3.7 Other Logic Operations

Boolean expressions for the 16 functions of two variables

Name	Operator Symbol	Function Values for x, y =				Algebraic Expression	Comment	Standard Library Gates
		00	01	10	11			
Zero		0	0	0	0	$F_0 = 0$	binary constant 0	
AND	$x \bullet y$	0	0	0	1	$F_1 = x\,y$	x and y	✓
Inhibition	x / y	0	0	1	0	$F_2 = x\,y'$	x but not y	
Transfer		0	0	1	1	$F_3 = x$	x	✓
Inhibition	y / x	0	1	0	0	$F_4 = x'\,y$	y but not x	
Transfer		0	1	0	1	$F_5 = y$	y	✓
XOR	$x \oplus y$	0	1	1	0	$F_6 = xy' + x'y$	x or y but not both	✓
OR	$x + y$	0	1	1	1	$F_7 = x + y$	x or y	✓
NOR	$x \downarrow y$	1	0	0	0	$F_8 = (x + y)'$	not OR	✓
Equivalence **XNOR**	$x \odot y$	1	0	0	1	$F_9 = xy + x'\,y'$	x equals y	✓
Complement **NOT**	y'	1	0	1	0	$F_{10} = y'$	not y	✓
Implication	$x \subset y$	1	0	1	1	$F_{11} = x + y'$	if y, then x	
Complement **NOT**	x'	1	1	0	0	$F_{12} = x'$	not x	✓
Implication	$x \supset y$	1	1	0	1	$F_{13} = x' + y$	if x, then y	
NAND	$x \uparrow y$	1	1	1	0	$F_{14} = (x\,y)'$	not AND	✓
One		1	1	1	1	$F_{15} = 1$	binary constant 1	

3.8 Digital Logic Gates

Gate library - the collection of logic gates that we use in constructing logic networks.
Standard gates - gates in the gate library.

Name	Graphic Symbol	Functional Expression	Number of Transistors	Gate Delay (ns) (with respect to the Inverter)
Inverter		$F = x'$	2	1
Driver		$F = x$	4	2
AND		$F = xy$	6	2.4
OR		$F = x + y$	6	2.4
NAND		$F = (xy)'$	4	1.4
NOR		$F = (x + y)'$	4	1.4
XOR		$F = x \oplus y$	14	4.2
XNOR		$F = x \odot y$	12	3.2

To show that XOR' = XNOR

$$
\begin{aligned}
(x \oplus y)' &= (x\,y' + x'\,y)' \\
&= (x\,y')'\,(x'\,y)' \\
&= (x' + y)\,(x + y') \\
&= x'\,x + x'\,y' + y\,x + y\,y' \\
&= x \odot y
\end{aligned}
$$

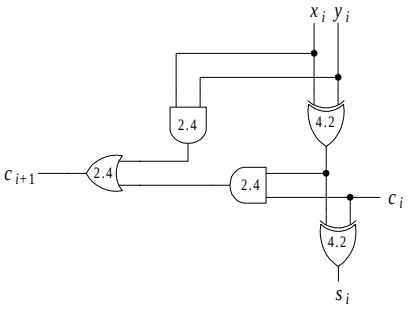
Ex. Full-adder design

$x_i + y_i + c_i$			c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Solution 1: First, we derive an expression that contains a minimum number of operators. The expression for s_i can be reduced to two XOR gates:

$$\begin{aligned} s_i &= x_i' y_i' c_i + x_i' y_i c_i' + x_i y_i' c_i' + x_i y_i c_i \\ &= (x_i' y_i + x_i y_i') c_i ' + (x_i' y_i' + x_i y_i) c_i \\ &= (x_i \oplus y_i) c_i' + (x_i \odot y_i) c_i \\ &= (x_i \oplus y_i) c_i' + (x_i \oplus y_i)' c_i \\ &= (x_i \oplus y_i) \oplus c_i \end{aligned}$$

$$\begin{aligned} c_{i+1} &= x_i' y_i c_i + x_i y_i' c_i + x_i y_i c_i' + x_i y_i c_i \\ &= x_i y_i (c_i' + c_i) + c_i (x_i' y_i + x_i y_i') \\ &= x_i y_i \bullet 1 + c_i (x_i \oplus y_i) \end{aligned}$$



