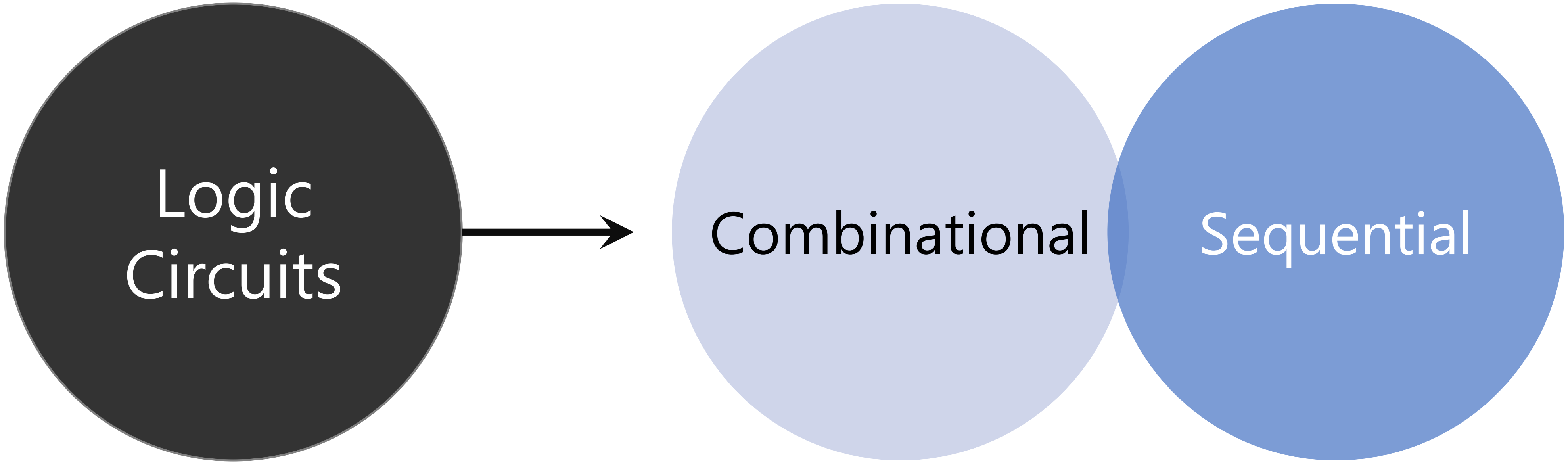


Logic Circuits

Combinational

Sequential

Chapter 4 Combinational Logic

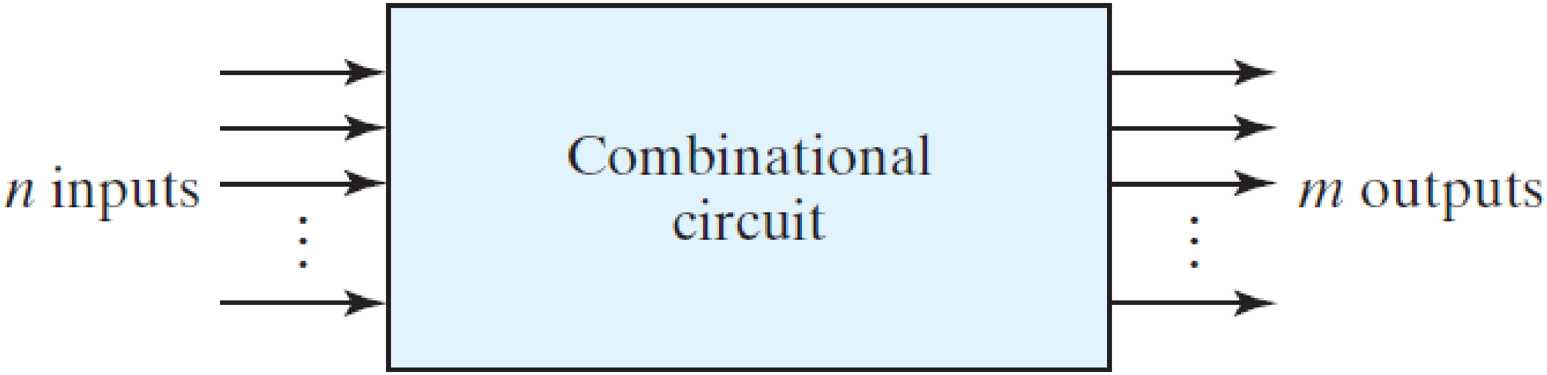


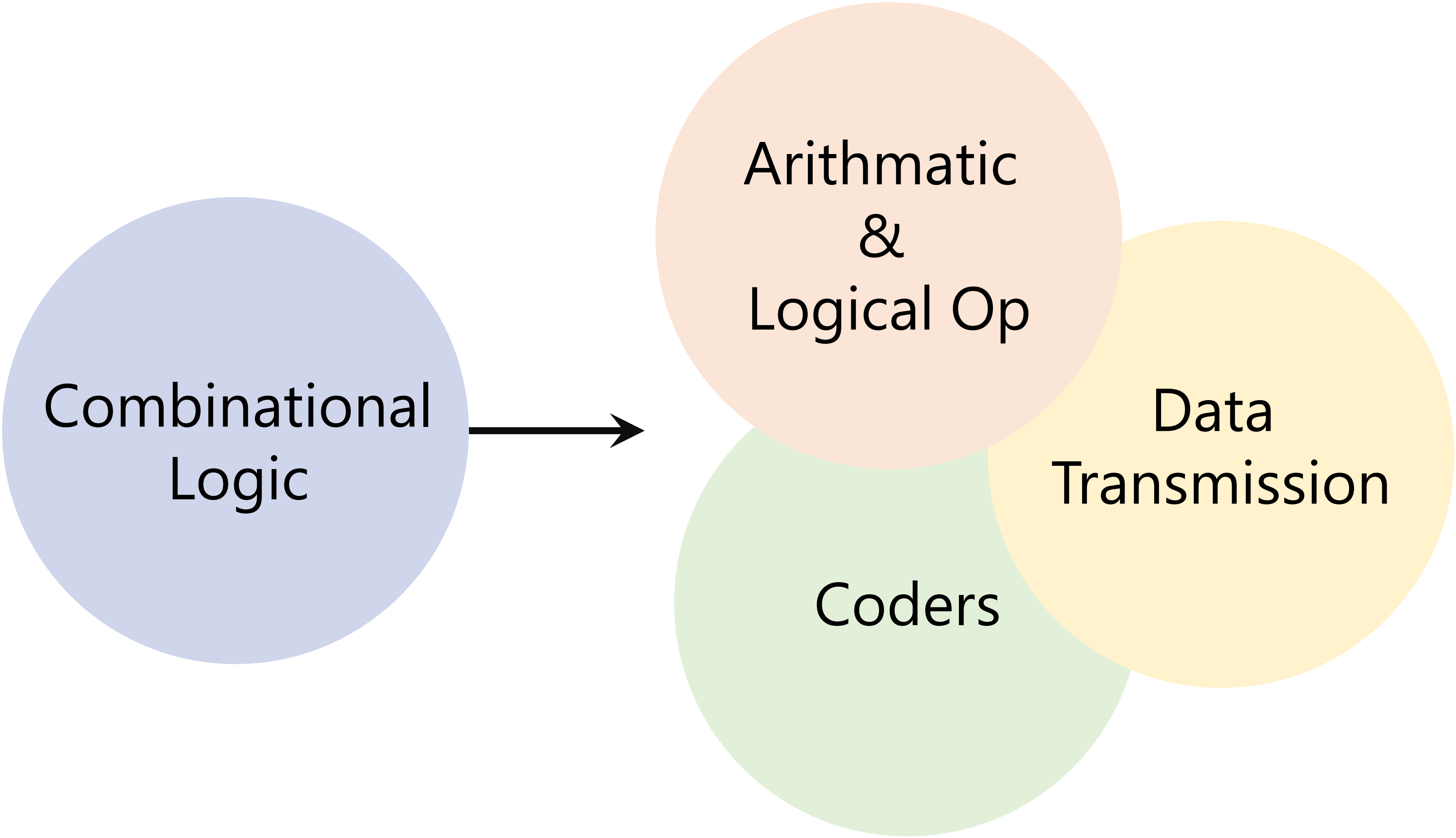
FIGURE 4.1
Block diagram of combinational circuit

Combinational Logic

aka. Combinational Circuit

Combination of logic gates on the present inputs → the outputs *at any time*!

A combinational circuit performs an operation that can be specified logically by a set of Boolean functions.

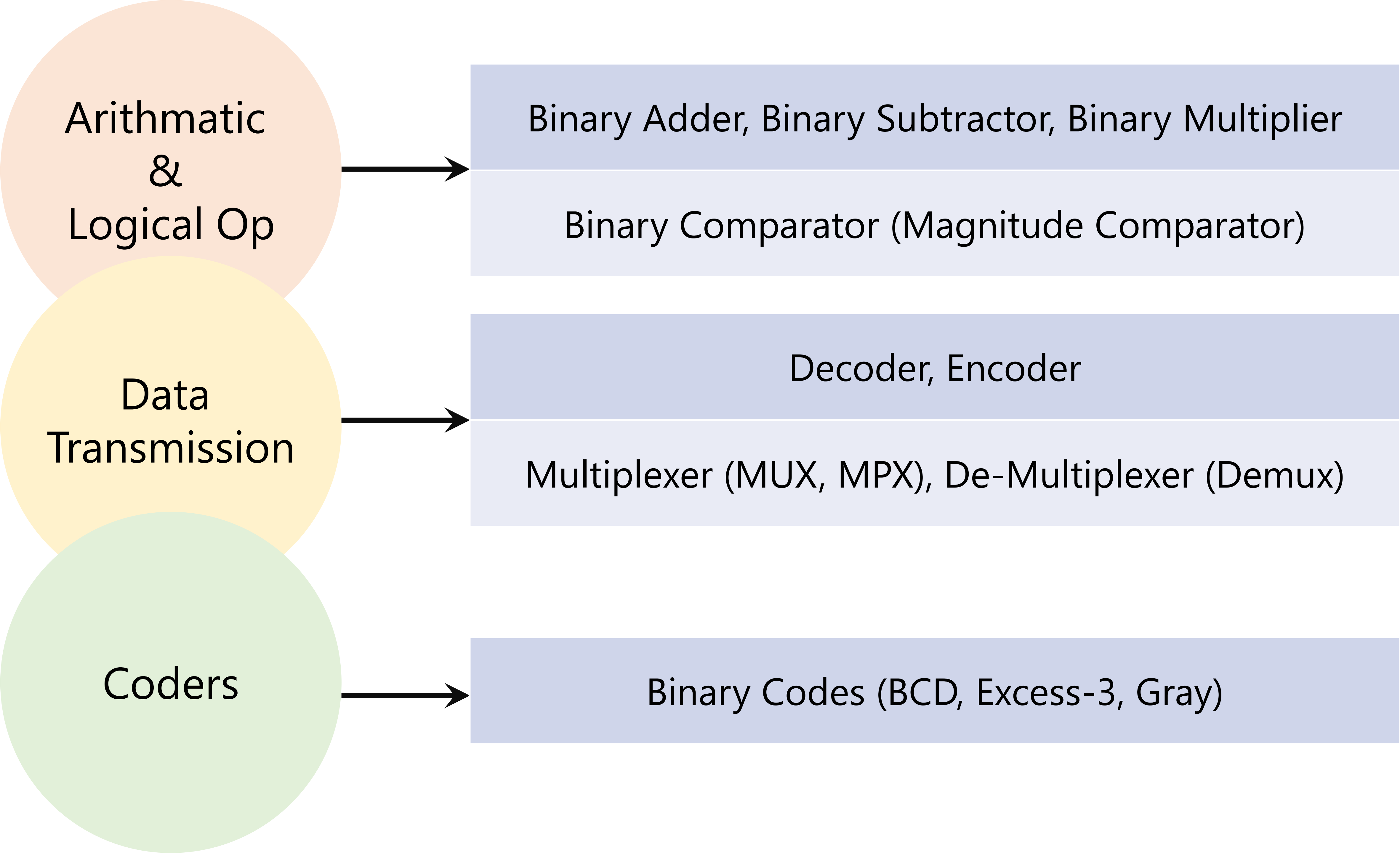


Combinational Logic

Arithmatic & Logical Op

Data Transmission

Coders



Arithmatic & Logical Op

Binary Adder, Binary Subtractor, Binary Multiplier

Binary Comparator (Magnitude Comparator)

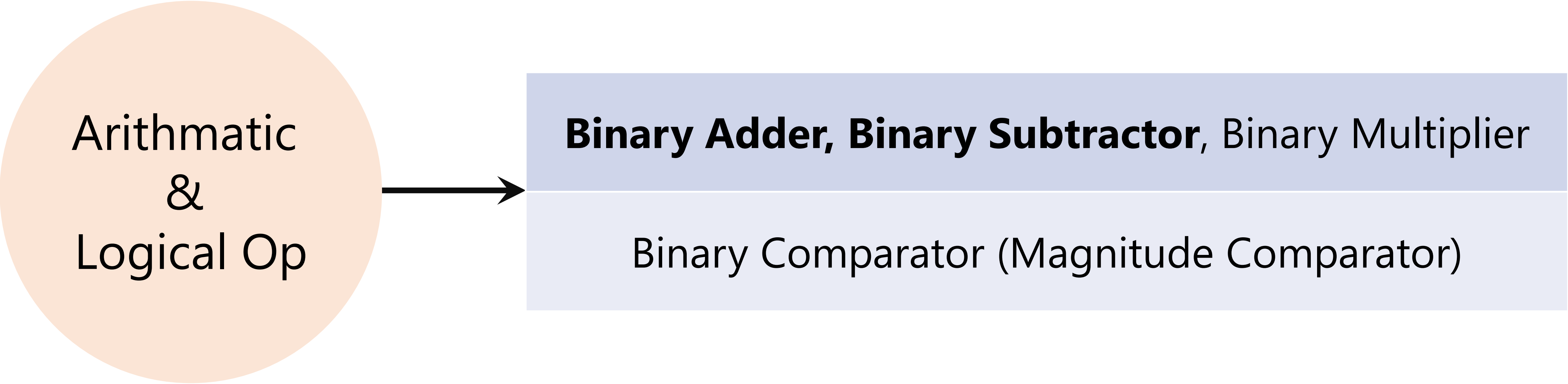
Data Transmission

Decoder, Encoder

Multiplexer (MUX, MPX), De-Multiplexer (Demux)

Coders

Binary Codes (BCD, Excess-3, Gray)

