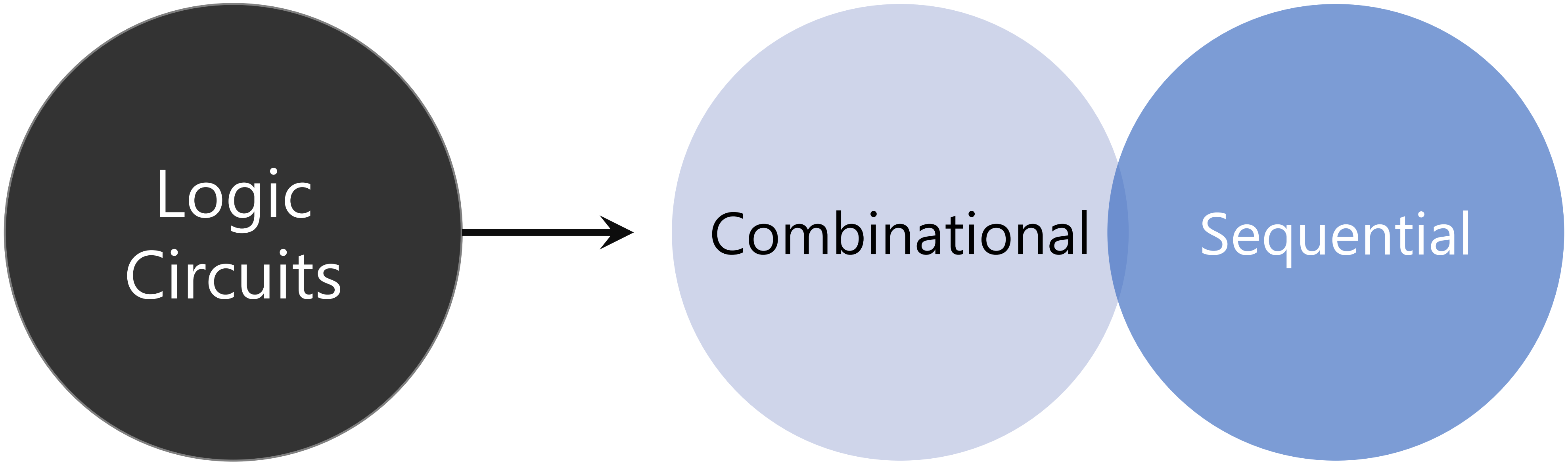

Logic
Circuits
Combinational
Sequential

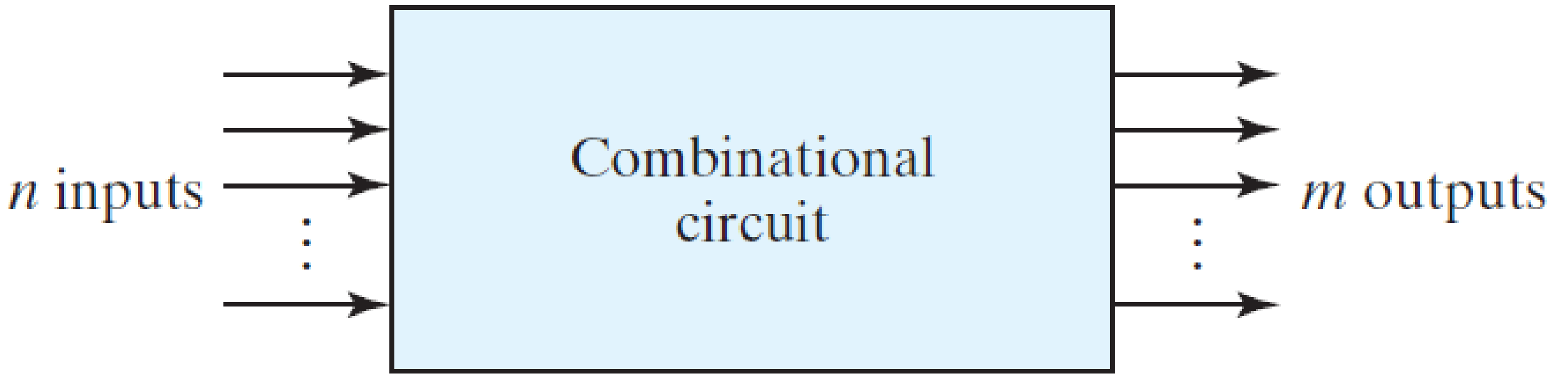
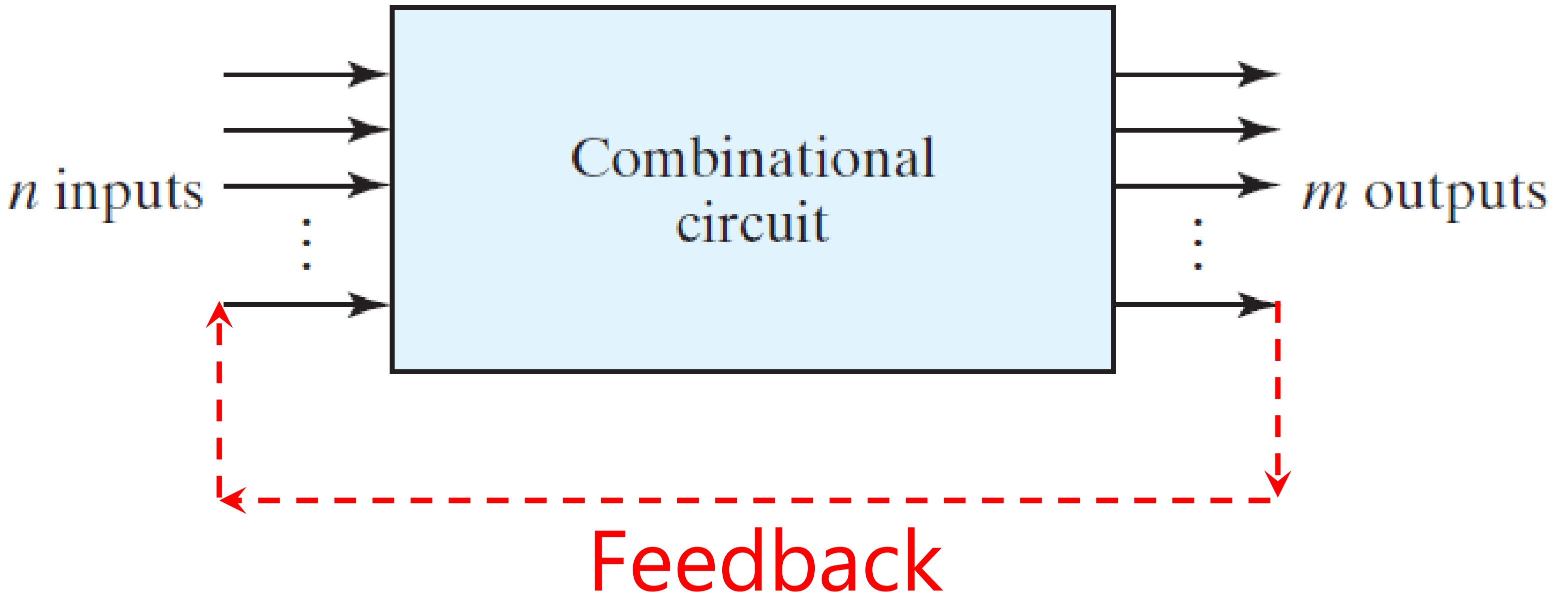


FIGURE 4.1
Block diagram of combinational circuit

Sequential Logic

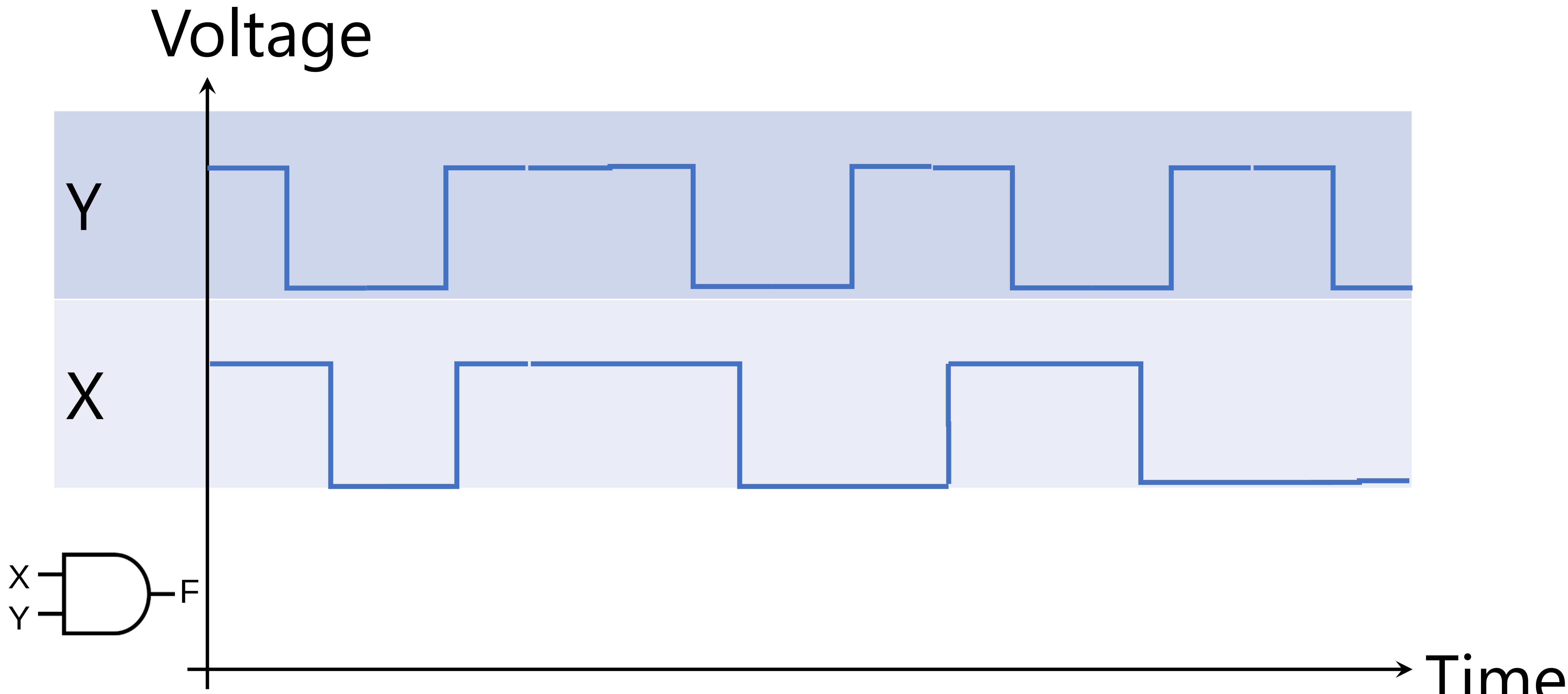


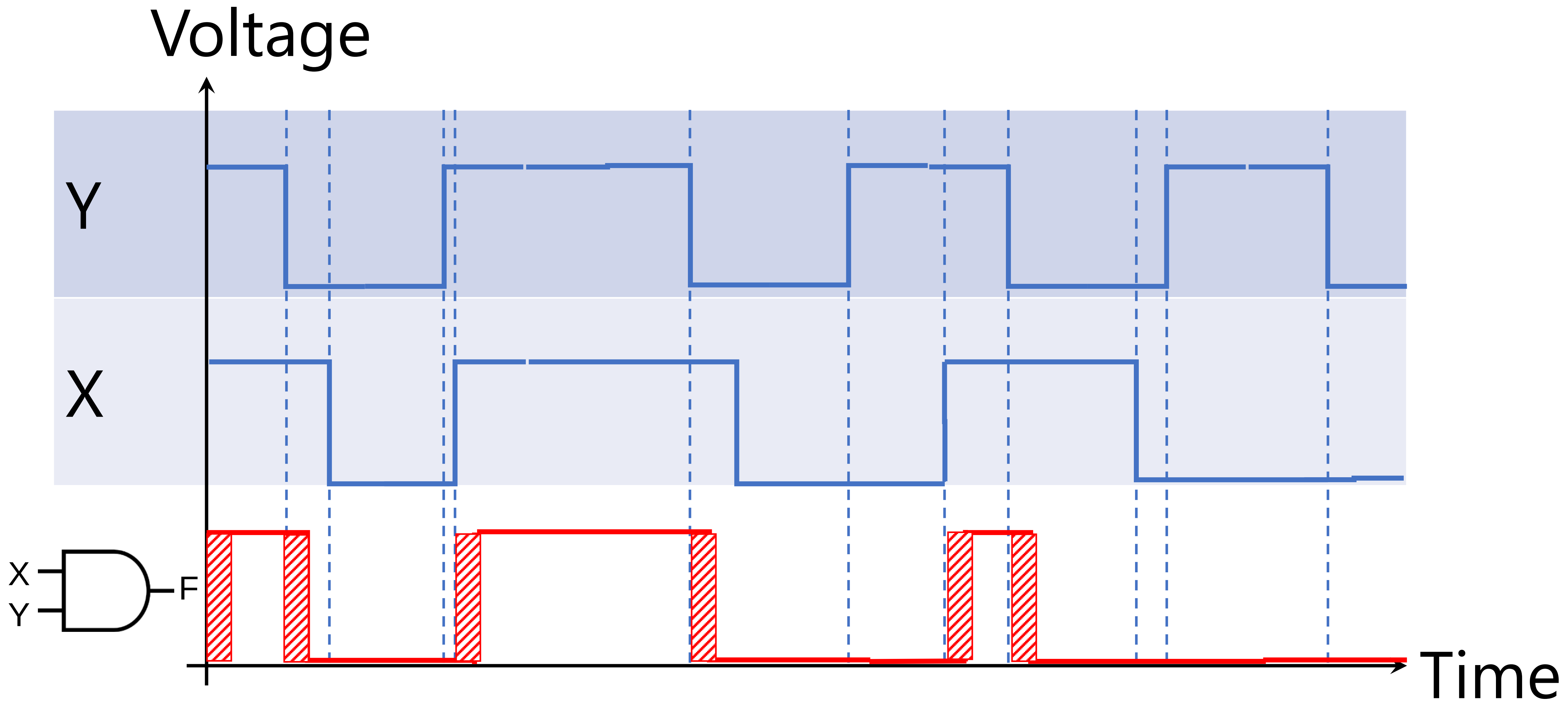
Combinational Logic

aka. Combinational Circuit

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

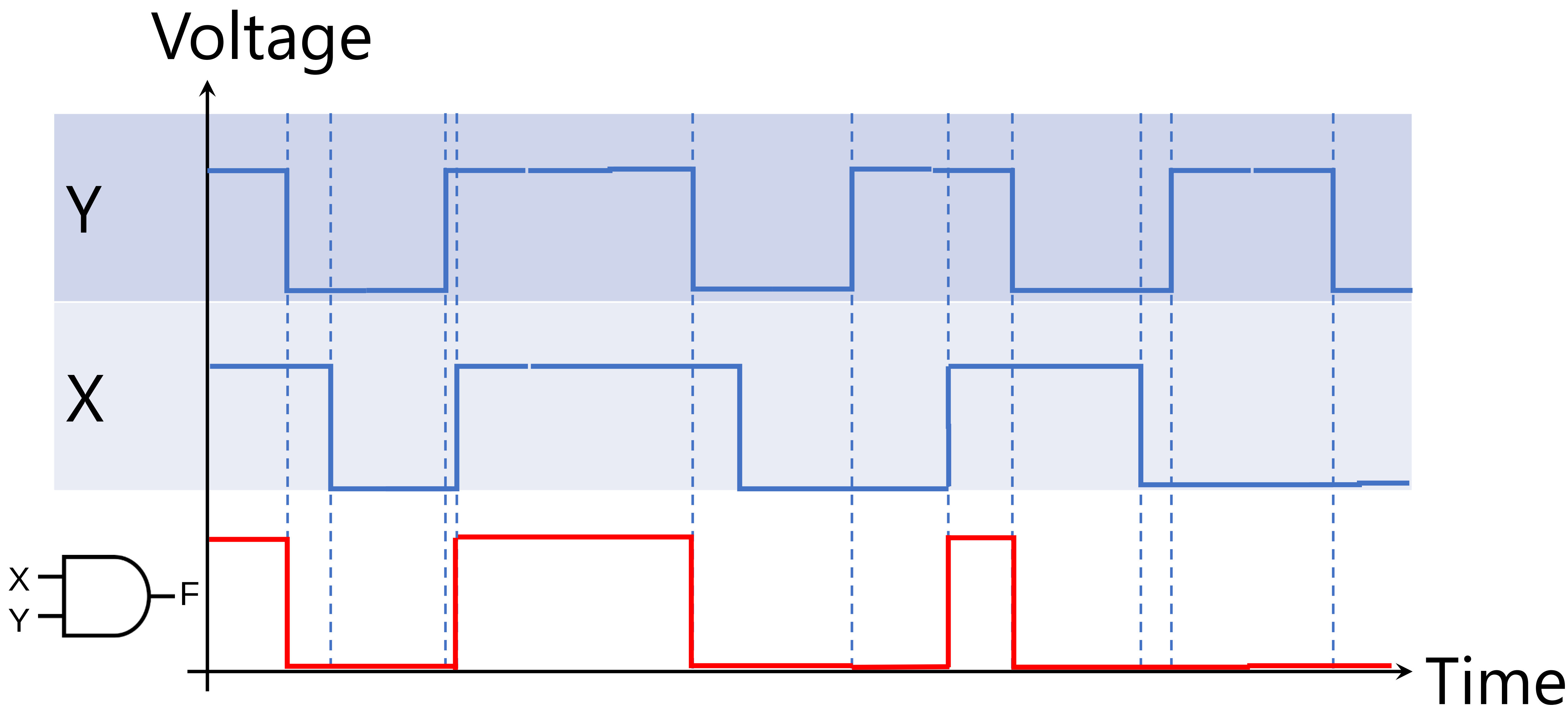
A combinational circuit performs an operation that can be specified logically by a set of Boolean functions.


Voltage
Y
X
X
Y
F
Time



Propagation Delay (Gate Delay) ≈ Δt

https://en.wikipedia.org/wiki/Propagation_delay#Electronics



Propagation Delay (Gate Delay) ≈ Δt ≈ 0

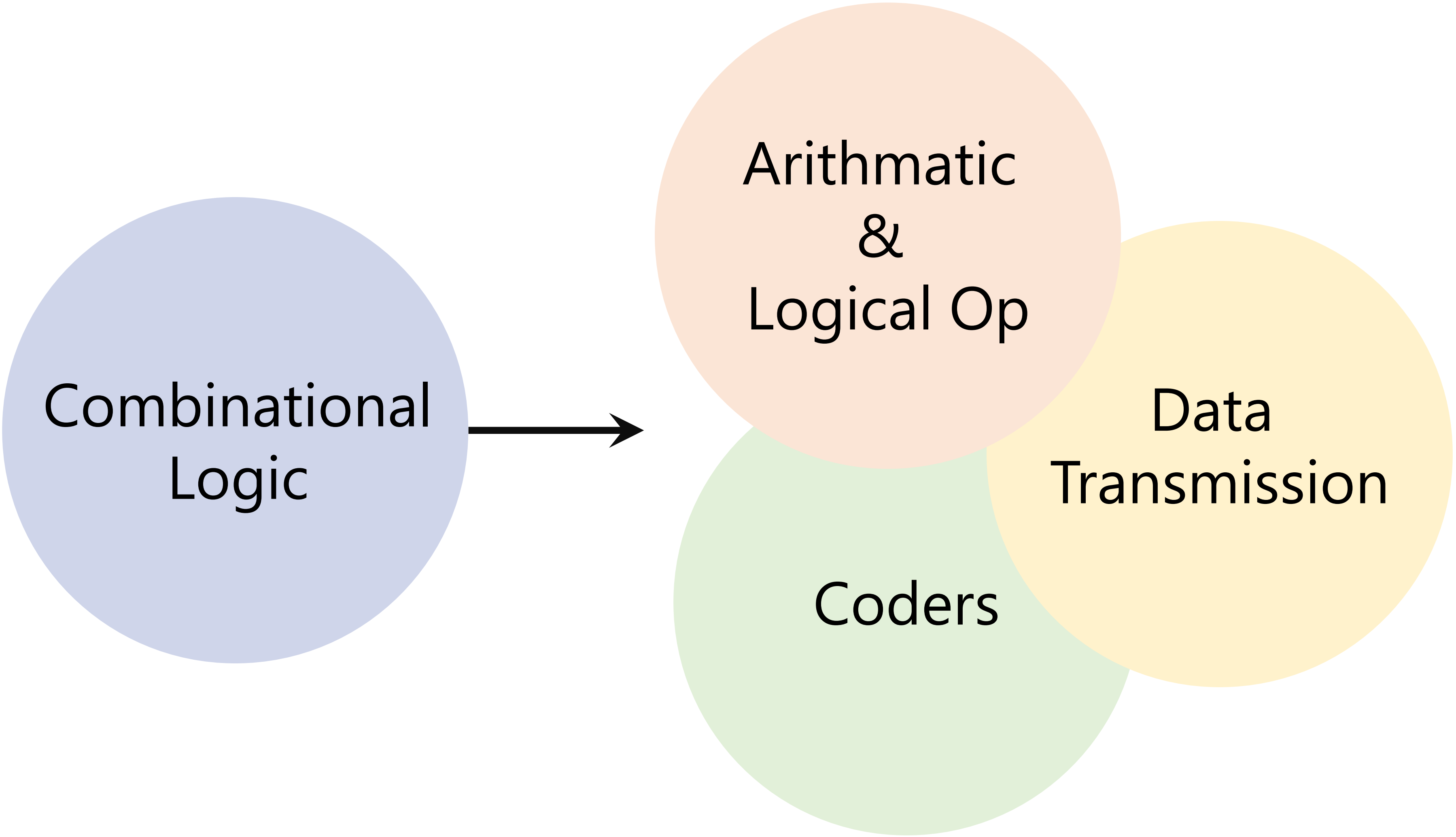
https://en.wikipedia.org/wiki/Propagation_delay#Electronics

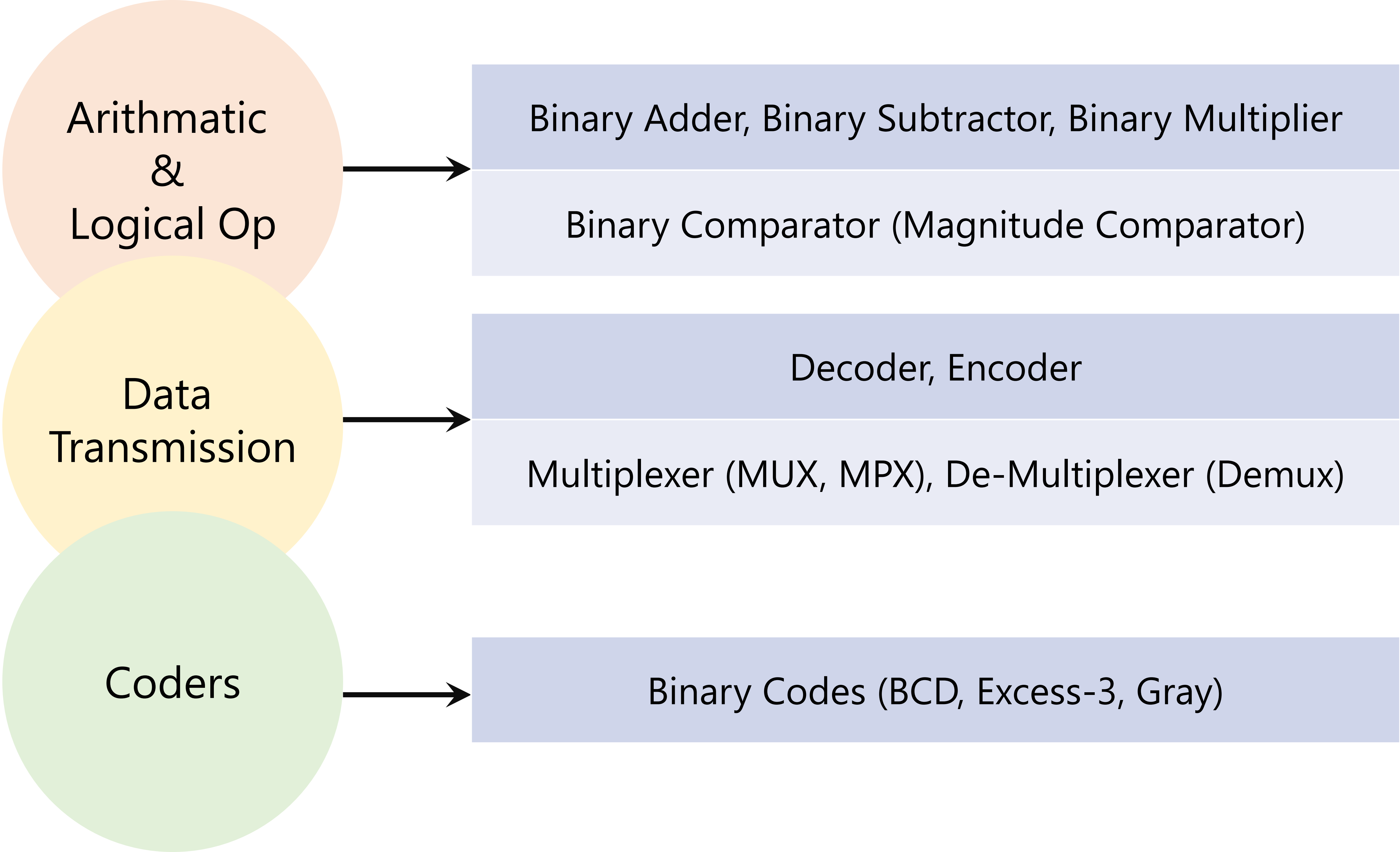
What we’ve done so far

Combinational Logic aka. Combinational Circuit

Design a combinational logic circuit:

1. Truth Table
2. Boolean Function (Algebraic Expression)
3. Minimization
    - Boolean Algebra
    - Karnaugh Map (K-Map)
    - Quine-McCluskey Algorithm
4. Logic Diagram


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

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

Range of inputs:
2 bits

Input binary variables:
X and Y

Range of outputs?

