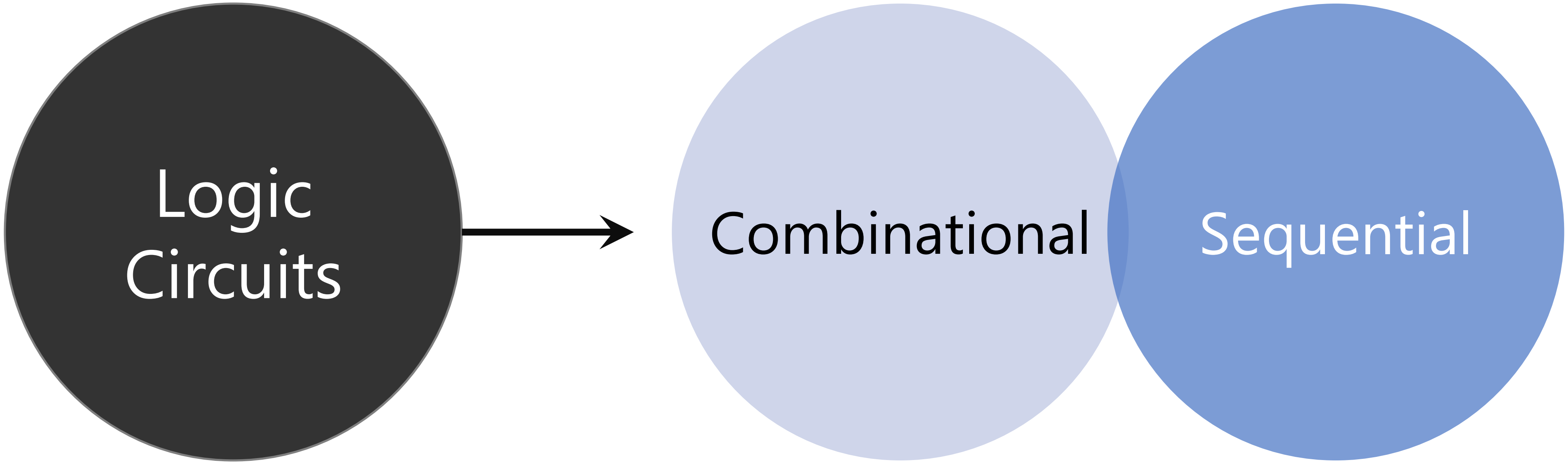

Logic
Circuits
Combinational
Sequential

Chapter 4 Combinational Logic

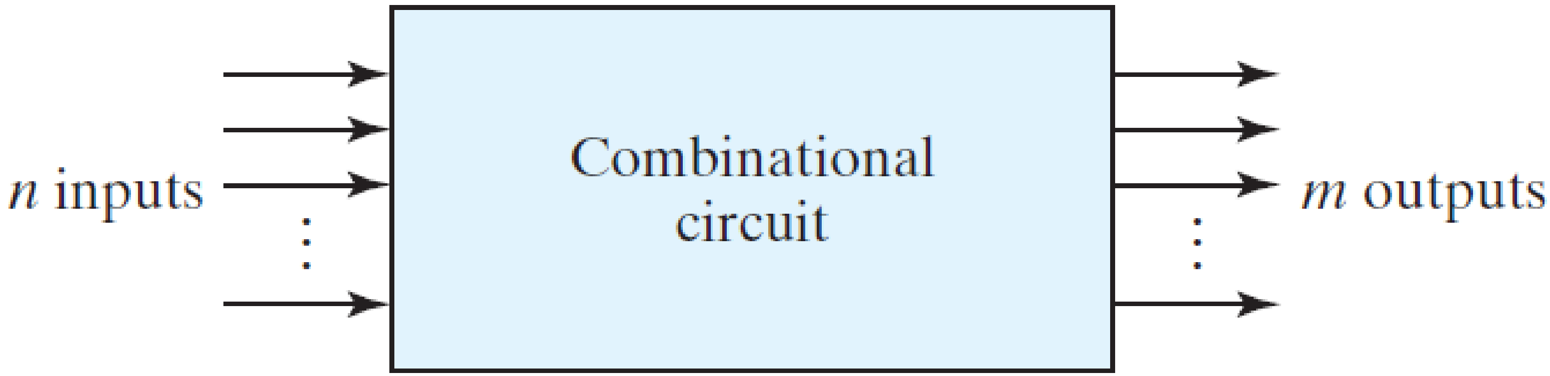


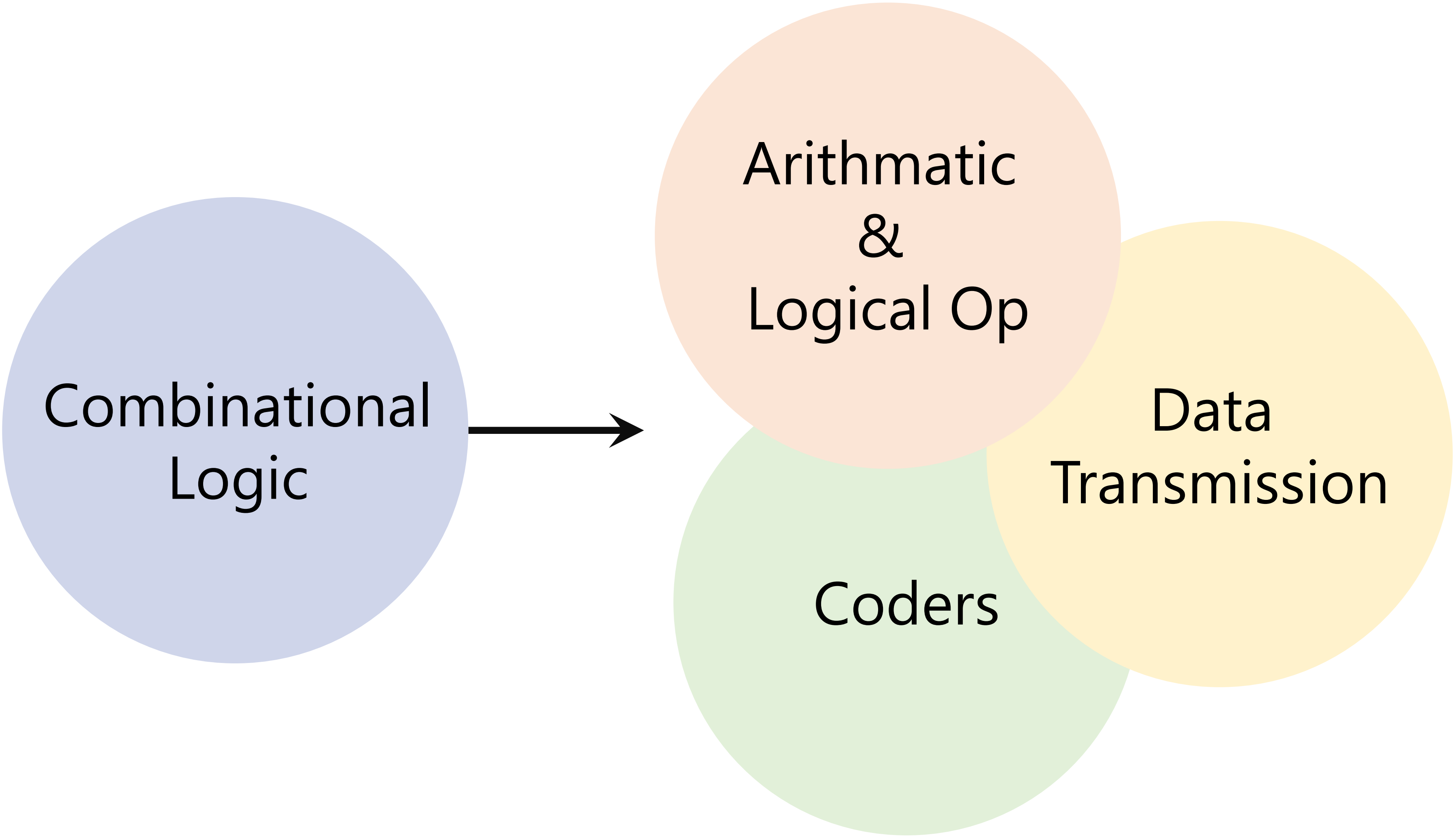
FIGURE 4.1
Block diagram of combinational circuit

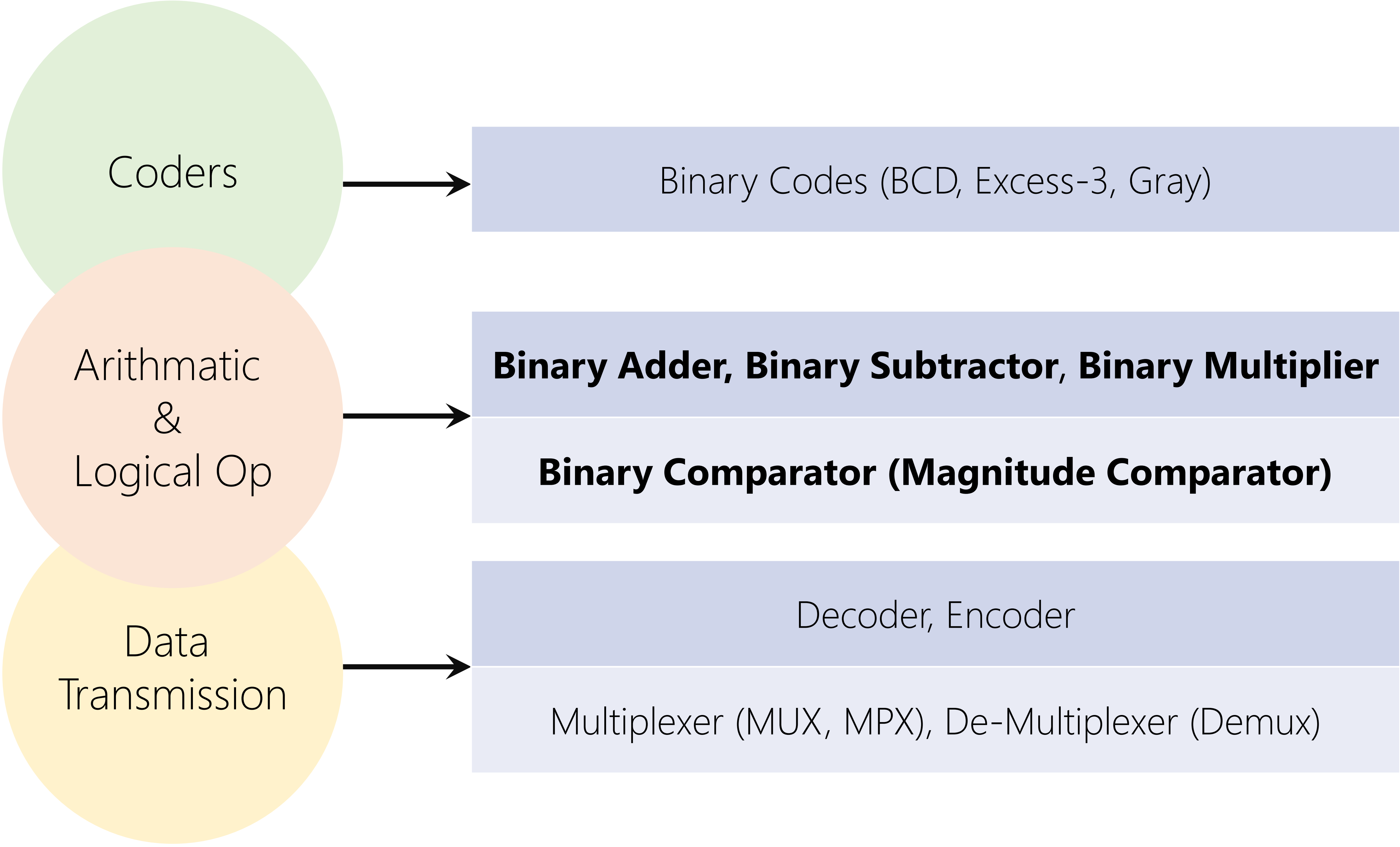
Combinational Logic

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.

