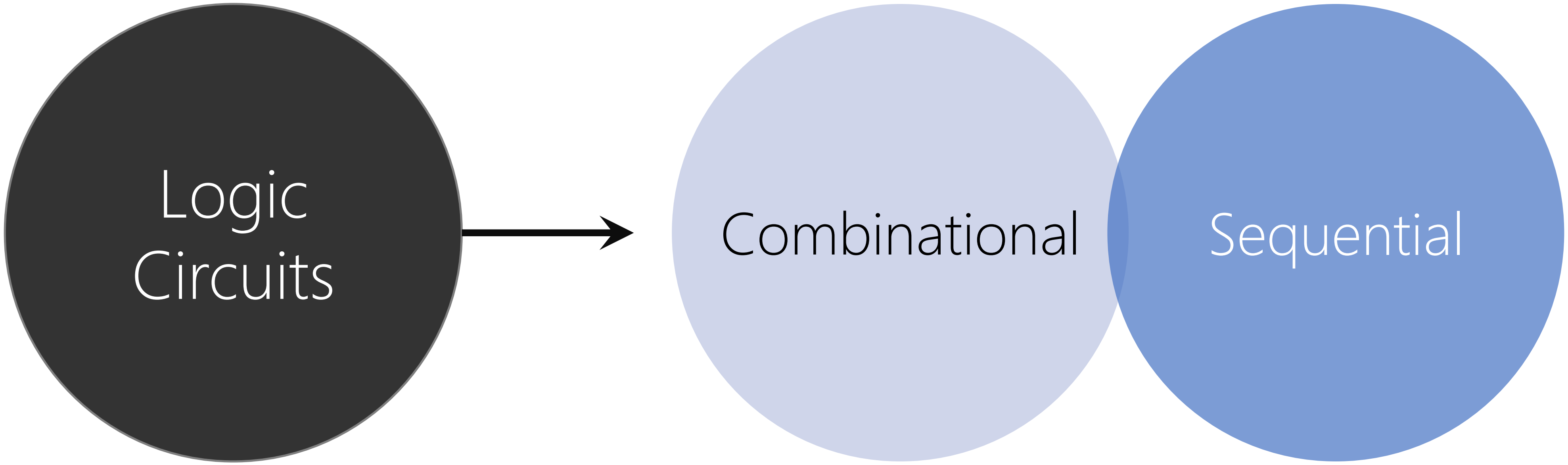

Logic Circuits
Combinational
Sequential

Chapter 4 Combinational Logic

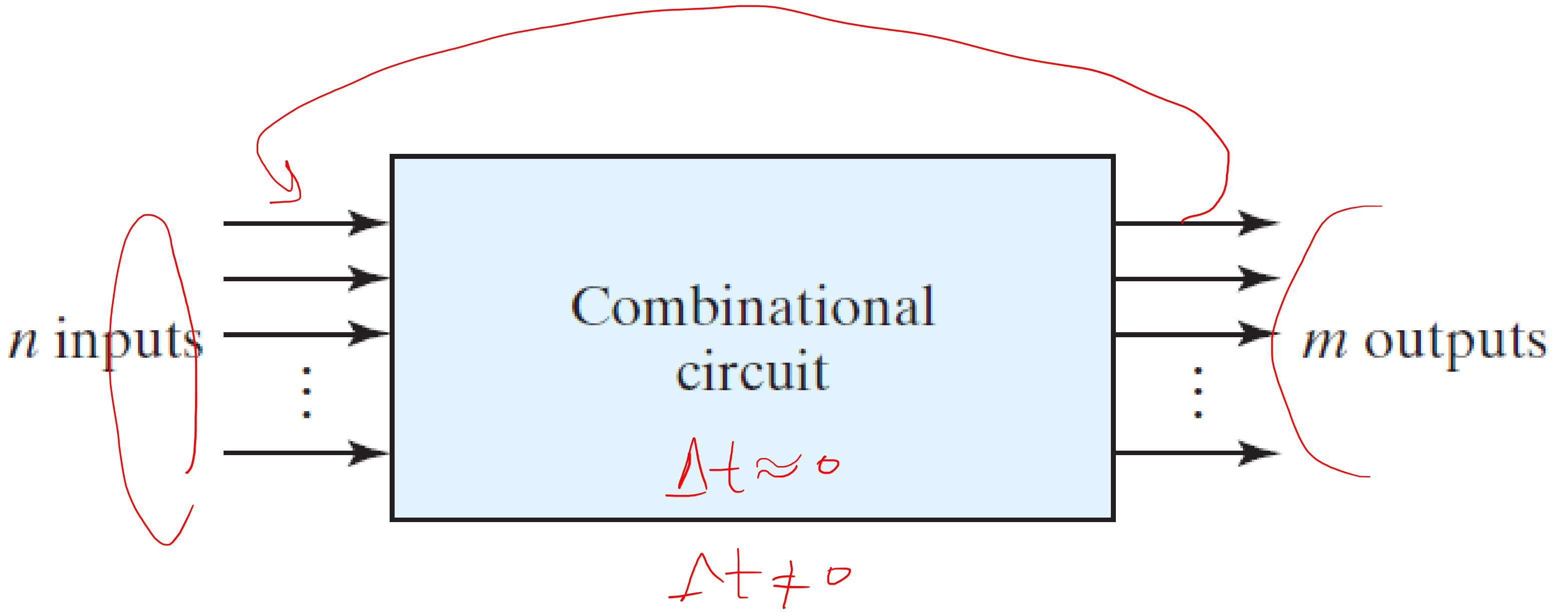


FIGURE 4.1
Block diagram of combinational circuit

kan

K-map = map

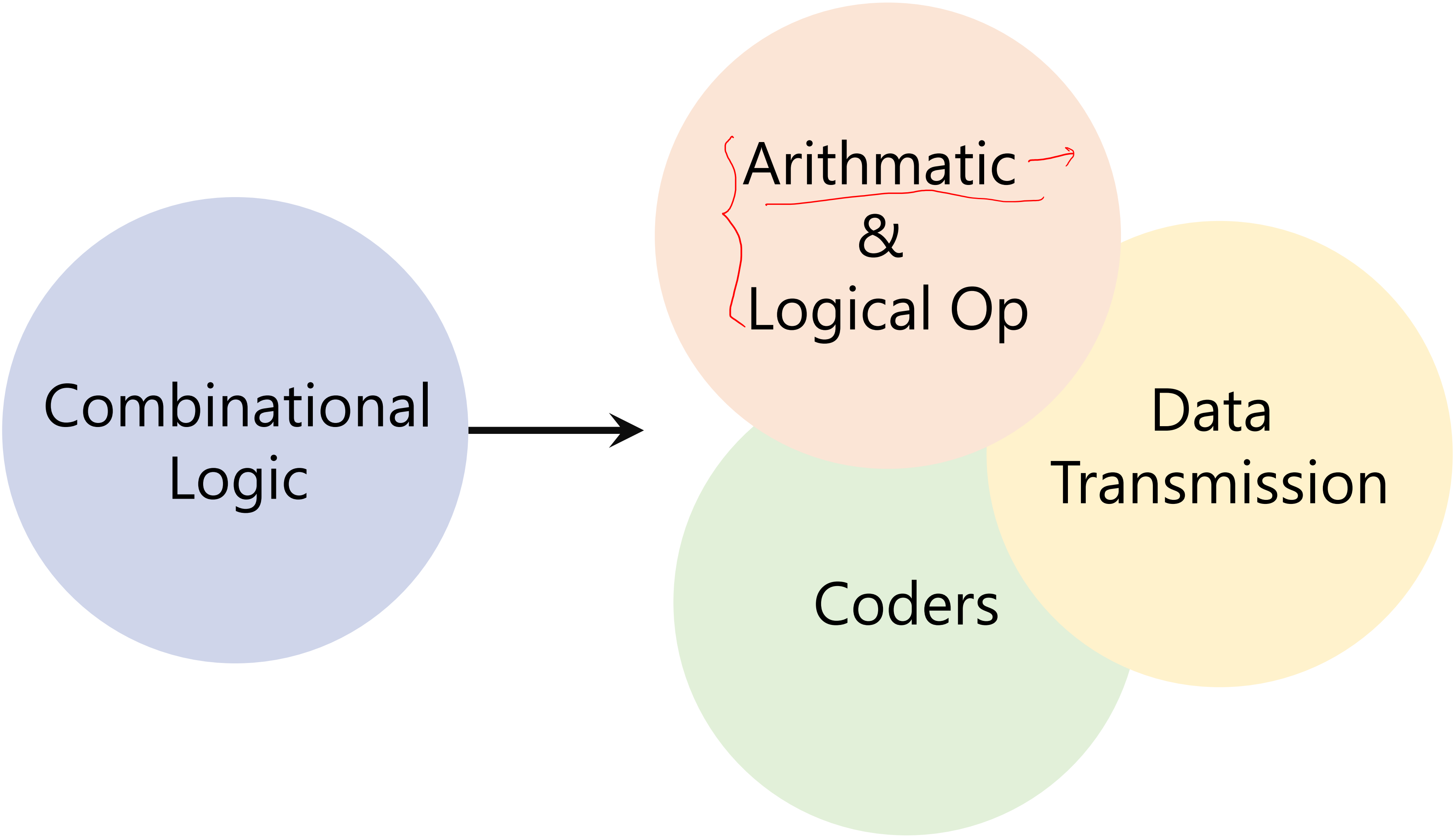
int

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

THE INTERNATIONAL

Calculator Collector

Spring 1993 Issue No. 1

The Cal Tech circa 1967 *Photo Courtesy Texas Instruments*

The Beginning

If you're past your mid-30s, you probably remember your first simple hand-held calculator costing over $50 (in early 1970's dollars). Depending how much older you are, your first could have been upwards to $400. And we're just talking the basic four functions here — addition, subtraction, multiplication, and division. Percentage and memory features were extra (if they were even available at that point in time).

Company Profile:

Bowmar

Who can forget the "Bowmar Brain" series of calculators from the early '70s?

