1	1
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	1
		1	1	0	1	1
		1	1	1	0	1
		1	1	1	1	1

identical

[http://www.absoluteastronomy.com/topics/Multiplexer]

Graphical Symbol and Truth Table



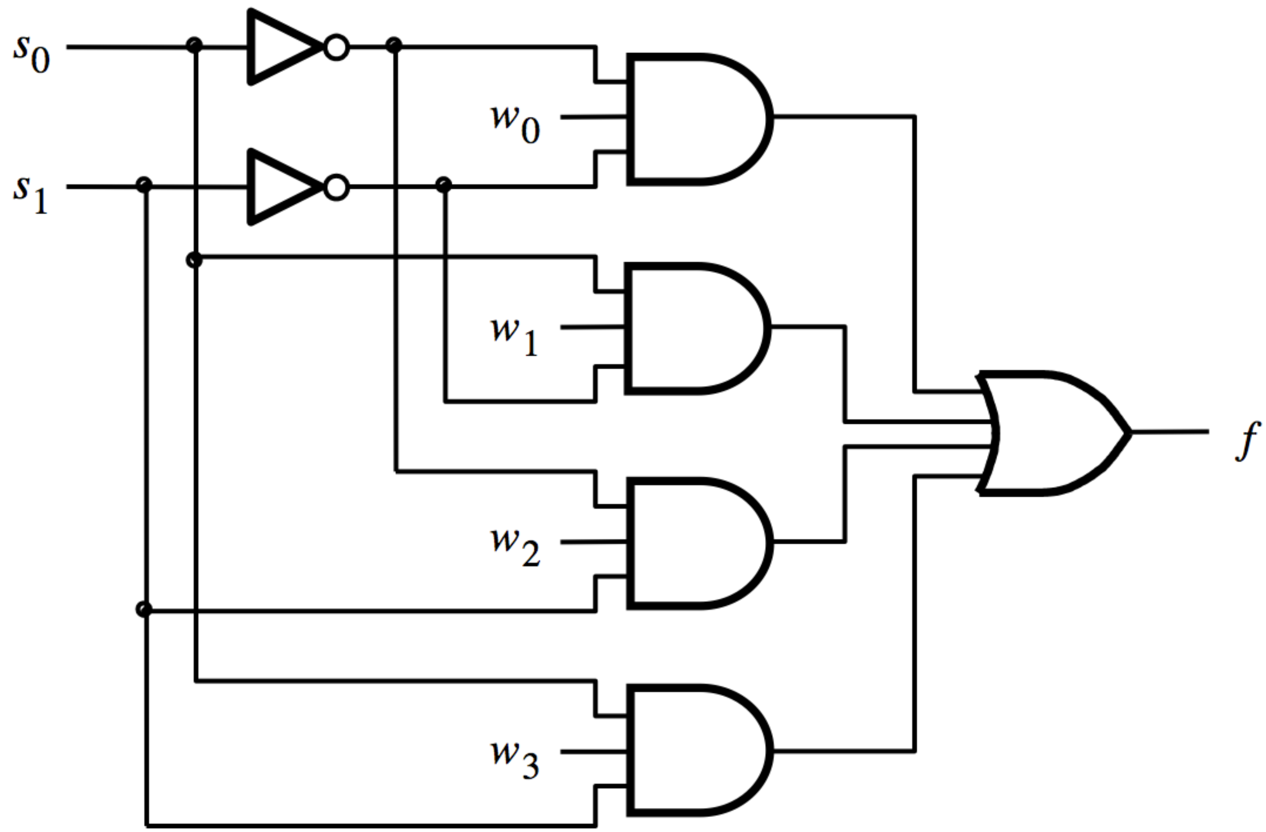
(a) Graphic symbol

s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

(b) Truth table

[Figure 4.2a-b from the textbook]

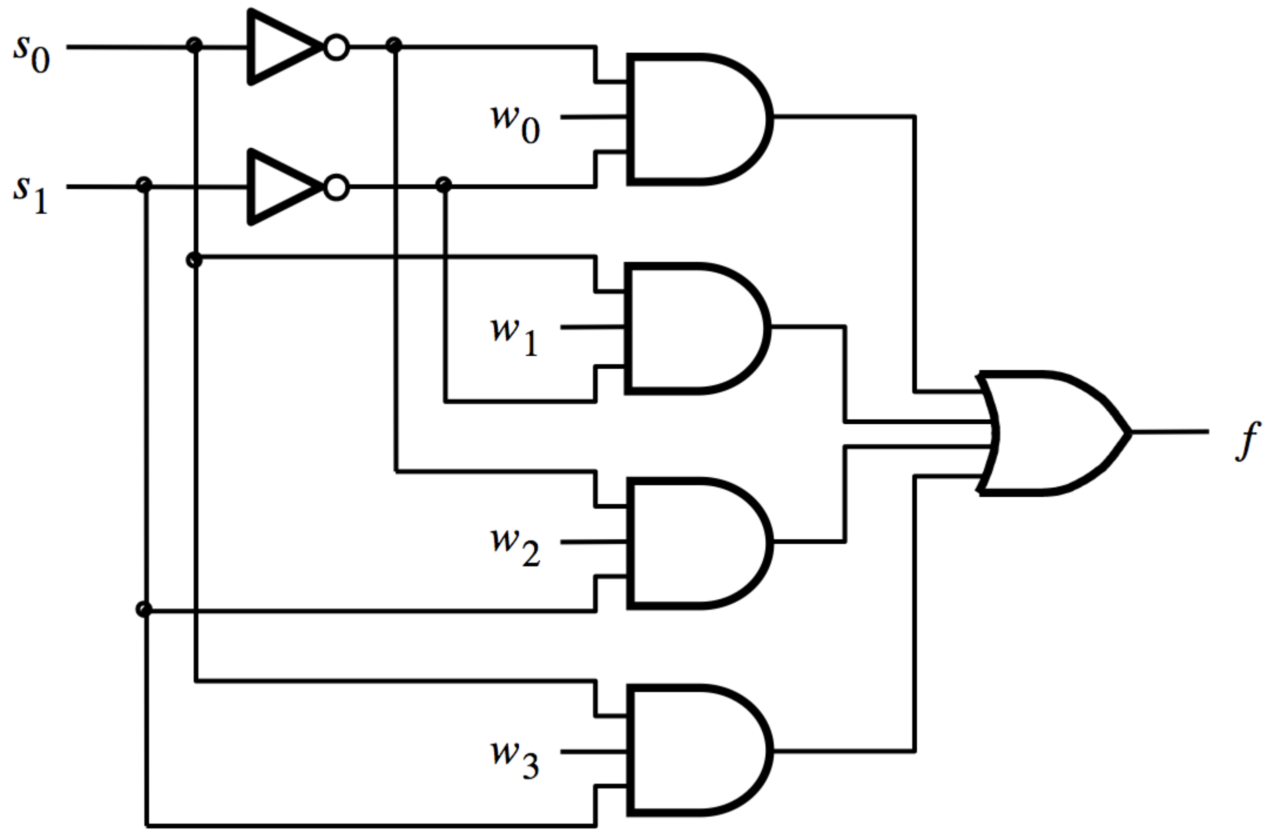
4-to-1 Multiplexer (SOP circuit)



$$f = \overline{s_1}\,\overline{s_0}\,w_0 + \overline{s_1}\,s_0\,w_1 + s_1\,\overline{s_0}\,w_2 + s_1\,s_0\,w_3$$

[Figure 4.2c from the textbook]

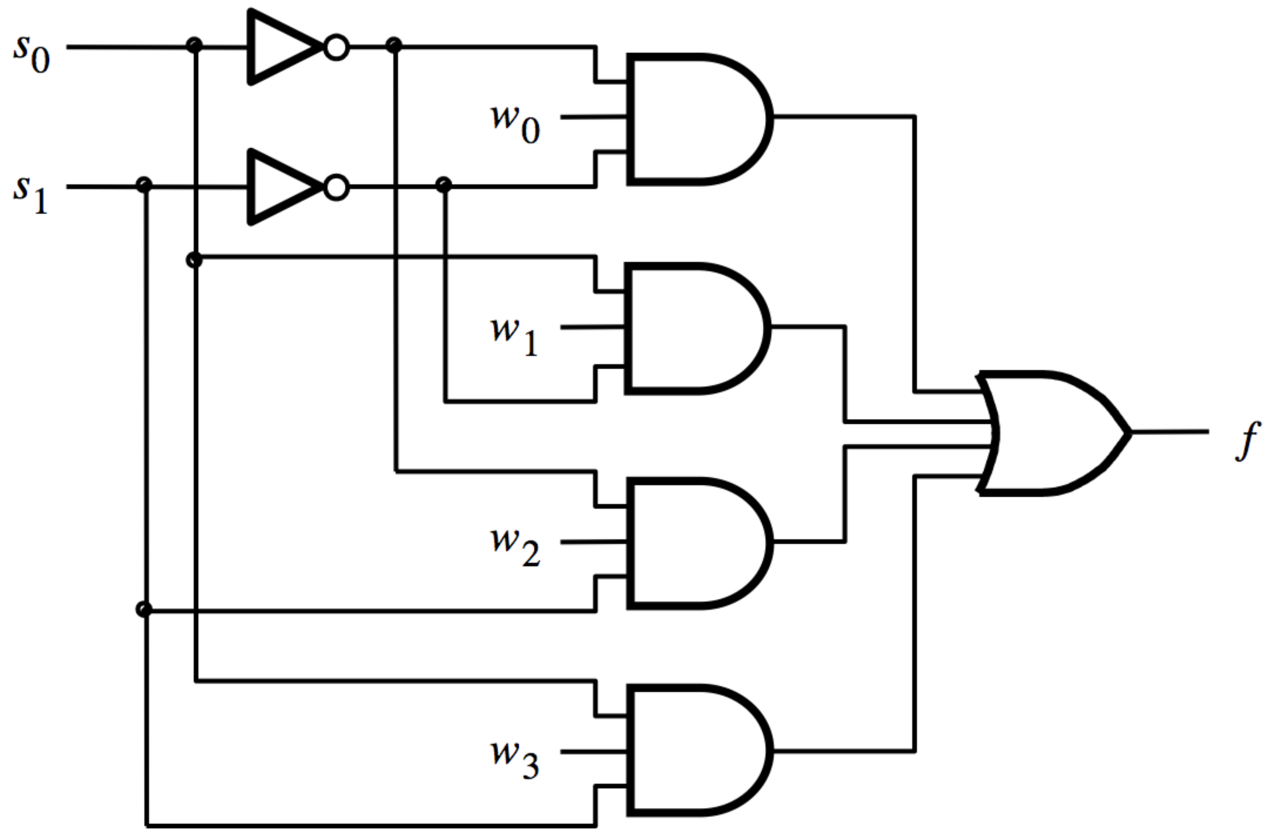
4-to-1 Multiplexer (SOP circuit)



$$f = \overline{s_1}\,\overline{s_0}\,w_0 + \overline{s_1}\,s_0\,w_1 + s_1\,\overline{s_0}\,w_2 + s_1\,s_0\,w_3$$

these are the four minterms in terms of s_1 and s_2

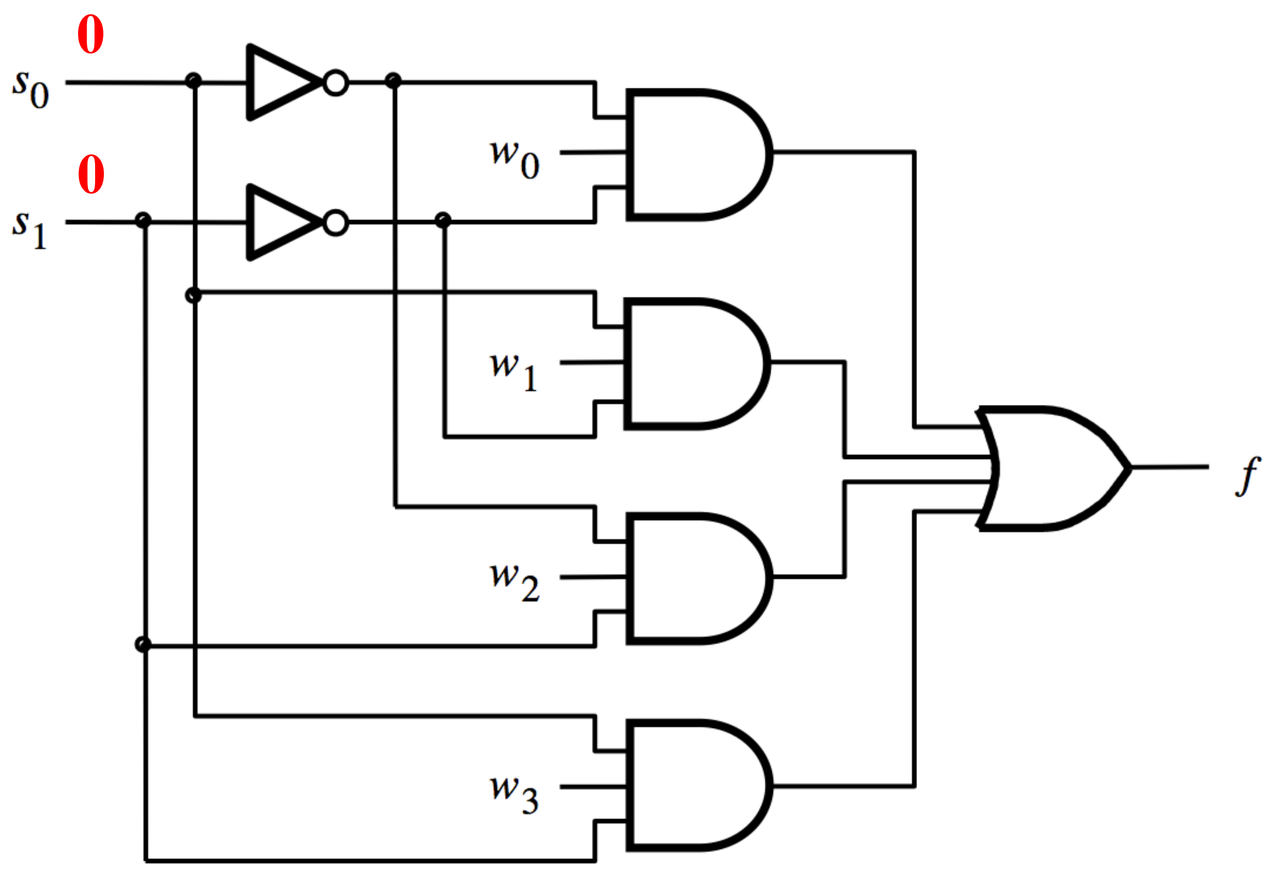
4-to-1 Multiplexer (SOP circuit)



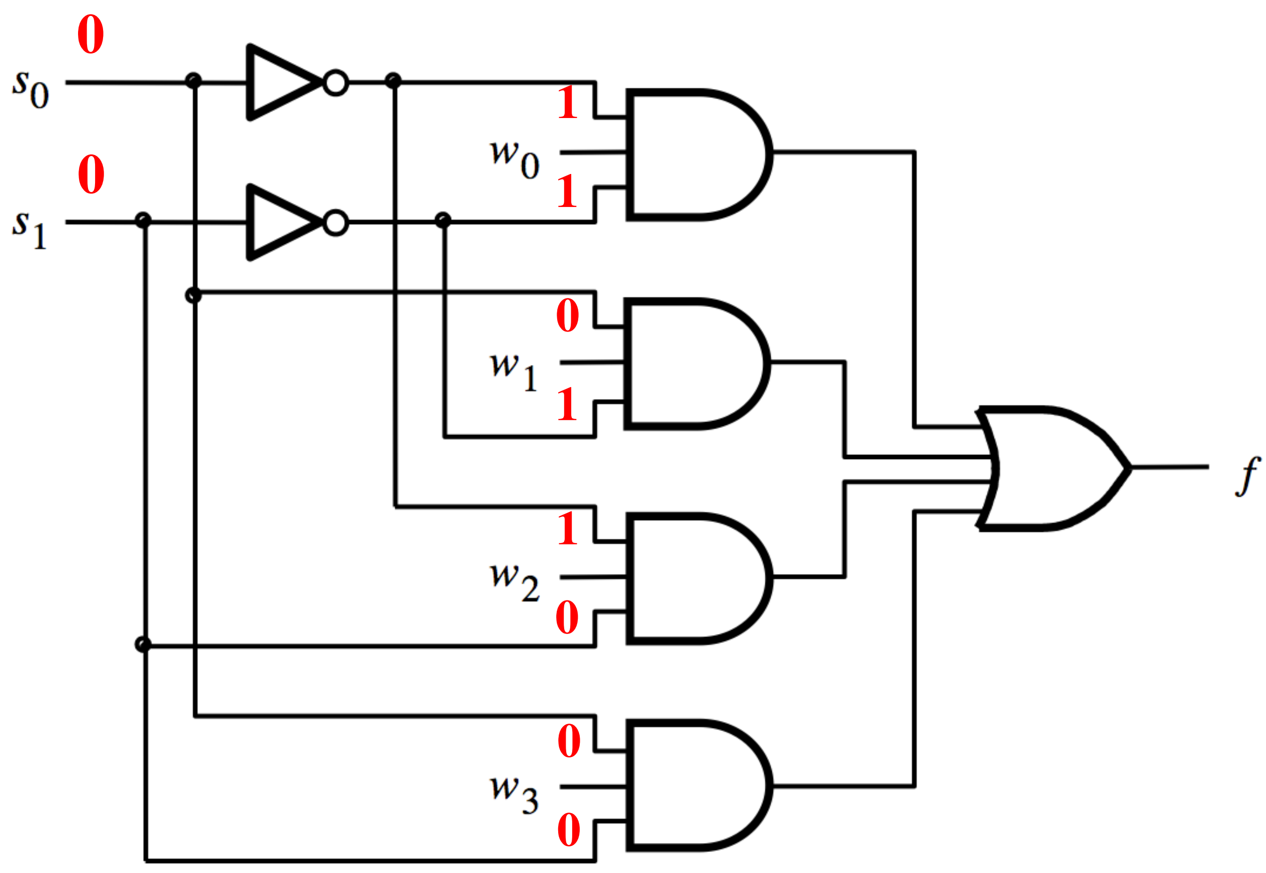
$$f = \overline{s_1}\,\overline{s_0}\,w_0 + \overline{s_1}\,s_0\,w_1 + s_1\,\overline{s_0}\,w_2 + s_1\,s_0\,w_3$$

these are the four w inputs to the 4-to-1 MUX

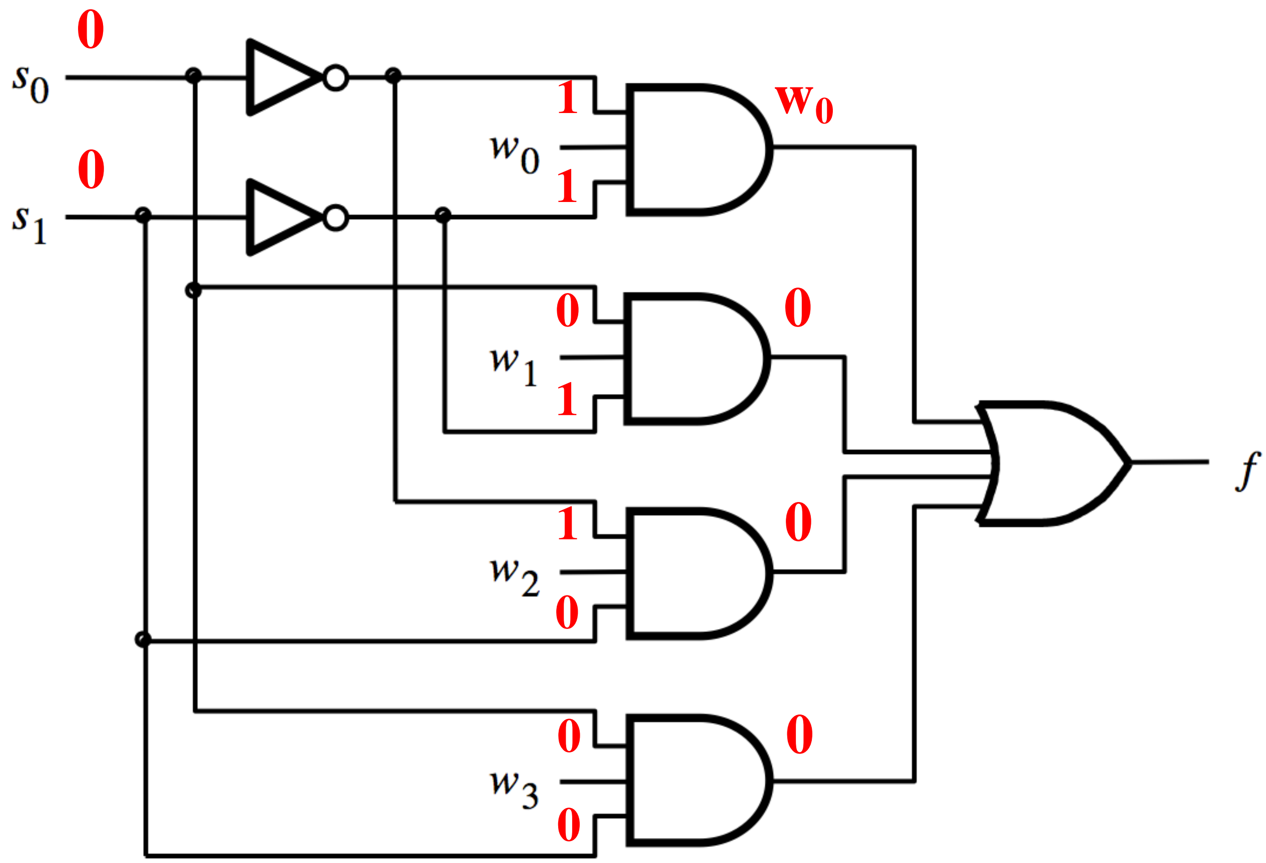
Analysis of the 4-to-1 Multiplexer
(s_1=0 and s_0=0)



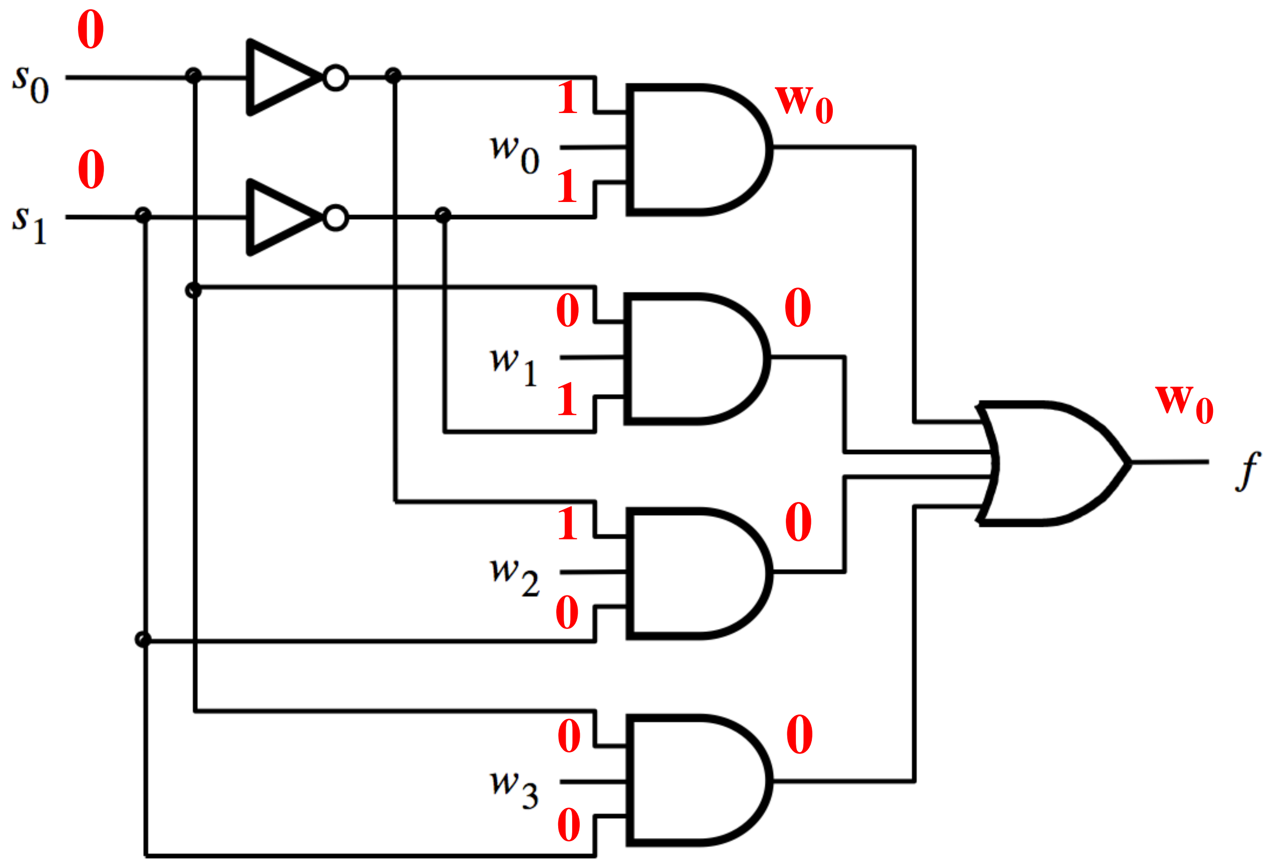
Analysis of the 4-to-1 Multiplexer
(s_1=0 and s_0=0)