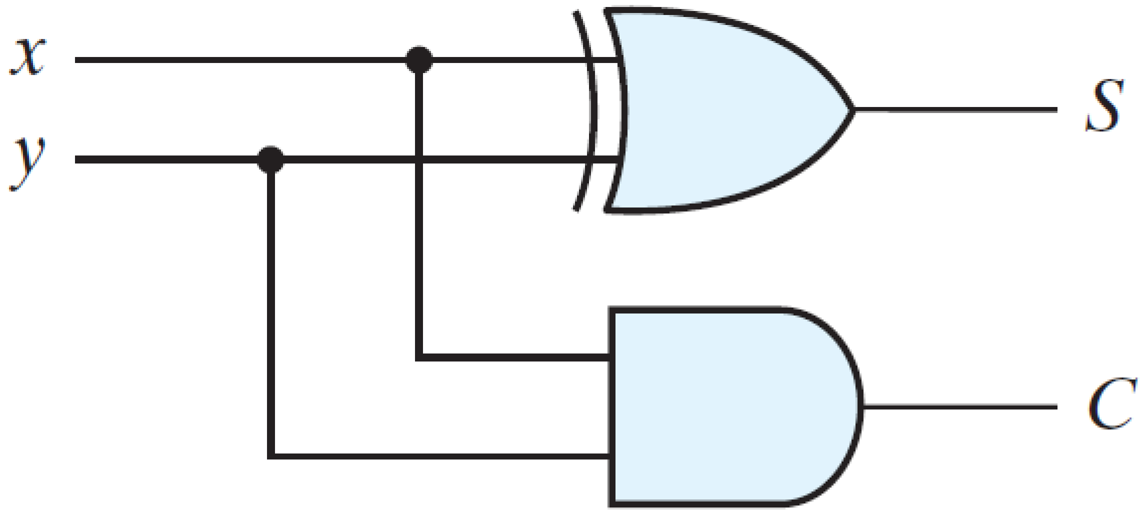

Combinational
Logic
Arithmatic
&
Logical Op
Data
Transmission
Coders


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)

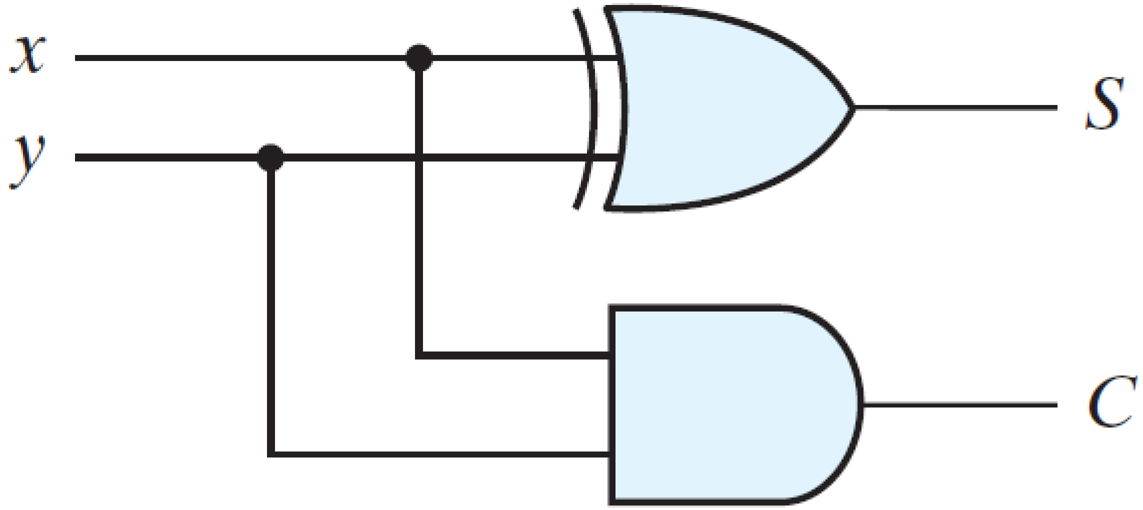
Binary Adder

Design a logic circuit that adds two binary digits (bit).

Y	X	$F_2 = C(Y,X) = YX$	$F_1 = S(Y,X) = Y'X + YX'$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

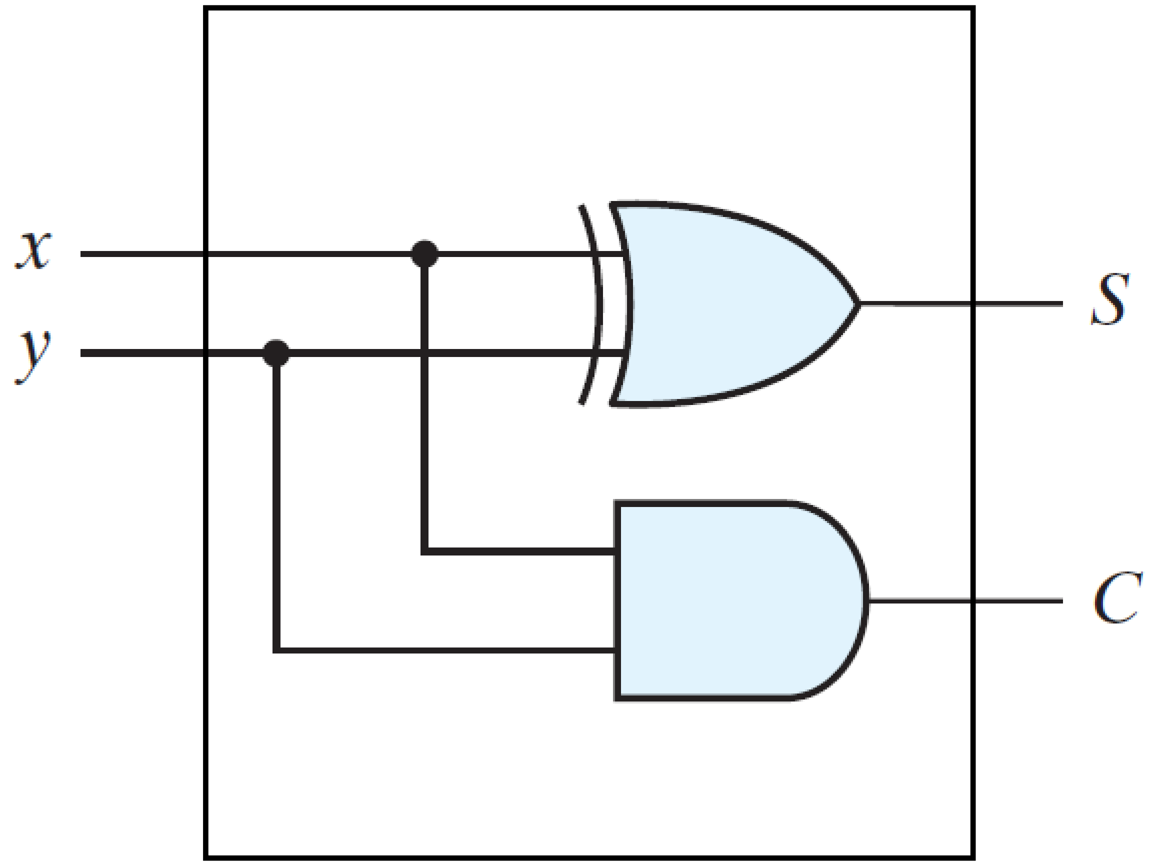


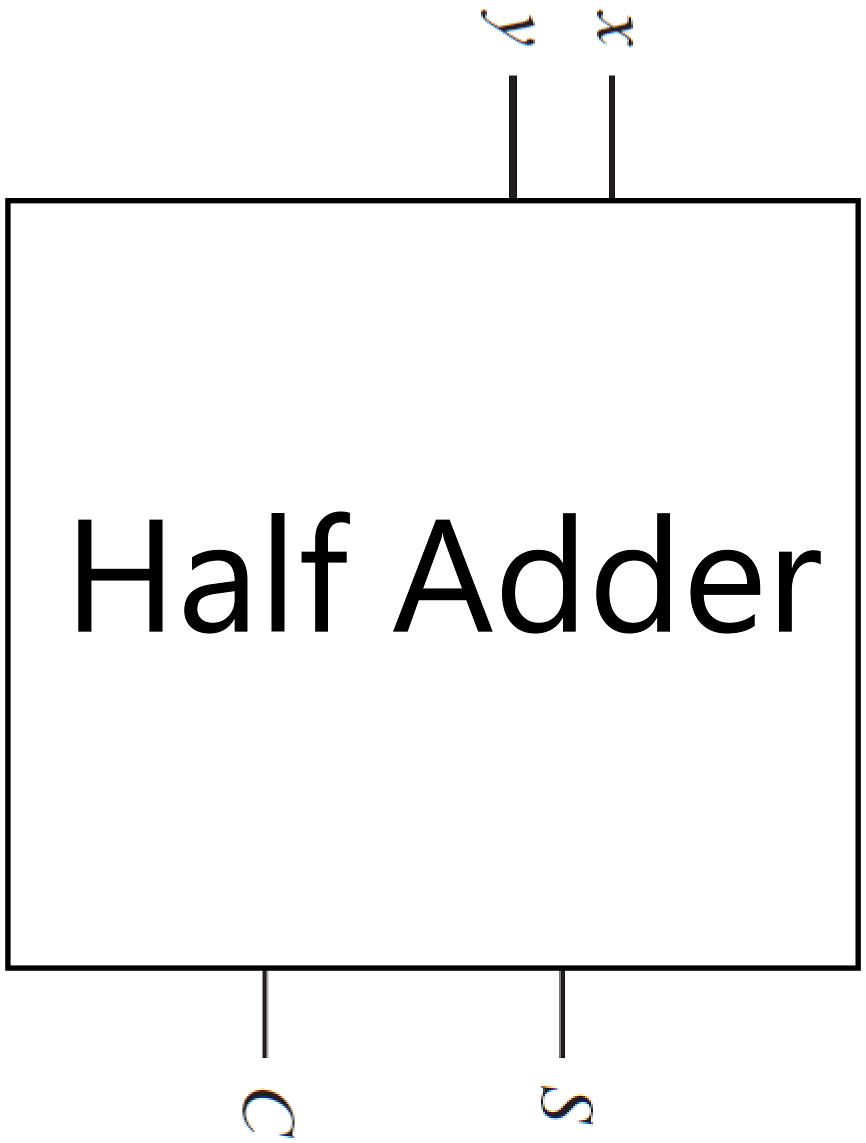
$$S = x \oplus y$$
$$C = xy$$



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

Half Adder: Just 2 bits: X+Y


x
y
S
C


y
x
Half Adder
C
S

Design a logic circuit that adds two binary numbers!

$$
\begin{array}{cr}
 & X_2X_1 \\
+ & Y_2Y_1 \\
\hline
C & S_2S_1
\end{array}
$$

Y_2	Y_1	X_2	X_1	$C(Y_1,Y_2,X_2,X_1)$	$S_2(Y_1,Y_2,X_2,X_1)$	$S_1(Y_1,Y_2,X_2,X_1)$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

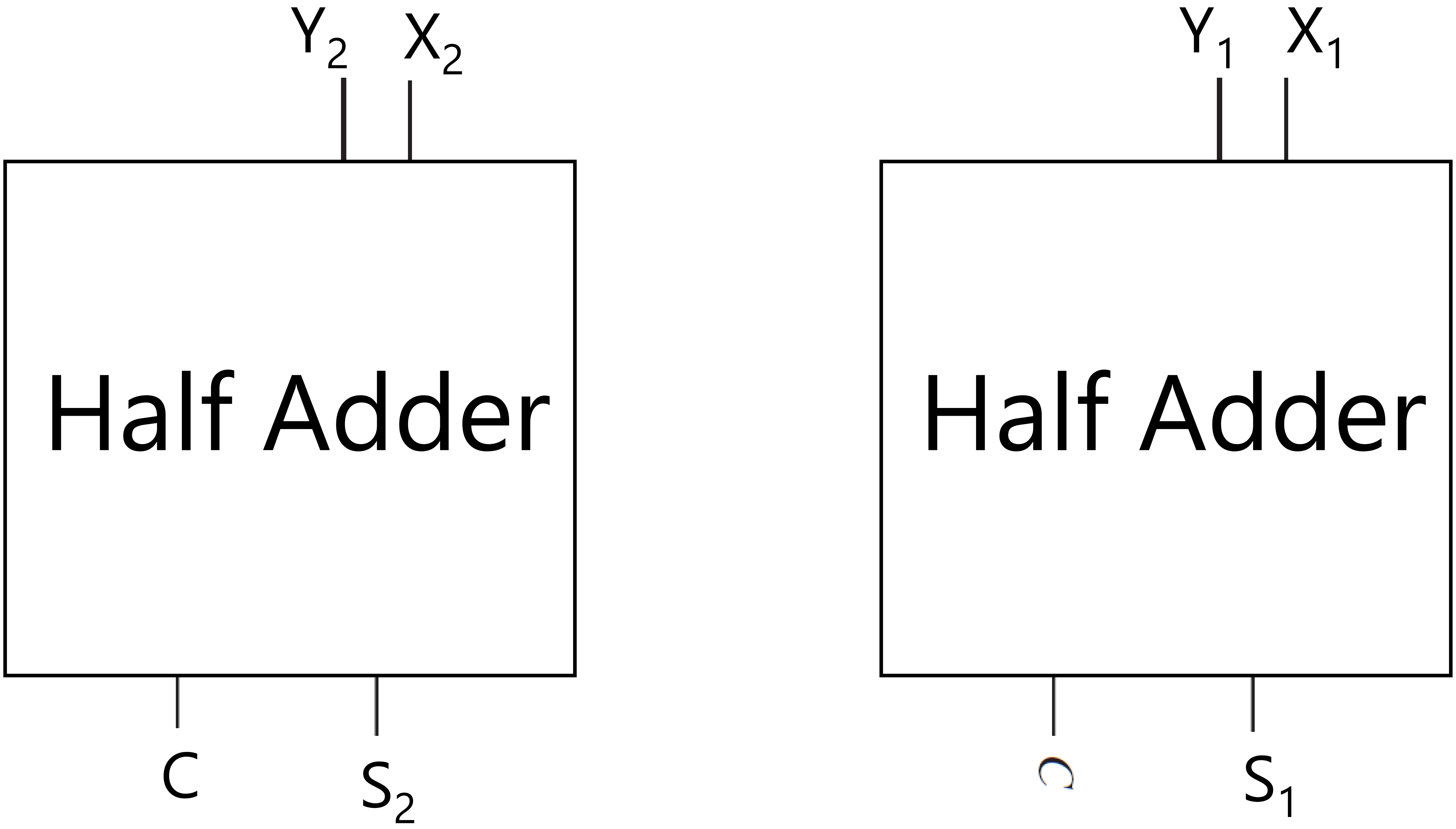
Y_2	Y_1	X_2	X_1	$C(Y_1,Y_2,X_2,X_1)$	$S_2(Y_1,Y_2,X_2,X_1)$	$S_1(Y_1,Y_2,X_2,X_1)$
0	0	0	0	0	0	0
0	0	0	1	0	[illegible]	1
0	0	1	0	0	[illegible]	0
0	0	1	1	0	[illegible]	1
0	1	0	0	0	[illegible]	1
0	1	0	1	0	[illegible]	0
0	1	1	0	0	[illegible]	1
0	1	1	1	[illegible]	[illegible]	[illegible]
1	0	0	0	[illegible]	[illegible]	[illegible]
1	0	0	1	0	[illegible]	1
1	0	1	0	1	[illegible]	0
1	0	1	1	1	[illegible]	1
1	1	0	0	0	[illegible]	1
1	1	0	1	1	[illegible]	0
1	1	1	0	1	[illegible]	1
1	1	1	1	1	1	0