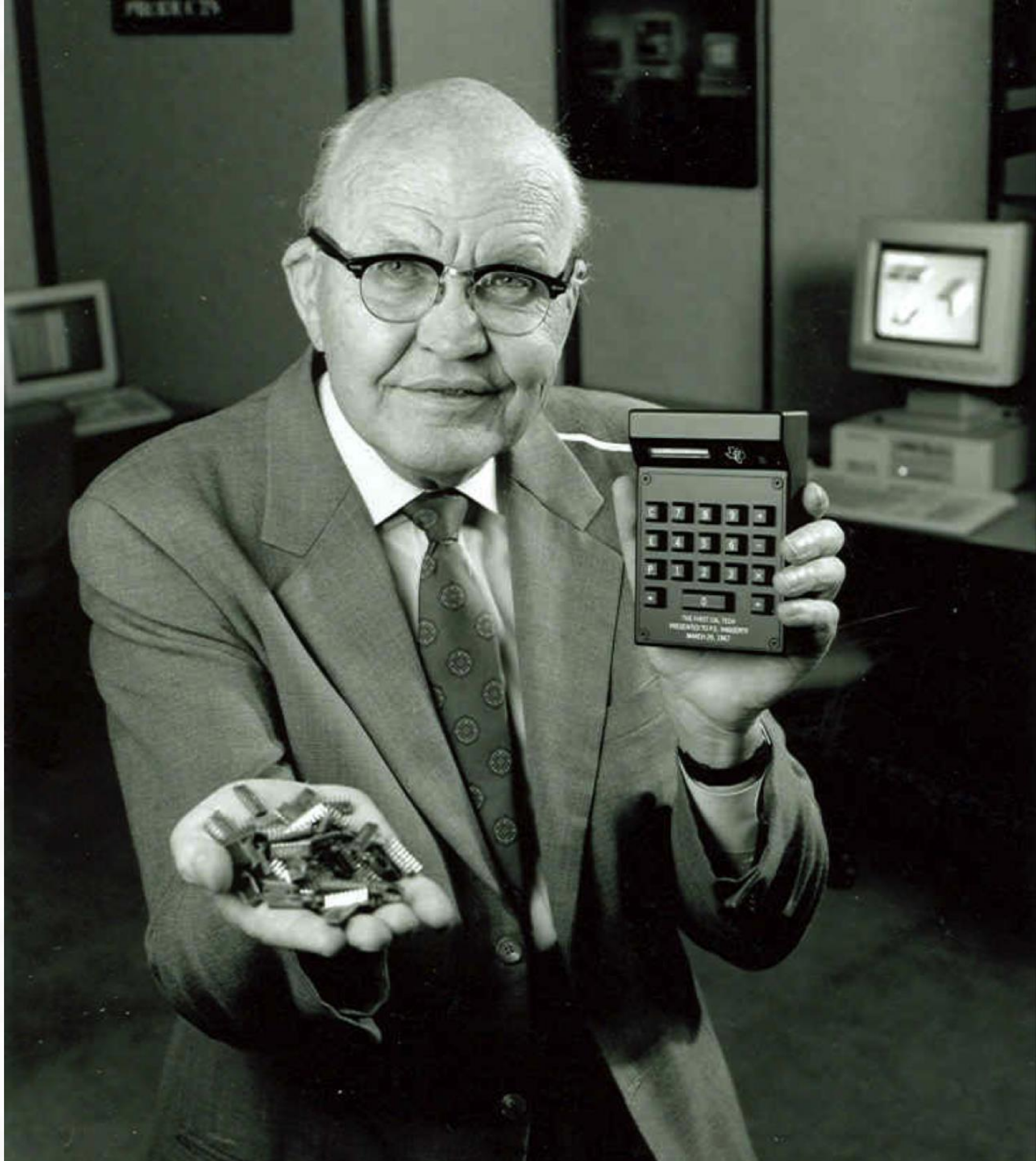
Bowmar was the first American company that made and sold their own line of portable electronic machines.

The story starts around 1970 when Bowmar, then a manufacturer of Light Emitting Diodes (LEDs), tried to sell their numeric display product to Japanese manufacturers for use in their electronic products.

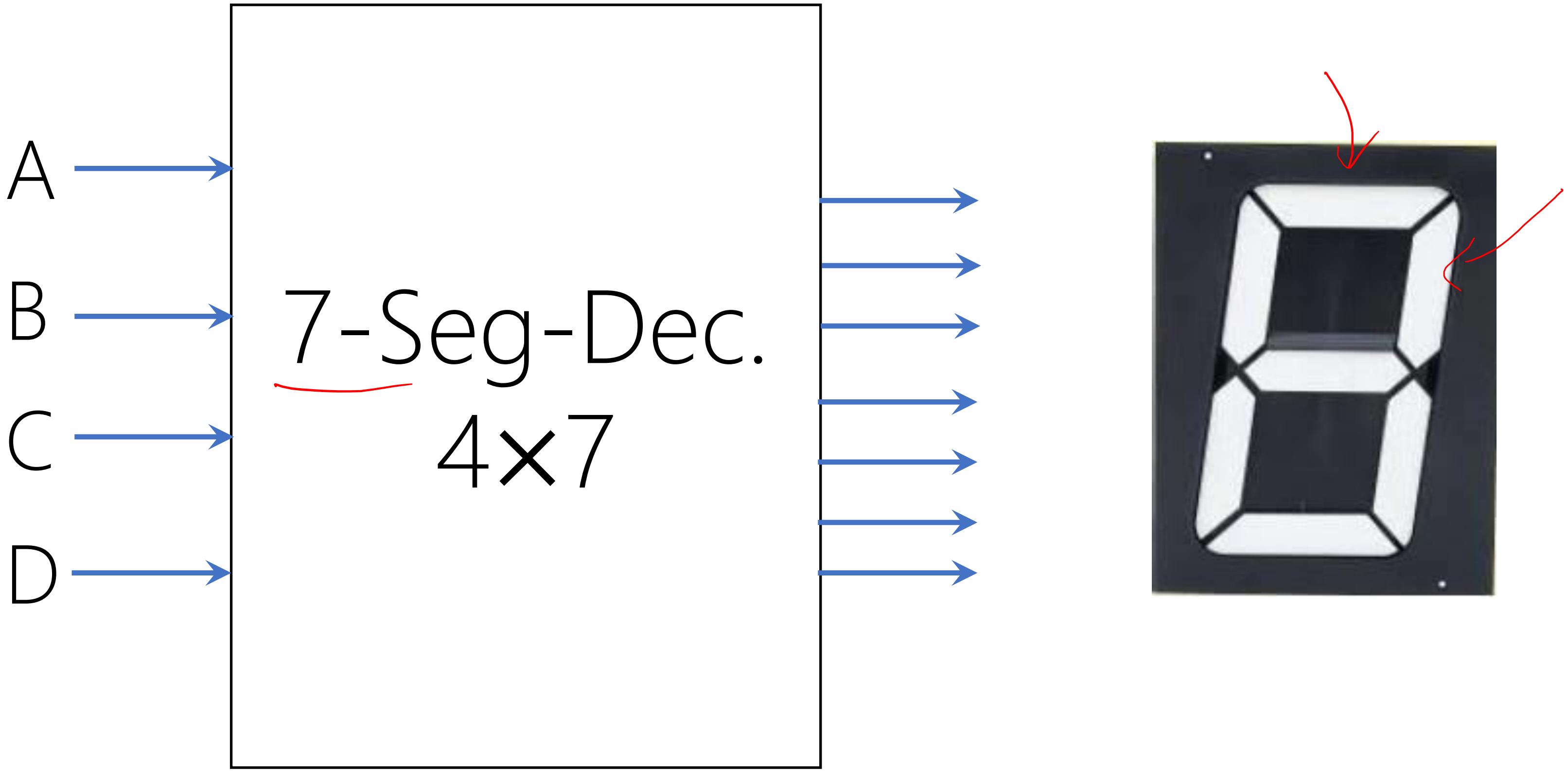
Bowmar wasn't too successful. The Japanese were using a flourescent style display that was cheaper and had a few design features the manufacturers liked better.