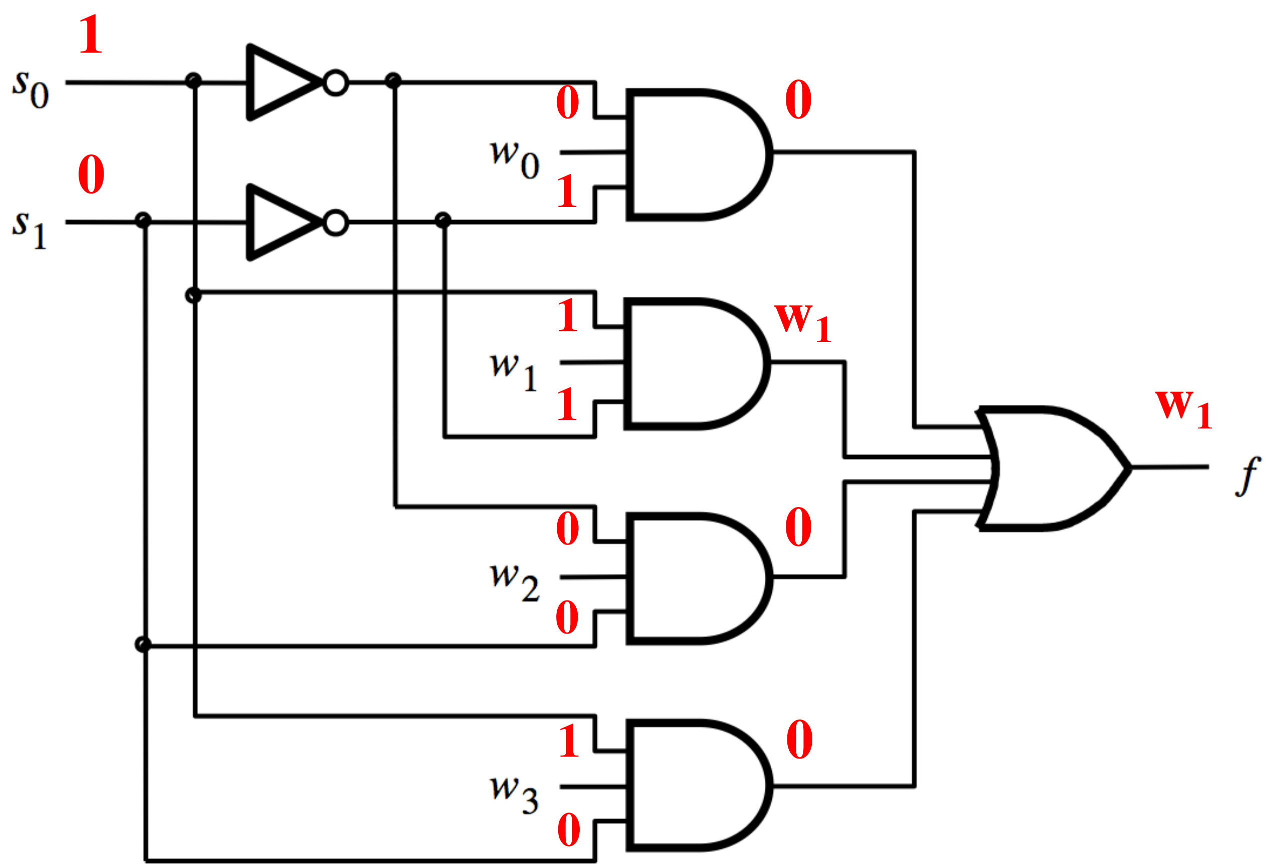
Analysis of the 4-to-1 Multiplexer
(s_1=0 and s_0=0)



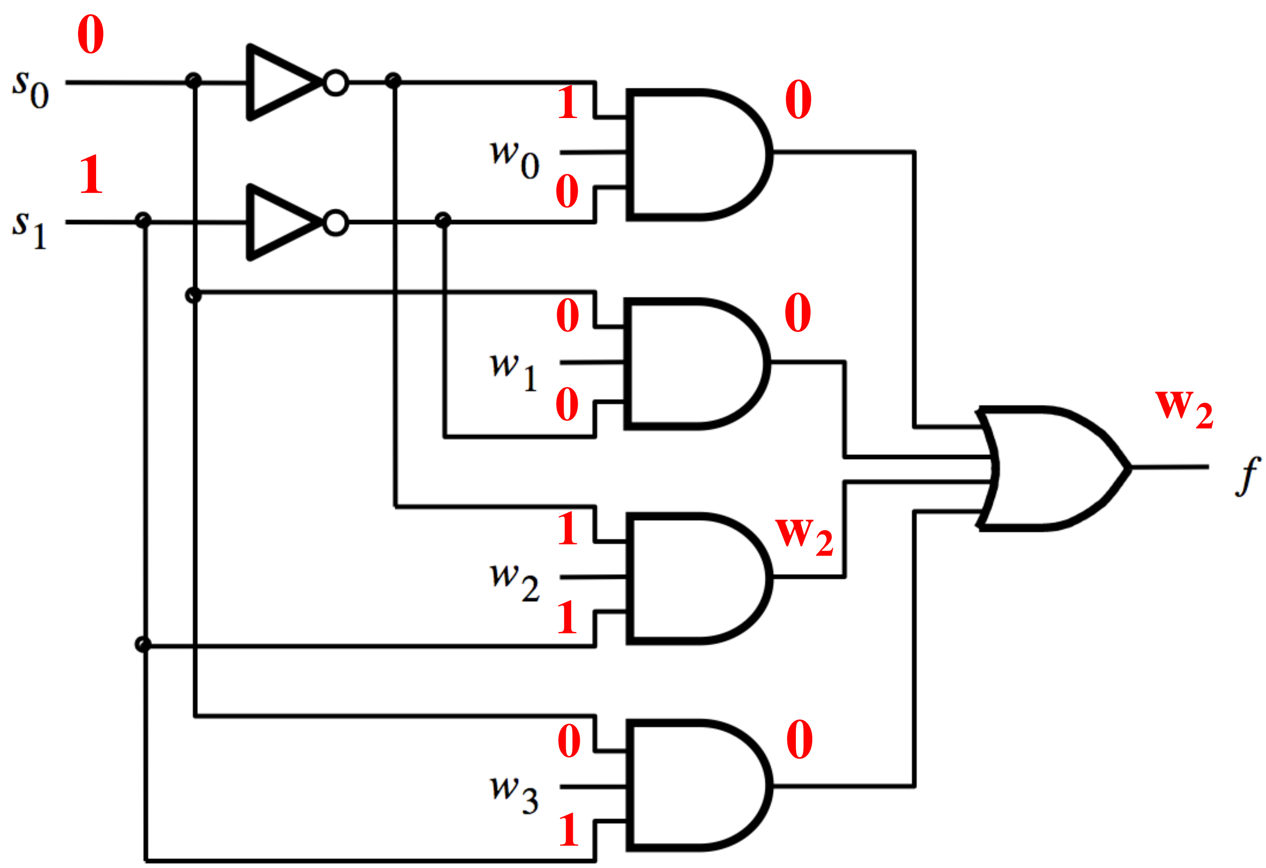
Analysis of the 4-to-1 Multiplexer
(s_1=0 and s_0=0)



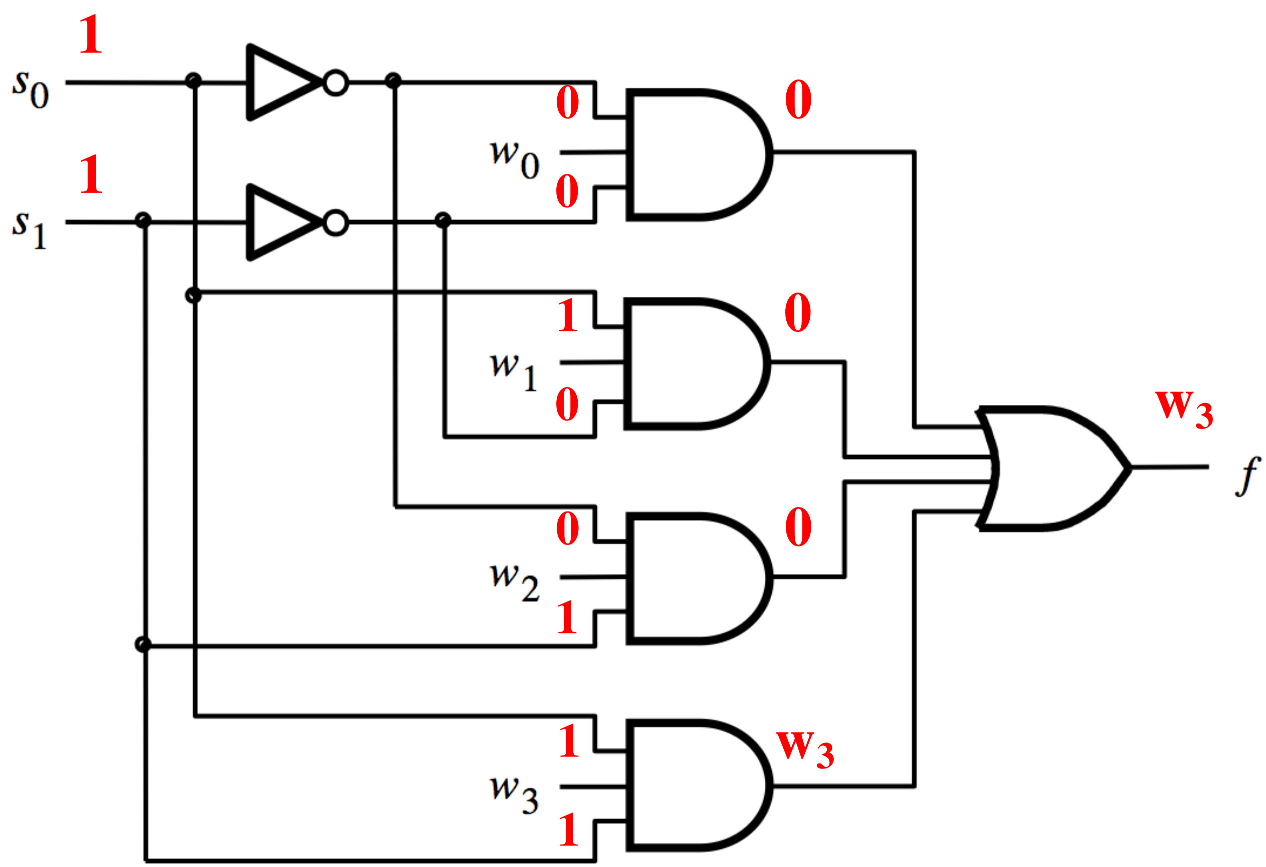
Analysis of the 4-to-1 Multiplexer
(s_1=0 and s_0=1)



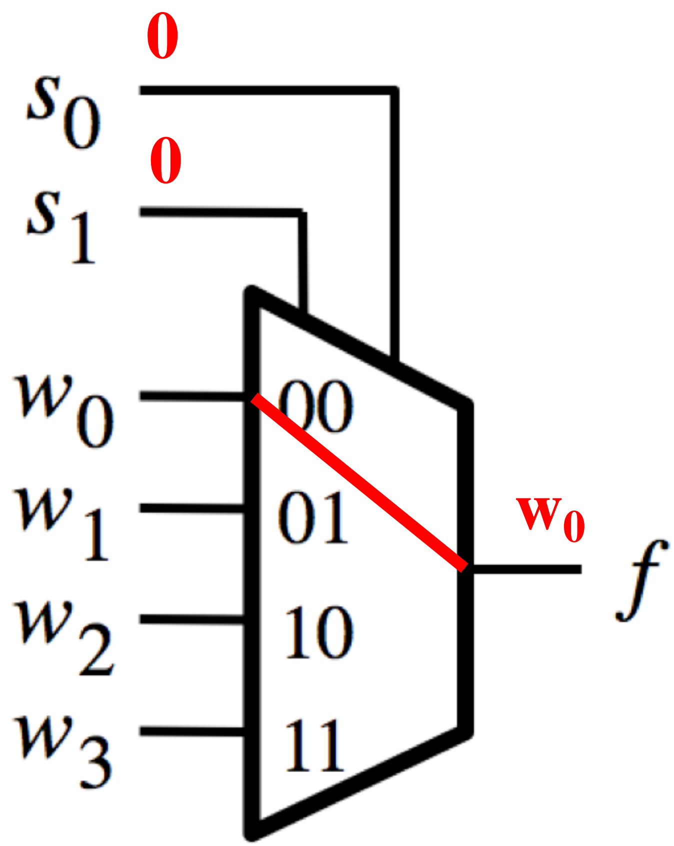
Analysis of the 4-to-1 Multiplexer
(s_1=1 and s_0=0)



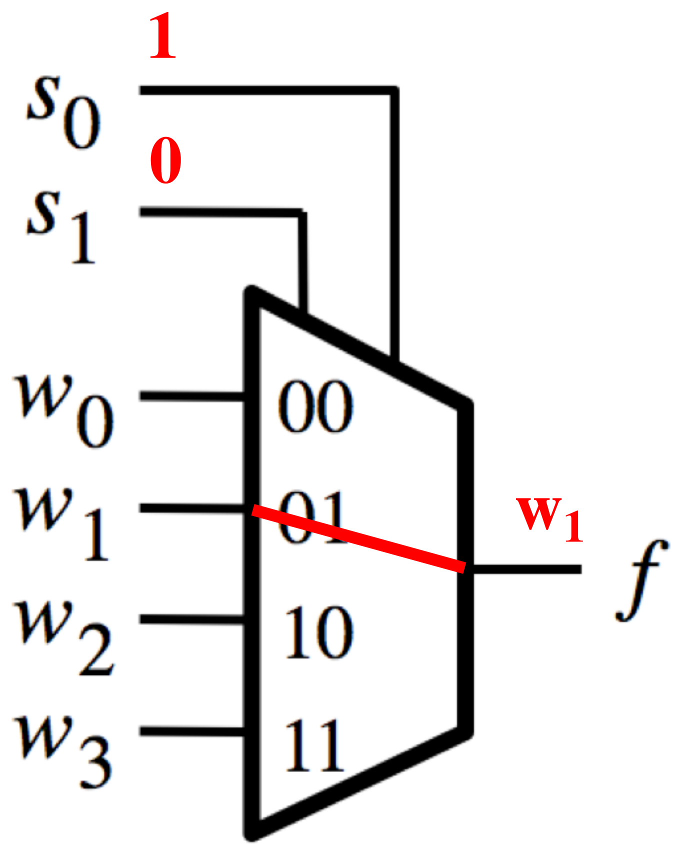
Analysis of the 4-to-1 Multiplexer
(s_1=1 and s_0=1)



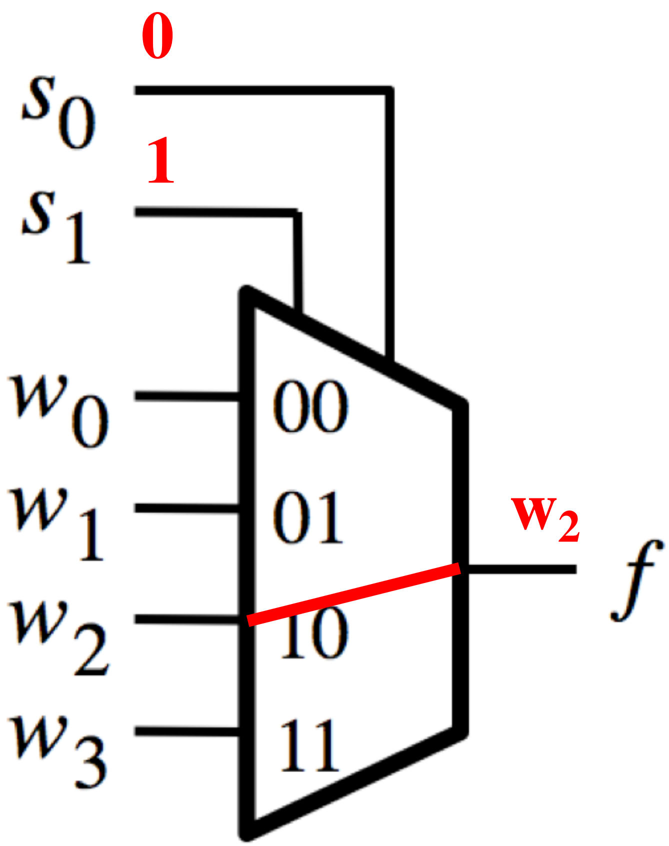
Analysis of the 4-to-1 Multiplexer
(s_1=0 and s_0=0)



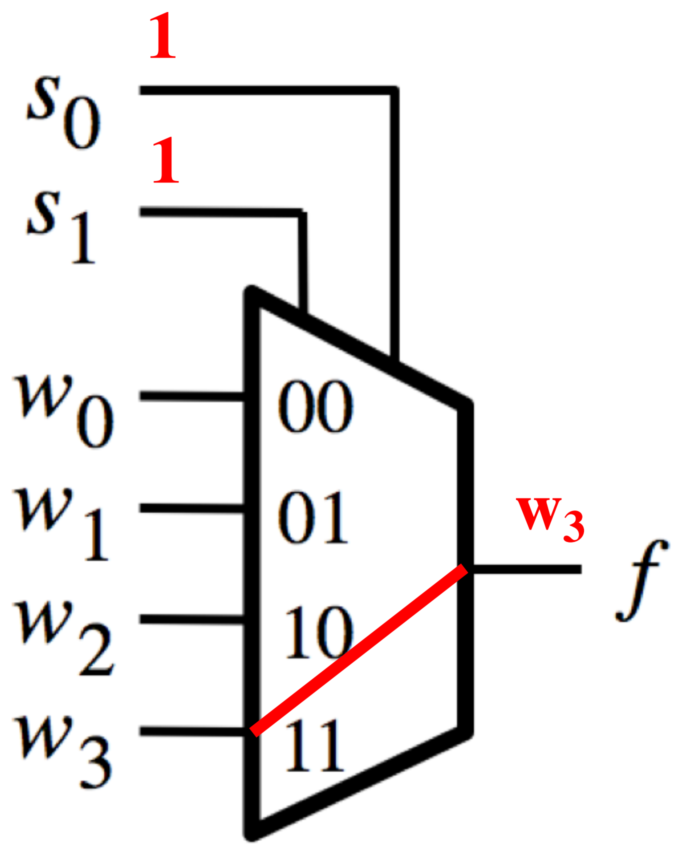
Analysis of the 4-to-1 Multiplexer
(s_1=0 and s_0=1)



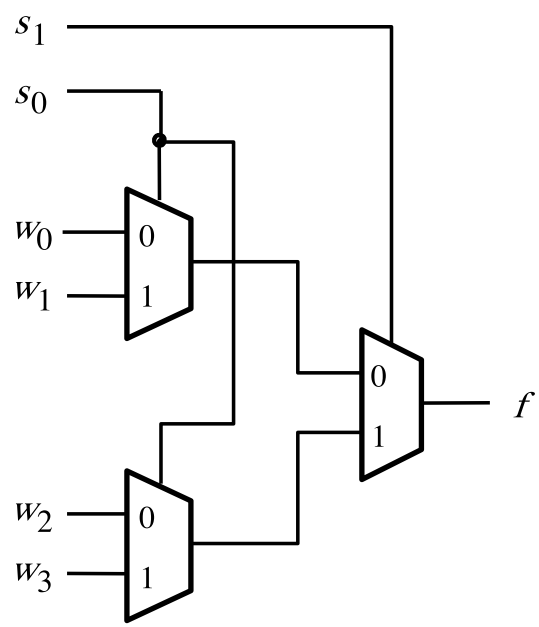
Analysis of the 4-to-1 Multiplexer
(s_1=1 and s_0=0)



Analysis of the 4-to-1 Multiplexer
(s_1=1 and s_0=1)



Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer

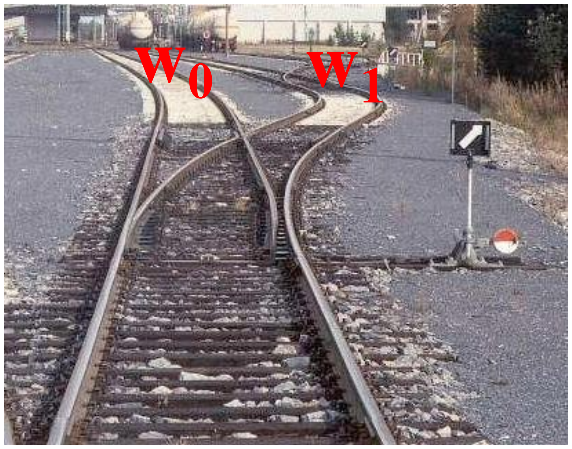


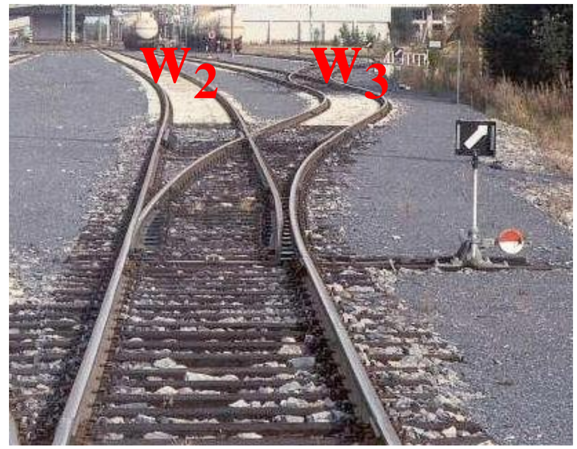
[Figure 4.3 from the textbook]

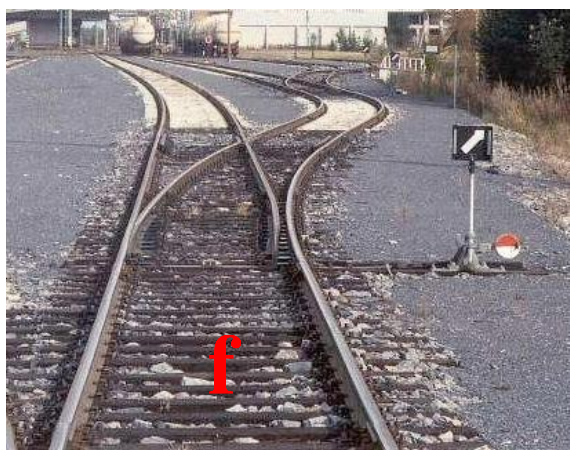
Analogy: Railroad Switches

http://en.wikipedia.org/wiki/Railroad_switch]

Analogy: Railroad Switches



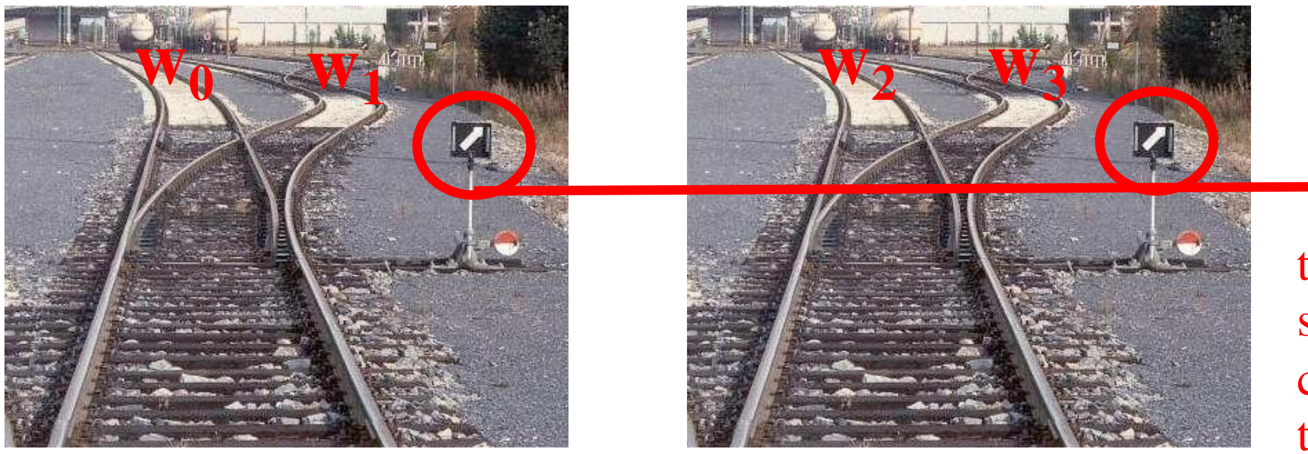




s_1

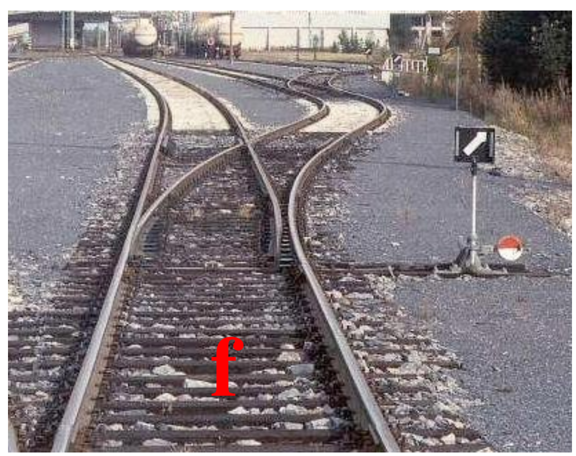
http://en.wikipedia.org/wiki/Railroad_switch]