Wait a sec!

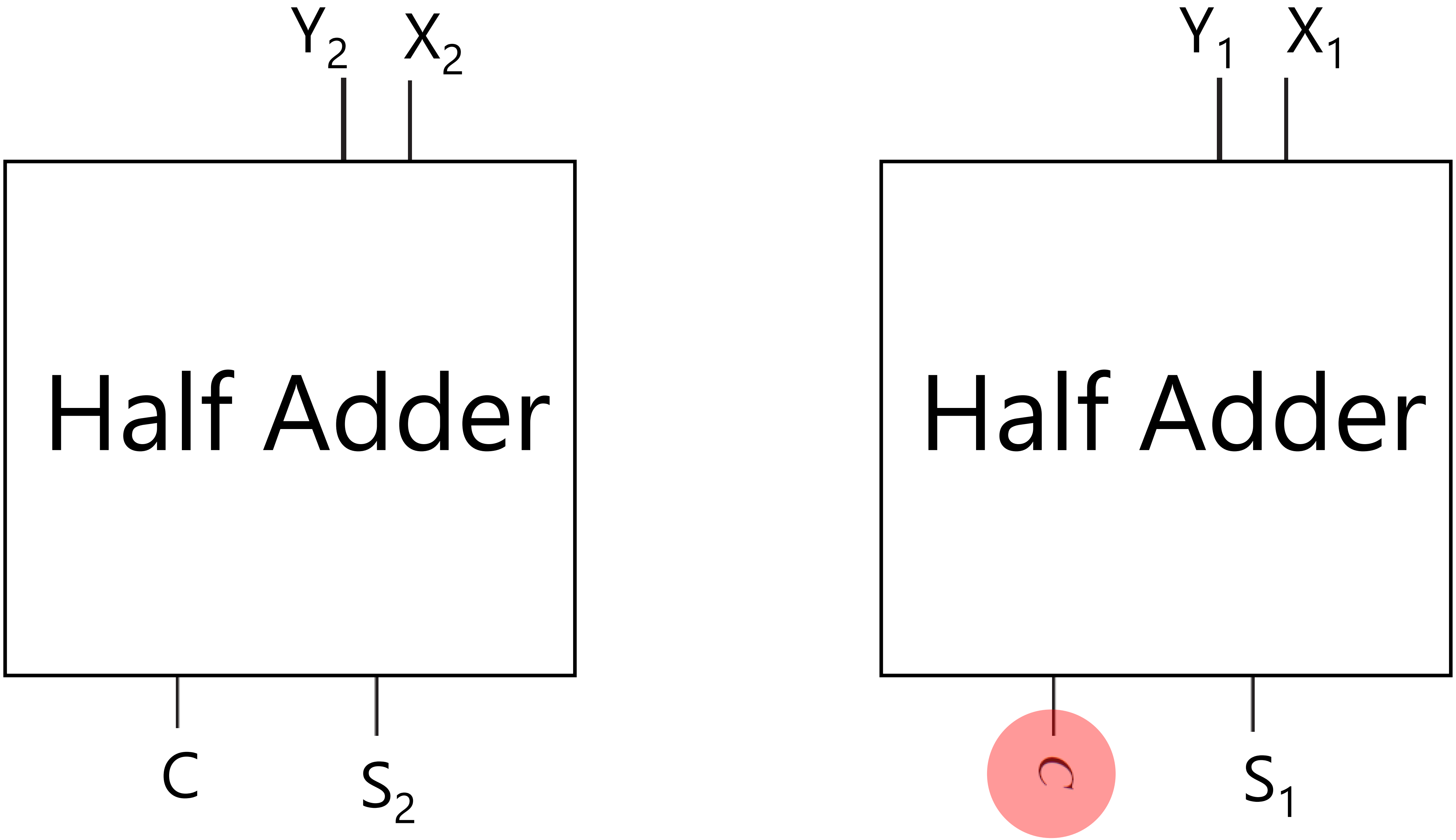
Can we re-use the half adder?

Half adder for adding 2 bits.
How about having 2 half adders for adding 2 × 2 bits?

$$\begin{array}{cr} & X_2X_1 \\ + & Y_2Y_1 \\ \hline \boxed{C} & S_2S_1 \end{array}$$

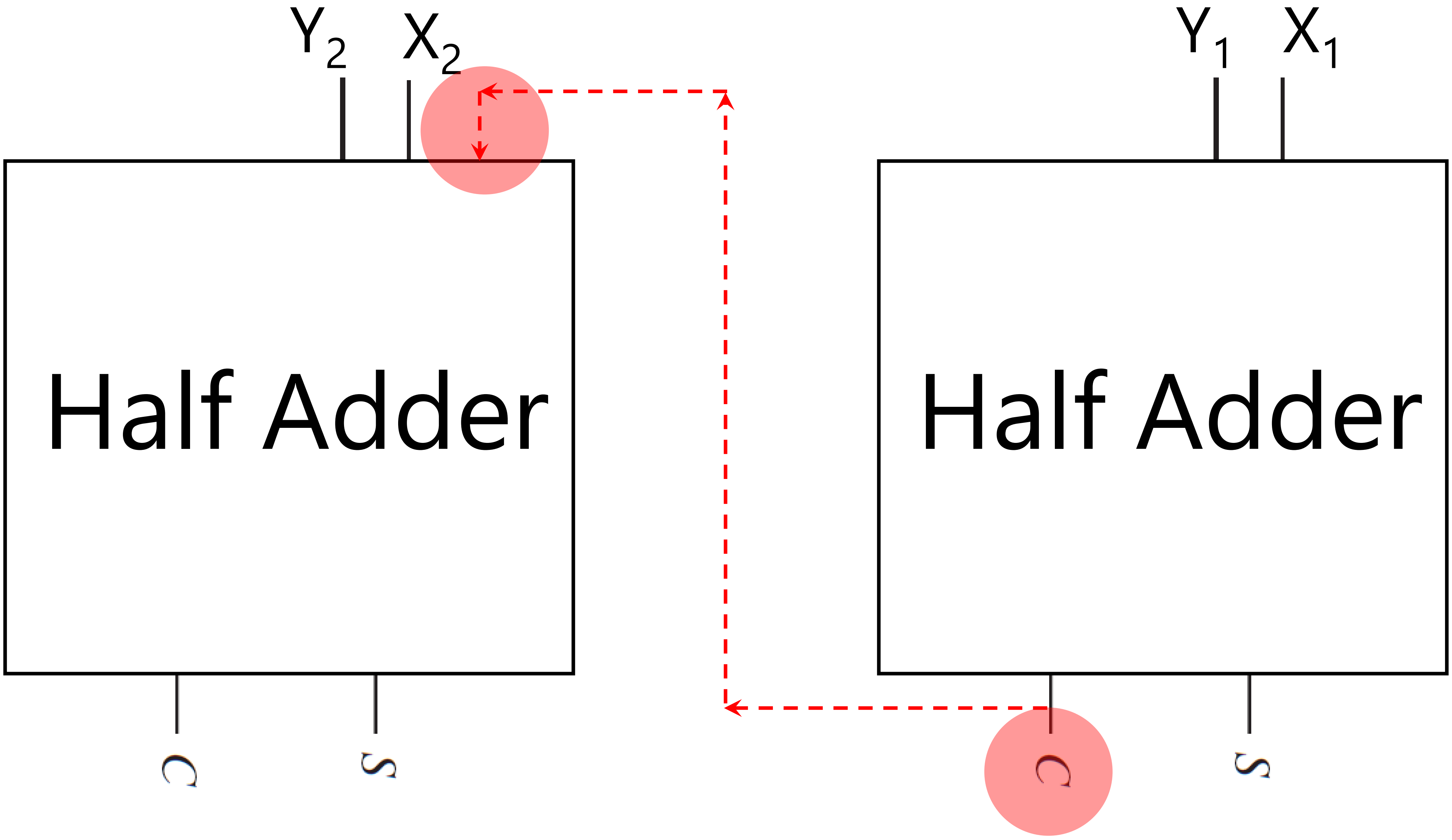


Y2 X2
Half Adder
C S2
Y1 X1
Half Adder
S1

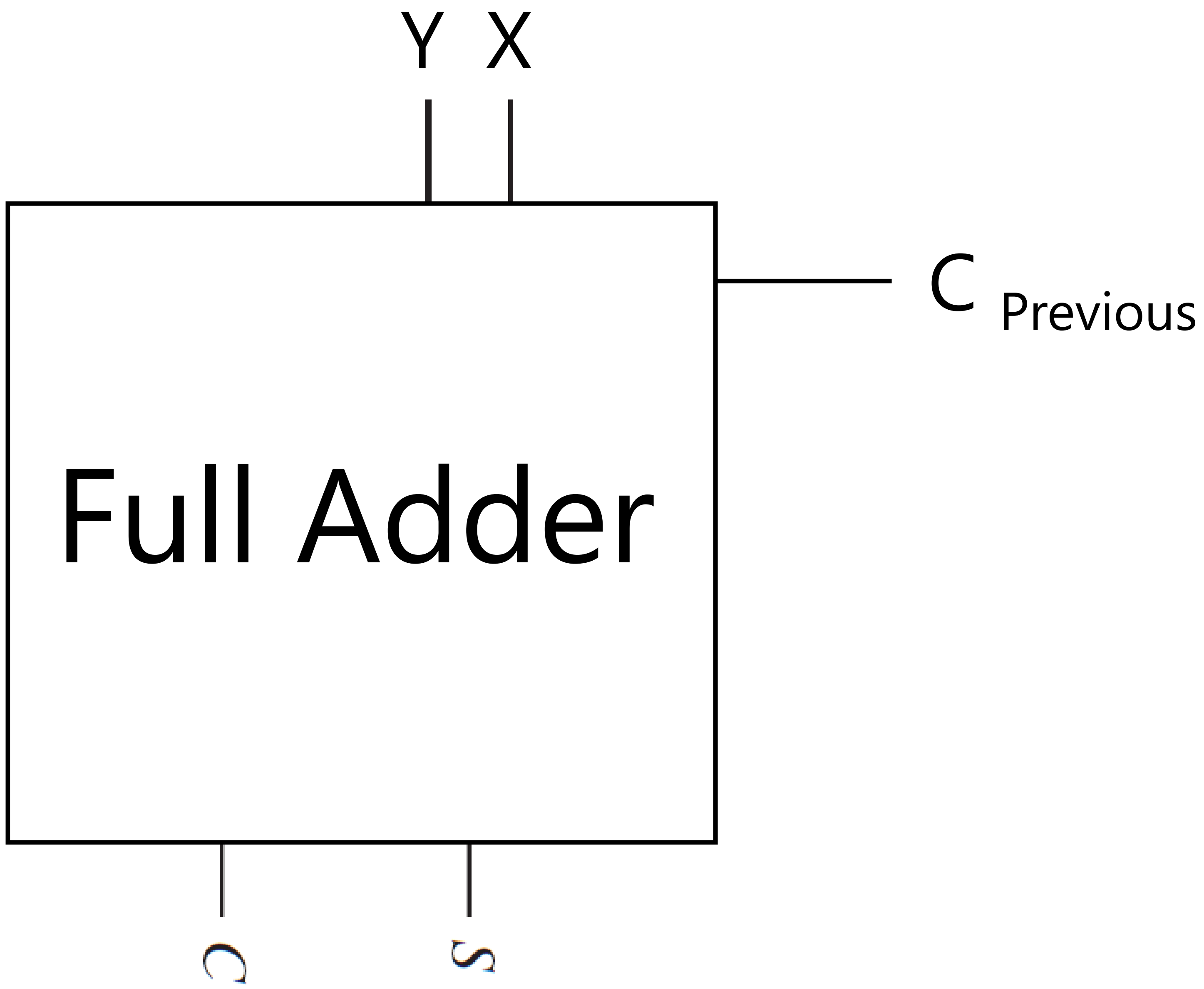


Y2
X2
Half Adder
C
S2
Y1
X1
Half Adder
C
S1

$$\begin{array}{rcc} \boxed{C=1} & & \\ & 0 & 1 \\ + & 0 & 1 \\ \hline \boxed{C=0} & 1 & 0 \end{array}$$


Y_2
X_2
Half Adder
C
S
Y_1
X_1
Half Adder
C
S

$$\begin{array}{cc} & \boxed{C_p} \\ & X \\ + & Y \\ \hline \boxed{C} & S \end{array}$$


Y X
C Previous
Full Adder
C
S

Design a logic circuit that adds two binary digits (bit) and a carry bit.