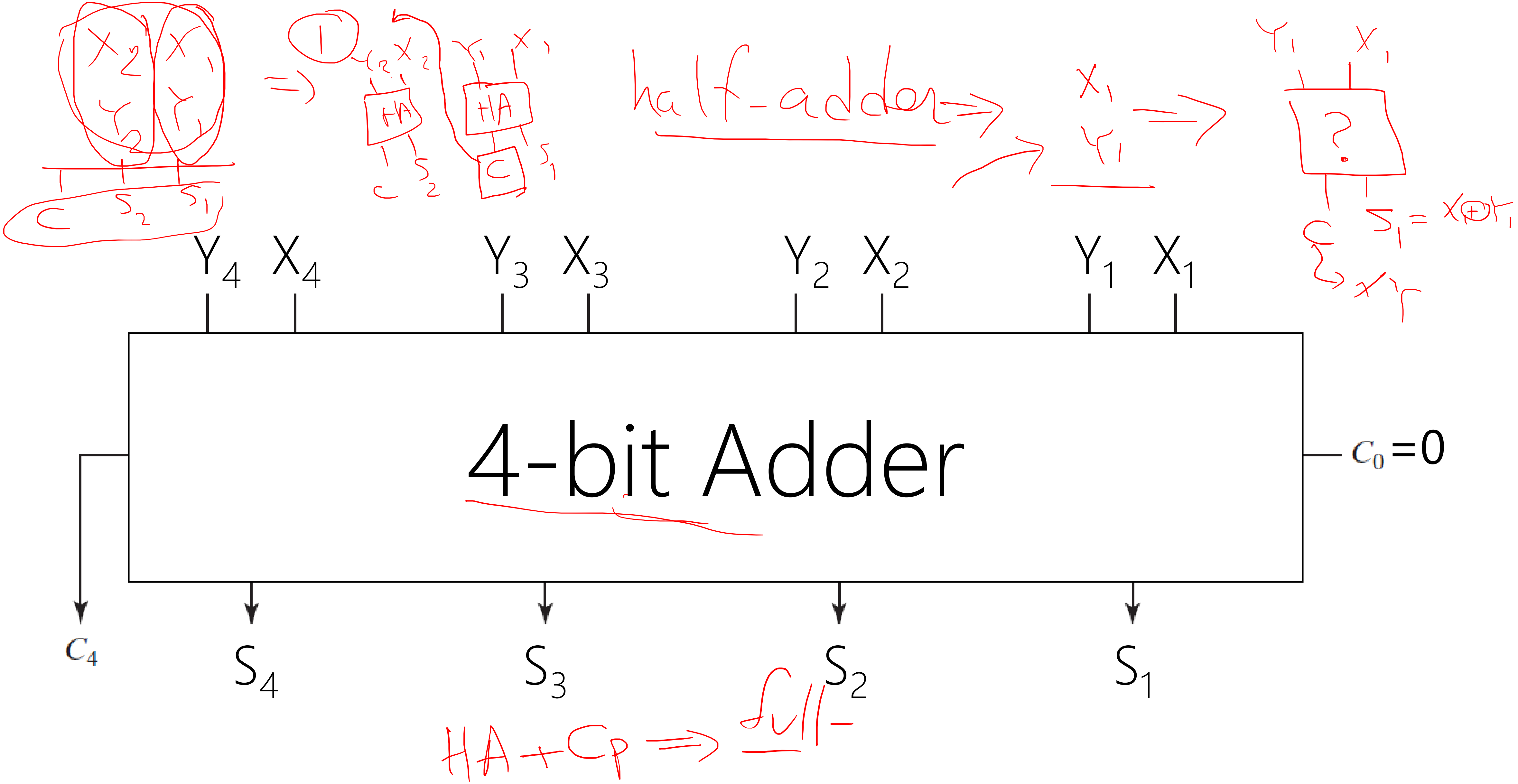
So, president Ed White, a consummate entrepreneur, and his staff came up with an even better idea — make the whole electronic calculator themselves.

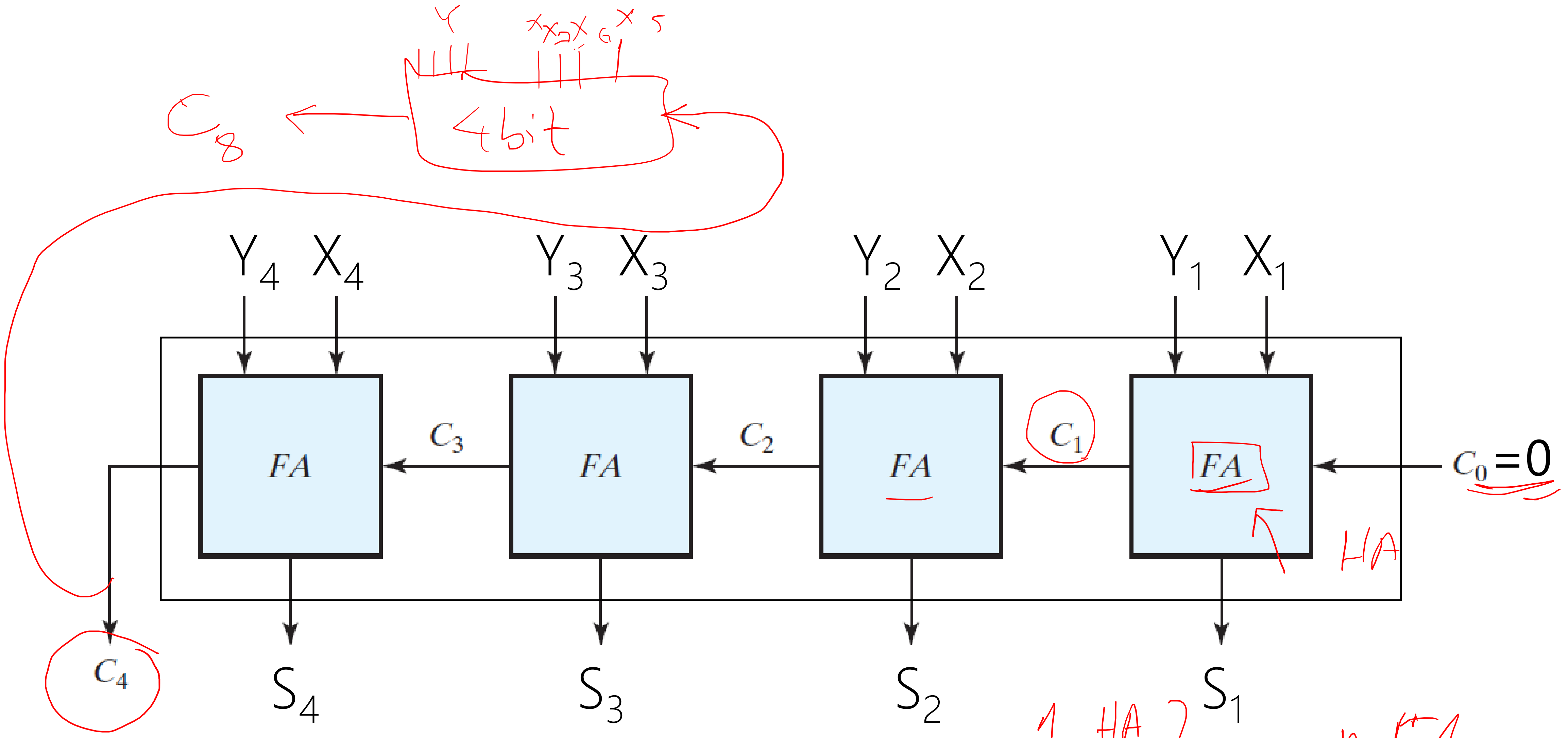
Up to now, most of the so-called "portable" calculators



Binary Number

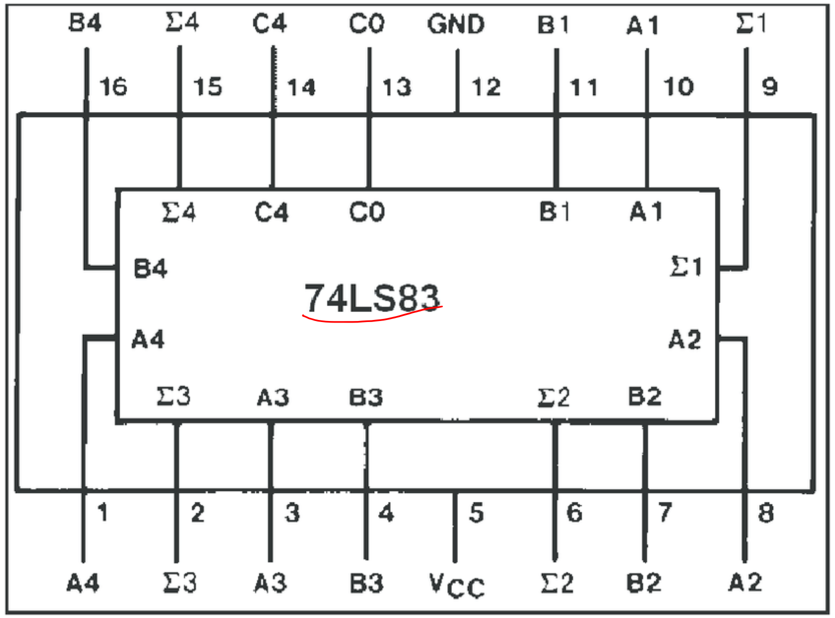



4-bit Adder
Y4 X4 Y3 X3 Y2 X2 Y1 X1
C0=0
C4
S4 S3 S2 S1
half-adder
HA
C
S1 = X⊕Y
HA + Cp => full-

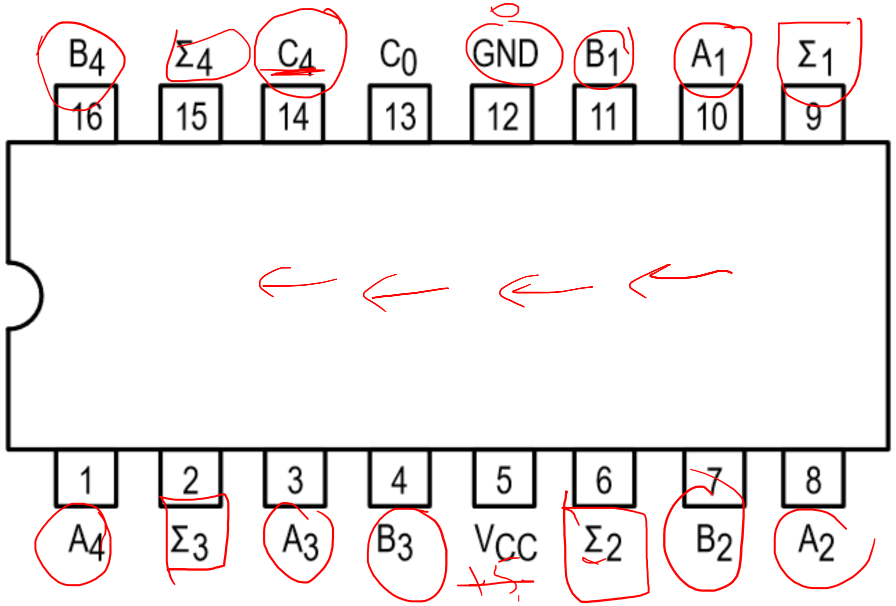


$$\text{n-bit} \Rightarrow \left.\begin{array}{l} 1 \ \text{HA} \\ (n-1) \ \text{FA} \end{array}\right\} \Rightarrow \text{n-FA}$$

$$
\begin{array}{rcl}
 & \boxed{C_3C_2C_1C_0} & 0 \\
 & X_4X_3X_2X_1 & \\
+ & Y_4Y_3Y_2Y_1 & \\
\hline
\boxed{C4} & S_4S_3S_2S_1 &
\end{array}
$$



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)

