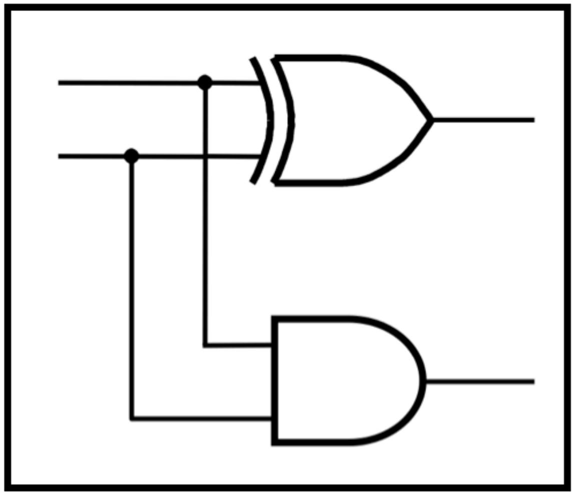


CprE 281: Digital Logic

Instructor: Alexander Stoytchev

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

Design Examples

CprE 281: Digital Logic
Iowa State University, Ames, IA

Administrative Stuff

- HW3 is out

- It is due on Monday Sep 12 @ 4pm.

- Please write clearly on the first page (in BLOCK CAPITAL letters) the following three things:

    - Your First and Last Name
    - Your Student ID Number
    - Your Lab Section Letter

- Also, please
    - Staple your pages

Administrative Stuff

TA Office Hours:

- 11:00am-1:00pm on Wednesdays (Jinyuan Jia)
  Location: TLA (Coover Hall - first floor)

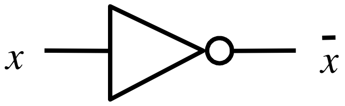
- 9:50am-11:50am on Thursday (Siyuan Lu)
  Location: TLA (Coover Hall - first floor)

Administrative Stuff

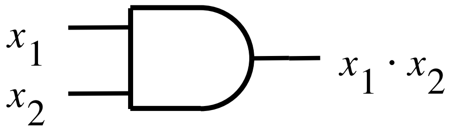
- Homework Solutions will be posted on BlackBoard

Quick Review

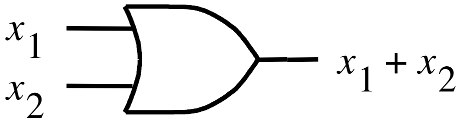
The Three Basic Logic Gates



NOT gate



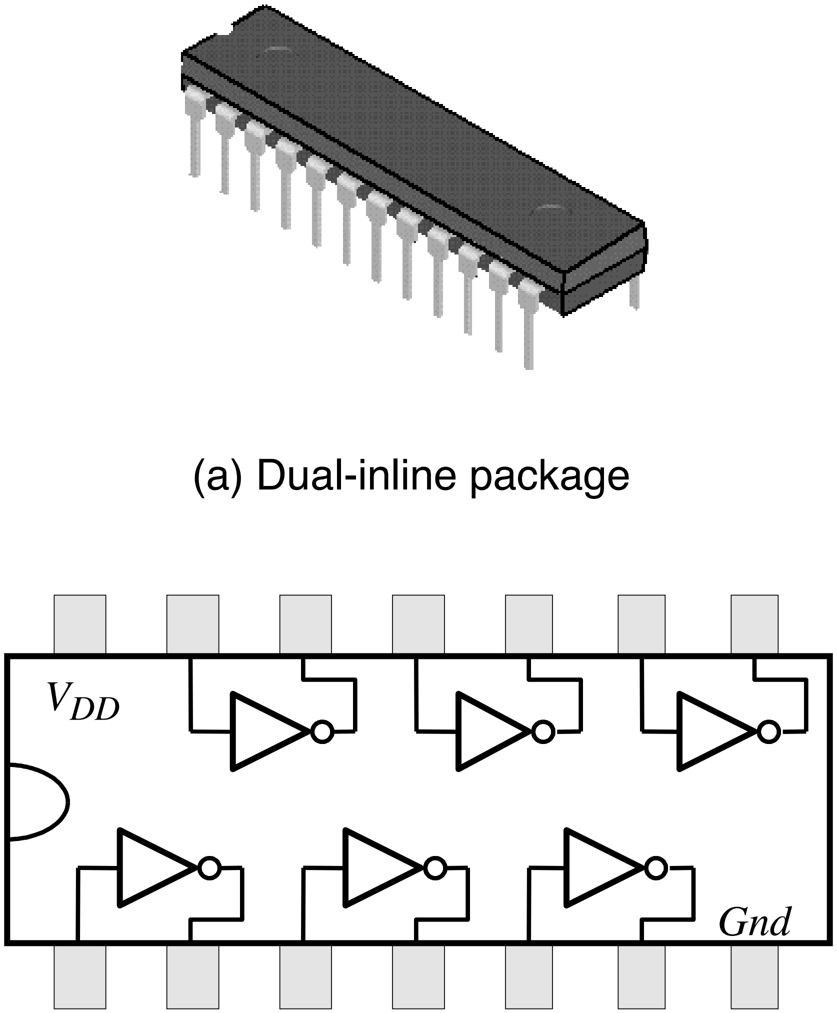
AND gate



OR gate

You can build any circuit using only these three gates

[Figure 2.8 from the textbook]



(a) Dual-inline package

(b) Structure of 7404 chip

Figure B.21. A 7400-series chip.

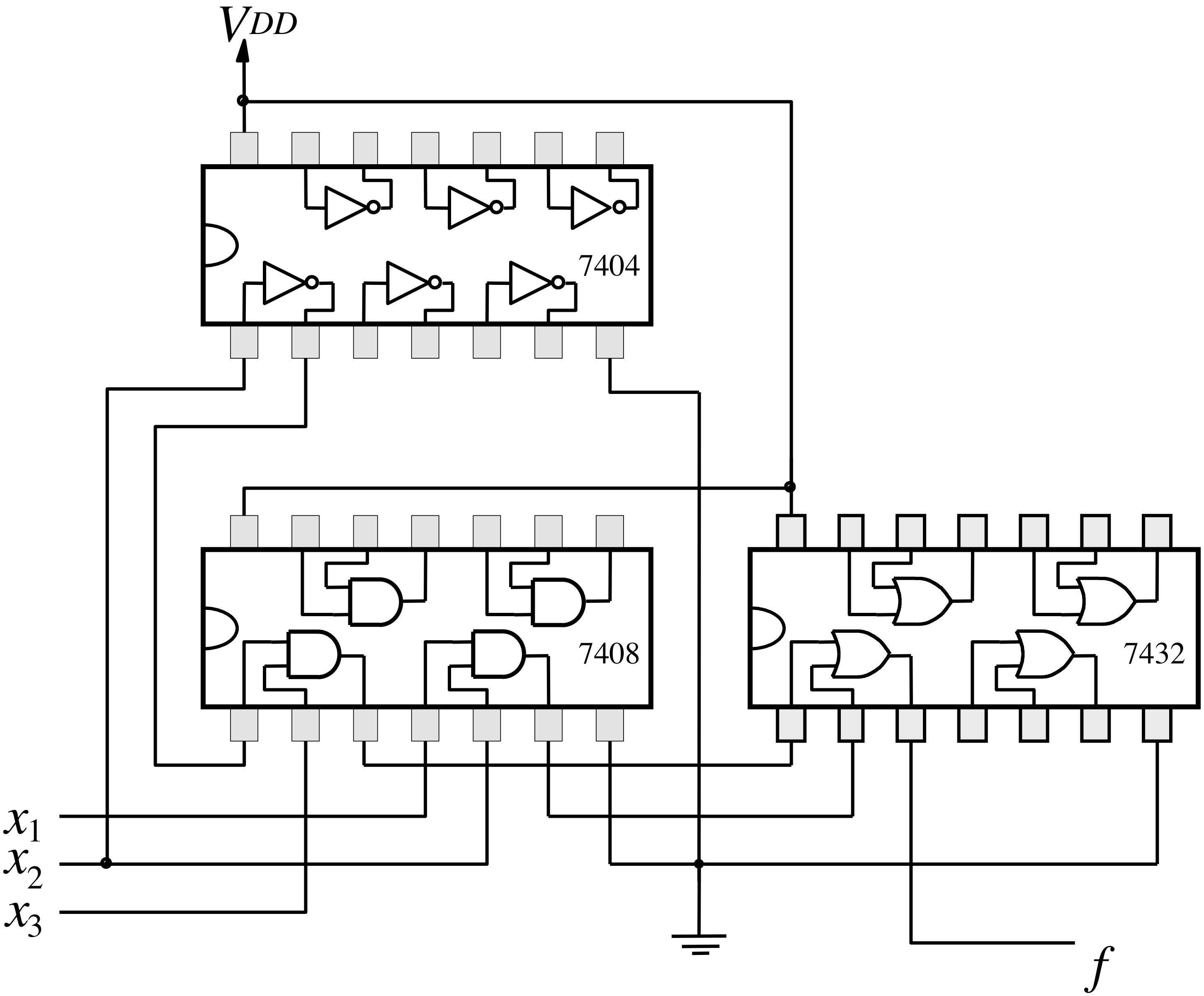
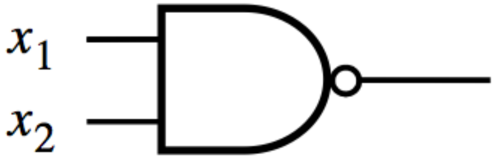


Figure B.22. An implementation of $f = x_1x_2 + \bar{x}_2x_3$.

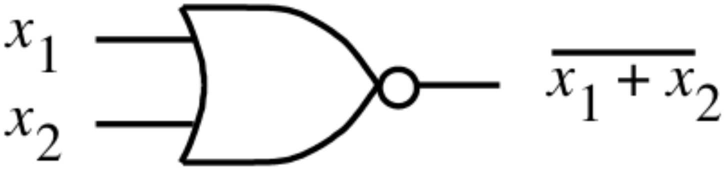
NAND Gate



$\overline{x_1 \cdot x_2}$

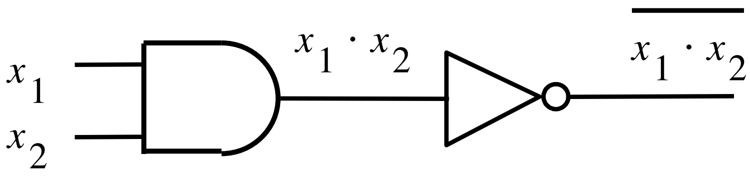
x_1	x_2	f
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate



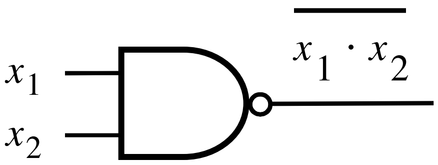
x_1	x_2	f
0	0	1
0	1	0
1	0	0
1	1	0

AND followed by NOT = NAND



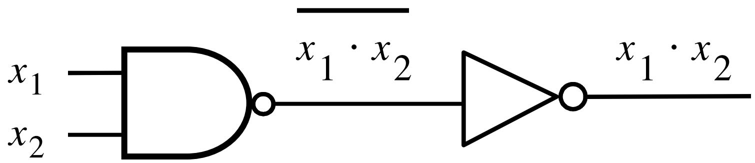
x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1

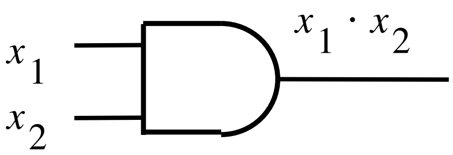
f
1
1
1
0



x_1	x_2	f
0	0	1
0	1	1
1	0	1
1	1	0

NAND followed by NOT = AND



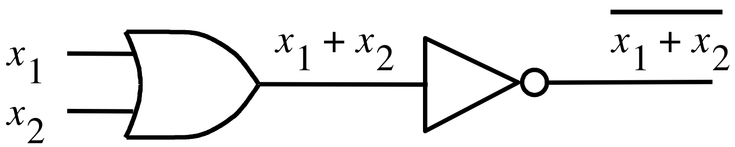


x_1	x_2	f
0	0	1
0	1	1
1	0	1
1	1	0

f
0
0
0
1

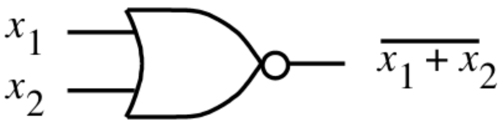
x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1

OR followed by NOT = NOR



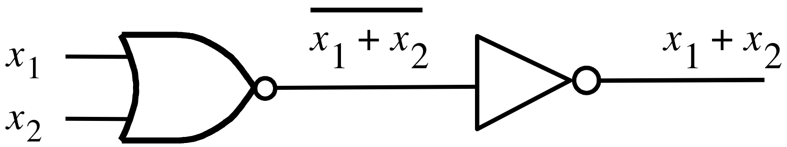
x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	1

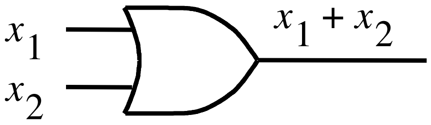
f
1
0
0
0



x_1	x_2	f
0	0	1
0	1	0
1	0	0
1	1	0

NOR followed by NOT = OR





x_1	x_2	f
0	0	1
0	1	0
1	0	0
1	1	0

f
0
1
1
1

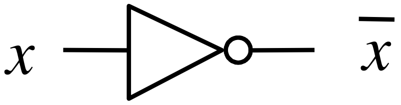
x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	1

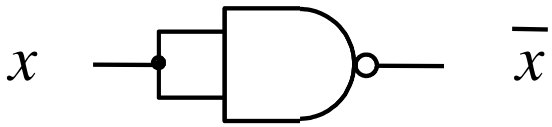
Why do we need two more gates?