C_p	Y	X	$C=\sum m(3,5,6,7)$	$S=\sum m(1,2,4,7)$
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

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

$$C=\sum m(3,5,6,7)$$

C_p \ YX	00	01	11	10
0	0 m_0	0 m_1	1 m_3	0 m_2
1	0 m_4	1 m_5	1 m_7	1 m_6

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

$$S=C'_pY'X+C'_pYX'+C_pY'X'+C_pYX$$

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

$$
\begin{aligned}
S &= C'_pY'X+C'_pYX'+C_pY'X'+C_pYX \\
&= C'_p(Y'X+YX')+C_p(Y'X'+YX)
\end{aligned}
$$

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 (m_0)	1 (m_1)	0 (m_3)	1 (m_2)
1	1 (m_4)	0 (m_5)	1 (m_7)	0 (m_6)

$$\begin{aligned} S &= C'_pY'X+C'_pYX'+C_pY'X'+C_pYX \\ &= C'_p(Y'X+YX')+C_p(Y'X'+YX) \\ &= C'_p(X\oplus Y)+C_p(Y'X'+YX) \end{aligned}$$

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

$$\begin{aligned} S &= C'_pY'X+C'_pYX'+C_pY'X'+C_pYX \\ &= C'_p(Y'X+YX')+C_p(Y'X'+YX) \\ &= C'_p(X\oplus Y)+C_p(Y'X'+YX) \\ &= C'_p(X\oplus Y)+C_p(X\odot Y) \end{aligned}$$

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

$$\begin{aligned} S &= C'_pY'X+C'_pYX'+C_pY'X'+C_pYX \\ &= C'_p(Y'X+YX')+C_p(Y'X'+YX) \\ &= C'_p(X\oplus Y)+C_p(Y'X'+YX) \\ &= C'_p(X\oplus Y)+C_p(X\odot Y) \\ &= C'_p(X\oplus Y)+C_p(X\oplus Y)' \end{aligned}$$

$$\begin{aligned} (X\oplus Y)' &= (Y'X+YX')' \\ &= (Y'X)'(YX')' \\ &= (Y+X')(Y'+X) \\ &= YY'+YX+X'Y'+X'X \\ &= 0+YX+X'Y'+0 \\ &= YX+X'Y' \\ &= Y\odot X \end{aligned}$$

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 (m_0)	1 (m_1)	0 (m_3)	1 (m_2)
1	1 (m_4)	0 (m_5)	1 (m_7)	0 (m_6)

$$\begin{aligned}
S&=C'_pY'X+C'_pYX'+C_pY'X'+C_pYX\\
&=C'_p(Y'X+YX')+C_p(Y'X'+YX)\\
&=C'_p(X\oplus Y)+C_p(Y'X'+YX)\\
&=C'_p(X\oplus Y)+C_p(X\odot Y)\\
&=C'_p(X\oplus Y)+C_p(X\oplus Y)'\\
&=C'_p\alpha+C_p\alpha'
\end{aligned}$$

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

$$
\begin{aligned}
S &= C'_pY'X + C'_pYX' + C_pY'X' + C_pYX \\
&= C'_p(Y'X + YX') + C_p(Y'X' + YX) \\
&= C'_p(X \oplus Y) + C_p(Y'X' + YX) \\
&= C'_p(X \oplus Y) + C_p(X \odot Y) \\
&= C'_p(X \oplus Y) + C_p(X \oplus Y)' \\
&= C'_p\alpha + C_p\alpha' \\
&= C_p \oplus \alpha
\end{aligned}
$$

$$S=\sum m(1,2,4,7)$$

C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

$$\begin{aligned}
S &= C'_pY'X+C'_pYX'+C_pY'X'+C_pYX \\
&= C'_p(Y'X+YX')+C_p(Y'X'+YX) \\
&= C'_p(X\oplus Y)+C_p(Y'X'+YX) \\
&= C'_p(X\oplus Y)+C_p(X\odot Y) \\
&= C'_p(X\oplus Y)+C_p(X\oplus Y)' \\
&= C'_p\alpha+C_p\alpha' \\
&= C_p\oplus\alpha \\
&= C_p\oplus(X\oplus Y)
\end{aligned}$$

$$S = \sum m(1,2,4,7)$$

$$S = C_p \oplus (X \oplus Y)$$

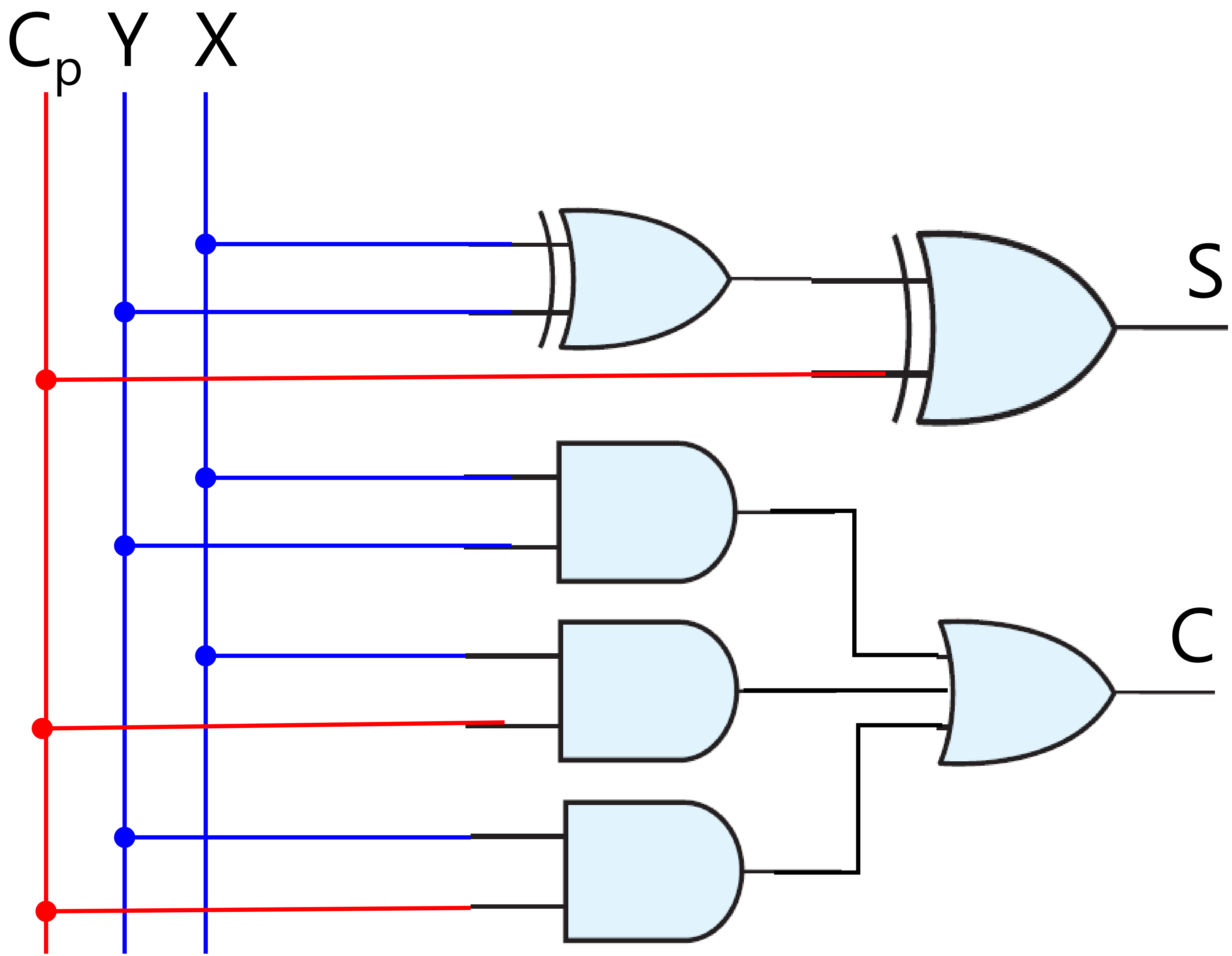
C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

$\oplus$ is associative, we can drop ( ). But let's keep them!