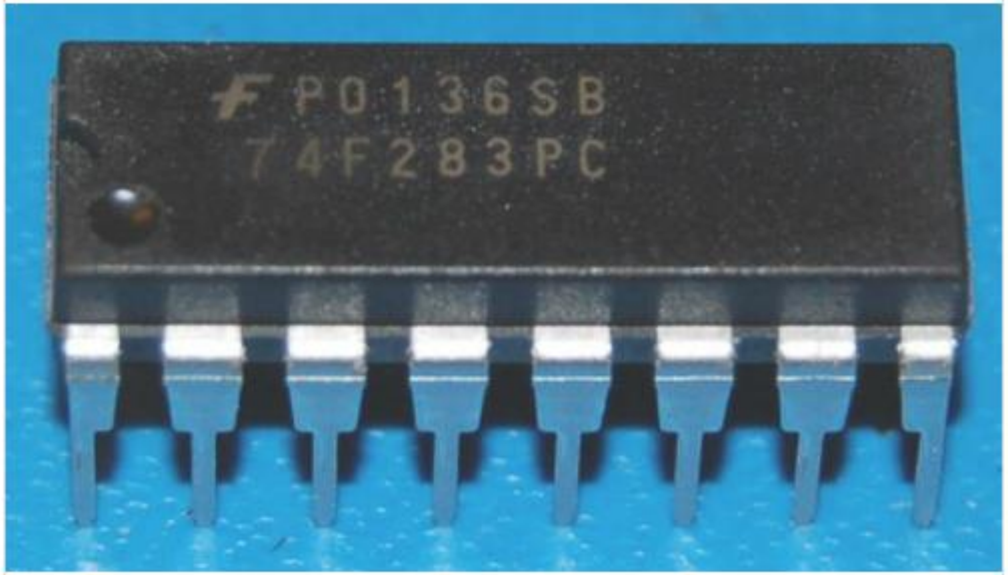
Hi! Sign in or register eBay Deals Brand Outlet Help & Contact Holidays At Home English Sell Watch List My eBay

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

ebay MONEY BACK GUARANTEE

Qty: 1

Buy It Now

Add to cart

Watch

Sold by

vedge23 (3476)

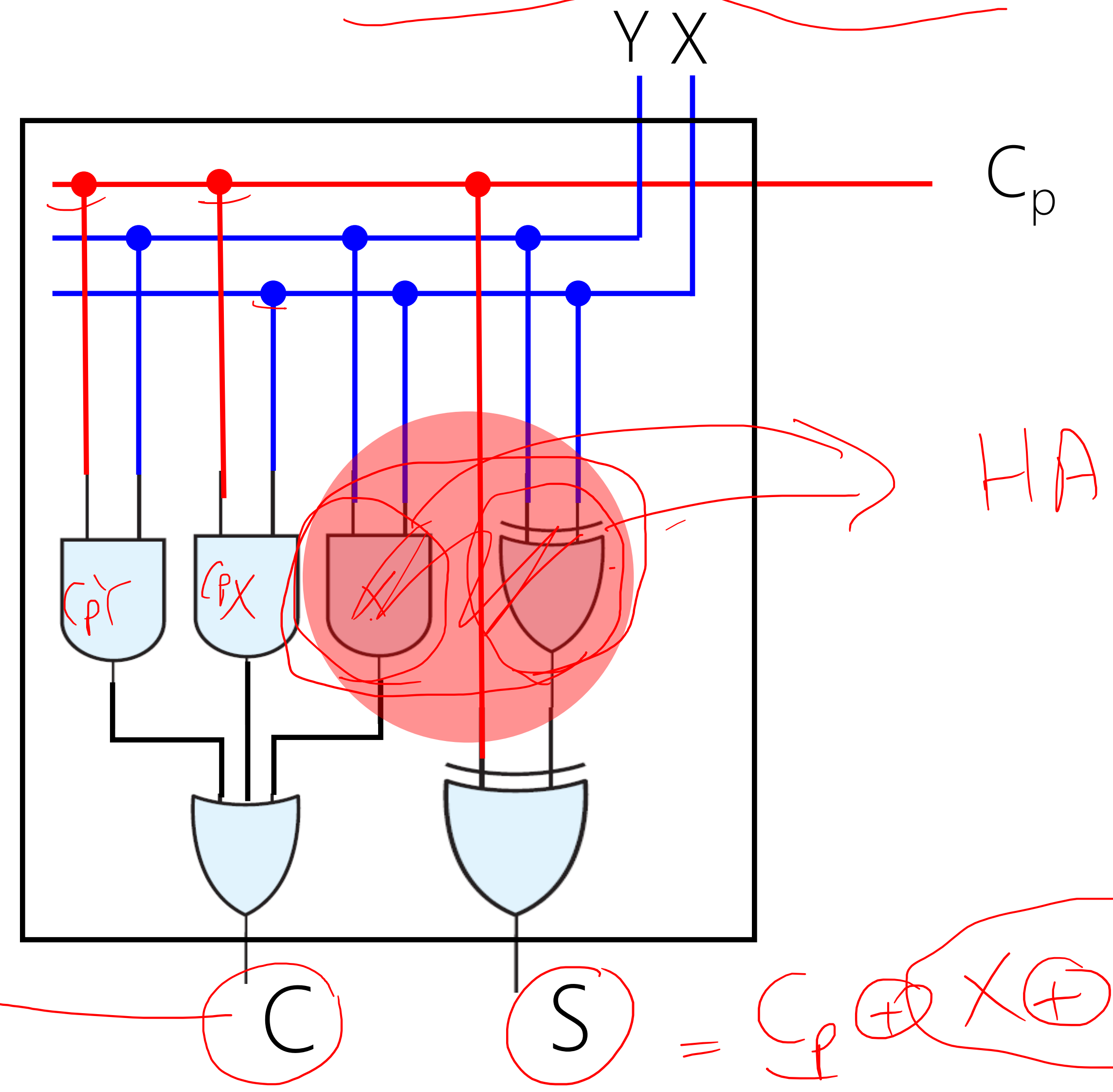
100.0% Positive feedback

Contact seller

More on Full-Adder

Full Adder $\stackrel{?}{=}$ Half Adder + ...