	0	0	1	1
+	0	1	0	1
	0	1	1	C=1 0

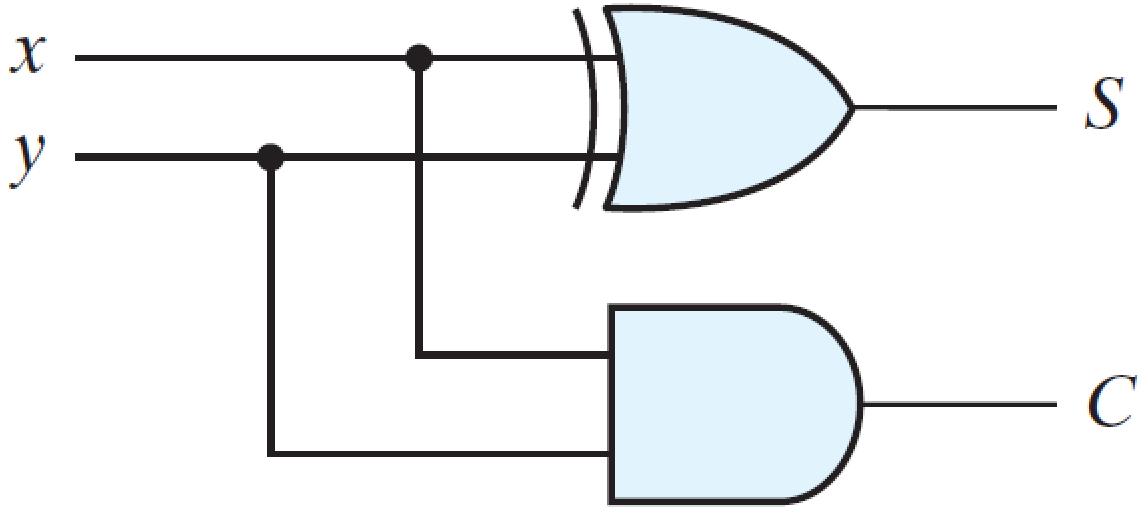
	0	0	1	1
+	0	1	0	1
	C=0 0	C=0 1	C=0 1	C=1 0

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

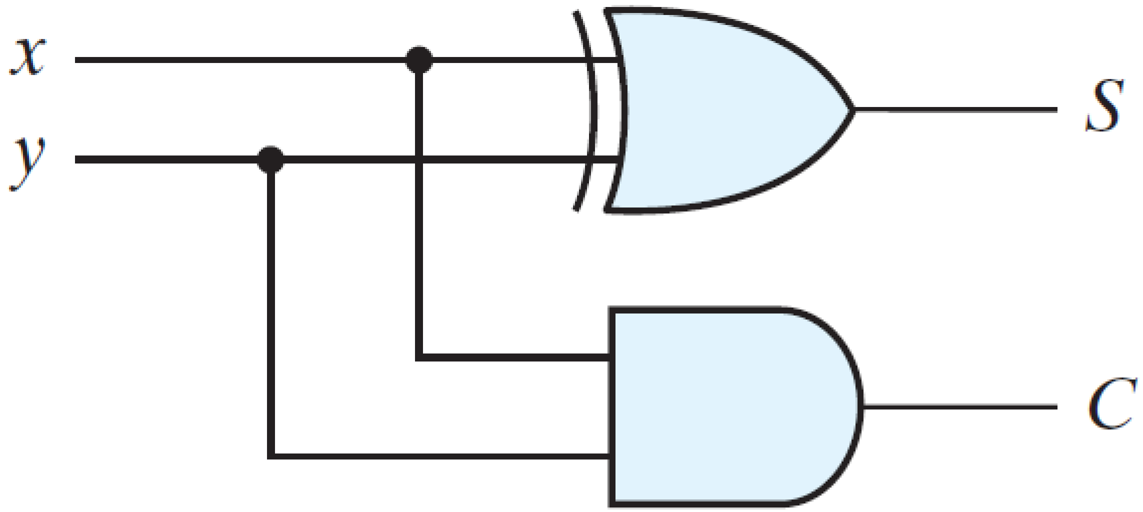
Range of outputs?
2 bits

Output binary variables: Carry and Sum

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

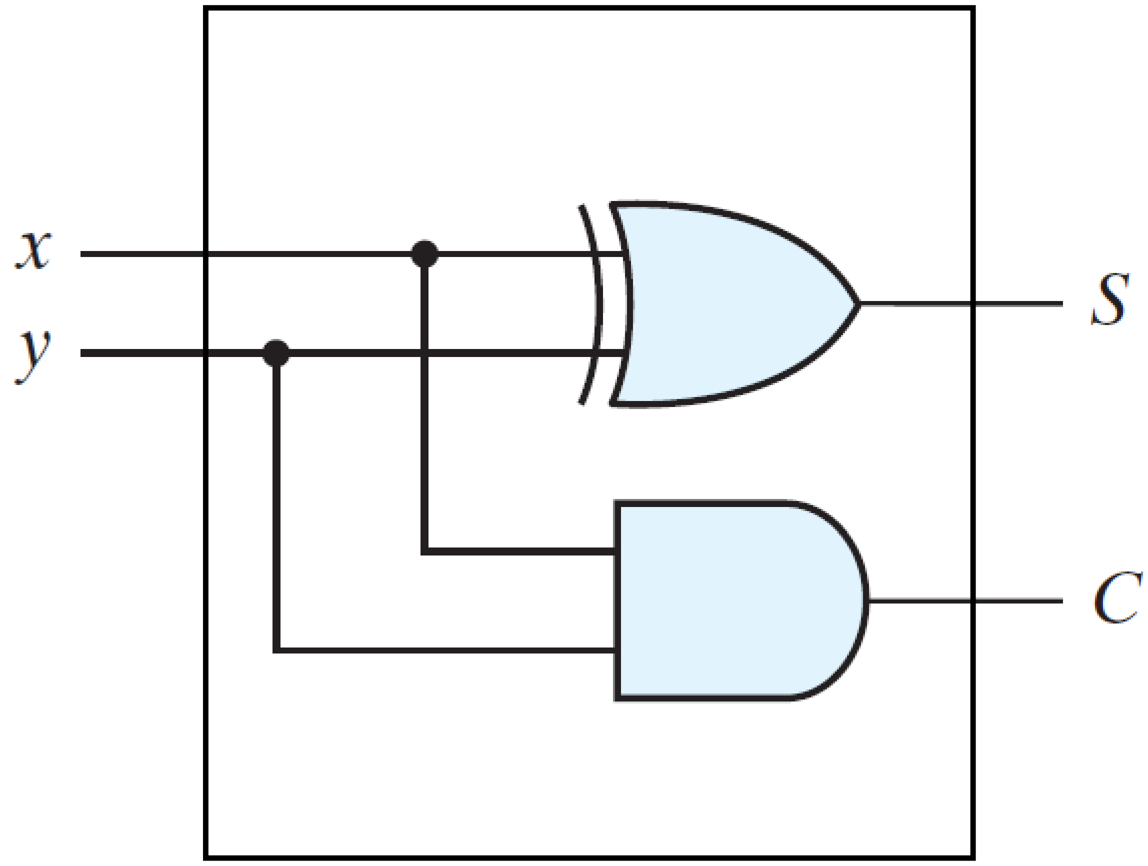


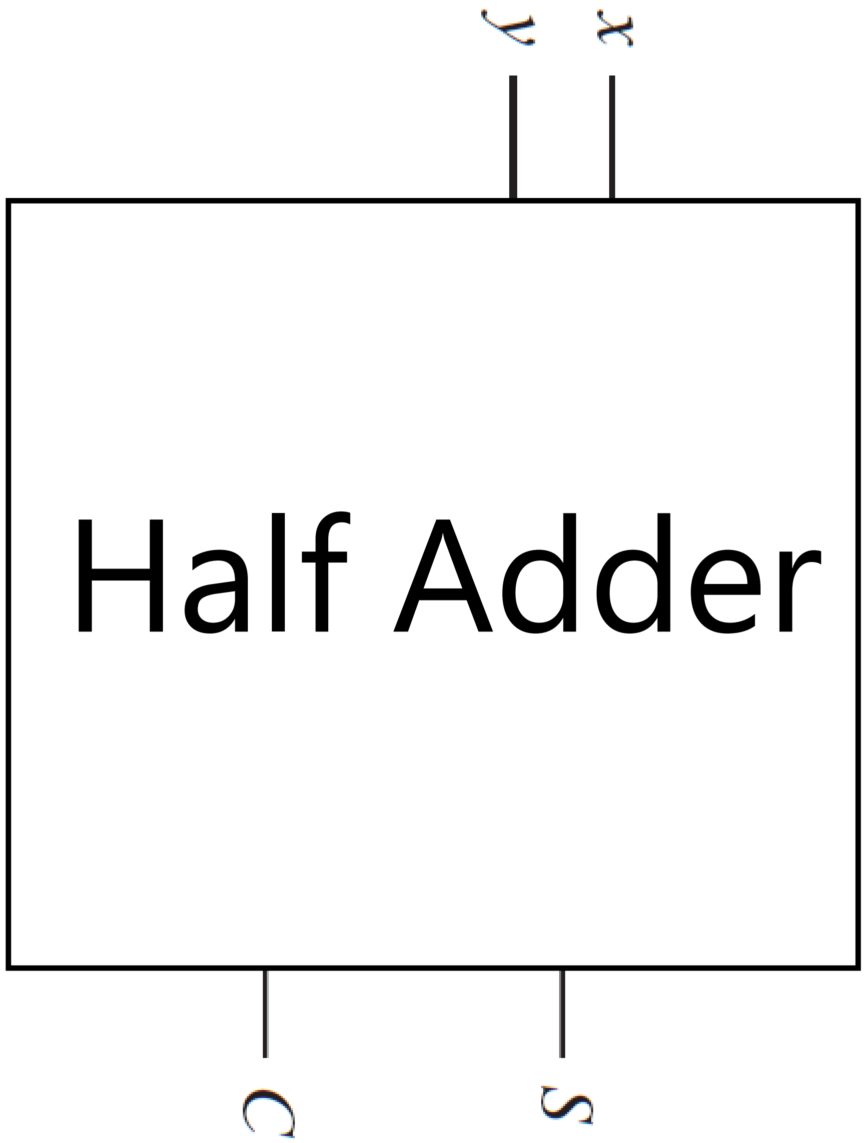
$$S = x \oplus y$$
$$C = xy$$



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

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!

Range of inputs:
2 binary numbers in range
$[00,11]_2$

Input binary variables:
$X=X_2X_1$ and $Y=Y_2Y_1$

Range of outputs?

	00	00	00	…	11
+	00	01	10	…	11
	C=0 00	C=0 01	C=0 10		C=1 10

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

Range of outputs?
Carry, S_2, S_1

Y_2	Y_1	X_2	X_1	$F_1=C(Y_1,Y_2,X_2,X_1)$	$F_2=S_1(Y_1,Y_2,X_2,X_1)$	$F_3=S_2(Y_1,Y_2,X_2,X_1)$
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

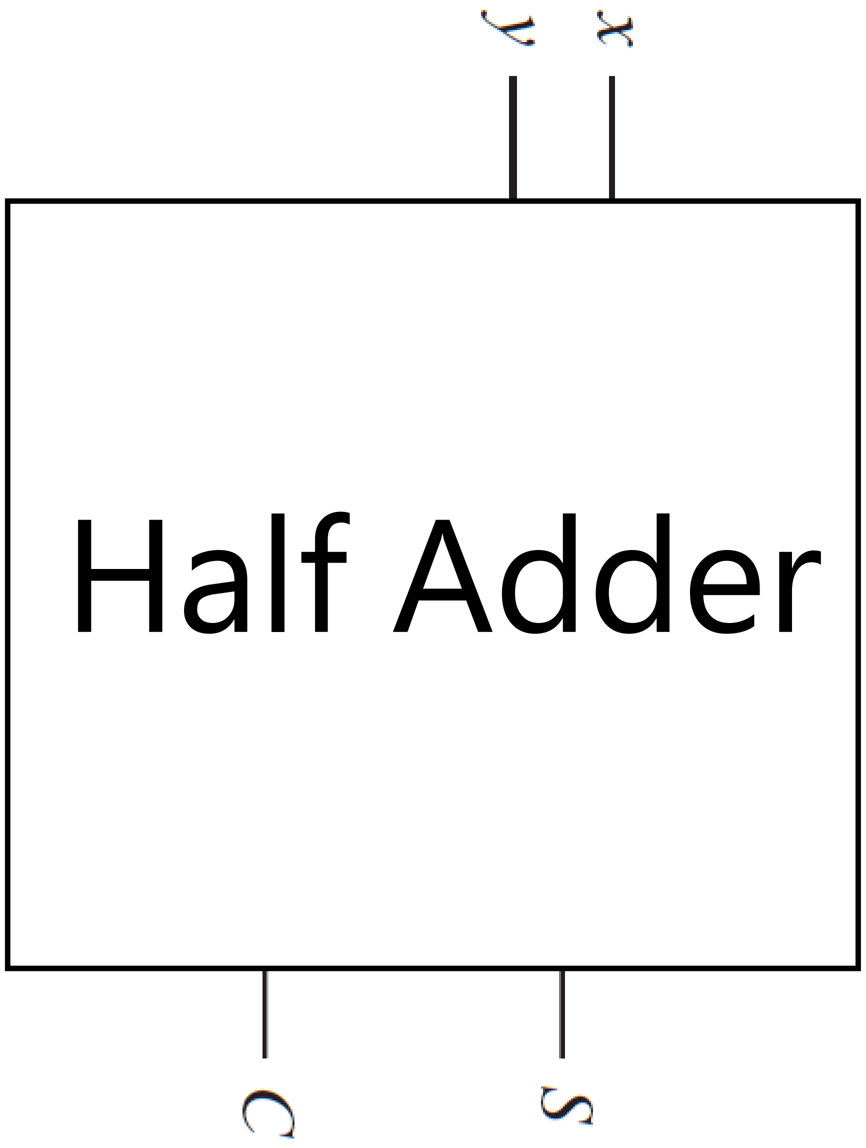
Y_2	Y_1	X_2	X_1	$F_1=C(Y_1,Y_2,X_2,X_1)$	$F_2=S_1(Y_1,Y_2,X_2,X_1)$	$F_3=S_2(Y_1,Y_2,X_2,X_1)$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	[illegible]	0
0	0	1	1	0	[illegible]	1
0	1	0	0	0	[illegible]	1
0	1	0	1	0	[illegible]	0
0	1	1	0	0	[illegible]	1
0	1	1	1	[illegible]	[illegible]	0
1	0	0	0	[illegible]	[illegible]	[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

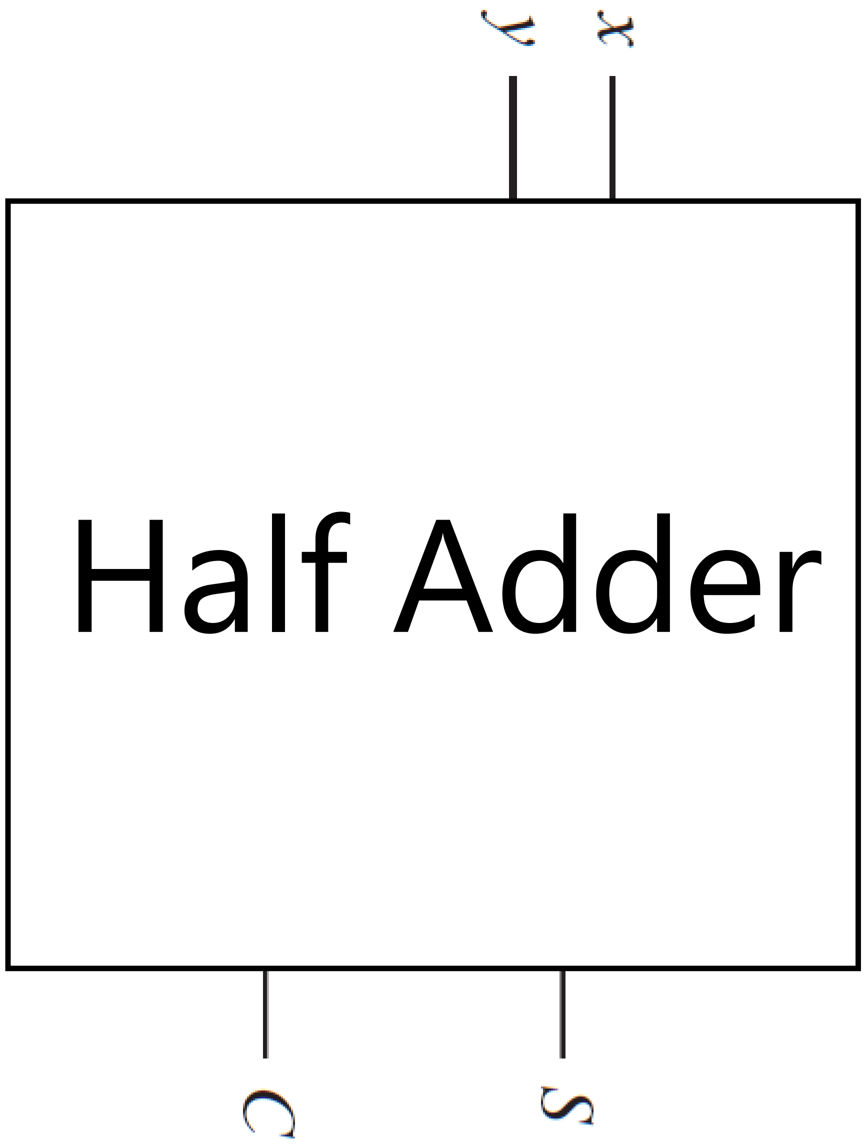
Wait a sec!

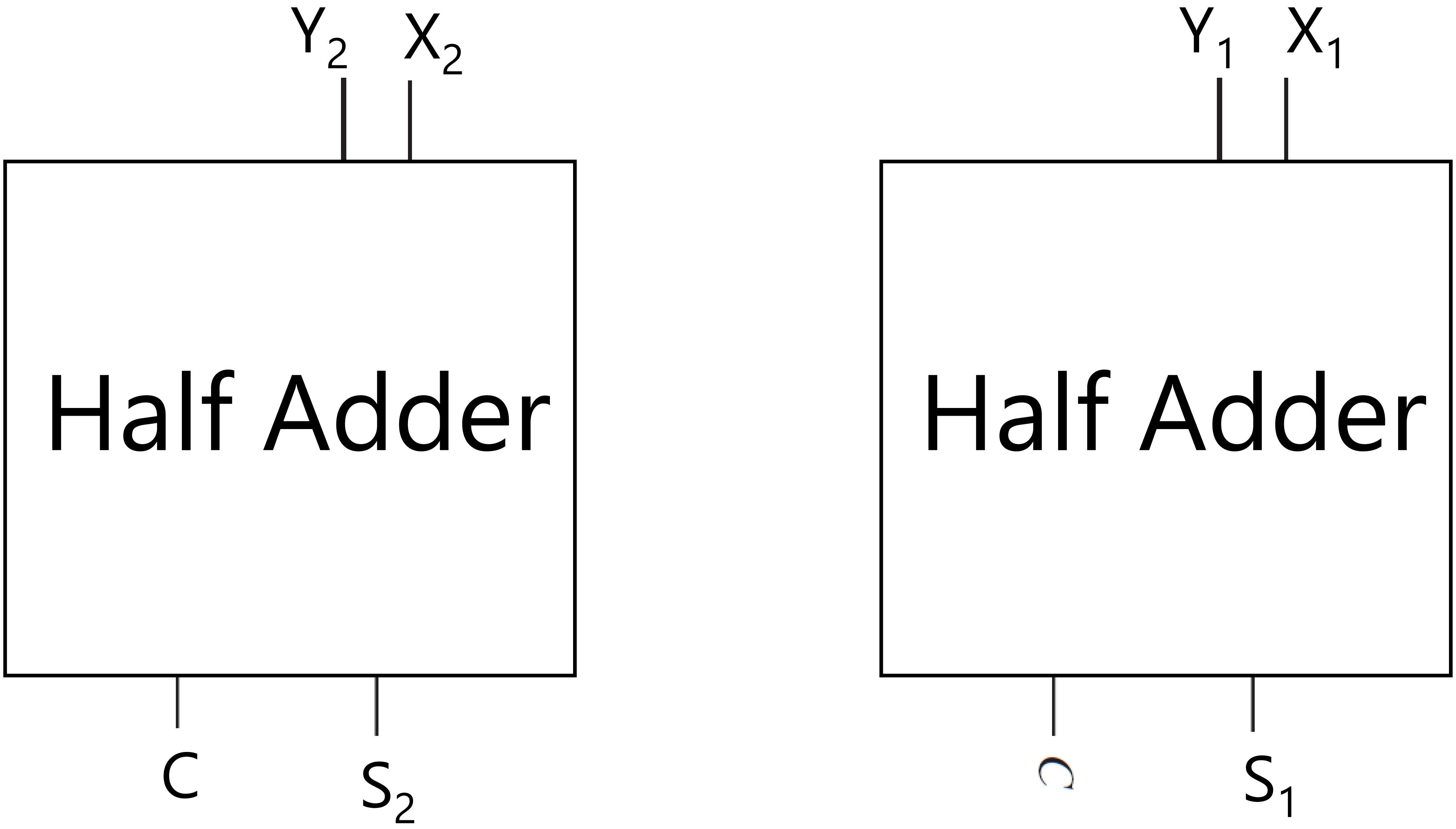
Can we re-use the half adder?