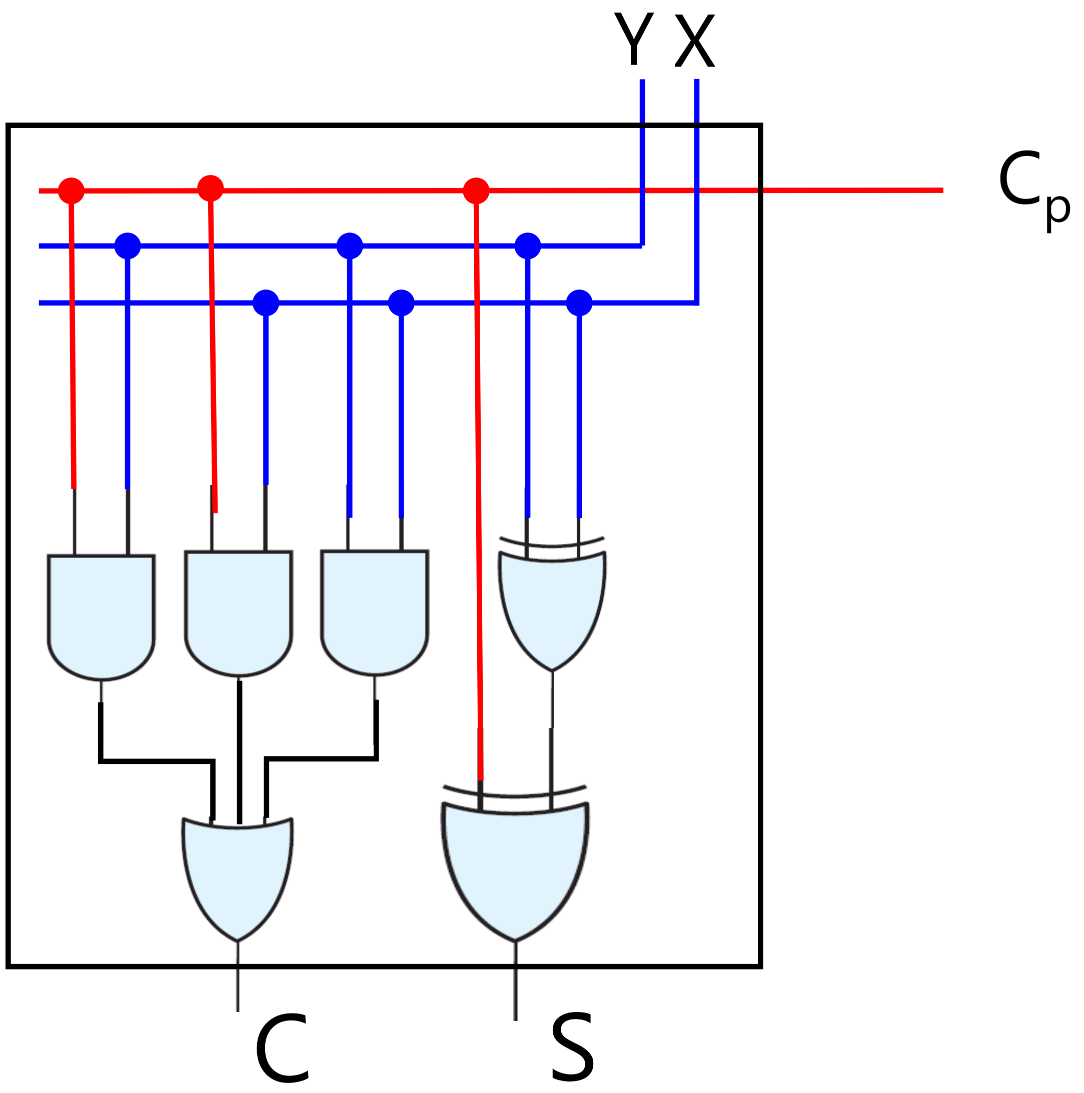
$$C = \sum m(3,5,6,7)$$

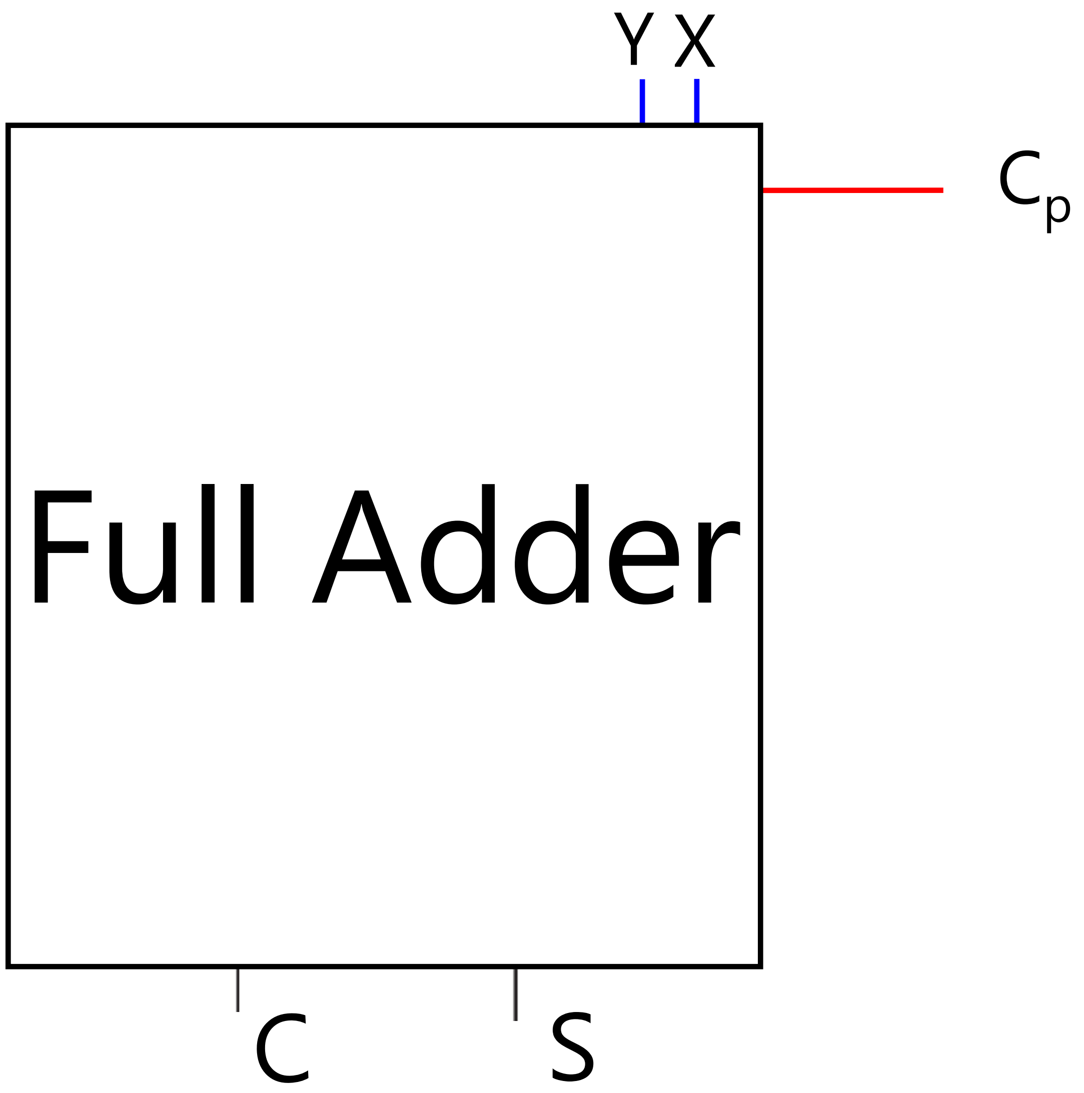
C_p \ YX	00	01	11	10
0	0 m_0	0 m_1	1 m_3	0 m_2
1	0 m_4	1 m_5	1 m_7	1 m_6

$$C = YX + C_pX + C_pY$$

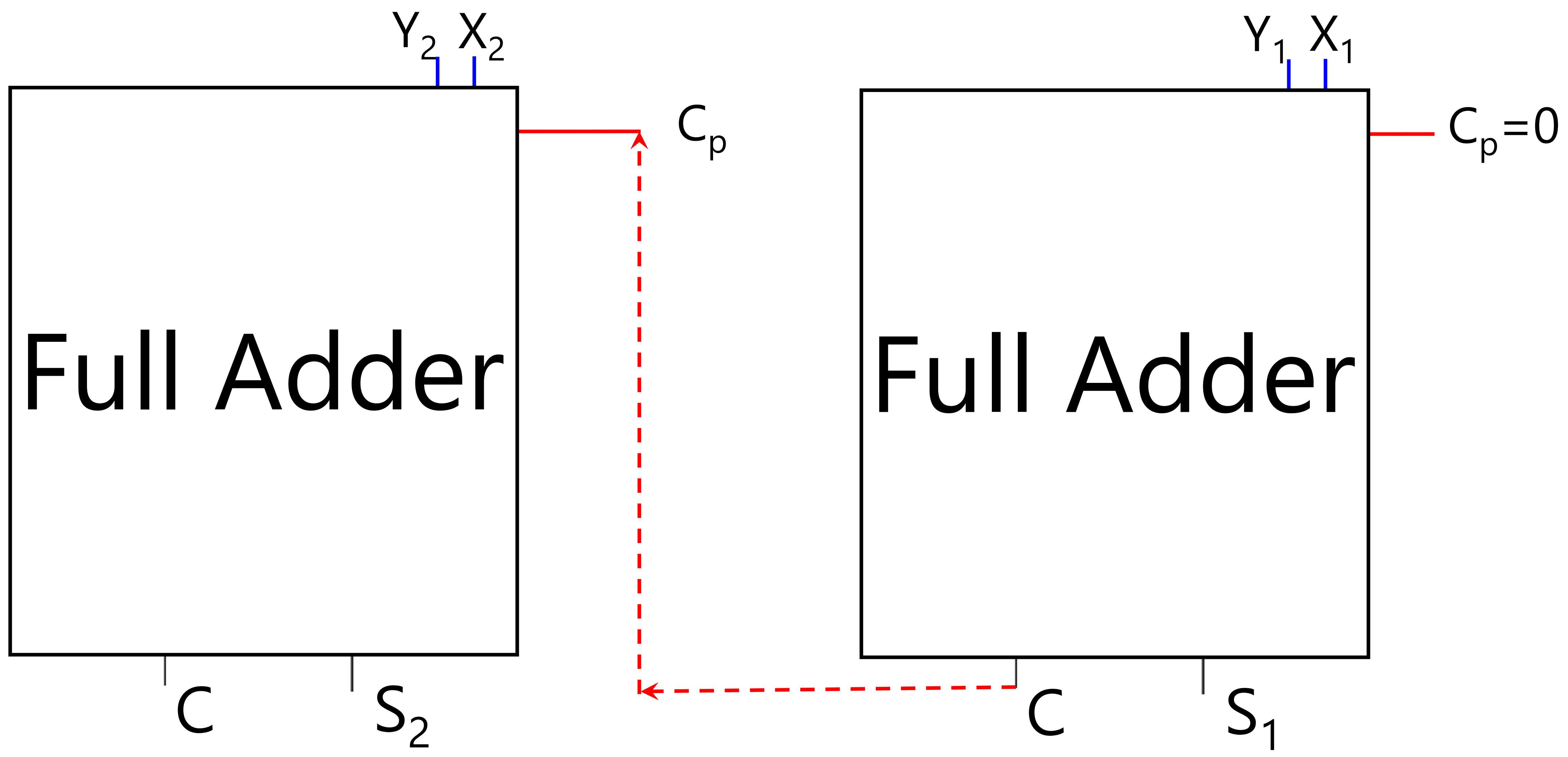


C_p
Y
X
S
C


Y X
C_p
C
S


Y
X
C_p
Full Adder
C
S

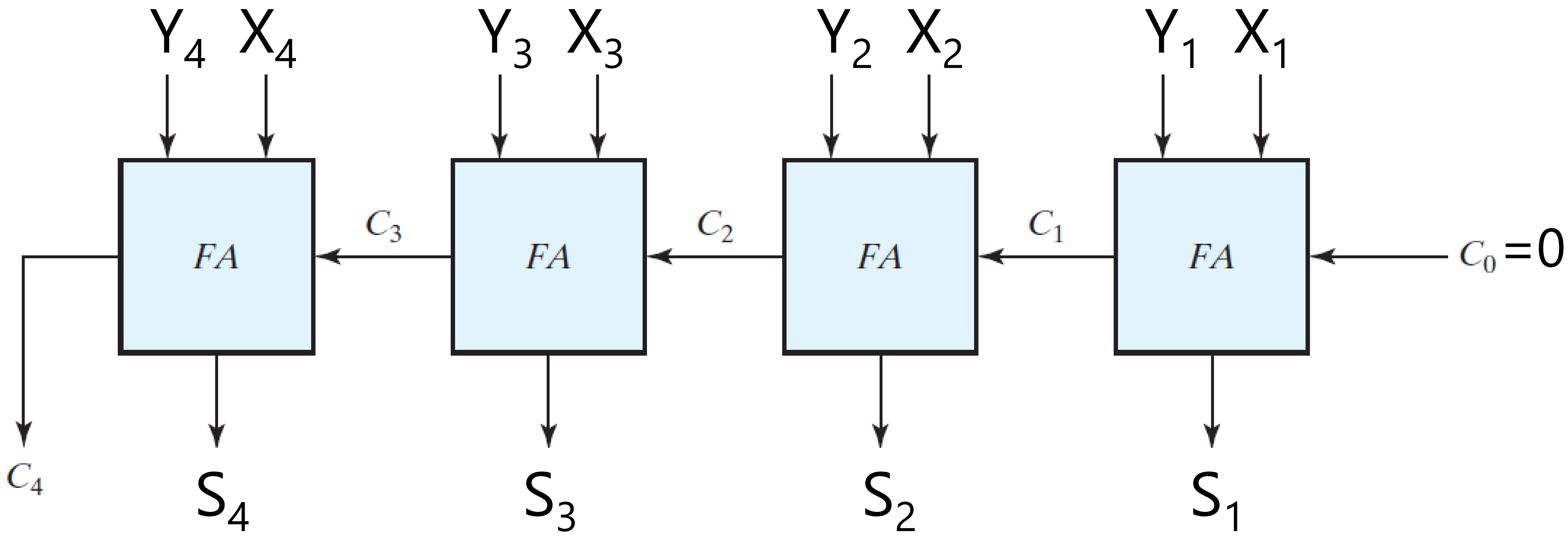
$$\begin{array}{r} X_2X_1 \\ +\quad Y_2Y_1 \\ \hline \boxed{C}\ S_2S_1 \end{array} \quad \Rightarrow \quad \begin{array}{r} \boxed{C}\,\boxed{0} \\ X_2X_1 \\ +\quad Y_2Y_1 \\ \hline \boxed{C}\ S_2S_1 \end{array}$$


Full Adder
Full Adder
Y_2
X_2
Y_1
X_1
C_p
$C_p=0$
C
S_2
C
S_1

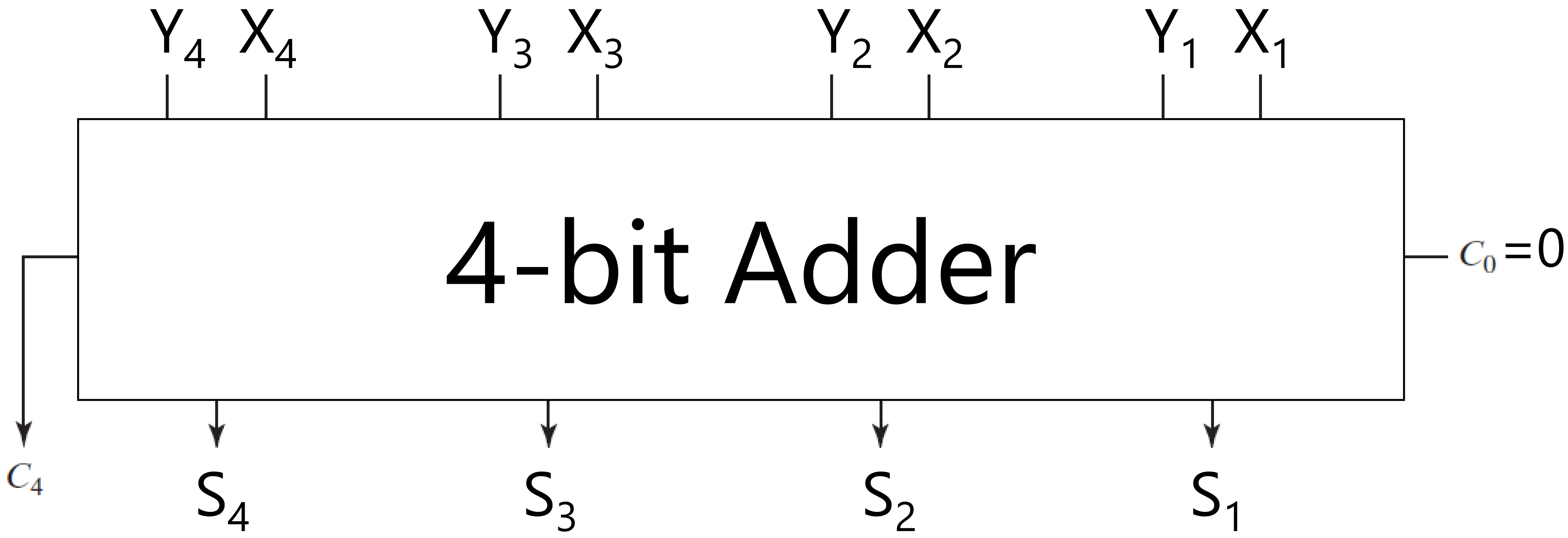
Design a logic circuit that adds two binary numbers!