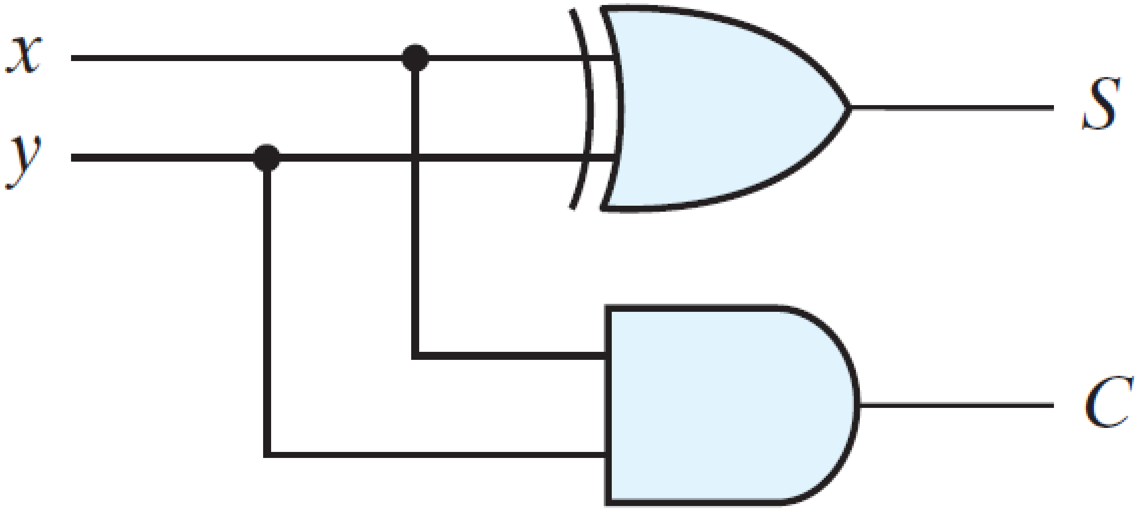


Arithmatic
&
Logical Op

Binary Adder, Binary Subtractor, Binary Multiplier

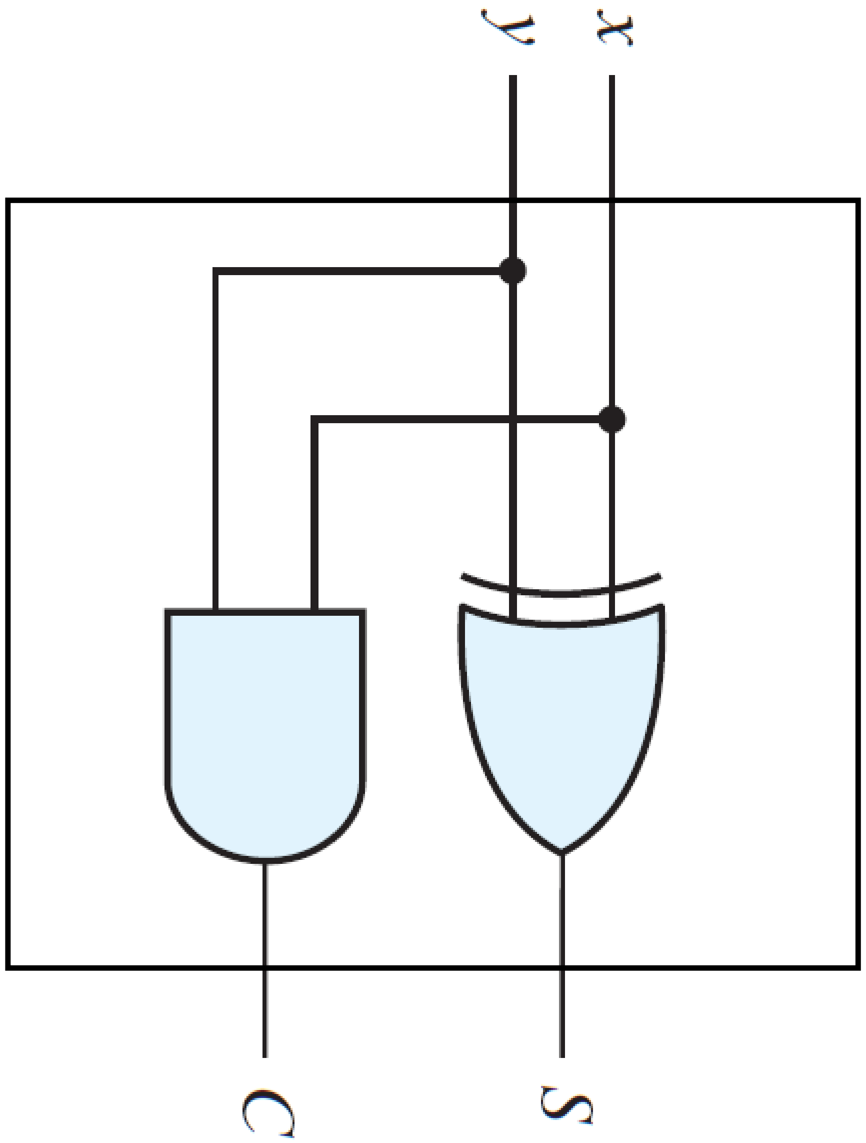
Binary Comparator (Magnitude Comparator)