They can be implemented with fewer transistors.

(more about this later)

Building a NOT Gate with NAND





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

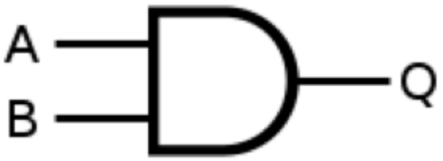
x	x	f
0	0	1
[redacted]	[redacted]	[redacted]
1	1	0

impossible
combinations

Thus, the two truth tables are equal!

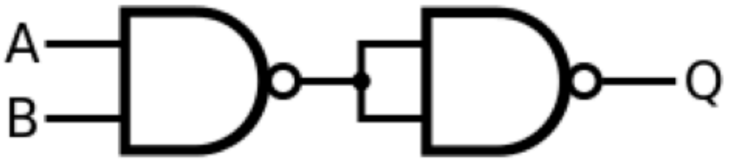
Building an AND gate with NAND gates

Desired AND Gate



Q = **A** AND **B**

NAND Construction



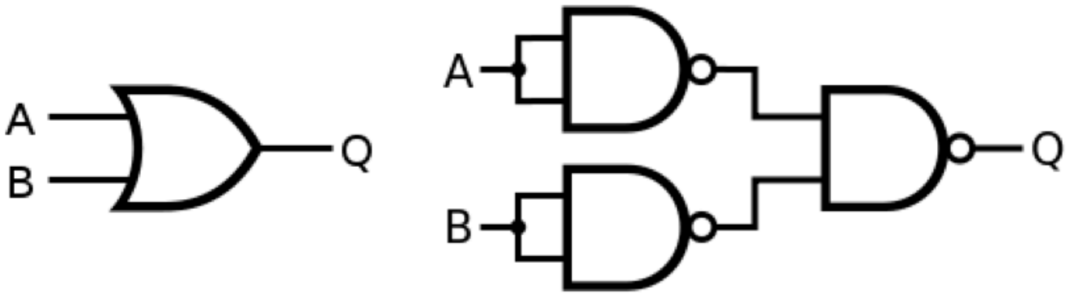
= NOT(NOT(**A** AND **B**))

Truth Table

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

[http://en.wikipedia.org/wiki/NAND_logic]

Building an OR gate with NAND gates



Desired OR Gate

NAND Construction

Q = **A** OR **B**

= NOT[NOT(**A** AND **A**) AND NOT(**B** AND **B**)]

Truth Table

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

[http://en.wikipedia.org/wiki/NAND_logic]

Implications

Any Boolean function can be implemented with only NAND gates!

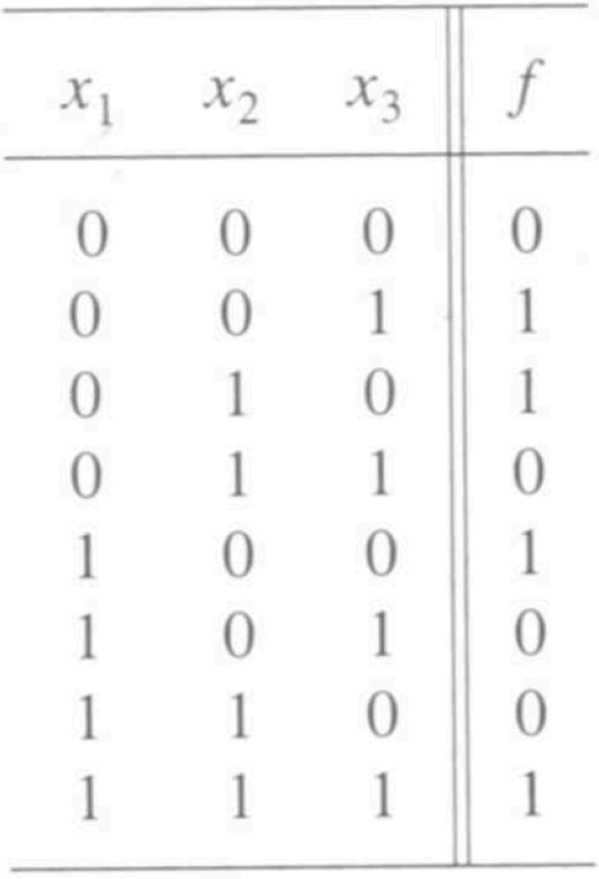
Implications

Any Boolean function can be implemented with only NAND gates!

The same is also true for NOR gates!

Another Synthesis Example

Truth table for a three-way light control



x_1	x_2	x_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

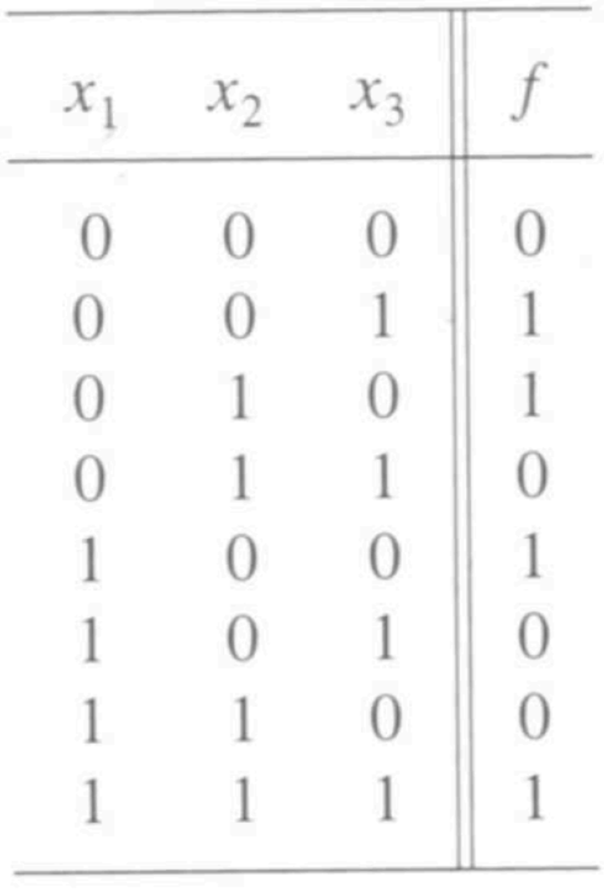
[Figure 2.31 from the textbook]

Minterms and Maxterms (with three variables)

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2 x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1 x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1 x_2 x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2 x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1 x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1 x_2 x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

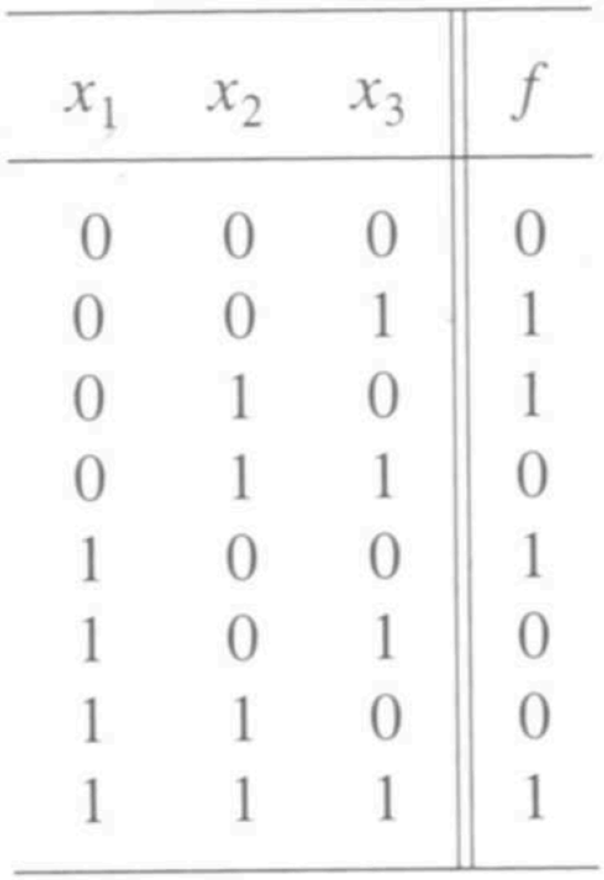
[Figure 2.22 from the textbook]

Let’s Derive the SOP form



x_1	x_2	x_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

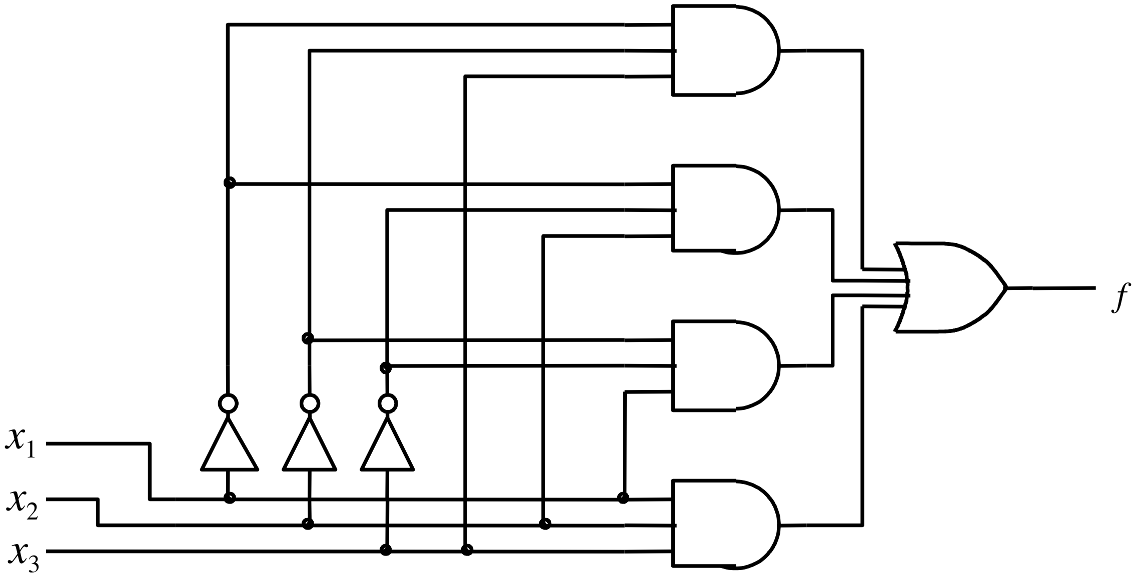
Let’s Derive the SOP form



x_1	x_2	x_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

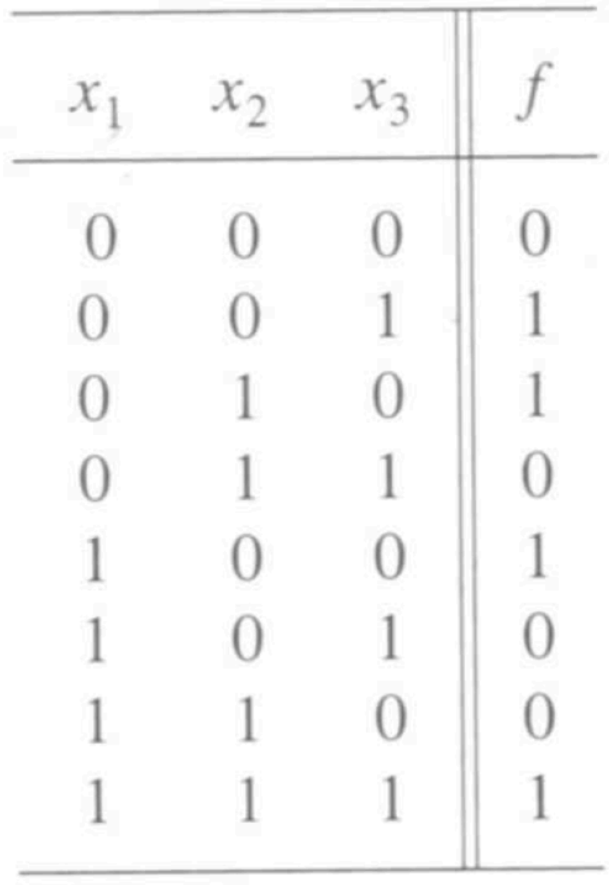
$$\begin{aligned} f &= m_1 + m_2 + m_4 + m_7 \\ &= \overline{x}_1\overline{x}_2x_3 + \overline{x}_1x_2\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 + x_1x_2x_3 \end{aligned}$$