Analogy: Railroad Switches



s_0

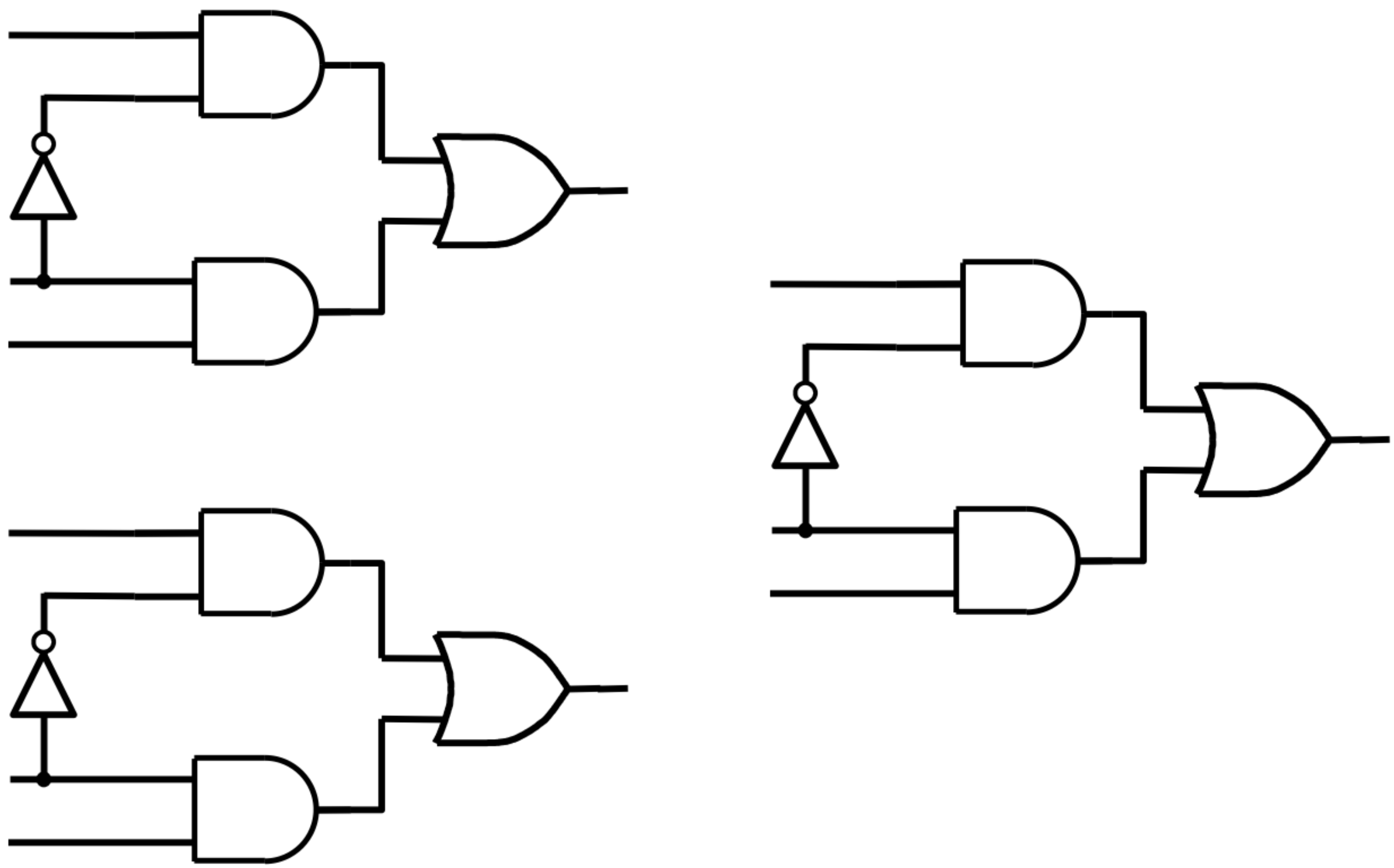
these two switches are controlled together



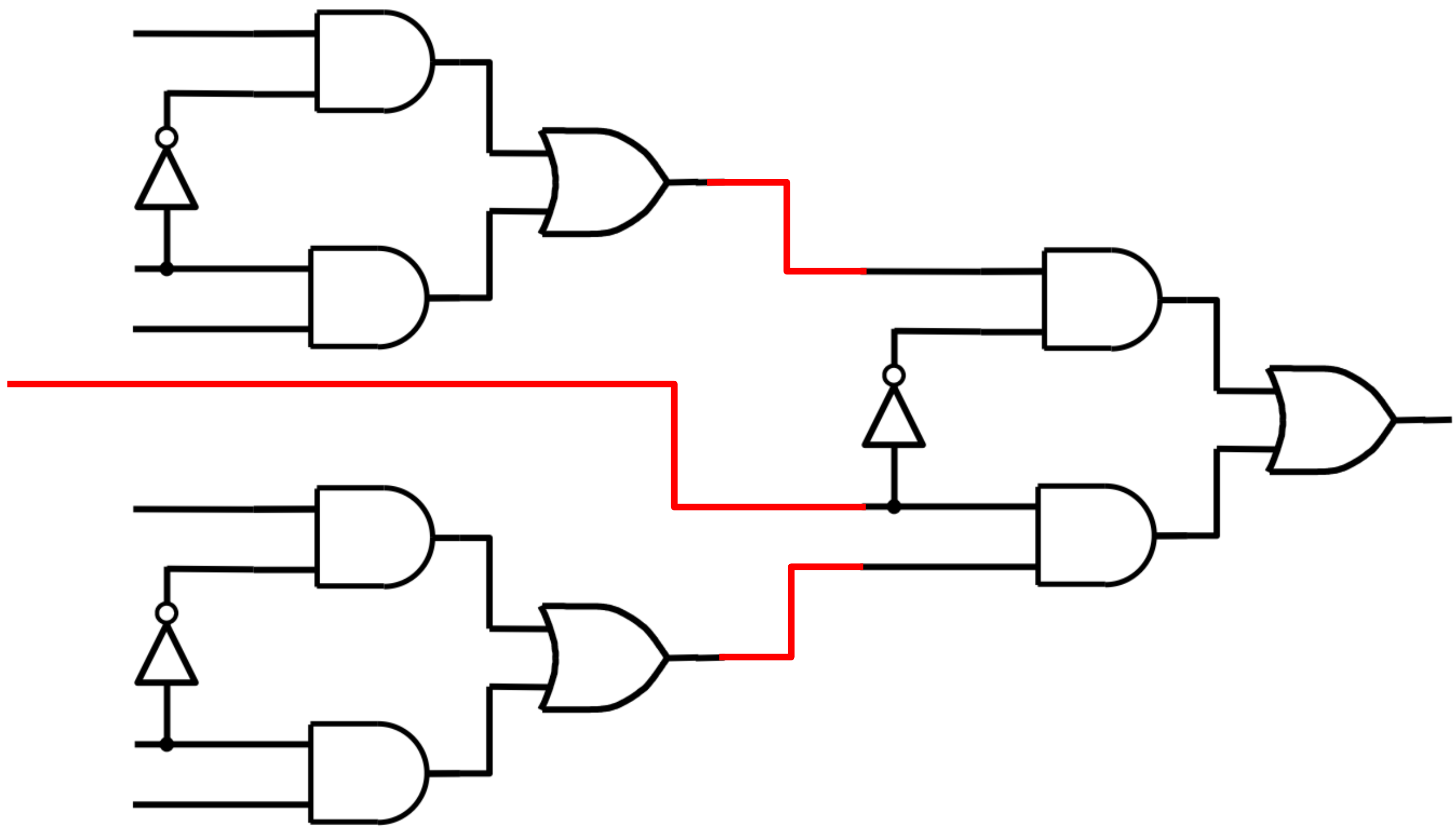
s_1

http://en.wikipedia.org/wiki/Railroad_switch]

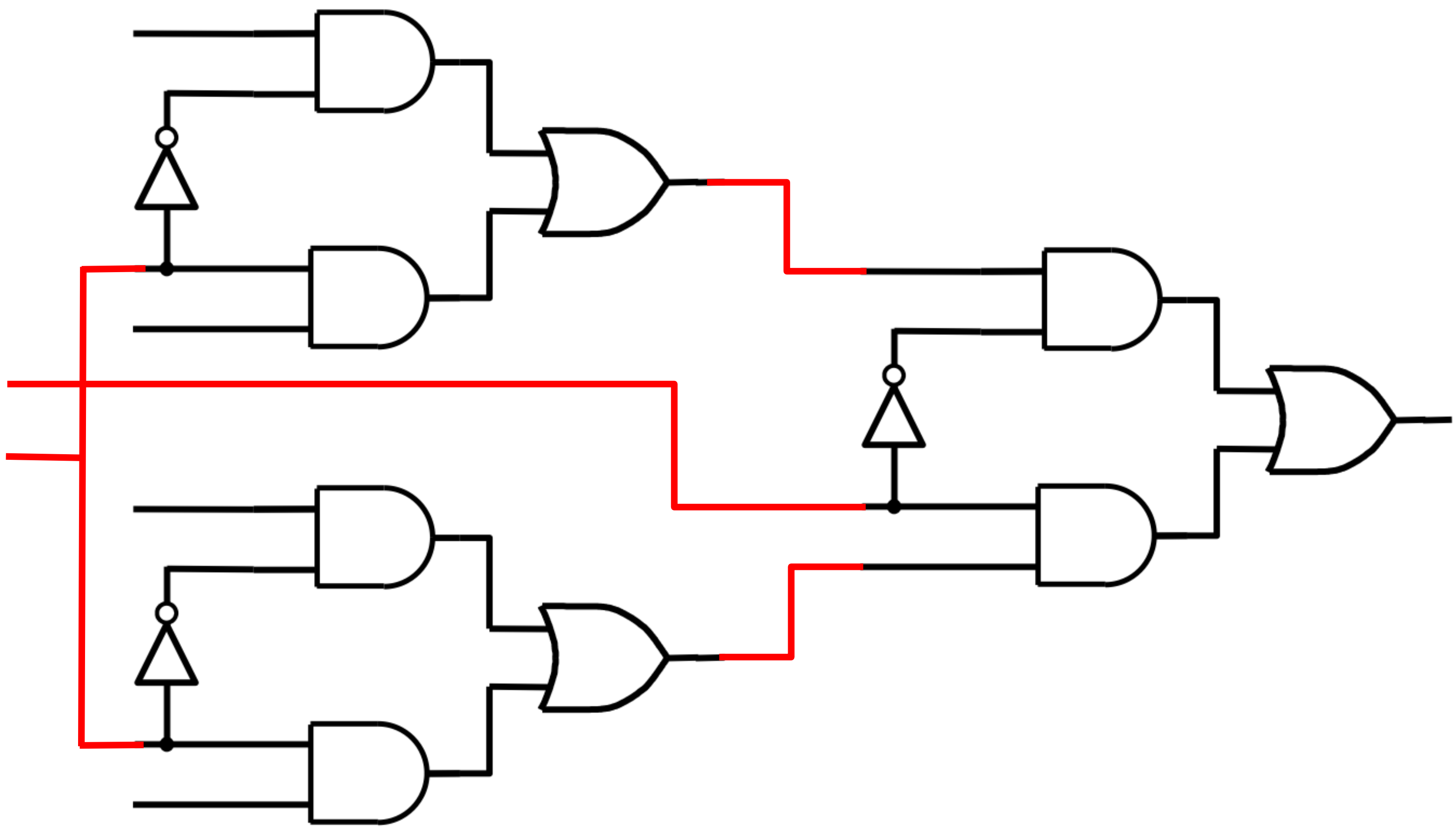
Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer



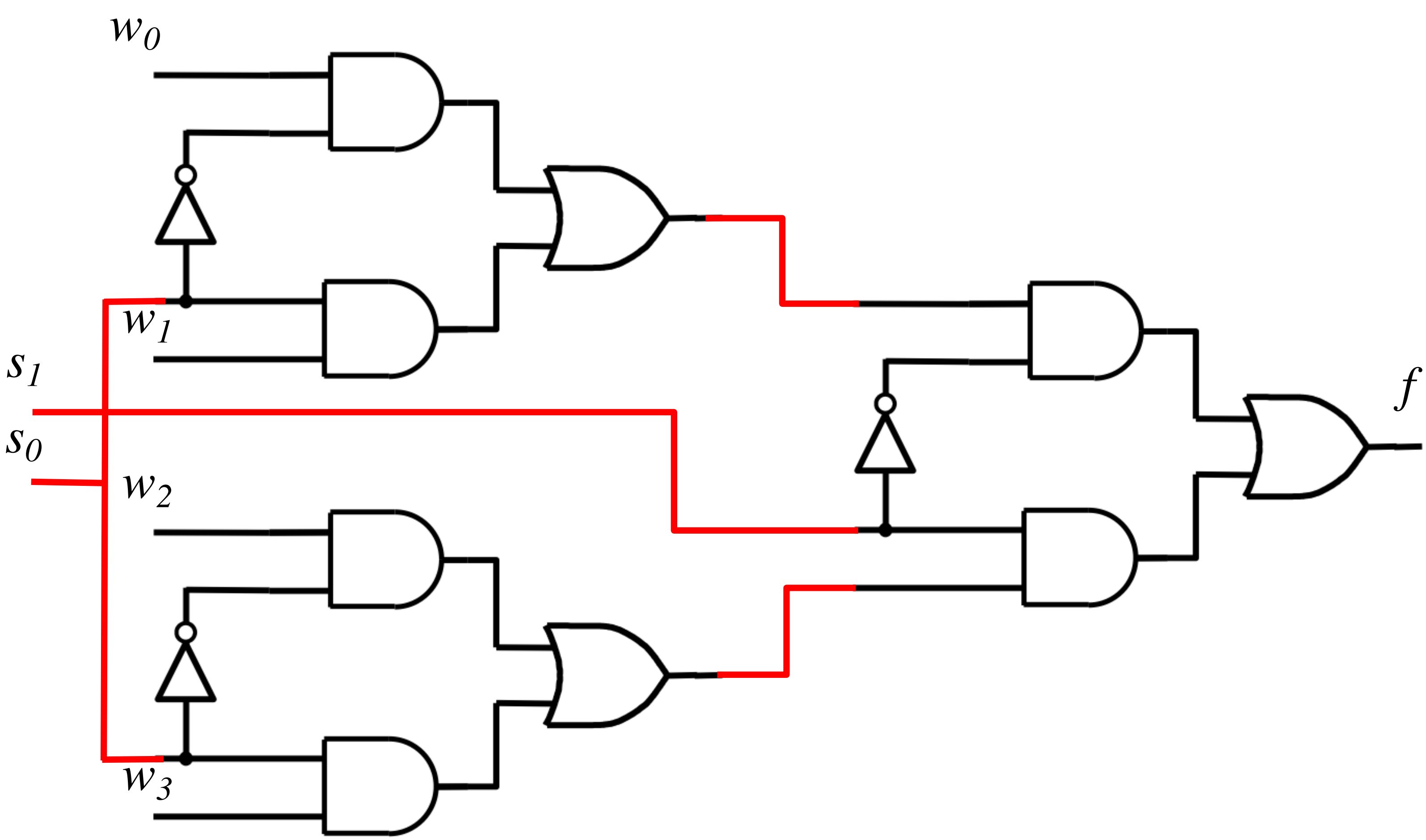
Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer



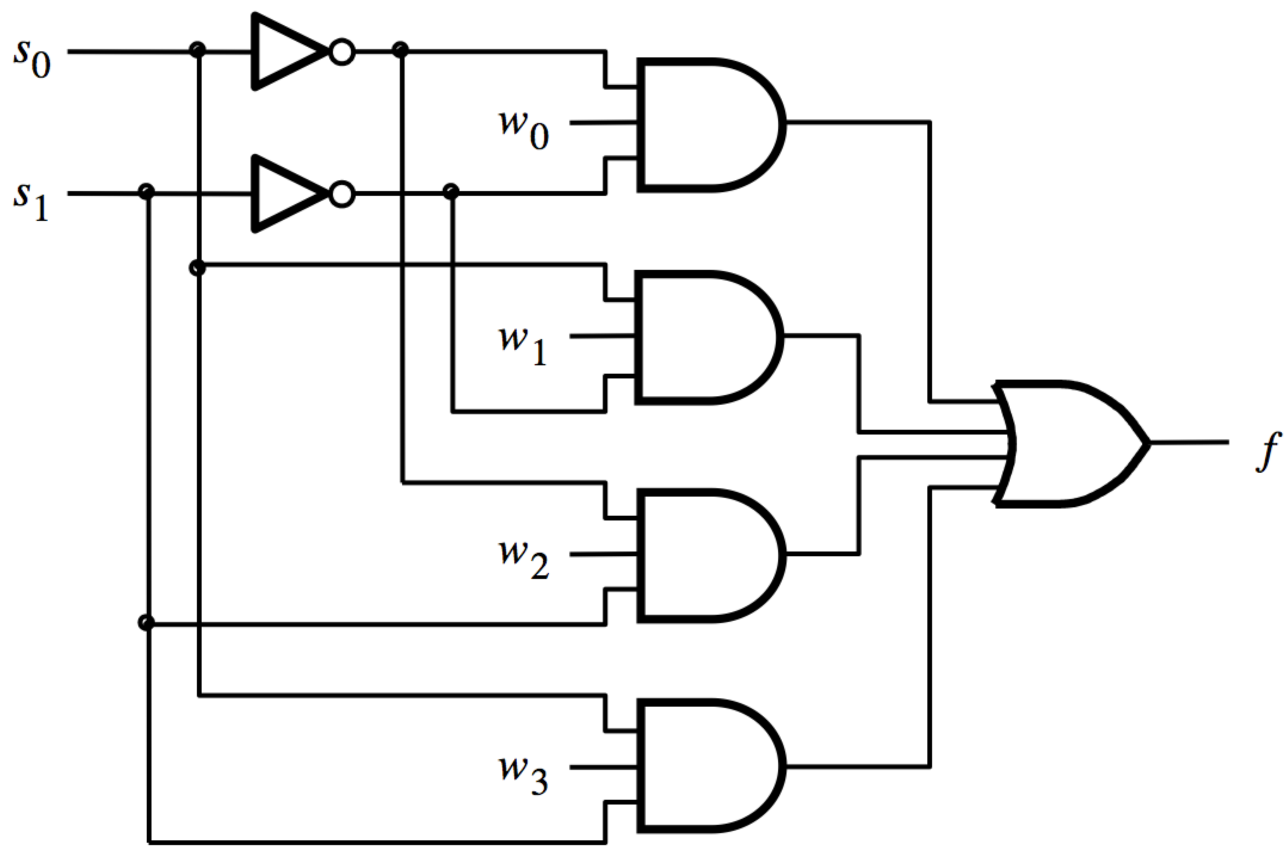
Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer



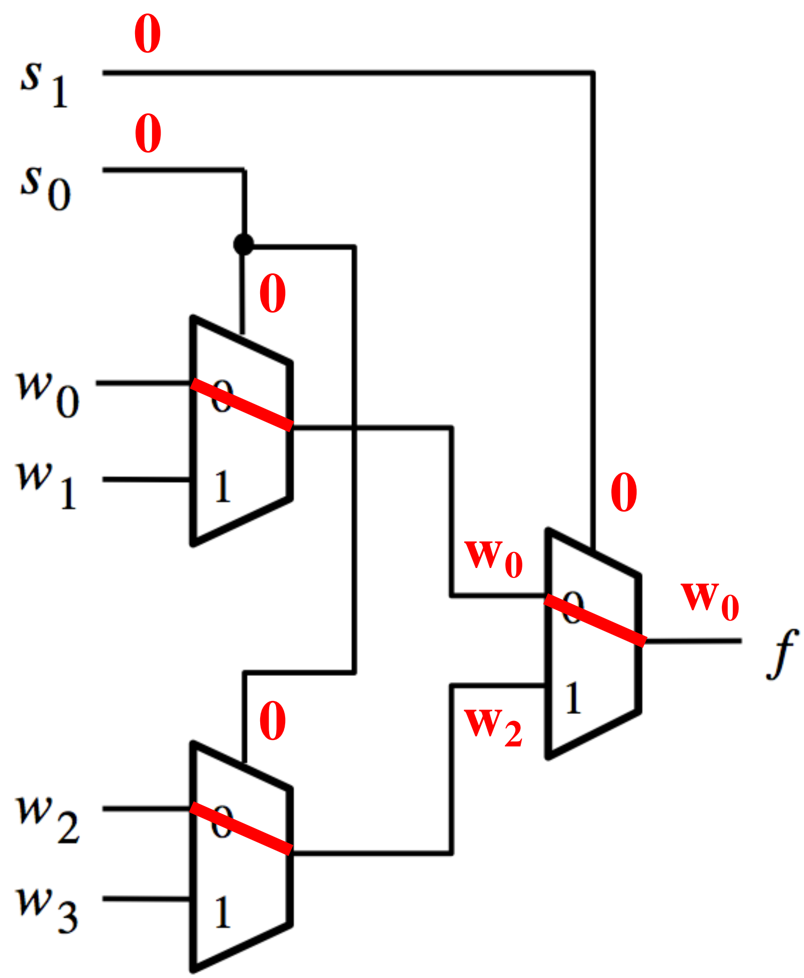
Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer



That is different from the SOP form of the 4-to-1 multiplexer shown below, which uses fewer gates

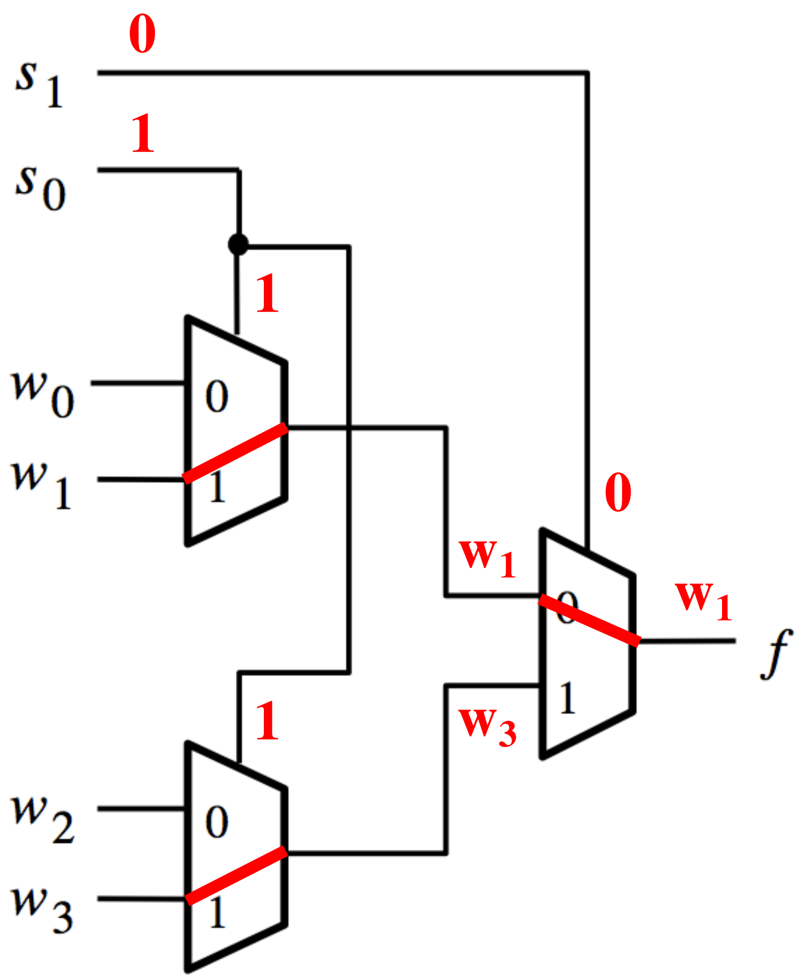


Analysis of the Hierarchical Implementation
(s_1=0 and s_0=0)



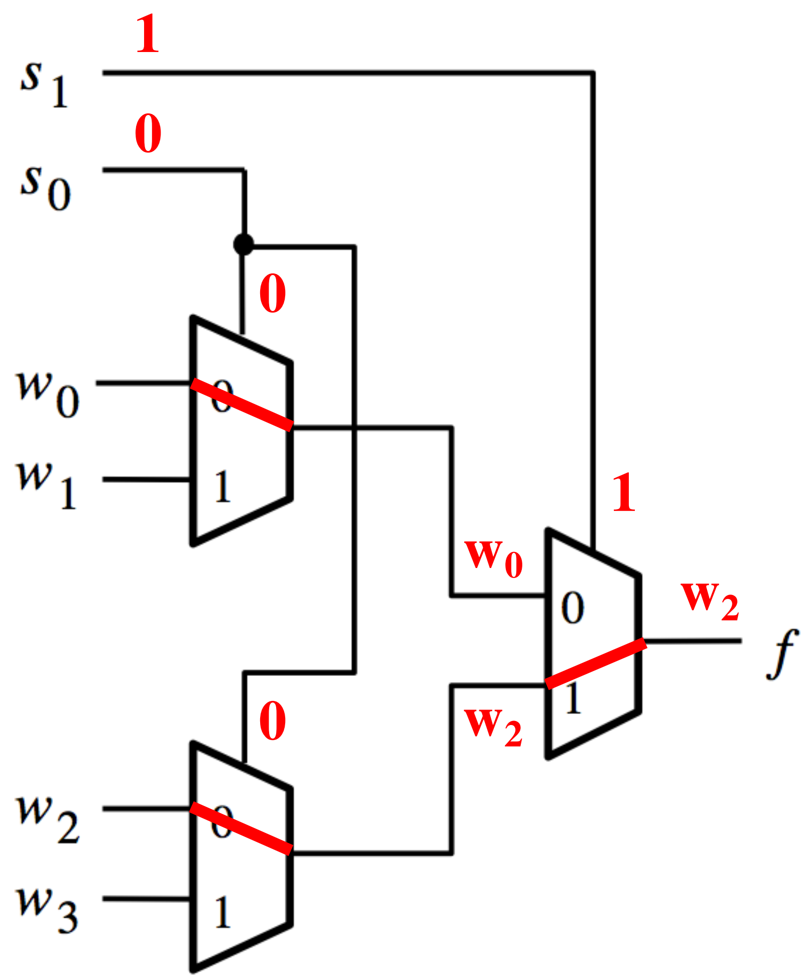
[Figure 4.3 from the textbook]

Analysis of the Hierarchical Implementation (s_1=0 and s_0=1)