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

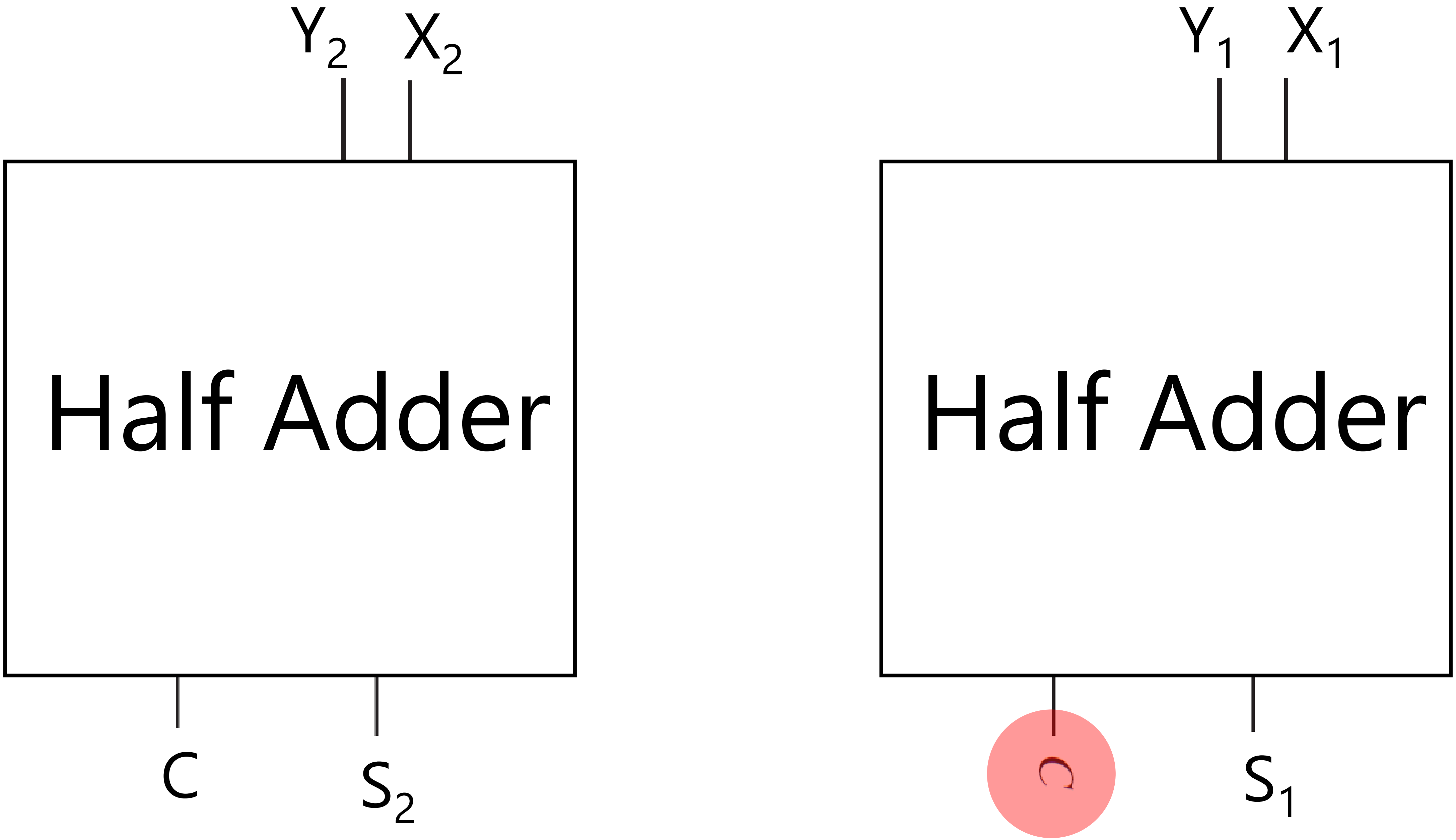
$$
\begin{array}{rcc}
 & & X_2X_1 \\
+ & & Y_2Y_1 \\
\hline
 & C & S_2S_1
\end{array}
$$


y
x
Half Adder
C
S


y
x
Half Adder
C
S

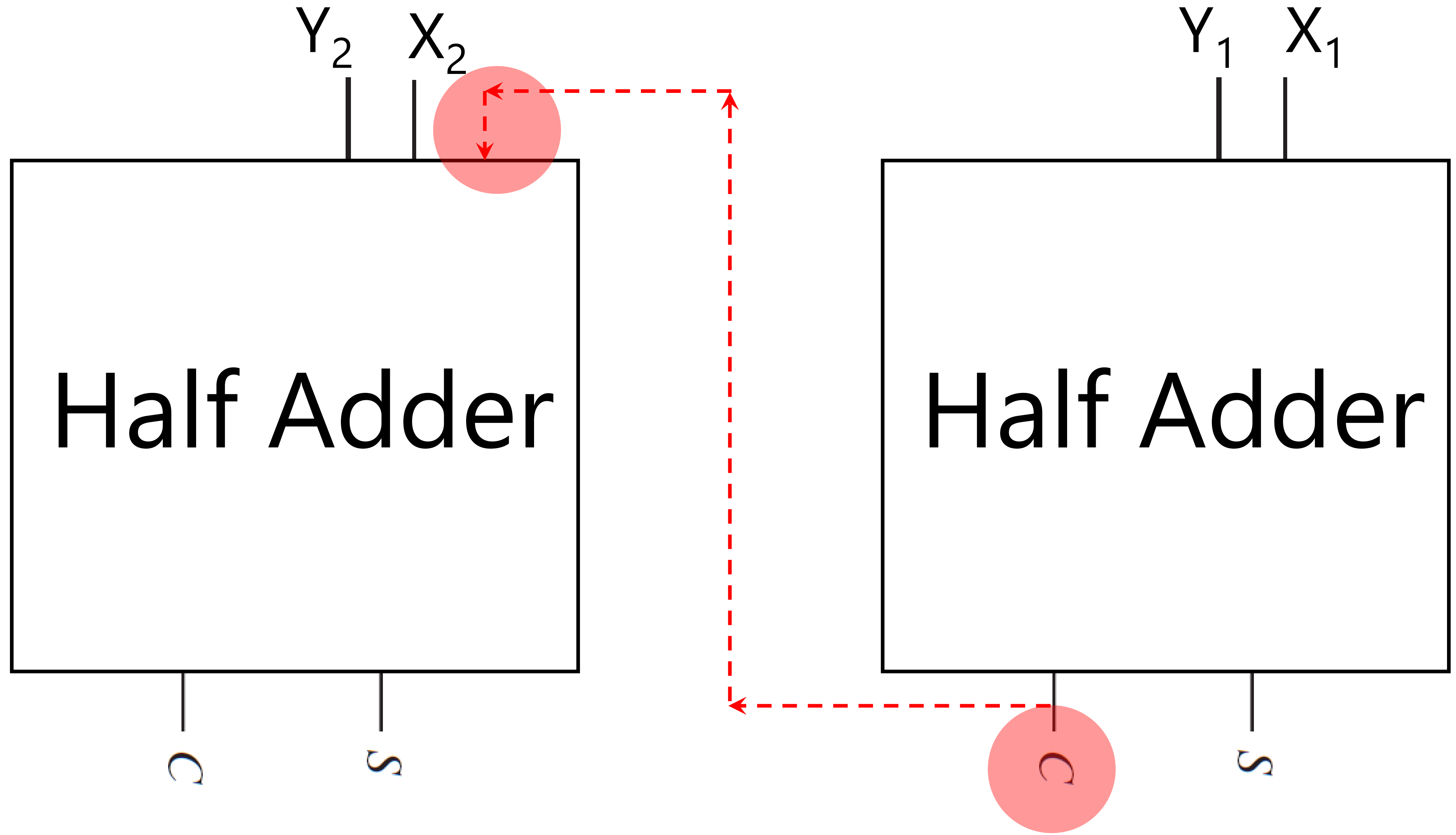


Y_2
X_2
Half Adder
C
S_2
Y_1
X_1
Half Adder
C
S_1

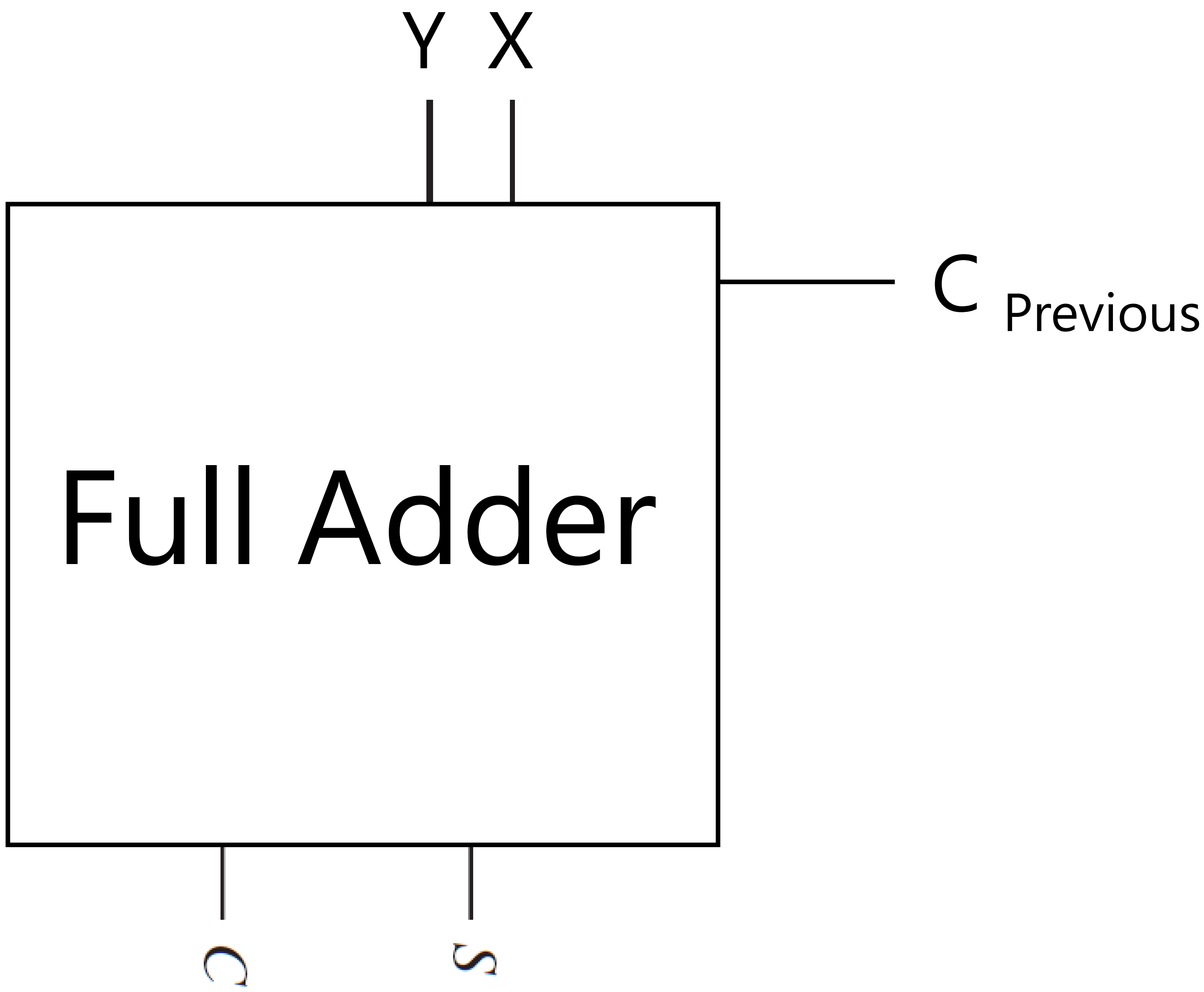


Y2
X2
Half Adder
C
S2
Y1
X1
Half Adder
C
S1

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



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


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

$$S=C'_pY'X+C'_pYX'+C_pY'X'+C_pYX$$
$$=C'_p(Y'X+YX')+C_p(Y'X'+YX)$$
$$=C'_p(X\oplus Y)+C_p(Y'X'+YX)$$

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

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

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

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

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

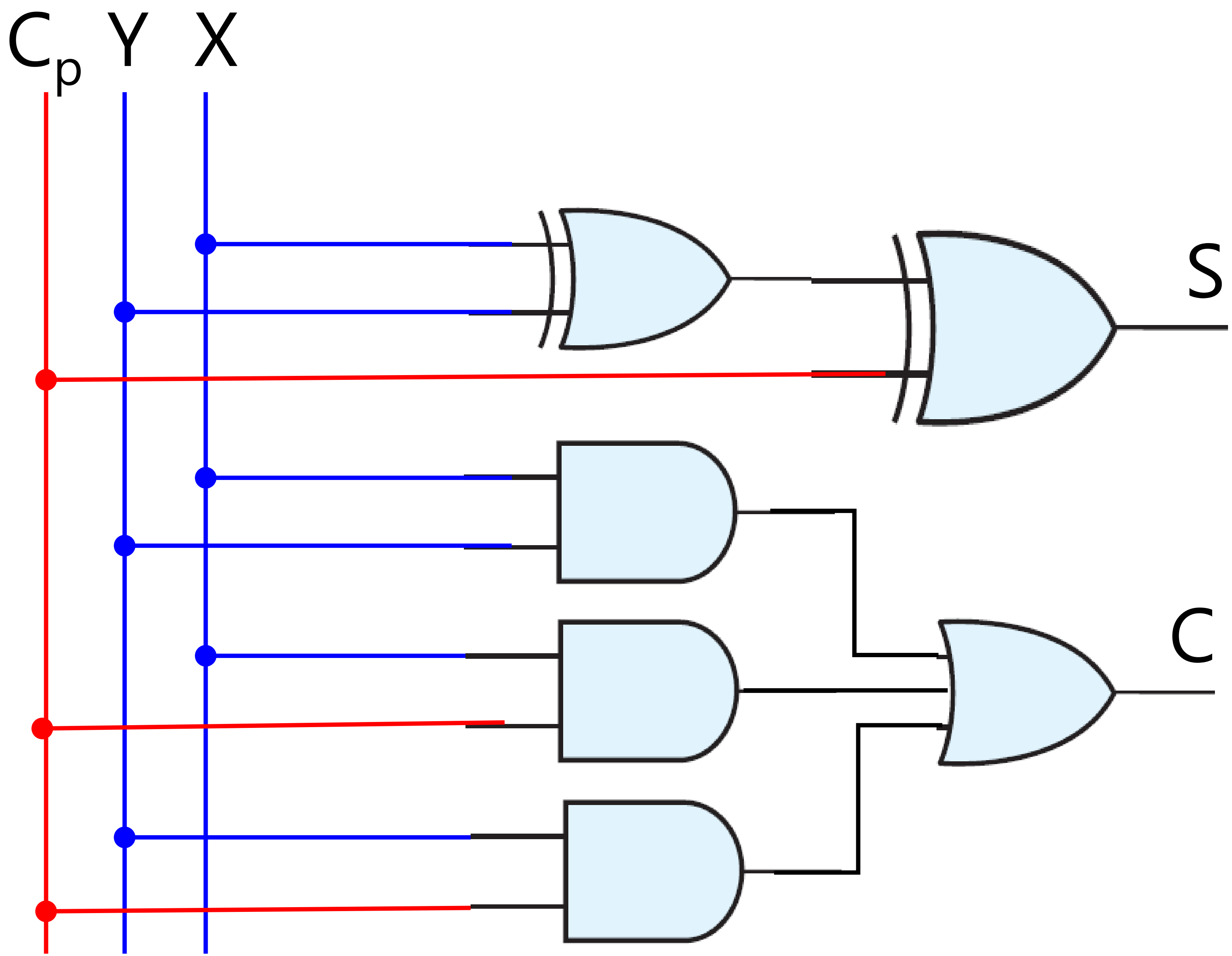
C_p \ YX	00	01	11	10
0	0 m_0	1 m_1	0 m_3	1 m_2
1	1 m_4	0 m_5	1 m_7	0 m_6