Sum-of-products realization



[Figure 2.32a from the textbook]

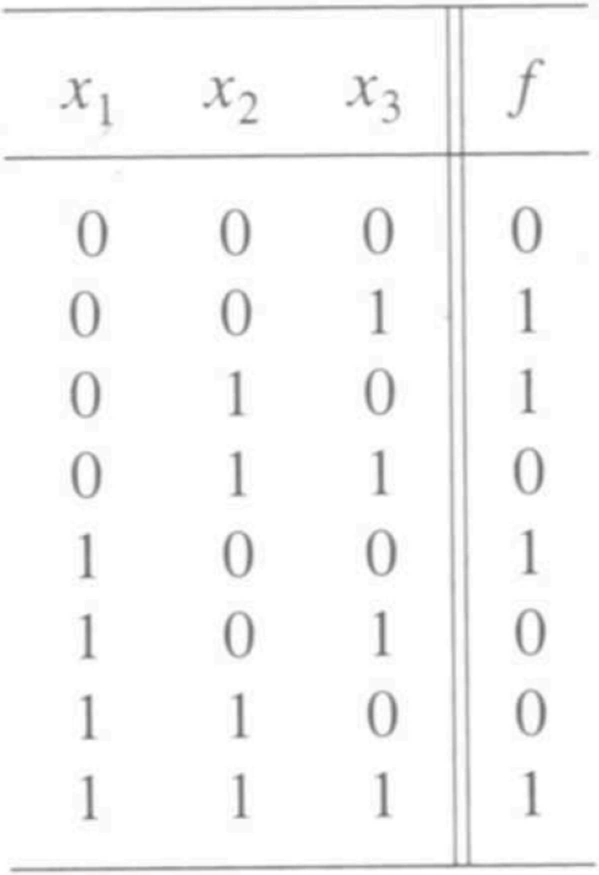
Let’s Derive the POS form



x_1	x_2	x_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 2.31 from the textbook]

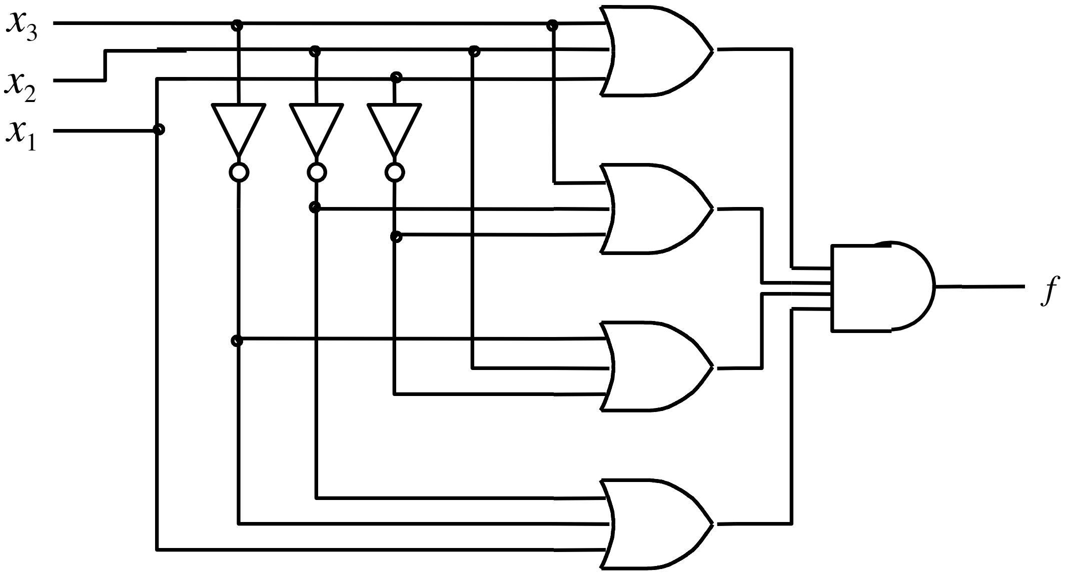
Let’s Derive the POS form



x_1	x_2	x_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

$$\begin{aligned} f &= M_0 \cdot M_3 \cdot M_5 \cdot M_6 \\ &= (x_1 + x_2 + x_3)(x_1 + \overline{x}_2 + \overline{x}_3)(\overline{x}_1 + x_2 + \overline{x}_3)(\overline{x}_1 + \overline{x}_2 + x_3) \end{aligned}$$

Product-of-sums realization



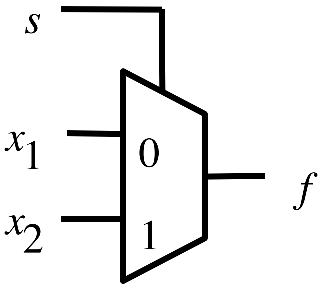
[Figure 2.32b from the textbook]

Multiplexers

2-1 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 2-1 Multiplexer

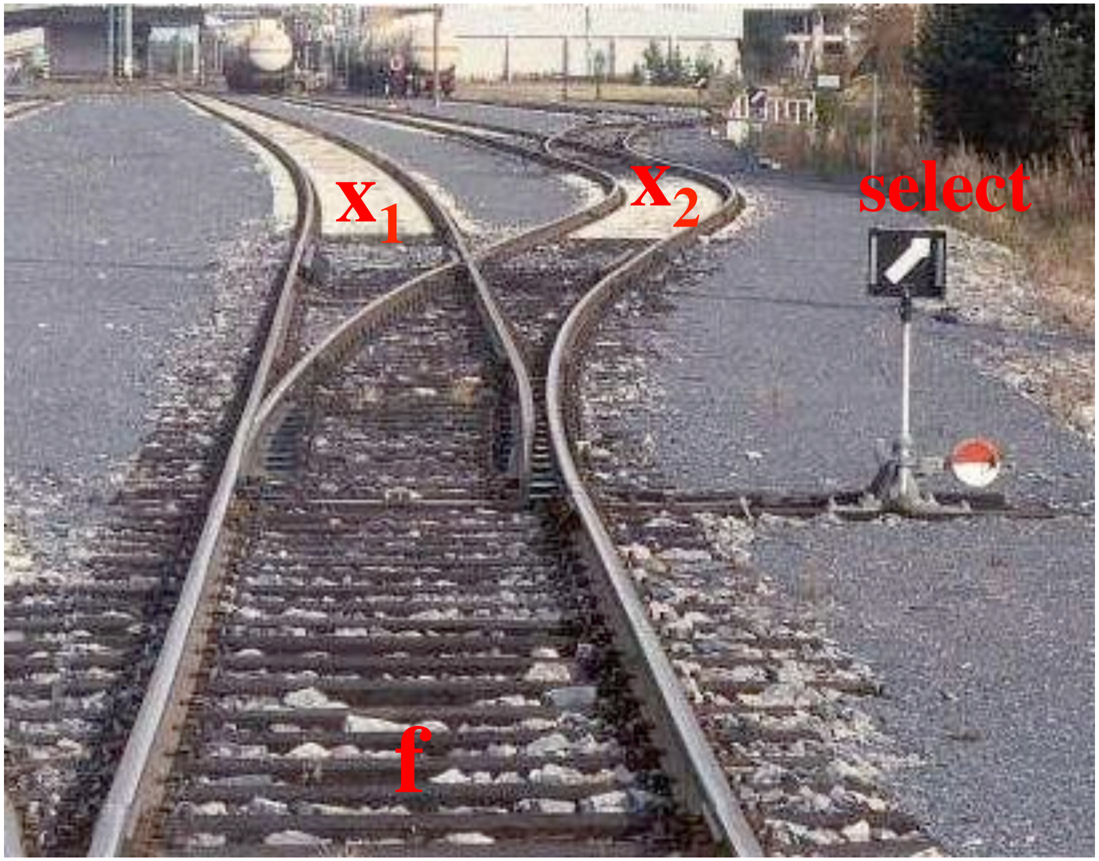


[Figure 2.33c from the textbook]

Analogy: Railroad Switch

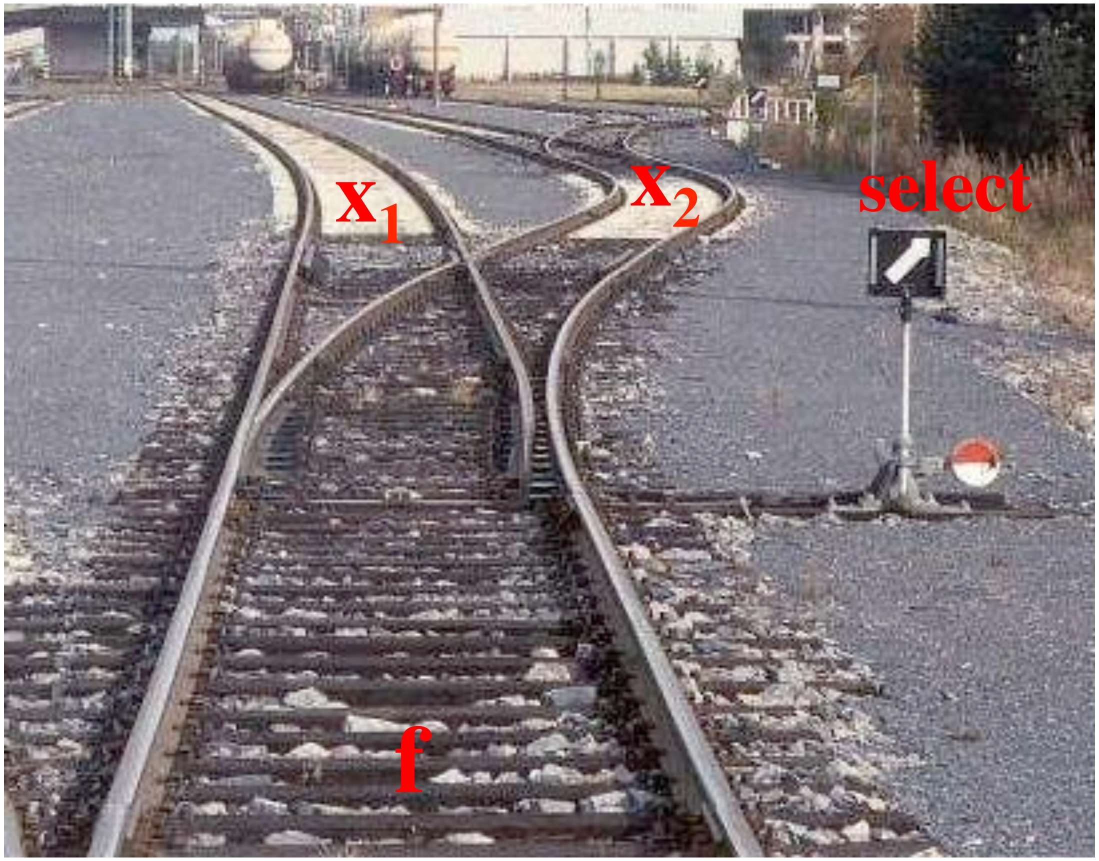
http://en.wikipedia.org/wiki/Railroad_switch]