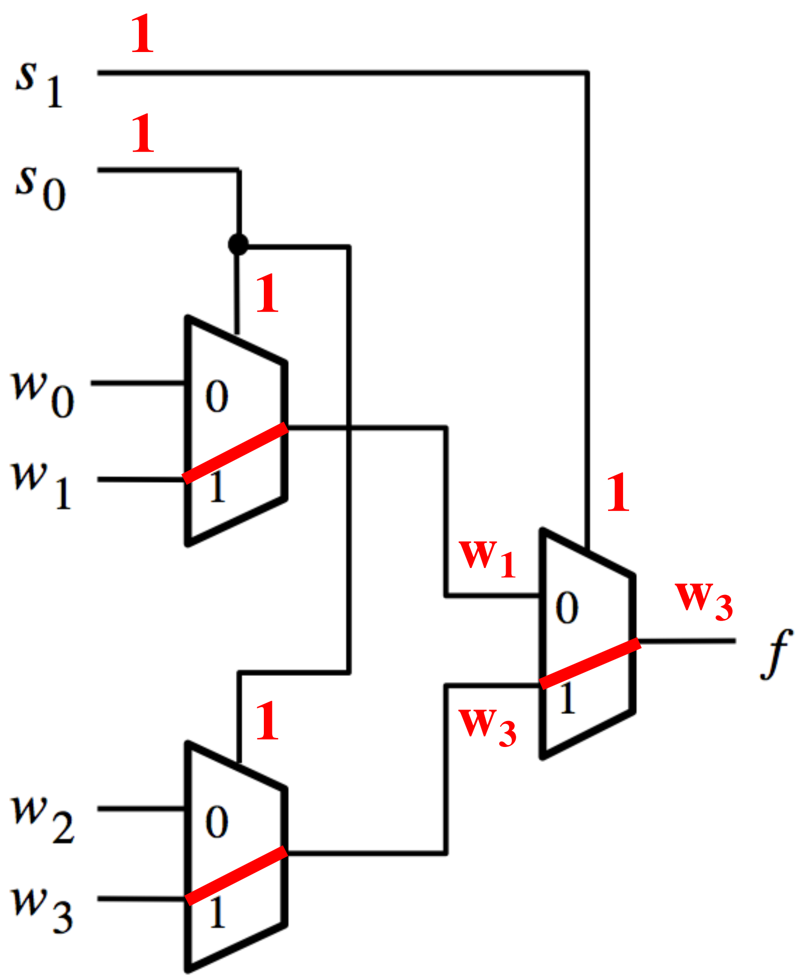
[Figure 4.3 from the textbook]

Analysis of the Hierarchical Implementation
(s_1=1 and s_0=0)



[Figure 4.3 from the textbook]

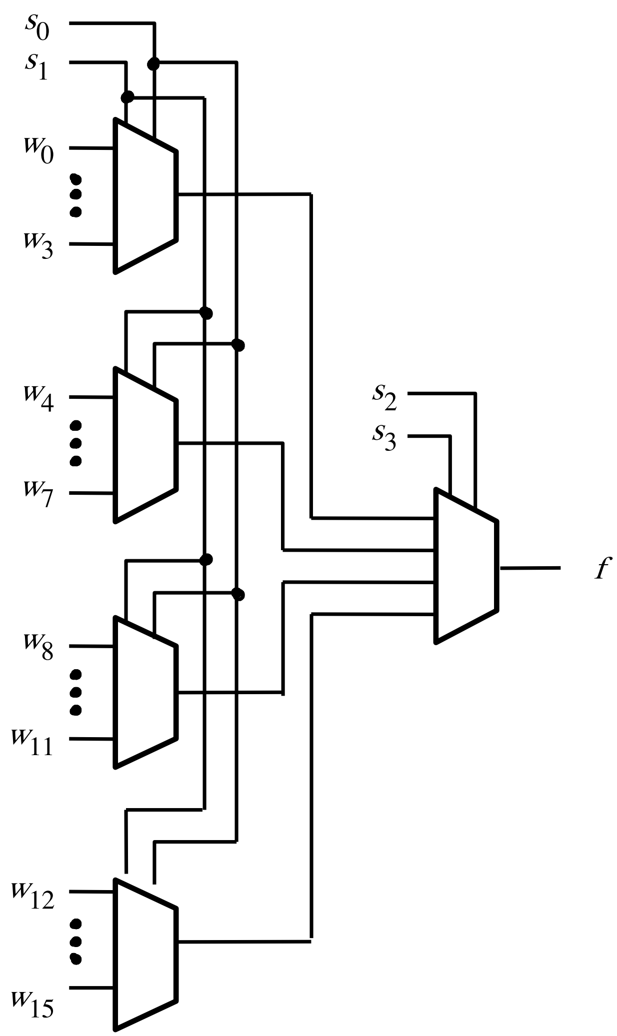
Analysis of the Hierarchical Implementation
(s_1=1 and s_0=1)



[Figure 4.3 from the textbook]

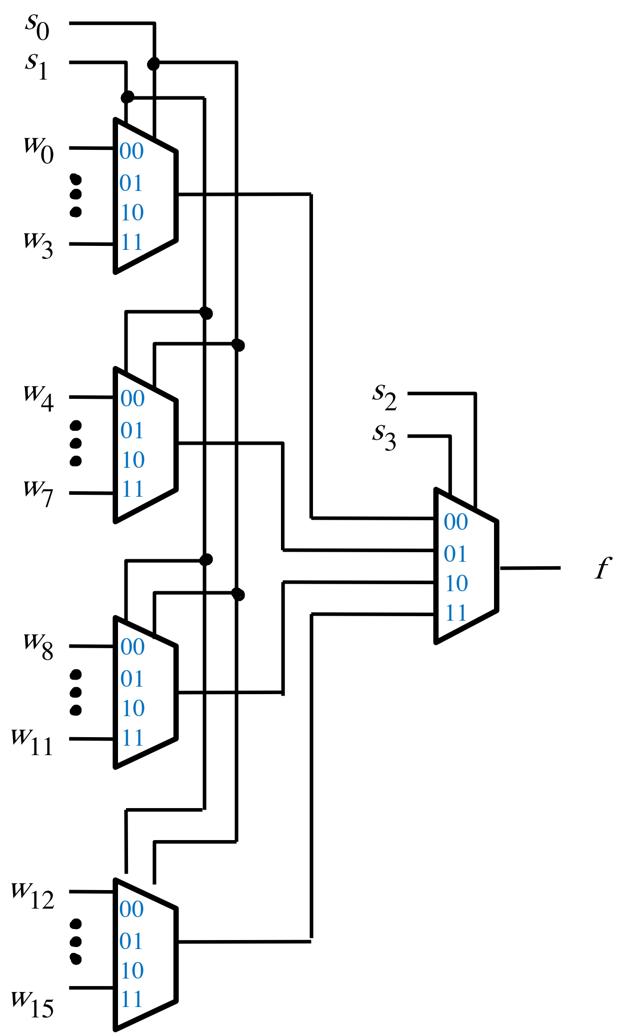
16-to-1 Multiplexer

16-1 Multiplexer



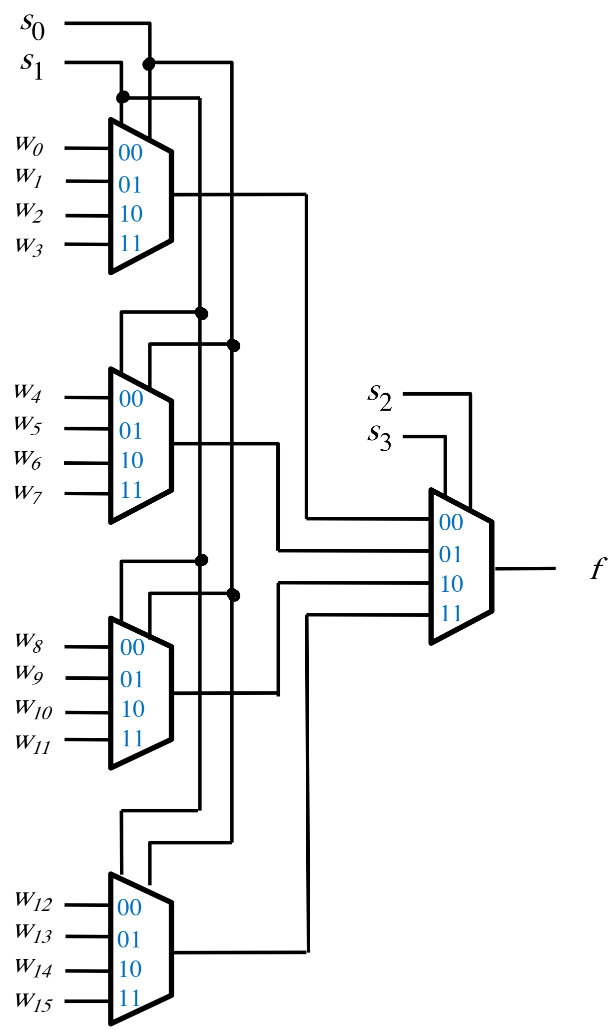
[Figure 4.4 from the textbook]

16-1 Multiplexer

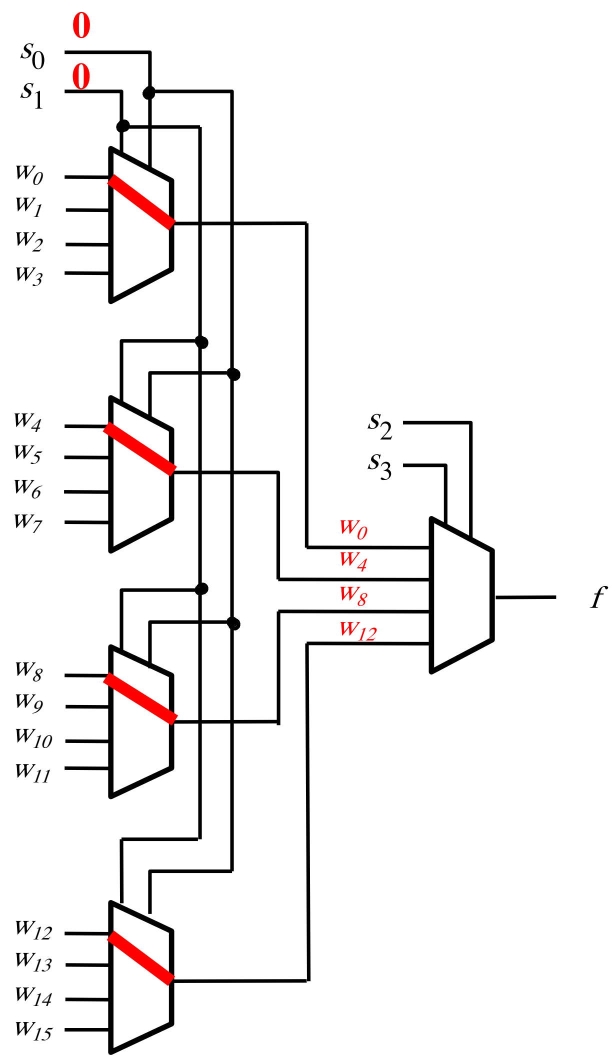


[Figure 4.4 from the textbook]

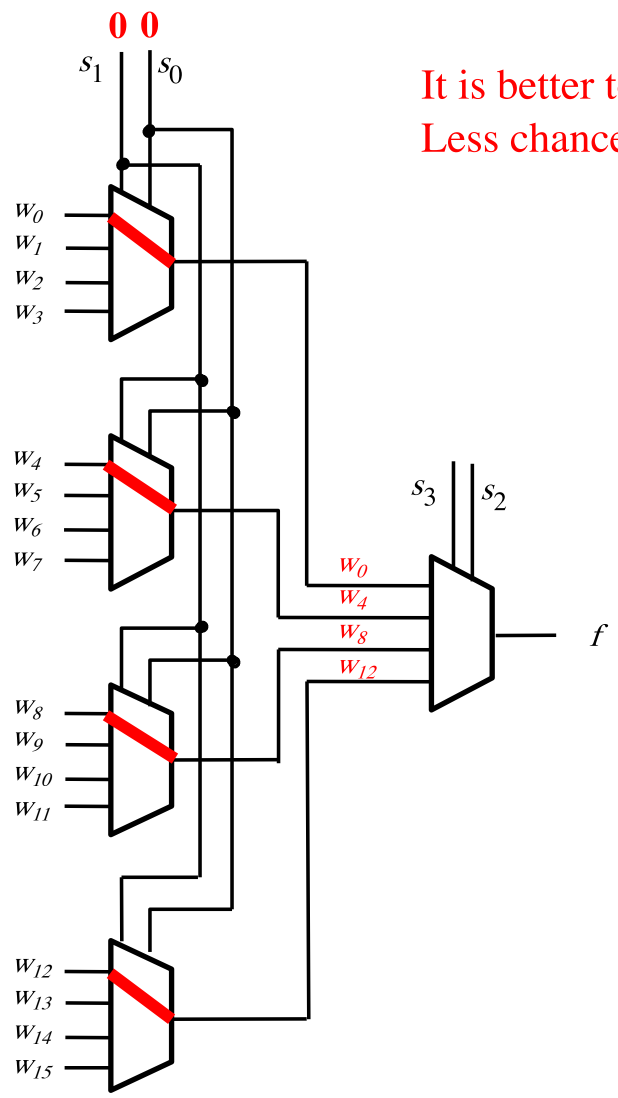
16-1 Multiplexer



16-1 Multiplexer

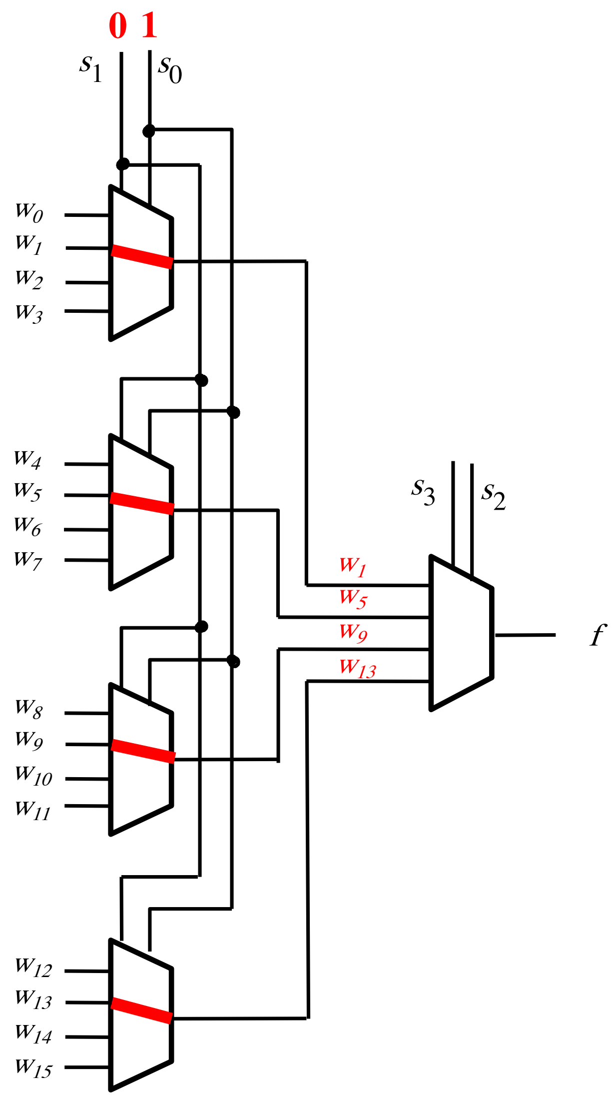


16-1 Multiplexer

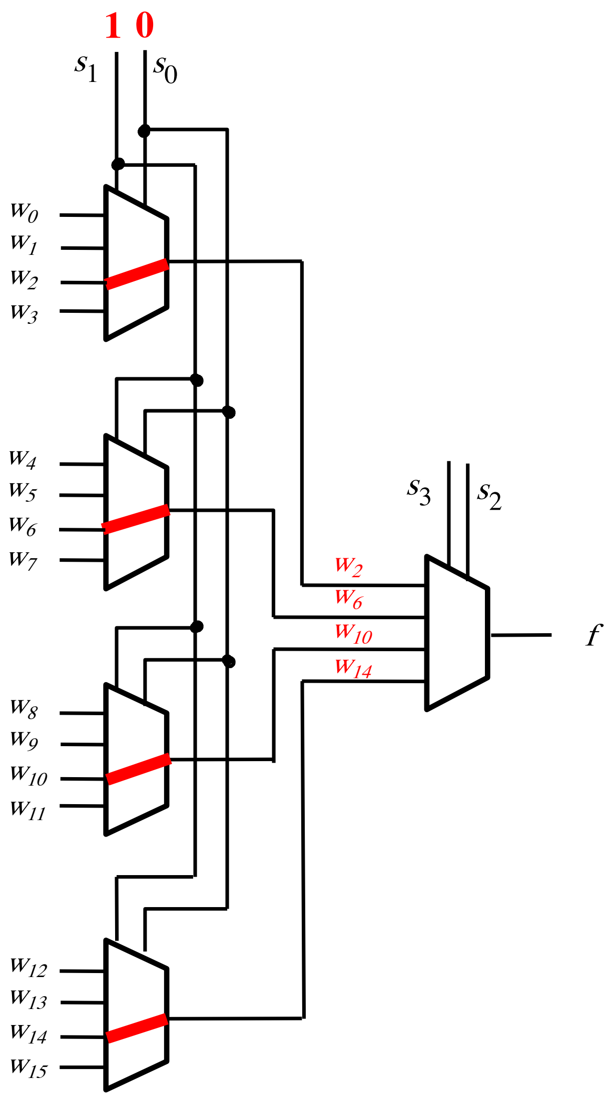


It is better to draw the select lines like this.
Less chance to confuse their order.

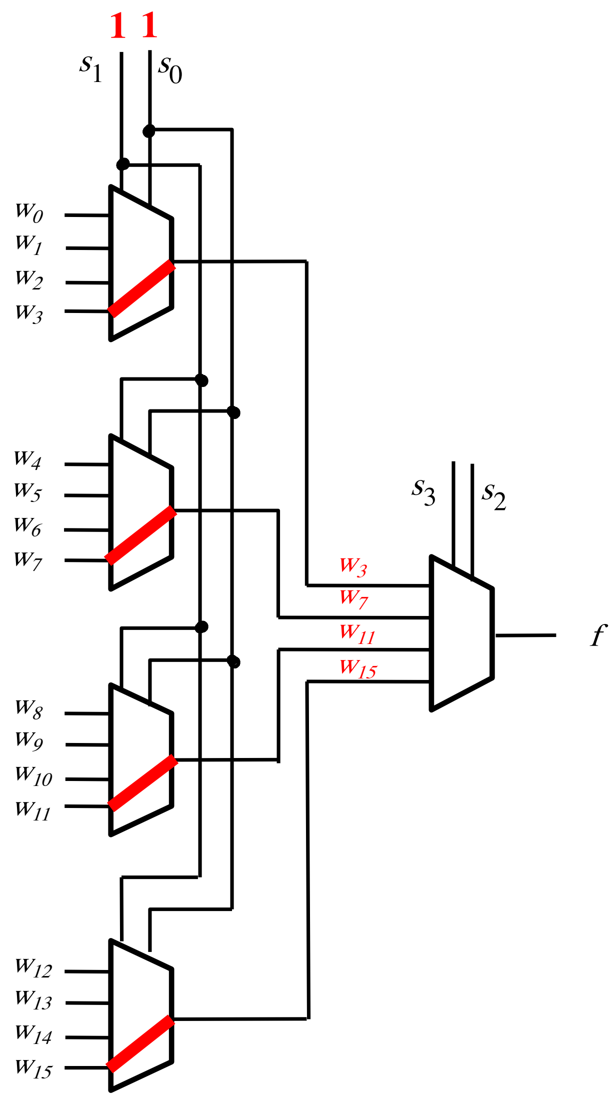
16-1 Multiplexer



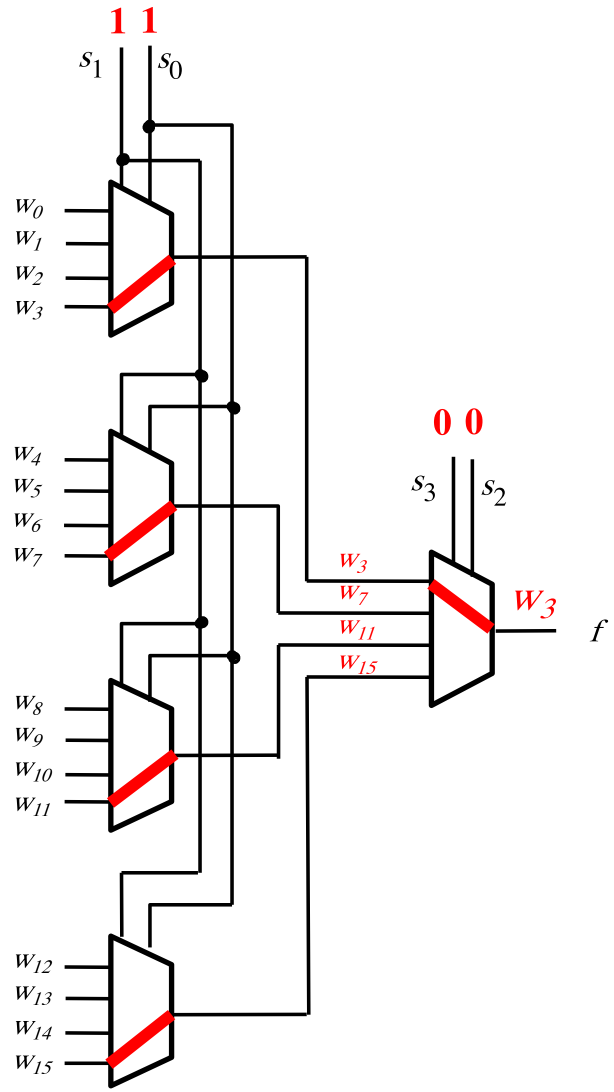
16-1 Multiplexer



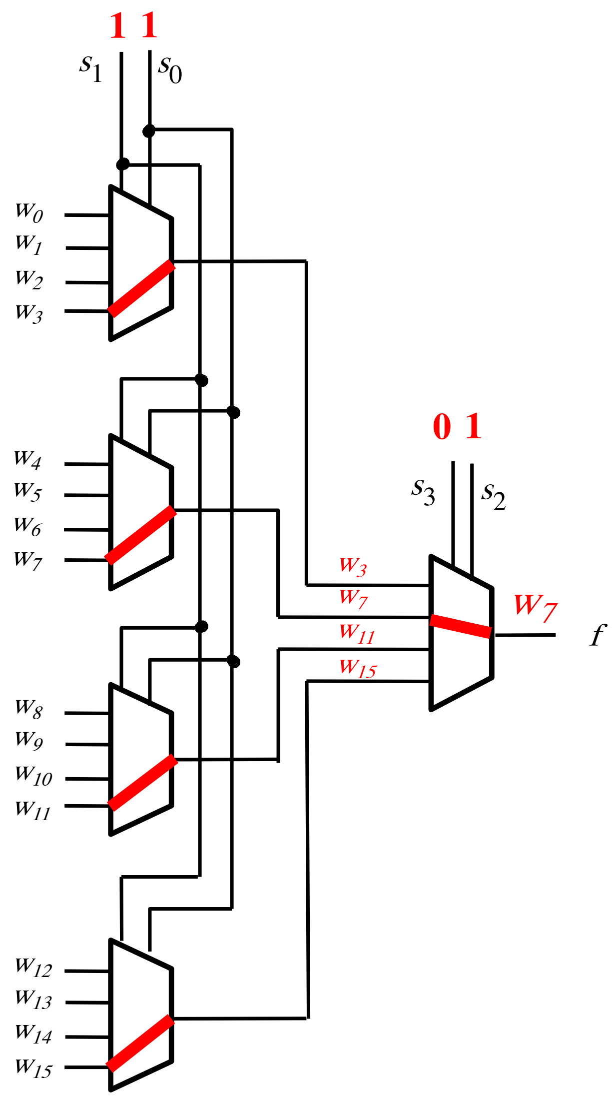
16-1 Multiplexer



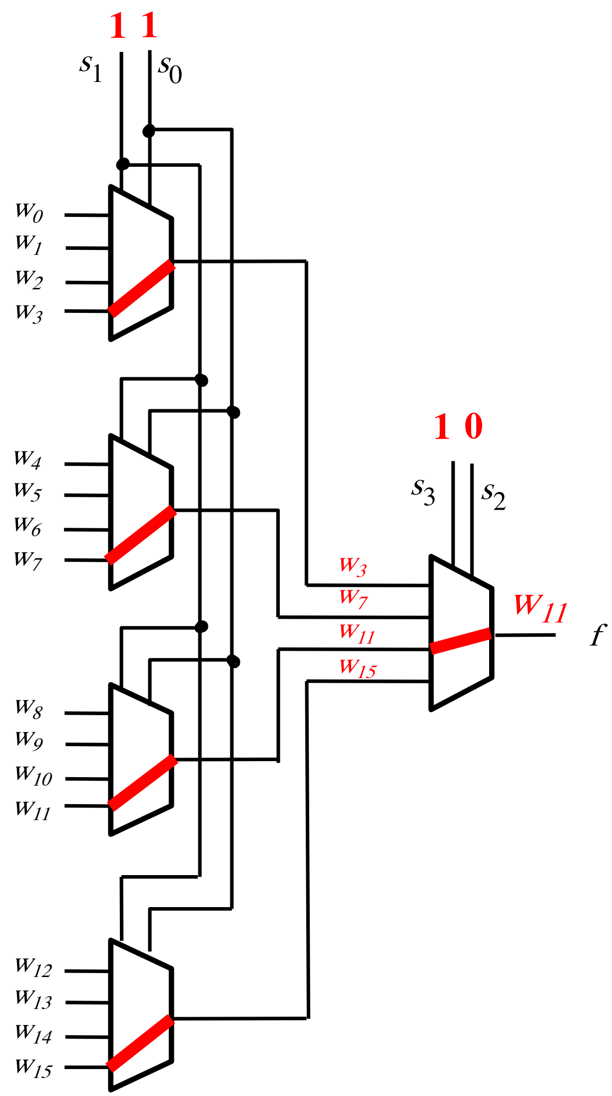
16-1 Multiplexer



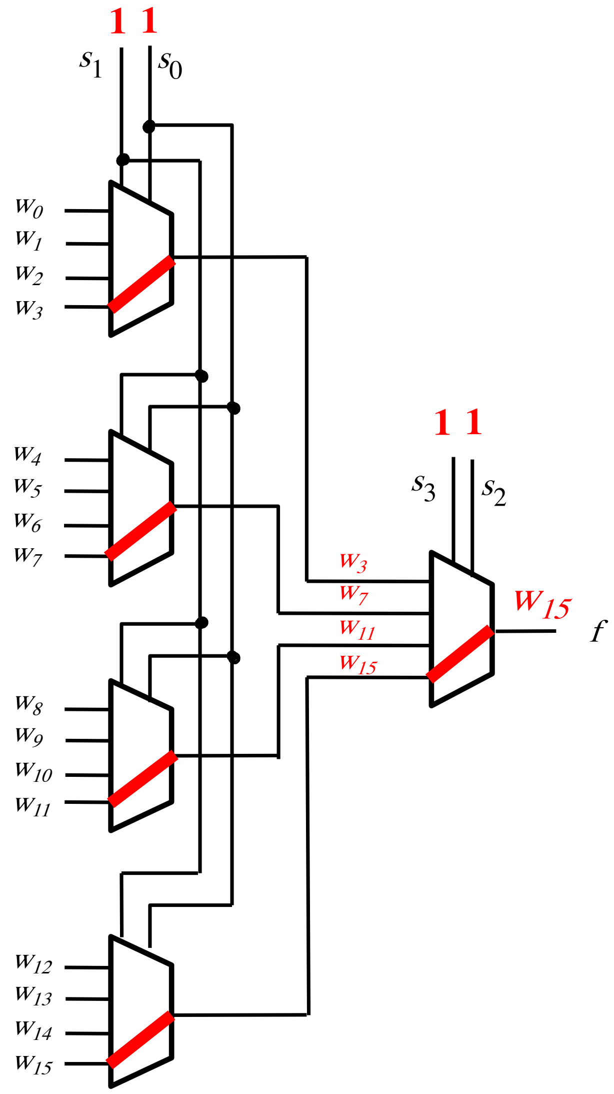
16-1 Multiplexer



16-1 Multiplexer



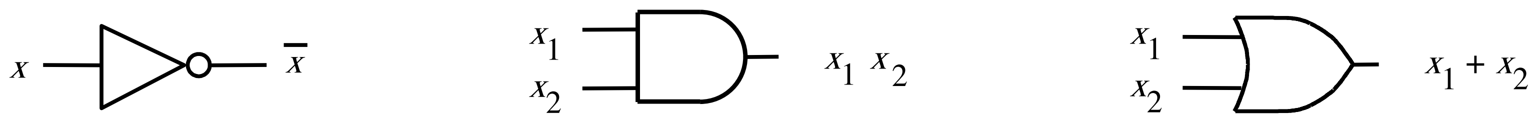
16-1 Multiplexer



[http://upload.wikimedia.org/wikipedia/commons/2/26/SunsetTracksCrop.JPG]