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

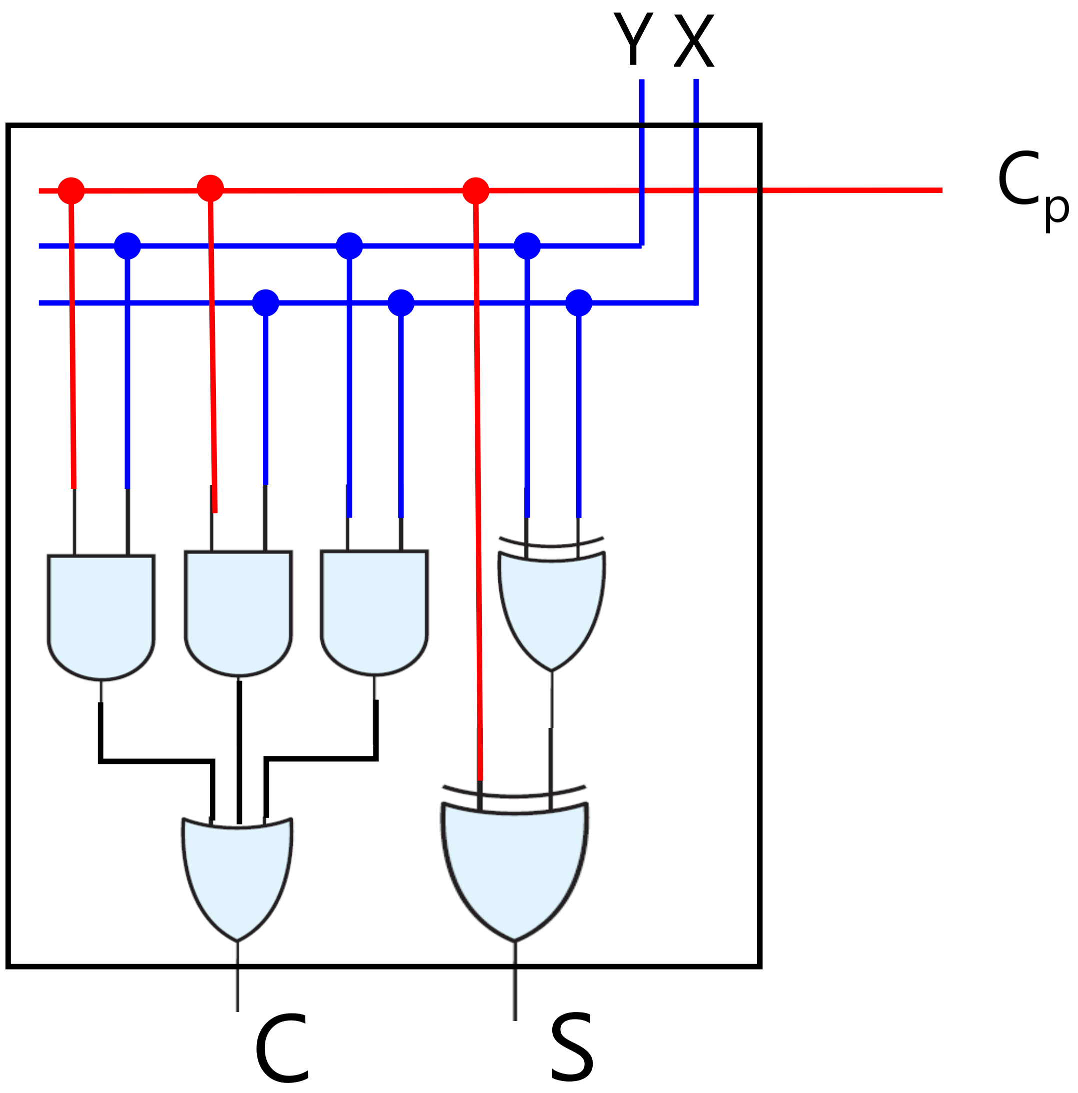
$$C=\sum m(3,5,6,7)$$

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

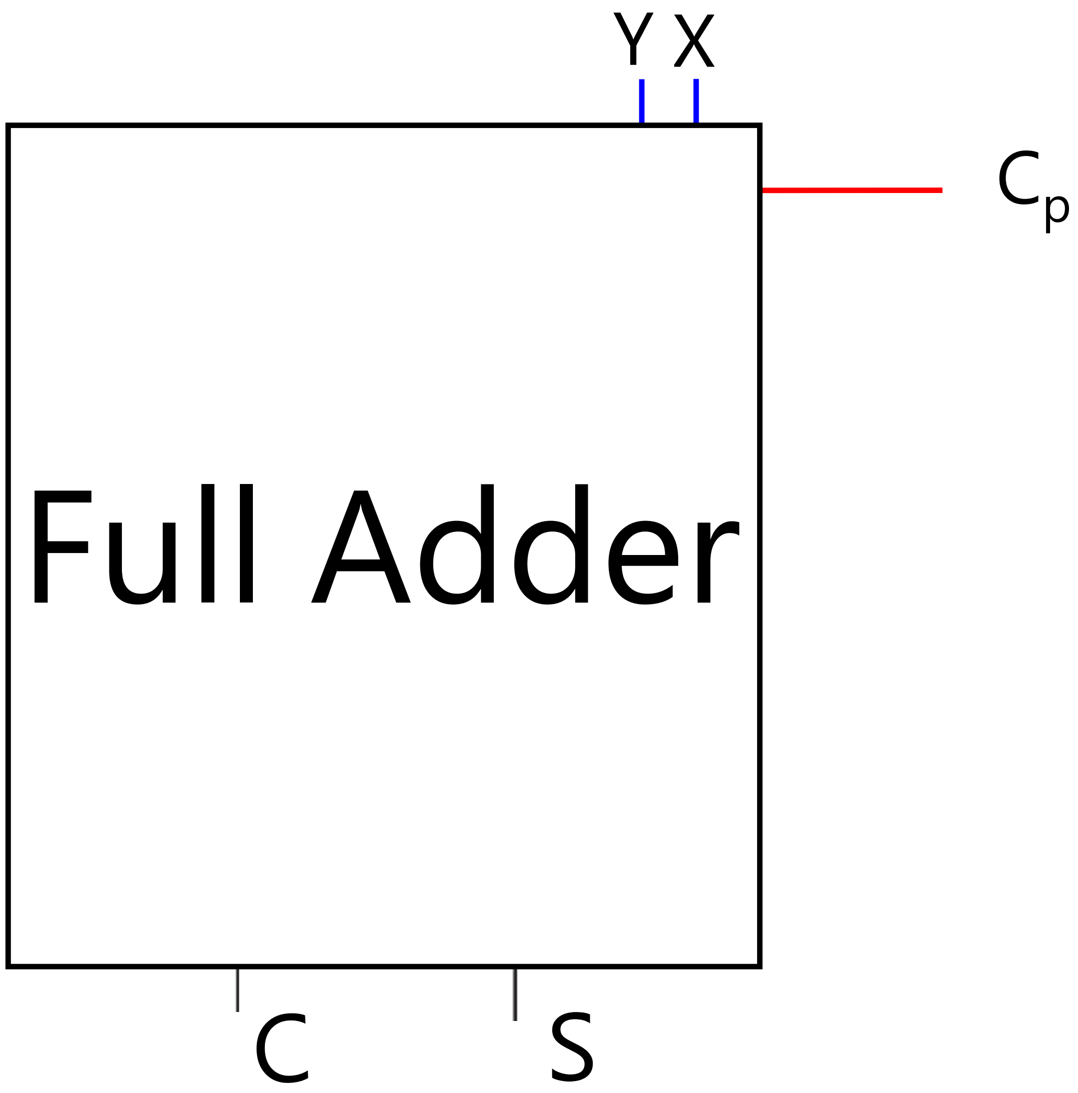
$$C=YX+C_pX+C_pY$$

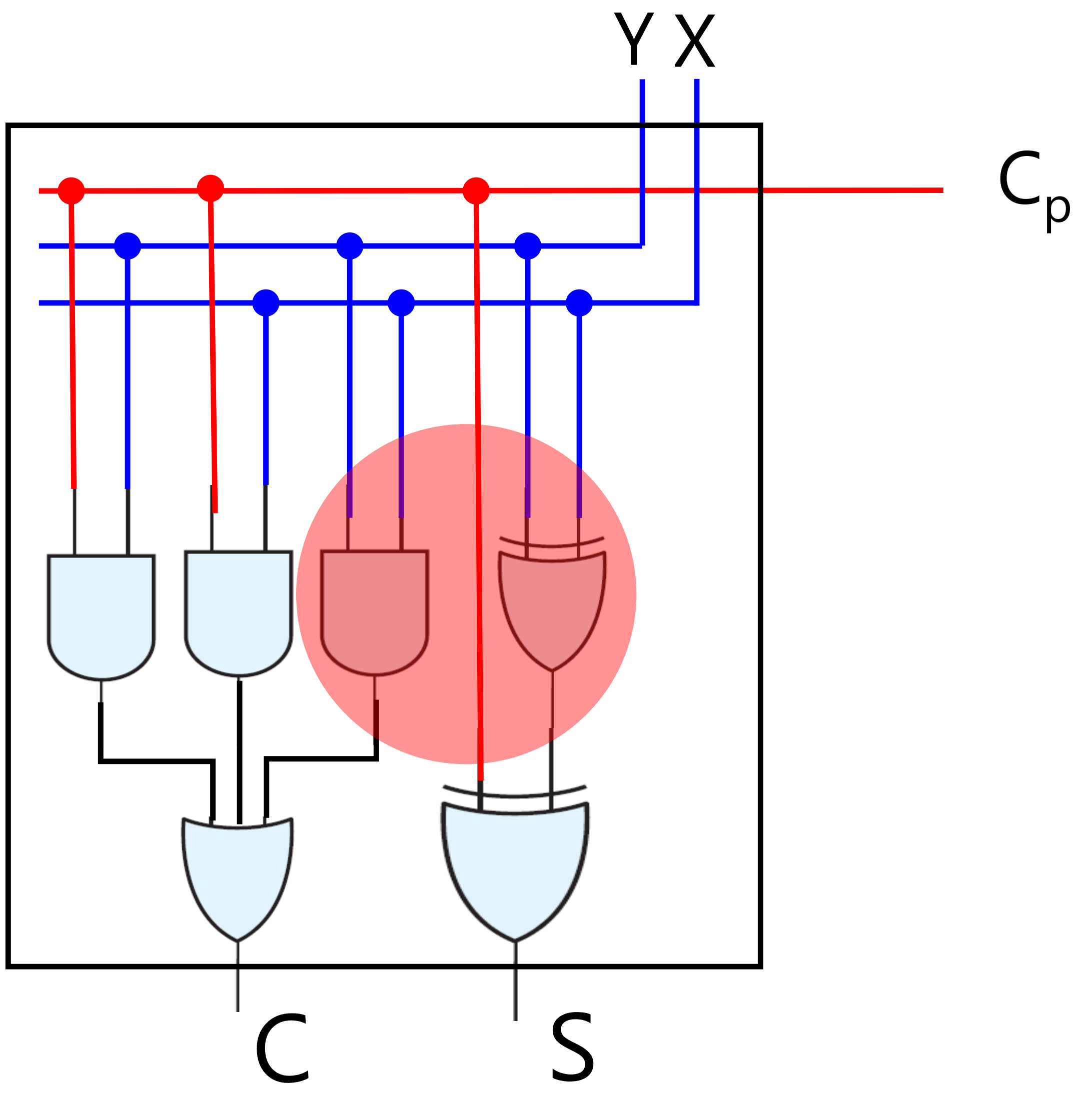


C_p
Y
X
S
C

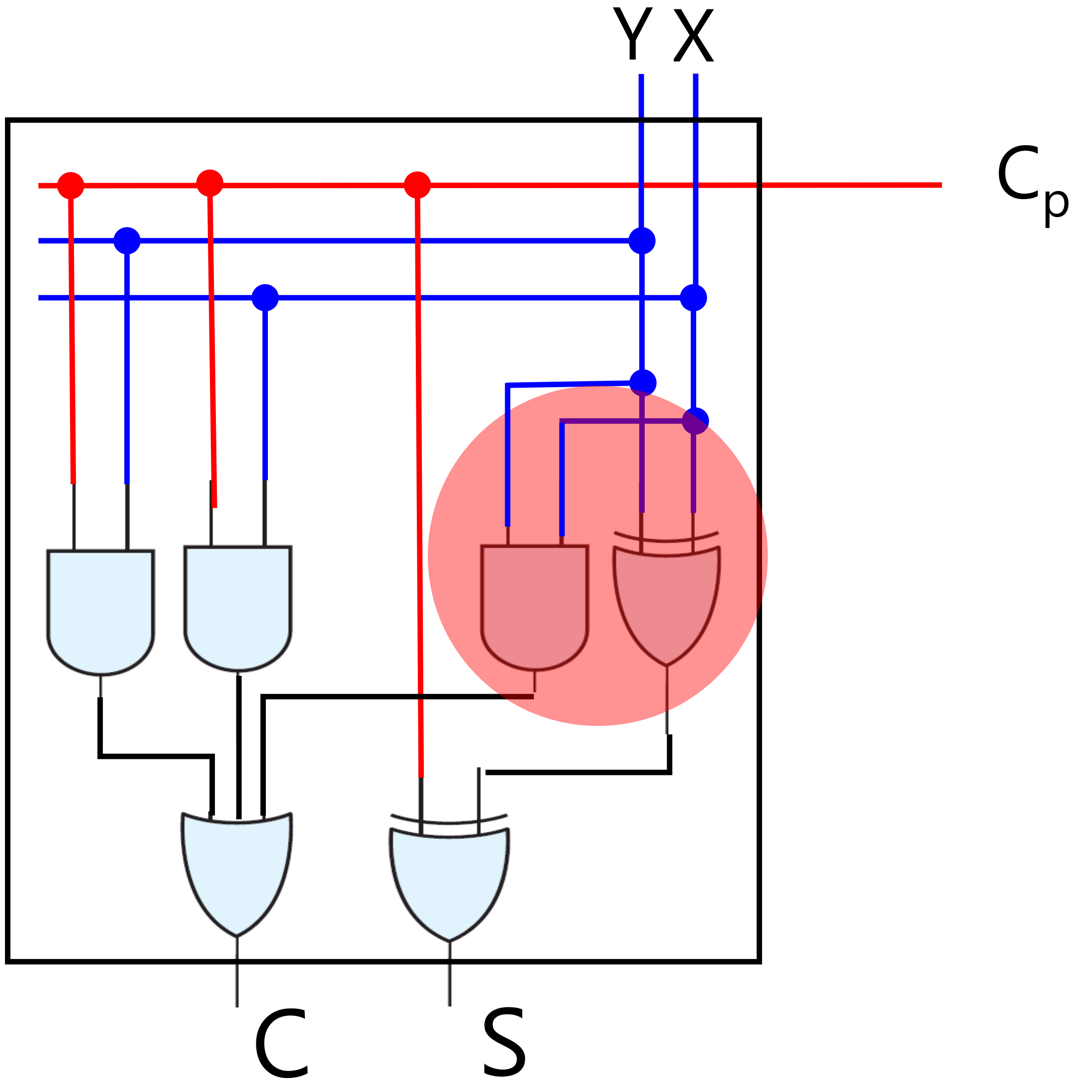


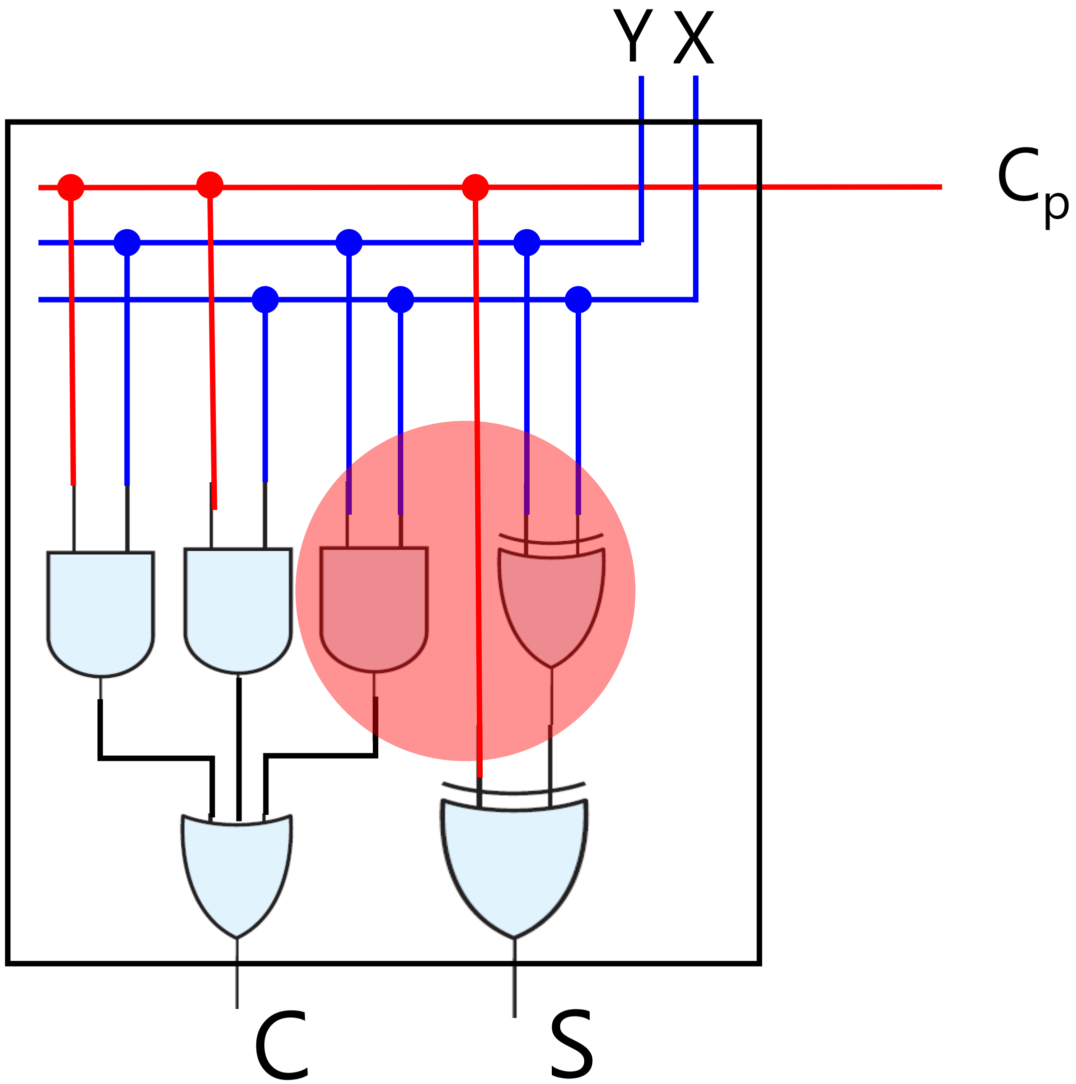
Y
X
C
p
C
S

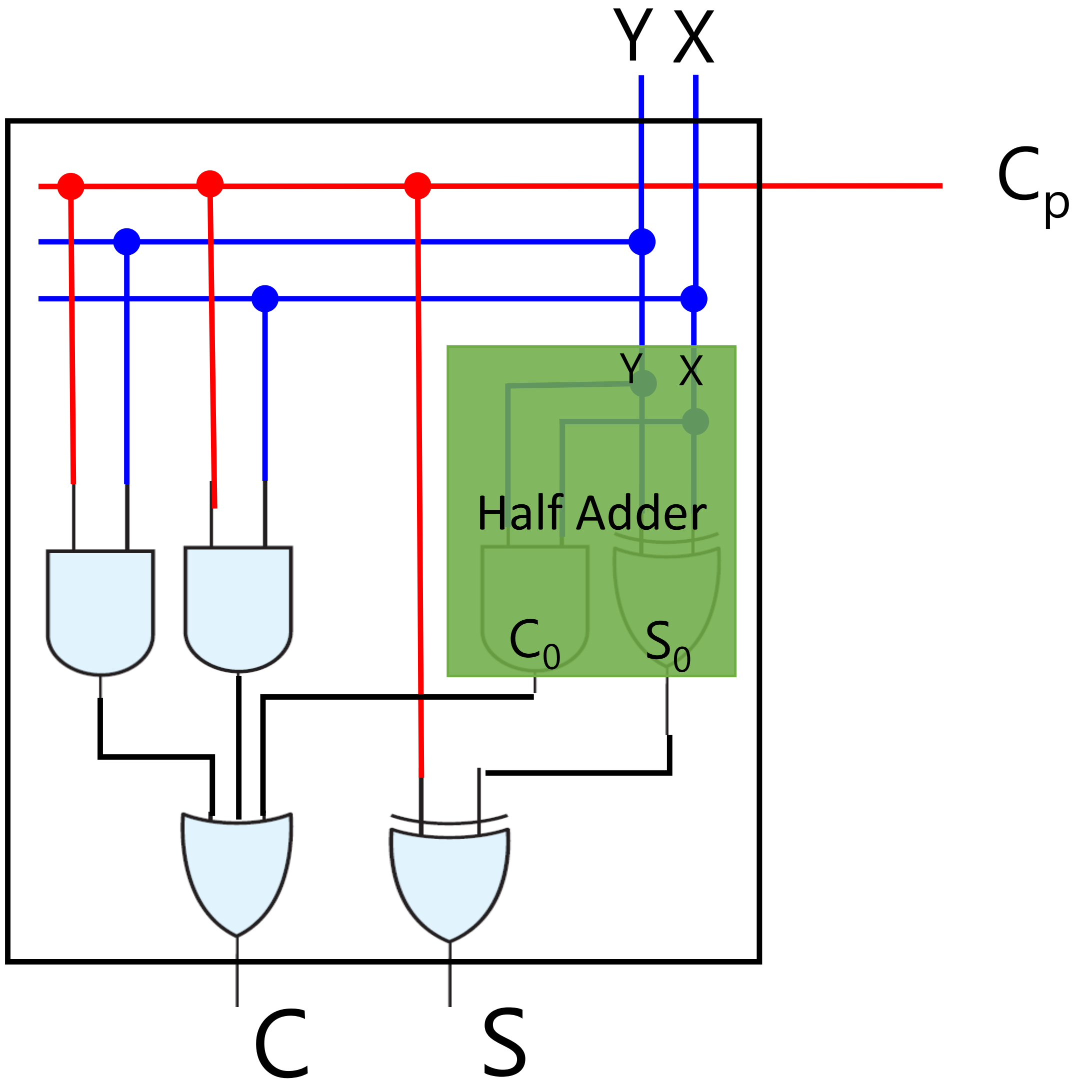

Y
X
C_p
Full Adder
C
S



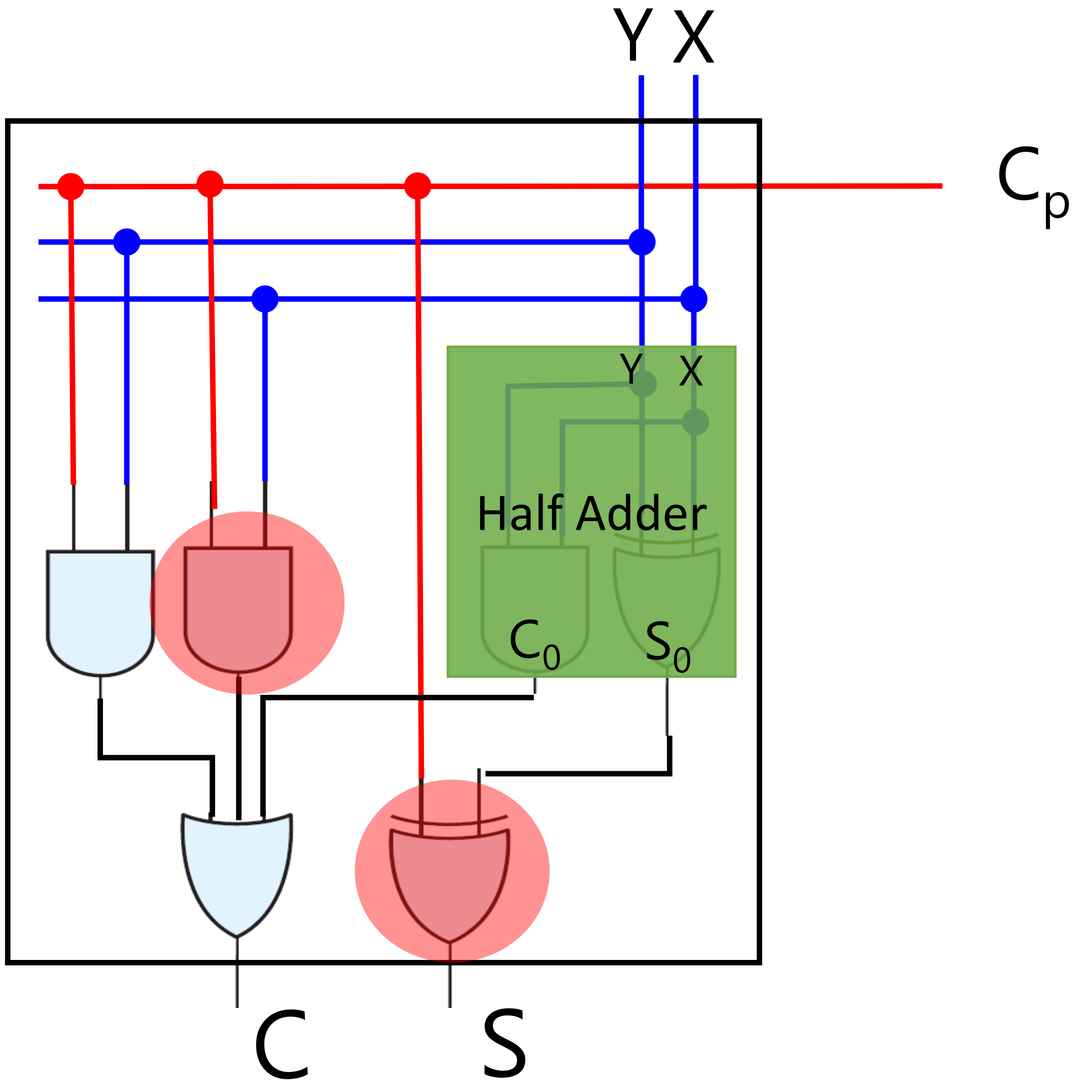
Y X
C_p
C
S


Y X
Cp
C
S


Y X
Cp
C
S

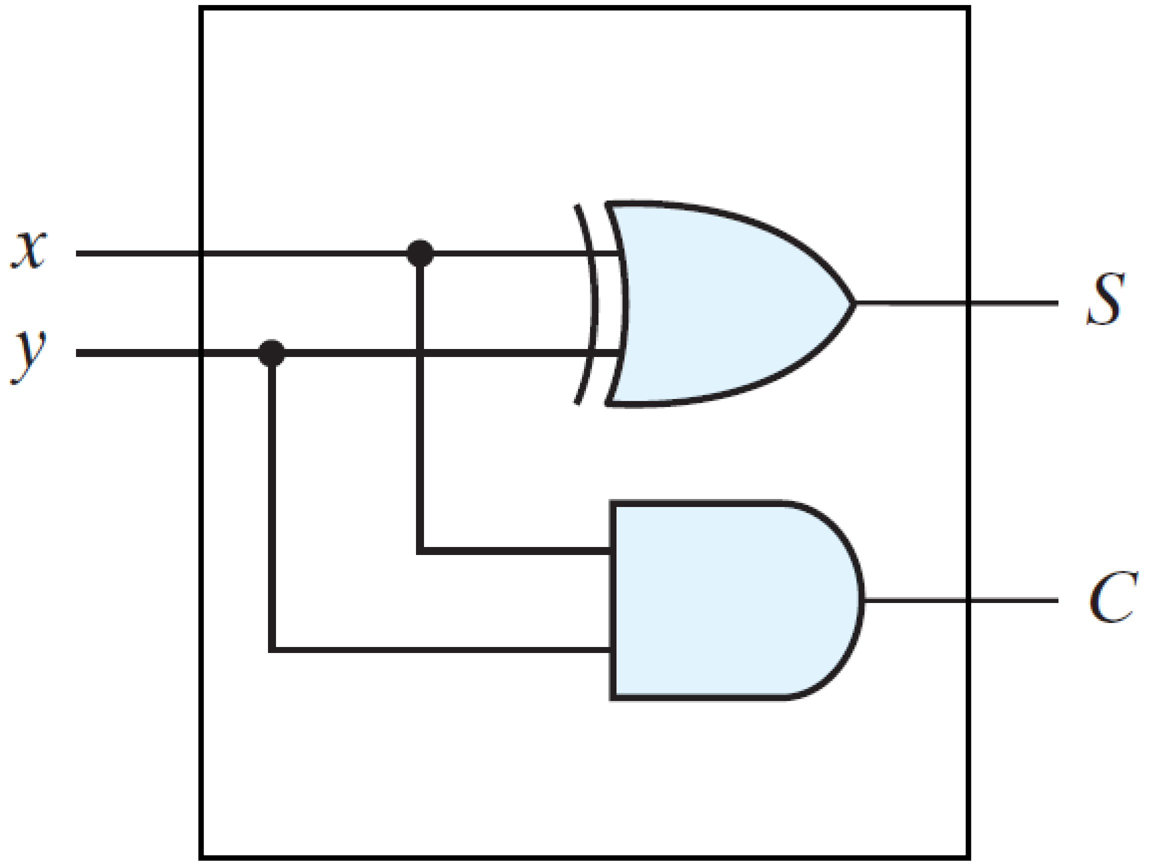


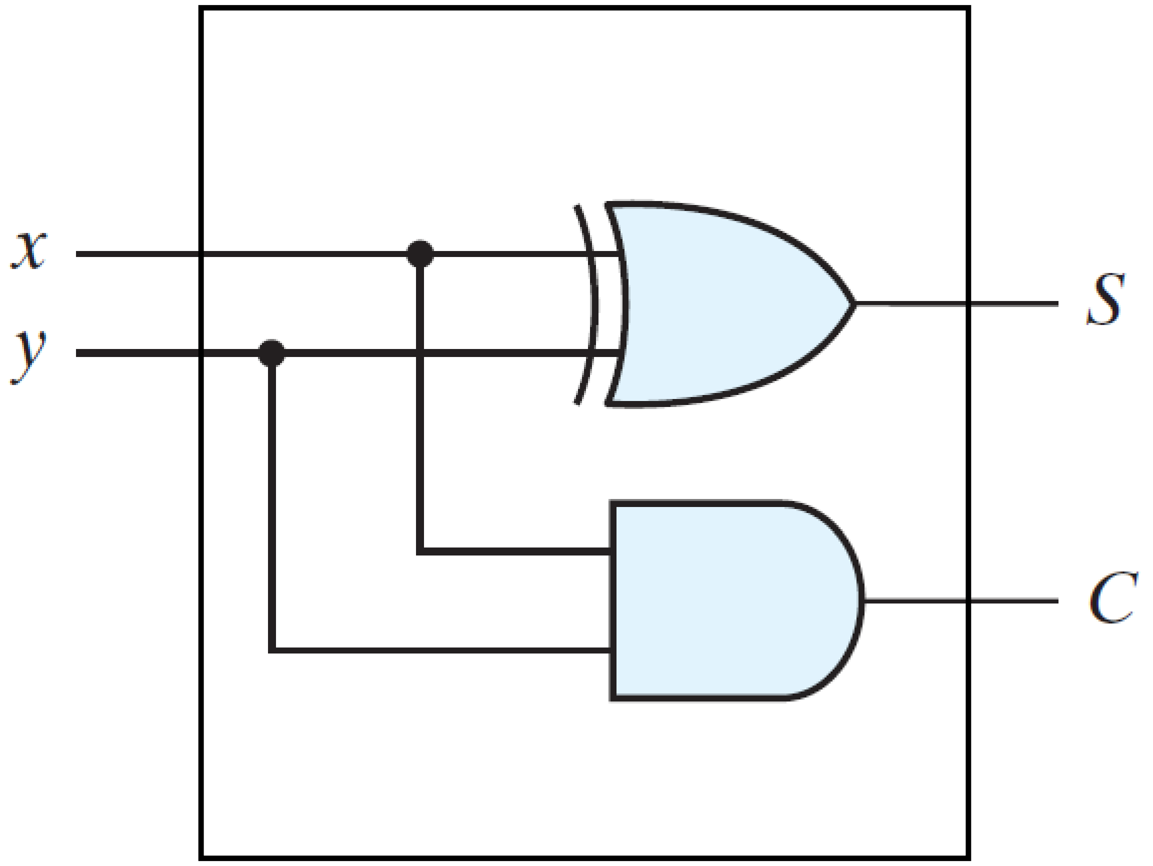
Full Adder = Half Adder + ...