Analogy: Railroad Switch



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

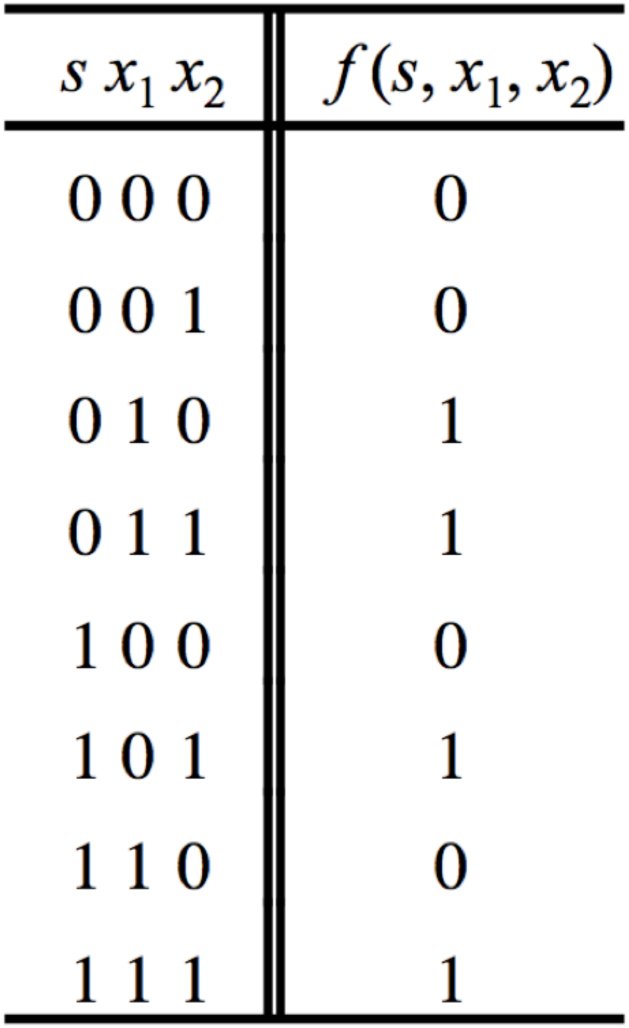
Analogy: Railroad Switch



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]

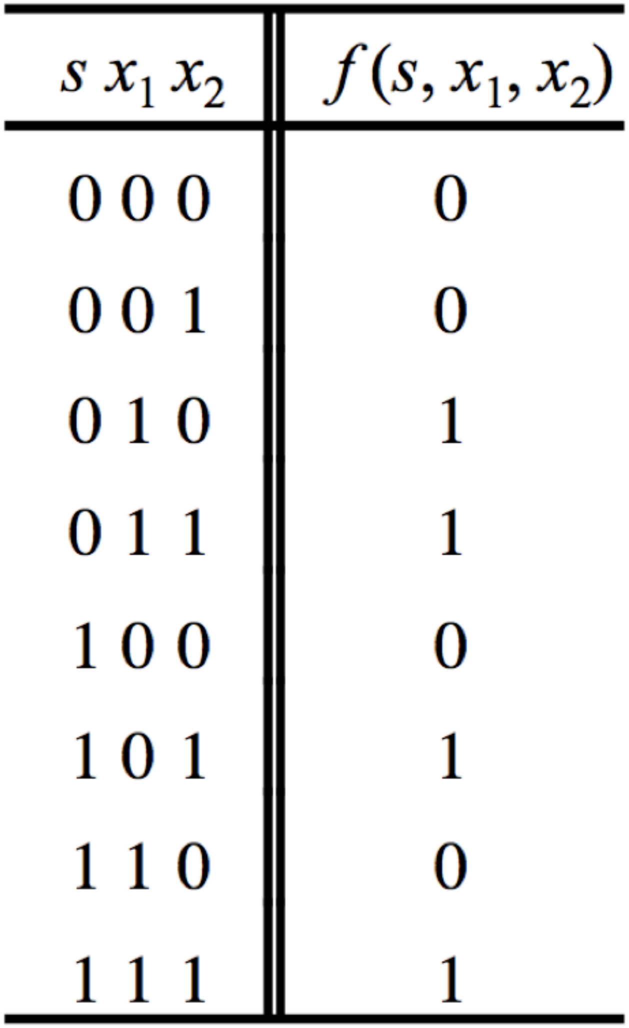
Truth Table for a 2-1 Multiplexer



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

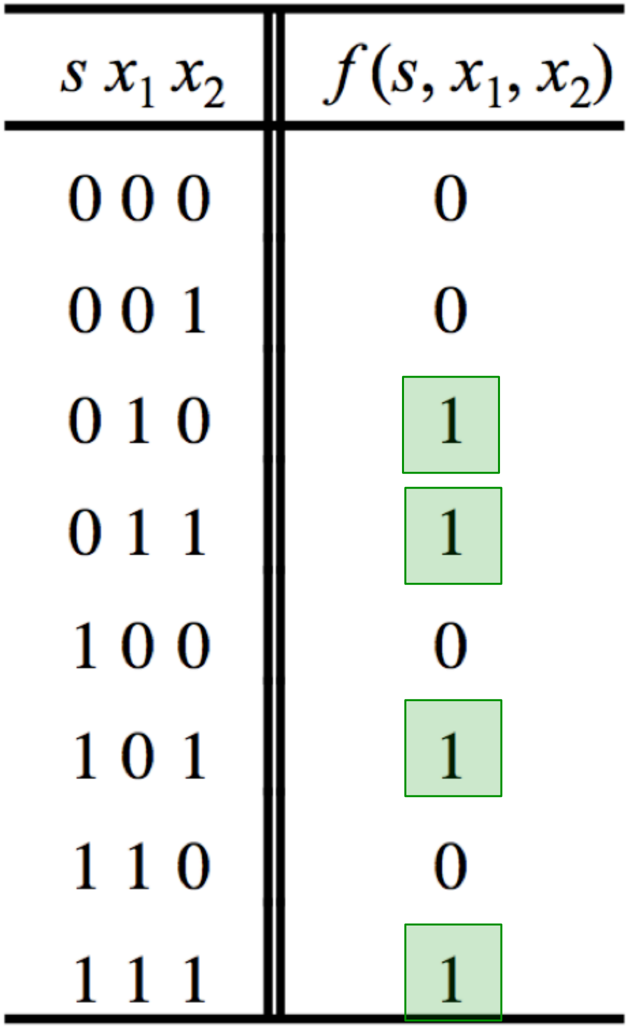
[Figure 2.33a from the textbook]

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	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	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	1	$\overline{s}\,x_1\,\overline{x}_2$
0 1 1	1	$\overline{s}\,x_1\,x_2$
1 0 0	0	
1 0 1	1	$s\,\overline{x}_1\,x_2$
1 1 0	0	
1 1 1	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	1	$\overline{s}\, x_1\, \overline{x}_2$
0 1 1	1	$\overline{s}\, x_1\, x_2$
1 0 0	0	
1 0 1	1	$s\, \overline{x}_1\, x_2$
1 1 0	0	
1 1 1	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

$$f\,(s, x_{1,}\, x_2) = \overline{s}\, x_1\, \overline{x_2} + \overline{s}\, x_1\, x_2 + s\, \overline{x_1}\, x_2 + s\, x_1\, x_2$$

$$f\,(s, x_{1,}\, x_2) = \overline{s}\, x_1\, (\overline{x_2} + x_2) + s\, (\overline{x_1} + x_1\,) x_2$$

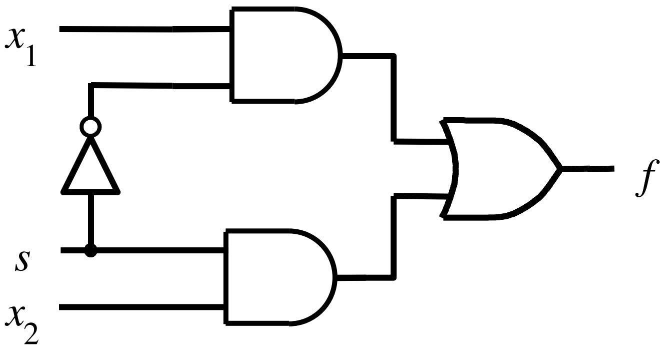
Let’s simplify this expression

$$f\,(s, x_{1,}\, x_2) = \overline{s}\, x_1\, \overline{x_2} + \overline{s}\, x_1\, x_2 + s\, \overline{x_1}\, x_2 + s\, x_1\, x_2$$

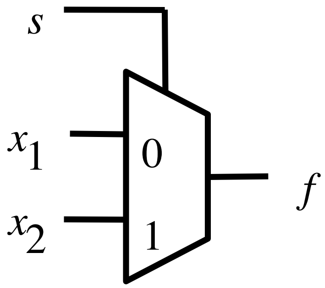
$$f\,(s, x_{1,}\, x_2) = \overline{s}\, x_1\, (\overline{x_2} + x_2) + s\, (\overline{x_1} + x_1\,)x_2$$

$$f\,(s, x_{1,}\, x_2) = \overline{s}\, x_1 + s\, x_2$$

Circuit for 2-1 Multiplexer



(b) Circuit



(c) Graphical symbol

[Figure 2.33b-c from the textbook]

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]

4-1 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

4-1 Multiplexer (Definition)

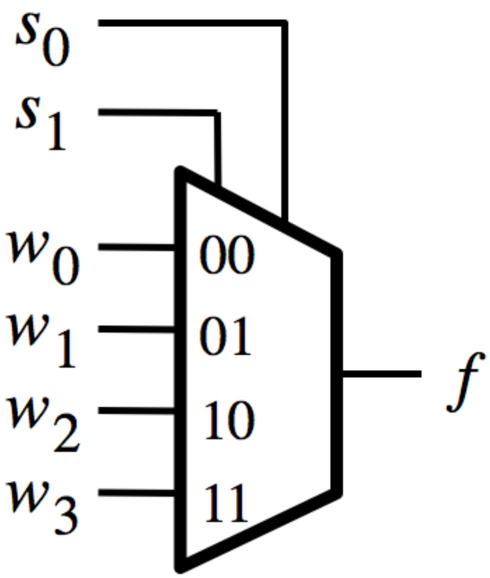
- Has four inputs: w_0 , w_1, w_2, w_3

- 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

We'll talk more about this when we get
to chapter 4, but here is a quick preview.