$\Rightarrow$ FA

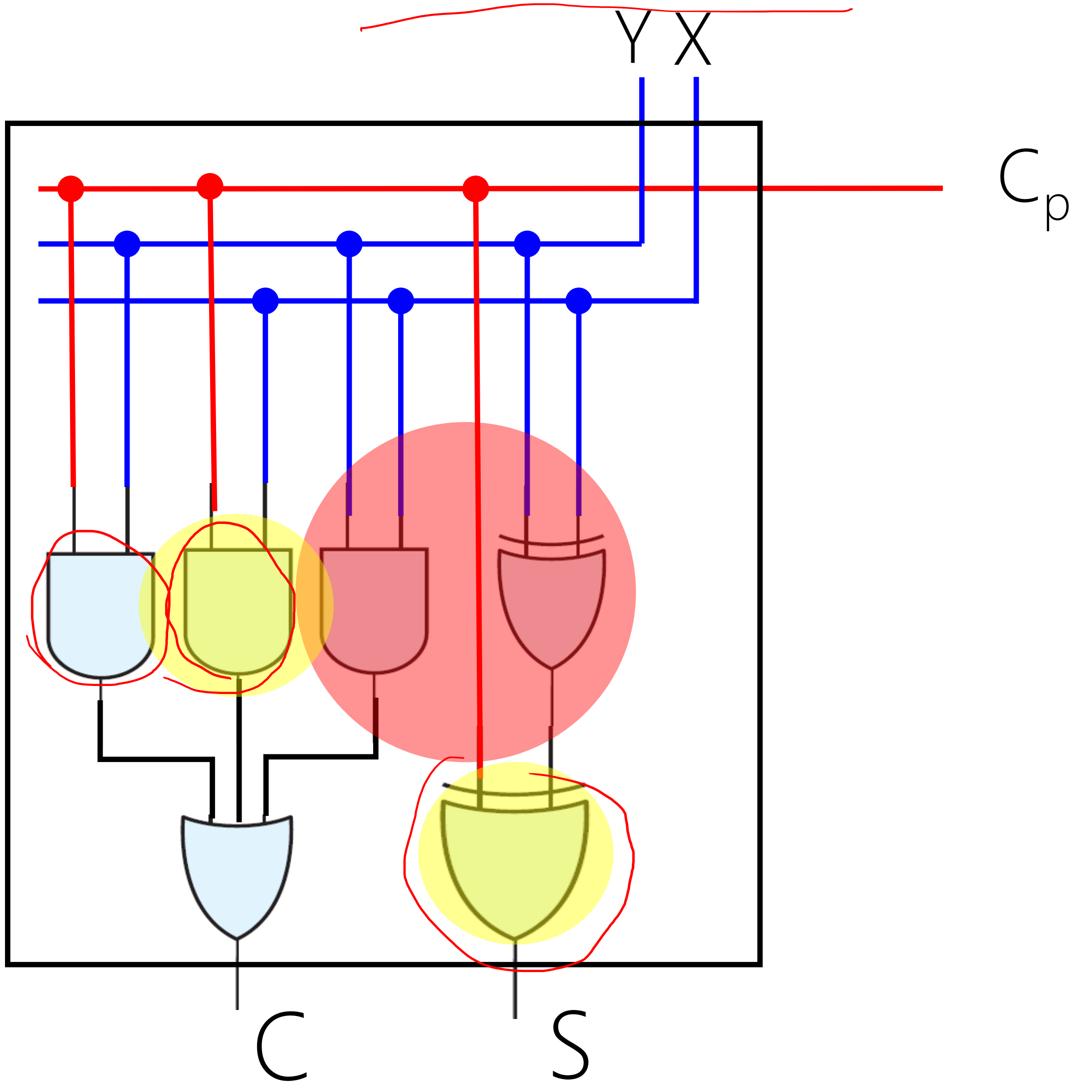


$XY + C_pX + C_pY$ $\leftarrow$ C

S $= C_p \oplus X \oplus Y$

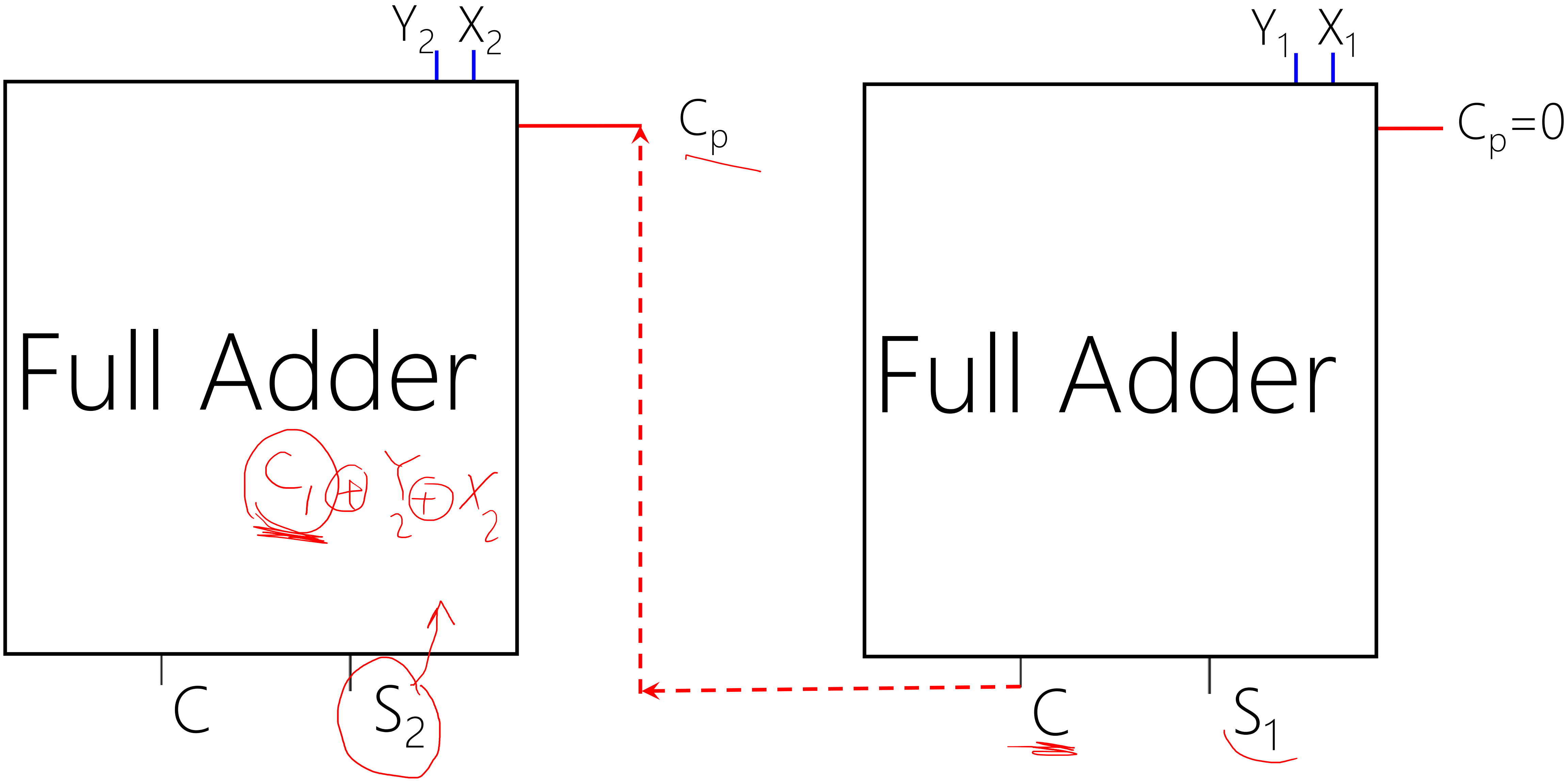
Lecture Assignment

Full Adder $=^?$ Half Adder + ...



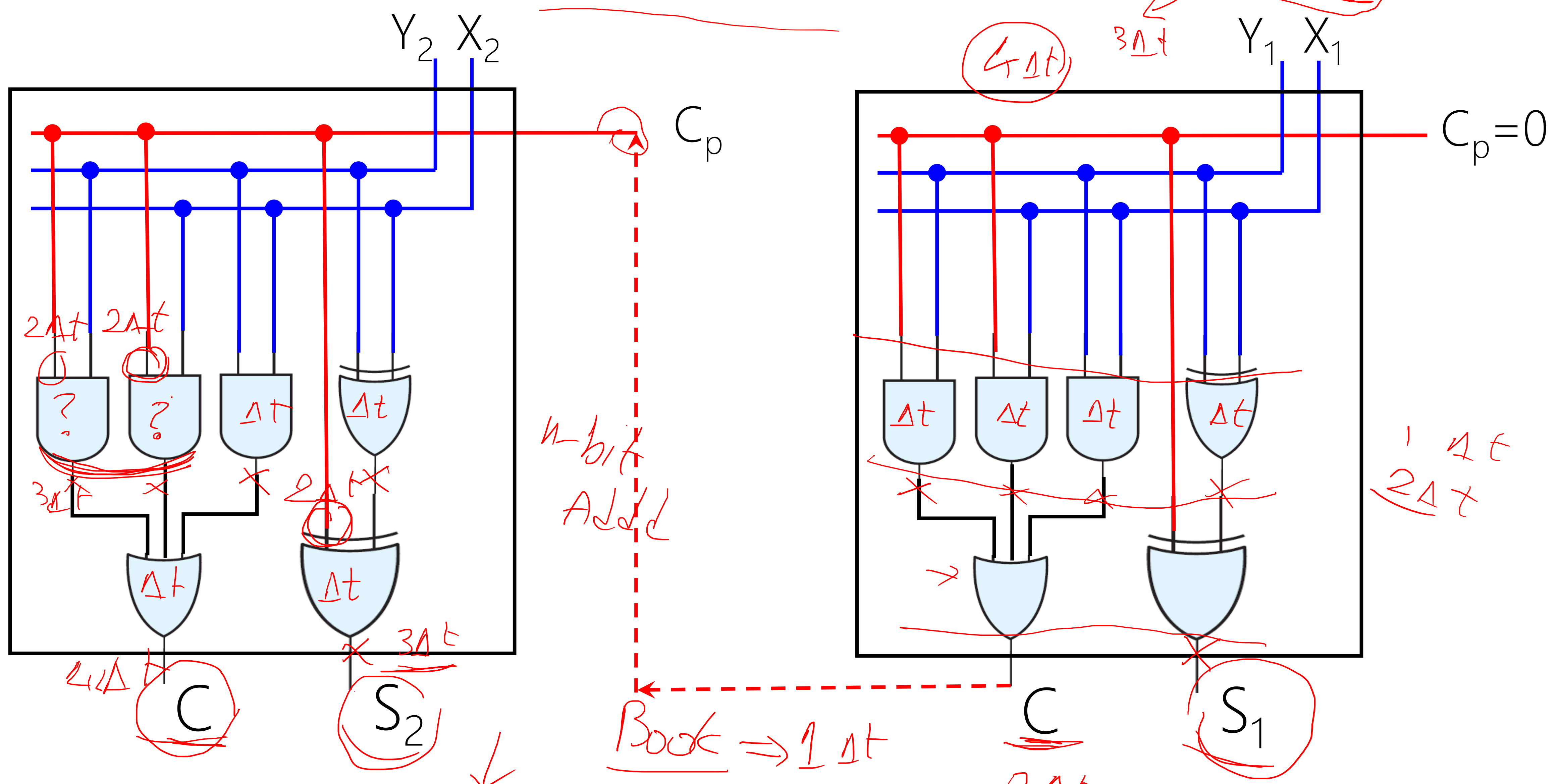
Lecture Assignment

Carry Propagation



Y_2 X_2
C_p
Y_1 X_1
$C_p=0$
Full Adder
Full Adder
$C_1 \oplus Y_2 \oplus X_2$
C
S_2
C
S_1

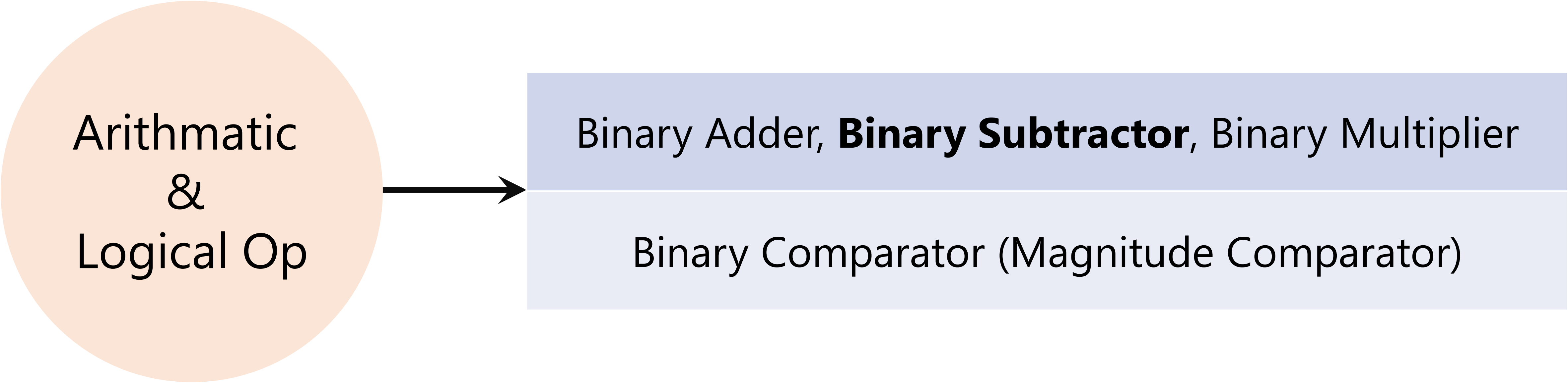
If gate delay is Δt, how long does it take to see the $S = S_2S_1$?