Input/Output Path	Delay (ns)
c_i to c_{i+1}	4.8
c_i to s_i	4.2
x_i, y_i to c_{i+1}	9.0
x_i, y_i to s_i	8.4

Number of transistors = 46

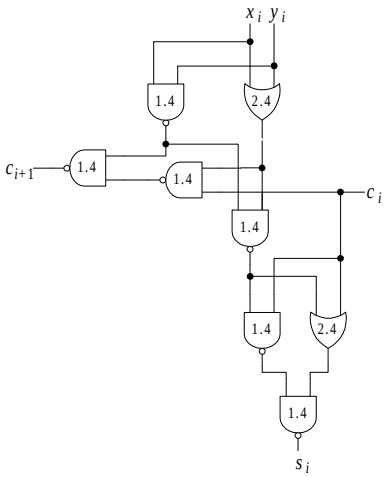
Solution 2: Transform the expressions using De Morgan's theorems.

$$\begin{aligned} s_i &= (x_i \oplus y_i) c_i' + (x_i \odot y_i) c_i \\ &= (x_i \odot y_i)' c_i' + (x_i \odot y_i) c_i \\ &= (x_i \odot y_i) \odot c_i \end{aligned}$$

$$\begin{aligned} c_{i+1} &= x_i' y_i c_i + x_i y_i' c_i + x_i y_i c_i' + x_i y_i c_i \\ &= x_i' y_i c_i + x_i y_i c_i + x_i y_i' c_i + x_i y_i c_i + x_i y_i c_i' + x_i y_i c_i \\ &= x_i y_i + x_i c_i + y_i c_i \\ &= x_i y_i + c_i (x_i + y_i) \\ &= [(x_i y_i)' \bullet (c_i (x_i + y_i))'] ' \end{aligned}$$

In addition, we can also implement $x_i \odot y_i$ with two NANDs and one OR gate:

$$\begin{aligned} x_i \odot y_i &= x_i y_i + x_i' y_i' \\ &= ((x_i y_i)' (x_i' y_i')')' \\ &= ((x_i y_i)' (x_i + y_i))' \end{aligned}$$

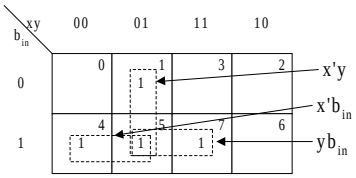


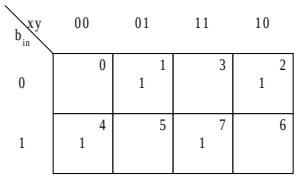
Input/Output Path	Delay (ns)
c_i to c_{i+1}	2.8
c_i to s_i	3.8
x_i, y_i to c_{i+1}	5.2
x_i, y_i to s_i	7.6

Number of transistors = 36

Ex. Full subtractor

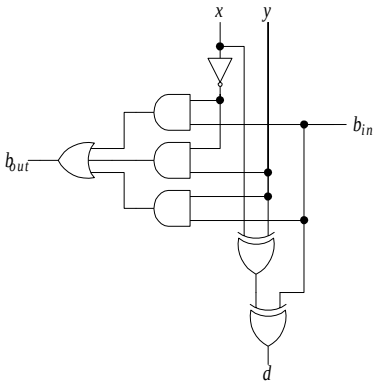
b_{in}	x	y	b_{out}	d	Comment
0	0	0	0	0	x-b_{in}-y = d = 0-0 = 0
0	0	1	1	1	0-0-1 = borrow 2-1 = 1
0	1	0	0	1	1-0-0=1 with no borrow
0	1	1	0	0	1-0-1=0 with no borrow
1	0	0	1	1	x=0-1=-1 because of b_{in}=1, therefore, borrow 2-1=1. Finally, 1-0=1
1	0	1	1	0	x-b_{in}-y=borrow 2-1-1=0
1	1	0	0	0	1-1-0=0 with no borrow
1	1	1	1	1	x-b_{in}=1-1=0. Then 0-1=borrow 2-1=1