Graphical Symbol and Truth Table



(a) Graphic symbol

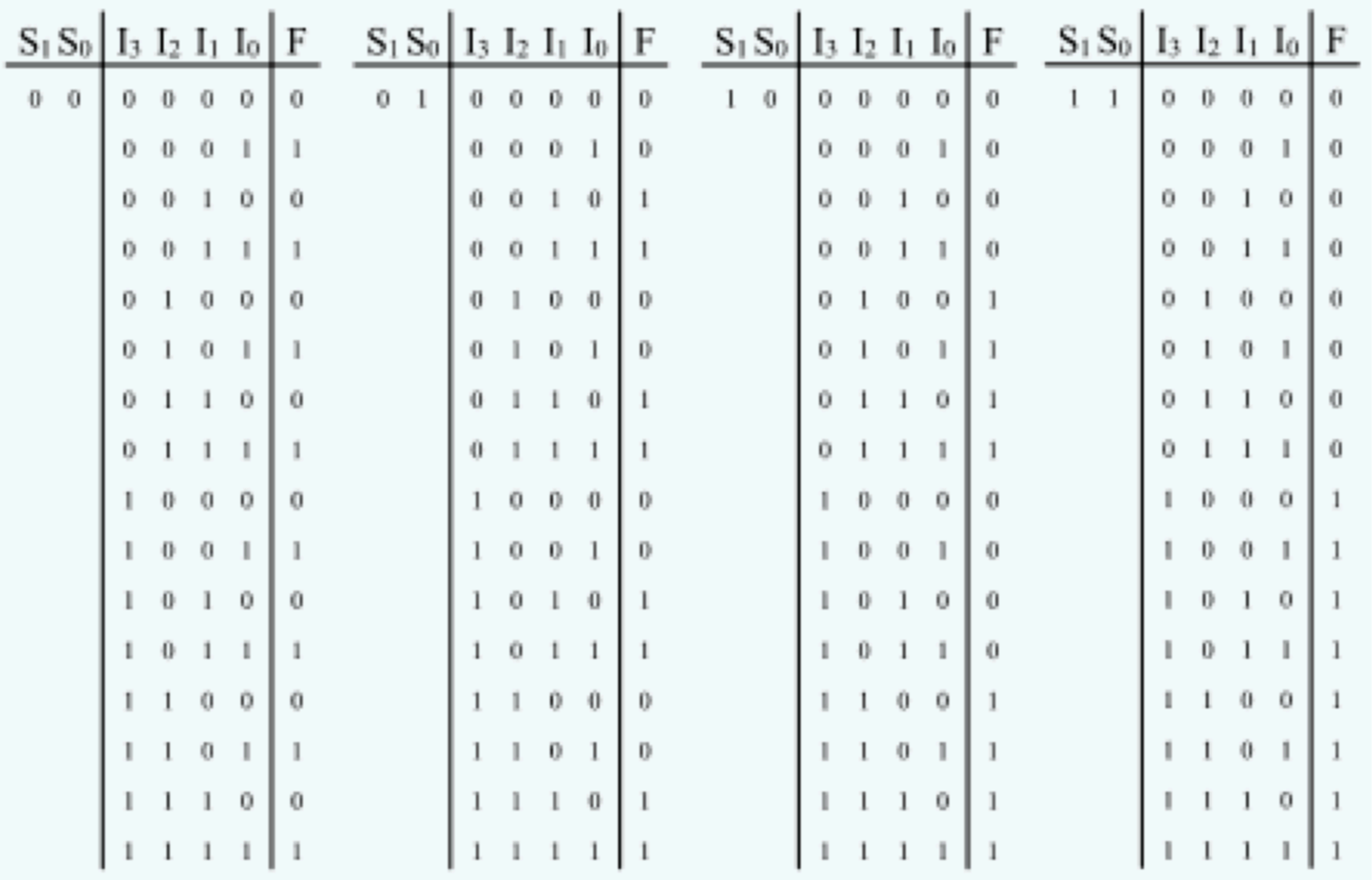
s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

(b) Truth table

[Figure 4.2a-b from the textbook]