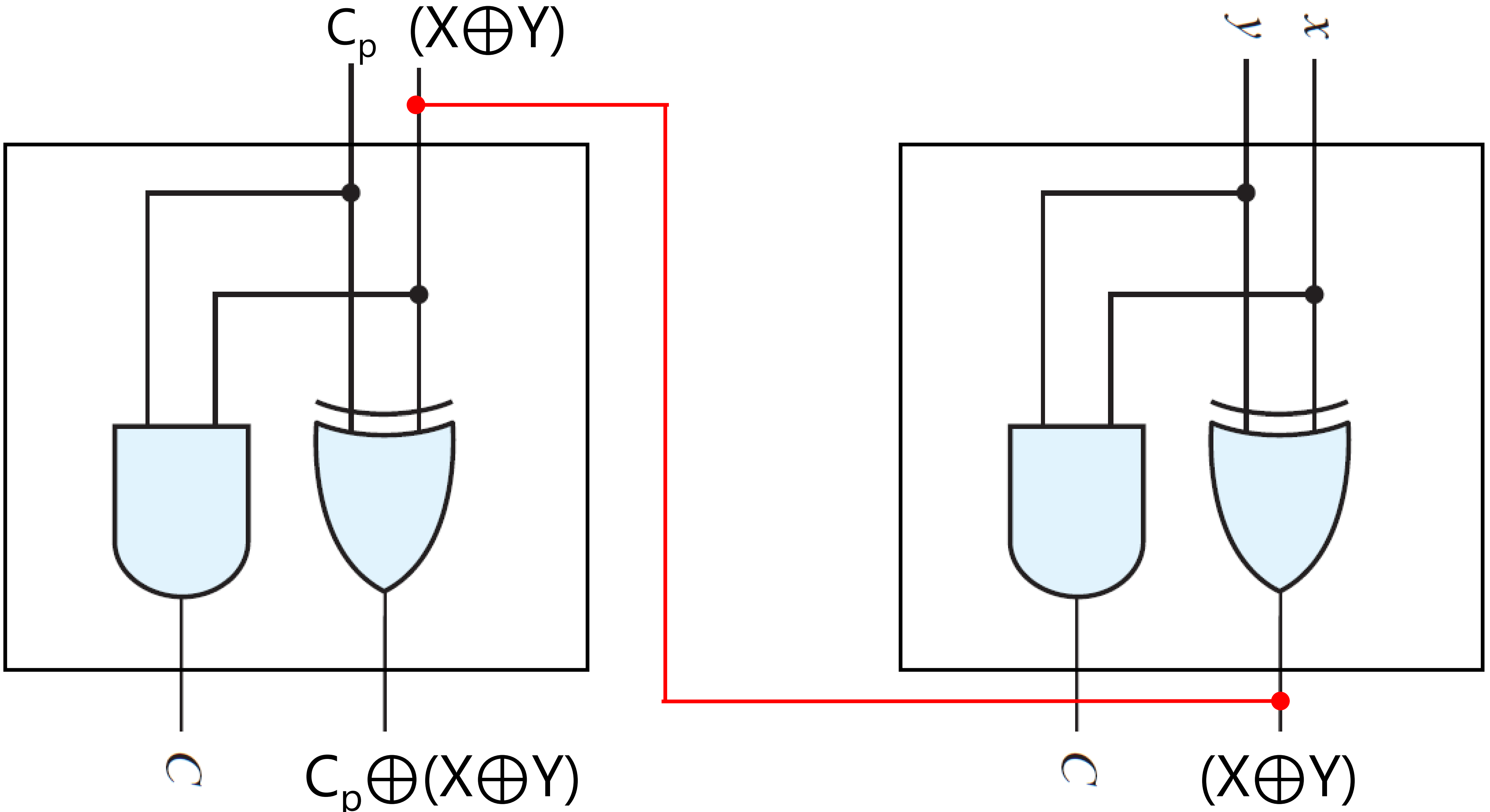


Full Adder $\stackrel{?}{=}$ 2 Half Adder + …

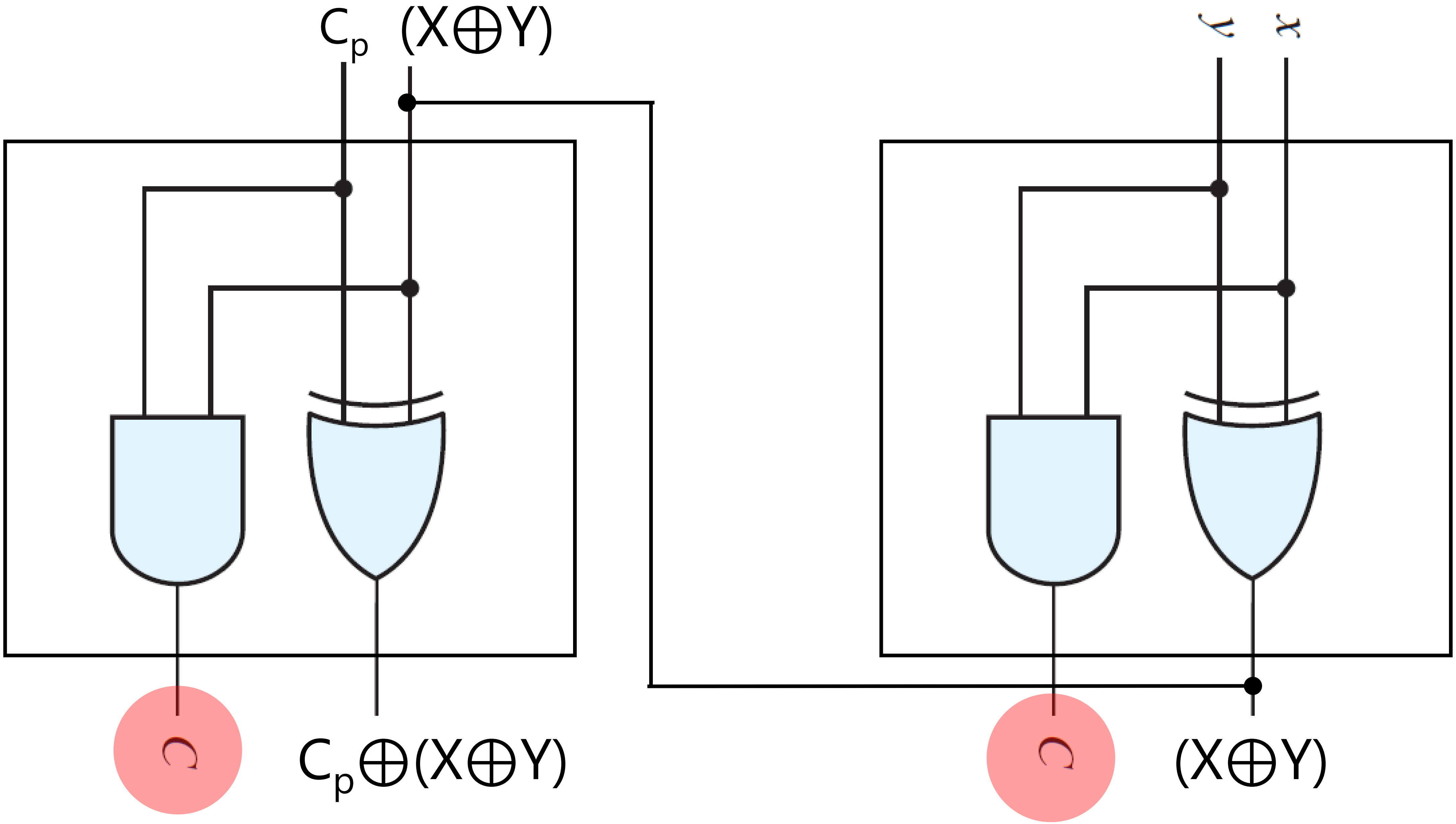
Full Adder $\propto$ 2 Half Adder


x
y
S
C

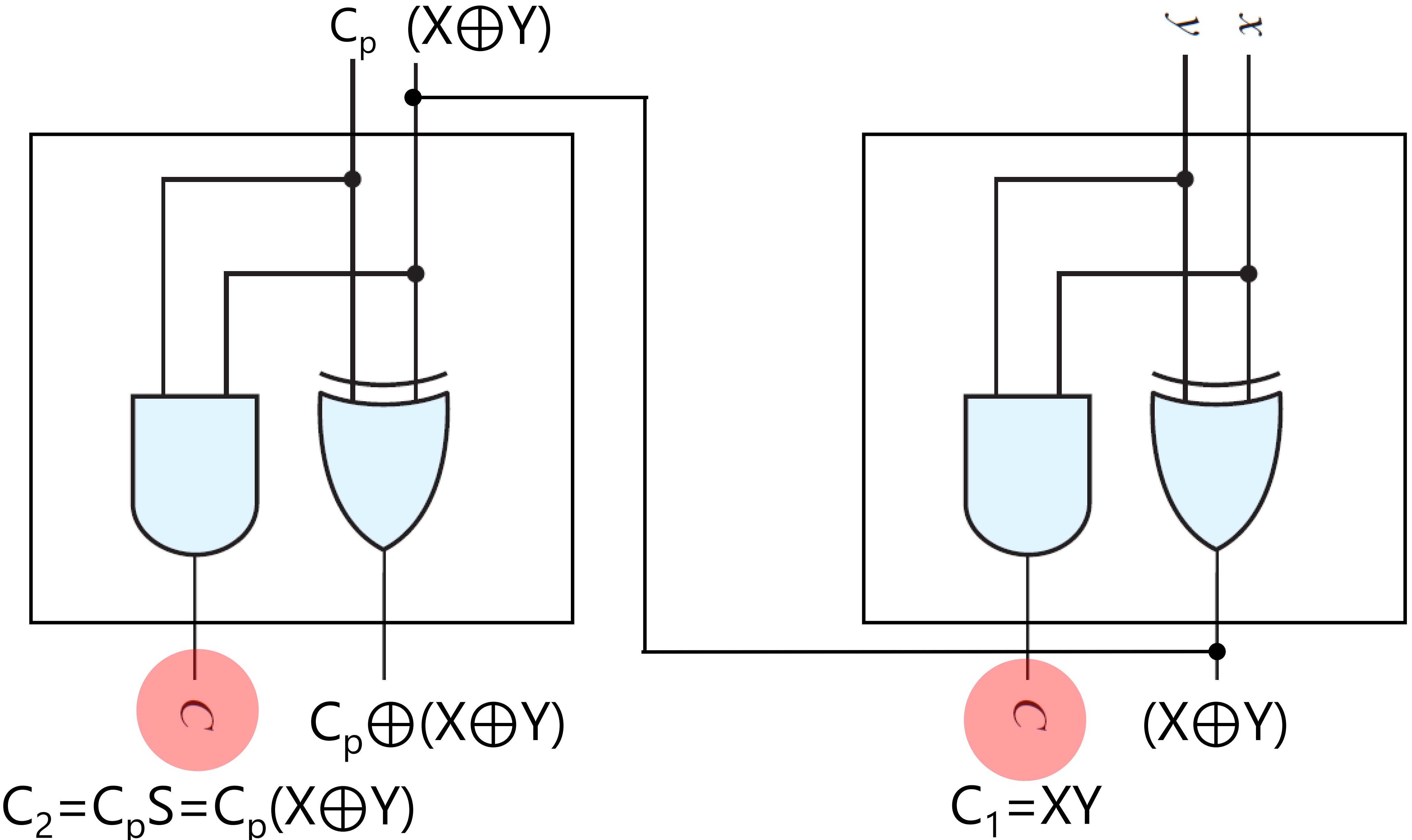

x
y
S
C



C_p
$(X \oplus Y)$
y
x
C
$C_p \oplus (X \oplus Y)$
C
$(X \oplus Y)$



Cp
(X⊕Y)
y
x
C
Cp⊕(X⊕Y)
C
(X⊕Y)



C_p
$(X \oplus Y)$
y
x
C
$C_p \oplus (X \oplus Y)$
C
$(X \oplus Y)$
$C_2 = C_pS = C_p(X \oplus Y)$
$C_1 = XY$