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

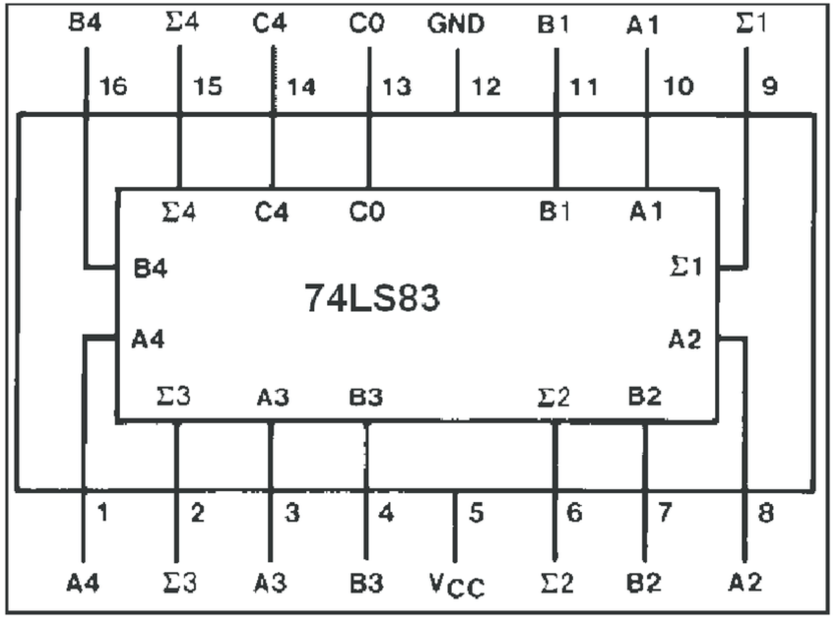
$C_0=0$



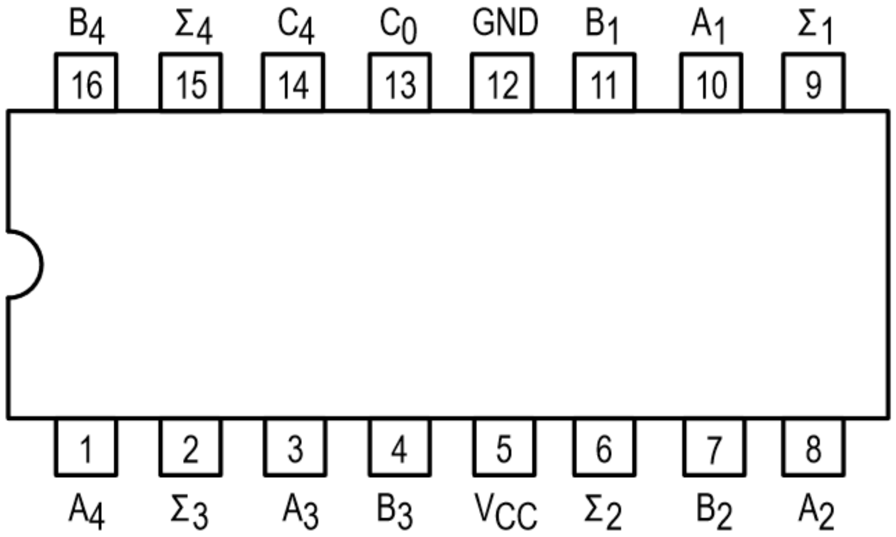
Y4 X4
Y3 X3
Y2 X2
Y1 X1
FA
FA
FA
FA
C3
C2
C1
C0 =0
C4
S4
S3
S2
S1



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

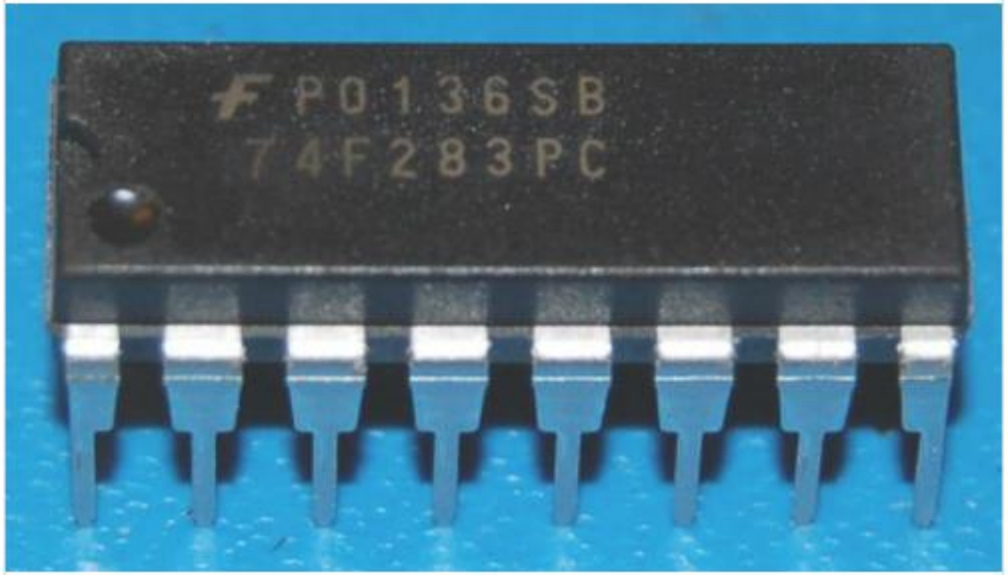


74LS83 pinout



Vcc	5.5V max, 5V Typical
$A_1 - A_4$	Operand A Inputs
$B_1 - B_4$	Operand B Inputs
C_0	Carry Input
$\Sigma_1 - \Sigma_4$	Sum Outputs (Note b)
C_4	Carry Output (Note b)

74283 - 74F283N 4-Bit Binary Full Adder w/ Fast Carry, DIP-16



C $6.55

+ C $4.89 Shipping

Get it by **Tue, Nov 10 - Tue, Nov 17** from Havre-aux-Maisons, QC, Canada

- **New** condition
- 30 day returns - Buyer pays return shipping |

Return policy

Read seller's description

See details

ebay MONEY BACK GUARANTEE

Qty: 1

Buy It Now

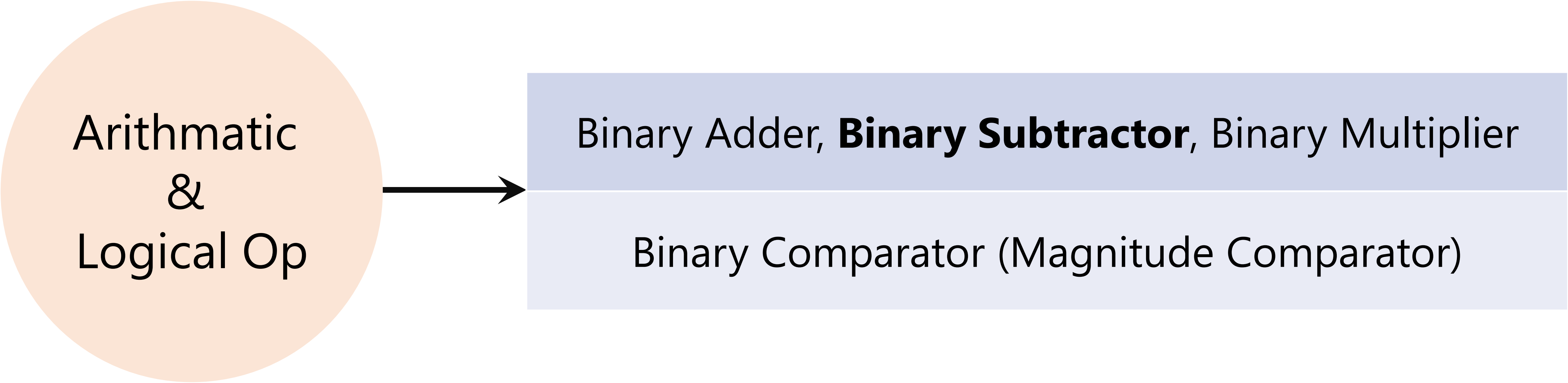
Add to cart

Watch

Sold by
vedge23 (3476)
100.0% Positive feedback
Contact seller

Binary Adder

Does it matter we have signed or unsigned binary numbers?
Justify your answer.


Arithmatic
&
Logical Op
Binary Adder, **Binary Subtractor**, Binary Multiplier
Binary Comparator (Magnitude Comparator)

Binary Subtractor

Signed-2’s-Complement

X – Y

Subtraction in Signed-2’s-Complement

$$X_n X_{n-1} \ldots X_2 X_1 - Y_n Y_{n-1} \ldots Y_2 Y_1$$

Subtraction in Signed-2's-Complement

X + 2’s-comp(Y)