Lecture Assignment

Full 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)

0 X_1	1	0	1
− 0 Y_1	− 0	− 1	− 1
B=0 D_1	B=0 D_1	$\boxed{B}$ D_1	B=0 D_1

2-bits
3-bit

Binary Subtractor

Half-Subtractor ⟶ Full-Subtractor ⟶ n-bit Full-Subtractor

Lecture Assignments

Binary Subtractor

Signed-2's-Complement

$X - Y$

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

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

Subtraction in Signed-2's-Complement

n-bit x_i y_i $c_0 = 0$ 1

bitwise

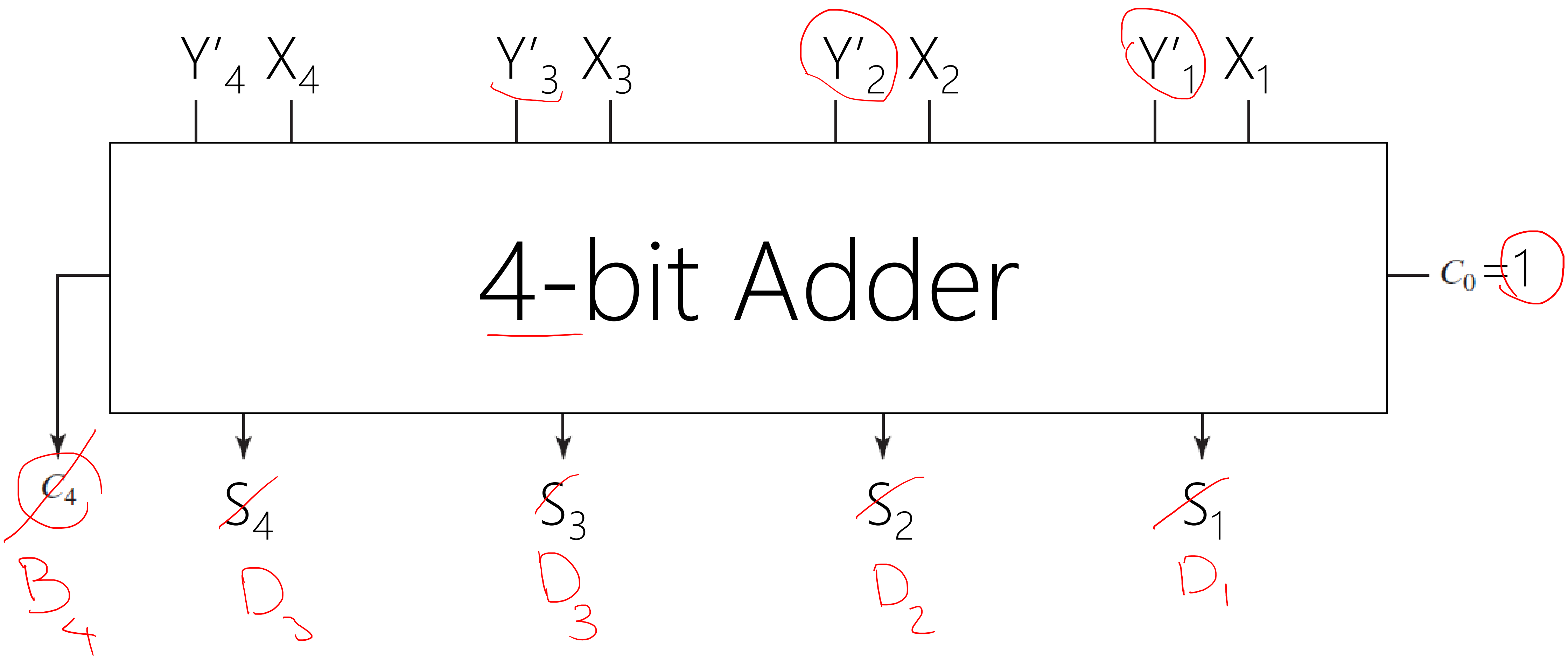