Binary Adder

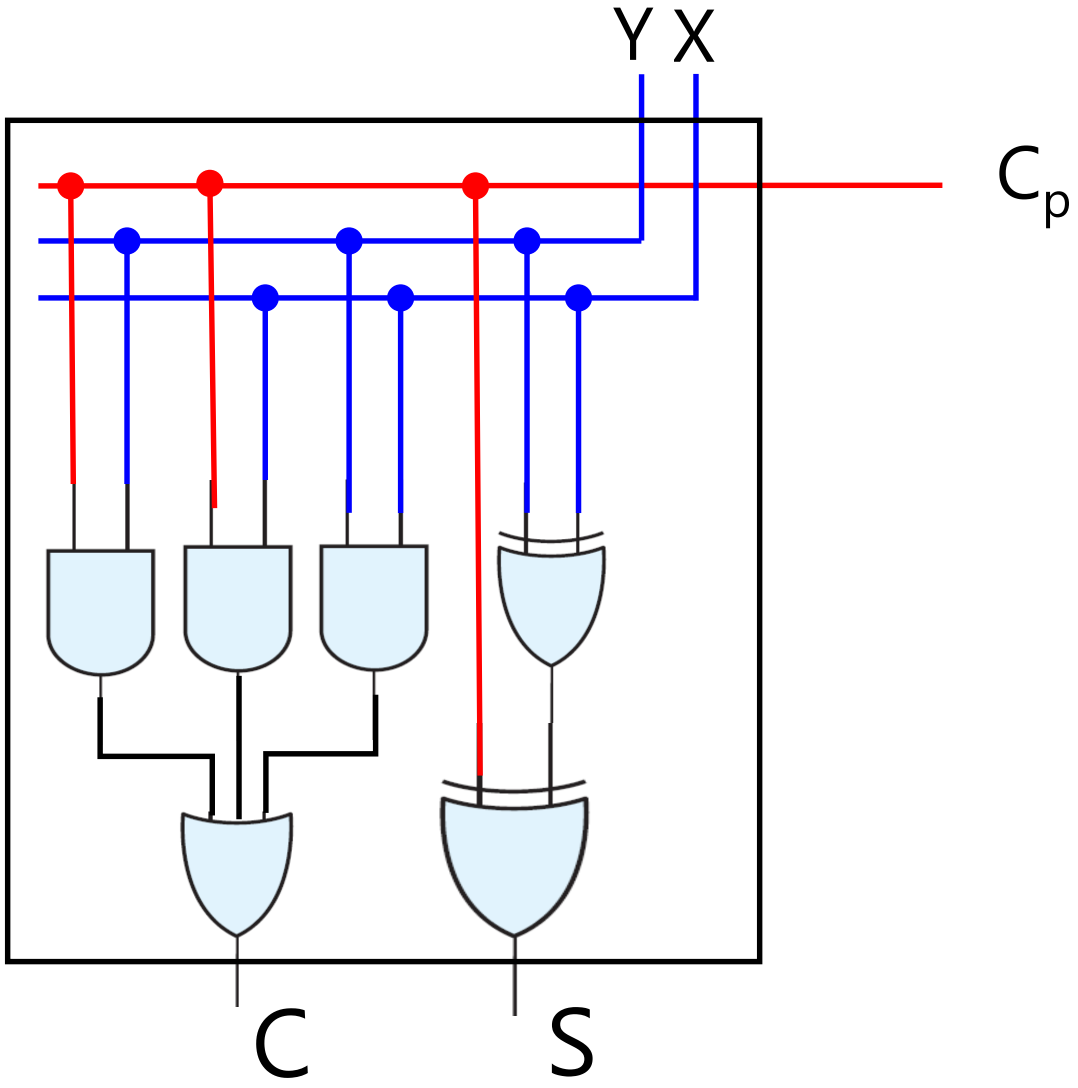


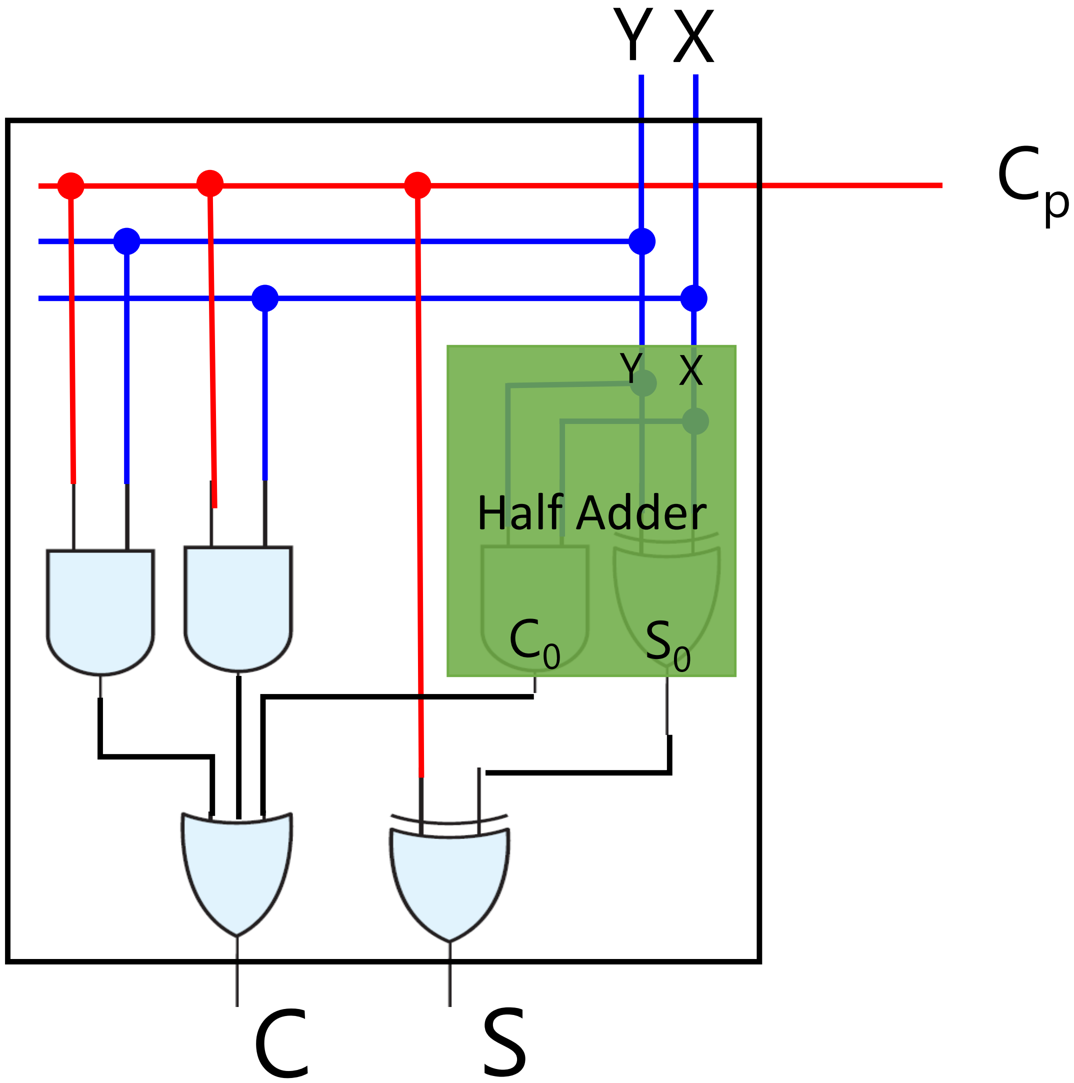
$$S = x \oplus y$$
$$C = xy$$

Half Adder

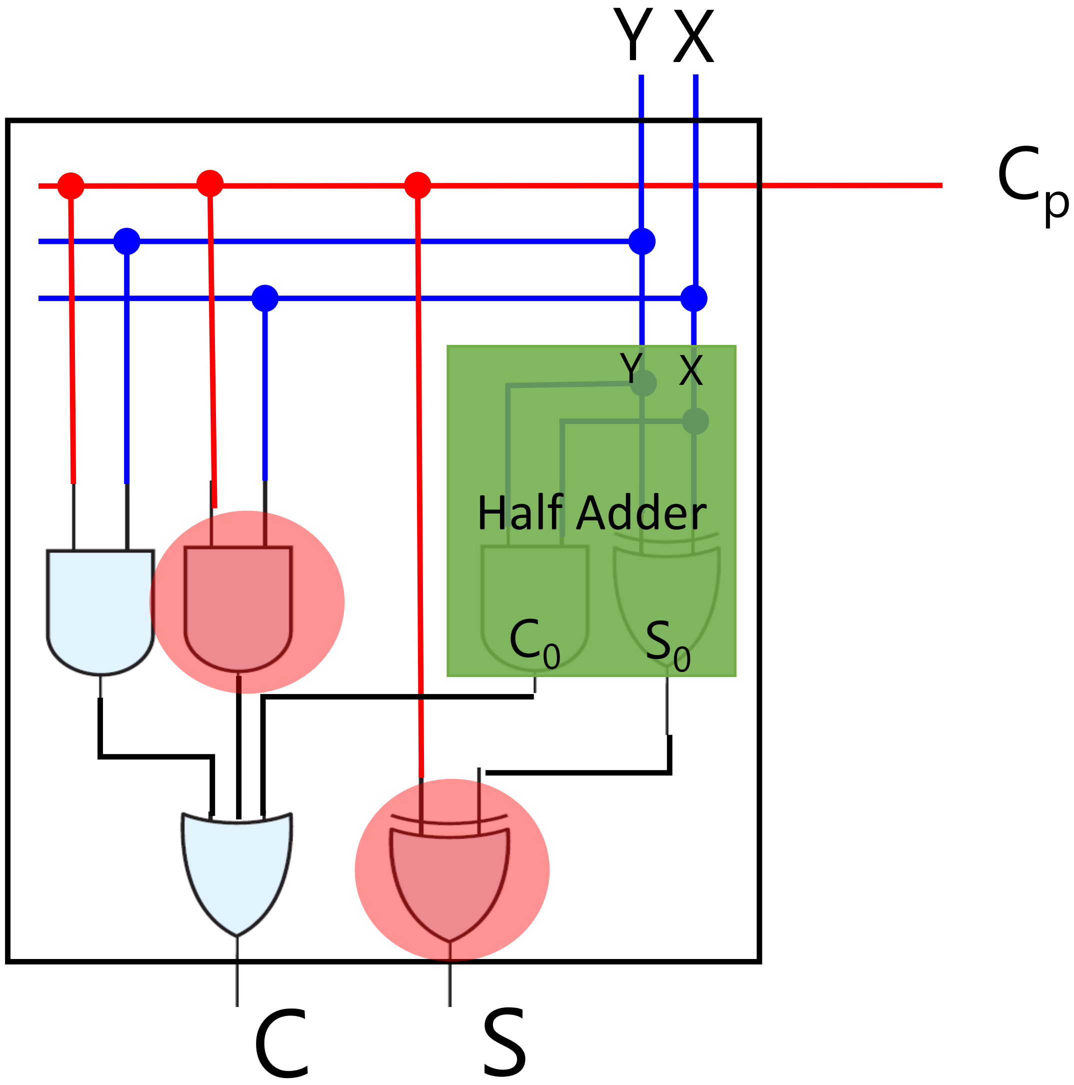

y
x
C
S

Full Adder I


Y
X
C_p
C
S

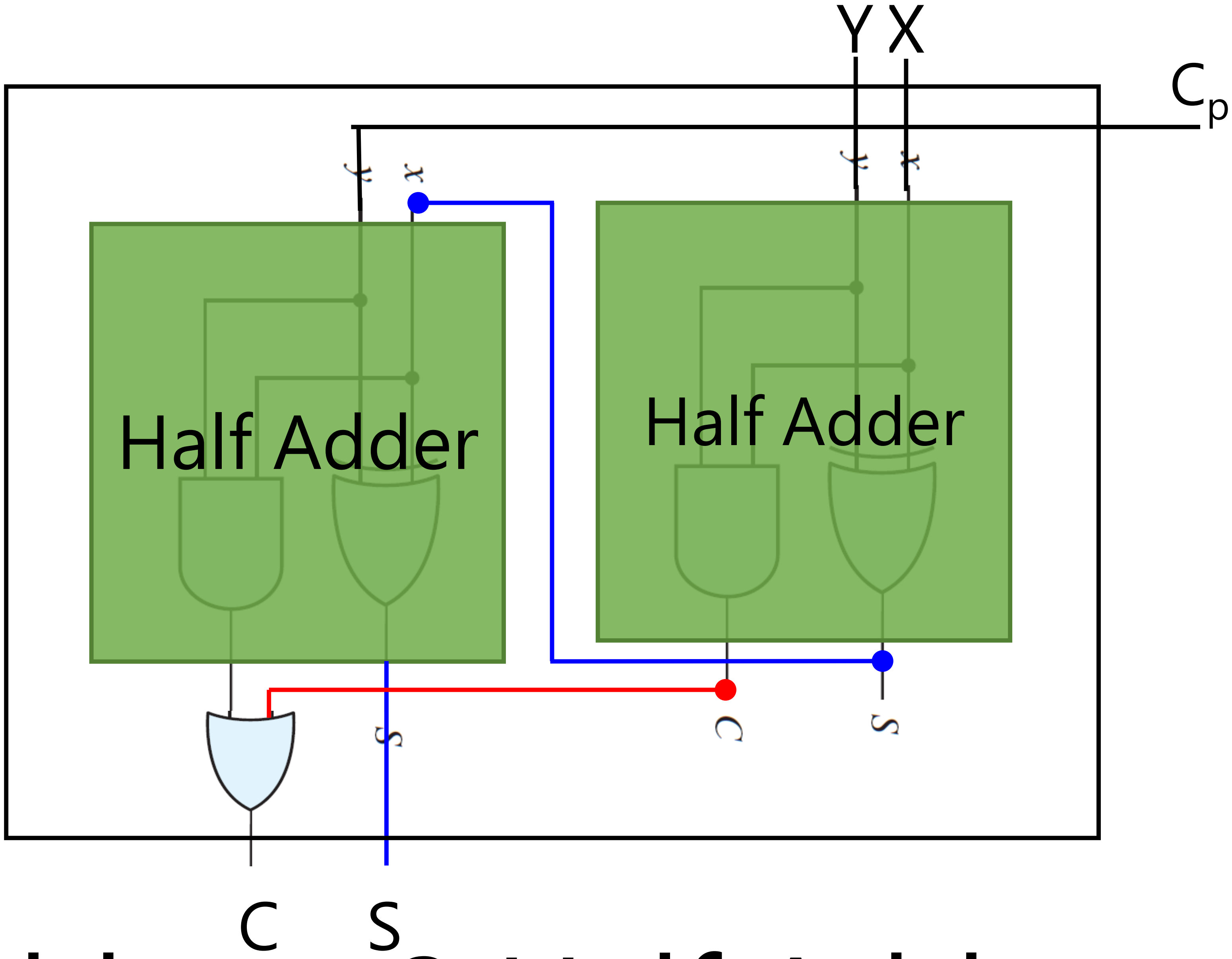


Full Adder = Half Adder + ...



Full Adder $\overset{?}{=}$ 2 Half Adder + …

Full Adder II

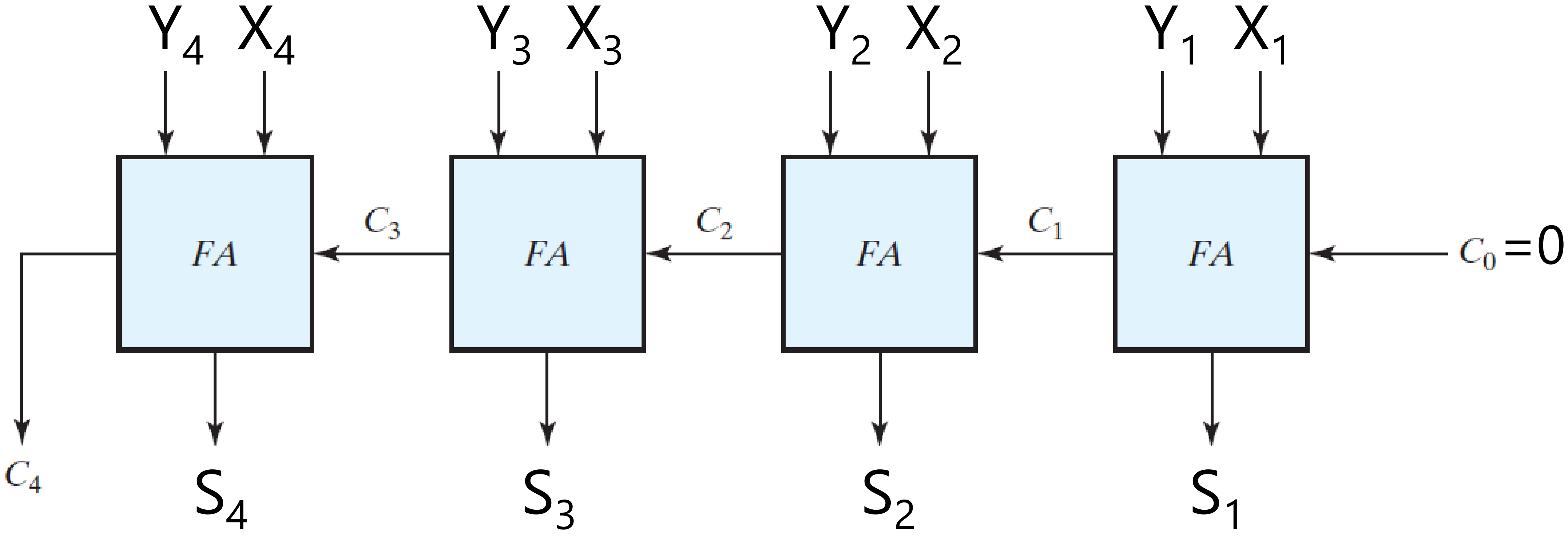


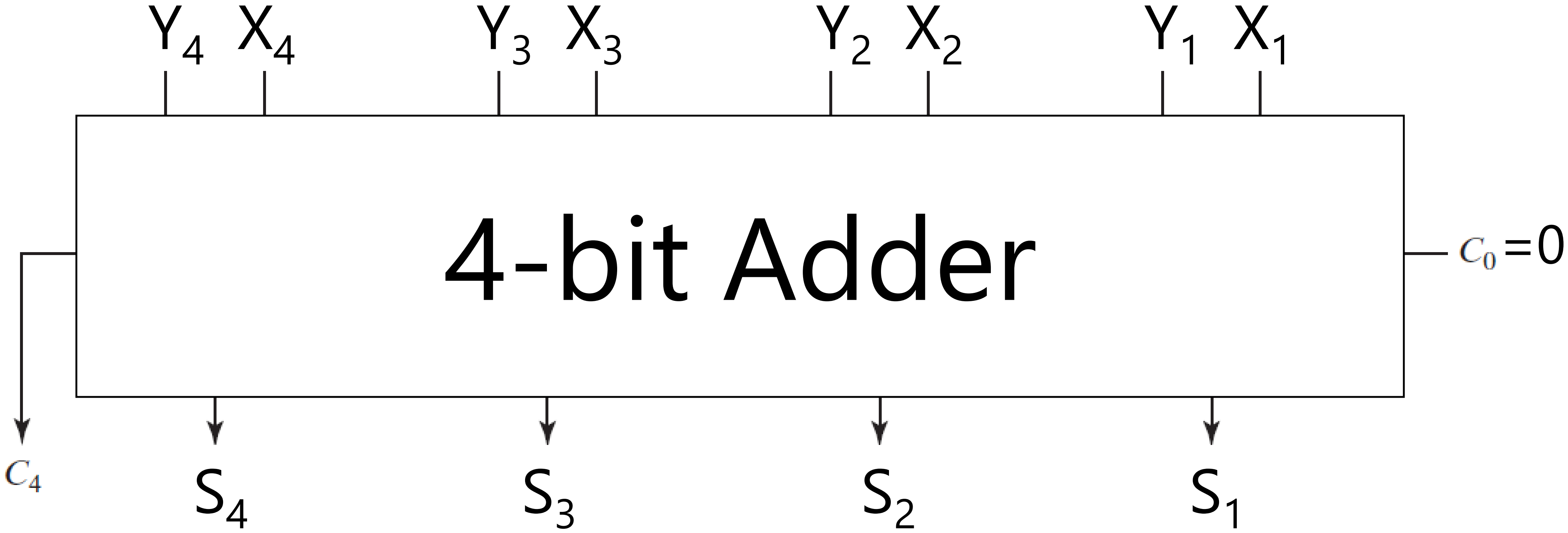
Full Adder = 2 Half Adder + OR

n-Bit Adder

$$\begin{array}{rr} & C_3C_2C_1C_0 \\ & X_4X_3X_2X_1 \\ + & Y_4Y_3Y_2Y_1 \\ \hline \text{C4} & S_4S_3S_2S_1 \end{array}$$

$C_0=0$


Y4 X4
Y3 X3
Y2 X2
Y1 X1
FA
FA
FA
FA
C3
C2
C1
C0 =0
C4
S4
S3
S2
S1


Y_4
X_4
Y_3
X_3
Y_2
X_2
Y_1
X_1
4-bit Adder
$c_0=0$
c_4
S_4
S_3
S_2
S_1

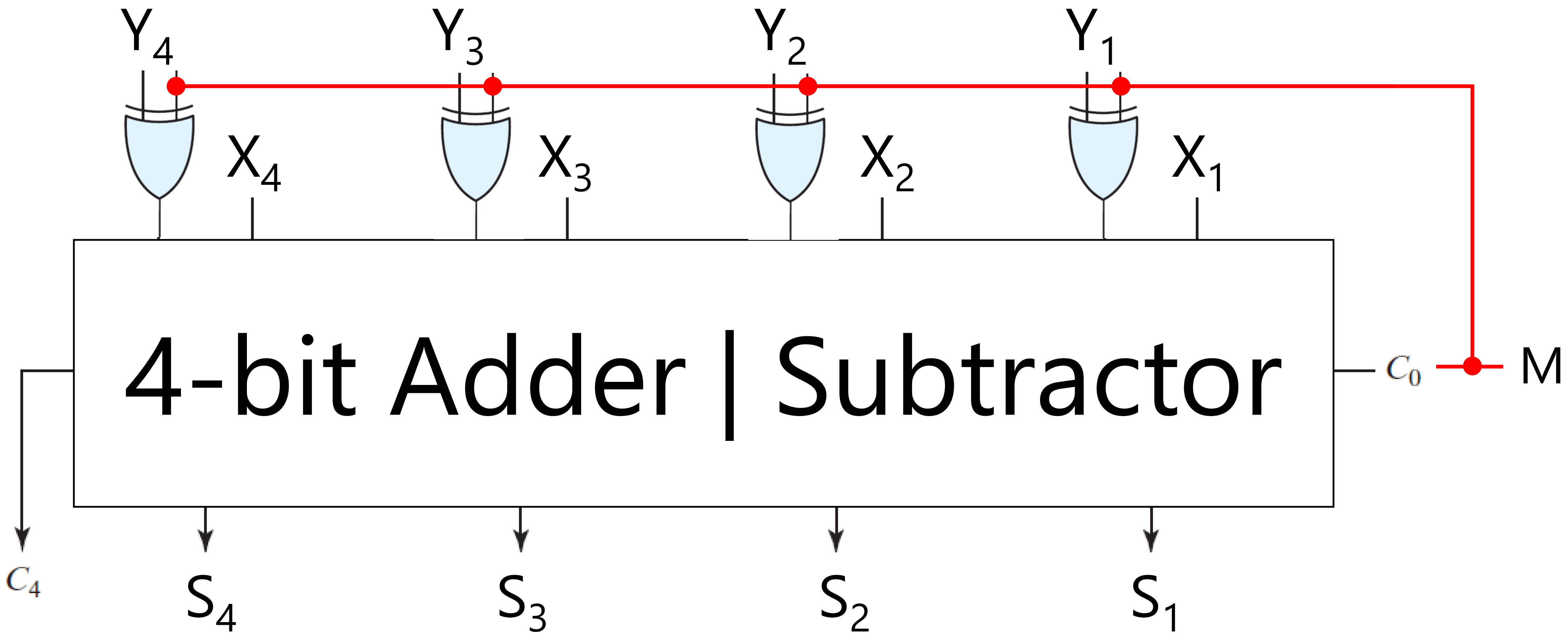
Carry Lookahead

$C_{1:n}$ → Constant Delay

Book Page 138-141

Binary Subtractor

Signed-2’s-Complement



M=0 → Adder
M=1 → Subtractor

Overflow

Signed-2’s-Complement

Design a logic circuit that detects overflow?

Signed-2’s-Complement

Truth Table

Y4	Y3	Y2	Y1	X4	X3	X2	X1	C0	OVF
4-Bit Adder									?

Y3	Y2	Y1	X3	X2	X1	C0	OVF
3-Bit Adder							?

Y2	Y1	X2	X1	C0	OVF
2-Bit Adder					?

Design a logic circuit that detects overflow?

Signed-2’s-Complement

Using Prior Knowledge

Design a logic circuit that detects overflow?

Signed-2’s-Complement

(I)

Subtraction → Addition with 2’s Comp.

Design a logic circuit that detects overflow?

Signed-2’s-Complement

(II)

Sum of Positive Numbers → Negative: OVF=1

Sum of Negative Numbers → Positive: OVF=1

Design a logic circuit that detects overflow?

Signed-2’s-Complement

(III)

Binary System→ The most significant bit → Sign

Base-r in Radix Complement

r^{n-1}	r^{n-2}	r^{n-3}	…	r^2	r^1	r^0

0 <= Positive Numbers <= $(r^n - 1) \div 2$

Base-2: 0,111,…,111
Base-4: 1,333,…,333
Base-8: 3,777,…,777
Base-10: 4,999,…,999
Base-16: 7,FFF,…,FFF

Nothing to do!

W03

Base-r in Radix Complement

r^{n-1}	r^{n-2}	r^{n-3}	…	r^2	r^1	r^0

$(r^n - 1) \div 2 + 1 <=$ Negative Numbers $<= (r^n - 1)$

Base-2: 1,000,…,000

Base-4: 2,000,…,000

Base-8: 4,000,…,000

Base-10: 5,000,…,000

Base-16: 8,000,…,000

W03

We see positive number, but we interpret negative!

$= -$ (r's comp. (#)) $= -$ ((r–1)'s comp. (#) + 1)

$$\begin{array}{cr} & C_3C_2C_1C_0 \\ & X_4X_3X_2X_1 \\ + & Y_4Y_3Y_2Y_1 \\ \cline{2-2} \text{C4} & S_4S_3S_2S_1 \end{array}$$

$$\begin{array}{lr} & C_3C_2C_1C_0 \\ & X_4=0\ X_3X_2X_1 \\ + & Y_4=0\ Y_3Y_2Y_1 \\ \hline C4 & S_4=1\ S_3S_2S_1 \end{array}$$

OVF!

$$
\begin{array}{rr}
 & C_3=1\ C_2C_1C_0 \\
 & X_4=0\ X_3X_2X_1 \\
+ & Y_4=0\ Y_3Y_2Y_1 \\
\hline
C4=0 & S_4=1\ S_3S_2S_1
\end{array}
$$

OVF!

$$\begin{array}{rr} & C_3C_2C_1C_0 \\ & X_4=1\ X_3X_2X_1 \\ + & Y_4=0\ Y_3Y_2Y_1 \\ \hline C4 & S_4=?\ S_3S_2S_1 \end{array}$$

S_4 is guaranteed to be correct in signed-2’s-comp.
Don’t believe it, try!

$$\begin{array}{rr} & C_3C_2C_1C_0 \\ & X_4=0\ X_3X_2X_1 \\ + & Y_4=1\ Y_3Y_2Y_1 \\ \hline C4 & S_4=?\ S_3S_2S_1 \end{array}$$

S_4 is guaranteed to be correct in signed-2’s-comp.

Don’t believe it, try!

$$\begin{array}{cl} & C_3C_2C_1C_0 \\ & X_4=1\ X_3X_2X_1 \\ + & Y_4=1\ Y_3Y_2Y_1 \\ \hline C4 & S_4=0\ S_3S_2S_1 \end{array}$$

OVF!