$$C_1 = XY$$
$$C_2 = C_p S = C_p(X \oplus Y)$$

$$\Rightarrow$$

$$C = XY + C_p X + C_p Y$$

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

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

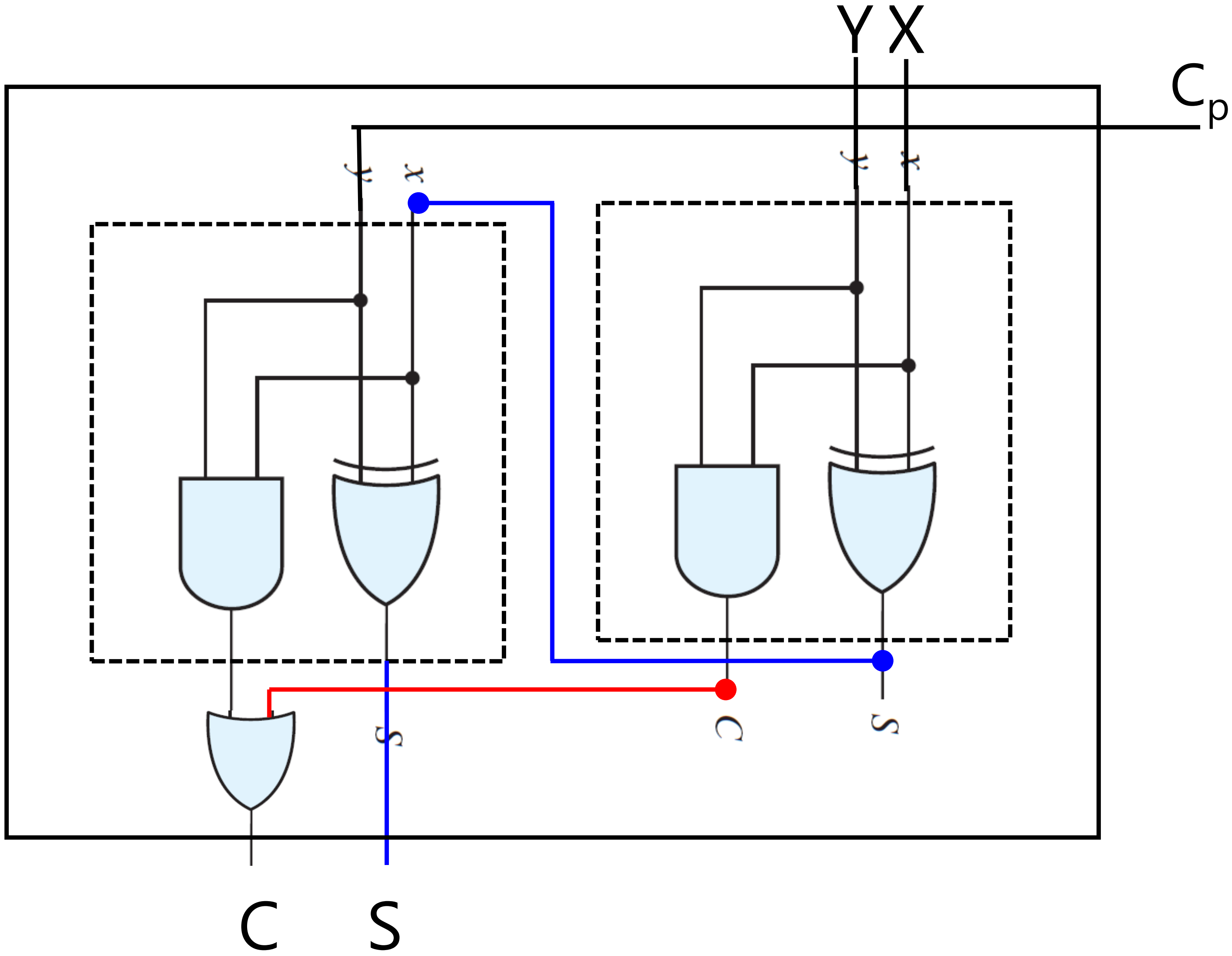
$$C=YX+C_pY'X+C_pYX'$$

$$C=YX+C_pY'X+C_pYX'$$

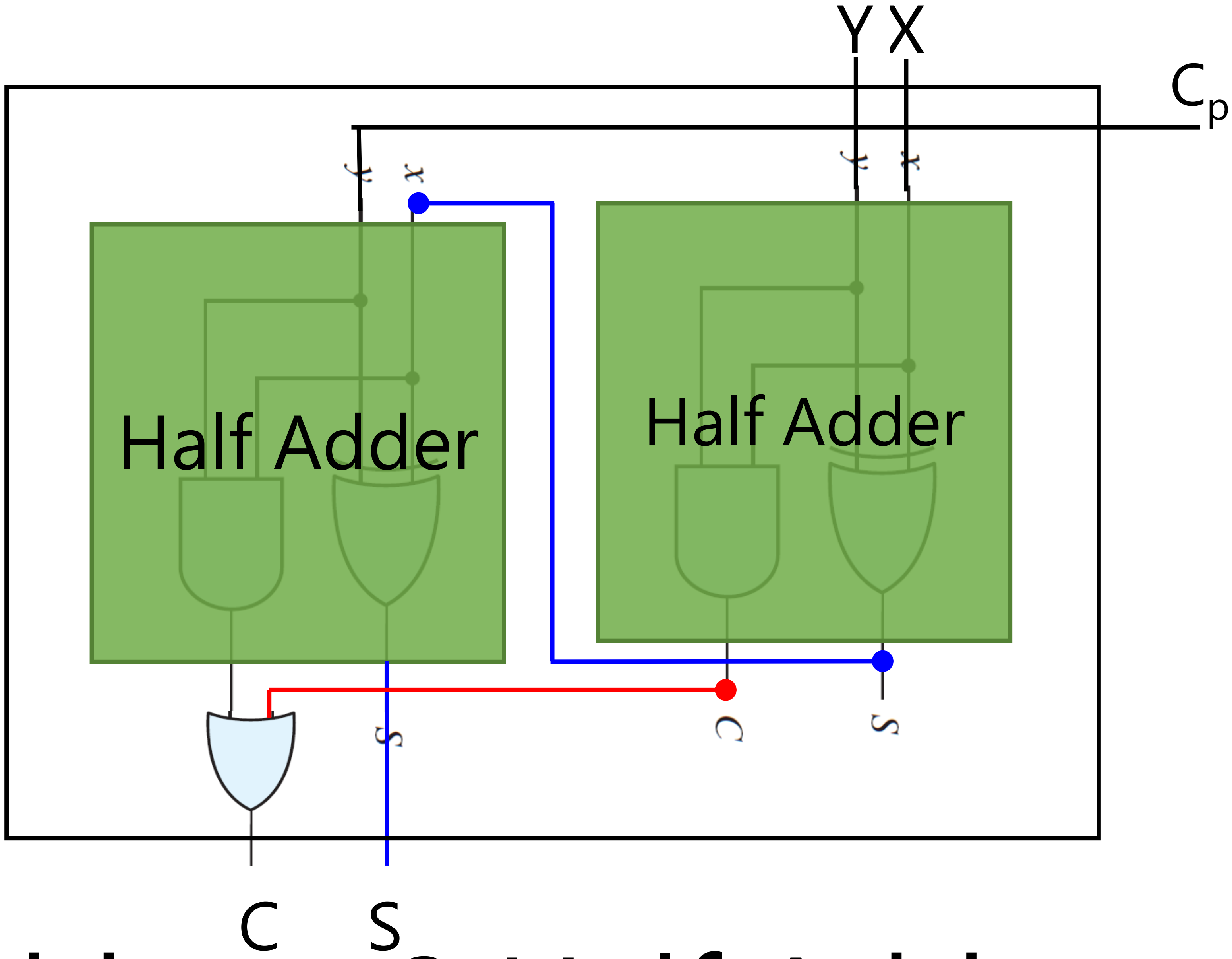
$$C=YX+C_p(Y'X+YX')$$

$$C=YX+C_p(X\oplus Y)$$

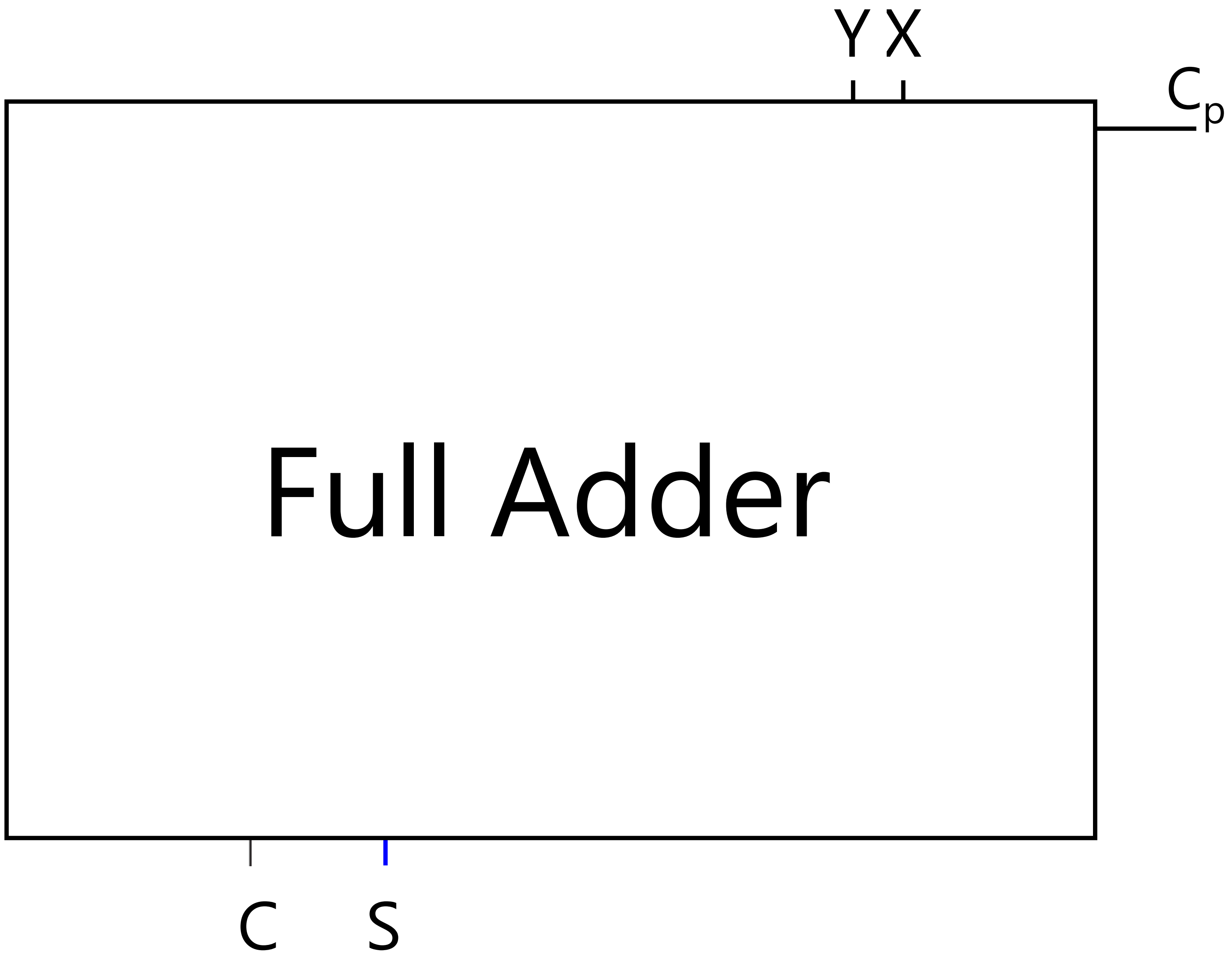
$$C=C_1+C_2$$



Y X
Cp
y x
y x
S
C
S
C S



Full Adder = 2 Half Adder + OR

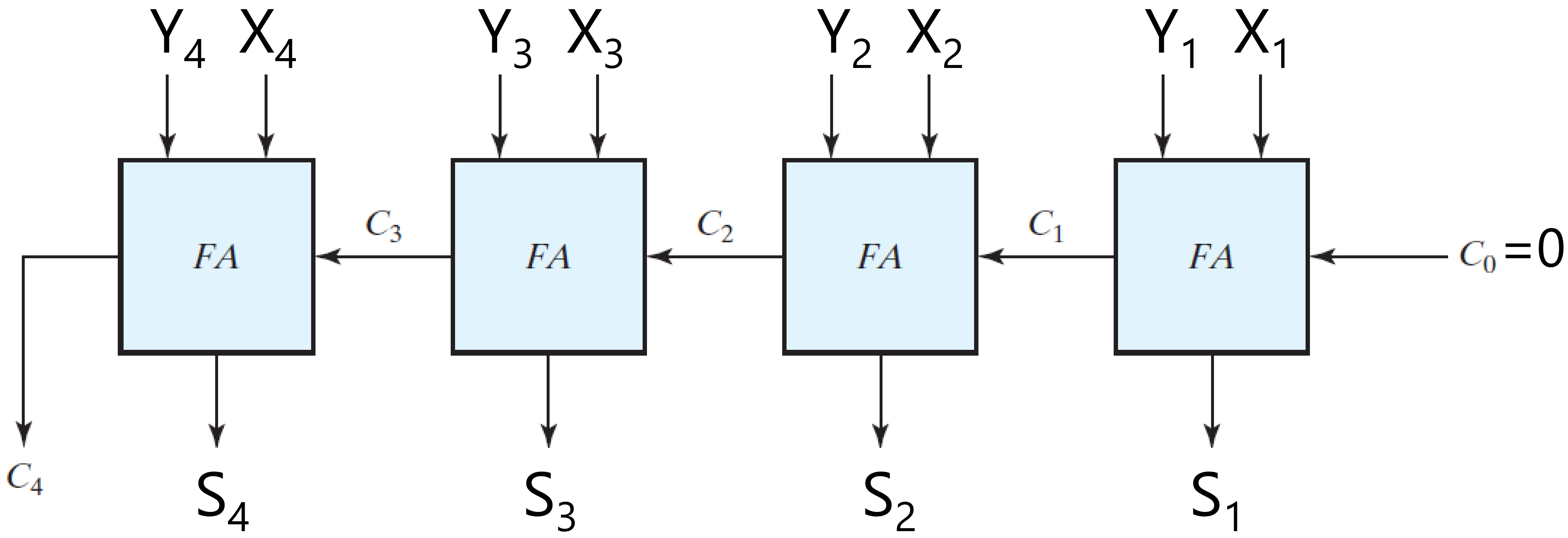


Y
X
C_p
Full Adder
C
S

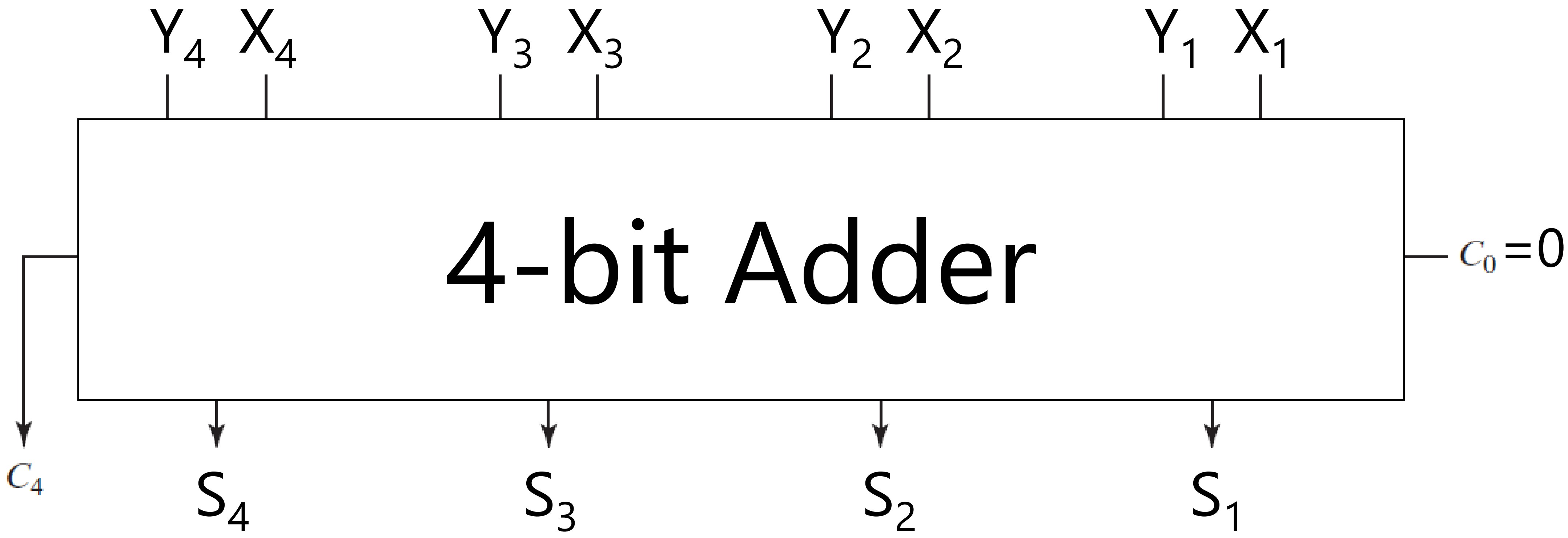
Design a logic circuit that adds two binary numbers!

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

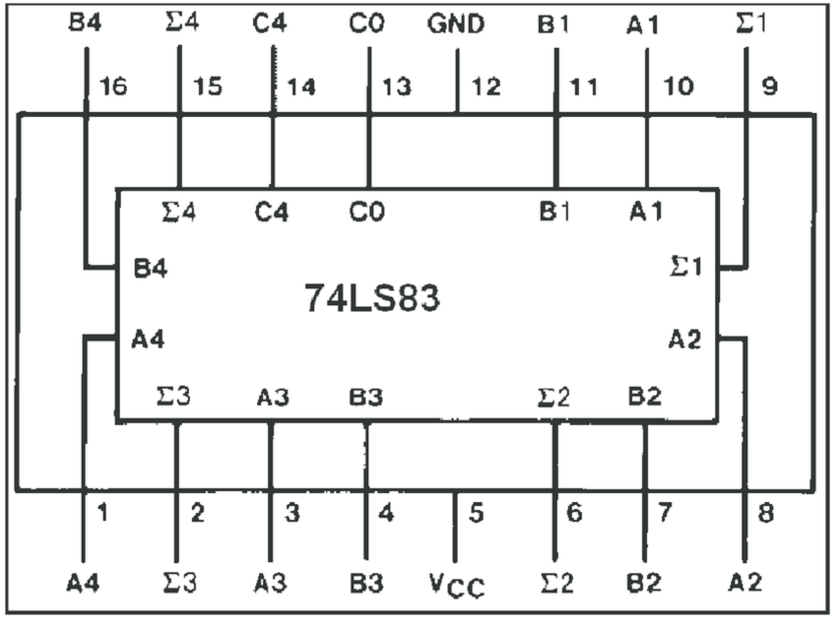
$C_0 = 0$



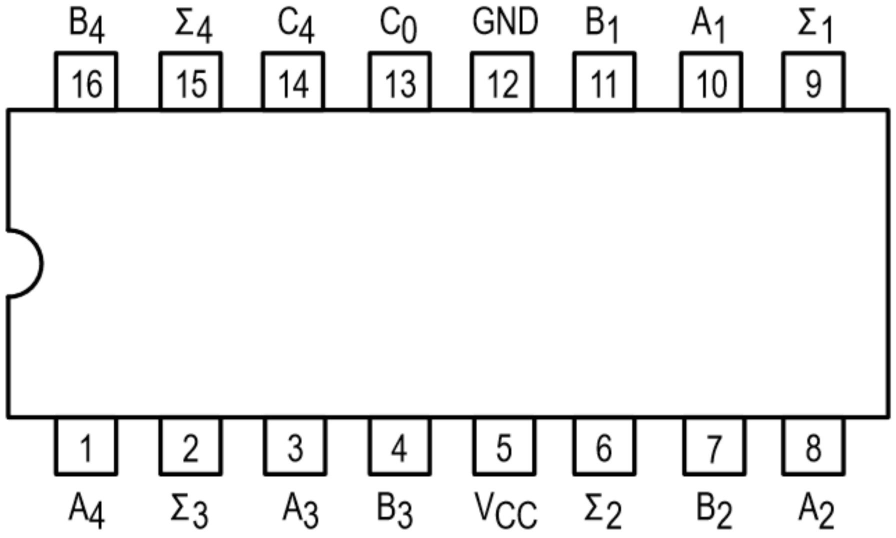
Y_4 X_4
Y_3 X_3
Y_2 X_2
Y_1 X_1
FA
FA
FA
FA
C_3
C_2
C_1
$C_0 = 0$
C_4
S_4
S_3
S_2
S_1



Y4 X4
Y3 X3
Y2 X2
Y1 X1
4-bit Adder
C0 =0
C4
S4
S3
S2
S1



74LS83 pinout



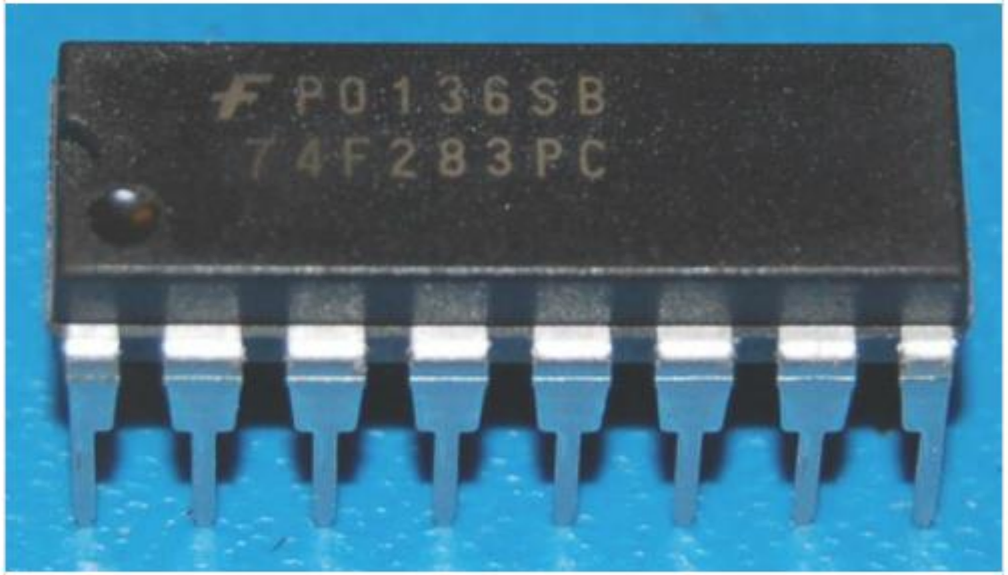
Vcc	5.5V max, 5V Typical
$A_1 - A_4$	Operand A Inputs
$B_1 - B_4$	Operand B Inputs
C_0	Carry Input
$\Sigma_1 - \Sigma_4$	Sum Outputs (Note b)
C_4	Carry Output (Note b)

Hi! Sign in or register eBay Deals Brand Outlet Help & Contact Holidays At Home English Sell Watch List My eBay

Shop by category Search for anything All Categories Search Advanced

eBay > Business & Industrial > Electrical Equipment & Supplies > Other Electrical Equipment & Supplies Share

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

MONEY BACK GUARANTEE

Qty: 1

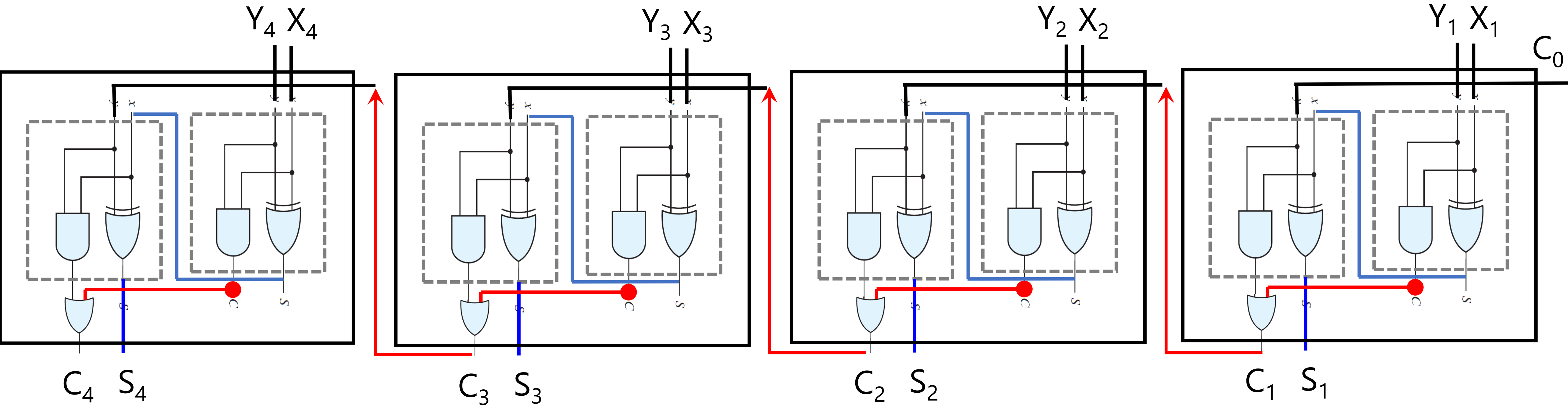
Buy It Now

Add to cart

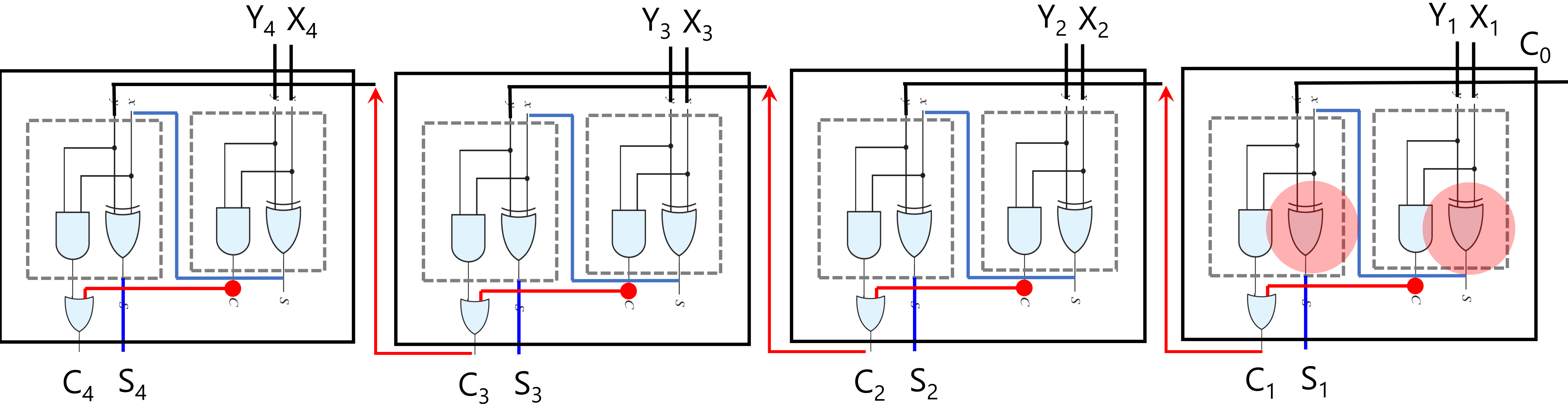
Watch

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

Carry Propagation



If gate delay is Δt, how long does it take to see the $S=S_4S_3S_2S_1$?



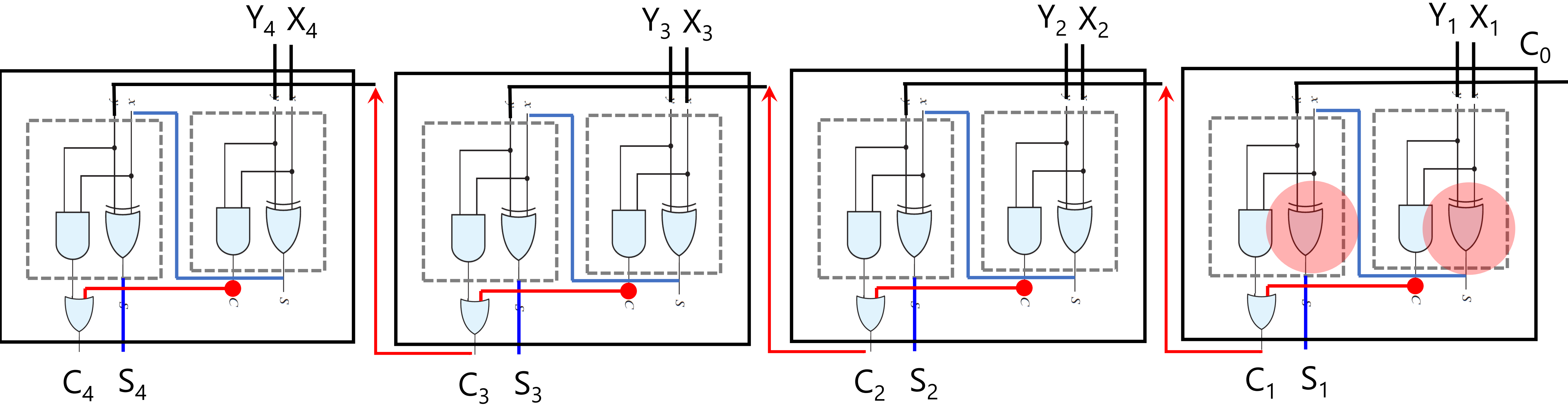
If gate delay is Δt, how long does it take to see the $S = S_4 S_3 S_2 S_1$?

S_1: $2 \times \Delta t$

S_2: $2 \times \Delta t + C_1$

S_3: $2 \times \Delta t + C_2$

S_4: $2 \times \Delta t + C_3$



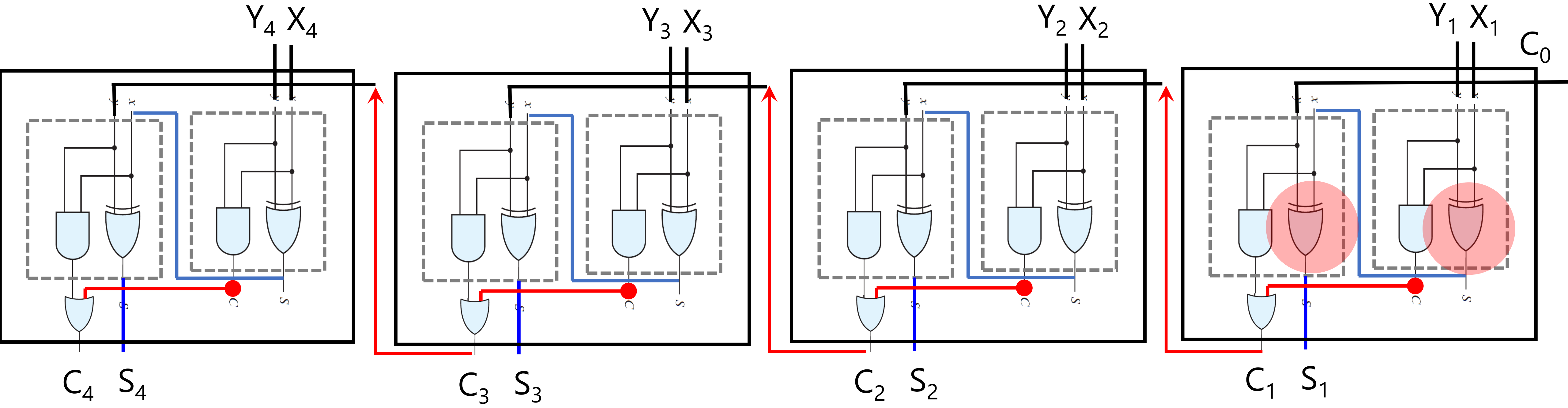
If gate delay is Δt, how long does it take to see the $S=S_4S_3S_2S_1$?