Multiplexers Are Special

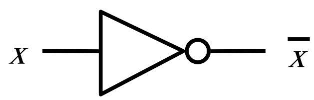
The Three Basic Logic Gates



NOT gate AND gate OR gate

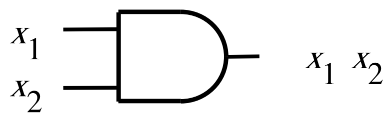
[Figure 2.8 from the textbook]

Truth Table for NOT



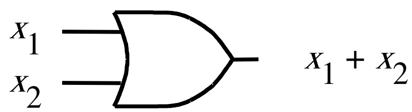
x	$\overline{x}$
0	1
1	0

Truth Table for AND



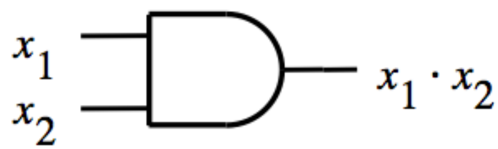
x_1	x_2	$x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table for OR

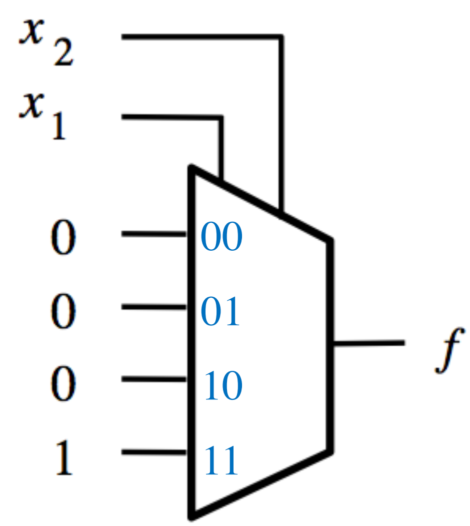


x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1

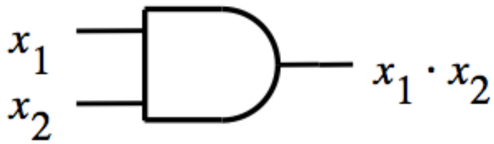
Building an AND Gate with 4-to-1 Mux



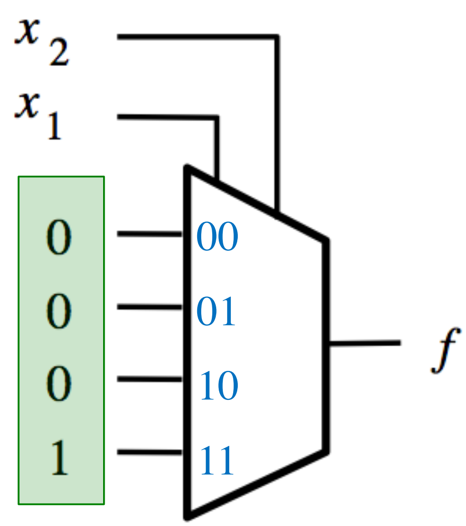
x_1	x_2	$x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1



Building an AND Gate with 4-to-1 Mux

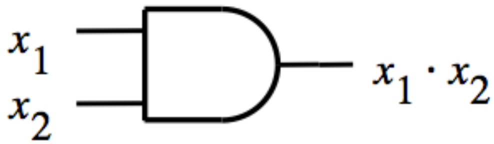


x_1	x_2	$x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1

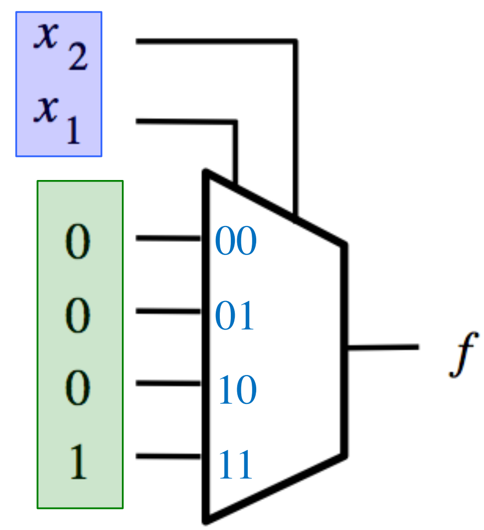


These two are the same.

Building an AND Gate with 4-to-1 Mux



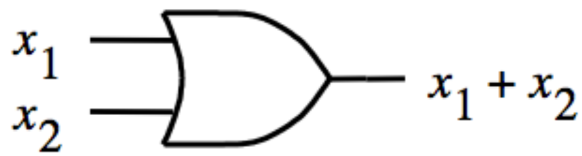
x_1	x_2	$x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1



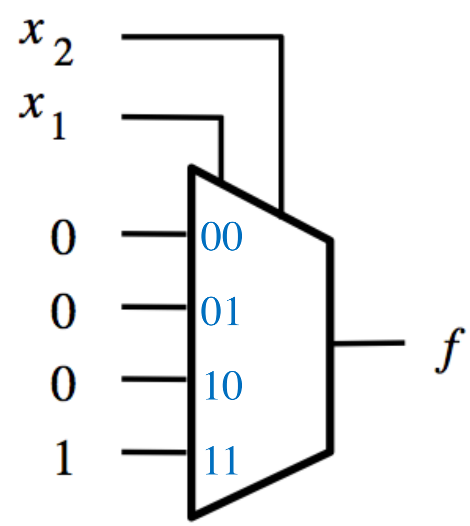
These two are the same.

And so are these two.

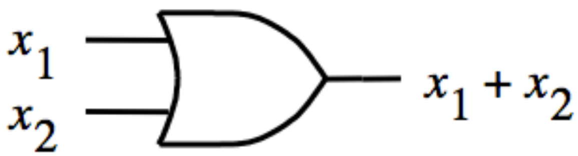
Building an OR Gate with 4-to-1 Mux



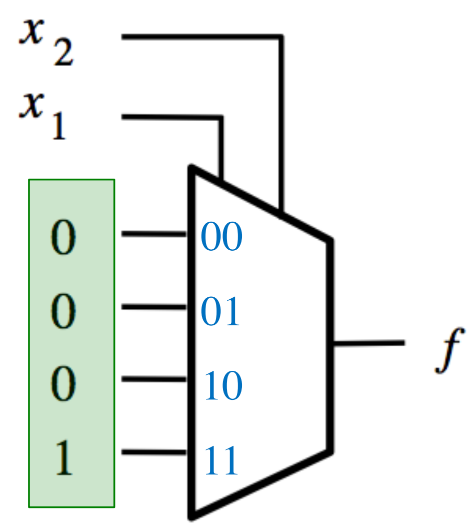
x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1



Building an OR Gate with 4-to-1 Mux

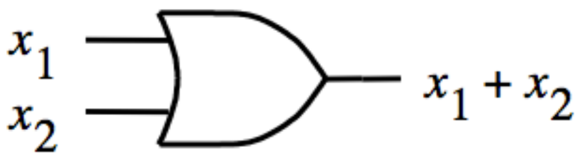


x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1

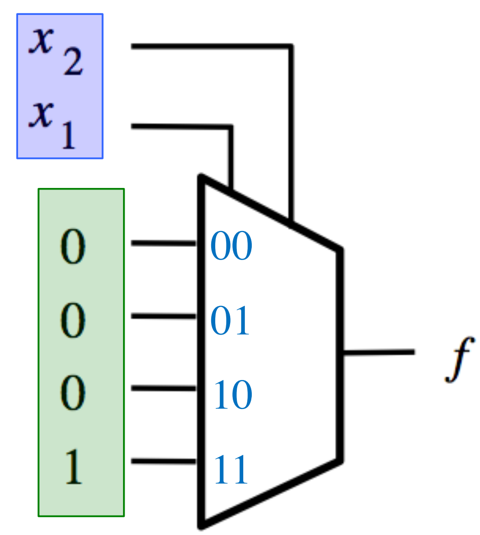


These two are the same.

Building an OR Gate with 4-to-1 Mux



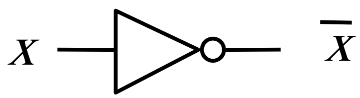
x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1



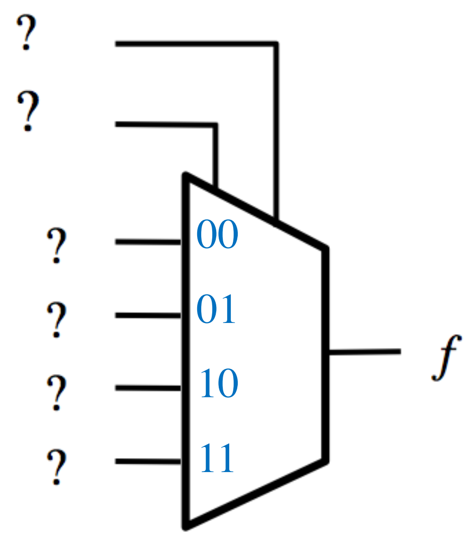
These two are the same.

And so are these two.

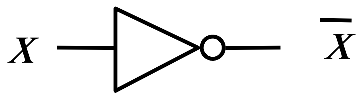
Building a NOT Gate with 4-to-1 Mux



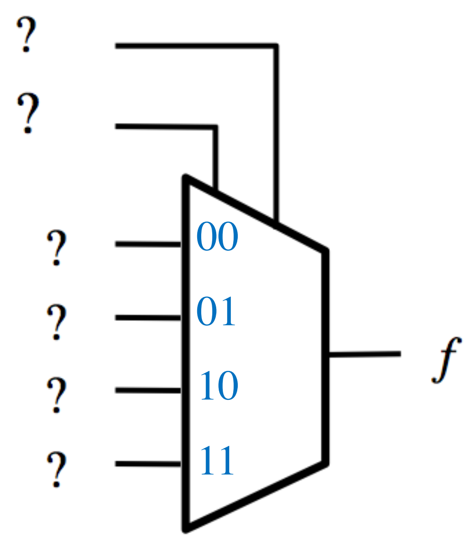
x	$\overline{x}$
0	1
1	0



Building a NOT Gate with 4-to-1 Mux

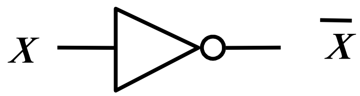


x	y	f
0	0	1
0	1	1
1	0	0
1	1	0

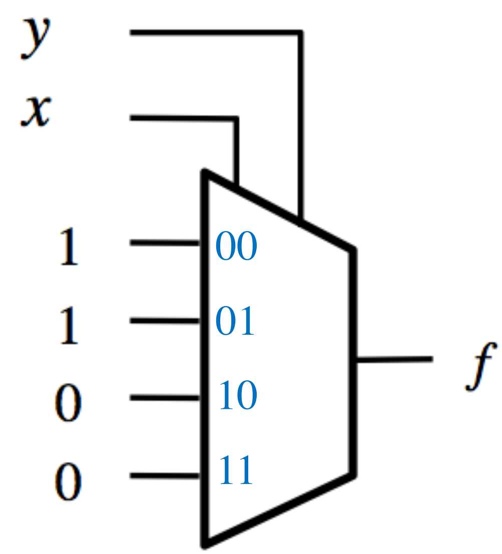


Introduce a dummy variable y.

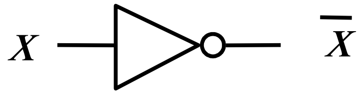
Building a NOT Gate with 4-to-1 Mux



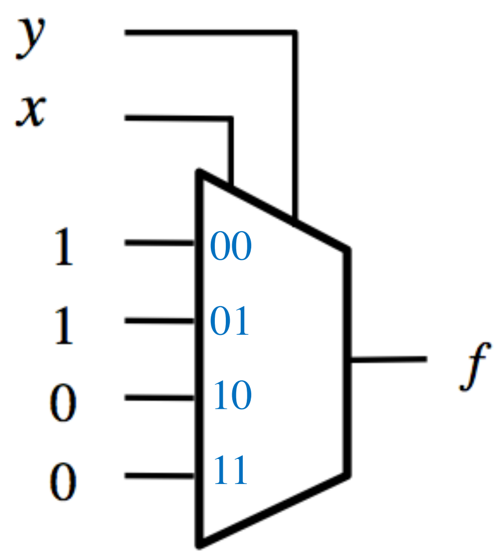
x	y	f
0	0	1
0	1	1
1	0	0
1	1	0



Building a NOT Gate with 4-to-1 Mux

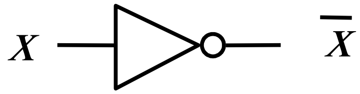


x	y	f
0	0	1
0	1	1
1	0	0
1	1	0

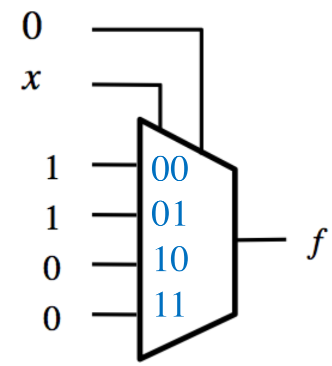


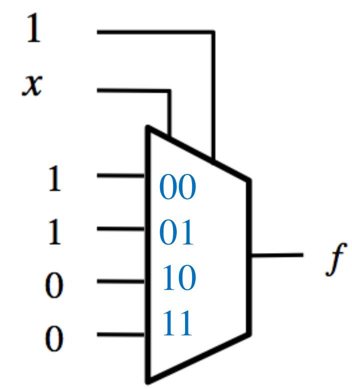
Now set y to either 0 or 1 (both will work). Why?

Building a NOT Gate with 4-to-1 Mux



x	$\overline{x}$
0	1
1	0



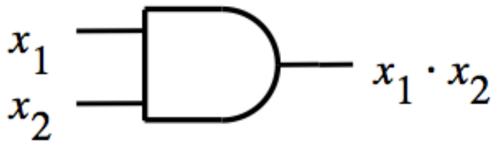


Two alternative solutions.

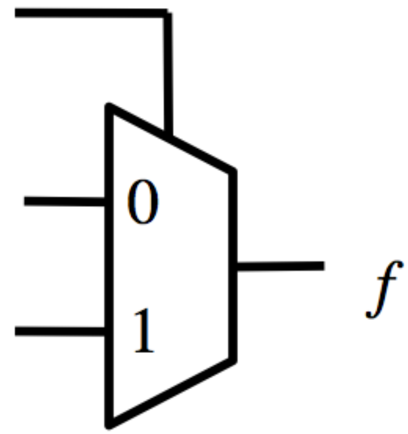
Implications

Any Boolean function can be implemented using only 4-to-1 multiplexers!

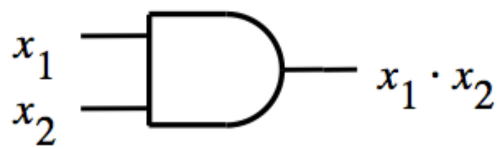
Building an AND Gate with 2-to-1 Mux



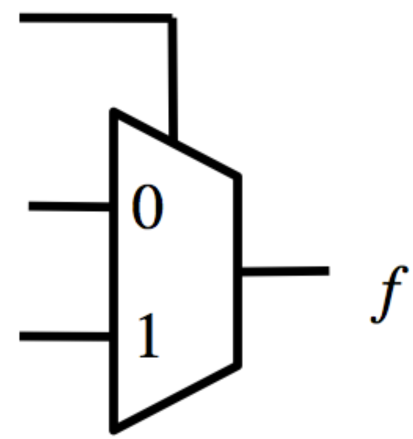
x_1	x_2	$x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1



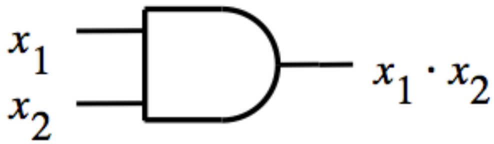
Building an AND Gate with 2-to-1 Mux



x_1	x_2	$x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1



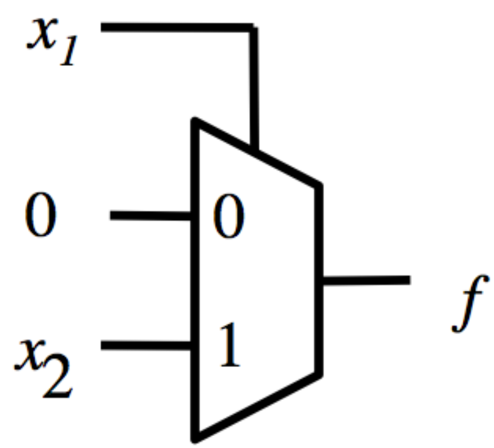
Building an AND Gate with 2-to-1 Mux



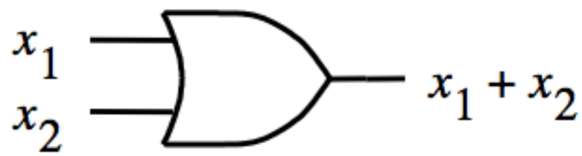
x_1	x_2	$x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1

0

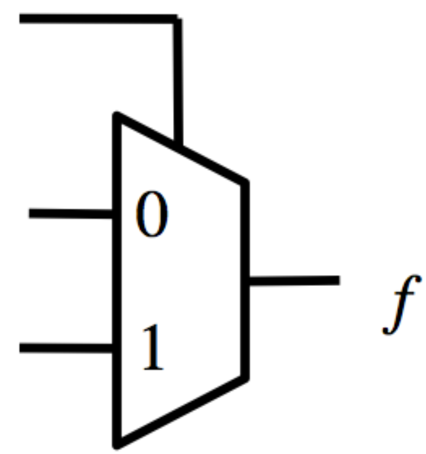
x_2



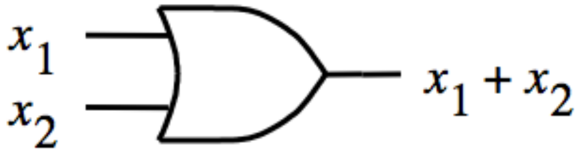
Building an OR Gate with 2-to-1 Mux



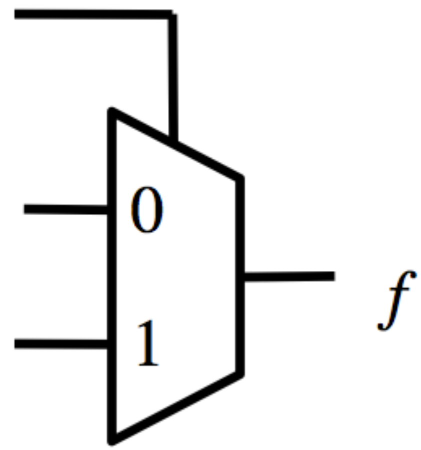
x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1



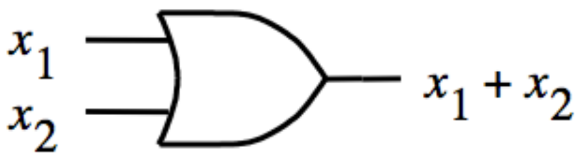
Building an OR Gate with 2-to-1 Mux



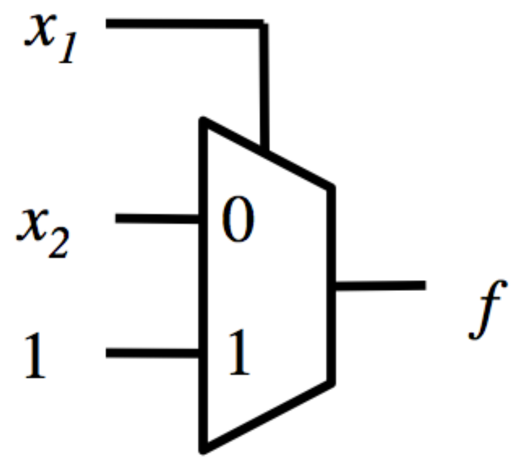
x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1



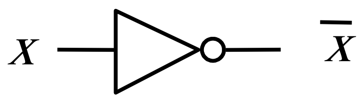
Building an OR Gate with 2-to-1 Mux



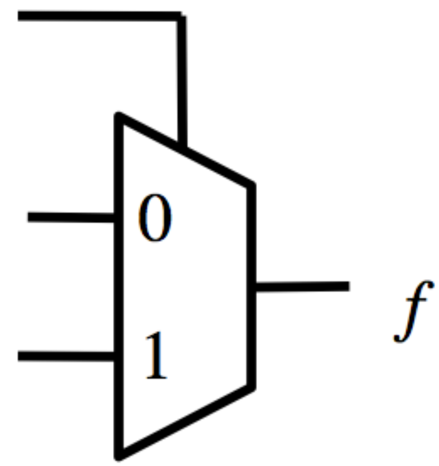
x_1	x_2	$x_1 + x_2$	
0	0	0	x_2
0	1	1	
1	0	1	1
1	1	1	



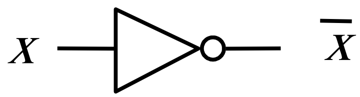
Building a NOT Gate with 2-to-1 Mux



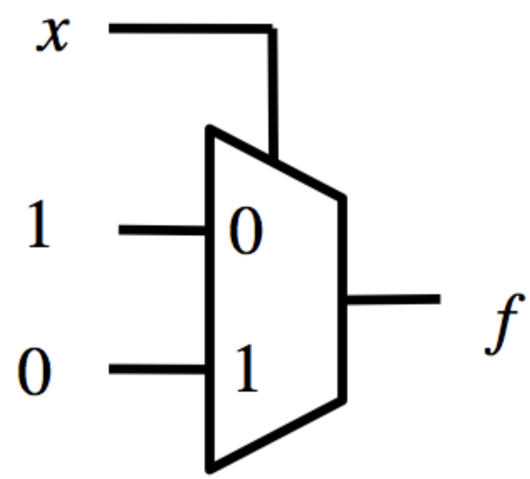
x	$\overline{x}$
0	1
1	0



Building a NOT Gate with 2-to-1 Mux



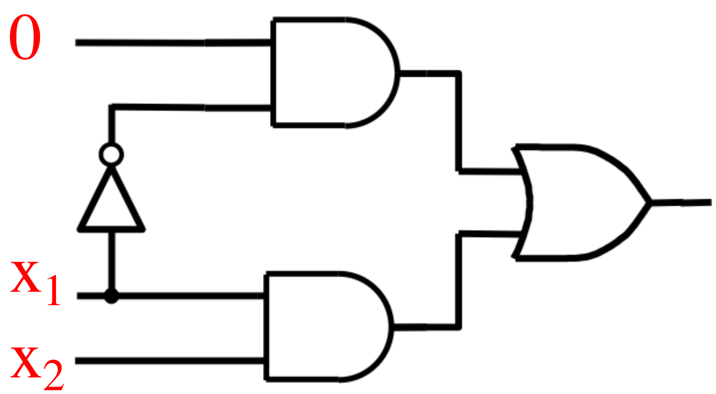
x	$\overline{x}$
0	1
1	0



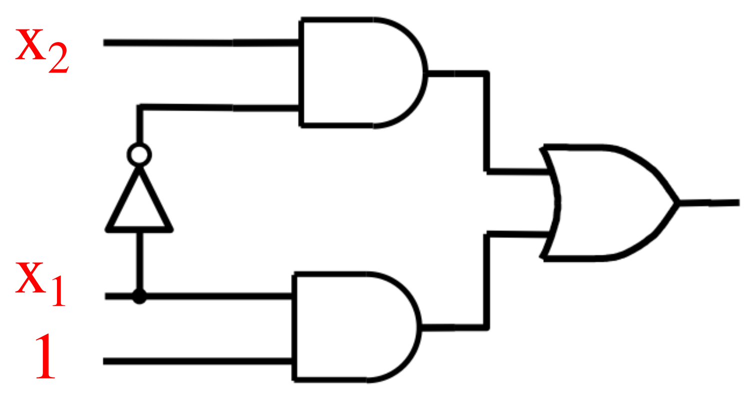
Implications

Any Boolean function can be implemented using only 2-to-1 multiplexers!