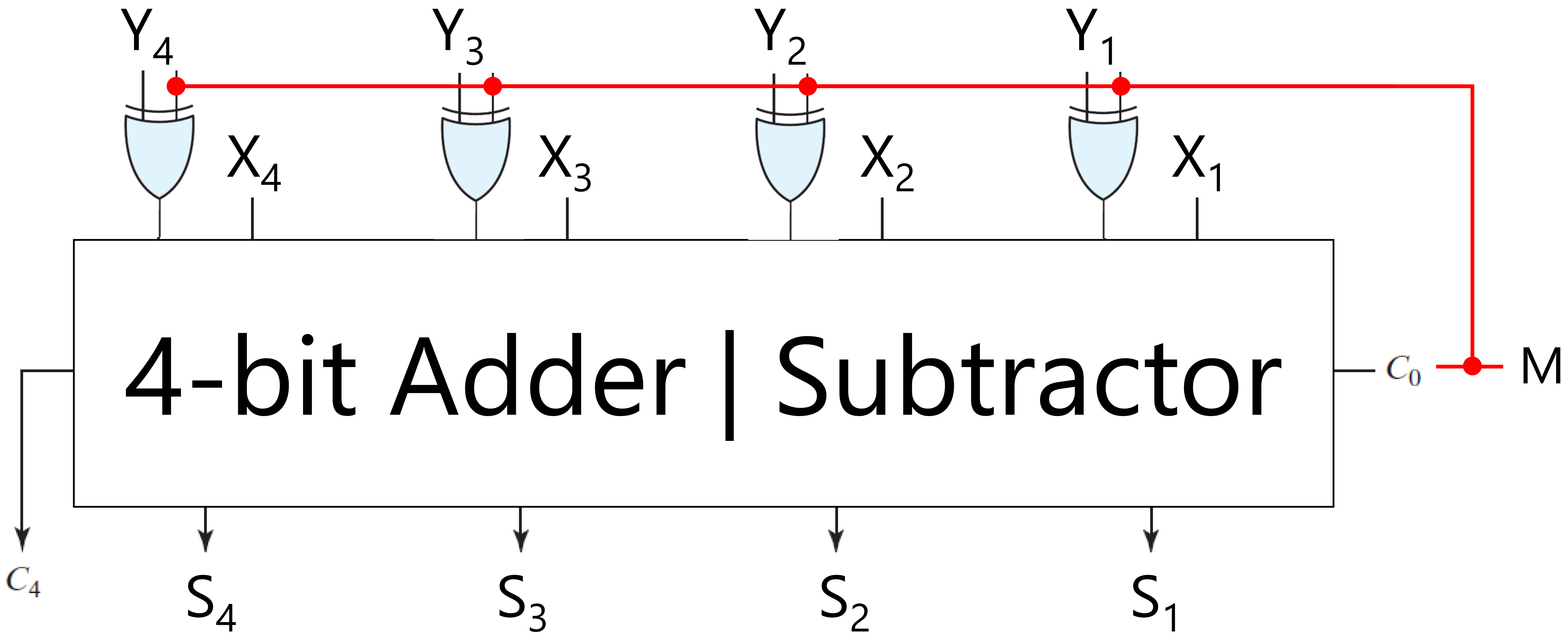
$$\begin{array}{rl} & C_3=0\ C_2C_1C_0 \\ & X_4=1\ X_3X_2X_1 \\ + & Y_4=1\ Y_3Y_2Y_1 \\ \hline C4=1 & S_4=0\ S_3S_2S_1 \end{array}$$

OVF!

Design a logic circuit that detects overflow?

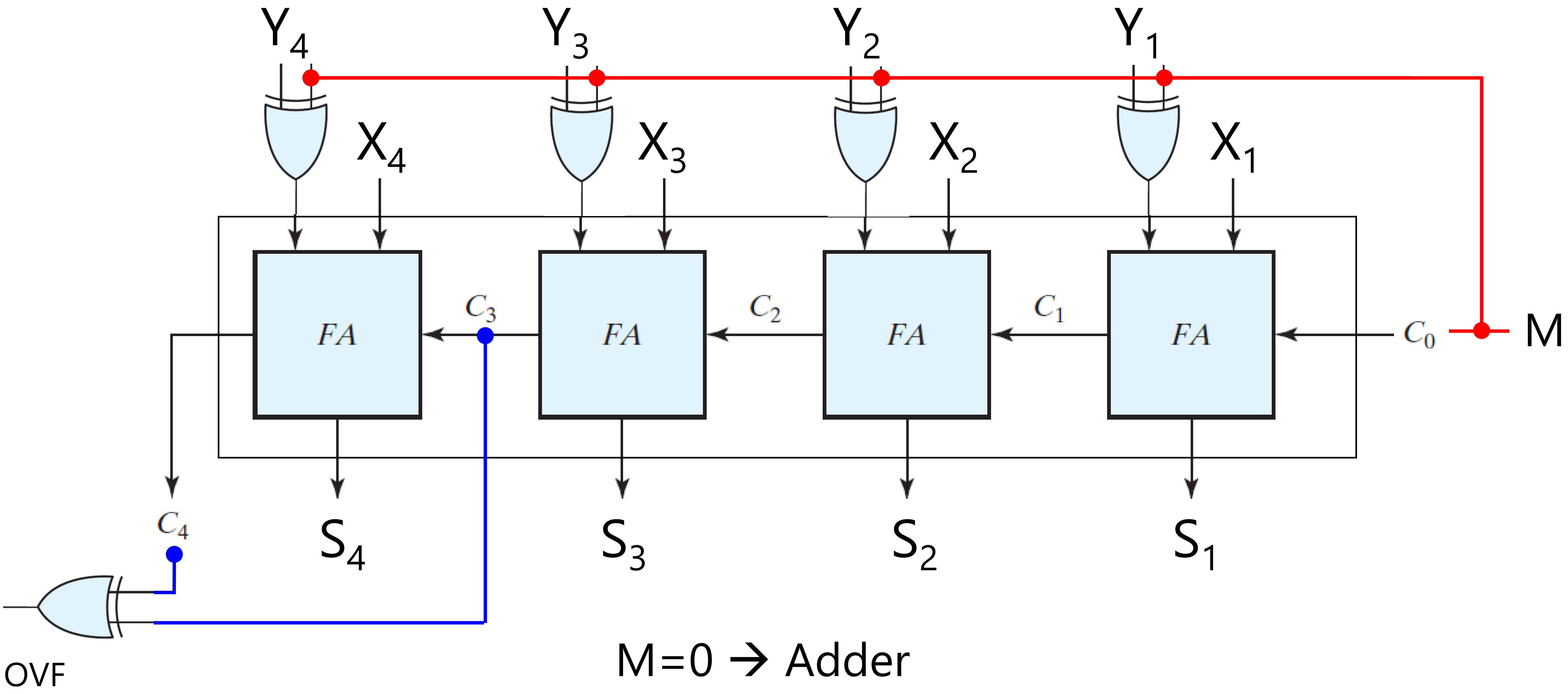
Signed-2's-Complement

$$OVF = C'_4C_3 + C_4C'_3 = C_4 \oplus C_3$$



M=0 → Adder

M=1 → Subtractor



M=0 → Adder

M=1 → Subtractor

Binary Adder | Subtractor | Overflow

Unsigned?

Hossein’s way!

Unsigned: Signed Positive

Binary Adder

Unsigned: Signed Positive

$$\begin{array}{lr} & C_3C_2C_1C_0 \\ & 0\ X_3X_2X_1 \\ + & 0\ Y_3Y_2Y_1 \\ \hline C4 & S_4S_3S_2S_1 \end{array}$$

Binary Subtractor

Unsigned: Signed Positive

$$\begin{array}{lr} & C_3C_2C_1C_0 \\ & 0\ X_3X_2X_1 \\ - & 0\ Y_3Y_2Y_1 \\ \hline C4 & S_4S_3S_2S_1 \end{array}$$

$$\begin{array}{lrr}
 & & C_3C_2C_1C_0 \\
 & & 0\ X_3X_2X_1 \\
+ & & \text{2's-comp}(0\ Y_3Y_2Y_1) \\
\hline
 & C4 & S_4S_3S_2S_1
\end{array}$$

Binary Overflow

Unsigned: Signed Positive

$$\begin{array}{cr} & C_3C_2C_1C_0 \\ & 0\ X_3X_2X_1 \\ + & 0\ Y_3Y_2Y_1 \\ \hline C4 & 1\ S_3S_2S_1 \end{array}$$

OVF!

$$\begin{array}{rr} & C_3=1C_2C_1C_0 \\ & 0\ X_3X_2X_1 \\ + & 0\ Y_3Y_2Y_1 \\ \hline C4 & 1\ S_3S_2S_1 \end{array}$$

OVF!

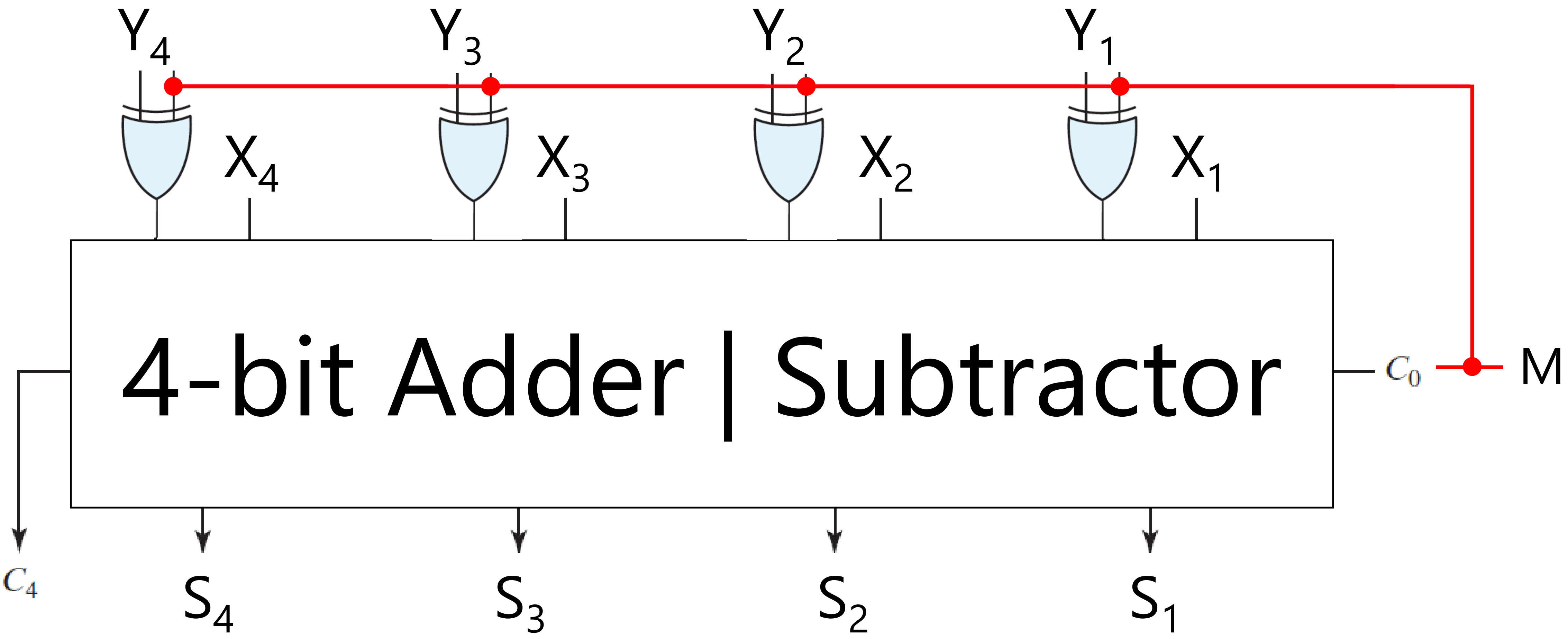
Binary Overflow

Unsigned: Signed Positive

OVF = C_3 for 3-bit unsigned adder!

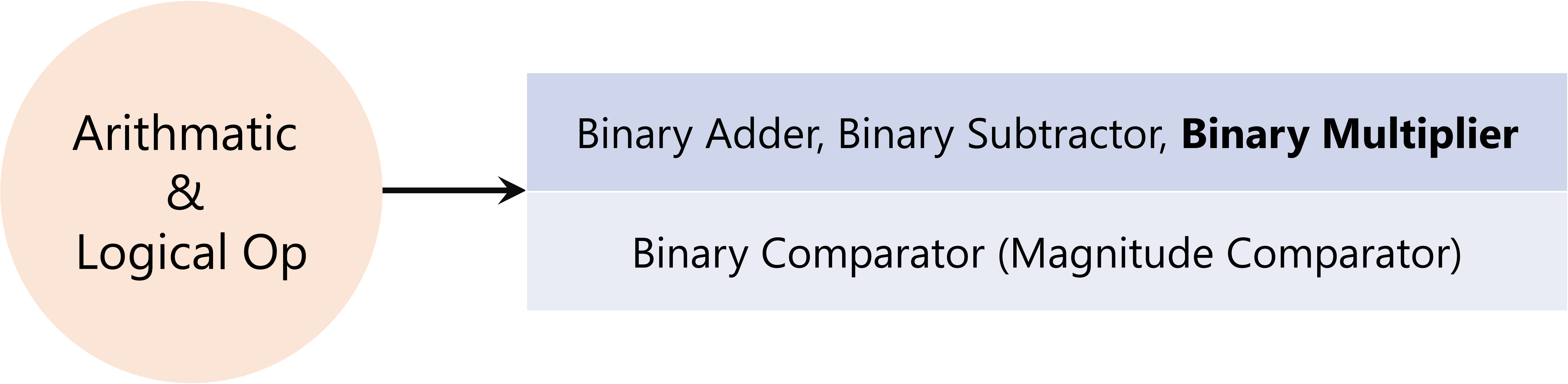
Book’s way!