S_1: $2 \times \Delta t$

S_2: $2 \times \Delta t + C_1$

S_3: $2 \times \Delta t + C_2$

S_4: $\underbrace{2 \times \Delta t}_{\text{In parallel}} + C_3$



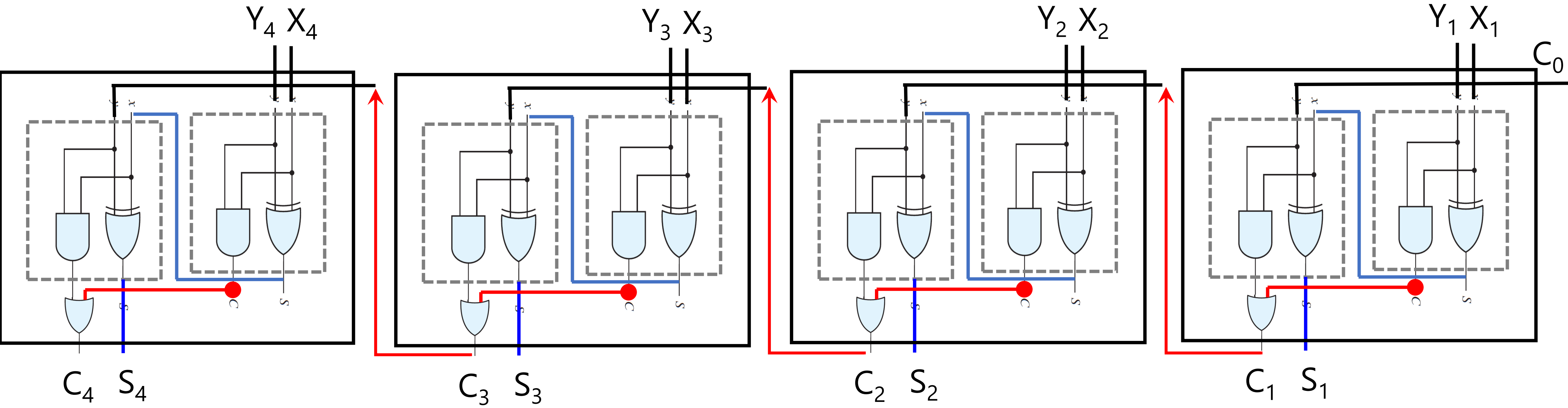
If gate delay is Δt, how long does it take to see the $S=S_4S_3S_2S_1$?

S_1: $2 \times \Delta t$

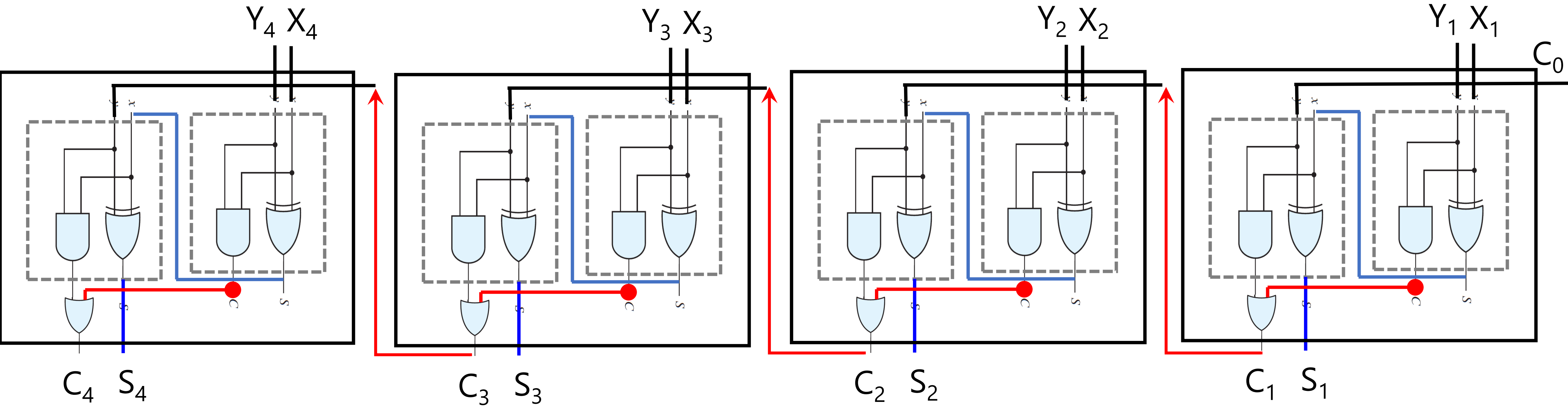
S_2: $2 \times \Delta t + C_1$

S_3: $2 \times \Delta t + C_2$

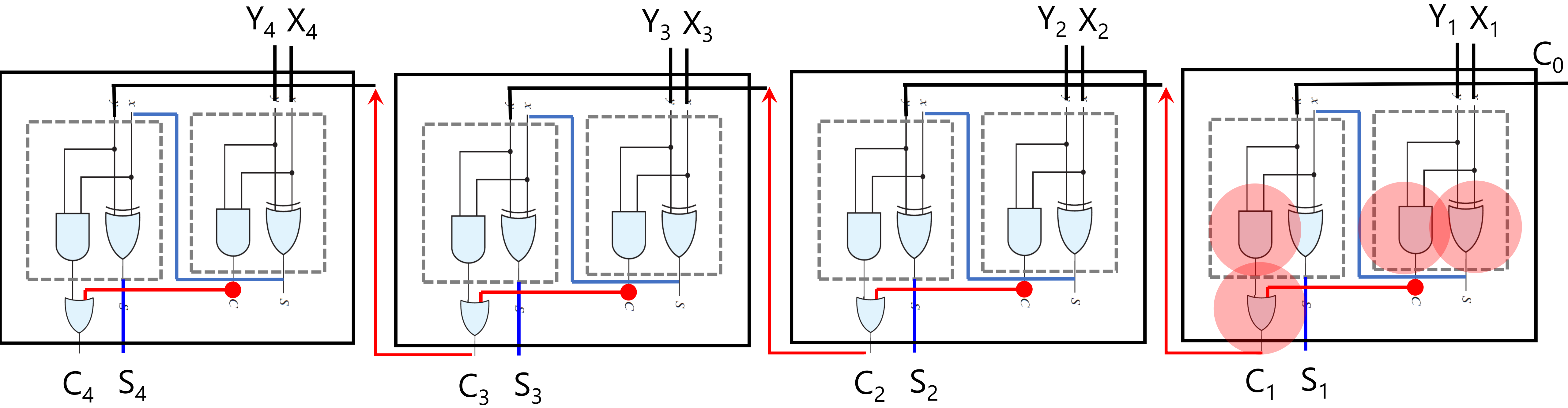
S_4: $\underbrace{2 \times \Delta t}_{\text{In parallel}} + C_3$



If gate delay is Δt, how long does it take to see the $S=S_n \ldots S_3 S_2 S_1$?

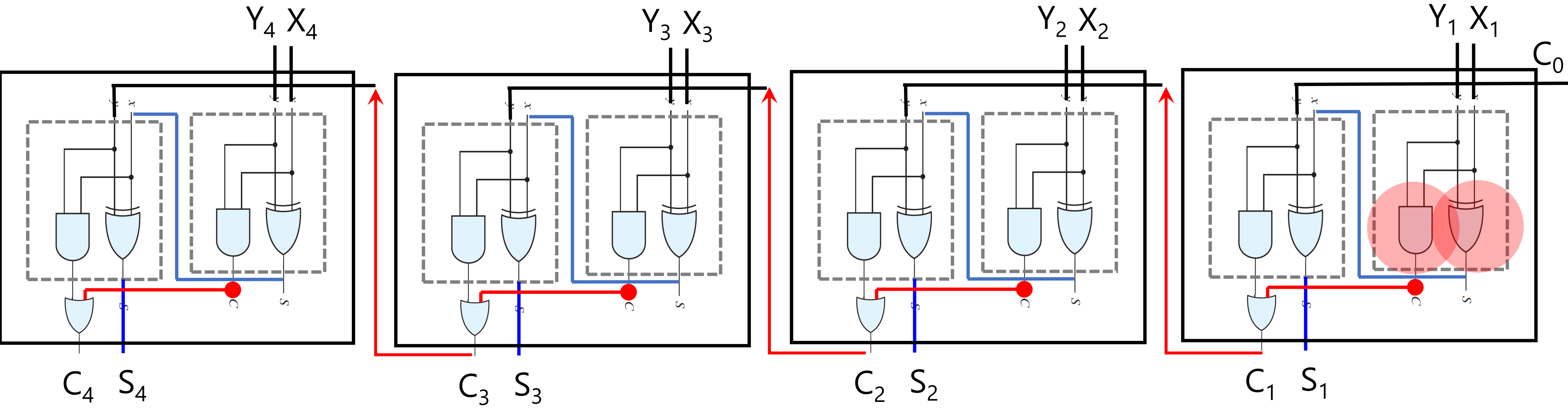


If gate delay is Δt, how long does it take to see the C_4?



If gate delay is Δt, how long does it take to see the C_4?

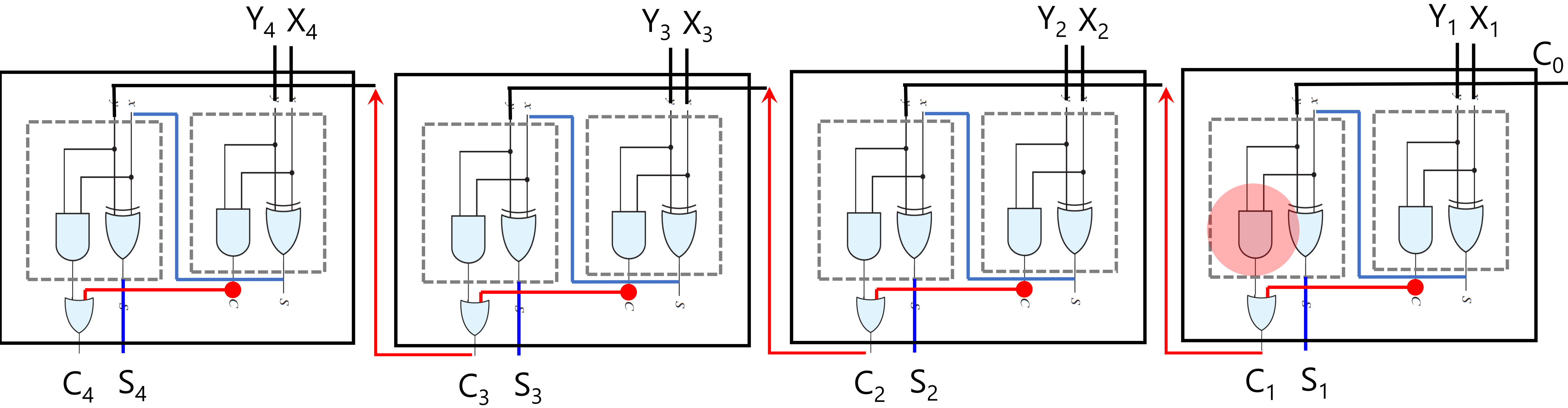
$C_1 = Y_1 X_1 + C_0(Y_1 \oplus X_1)$



If gate delay is Δt, how long does it take to see the C_4?

$C_1 = Y_1X_1 + C_0(Y_1 \oplus X_1) \rightarrow 1 \times \Delta t$

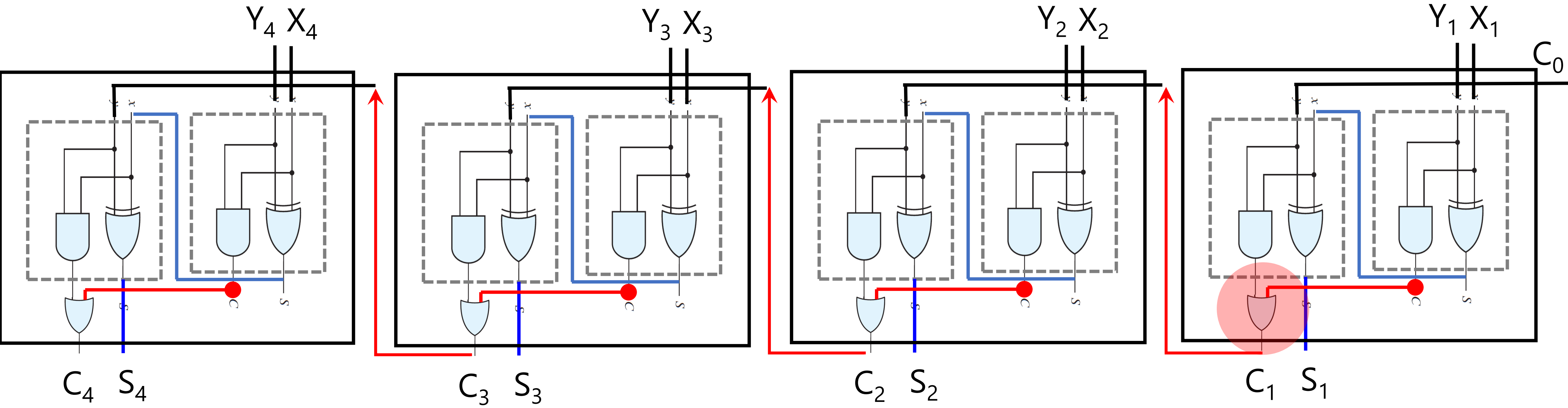
AND and XOR can be done in parallel.



If gate delay is Δt, how long does it take to see the C_4?

$C_1 = Y_1 X_1 + C_0\,(Y_1 \oplus X_1) \rightarrow 1 \times \Delta t + 1 \times \Delta t$

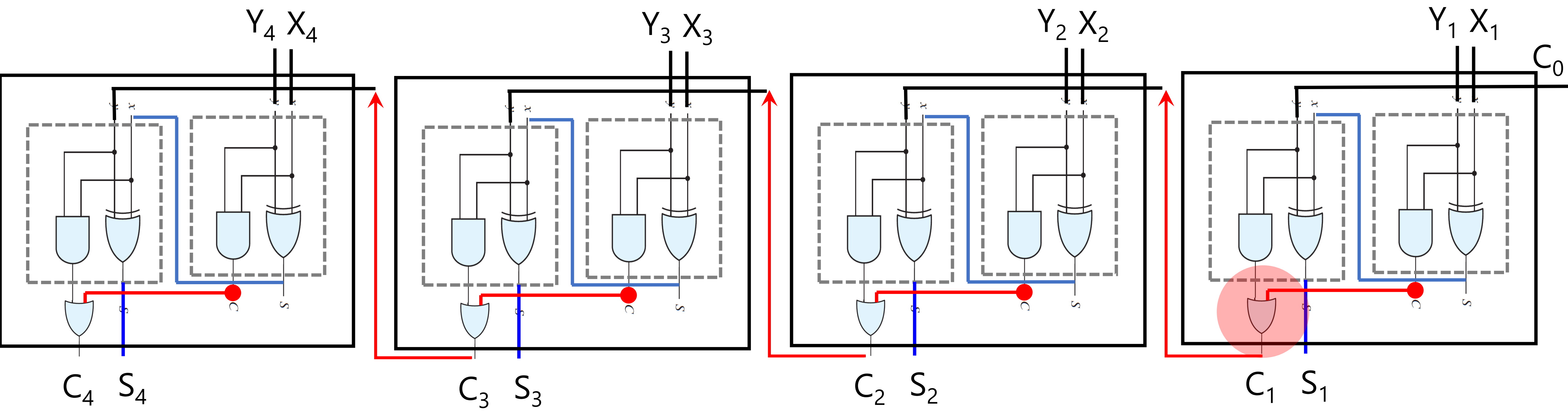
C_0 AND



If gate delay is Δt, how long does it take to see the C_4?

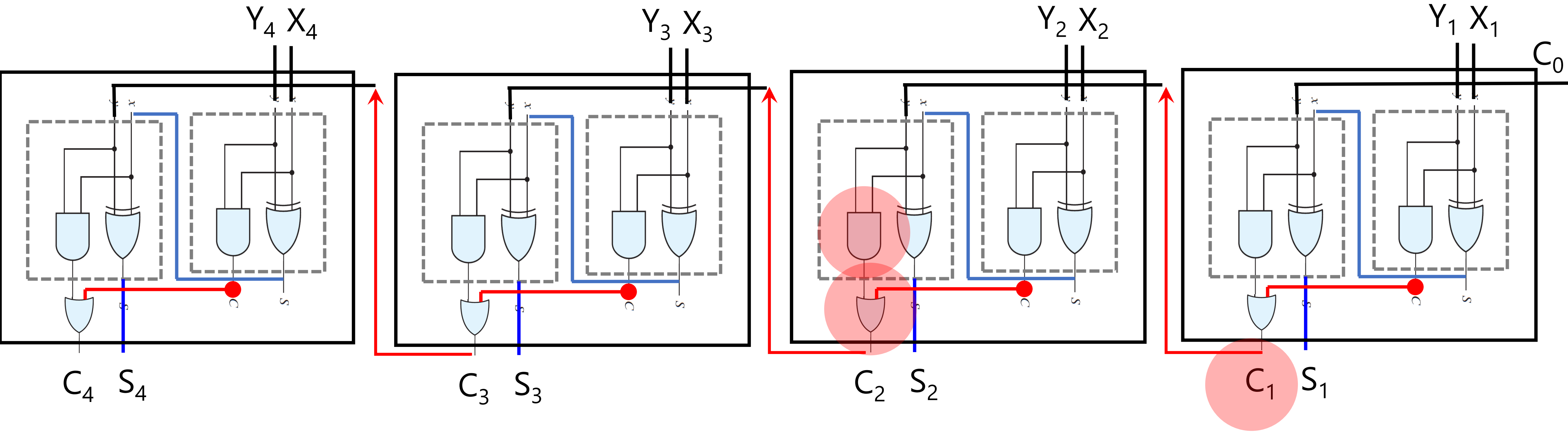
$C_1 = Y_1 X_1 + C_0 (Y_1 \oplus X_1) \rightarrow 1 \times \Delta t + 1 \times \Delta t + 1 \times \Delta t$

OR



If gate delay is Δt, how long does it take to see the C_4?

$C_1 = Y_1X_1 + C_0(Y_1 \oplus X_1) \rightarrow 1 \times \Delta t + 1 \times \Delta t + 1 \times \Delta t = 3 \times \Delta t$

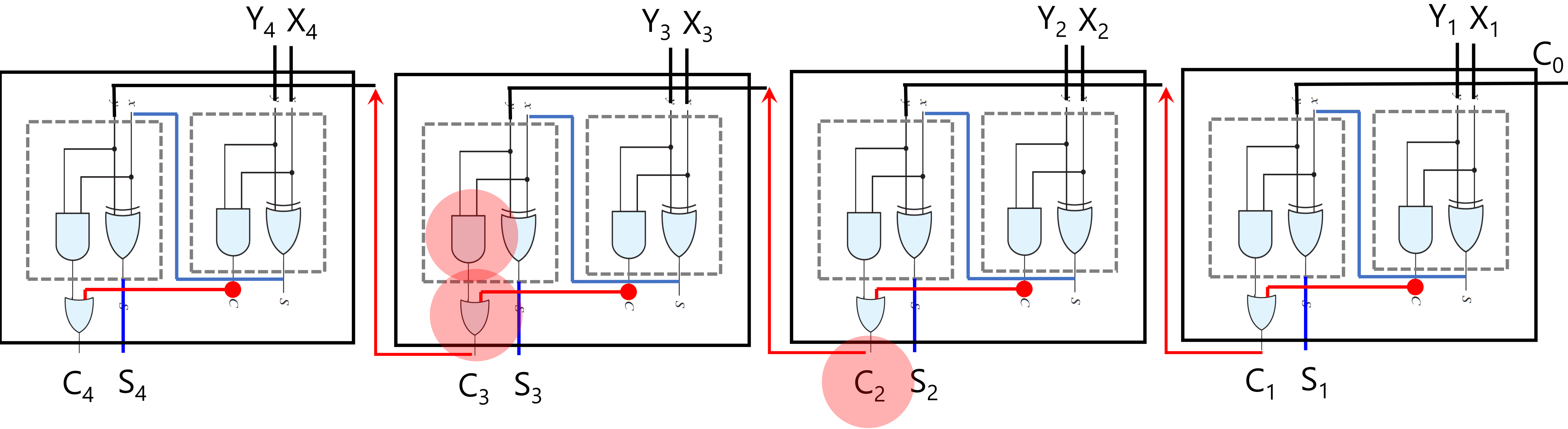


If gate delay is Δt, how long does it take to see the C_4?

$$C_2 = Y_2X_2 + {\color{red}C_1}(Y_2 \oplus X_2) \rightarrow C_1 + 2 \times \Delta t = 5 \times \Delta t$$

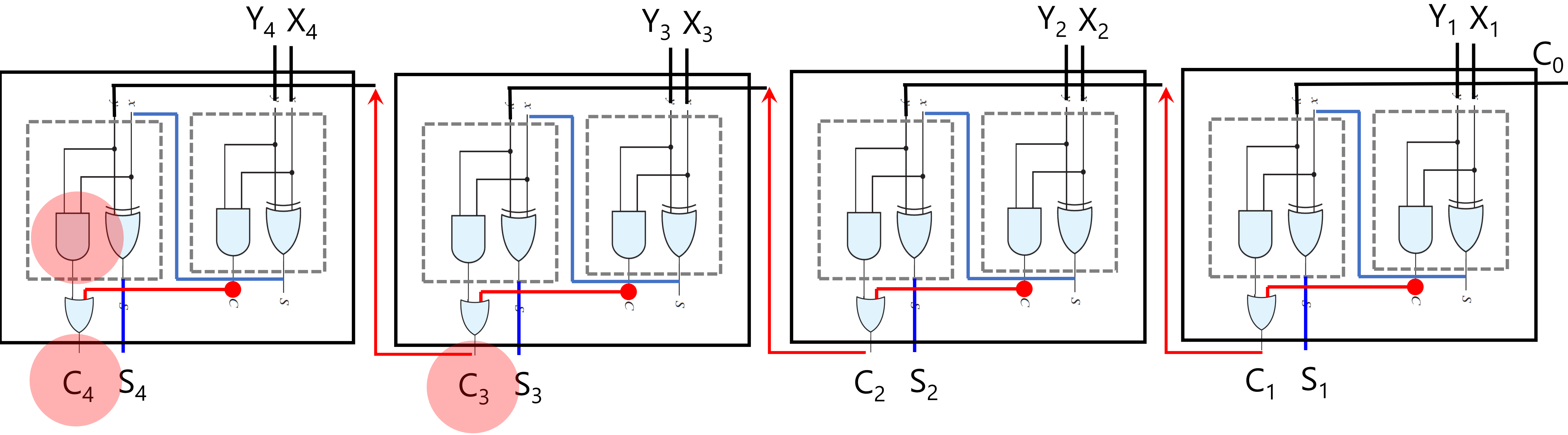
C_1 AND then OR

In the meantime, in parallel, we can do the Y_2X_2 and $Y_2 \oplus X_2$



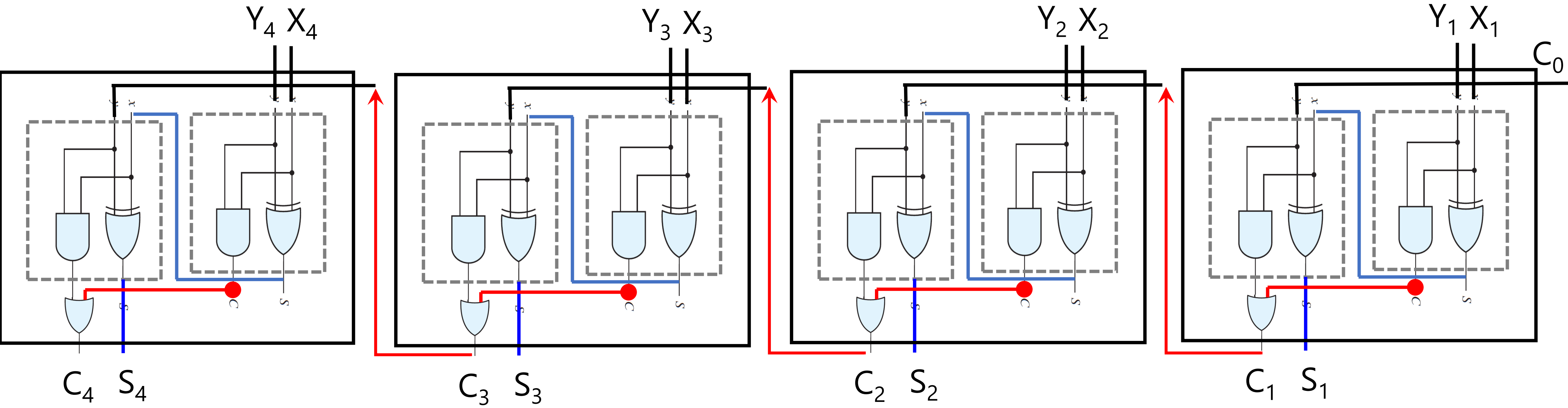
If gate delay is Δt, how long does it take to see the C_4?