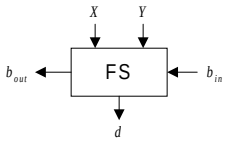




$$b_{out} = x'b_{in} + x'y + yb_{in}$$

$$\begin{aligned} d &= x'y'b_{in} + xyb_{in} + x'yb'_{in} + xy'b'_{in} \\ &= b_{in}(x'y' + xy) + b'_{in}(x'y + xy') \\ &= b_{in}(x \oplus y)' + b'_{in}(x \oplus y) \\ &= b_{in} \oplus x \oplus y \end{aligned}$$





Circuit for Full Subtractor

3.9a Multiple Inputs

The AND and OR operations are both commutative and associative, so they can easily be extended to have multiple inputs.

The NAND and NOR operations are commutative but not associative, but we can think of multiple-input NAND and NOR operations as the complements of the AND and OR.

NAND($x, y, z, \ldots$) = ($x\ y\ z\ \ldots$)' $\neq x \uparrow y \uparrow z \uparrow \ldots$

NOR($x, y, z, \ldots$) = ($x + y + z + \ldots$)' $\neq x \downarrow y \downarrow z \downarrow \ldots$

The XOR and XNOR operations are both commutative and associative, so they can be extended to have multiple inputs.

$x \oplus y \oplus z = (x\ y' + x'\ y) \oplus z$

$= (x\ y' + x'\ y)\ z' + (x\ y' + x'\ y)'\ z$

$= (x\ y' + x'\ y)\ z' + ((x\ y')' \bullet (x'\ y)')\ z$

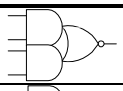
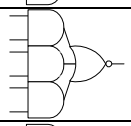
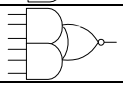
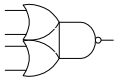
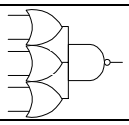
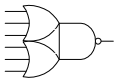
$= (x\ y' + x'\ y)\ z' + (x' + y) \bullet (x + y')\ z$

$= x\ y'\ z' + x'\ y\ z' + \sout{x'\ x\ z} + x'\ y'\ z + y\ x\ z + \sout{y\ y'\ z}$

$= x\ y'\ z' + x'\ y\ z' + x'\ y'\ z + y\ x\ z$

Name	Graphic Symbol	Functional Expression	Number of Transistors	Gate Delay (ns) (with respect to the Inverter)
3-input AND		$F = xyz$	8	2.8
4-input AND		$F = wxyz$	10	3.2
3-input OR		$F = x + y + z$	8	2.8
4-input OR		$F = w + x + y + z$	10	3.2
3-input NAND		$F = (xyz)'$	6	1.8
4-input NAND		$F = (wxyz)'$	8	2.2
3-input NOR		$F = (x + y + z)'$	6	1.8
4-input NOR		$F = (w + x + y + z)'$	8	2.2

3.9b Multiple Operators

Name	Graphic Symbol	Functional Expression	Number of Transistors	Gate Delay (ns) (with respect to the Inverter)
2-wide, 2-input AOI		$F = (wx + yz)'$	8	2.0
3-wide, 2-input AOI		$F = (uv + wx + yz)'$	12	2.4
2-wide, 3-input AOI		$F = (uvw + xyz)'$	12	2.2
2-wide, 2-input OAI		$F = ((w+x)(y+z))'$	8	2.0
3-wide, 2-input OAI		$F = ((u+v)\ (w+x)(y+z))'$	12	2.2
2-wide, 3-input OAI		$F = ((u+v+w)(\ x+y+z))'$	12	2.4