Unsigned



M=0 → Adder

M=1 → Subtractor



Arithmatic
&
Logical Op

Binary Adder, Binary Subtractor, **Binary Multiplier**

Binary Comparator (Magnitude Comparator)

Binary Multiplier

Unsigned

Binary Multiplier

Unsigned

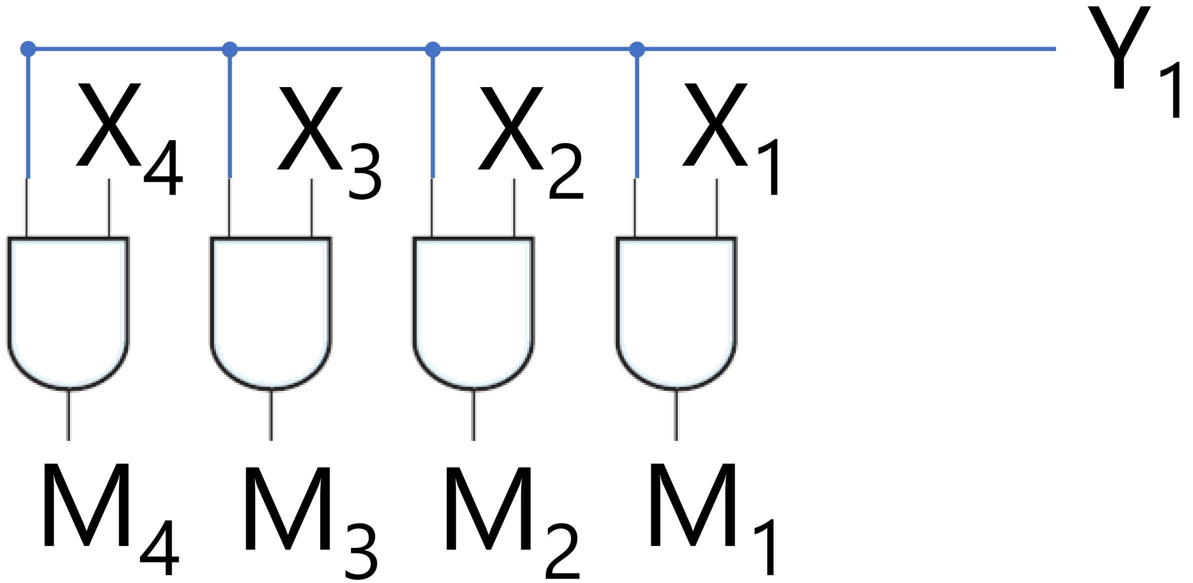
$$\begin{array}{lr} & X_4X_3X_2X_1 \\ \times & Y_1 \\ \hline & M_4M_3M_2M_1 \end{array}$$

$$\begin{array}{lr} & X_4X_3X_2X_1 \\ \times & Y_1 \\ \hline & M_1=Y_1X_1 \end{array}$$

$$\begin{array}{lr} & X_4X_3X_2X_1 \\ \times & Y_1 \\ \hline & M_2=Y_1X_2 \end{array}$$

$$\begin{array}{lr} & X_4X_3X_2X_1 \\ \times & Y_1 \\ \hline & M_3=Y_1X_3 \end{array}$$

$$\begin{array}{lr} & X_4X_3X_2X_1 \\ \times & Y_1 \\ \hline & M_4=Y_1X_4 \end{array}$$


Y1
X4
X3
X2
X1
M4
M3
M2
M1

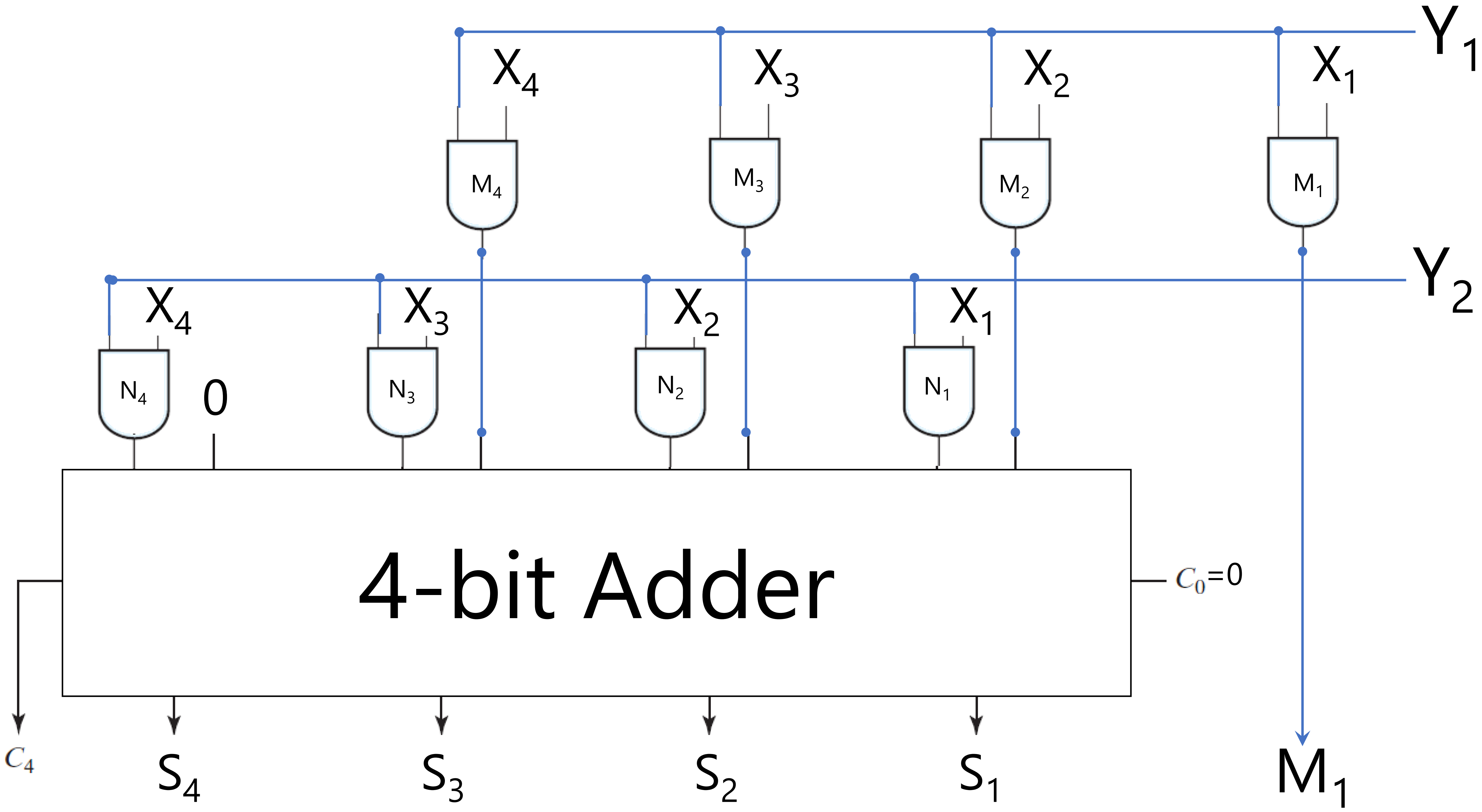
$$\begin{array}{rr} & X_4X_3X_2X_1 \\ \times & Y_2Y_1 \\ \hline & M_4\,M_3\,M_2\,M_1 \end{array}$$

$$\begin{array}{cccccc} & & X_4 & X_3 & X_2 & X_1 \\ \times & & & & Y_2 & Y_1 \\ \hline & & M_4 & M_3 & M_2 & M_1 \\ & N_4 & N_3 & N_2 & N_1 & 0 \end{array}$$

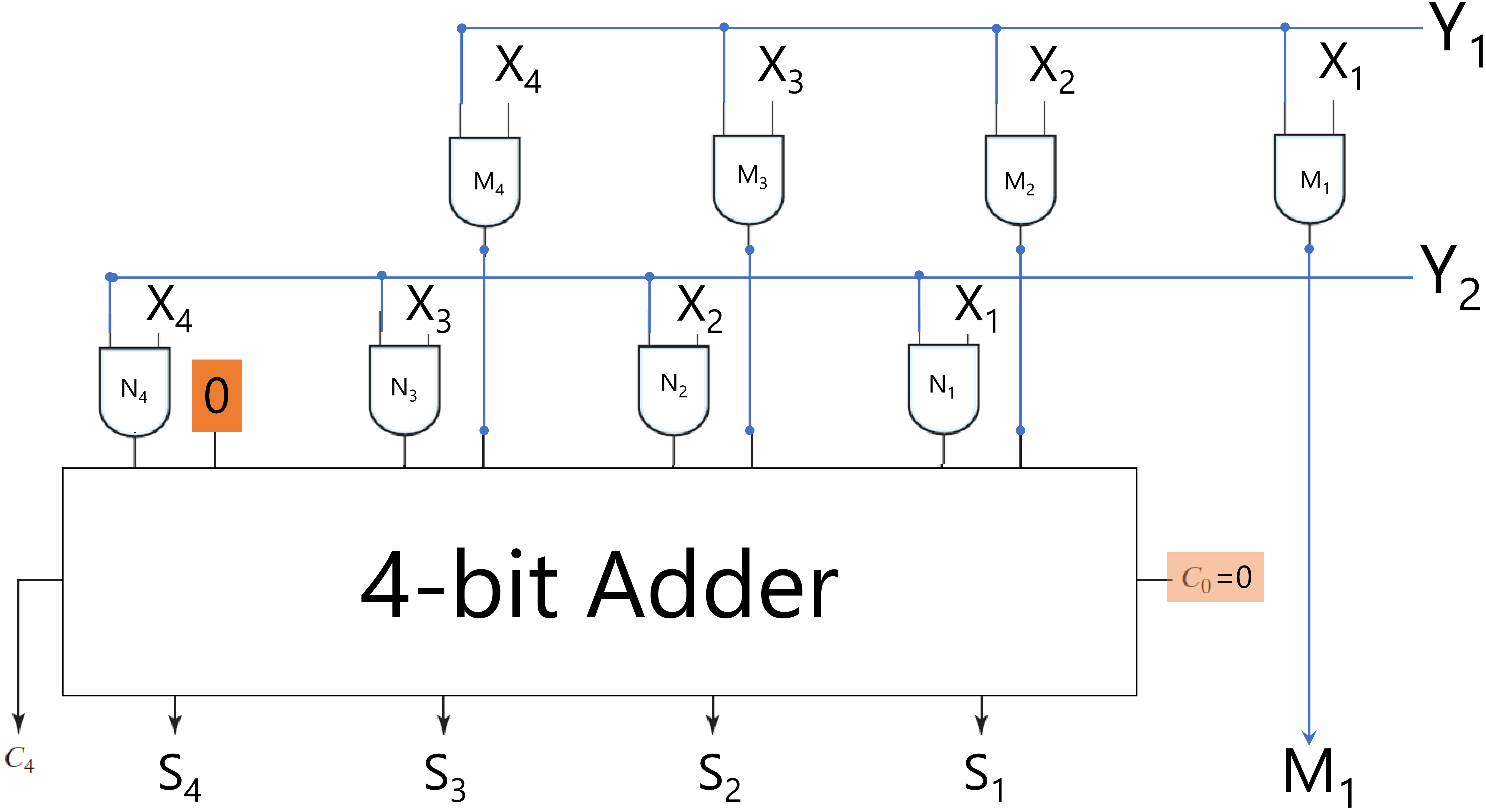
$$
\begin{array}{cccccc}
 & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_2 & Y_1 \\
\hline
 & & M_4 & M_3 & M_2 & M_1 \\
+ & N_4 & N_3 & N_2 & N_1 & 0 \\
\hline
\end{array}
$$

$$\begin{array}{rccccc}
 & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_2 & Y_1 \\
\hline
 & & M_4 & M_3 & M_2 & M_1 \\
+ & N_4 & N_3 & N_2 & N_1 & 0 \\
\hline
 & S_4 & S_3 & S_2 & S_1 & M_1
\end{array}$$

$$\begin{array}{cccccc}
 & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_2 & Y_1 \\
\hline
 & & M_4 & M_3 & M_2 & M_1 \\
+ & N_4 & N_3 & N_2 & N_1 & 0 \\
\hline
 & S_4 & S_3 & S_2 & S_1 & M_1
\end{array}$$



Y1
X4
X3
X2
X1
M4
M3
M2
M1
Y2
X4
X3
X2
X1
N4
0
N3
N2
N1
4-bit Adder
C0=0
C4
S4
S3
S2
S1
M1

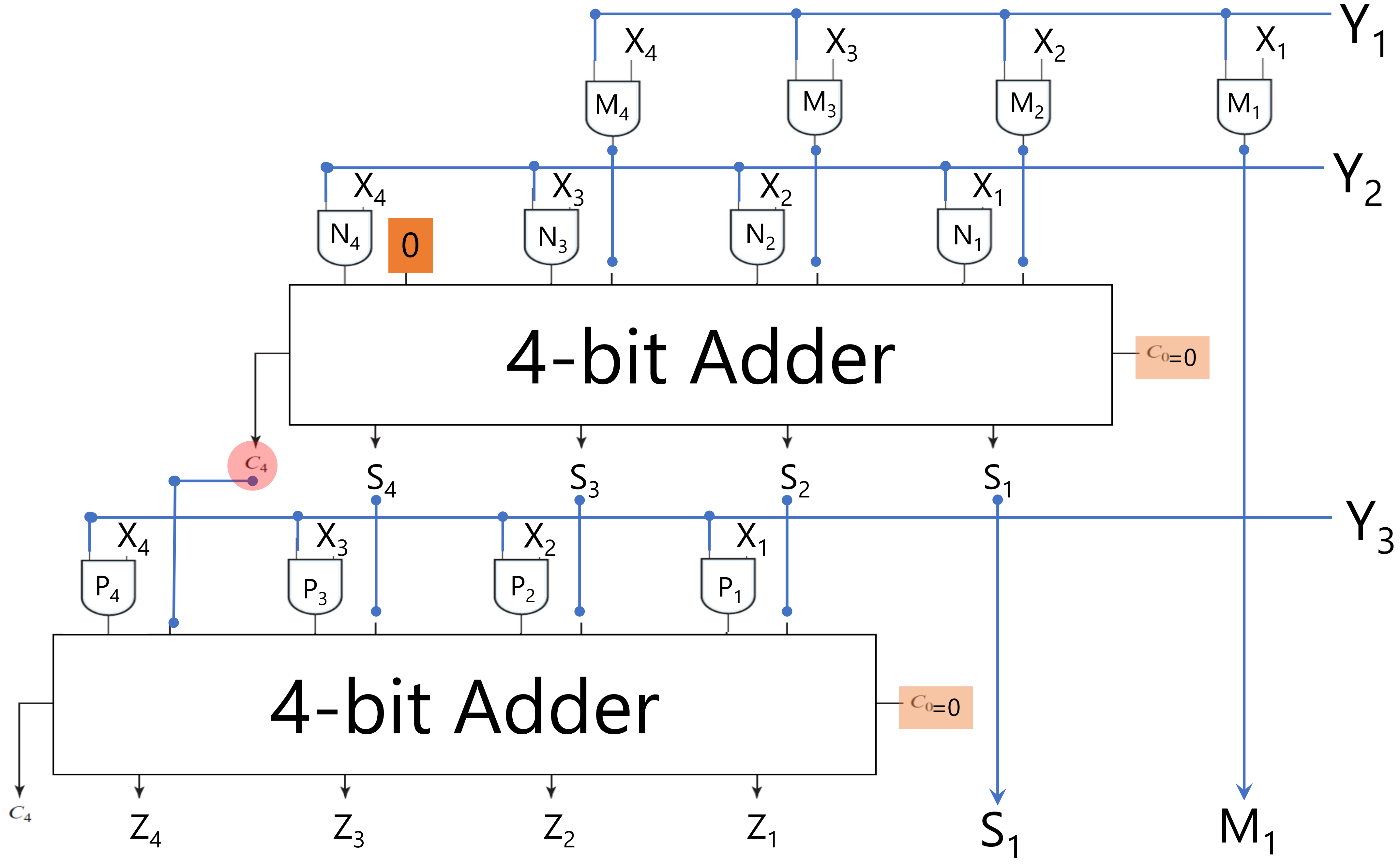

Y_1
X_4
X_3
X_2
X_1
M_4
M_3
M_2
M_1
Y_2
X_4
X_3
X_2
X_1
N_4
0
N_3
N_2
N_1
4-bit Adder
$C_0 = 0$
C_4
S_4
S_3
S_2
S_1
M_1

$$
\begin{array}{rcccccc}
 & & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_3 & Y_2 & Y_1 \\
\hline
 & & & M_4 & M_3 & M_2 & M_1 \\
+ & & N_4 & N_3 & N_2 & N_1 & 0 \\
 & P_4 & P_3 & P_2 & P_1 & 0 & 0 \\
\hline
\end{array}
$$

$$\begin{array}{rcccccc}
 & & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_3 & Y_2 & Y_1 \\
\hline
+ & & S_4 & S_3 & S_2 & S_1 & M_1 \\
 & P_4 & P_3 & P_2 & P_1 & 0 & 0 \\
\hline
\end{array}$$

$$
\begin{array}{c cccccc}
 & & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_3 & Y_2 & Y_1 \\
\hline
+ & & S_4 & S_3 & S_2 & S_1 & M_1 \\
 & P_4 & P_3 & P_2 & P_1 & 0 & 0 \\
\hline
 & Z_4 & Z_3 & Z_2 & Z_1 & S_1 & M_1
\end{array}
$$

$$
\begin{array}{ccccccc}
 & & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_3 & Y_2 & Y_1 \\
\hline
+ & & S_4 & S_3 & S_2 & S_1 & M_1 \\
 & P_4 & P_3 & P_2 & P_1 & 0 & 0 \\
\hline
 & Z_4 & Z_3 & Z_2 & Z_1 & S_1 & M_1
\end{array}
$$



Y_1
X_4
X_3
X_2
X_1
M_4
M_3
M_2
M_1
Y_2
X_4
X_3
X_2
X_1
N_4
0
N_3
N_2
N_1
4-bit Adder
C_0=0
C_4
S_4
S_3
S_2
S_1
Y_3
X_4
X_3
X_2
X_1
P_4
P_3
P_2
P_1
4-bit Adder
C_0=0
C_4
Z_4
Z_3
Z_2
Z_1
S_1
M_1

Binary Multiplier

Unsigned

n-bit X × m-bit Y

→ how many output bit?