Subtraction in Signed-2’s-Complement

X + 1’s-comp(Y) + 1

Subtraction in Signed-2’s-Complement

bitwise

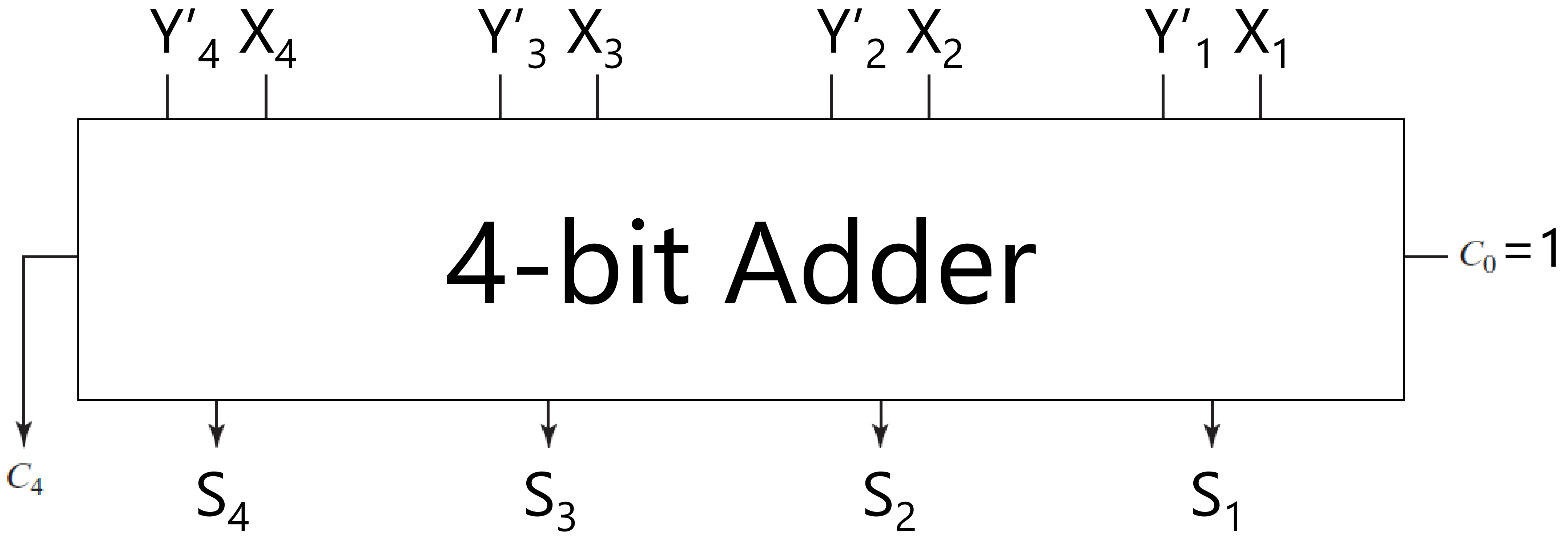
$$X + Y' + 1$$

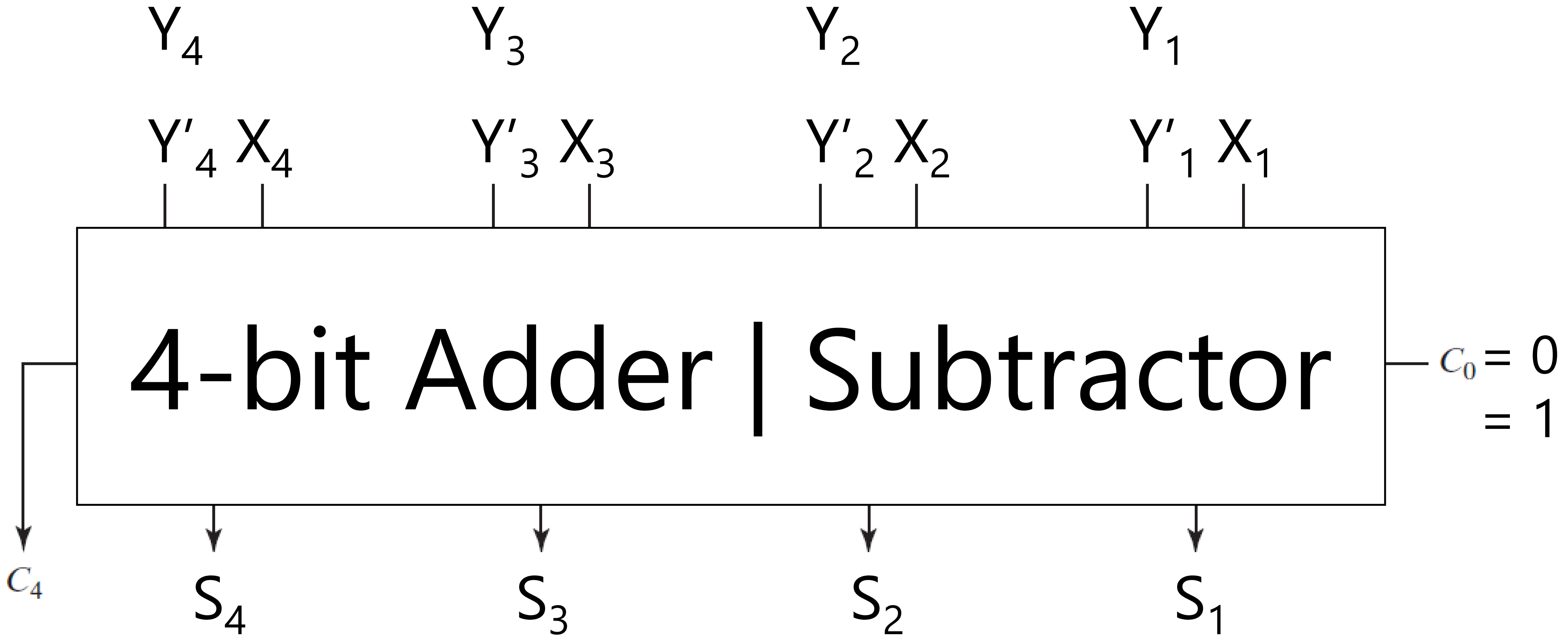
Subtraction in Signed-2's-Complement

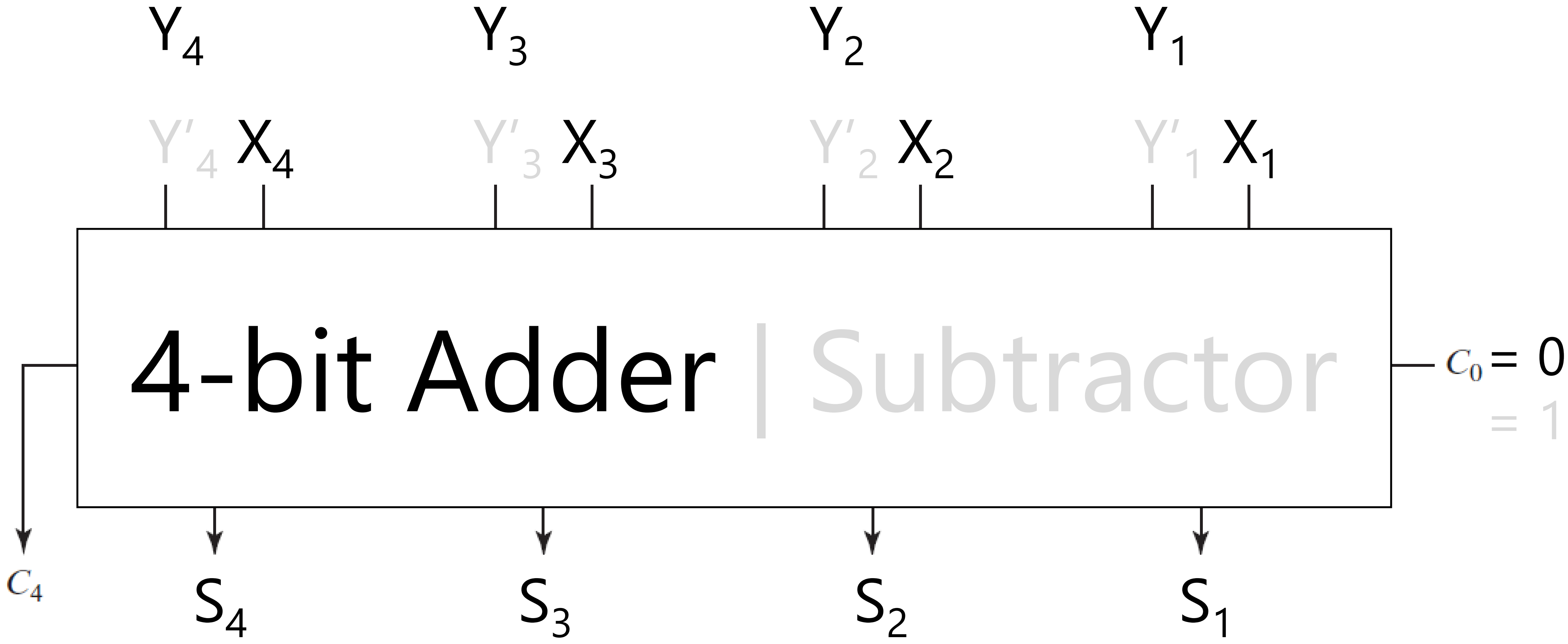
bitwise

$$X + Y' + (C_0 = 1)$$

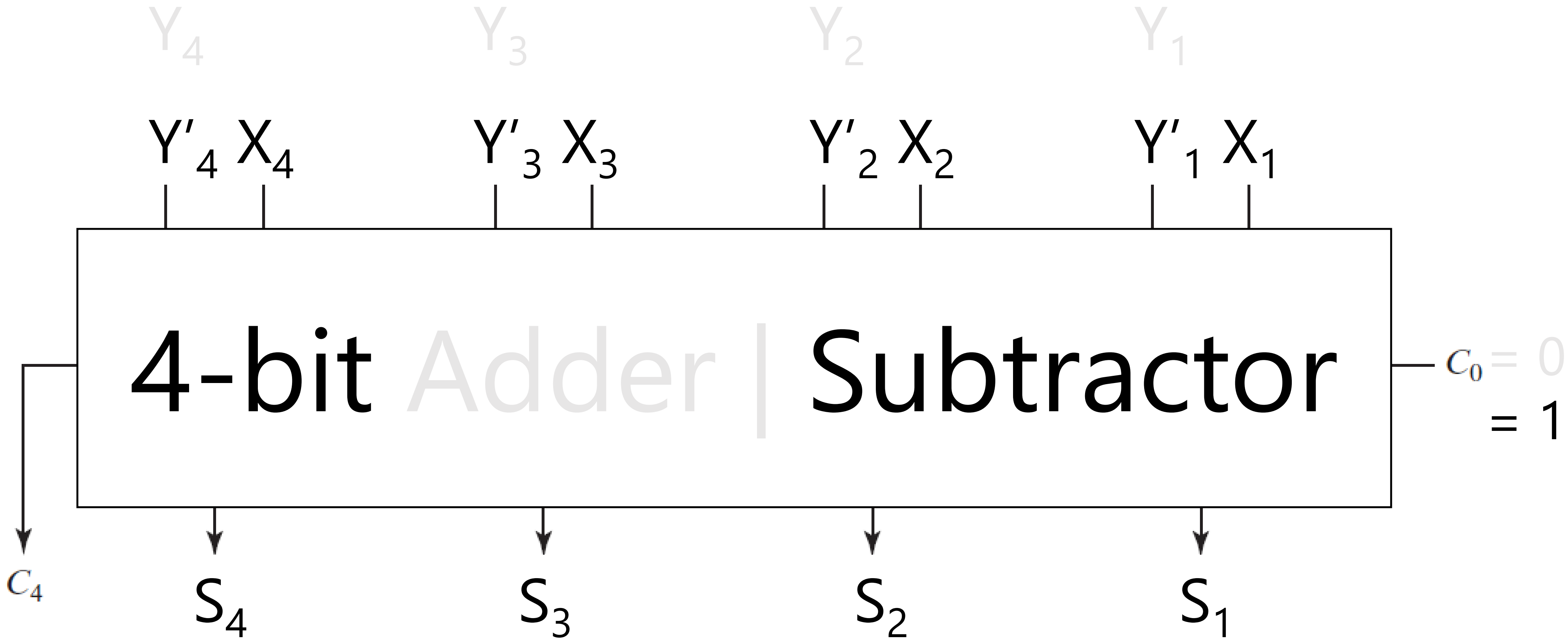
Subtraction in Signed-2's-Complement


Y'4 X4
Y'3 X3
Y'2 X2
Y'1 X1
4-bit Adder
C0 =1
C4
S4
S3
S2
S1


Y_4
Y_3
Y_2
Y_1
Y'_4 X_4
Y'_3 X_3
Y'_2 X_2
Y'_1 X_1
4-bit Adder | Subtractor
$C_0 = 0$
$= 1$
C_4
S_4
S_3
S_2
S_1



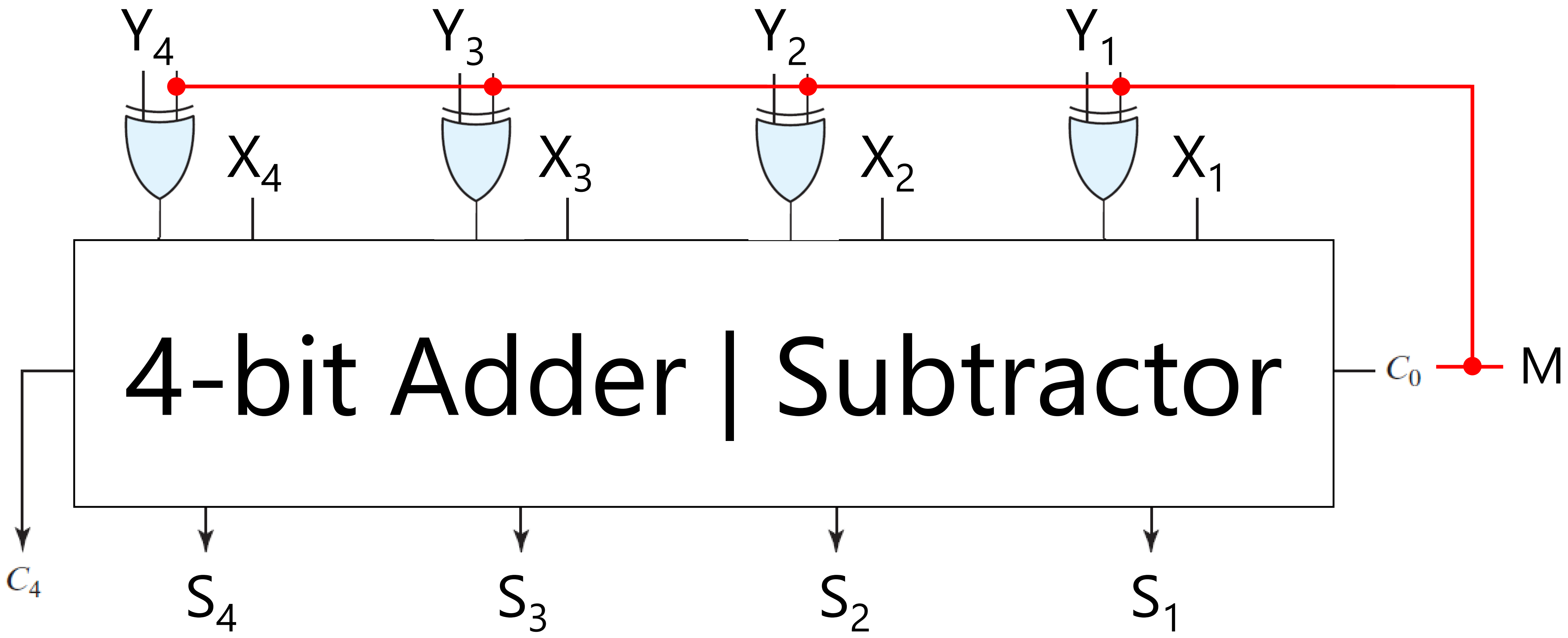
Y4
Y3
Y2
Y1
Y'4 X4
Y'3 X3
Y'2 X2
Y'1 X1
4-bit Adder | Subtractor
C0 = 0
= 1
C4
S4
S3
S2
S1