$C_3 = Y_3X_3 + {\color{red}C_2}(Y_3 \oplus X_3) \rightarrow C_2 + 2 \times \Delta t = 7 \times \Delta t$



If gate delay is Δt, how long does it take to see the C_4?

$C_4 = Y_4X_4 + {\color{red}C_3}\,(Y_4 \oplus X_4) \rightarrow C_3 + 2 \times \Delta t = 9 \times \Delta t$



If gate delay is Δt, how long does it take to see the C_n?

Carry Lookahead

$C_{1:n}$ → Constant Delay

Book Page 138-141

Binary Adder

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

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

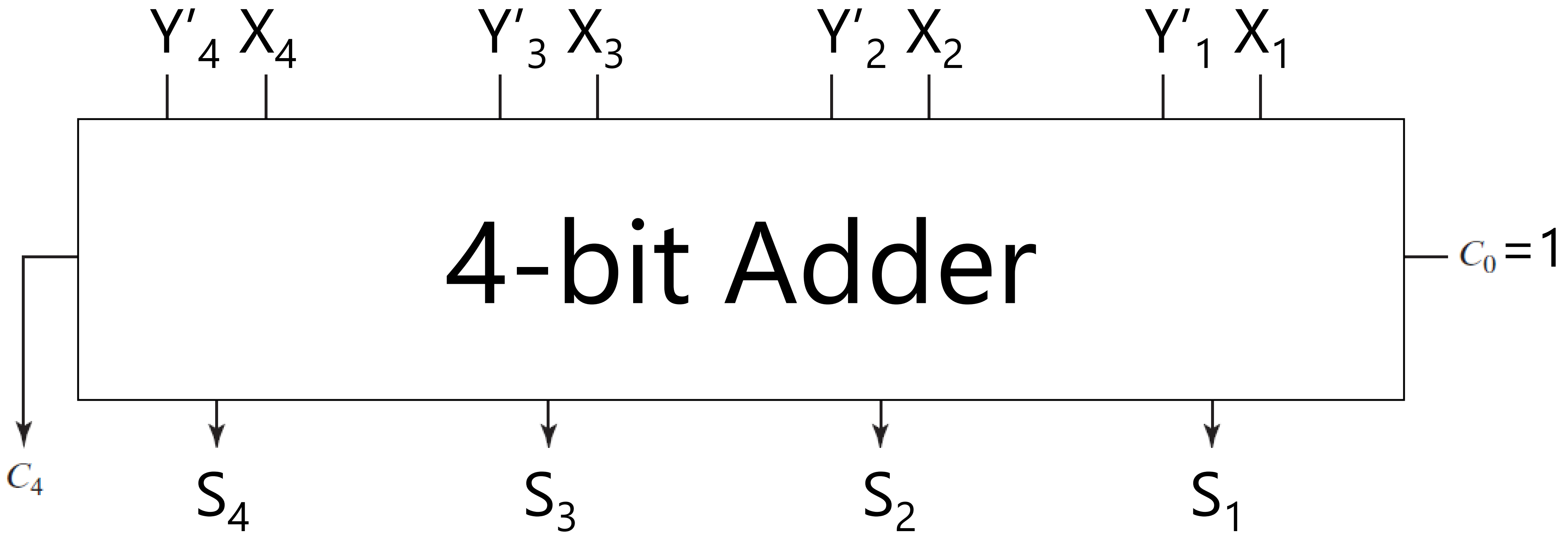
$$X + Y' + 1$$

Subtraction in Signed-2's-Complement

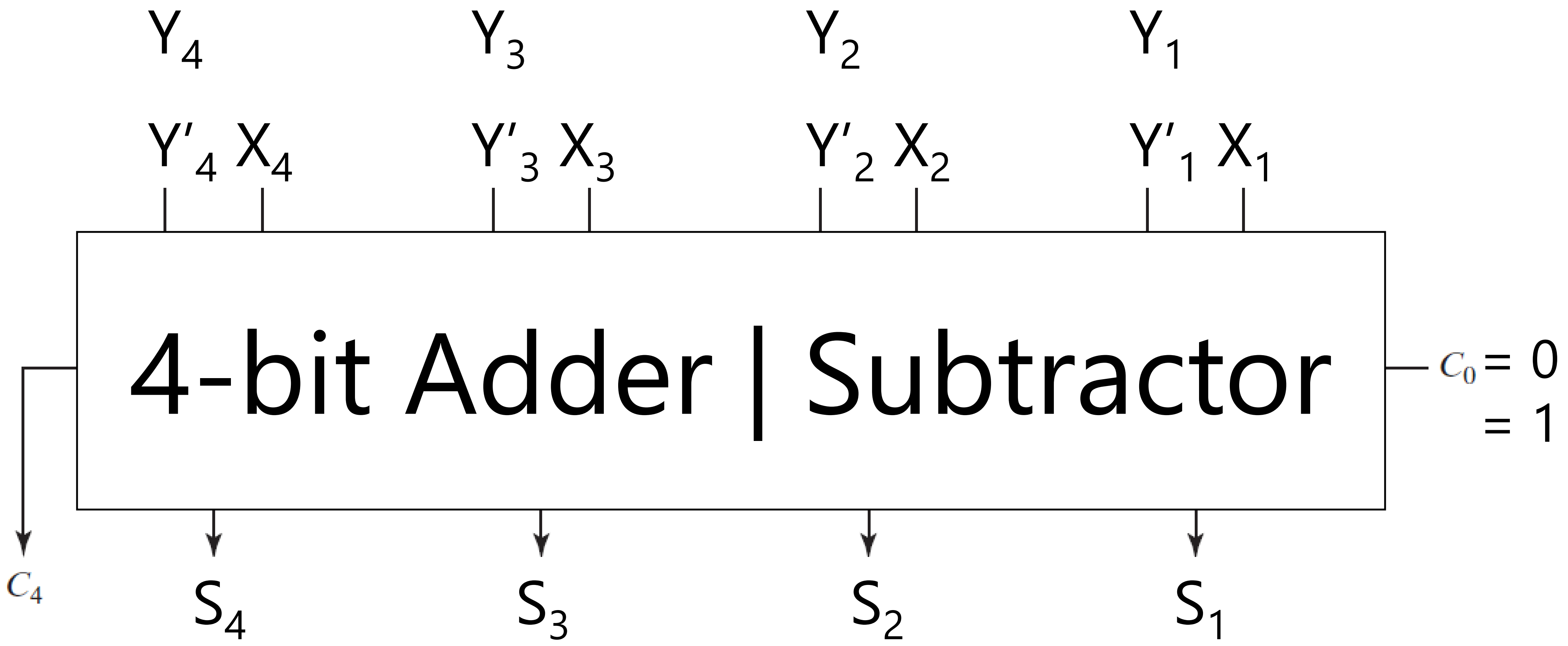
bitwise

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

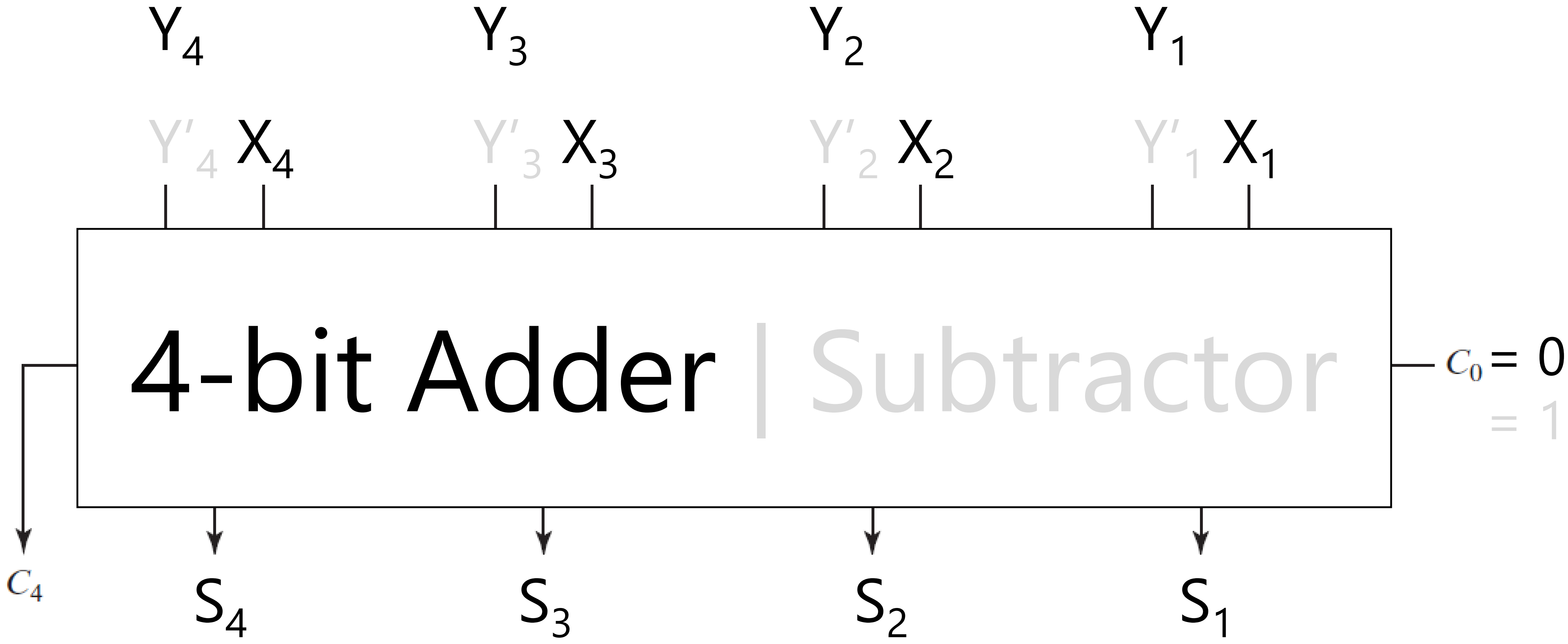
Subtraction in Signed-2's-Complement



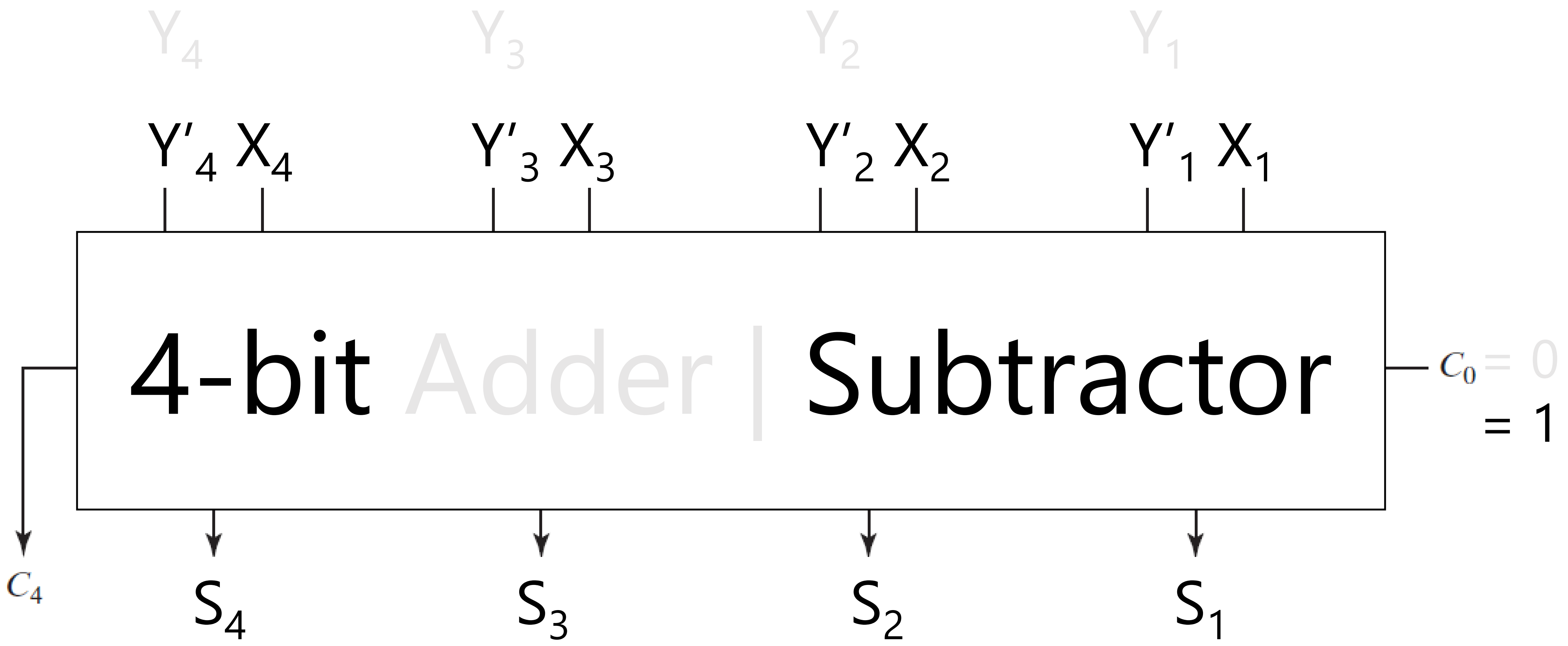
Y'4 X4
Y'3 X3
Y'2 X2
Y'1 X1
4-bit Adder
C0 =1
C4
S4
S3
S2
S1



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

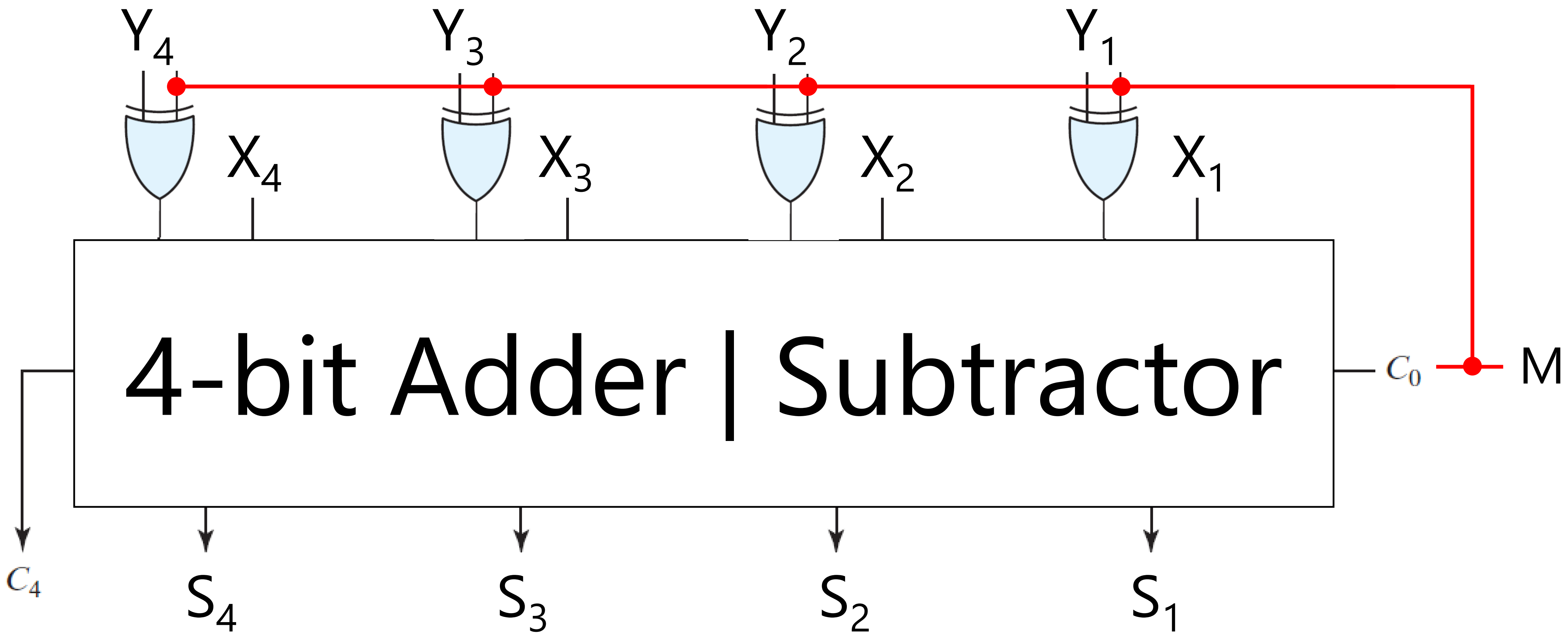


Y4
Y3
Y2
Y1
Y'4 X4
Y'3 X3
Y'2 X2
Y'1 X1
4-bit Adder | Subtractor
C0 = 0
= 1
C4
S4
S3
S2
S1

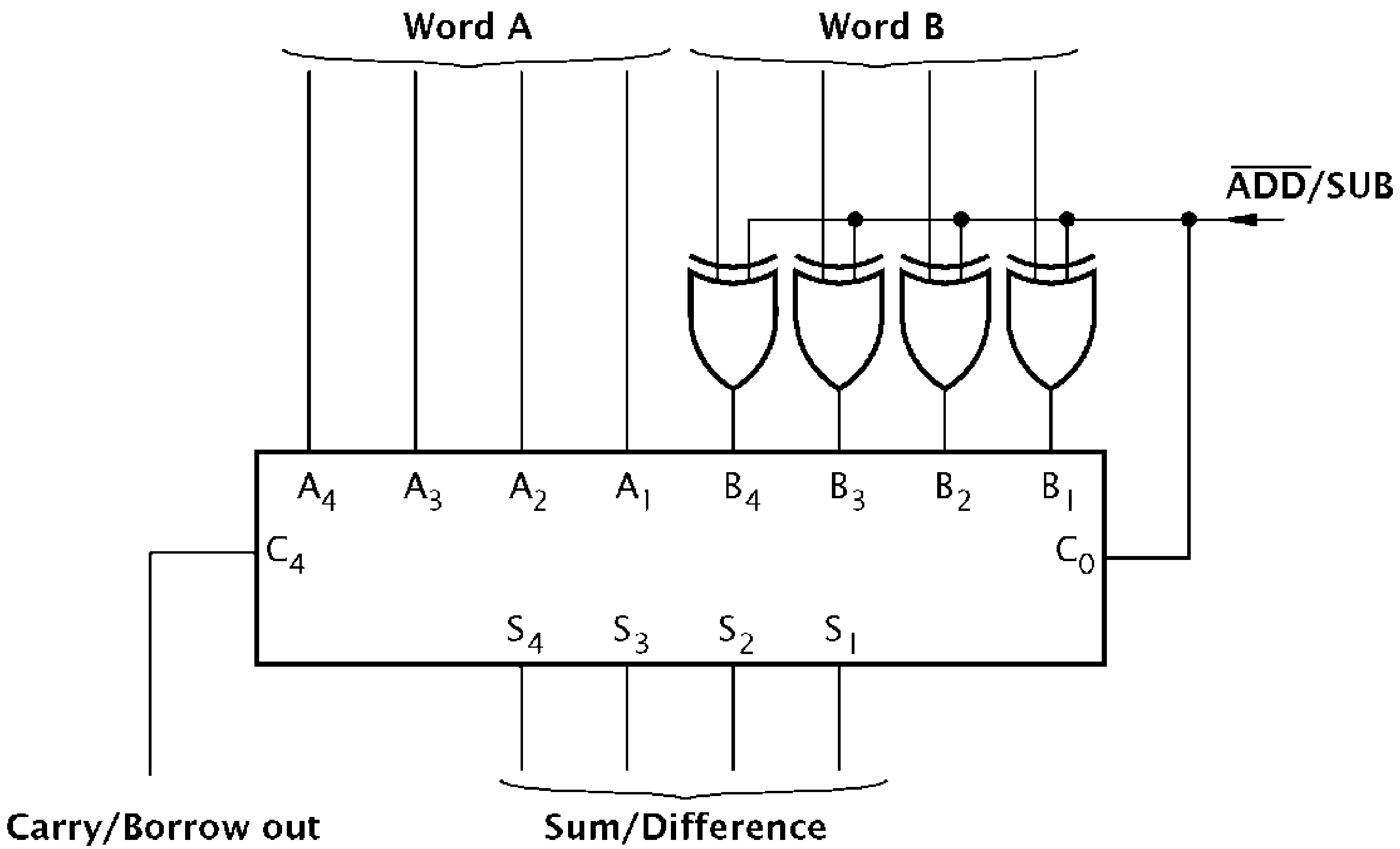

Y4
Y3
Y2
Y1
Y'4 X4
Y'3 X3
Y'2 X2
Y'1 X1
4-bit Adder | Subtractor
C0 = 0
= 1
C4
S4
S3
S2
S1

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

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



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

Binary Subtractor

Unsigned?

Overflow

Signed-2’s-Complement