Binary Multiplier

Unsigned

n-bit X × m-bit Y

→ how many ANDs?

Binary Multiplier

Unsigned

n-bit X × m-bit Y

→ how many k-bit adders?

Binary Multiplier

Unsigned

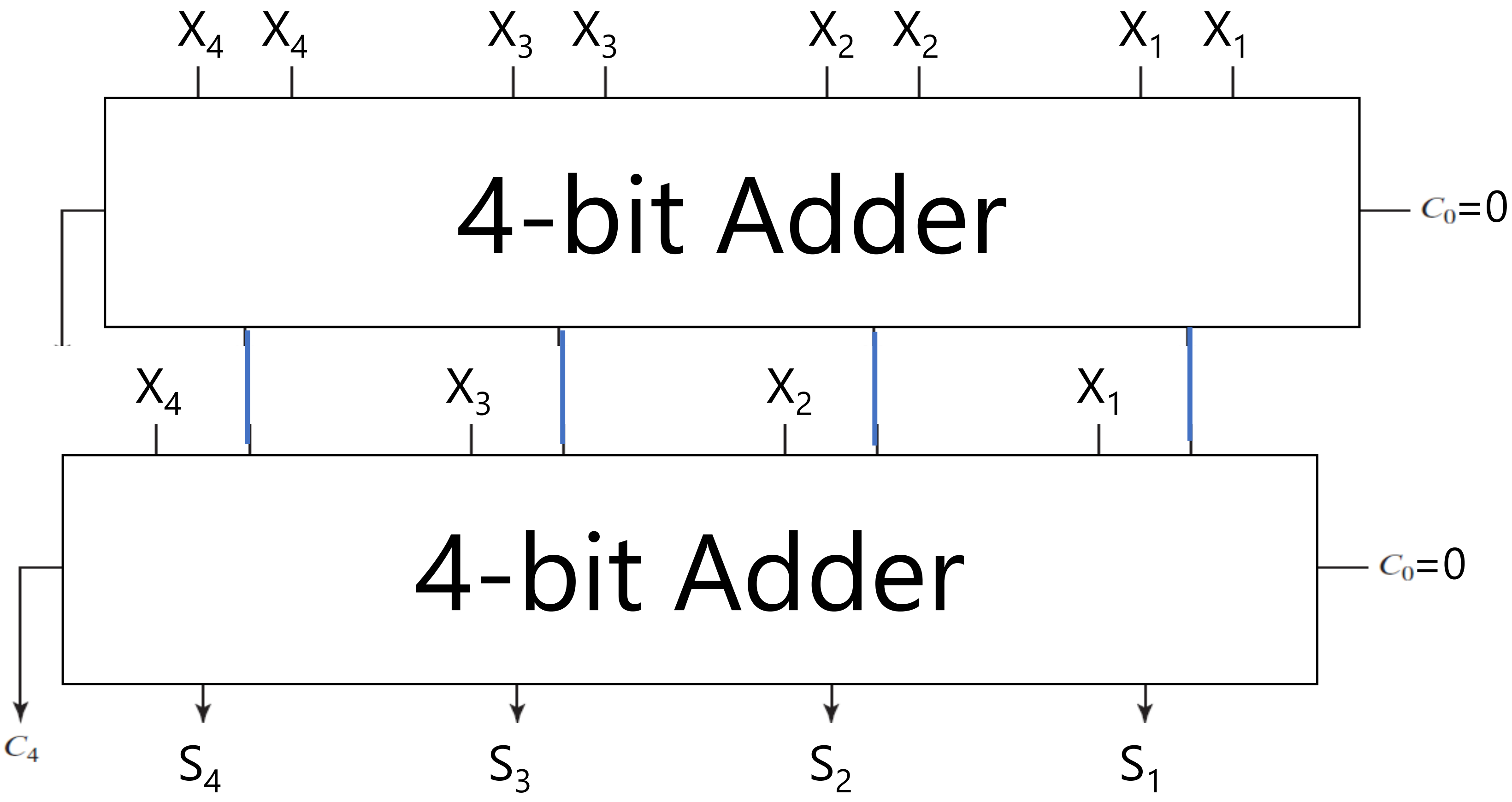
n-bit X × m-bit Y

→ what is k in k-bit adders?

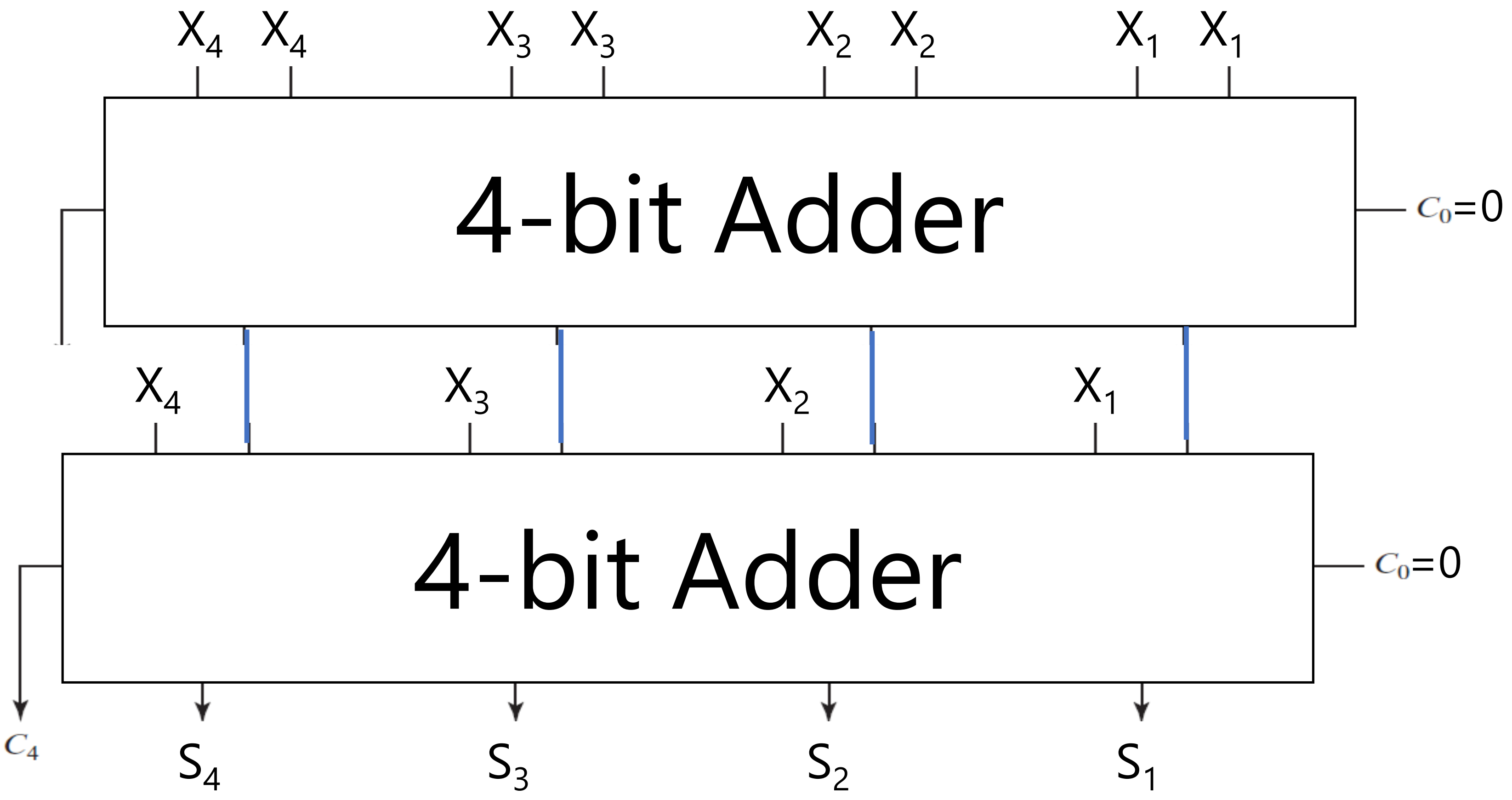
Binary Multiplier II

Unsigned

$$\underbrace{\text{n-bit X + n-bit X + … + n-bit X}}_{\text{m-bit Y times!}}$$

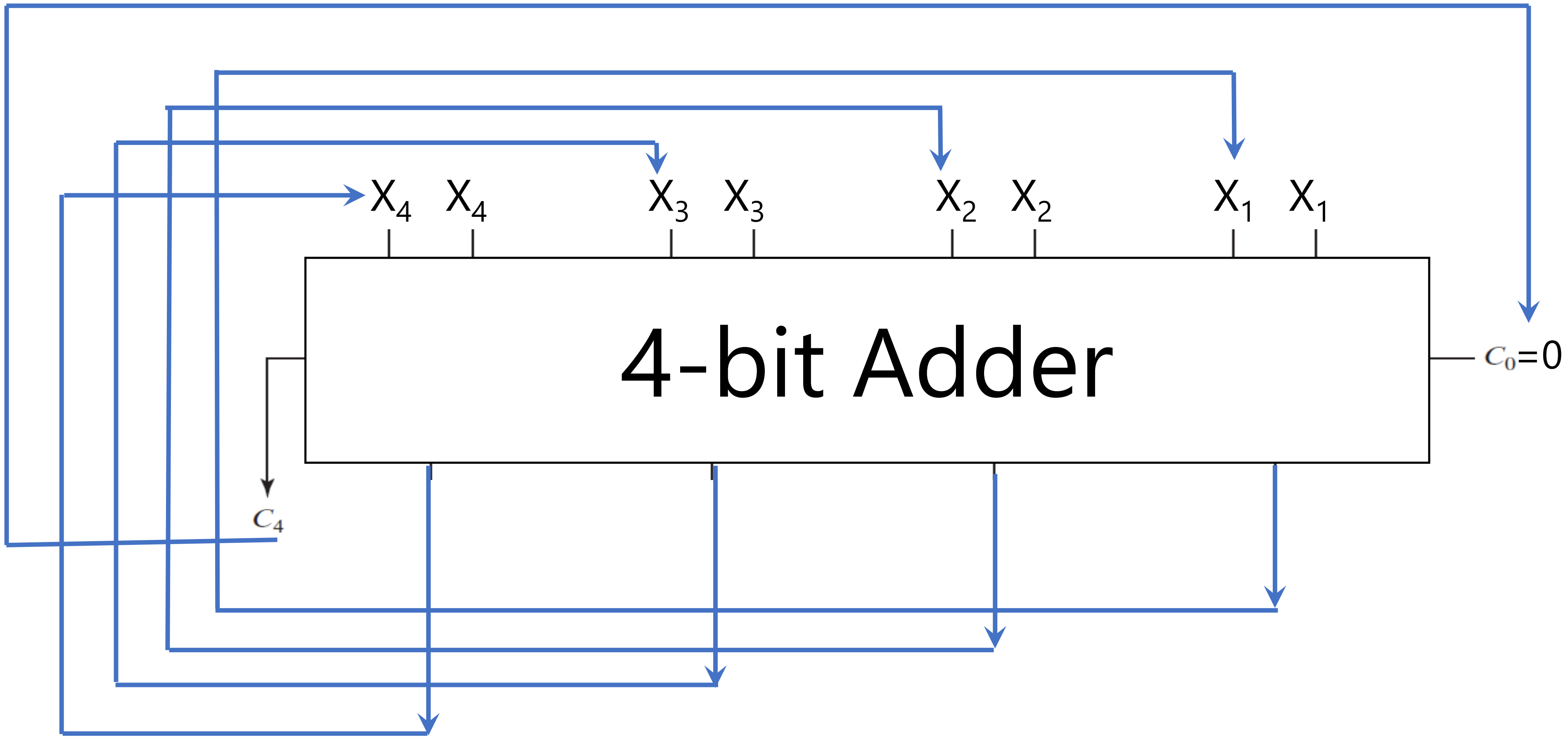


$$X \times (11)_2 = X+X+X$$



$$X \times (11)_2 = X+X+X$$

If you change Y, you have to change circuit!!



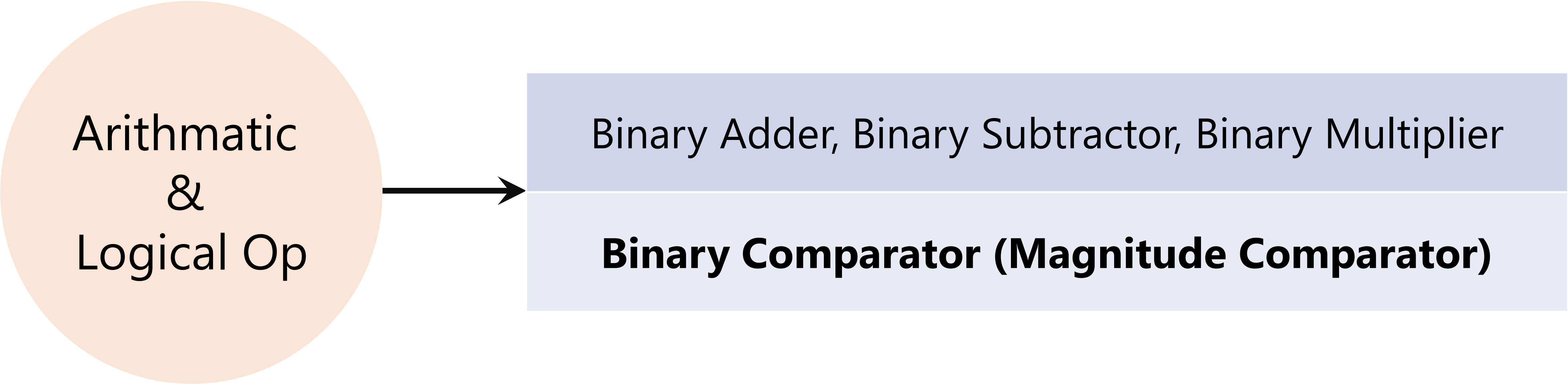
X × Y = X+…+X → When to stop?