$$X + Y' + 1$$

Subtraction in Signed-2's-Complement

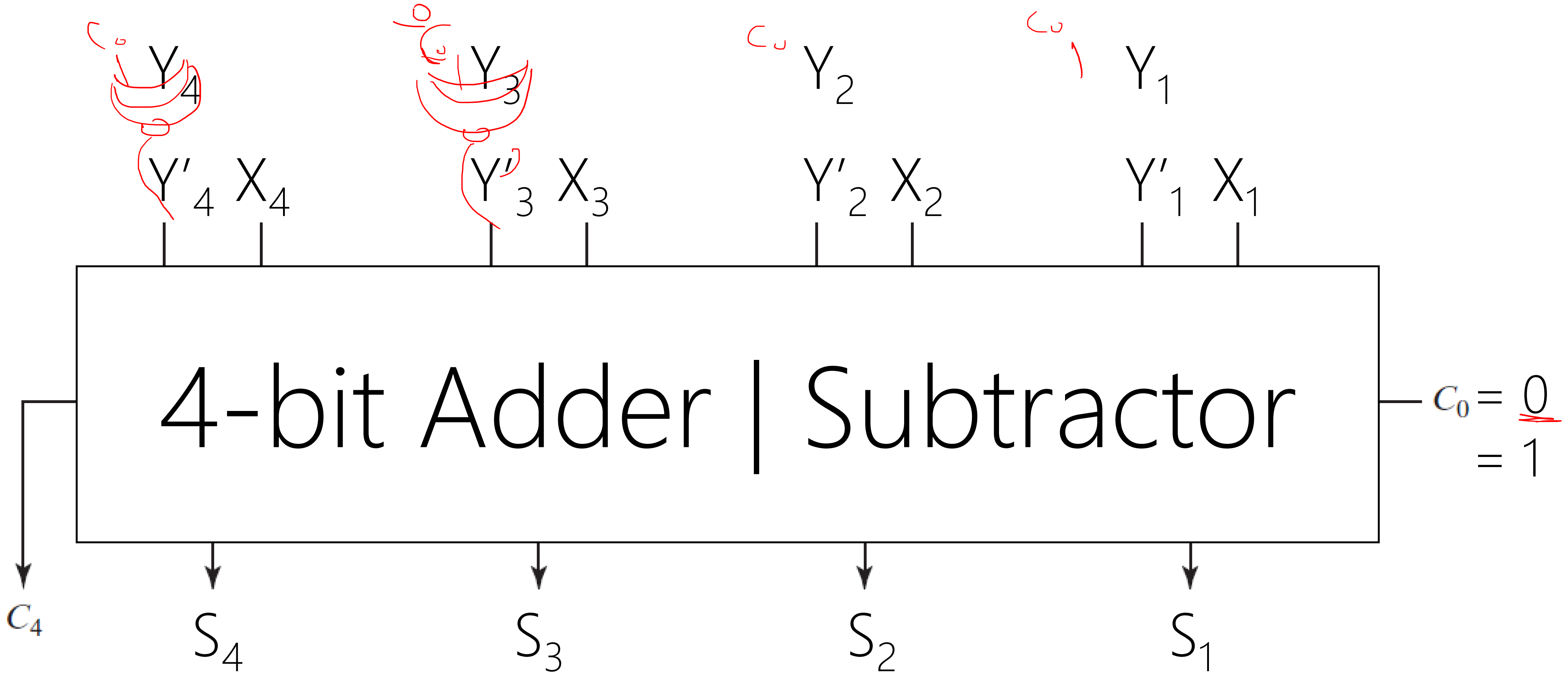
bitwise

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

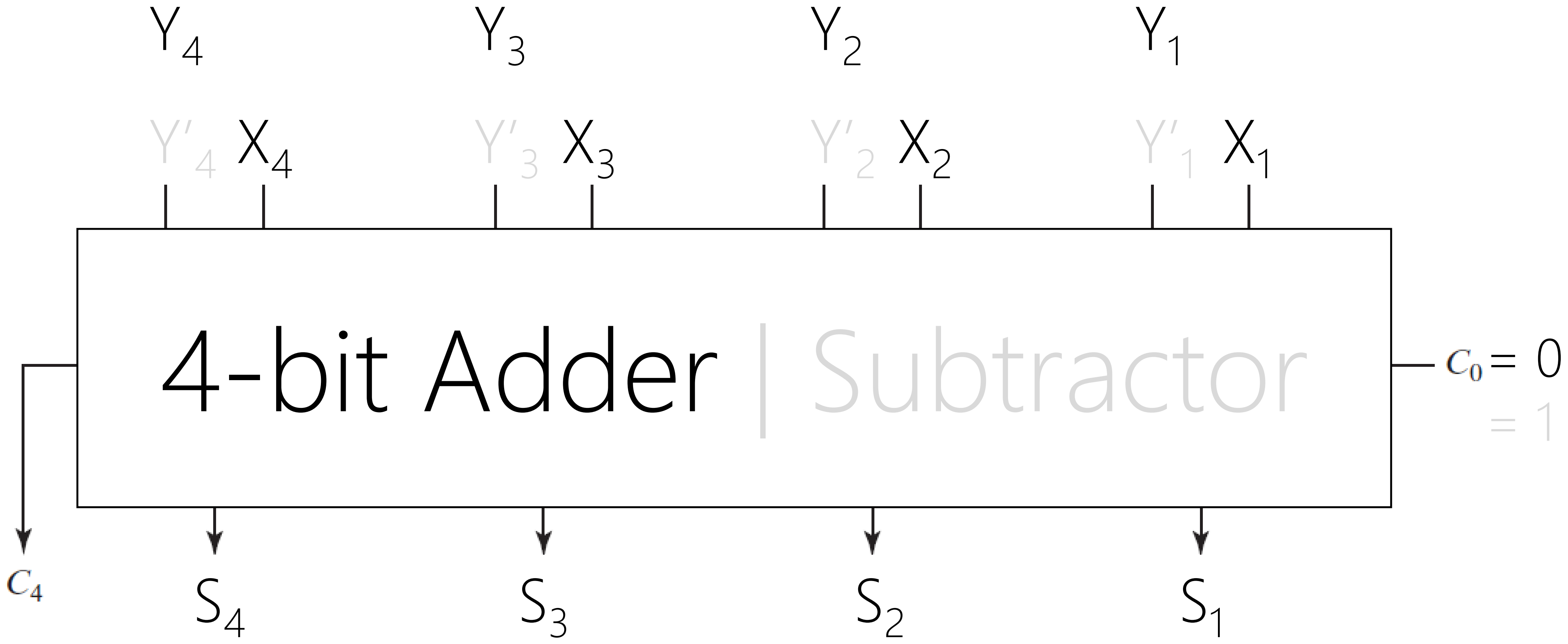
Subtraction in Signed-2's-Complement



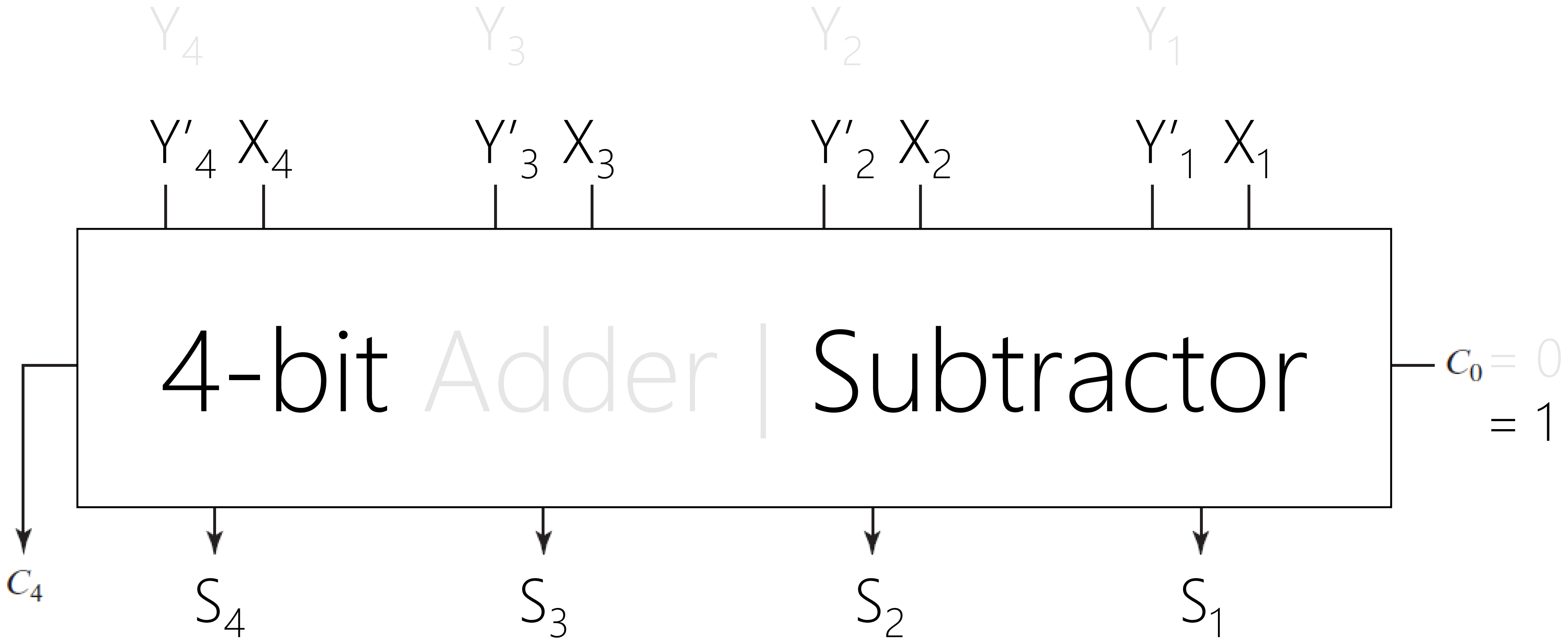
Y'4 X4
Y'3 X3
Y'2 X2
Y'1 X1
4-bit Adder
C0 = 1
C4
S4
S3
S2
S1
B4
D3
D3
D2
D1



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

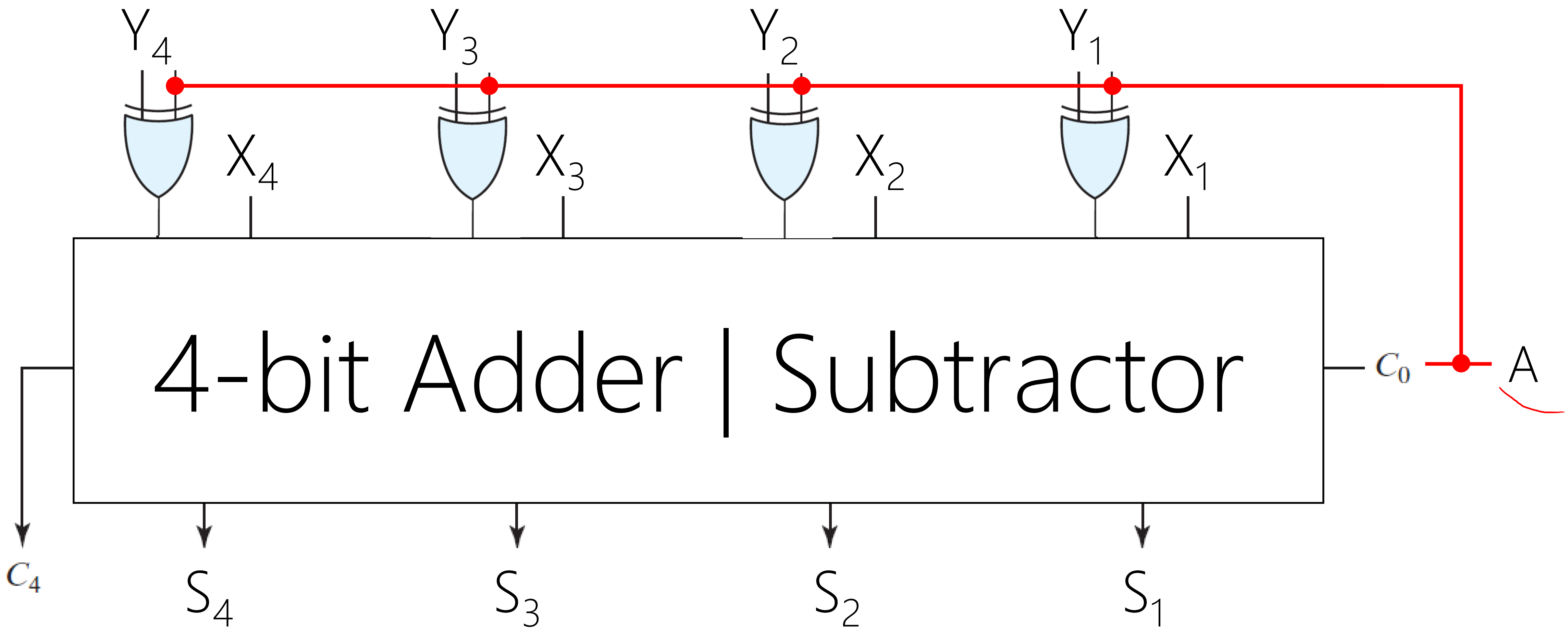


Y4
Y3
Y2
Y1
Y'4 X4
Y'3 X3
Y'2 X2
Y'1 X1
4-bit Adder | Subtractor
c0 = 0
= 1
c4
S4
S3
S2
S1