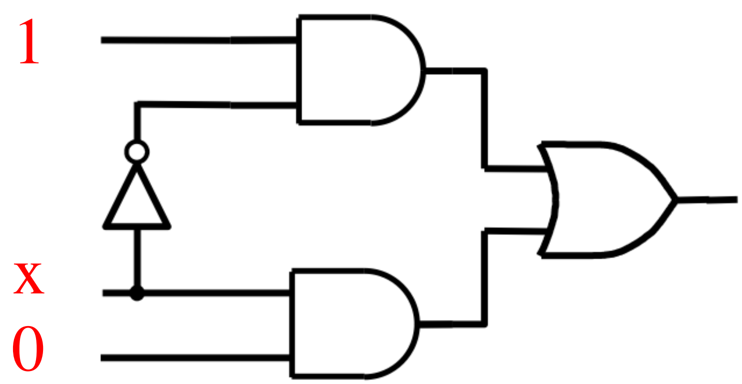
AND



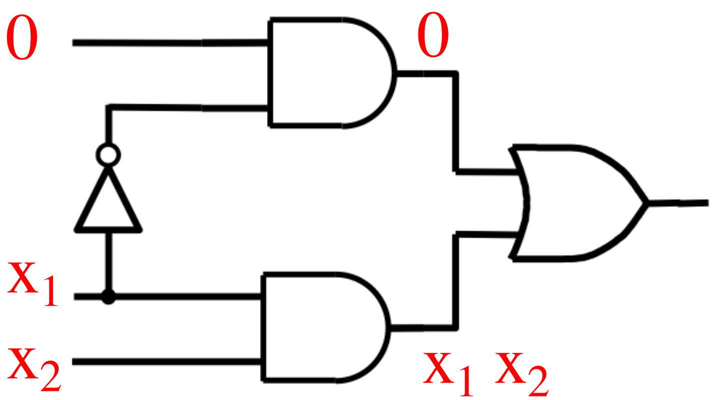
OR



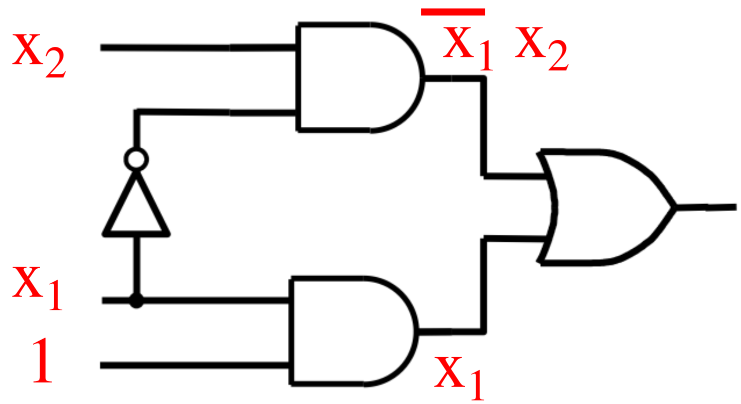
NOT



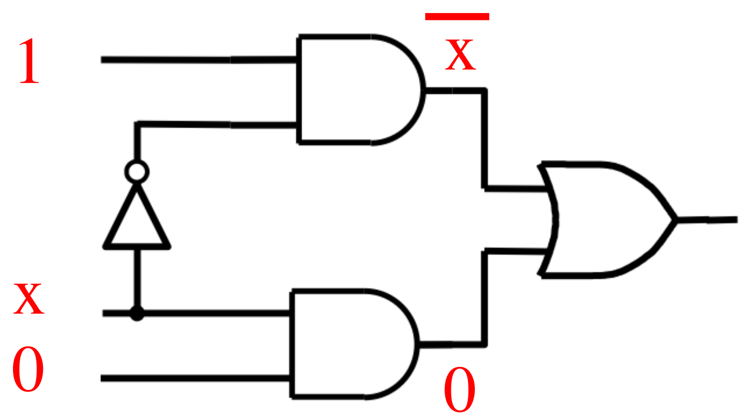
AND



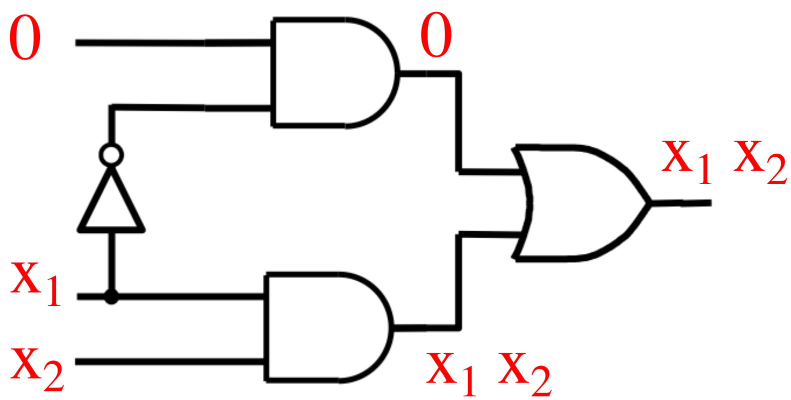
OR



NOT

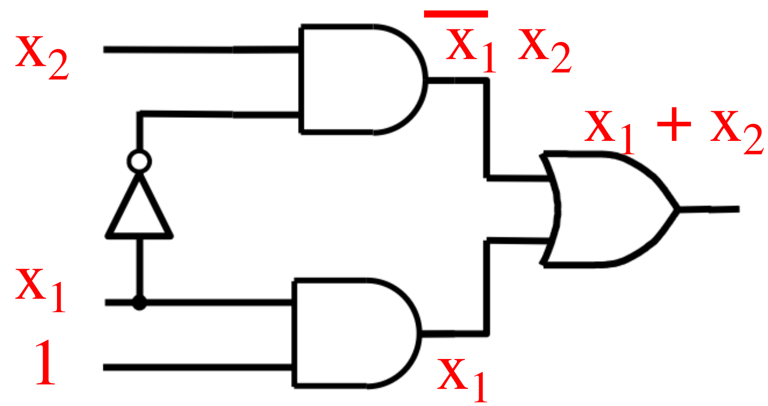


AND



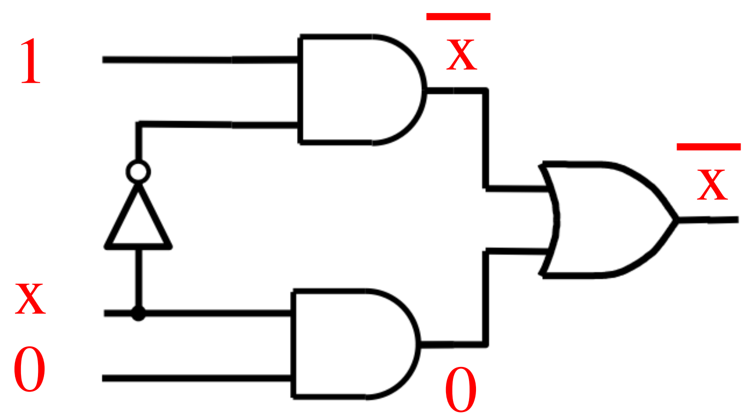
0
x_1
x_2
0
$x_1 \; x_2$
$x_1 \; x_2$

OR



x_2
x_1
1
$\overline{x_1} \; x_2$
x_1
$x_1 + x_2$

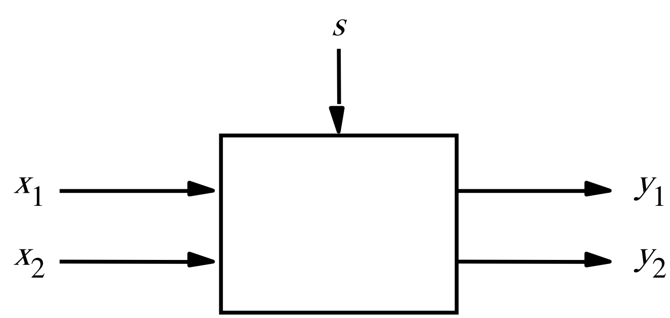
NOT



1
x
0
$\overline{x}$
0
$\overline{x}$

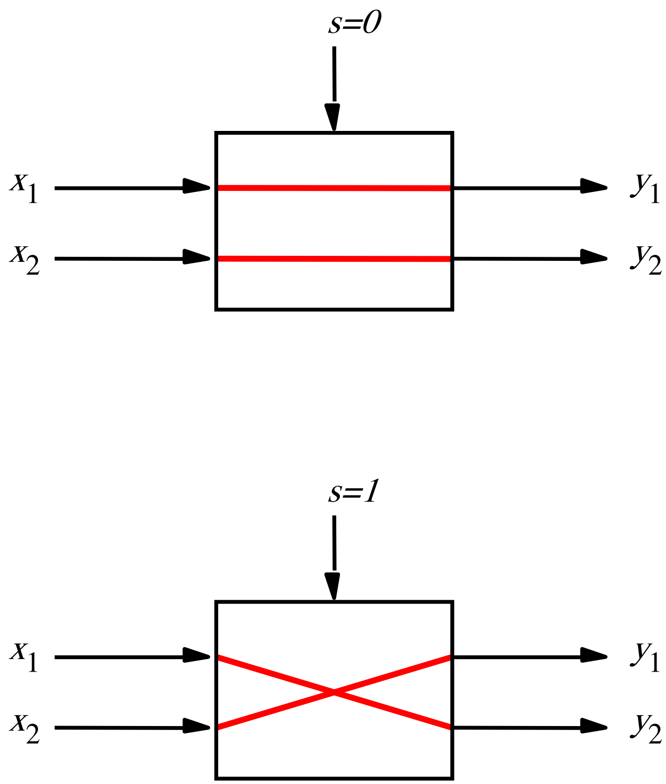
Switch Circuit

2 x 2 Crossbar switch

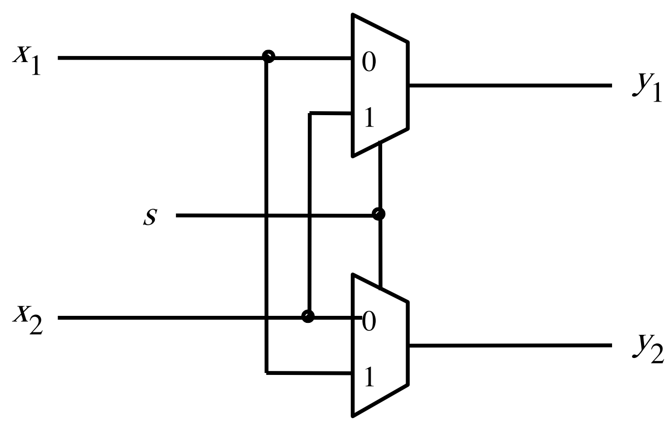


[Figure 4.5a from the textbook]

2 x 2 Crossbar switch

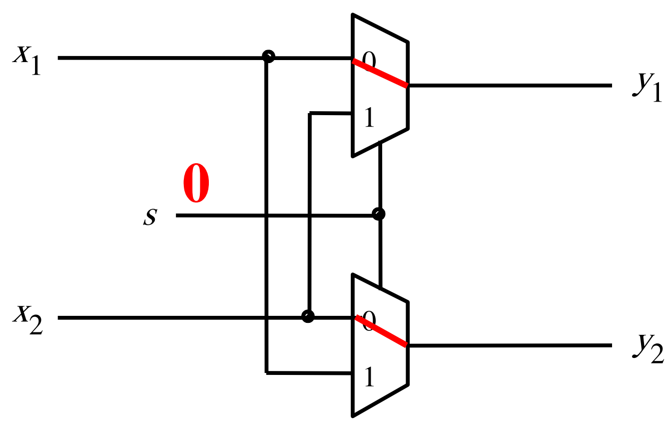


Implementation of a 2 x 2 crossbar switch with multiplexers



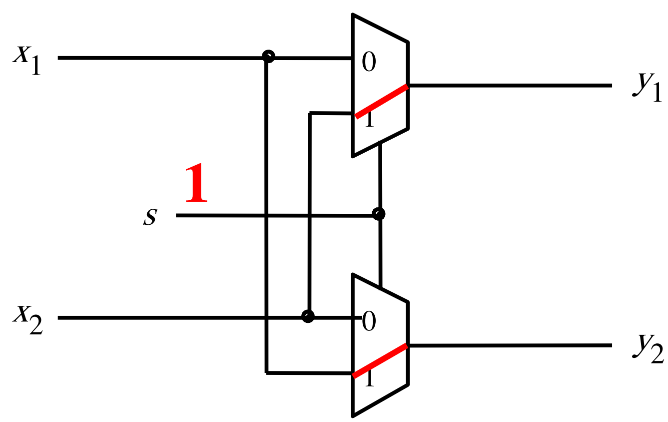
[Figure 4.5b from the textbook]

Implementation of a 2 x 2 crossbar switch with multiplexers



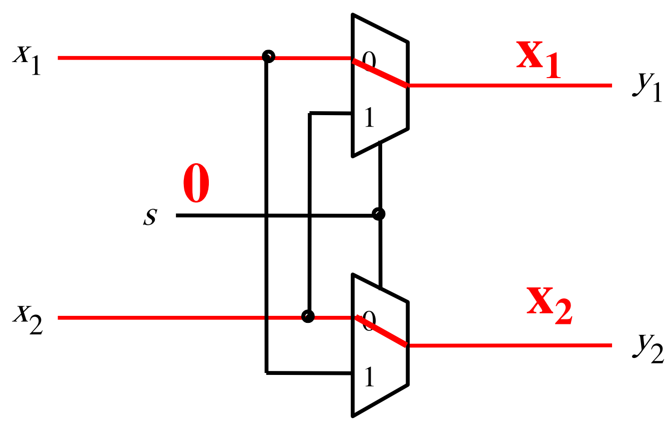
[Figure 4.5b from the textbook]

Implementation of a 2 x 2 crossbar switch with multiplexers



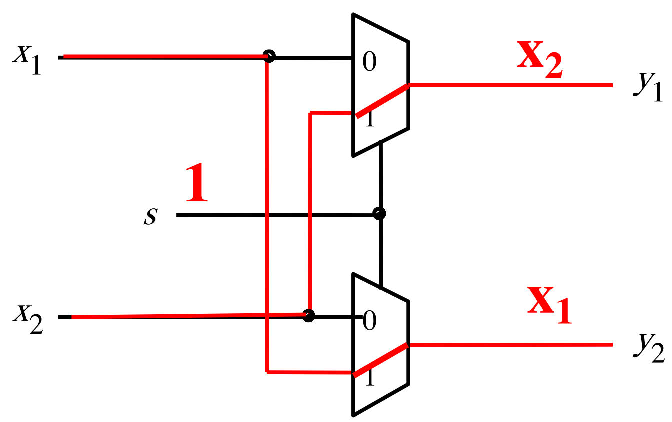
[Figure 4.5b from the textbook]

Implementation of a 2 x 2 crossbar switch with multiplexers



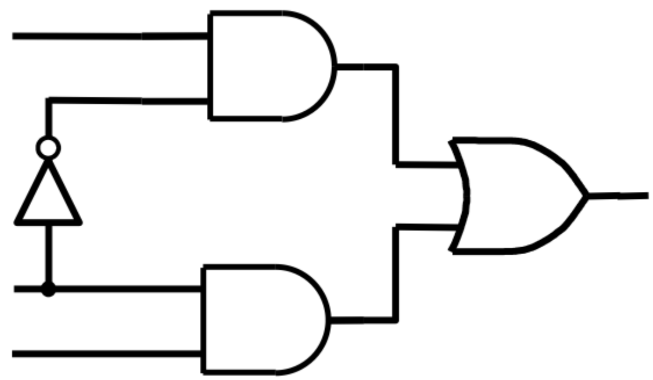
[Figure 4.5b from the textbook]

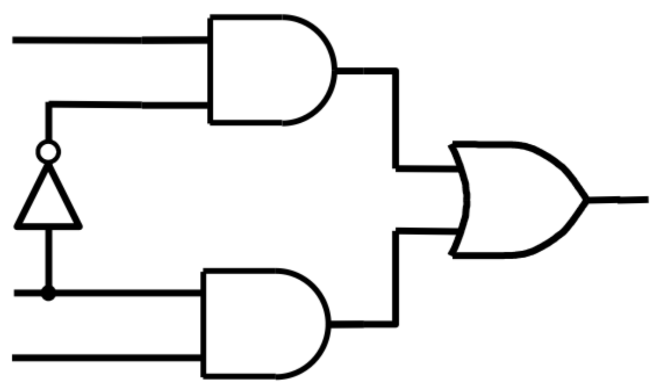
Implementation of a 2 x 2 crossbar switch with multiplexers



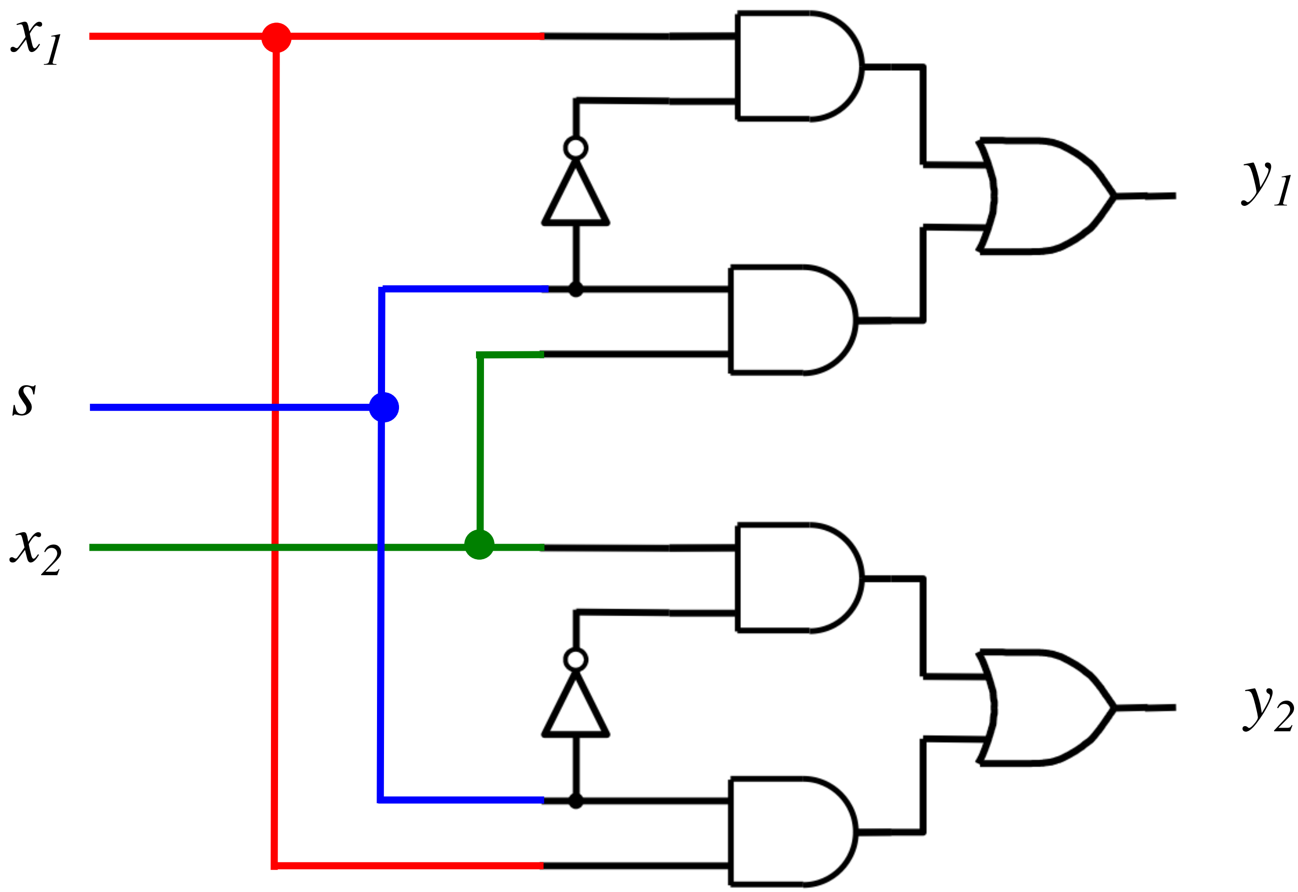
[Figure 4.5b from the textbook]

Implementation of a 2 x 2 crossbar switch with multiplexers





Implementation of a 2 x 2 crossbar switch with multiplexers

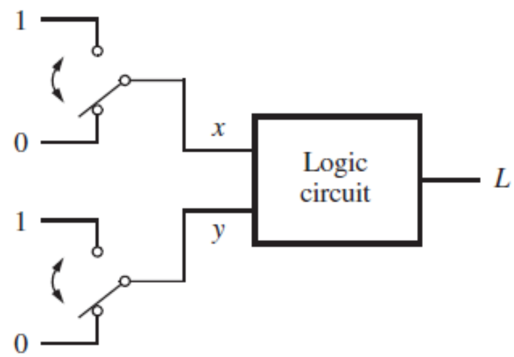


Synthesis of Logic Circuits Using Multiplexers

Synthesis of Logic Circuits Using Multiplexers

Note: This method is NOT the same as simply replacing each logic gate with a multiplexer! It is a lot more efficient.