Feedback → Sequential Logic

Binary Multiplier

Signed?

n-bit X × m-bit Y



Arithmatic
&
Logical Op

Binary Adder, Binary Subtractor, Binary Multiplier

Binary Comparator (Magnitude Comparator)

Binary Comparator

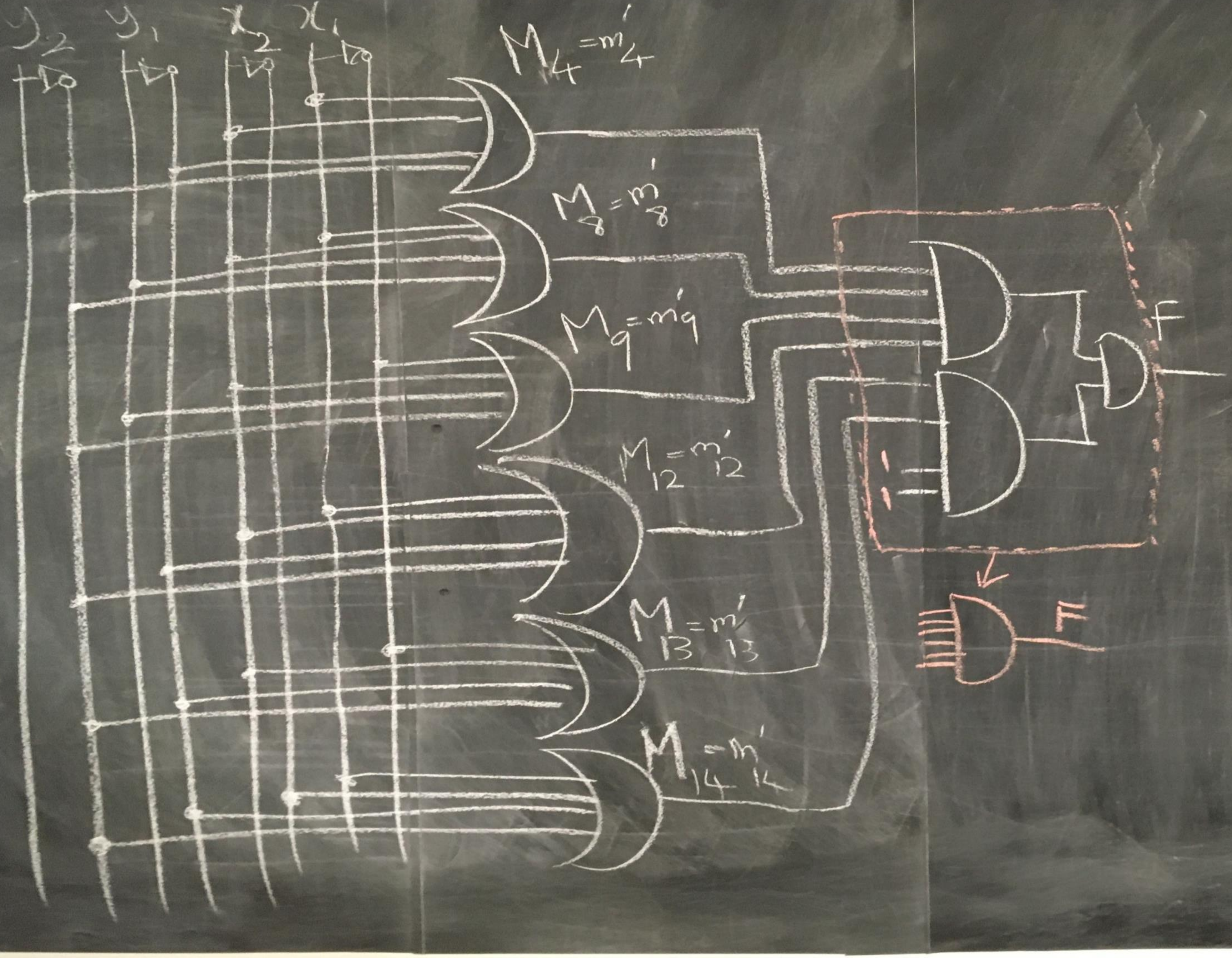
Unsigned

$X > Y$ $X==Y$ $X < Y$

Given two unsigned numbers x and y, design a logic circuit to see

$$x \overset{?}{\geq} y$$

Y2	Y1	X2	X1	F(Y2,Y1,X2,X1)=Σ m(0,1,2,3,5,6,7,10,11,15)	F(Y2,Y1,X2,X1)=Π M(4,8,9,12,13,14)
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0



$$F = \Pi M(4,8,9,12,13,14)$$

$$= M_4 M_8 M_9 M_{12} M_{13} M_{14}$$

$$= m'_4 m'_8 m'_9 m'_{12} m'_{13} m'_{14}$$

$= (y'_2 y_1 x'_2 x'_1)'$ $\quad m'_4 \rightarrow y_2 + y'_1 + x_2 + x_1$

$(y_2 y'_1 x'_2 x'_1)'$ $\quad m'_8 \rightarrow y'_2 + y_1 + x_2 + x_1$

$(y_2 y'_1 x'_2 x_1)'$ $\quad m'_9 \rightarrow y'_2 + y_1 + x_2 + x'_1$

$(y_2 y_1 x'_2 x'_1)'$ $\quad m'_{12} \rightarrow y'_2 + y'_1 + x_2 + x_1$

$(y_2 y_1 x'_2 x_1)'$ $\quad m'_{13} \rightarrow y'_2 + y'_1 + x_2 + x'_1$

$(y_2 y_1 x_2 x'_1)'$ $\quad m'_{14} \rightarrow y'_2 + y'_1 + x'_2 + x_1$

Given two unsigned numbers x and y, design a logic circuit to see

$$x > y\,;x == y\,;\,x < y$$

Y2	Y1	X2	X1	$F_1 = (X > Y)$	$F_2 = (X == Y)$	$F_3 = (X < Y)$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Y2	Y1	X2	X1	$F_1 = (X > Y)$	$F_2 = (X == Y)$	$F_3 = (X < Y)$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1		0
0	0	1	1	1		0
0	1	0	0	0		1
0	1	0	1	0		0
0	1	1	0	1		0
0	1	1	1	1		0
1	0	0	0	0		1
1	0	0	1	0		1
1	0	1	0	0		0
1	0	1	1	1		0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

If X and Y
3, 4, 5, …
bits?!

Binary Subtractor

Unsigned by Hossein’s Way!

$$\begin{array}{lr}
 & C_3C_2C_1C_0 \\
 & 0\ X_3X_2X_1 \\
- & 0\ Y_3Y_2Y_1 \\
\hline
C4 & S_4S_3S_2S_1
\end{array}$$

$$\begin{array}{llr} & & C_3C_2C_1C_0 \\ & & 0\ X_3X_2X_1 \\ + & & \text{2's-comp}(0\ Y_3Y_2Y_1) \\ \hline & \text{C4} & S_4S_3S_2S_1 \end{array}$$

$$\begin{array}{lrr} & & \boxed{C_3C_2C_1C_0} \\ & & 0\ X_3X_2X_1 \\ + & & \text{2's-comp}(0\ Y_3Y_2Y_1) \\ \hline & \boxed{C4} & S_4S_3S_2S_1 \end{array}$$

If S'_4 then $X \geq Y$

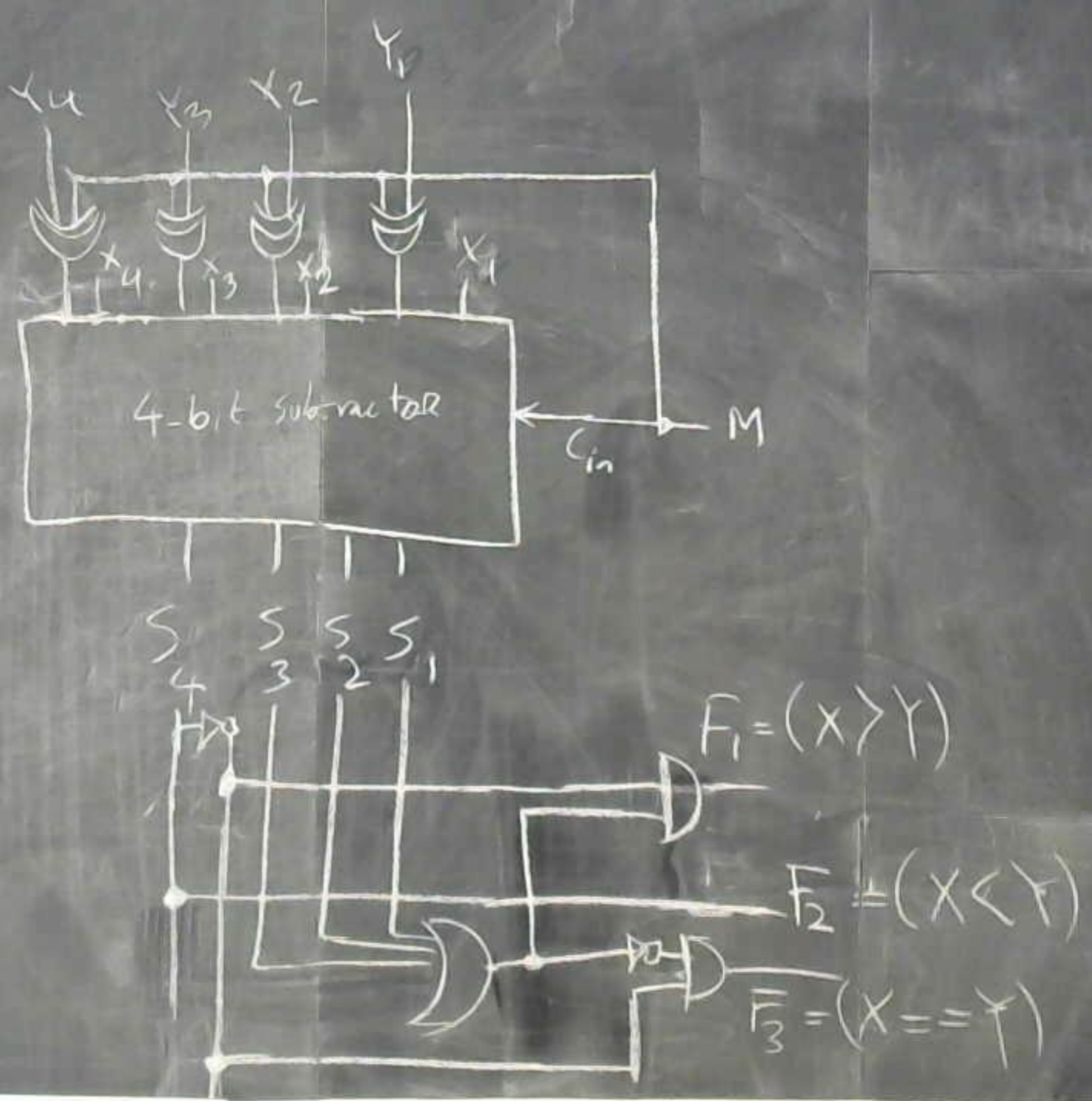
If S'_4 AND $(S_3+S_2+S_1)=1$ then $X > Y$

$$\begin{array}{lrr}
 & & C_3C_2C_1C_0 \\
 & & 0\ X_3X_2X_1 \\
+ & & \text{2's-comp}(0\ Y_3Y_2Y_1) \\
\hline
 & C4 & S_4S_3S_2S_1
\end{array}$$

If S_4 then $X < Y$

$$\begin{array}{lrr} & & \boxed{C_3C_2C_1C_0} \\ & & 0\ X_3X_2X_1 \\ + & & \text{2's-comp}(0\ Y_3Y_2Y_1) \\ \hline & \boxed{C4} & S_4S_3S_2S_1 \end{array}$$

If S'_4 AND $(S_3+S_2+S_1)'=1$ then $X == Y$

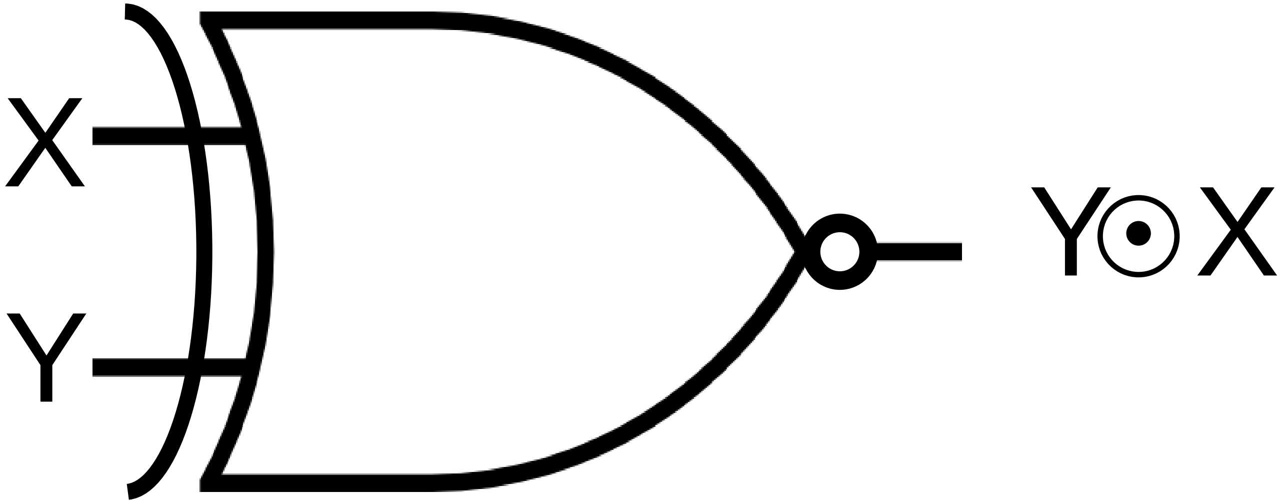

Y4
Y3
Y2
Y1
X4
X3
X2
X1
4-bit subtractor
Cin
M
S4
S3
S2
S1
F1 = (X>Y)
F2 = (X<Y)
F3 = (X==Y)

XNOR

Equality Gate

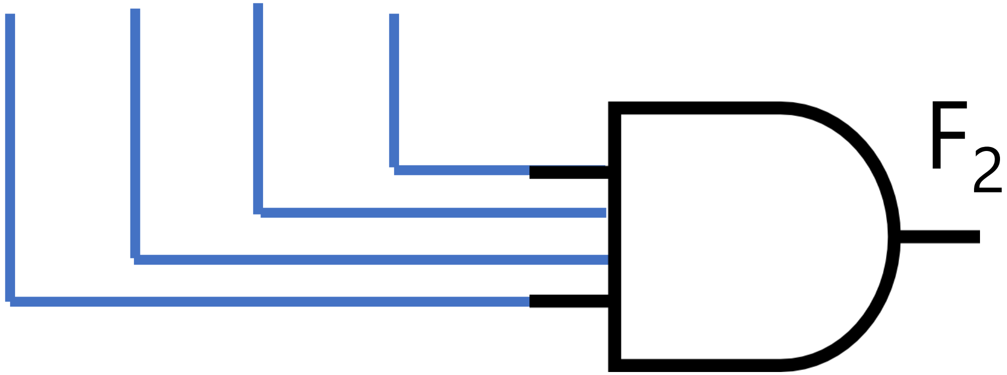
NOT Exclusive-OR (XNOR)

W04



Y	X	$F = F(Y,X) = Y'X'+YX = m_0+m_3$
0	0	1
0	1	0
1	0	0
1	1	1

$$\begin{array}{rcccc}
 & X_4 & X_3 & X_2 & X_1 \\
\odot & Y_4 & Y_3 & Y_2 & Y_1 \\
\hline
 & 1 & 1 & 1 & 1
\end{array}$$



$F_2=(X==Y)$

$$
\begin{array}{r}
X_4{=}1\ X_3X_2X_1 \\
Y_4{=}0\ Y_3Y_2Y_1 \\
\hline
X_4Y'_4 \rightarrow X > Y
\end{array}
$$

$$\begin{array}{r} X_4 = 0 \; X_3 X_2 X_1 \\ Y_4 = 1 \; Y_3 Y_2 Y_1 \\ \hline X'_4 Y_4 \rightarrow X < Y \end{array}$$

$$\begin{array}{r} X_4\ X_3X_2X_1 \\ Y_4\ Y_3Y_2Y_1 \\ \hline X_4 \odot Y_4 = 1 \end{array}$$

$$\begin{array}{c} X_4\ X_3=1\ X_2X_1 \\ Y_4\ Y_3=0\ Y_2Y_1 \\ \hline X_4 \odot Y_4=1 \\ X'_3Y_3 \rightarrow X > Y \end{array}$$

$$X_4\ X_3{=}0\ X_2X_1$$

$$Y_4\ Y_3{=}1\ Y_2Y_1$$

$$X_4 \odot Y_4 = 1$$

$$X_3Y'_3 \rightarrow X < Y$$

$$
\begin{aligned}
F1=(X>Y)= {} & X_4Y'_4+ \\
& (X_4\odot Y_4)X_3Y'_3+ \\
& (X_4\odot Y_4)(X_3\odot Y_3)X_2Y'_2+ \\
& (X_4\odot Y_4)(X_3\odot Y_3)(X_2\odot Y_2)X_1Y'_1
\end{aligned}
$$

$$
\begin{aligned}
F1=(X<Y)= \; & X'_4Y_4+ \\
& (X_4\odot Y_4)X'_3Y_3+ \\
& (X_4\odot Y_4)(X_3\odot Y_3)X'_2Y_2+ \\
& (X_4\odot Y_4)(X_3\odot Y_3)(X_2\odot Y_2)X'_1Y_1
\end{aligned}
$$