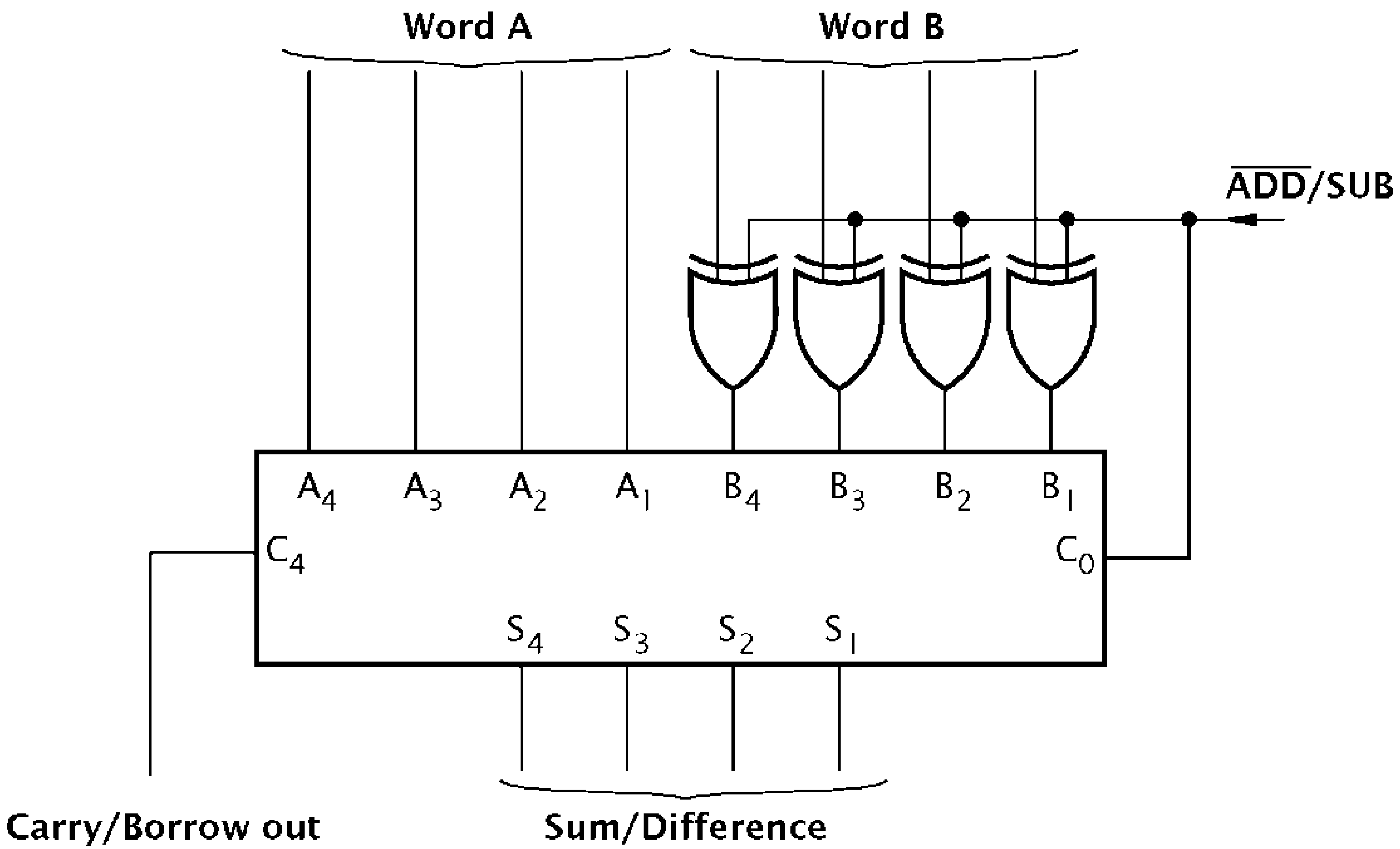
Y4 Y3 Y2 Y1
Y'4 X4 Y'3 X3 Y'2 X2 Y'1 X1
4-bit Adder | Subtractor
C0 = 0
= 1
C4
S4 S3 S2 S1

$$A ? 0 = A$$
$$A ? 1 = A'$$

$$A \oplus 0 = A$$
$$A \oplus 1 = A'$$



M=0 → Adder
M=1 → Subtractor


Word A
Word B
$\overline{ADD}$/SUB
A_4
A_3
A_2
A_1
B_4
B_3
B_2
B_1
C_4
C_0
S_4
S_3
S_2
S_1
Carry/Borrow out
Sum/Difference

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	C	OVF
4-Bit Adder									?

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

Y2	Y1	X2	X1	C	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!

Number System

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

Number System

We see positive number, but we interpret negative!

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

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

Either was addition originally
Or was subtraction and became addition with 2's-comp

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

OVF!

$$
\begin{array}{rl}
 & \boxed{C_3 = 1\ C_2 C_1 C_0} \\
 & X_4 = 0\ X_3 X_2 X_1 \\
+ & Y_4 = 0\ Y_3 Y_2 Y_1 \\
\hline
\boxed{C4=0} & S_4 = 1\ S_3 S_2 S_1
\end{array}
$$

OVF!

$C'_4 C_3$

$$\begin{array}{rl} & \boxed{C_3 C_2 C_1 C_0} \\ & X_4{=}1\ X_3 X_2 X_1 \\ + & Y_4{=}0\ Y_3 Y_2 Y_1 \\ \hline \boxed{C4} & S_4{=}?\ S_3 S_2 S_1 \end{array}$$

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

Don't believe it, try!

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

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

Don't believe it, try!

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

OVF!

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

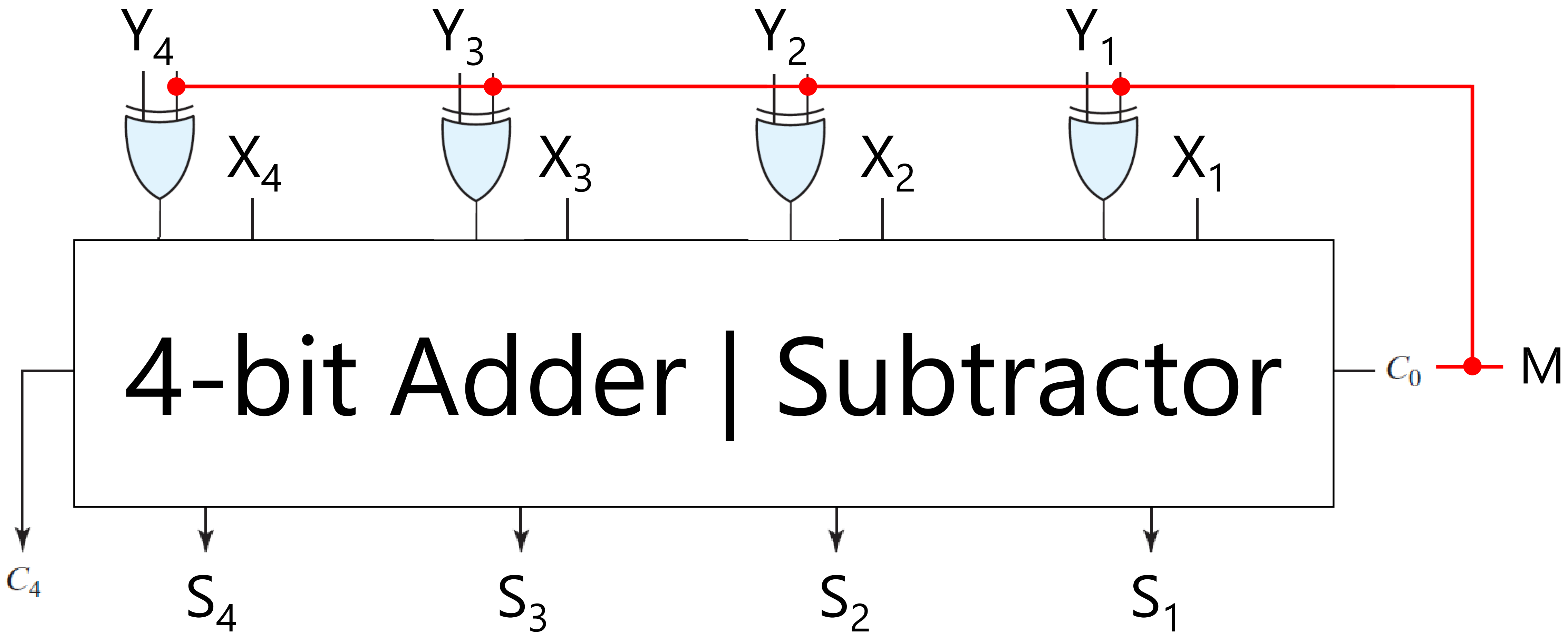
OVF!

$C_4C'_3$

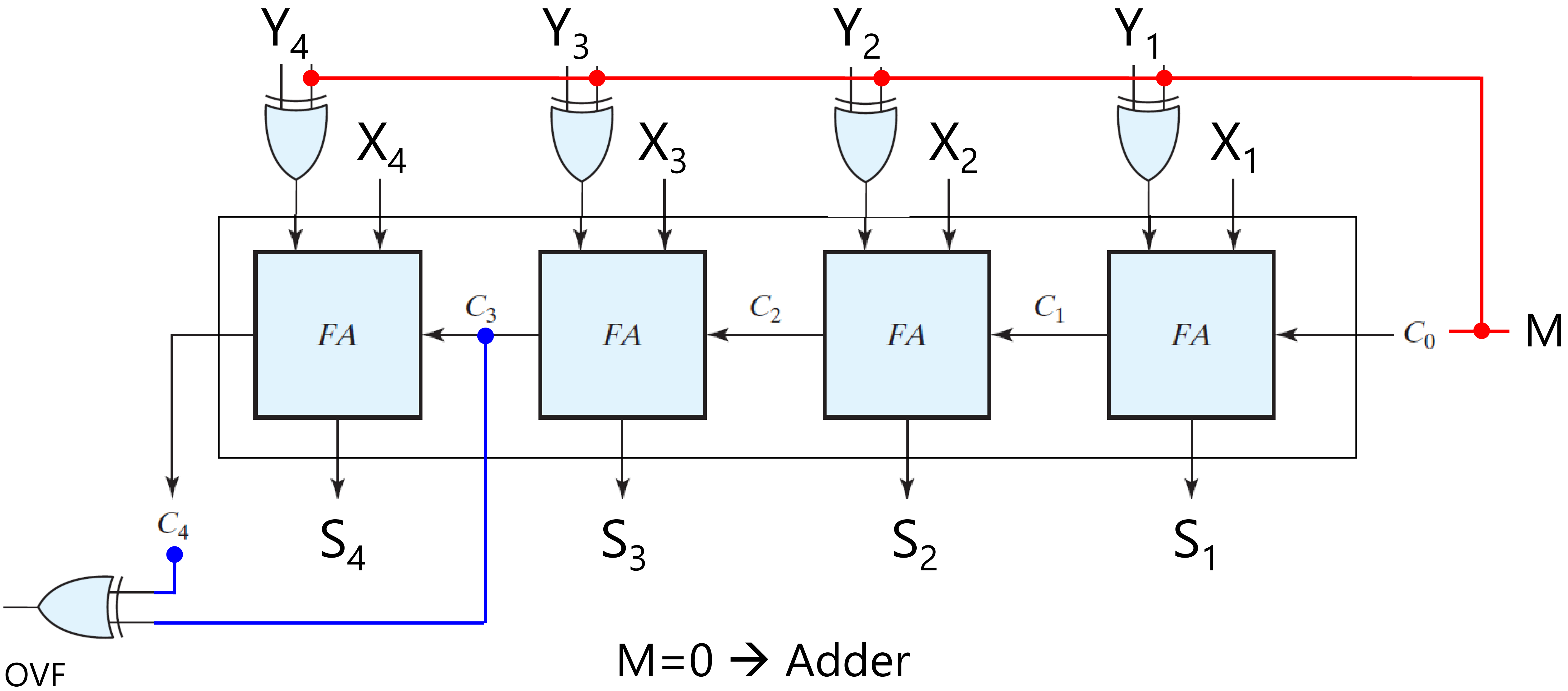
Design a logic circuit that detects overflow?

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?