The XOR Logic Gate



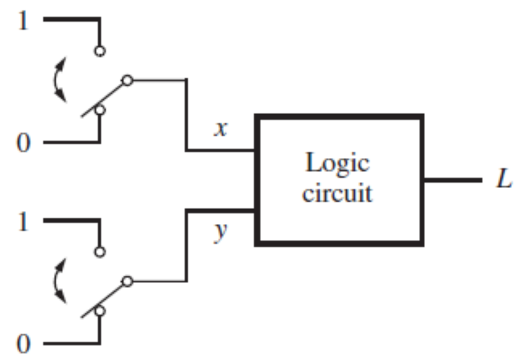
(a) Two switches that control a light

x	y	L
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth table

[Figure 2.11 from the textbook]

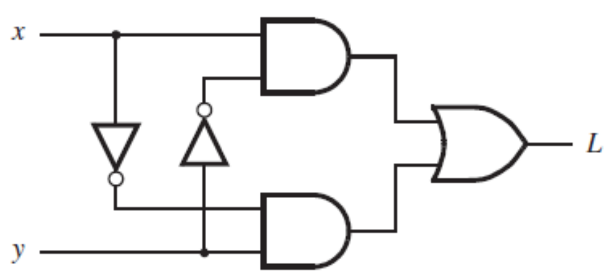
The XOR Logic Gate



(a) Two switches that control a light

x	y	L
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth table



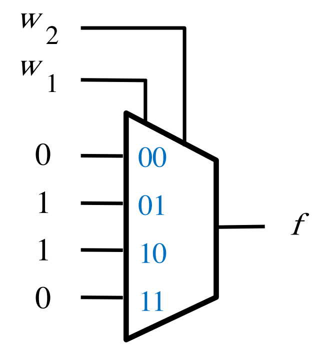
(c) Logic network

(d) XOR gate symbol

[Figure 2.11 from the textbook]

Implementation of a logic function with a 4-to-1 multiplexer

w_1	w_2	f
0	0	0
0	1	1
1	0	1
1	1	0



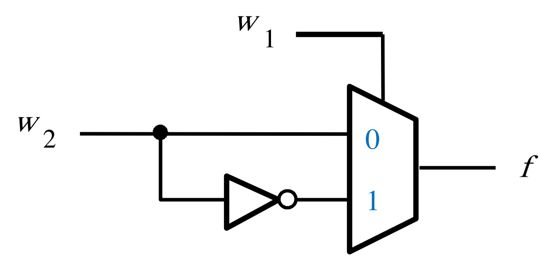
[Figure 4.6a from the textbook]

Implementation of the same logic function with a 2-to-1 multiplexer

w_1	w_2	f
0	0	0
0	1	1
1	0	1
1	1	0

w_1	f
0	w_2
1	$\overline{w_2}$

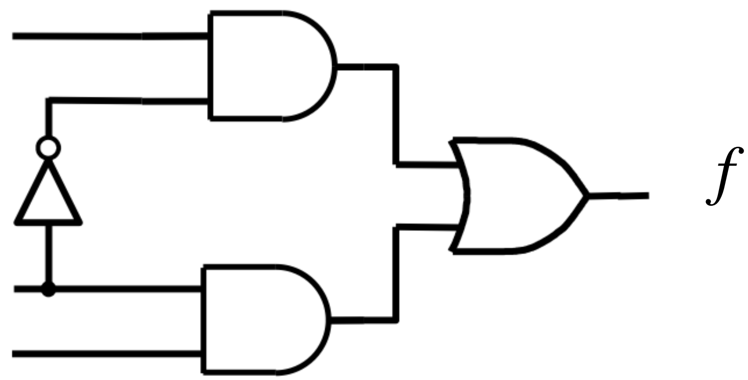
(b) Modified truth table



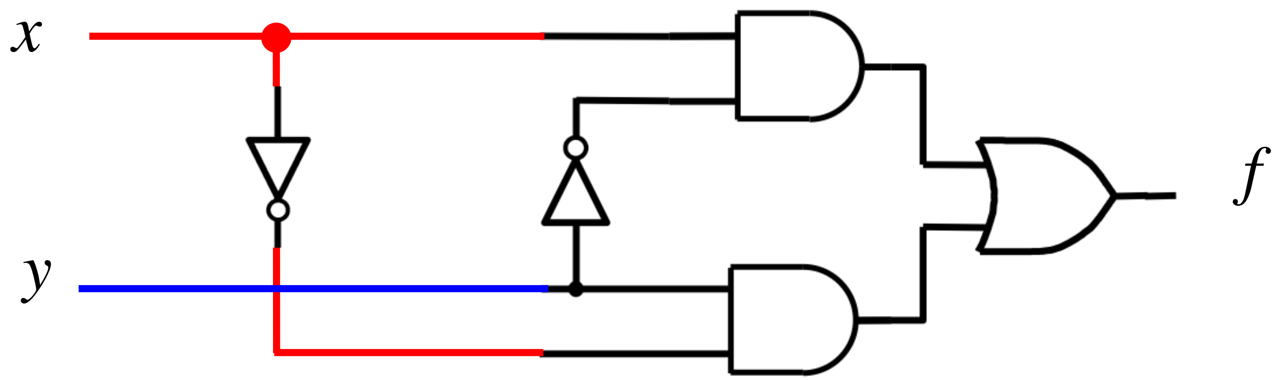
(c) Circuit

[Figure 4.6b-c from the textbook]

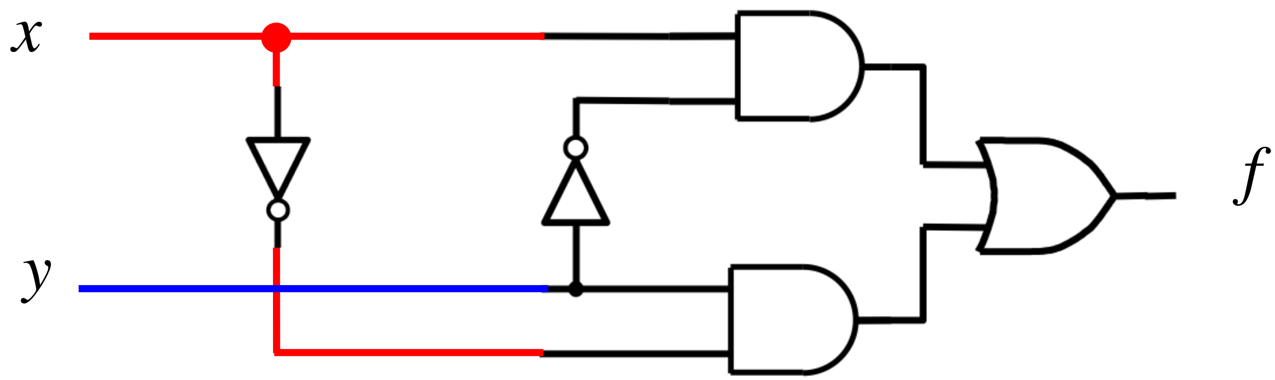
Implementation of the XOR Logic Gate with a 2-to-1 multiplexer and one NOT



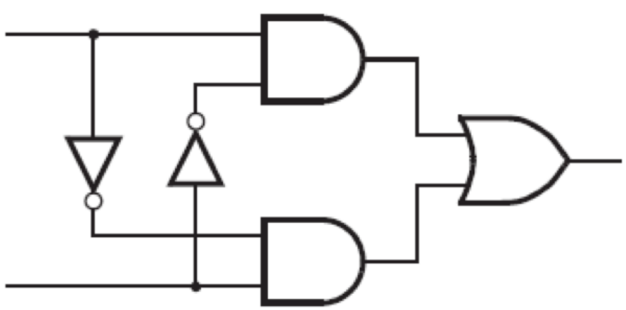
Implementation of the XOR Logic Gate with a 2-to-1 multiplexer and one NOT



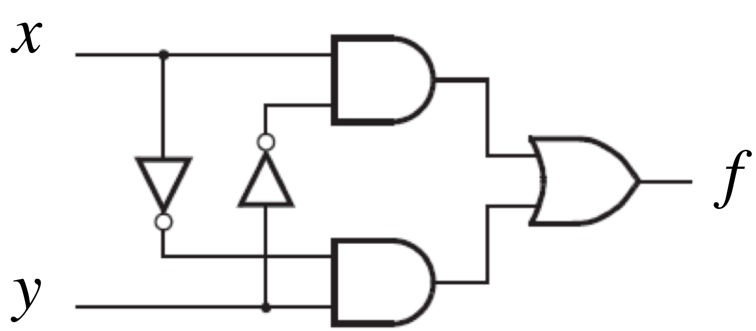
Implementation of the XOR Logic Gate with a 2-to-1 multiplexer and one NOT

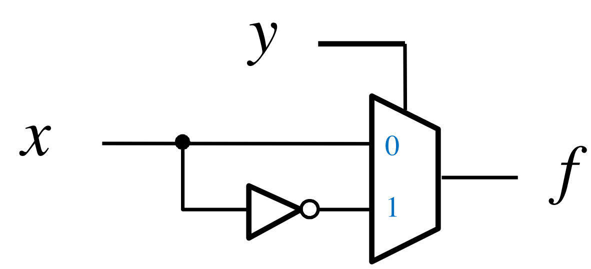


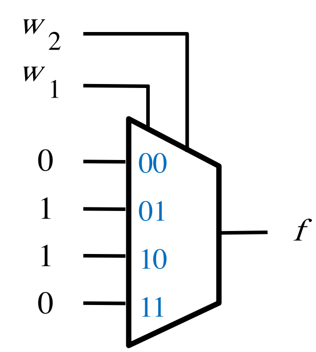
These two circuits are equivalent
(the wires of the bottom AND gate are flipped)



In other words, all four of these are equivalent!







Implementation of another logic function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

[Figure 4.7 from the textbook]

Implementation of another logic function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

[Figure 4.7 from the textbook]

Implementation of another logic function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

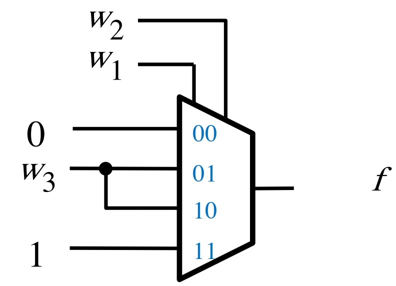
w_1	w_2	f
0	0	0
0	1	w_3
1	0	w_3
1	1	1

[Figure 4.7 from the textbook]

Implementation of another logic function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	w_2	f
0	0	0
0	1	w_3
1	0	w_3
1	1	1



[Figure 4.7 from the textbook]

Another Example
(3-input XOR)

Implementation of 3-input XOR with 2-to-1 Multiplexers

w_1	w_2	w_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

[Figure 4.8a from the textbook]

Implementation of 3-input XOR with 2-to-1 Multiplexers

w_1	w_2	w_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$w_2 \oplus w_3$

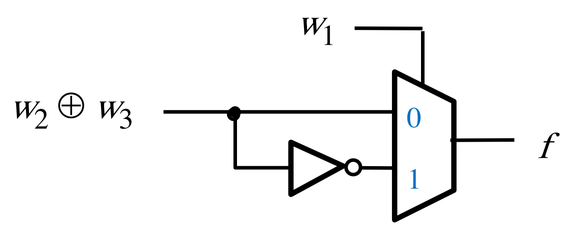
$\overline{w_2 \oplus w_3}$

[Figure 4.8a from the textbook]

Implementation of 3-input XOR with 2-to-1 Multiplexers

w_1	w_2	w_3	f	
0	0	0	0	
0	0	1	1	$w_2 \oplus w_3$
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	0	$\overline{w_2 \oplus w_3}$
1	1	0	0	
1	1	1	1	

(a) Truth table



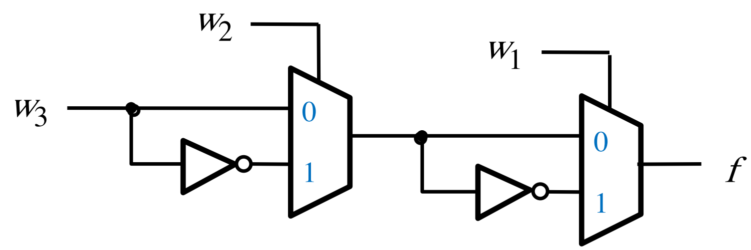
(b) Circuit

[Figure 4.8 from the textbook]

Implementation of 3-input XOR with 2-to-1 Multiplexers

w_1	w_2	w_3	f	
0	0	0	0	w_3
0	0	1	1	
0	1	0	1	$\overline{w_3}$
0	1	1	0	
1	0	0	1	
1	0	1	0	
1	1	0	0	
1	1	1	1	

(a) Truth table



(b) Circuit

[Figure 4.8 from the textbook]

Implementation of 3-input XOR with a 4-to-1 Multiplexer

w_1	w_2	w_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

[Figure 4.9a from the textbook]

Implementation of 3-input XOR with a 4-to-1 Multiplexer

w_1	w_2	w_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

[Figure 4.9a from the textbook]

Implementation of 3-input XOR with a 4-to-1 Multiplexer

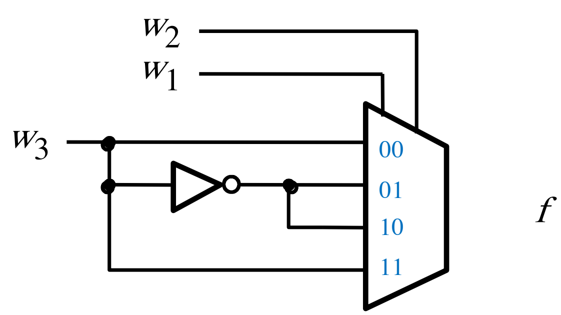
w_1	w_2	w_3	f	
0	0	0	0	w_3
0	0	1	1	
0	1	0	1	$\overline{w_3}$
0	1	1	0	
1	0	0	1	$\overline{w_3}$
1	0	1	0	
1	1	0	0	w_3
1	1	1	1	

[Figure 4.9a from the textbook]

Implementation of 3-input XOR with a 4-to-1 Multiplexer

w_1	w_2	w_3	f	
0	0	0	0	w_3
0	0	1	1	
0	1	0	1	$\overline{w_3}$
0	1	1	0	
1	0	0	1	$\overline{w_3}$
1	0	1	0	
1	1	0	0	w_3
1	1	1	1	

(a) Truth table



(b) Circuit

[Figure 4.9 from the textbook]

Circuit Synthesis with Multiplexers Using Shannon's Expansion

Three-input majority function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

[Figure 4.10a from the textbook]

Three-input majority function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

SOP expression for *f* :

$$f = \overline{w}_1 w_2 w_3 + w_1 \overline{w}_2 w_3 + w_1 w_2 \overline{w}_3 + w_1 w_2 w_3$$

[Figure 4.10a from the textbook]

Three-input majority function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	f
0	
1	

Divide-and-conquer method (a.k.a., Shannon's expansion)

[Figure 4.10a from the textbook]

Three-input majority function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	f
0	$w_2 w_3$
1	$w_2 + w_3$

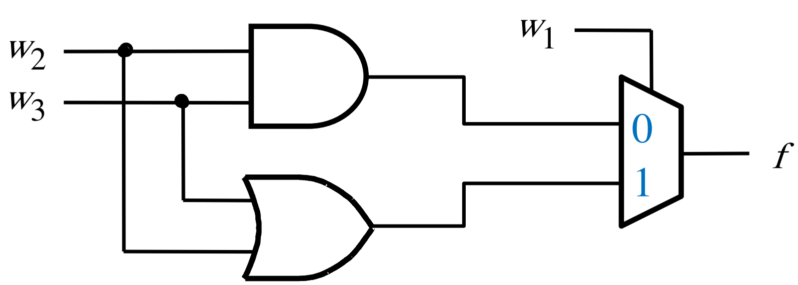
[Figure 4.10a from the textbook]

Three-input majority function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	f
0	$w_2 w_3$
1	$w_2 + w_3$

(b) Truth table



(b) Circuit

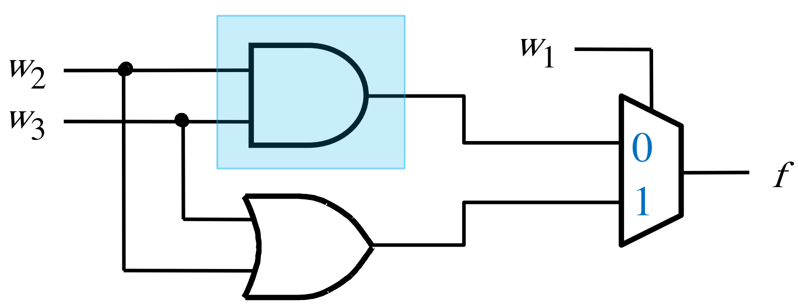
[Figure 4.10a from the textbook]

Three-input majority function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	f
0	$w_2 w_3$
1	$w_2 + w_3$

(b) Truth table



(b) Circuit

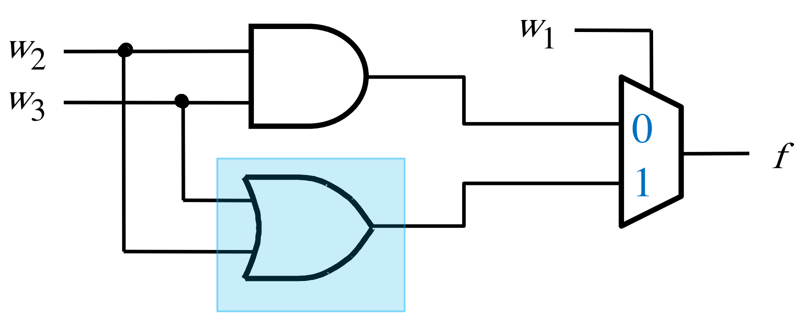
[Figure 4.10a from the textbook]

Three-input majority function

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	f
0	$w_2 w_3$
1	$w_2 + w_3$

(b) Truth table



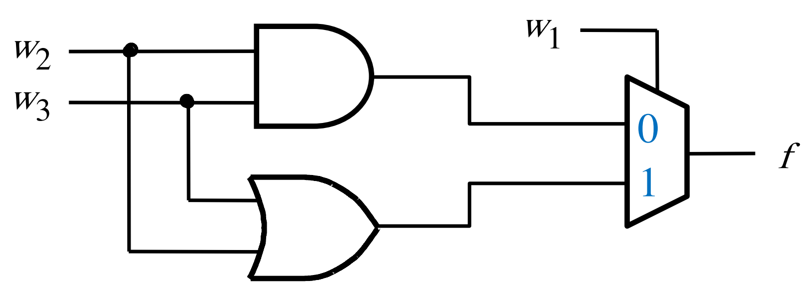
(b) Circuit

[Figure 4.10a from the textbook]

Three-input majority function

$$f = \overline{w}_1 w_2 w_3 + w_1 \overline{w}_2 w_3 + w_1 w_2 \overline{w}_3 + w_1 w_2 w_3$$

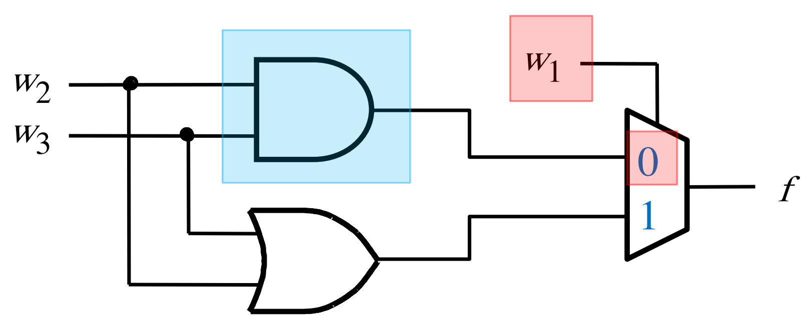
$$\begin{aligned} f &= \overline{w}_1 (w_2 w_3) + w_1 (\overline{w}_2 w_3 + w_2 \overline{w}_3 + w_2 w_3) \\ &= \overline{w}_1 (w_2 w_3) + w_1 (w_2 + w_3) \end{aligned}$$



Three-input majority function

$$f = \overline{w}_1 w_2 w_3 + w_1 \overline{w}_2 w_3 + w_1 w_2 \overline{w}_3 + w_1 w_2 w_3$$

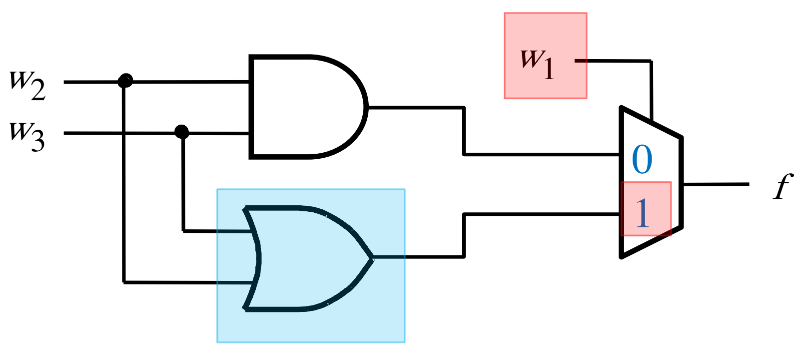
$$\begin{aligned} f &= \overline{w}_1 (w_2 w_3) + w_1 (\overline{w}_2 w_3 + w_2 \overline{w}_3 + w_2 w_3) \\ &= \boxed{\overline{w}_1} \boxed{(w_2 w_3)} + w_1 (w_2 + w_3) \end{aligned}$$



Three-input majority function

$$f = \overline{w}_1 w_2 w_3 + w_1 \overline{w}_2 w_3 + w_1 w_2 \overline{w}_3 + w_1 w_2 w_3$$

$$\begin{aligned} f &= \overline{w}_1 (w_2 w_3) + w_1 (\overline{w}_2 w_3 + w_2 \overline{w}_3 + w_2 w_3) \\ &= \overline{w}_1 (w_2 w_3) + w_1 (w_2 + w_3) \end{aligned}$$



Shannon's Expansion Theorem

Shannon's Expansion Theorem

Any Boolean function $f(w_1, \ldots, w_n)$ can be rewritten in the form:

$$f(w_1, w_2, \ldots, w_n) = \overline{w}_1 \cdot f(0, w_2, \ldots, w_n) + w_1 \cdot f(1, w_2, \ldots, w_n)$$

Shannon's Expansion Theorem

Any Boolean function $f(w_1, \ldots, w_n)$ can be rewritten in the form:

$$f(w_1, w_2, \ldots, w_n) = \overline{w}_1 \cdot f(0, w_2, \ldots, w_n) + w_1 \cdot f(1, w_2, \ldots, w_n)$$

$$f = \overline{w}_1 f_{\overline{w}_1} + w_1 f_{w_1}$$

Shannon's Expansion Theorem

Any Boolean function $f(w_1, \ldots, w_n)$ can be rewritten in the form:

$$f(w_1, w_2, \ldots, w_n) = \overline{w}_1 \cdot f(0, w_2, \ldots, w_n) + w_1 \cdot f(1, w_2, \ldots, w_n)$$

$$f = \overline{w}_1 \underbrace{f_{\overline{w}_1}}_{\text{cofactor}} + w_1 \underbrace{f_{w_1}}_{\text{cofactor}}$$

Shannon's Expansion Theorem
(example with only one select variable)
(used for 2-to-1 mux implementation)

Factor and implement the following function with a 2-to-1 multiplexer

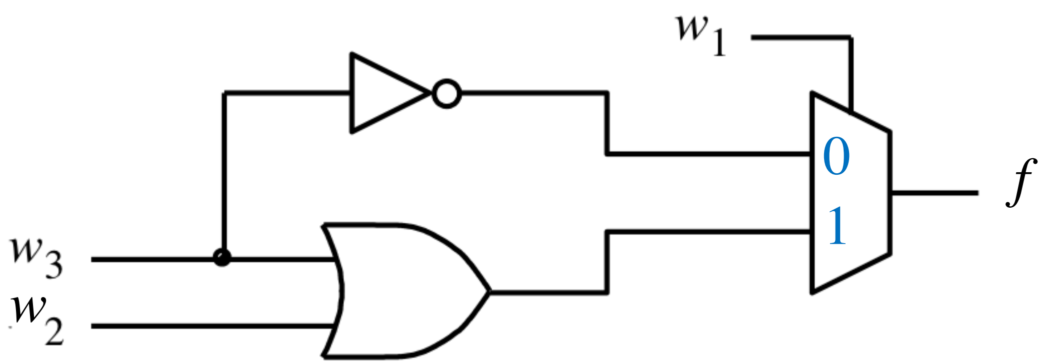
$$f = \overline{w}_1\overline{w}_3 + w_1w_2 + w_1w_3$$

Factor and implement the following function with a 2-to-1 multiplexer

$$f = \overline{w}_1\overline{w}_3 + w_1w_2 + w_1w_3$$

$$\begin{aligned} f &= \overline{w}_1 f_{\overline{w}_1} + w_1 f_{w_1} \\ &= \overline{w}_1(\overline{w}_3) + w_1(w_2 + w_3) \end{aligned}$$

Factor and implement the following function with a 2-to-1 multiplexer



$$f = \overline{w}_1 f_{\overline{w}_1} + w_1 f_{w_1}$$
$$= \overline{w}_1(\overline{w}_3) + w_1(w_2 + w_3)$$

[Figure 4.11a from the textbook]

Another Example

Shannon's Expansion Theorem Example

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

Shannon's Expansion Theorem Example

set $w_1 = 0$

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

Shannon's Expansion Theorem Example

set $w_1 = 0$

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

$$f_{\overline{w_1}}(0, w_2, w_3) = \underbrace{0 \cdot w_2 + 0 \cdot w_3 + w_2 w_3}_{w_2 w_3}$$

Shannon's Expansion Theorem Example

set $w_1 = 1$

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

Shannon's Expansion Theorem Example

set $w_1 = 1$

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

$$f(\underset{w_1}{\mathbf{1}}, w_2, w_3) = \underbrace{\mathbf{1} \bullet w_2 + \mathbf{1} \bullet w_3 + w_2 w_3}_{w_2 + w_3}$$

Shannon's Expansion Theorem Example

started with this

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

Shannon's Expansion Theorem Example

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

$$\begin{aligned} f &= \overline{w}_1(0 \cdot w_2 + 0 \cdot w_3 + w_2 w_3) + w_1(1 \cdot w_2 + 1 \cdot w_3 + w_2 w_3) \\ &= \overline{w}_1(w_2 w_3) + w_1(w_2 + w_3) \end{aligned}$$

factored it in this form

Shannon's Expansion Theorem Example

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

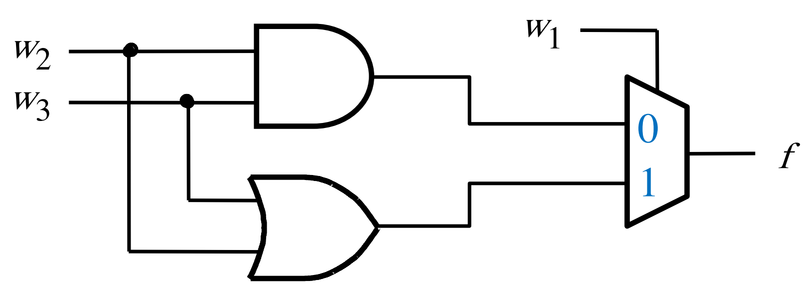
$$\begin{aligned} f &= \overline{w}_1(0 \cdot w_2 + 0 \cdot w_3 + w_2 w_3) + w_1(1 \cdot w_2 + 1 \cdot w_3 + w_2 w_3) \\ &= \overline{w}_1(w_2 w_3) + w_1(w_2 + w_3) \end{aligned}$$

$$f = \overline{w}_1 f_{\overline{w}_1} + w_1 f_{w_1}$$

which fits the general formula

Shannon's Expansion Theorem Example

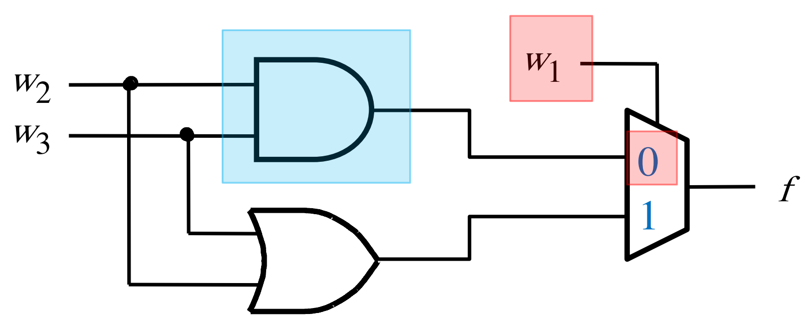
$$
\begin{aligned}
f &= \overline{w}_1(0 \cdot w_2 + 0 \cdot w_3 + w_2 w_3) + w_1(1 \cdot w_2 + 1 \cdot w_3 + w_2 w_3) \\
&= \overline{w}_1(w_2 w_3) + w_1(w_2 + w_3)
\end{aligned}
$$



Shannon's Expansion Theorem Example

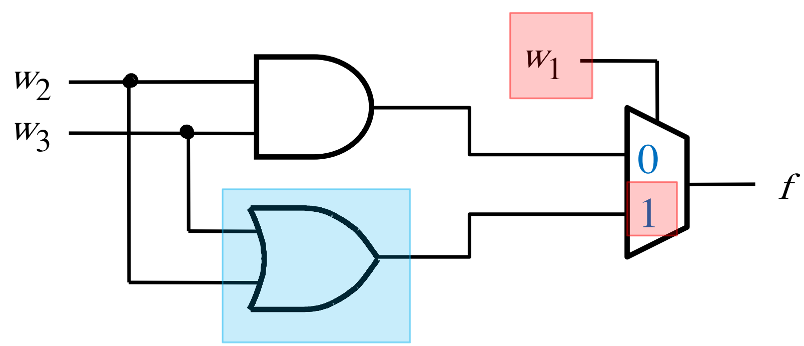
$$f = \overline{w}_1(0 \cdot w_2 + 0 \cdot w_3 + w_2 w_3) + w_1(1 \cdot w_2 + 1 \cdot w_3 + w_2 w_3)$$

$$= \overline{w}_1 (w_2 w_3) + w_1(w_2 + w_3)$$



Shannon's Expansion Theorem Example

$$
\begin{aligned}
f &= \overline{w}_1(0 \cdot w_2 + 0 \cdot w_3 + w_2w_3) + w_1(1 \cdot w_2 + 1 \cdot w_3 + w_2w_3) \\
&= \overline{w}_1(w_2w_3) + \boxed{w_1}\boxed{(w_2 + w_3)}
\end{aligned}
$$



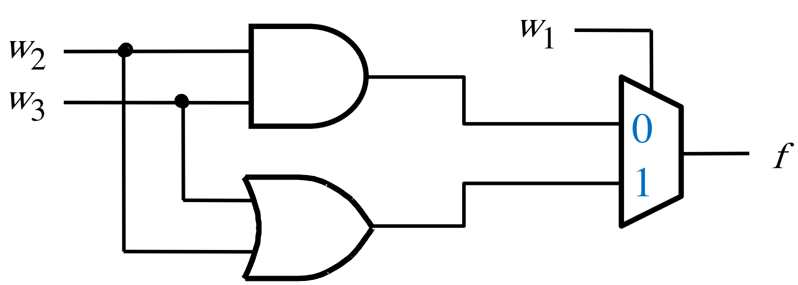
Earlier we derived the same result from the truth table

using the divide-and-conquer method (a.k.a., Shannon's theorem)

w_1	w_2	w_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

w_1	f
0	$w_2 w_3$
1	$w_2 + w_3$

(b) Truth table



(b) Circuit

[Figure 4.10a from the textbook]

Shannon's Expansion Theorem
(example with two select variables)
(used for 4-to-1 mux implementation)

Shannon's Expansion Theorem
(In terms of more than one variable)

$$\begin{aligned} f(w_1, \ldots, w_n) = {} & \overline{w}_1 \overline{w}_2 \cdot f(0, 0, w_3, \ldots, w_n) + \overline{w}_1 w_2 \cdot f(0, 1, w_3, \ldots, w_n) \\ & + w_1 \overline{w}_2 \cdot f(1, 0, w_3, \ldots, w_n) + w_1 w_2 \cdot f(1, 1, w_3, \ldots, w_n) \end{aligned}$$

This form is suitable for implementation with a 4x1 multiplexer.