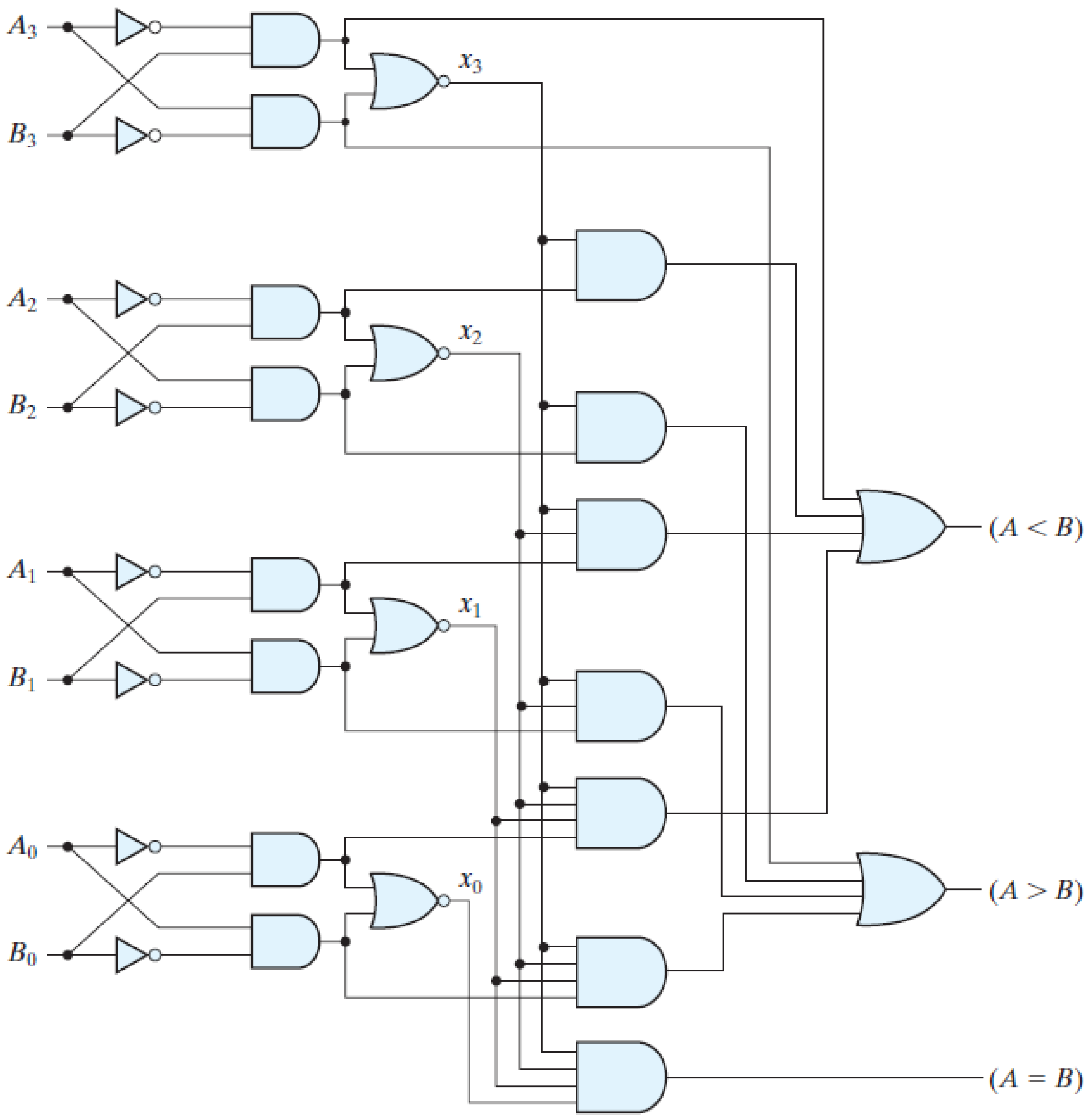


FIGURE 4.17
Four-bit magnitude comparator

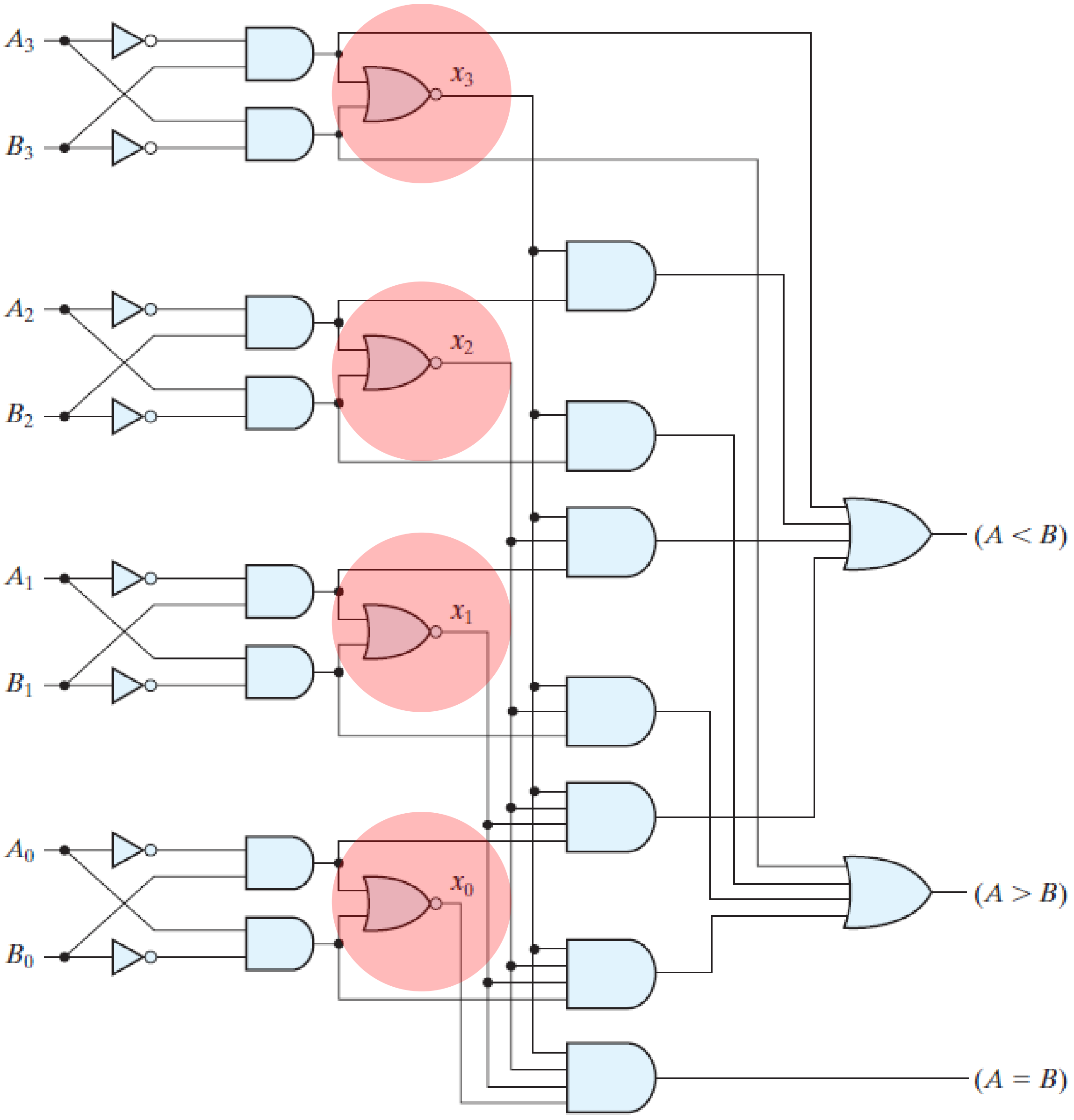
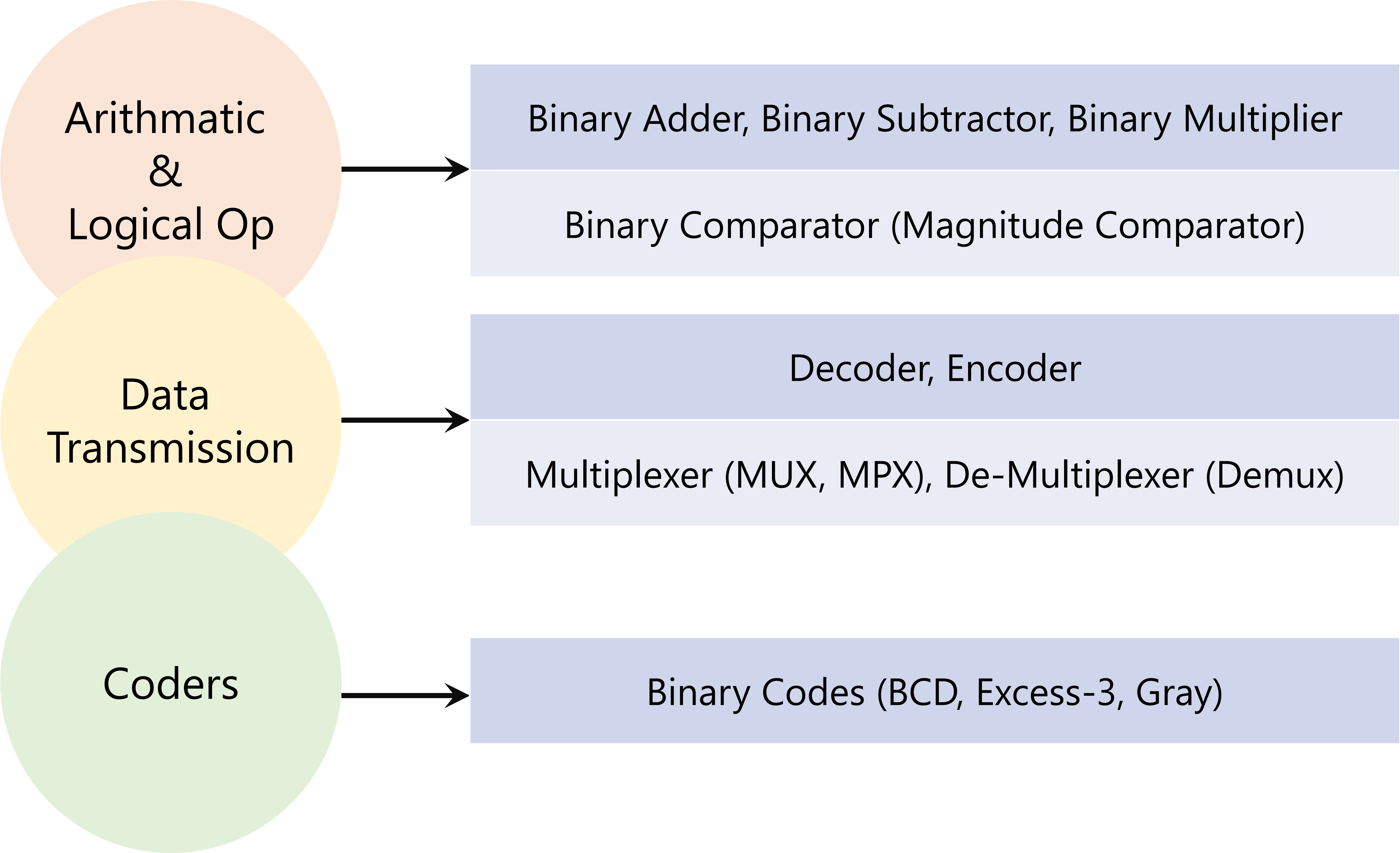


FIGURE 4.17
Four-bit magnitude comparator



Arithmatic
&
Logical Op

Binary Adder, Binary Subtractor, Binary Multiplier

Binary Comparator (Magnitude Comparator)

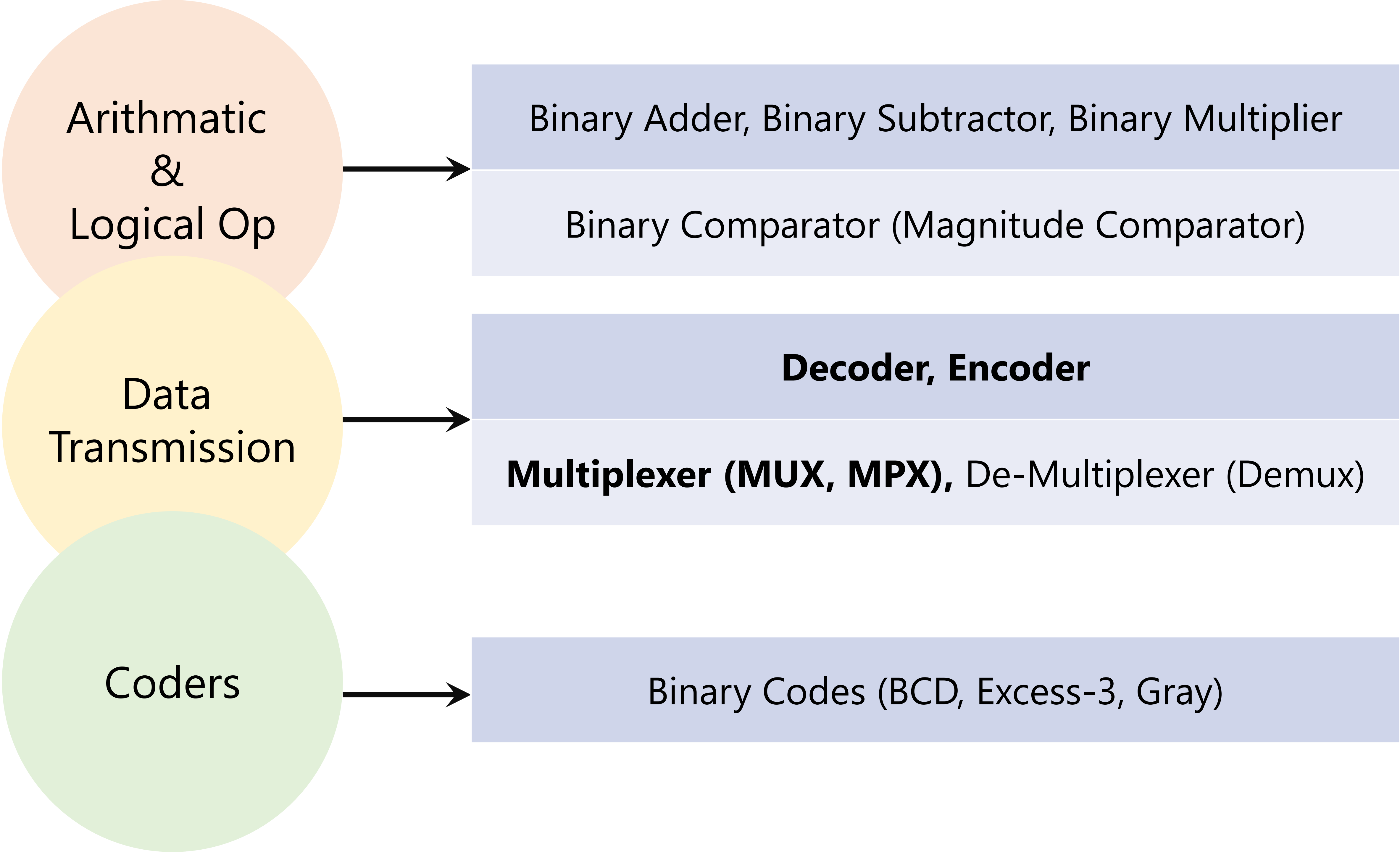
Data
Transmission

Decoder, Encoder

Multiplexer (MUX, MPX), De-Multiplexer (Demux)

Coders

Binary Codes (BCD, Excess-3, Gray)



Arithmatic
&
Logical Op

Binary Adder, Binary Subtractor, Binary Multiplier

Binary Comparator (Magnitude Comparator)

Data
Transmission

Decoder, Encoder

Multiplexer (MUX, MPX), De-Multiplexer (Demux)

Coders

Binary Codes (BCD, Excess-3, Gray)