The long-form truth table

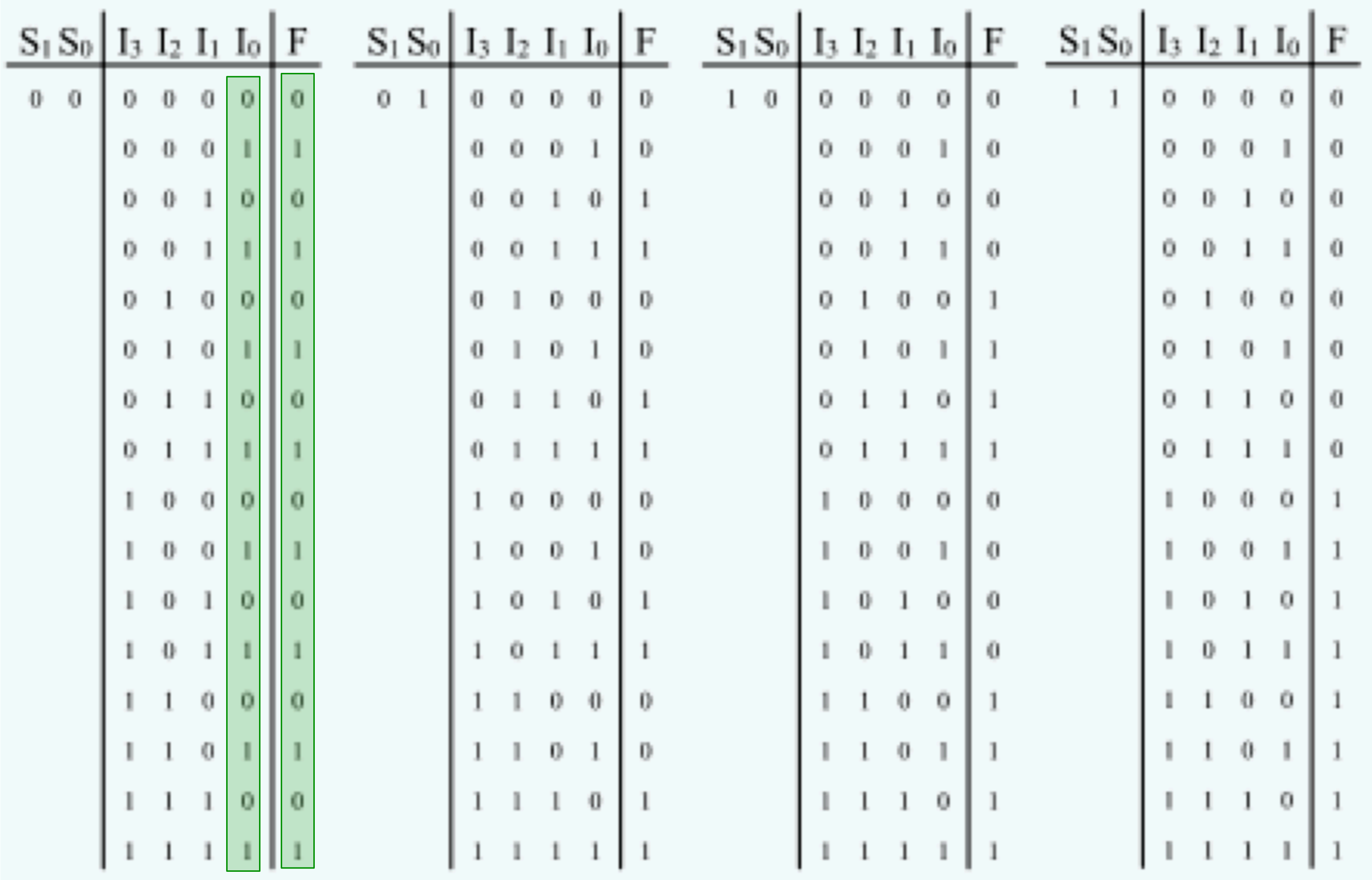
The long-form truth table



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

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

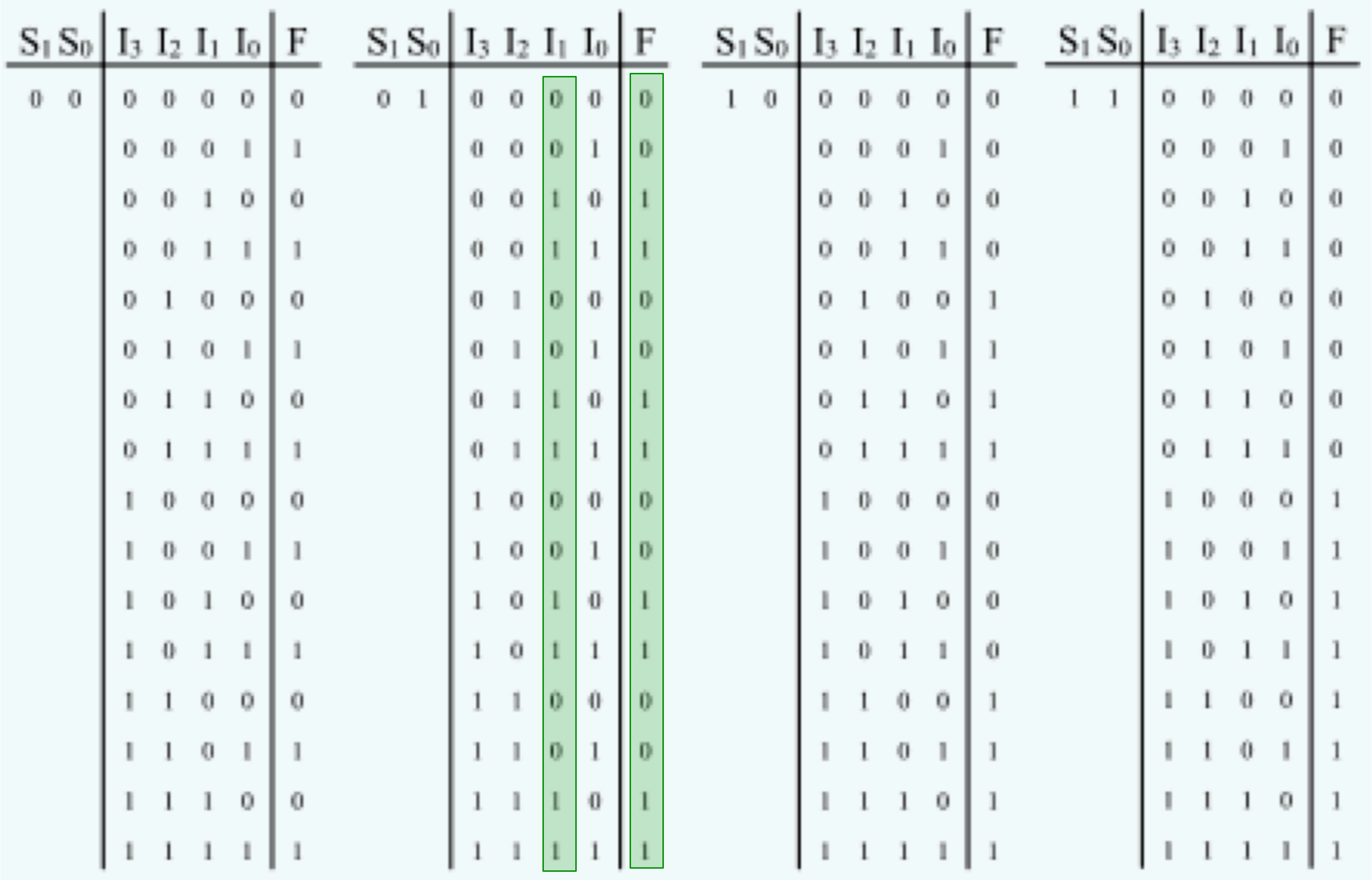
The long-form truth table



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

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

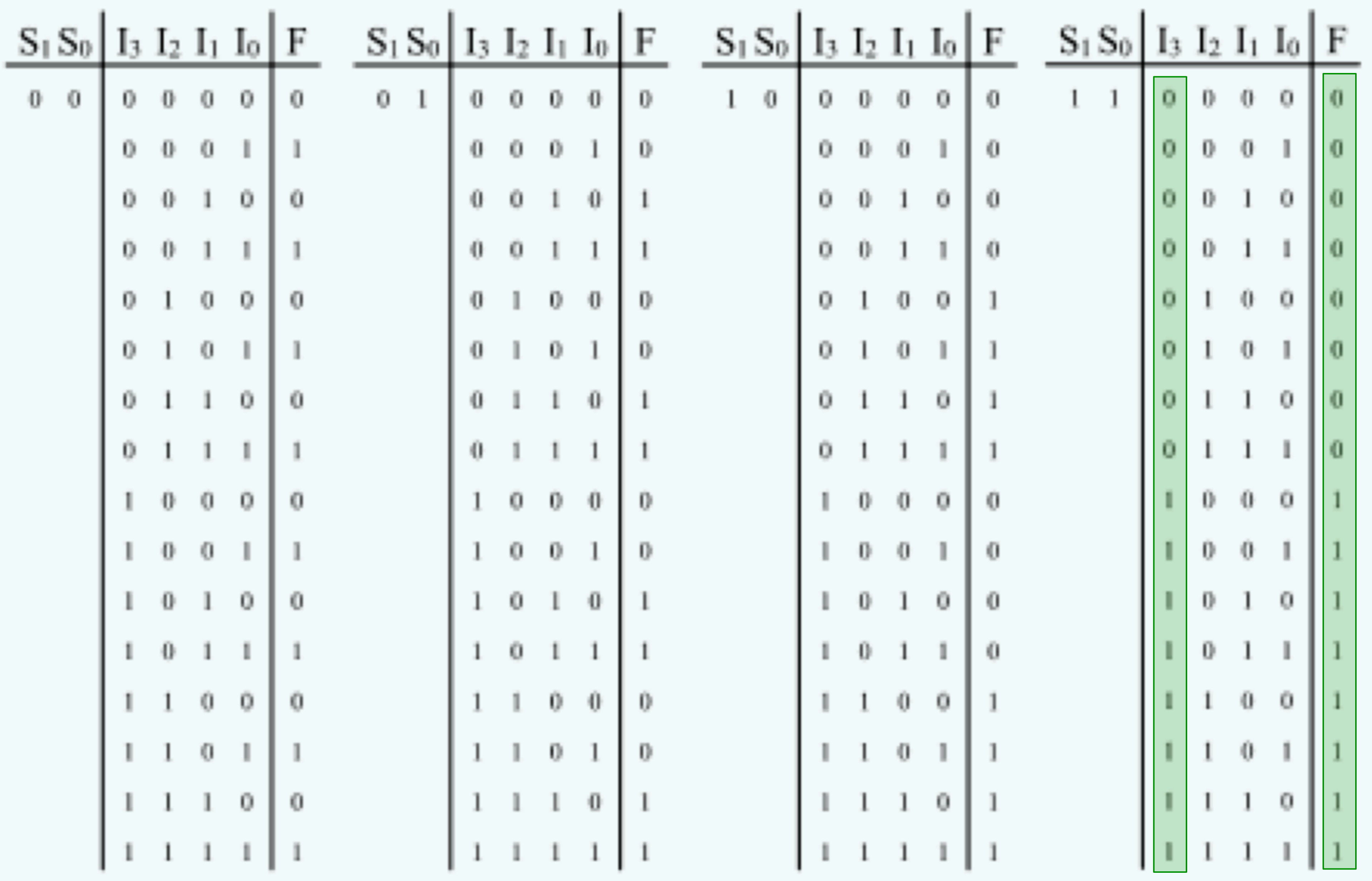
The long-form truth table



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

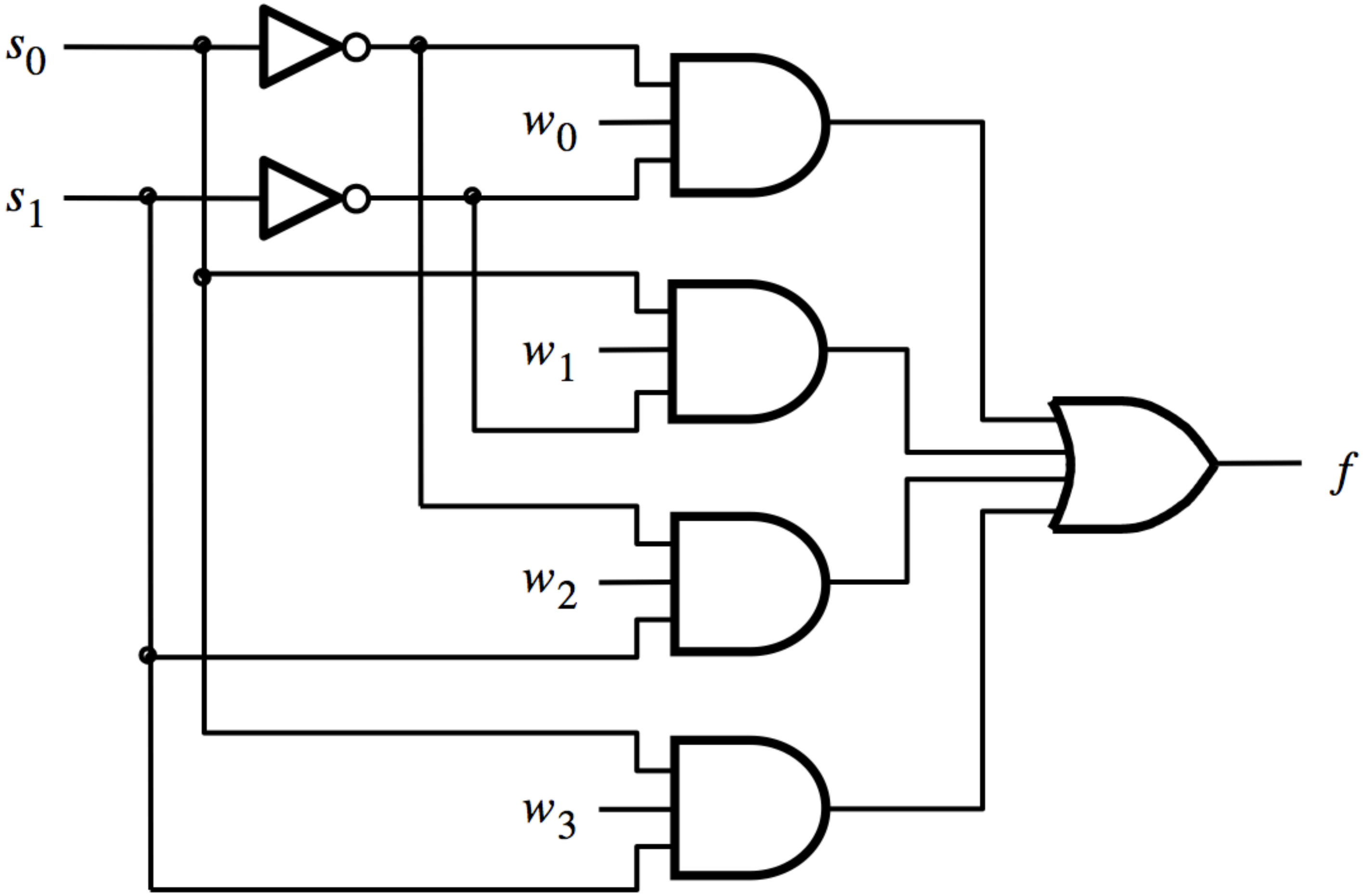
S_1	S_0	I_3	I_2	I_1	I_0	F
0	1	0	0	0	0	0
		0	0	0	1	0
		0	0	1	0	1
		0	0	1	1	1
		0	1	0	0	0
		0	1	0	1	0
		0	1	1	0	1
		0	1	1	1	1
		1	0	0	0	0
		1	0	0	1	0
		1	0	1	0	1
		1	0	1	1	1
		1	1	0	0	0
		1	1	0	1	0
		1	1	1	0	1
		1	1	1	1	1

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

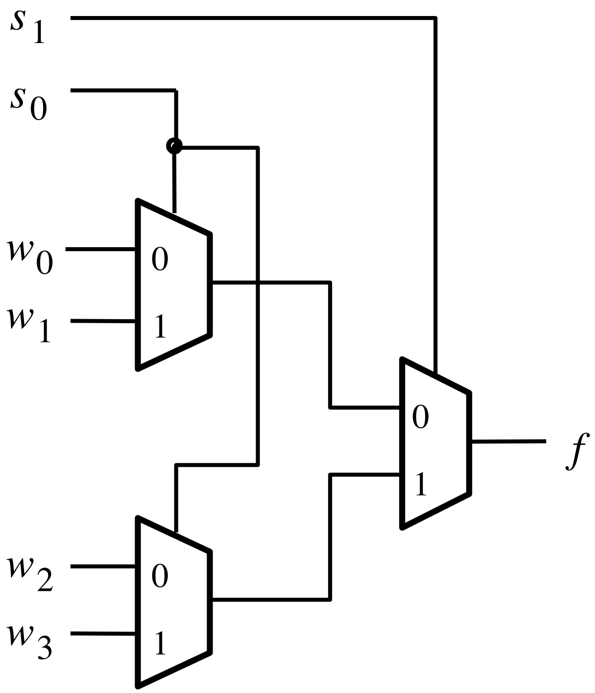
[http://www.absoluteastronomy.com/topics/Multiplexer]

4-1 Multiplexer (SOP circuit)



[Figure 4.2c from the textbook]

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

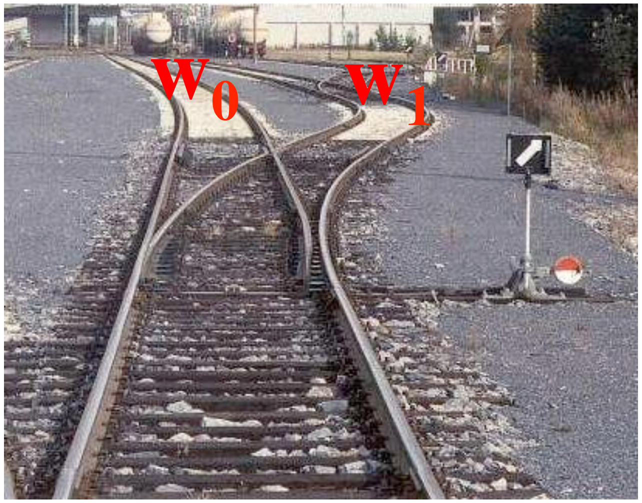


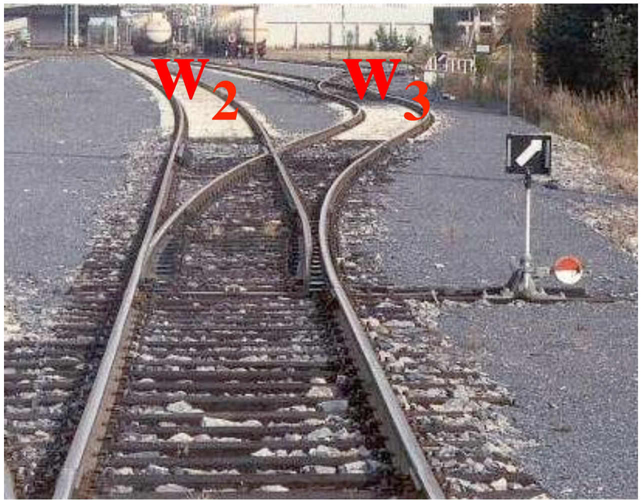
[Figure 4.3 from the textbook]