Factor and implement the following function with a 4-to-1 multiplexer

$$f(w_1, w_2, w_3) = \overline{w}_1\overline{w}_3 + w_1 w_2 + w_1 w_3$$

Factor and implement the following function with a 4-to-1 multiplexer

$w_1 = 0$ $w_2 = 0$

$$f(w_1, w_2, w_3) = \underbrace{1 \cdot \overline{w}_3 + 0 \cdot 0 + 0 \cdot w_3}_{\overline{w}_3}$$

Factor and implement the following function with a 4-to-1 multiplexer

$w_1 = 0$ $w_2 = 1$

$$f(w_1, w_2, w_3) = \underbrace{1 \bullet \overline{w}_3 + 0 \bullet 1 + 0 \bullet w_3}_{\overline{w}_3}$$

Factor and implement the following function with a 4-to-1 multiplexer

$w_1 = 1 \qquad w_2 = 0$

$$f(w_1, w_2, w_3) = \underbrace{0 \bullet \overline{w}_3 + 1 \bullet 0 + 1 \bullet w_3}_{w_3}$$

Factor and implement the following function with a 4-to-1 multiplexer

$w_1 = 1$ $w_2 = 1$

$$f(w_1, w_2, w_3) = \underbrace{0 \cdot \overline{w}_3 + 1 \cdot 1 + 1 \cdot w_3}_{1}$$

Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1\overline{w}_3 + w_1 w_2 + w_1 w_3$$

$$\begin{aligned} f &= \overline{w}_1\overline{w}_2 f_{\overline{w}_1\overline{w}_2} + \overline{w}_1 w_2 f_{\overline{w}_1 w_2} + w_1\overline{w}_2 f_{w_1\overline{w}_2} + w_1 w_2 f_{w_1 w_2} \\ &= \overline{w}_1\overline{w}_2(\overline{w}_3) + \overline{w}_1 w_2(\overline{w}_3) + w_1\overline{w}_2(w_3) + w_1 w_2(1) \end{aligned}$$

Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1\overline{w}_3 + w_1w_2 + w_1w_3$$

$$\begin{aligned} f &= \overline{w}_1\overline{w}_2 f_{\overline{w}_1\overline{w}_2} + \overline{w}_1 w_2 f_{\overline{w}_1 w_2} + w_1\overline{w}_2 f_{w_1\overline{w}_2} + w_1 w_2 f_{w_1 w_2} \\ &= \overline{w}_1\overline{w}_2(\overline{w}_3) + \overline{w}_1 w_2(\overline{w}_3) + w_1\overline{w}_2(w_3) + w_1 w_2(1) \end{aligned}$$

these are the 4 minterms in terms of w_1 and w_2

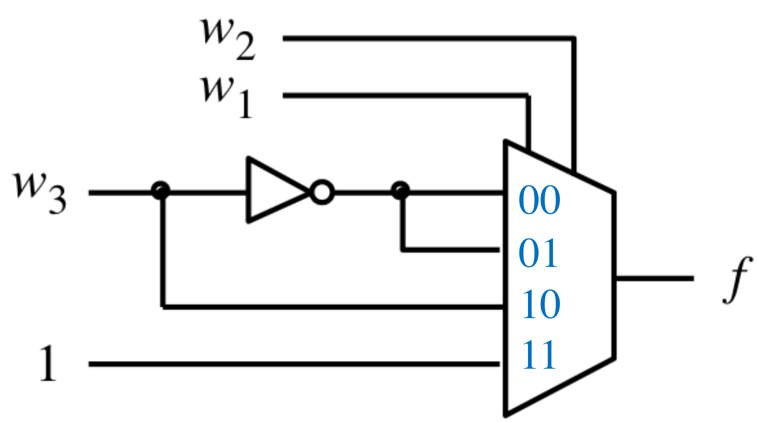
Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1 \overline{w}_3 + w_1 w_2 + w_1 w_3$$

$$\begin{aligned} f &= \overline{w}_1 \overline{w}_2 f_{\overline{w}_1 \overline{w}_2} + \overline{w}_1 w_2 f_{\overline{w}_1 w_2} + w_1 \overline{w}_2 f_{w_1 \overline{w}_2} + w_1 w_2 f_{w_1 w_2} \\ &= \overline{w}_1 \overline{w}_2 (\overline{w}_3) + \overline{w}_1 w_2 (\overline{w}_3) + w_1 \overline{w}_2 (w_3) + w_1 w_2 (1) \end{aligned}$$

these are the 4 cofactors

Factor and implement the following function with a 4-to-1 multiplexer



$$f = \overline{w}_1\overline{w}_2 f_{\overline{w}_1\overline{w}_2} + \overline{w}_1 w_2 f_{\overline{w}_1 w_2} + w_1\overline{w}_2 f_{w_1\overline{w}_2} + w_1 w_2 f_{w_1 w_2}$$

$$= \overline{w}_1\overline{w}_2(\overline{w}_3) + \overline{w}_1 w_2(\overline{w}_3) + w_1\overline{w}_2(w_3) + w_1 w_2(1)$$

[Figure 4.11b from the textbook]

Alternative Derivation for the same Problem

(using Boolean Algebra to derive the cofactors)

Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1\overline{w}_3 + w_1w_2 + w_1w_3$$

Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1\overline{w}_3 + w_1w_2 + w_1w_3$$

$$= \overline{w}_1 \, (\overline{w}_2 + w_2) \, \overline{w}_3 + w_1 \, w_2 + w_1 \, (\overline{w}_2 + w_2) \, w_3$$

Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1 \overline{w}_3 + w_1 w_2 + w_1 w_3$$

$$= \overline{w}_1 (\overline{w}_2 + w_2) \overline{w}_3 + w_1 w_2 + w_1 (\overline{w}_2 + w_2) w_3$$

$$= \overline{w}_1 \overline{w}_2 \overline{w}_3 + \overline{w}_1 w_2 \overline{w}_3 + w_1 w_2 + w_1 \overline{w}_2 w_3 + w_1 w_2 w_3$$

$$= \overline{w}_1 \overline{w}_2 \overline{w}_3 + \overline{w}_1 w_2 \overline{w}_3 + w_1 \overline{w}_2 w_3 + w_1 w_2 (1 + w_3)$$

$$= \overline{w}_1 \overline{w}_2 (\overline{w}_3) + \overline{w}_1 w_2 (\overline{w}_3) + w_1 \overline{w}_2 (w_3) + w_1 w_2 (1)$$

Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1\overline{w}_3 + w_1 w_2 + w_1 w_3$$

$$= \overline{w_1}\,(\overline{w_2} + w_2)\,\overline{w_3} + w_1 w_2 + w_1\,(\overline{w_2} + w_2)\,w_3$$

$$= \overline{w_1}\,\overline{w_2}\,\overline{w_3} + \overline{w_1}\,w_2\,\overline{w_3} + w_1 w_2 + w_1\,\overline{w_2}\,w_3 + w_1 w_2 w_3$$

$$= \overline{w_1}\,\overline{w_2}\,\overline{w_3} + \overline{w_1}\,w_2\,\overline{w_3} + w_1\,\overline{w_2}\,w_3 + w_1 w_2\,(1 + w_3)$$

$$= \overline{w_1}\,\overline{w_2}\,(\overline{w_3}) + \overline{w_1}\,w_2\,(\overline{w_3}) + w_1\,\overline{w_2}\,(w_3) + w_1 w_2\,(1)$$

these are the 4 minterms in terms of w_1 and w_2

Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1\overline{w}_3 + w_1 w_2 + w_1 w_3$$

$$= \overline{w}_1 (\overline{w}_2 + w_2)\, \overline{w}_3 + w_1 w_2 + w_1 (\overline{w}_2 + w_2)\, w_3$$

$$= \overline{w}_1 \overline{w}_2 \overline{w}_3 + \overline{w}_1 w_2 \overline{w}_3 + w_1 w_2 + w_1 \overline{w}_2 w_3 + w_1 w_2 w_3$$

$$= \overline{w}_1 \overline{w}_2 \overline{w}_3 + \overline{w}_1 w_2 \overline{w}_3 + w_1 \overline{w}_2 w_3 + w_1 w_2 (1 + w_3)$$

$$= \overline{w}_1 \overline{w}_2 (\overline{w}_3) + \overline{w}_1 w_2 (\overline{w}_3) + w_1 \overline{w}_2 (w_3) + w_1 w_2 (1)$$

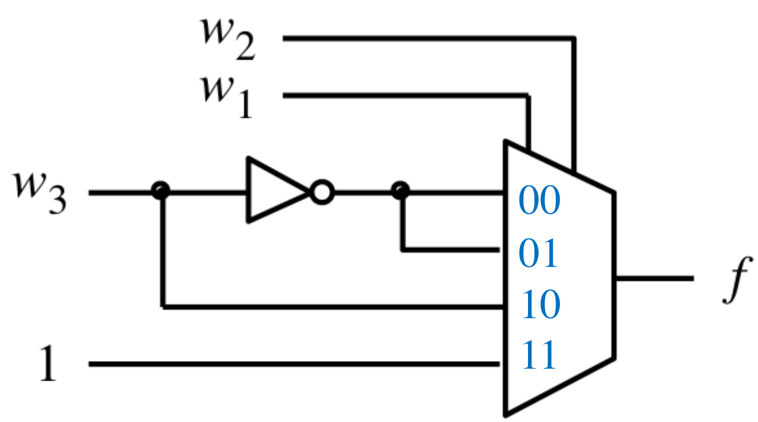
these are the 4 cofactors

Factor and implement the following function with a 4-to-1 multiplexer

$$f = \overline{w}_1\overline{w}_3 + w_1w_2 + w_1w_3$$

$$\begin{aligned} f &= \overline{w}_1\overline{w}_2 f_{\overline{w}_1\overline{w}_2} + \overline{w}_1 w_2 f_{\overline{w}_1 w_2} + w_1\overline{w}_2 f_{w_1\overline{w}_2} + w_1 w_2 f_{w_1 w_2} \\ &= \overline{w}_1\overline{w}_2(\overline{w}_3) + \overline{w}_1 w_2(\overline{w}_3) + w_1\overline{w}_2(w_3) + w_1 w_2(1) \end{aligned}$$

Factor and implement the following function with a 4-to-1 multiplexer



$$f = \overline{w}_1\overline{w}_2 f_{\overline{w}_1\overline{w}_2} + \overline{w}_1 w_2 f_{\overline{w}_1 w_2} + w_1\overline{w}_2 f_{w_1\overline{w}_2} + w_1 w_2 f_{w_1 w_2}$$
$$= \overline{w}_1\overline{w}_2(\overline{w}_3) + \overline{w}_1 w_2(\overline{w}_3) + w_1\overline{w}_2(w_3) + w_1 w_2(1)$$

[Figure 4.11b from the textbook]

Yet Another Example
(with hierarchical structure)

Factor and implement the following function using <u>only</u> 2-to-1 multiplexers

$$f = w_1 w_2 + w_1 w_3 + w_2 w_3$$

Factor and implement the following function using <u>only</u> 2-to-1 multiplexers

$$f = w_1 w_2 + w_1 w_3 + w_2 w_3$$

$$\begin{aligned} f &= \overline{w}_1(w_2 w_3) + w_1(w_2 + w_3 + w_2 w_3) \\ &= \overline{w}_1(w_2 w_3) + w_1(w_2 + w_3) \end{aligned}$$

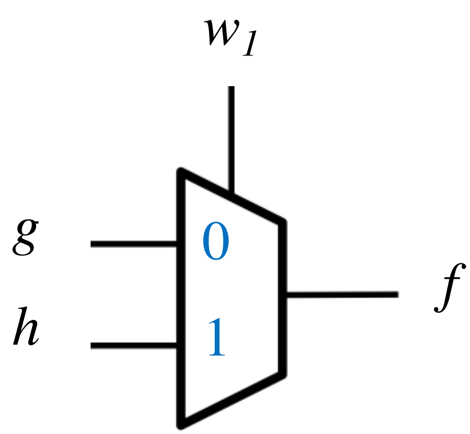
Factor and implement the following function using <u>only</u> 2-to-1 multiplexers

$$f = w_1 w_2 + w_1 w_3 + w_2 w_3$$

$$\begin{aligned} f &= \overline{w}_1 (w_2 w_3) + w_1 (w_2 + w_3 + w_2 w_3) \\ &= \overline{w}_1 (\underbrace{w_2 w_3}) + w_1 (\underbrace{w_2 + w_3}) \end{aligned}$$

$$g = w_2 w_3 \qquad h = w_2 + w_3$$

Factor and implement the following function using <u>only</u> 2-to-1 multiplexers



$$f = \overline{w}_1(w_2w_3) + w_1(w_2 + w_3 + w_2w_3)$$

$$= \overline{w}_1(\underbrace{w_2w_3}) + w_1(\underbrace{w_2 + w_3})$$

$$g = w_2w_3 \qquad h = w_2 + w_3$$

Factor and implement the following function using <u>only</u> 2-to-1 multiplexers

$$g = w_2 w_3$$

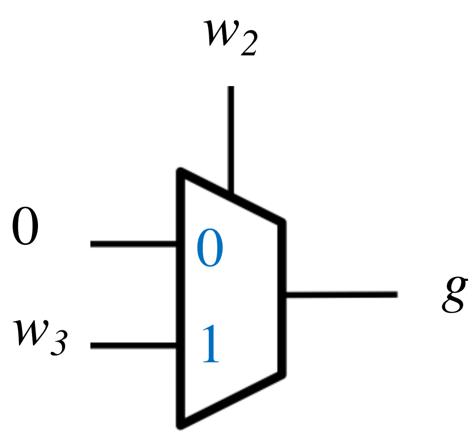
$$h = w_2 + w_3$$

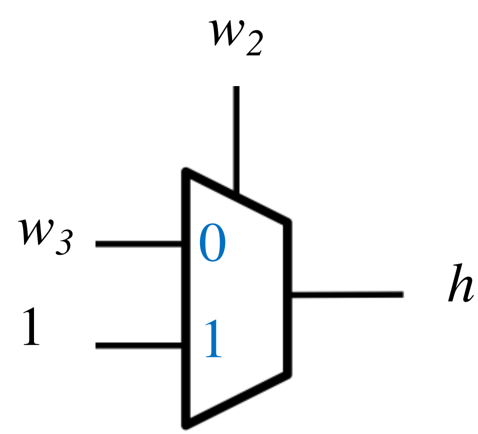
Factor and implement the following function using <u>only</u> 2-to-1 multiplexers

$$g = w_2 w_3 \qquad\qquad h = w_2 + w_3$$

$$g = \overline{w}_2(0) + w_2(w_3) \qquad\qquad h = \overline{w}_2(w_3) + w_2(1)$$

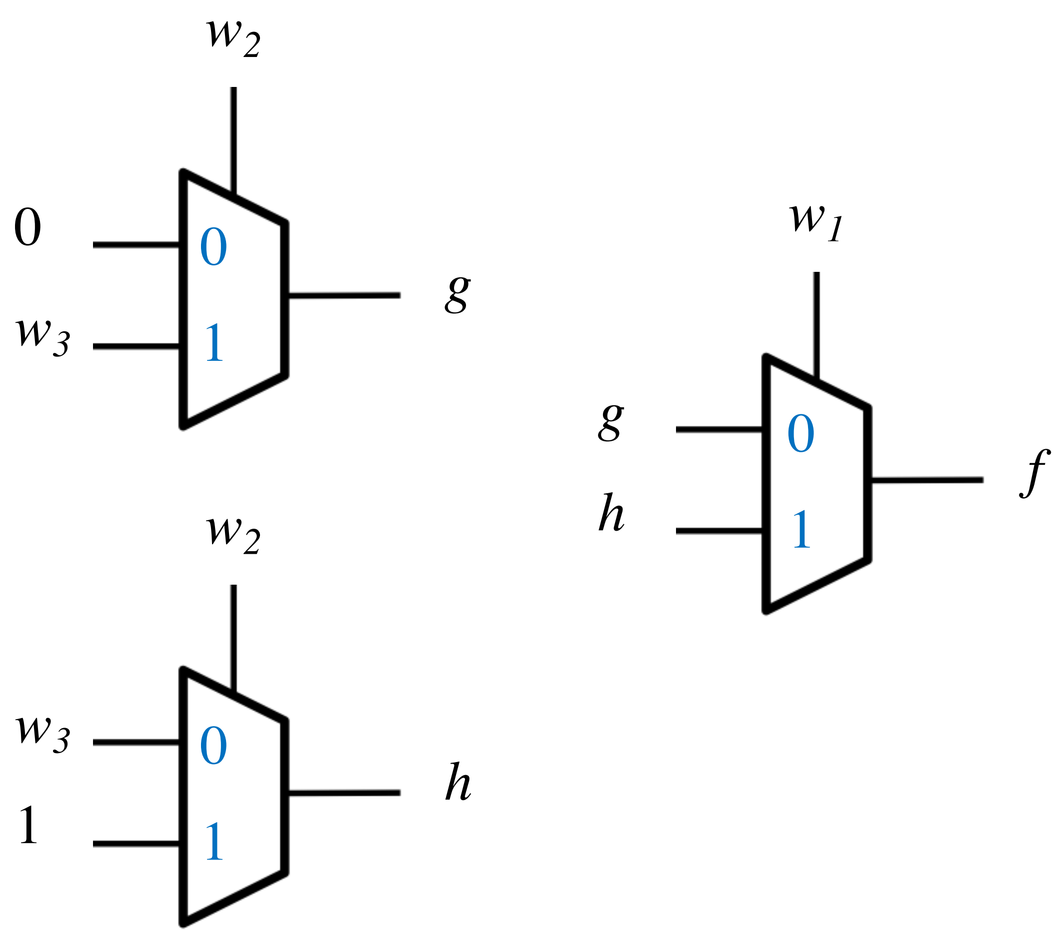
Factor and implement the following function using only 2-to-1 multiplexers



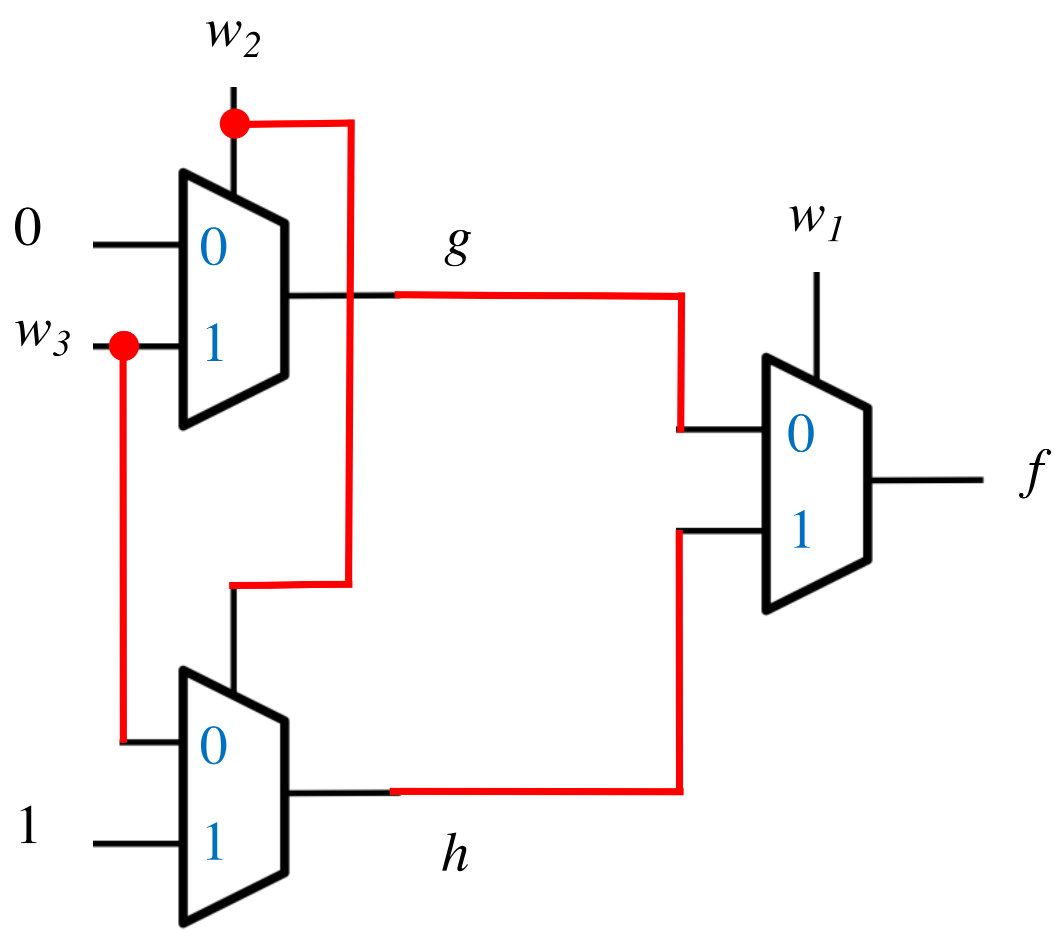


$$g = \overline{w}_2(0) + w_2(w_3) \qquad h = \overline{w}_2(w_3) + w_2(1)$$

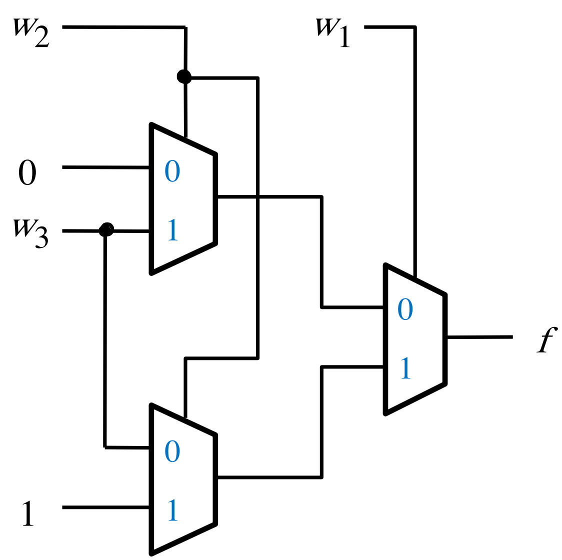
Finally, we are ready to draw the circuit



Finally, we are ready to draw the circuit



Finally, we are ready to draw the circuit



[Figure 4.12 from the textbook]

Questions?

THE END