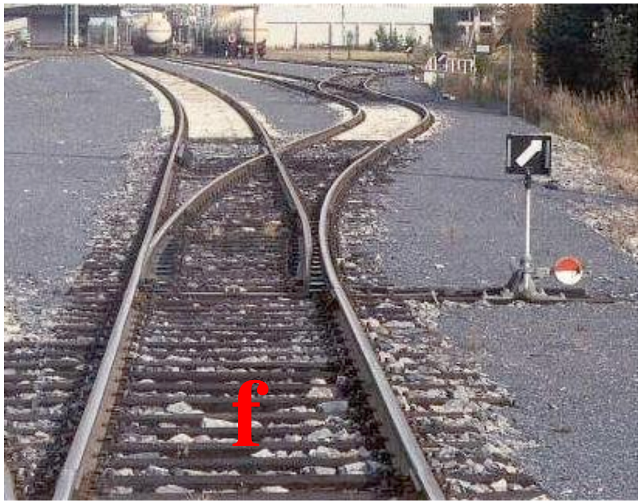
Analogy: Railroad Switches

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

Analogy: Railroad Switches



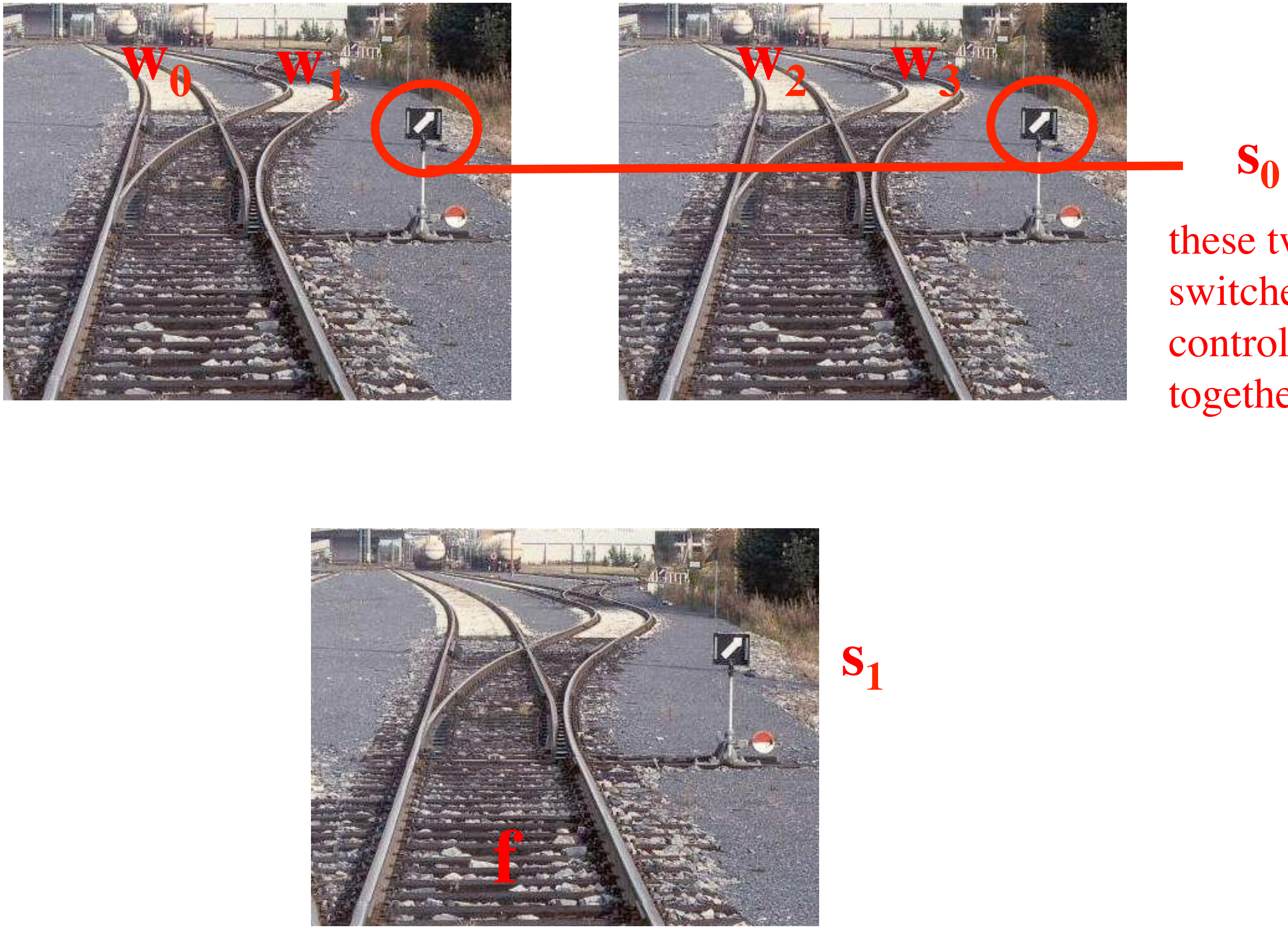




s_1

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

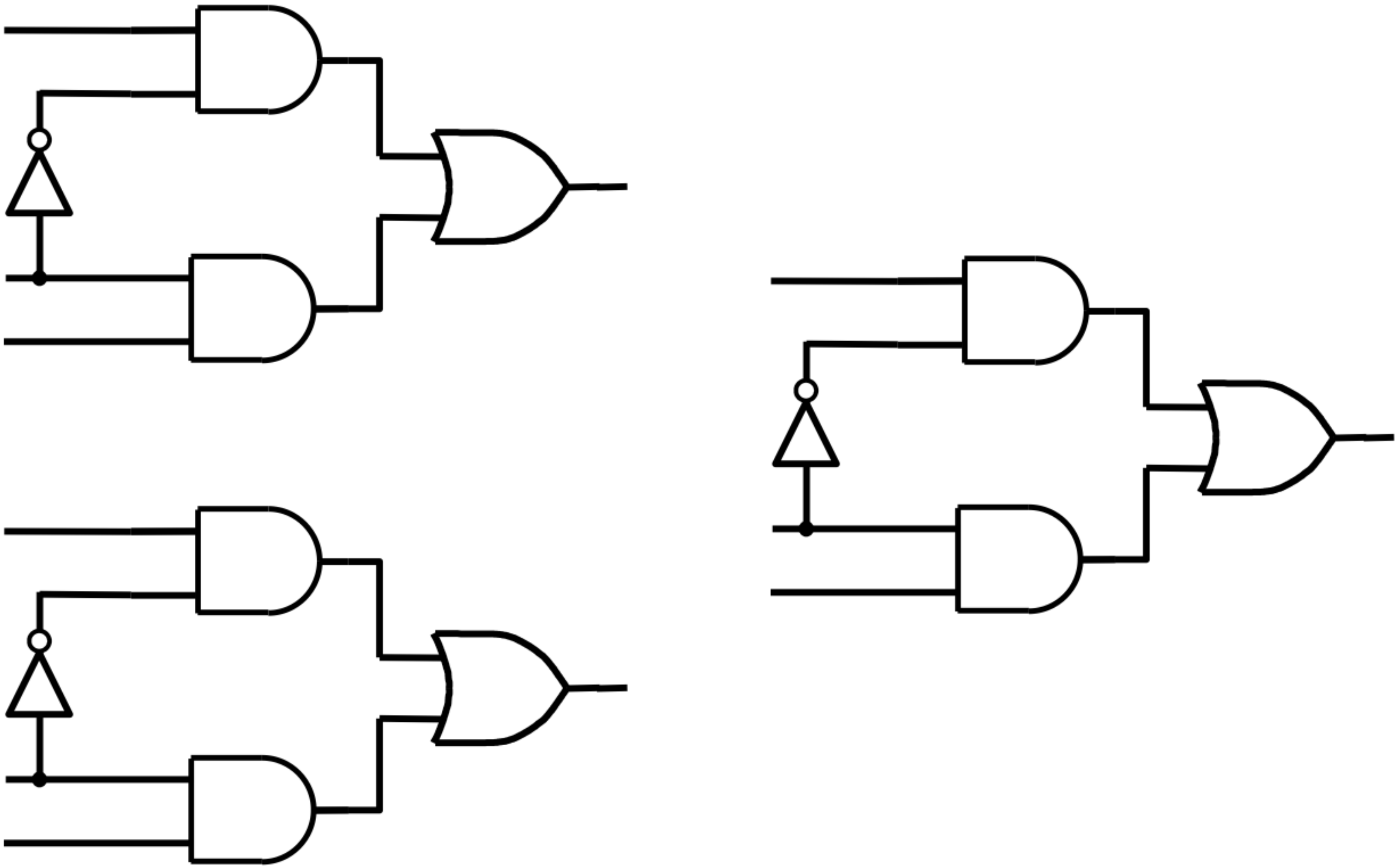
Analogy: Railroad Switches



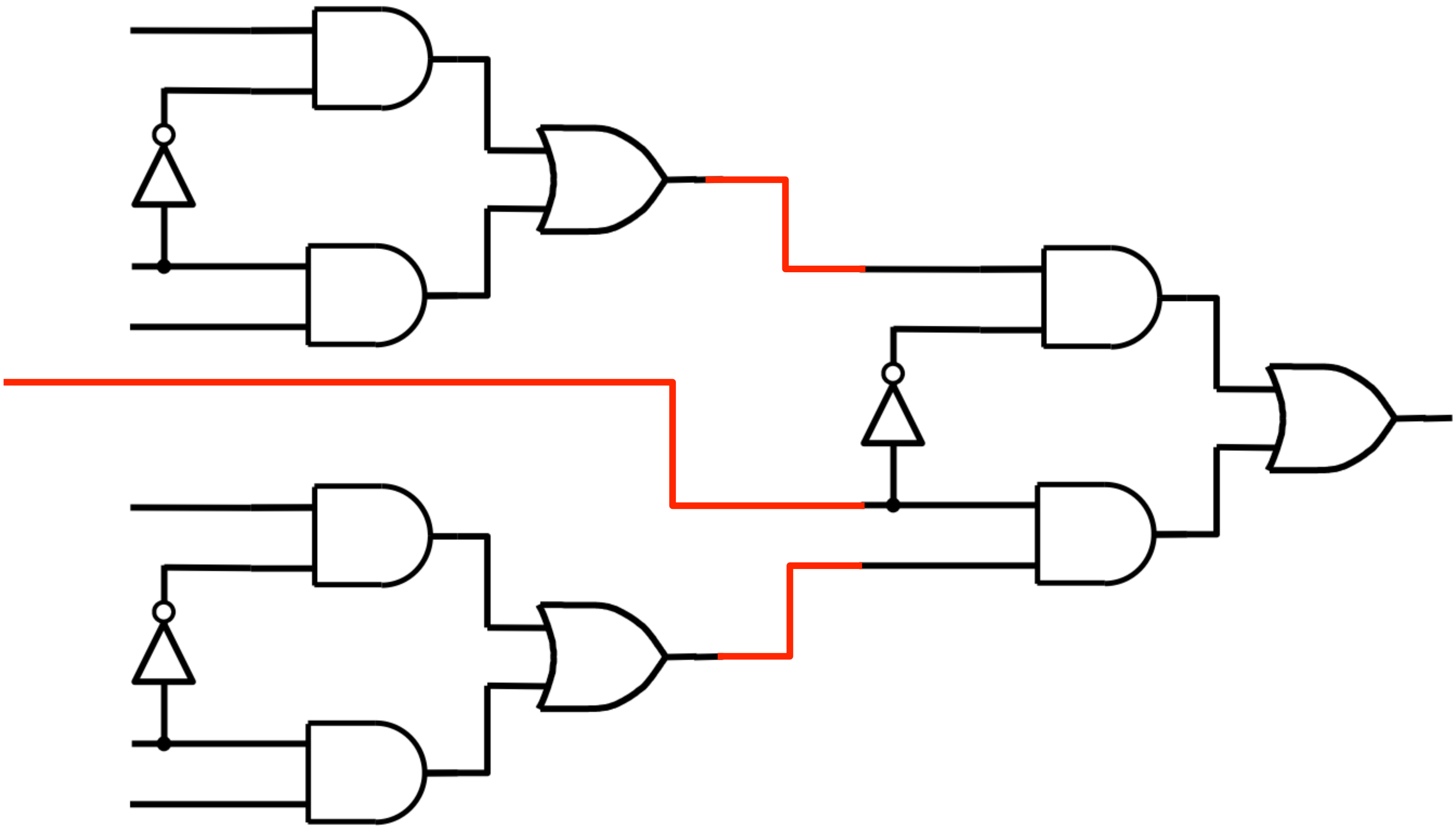
these two switches are controlled together

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

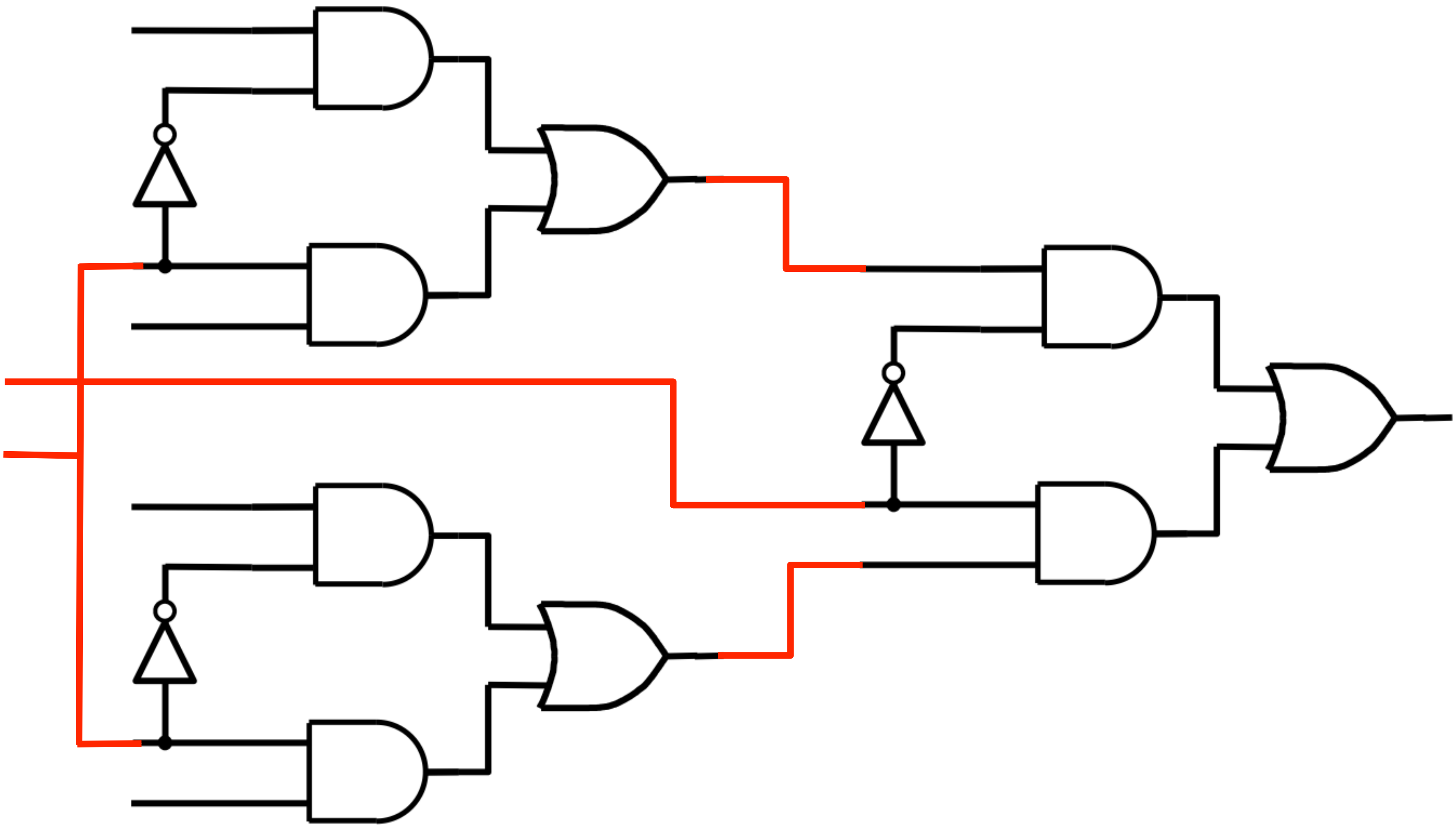
Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer



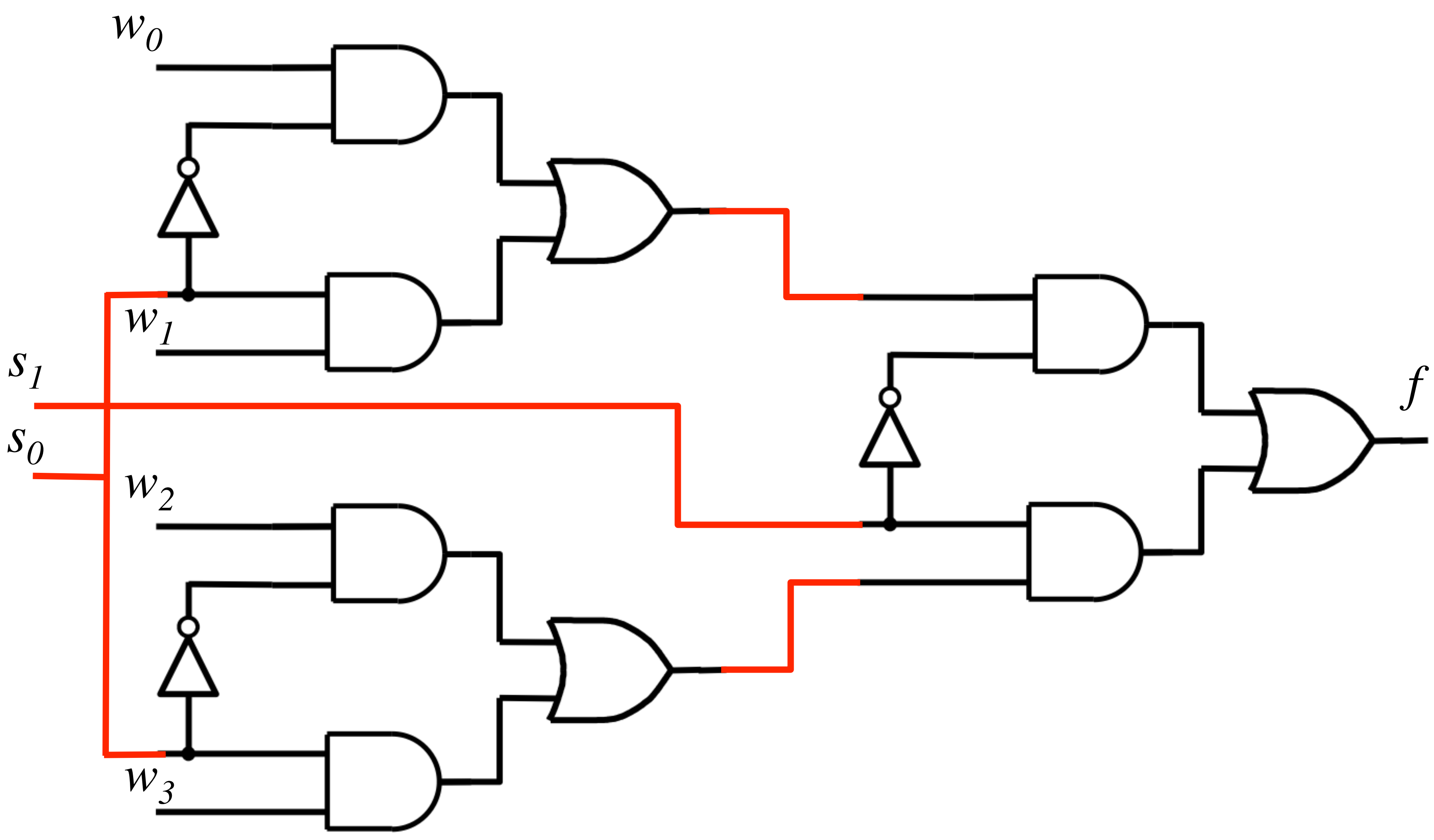
Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer



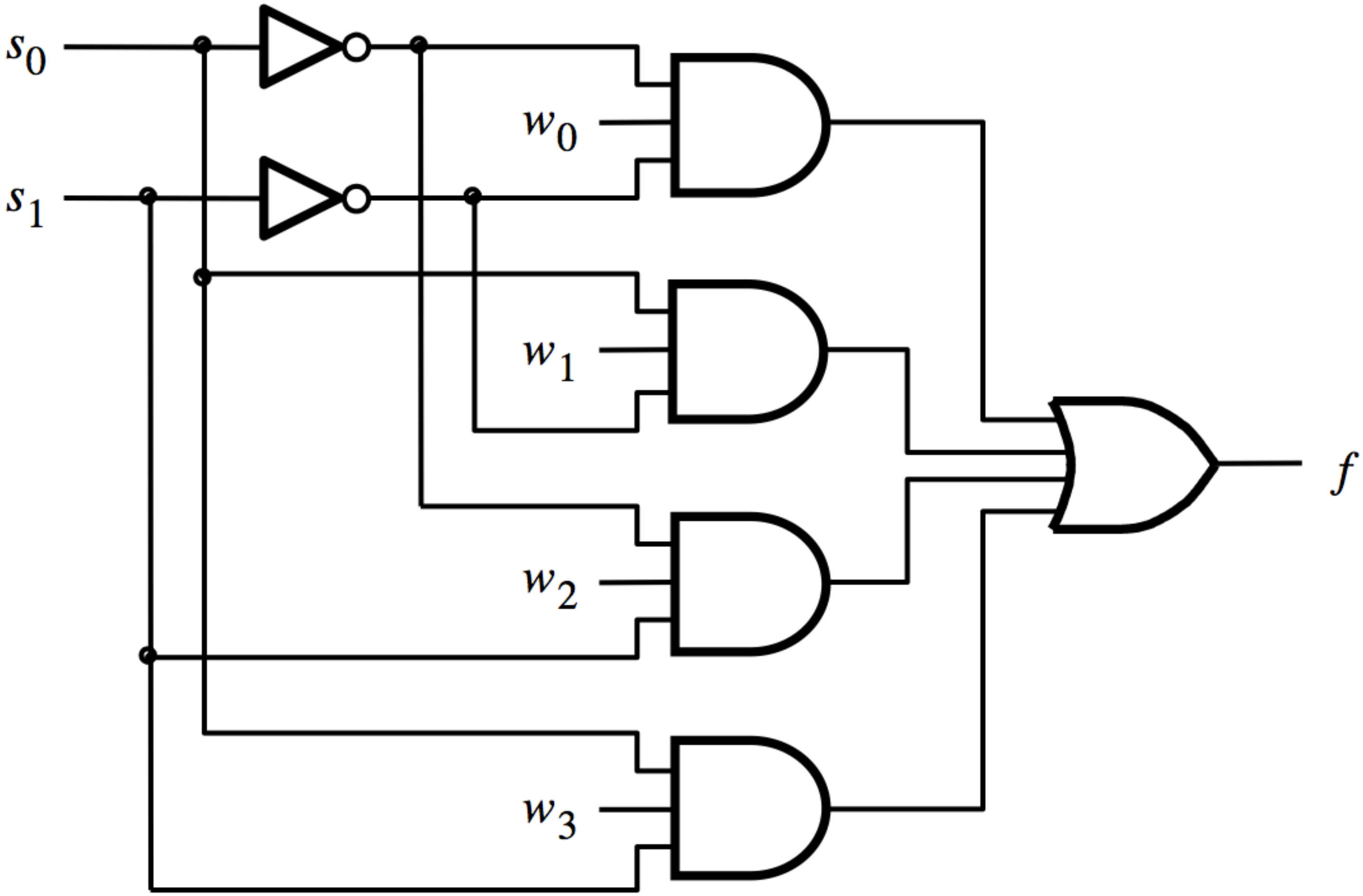
Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer



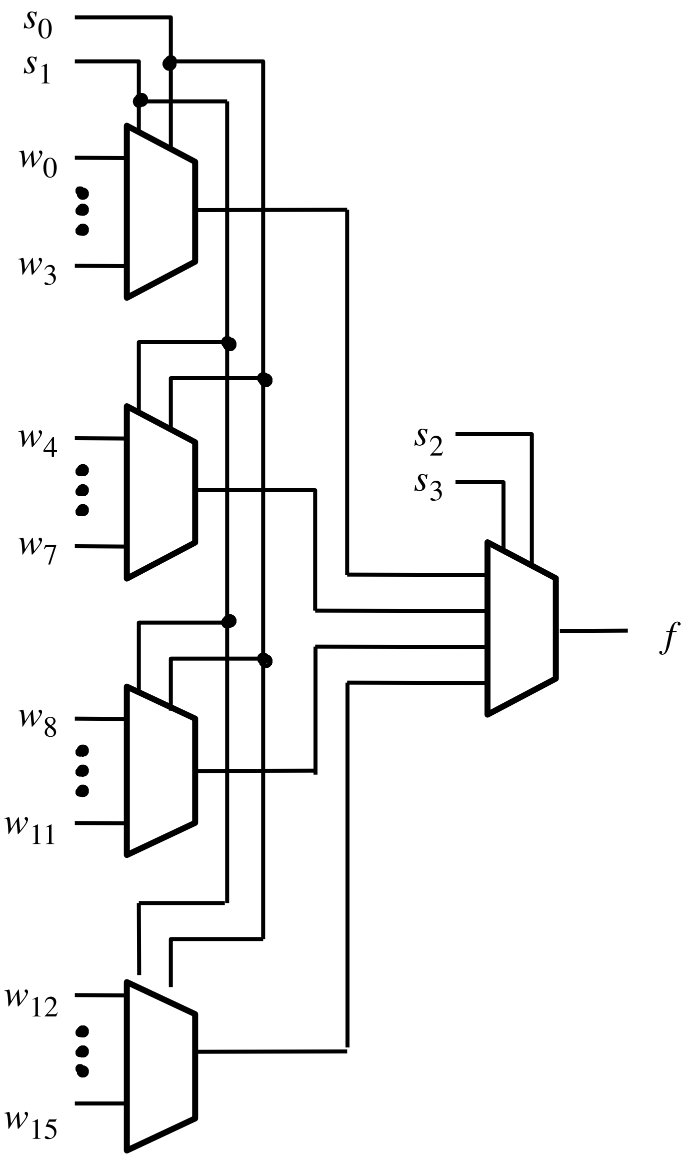
Using three 2-to-1 multiplexers to build one 4-to-1 multiplexer



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



16-1 Multiplexer



[Figure 4.4 from the textbook]

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

Questions?

THE END