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

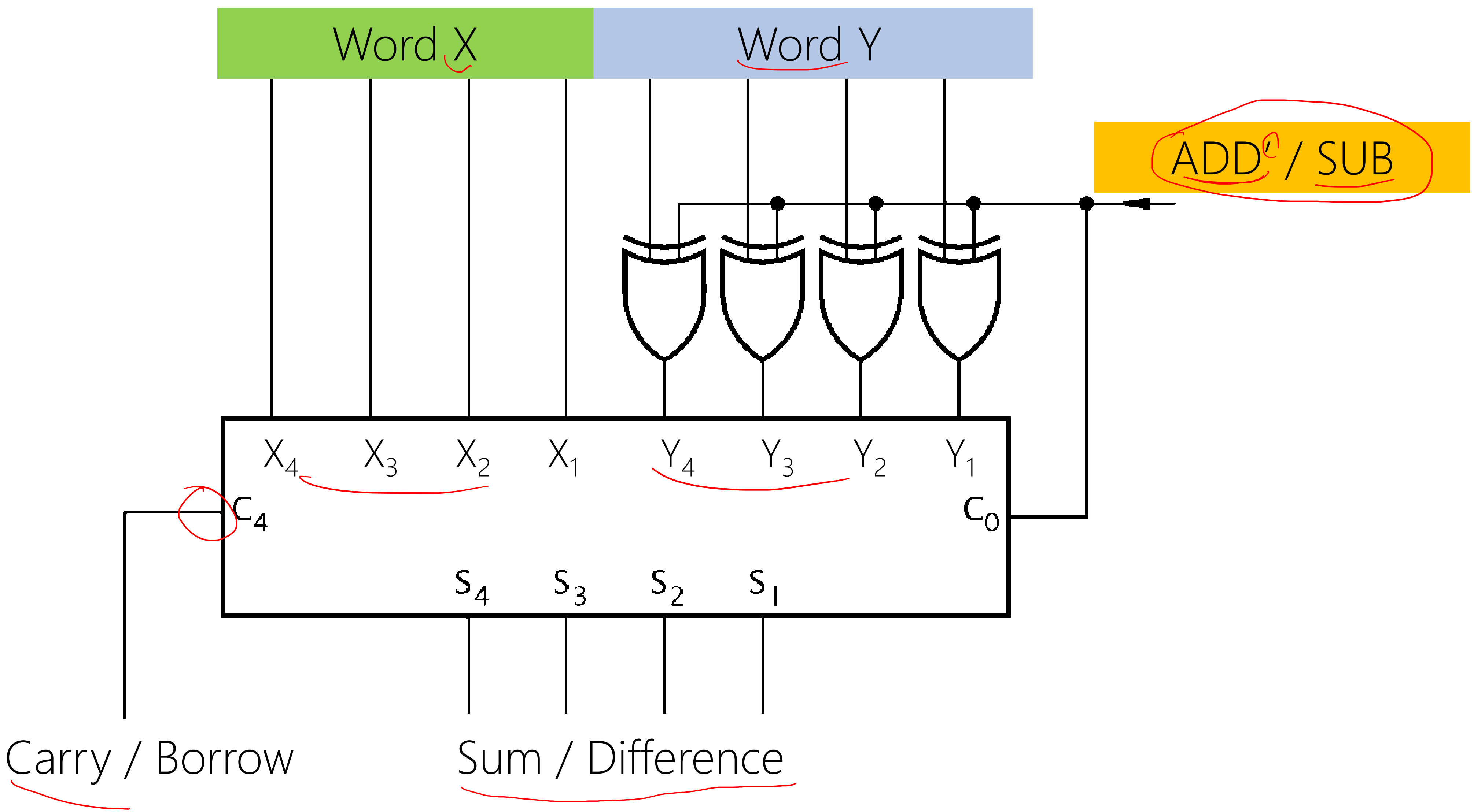
$$A \oplus 0 = A$$

$$A \oplus 1 = A'$$



A=0 → Adder

A=1 → Subtractor



Word X
Word Y
ADD' / SUB
X4
X3
X2
X1
Y4
Y3
Y2
Y1
C4
C0
S4
S3
S2
S1
Carry / Borrow
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									?

$2^9 = f(a, b, c \ldots) =$

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

Y2	Y1	X2	X1	C	OVF
2-Bit Adder					?

overflow

Signed-2's-Complement

Using Prior Knowledge

overflow

Signed-2's-Complement

(I)

Subtraction → Addition with 2's Comp.

overflow

Signed-2's-Complement

(II)

Sum of Positive Numbers → Negative: OVF=1

Sum of Negative Numbers → Positive: OVF=1

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

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

Nothing to do!

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

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)

$$
\begin{array}{lr}
 & \boxed{C_3C_2C_1C_0} \\
 & 1 \quad 0 \quad X_4X_3X_2X_1 \\
+ & 1 \quad 0 \quad Y_4Y_3Y_2Y_1 \\
\hline
\boxed{C4} & 0 \quad 1 \quad S_4S_3S_2S_1
\end{array}
$$

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

$OVF = C_3 C_4'$

1

$C_3C_2C_1C_0$

$X_4 = 0\ X_3X_2X_1$

$+ \quad Y_4 = 0\ Y_3Y_2Y_1$

$C4 \quad S_4 = 1\ S_3S_2S_1$

0

OVF

$\boxed{1}$

OVF!

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

OVF!

C'_4C_3

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

OVF = 0

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

Don't believe it, try!

$$\boxed{C_3C_2C_1C_0}$$

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

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

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

1 0

1

1

OVF!

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

OVF!

$C_4C'_3$

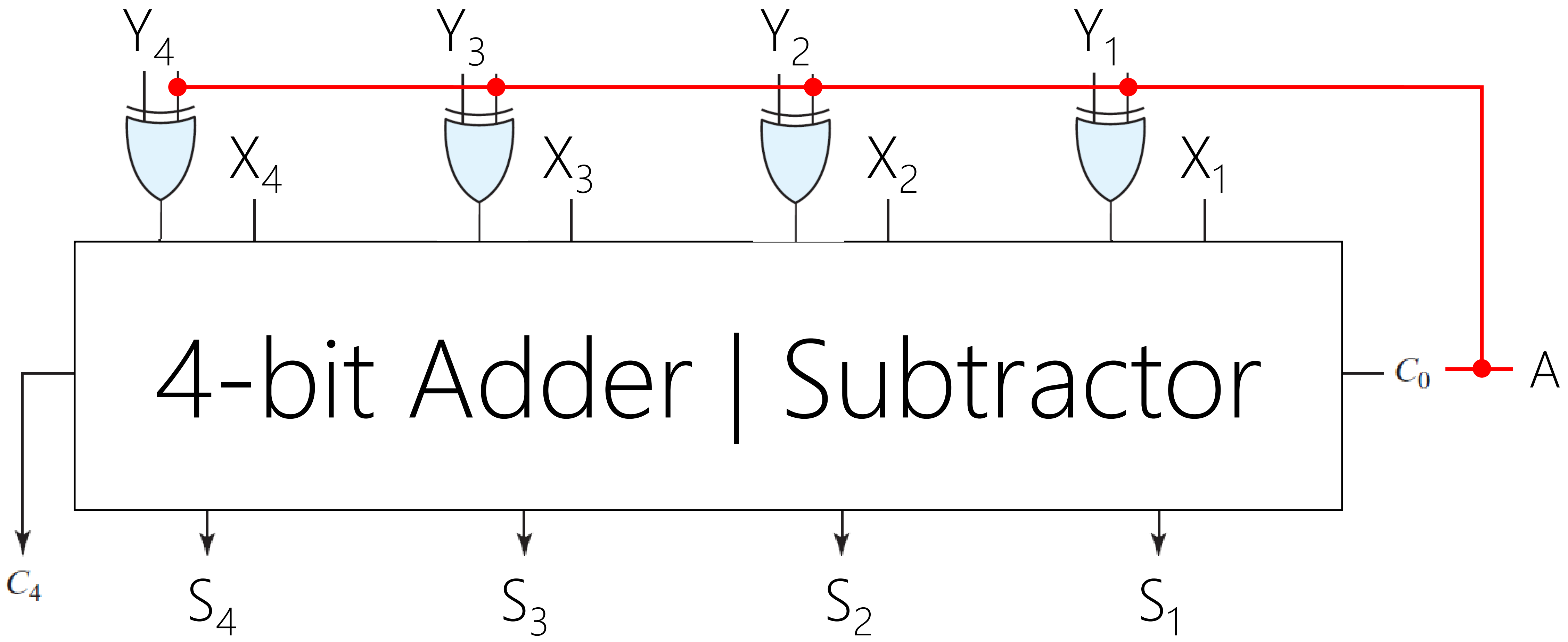
n_bit adder

$C_n \oplus C_{n-1}$

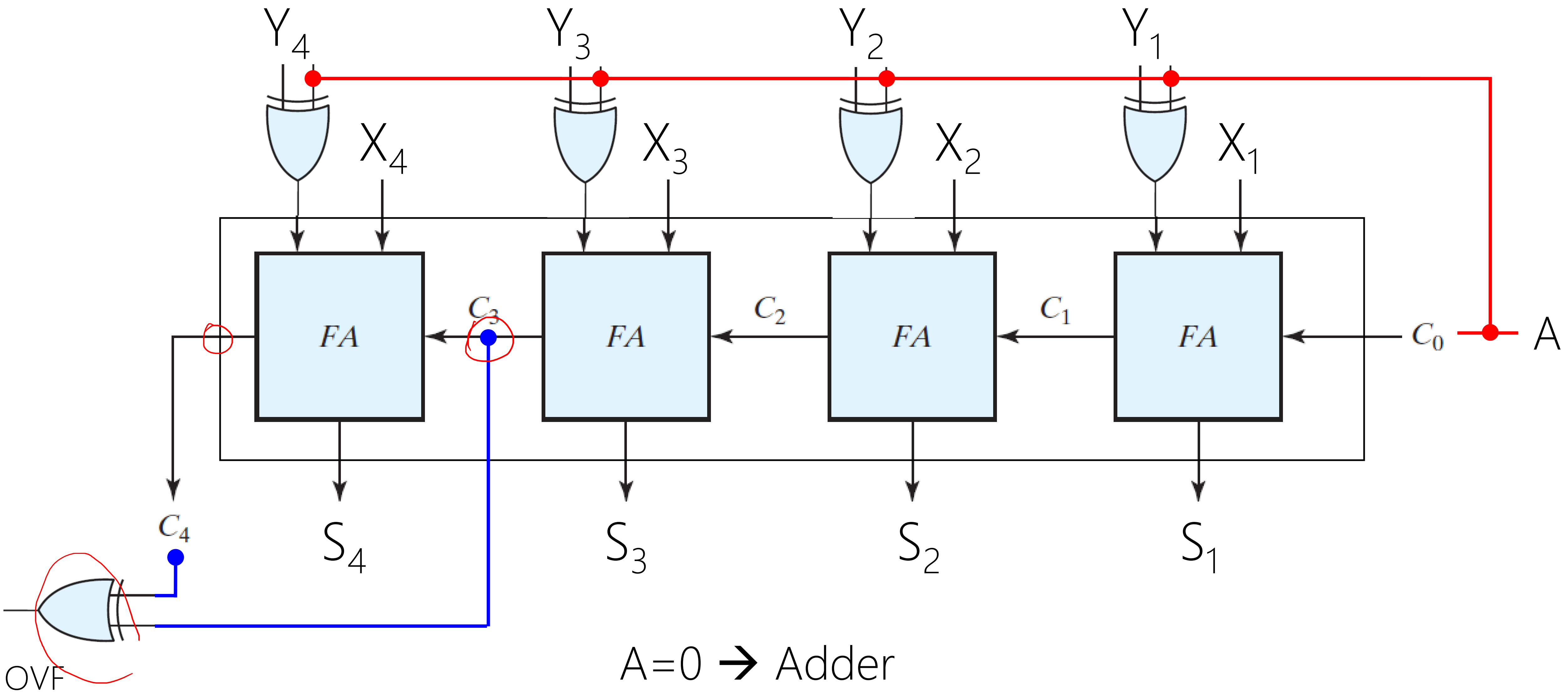
overflow

Signed-2's-Complement

$$OVF = \underbrace{C'_4C_3}_{X'Y} + \underbrace{C_4C'_3}_{XY'} = C_4 \oplus C_3$$



A=0 → Adder
A=1 → Subtractor

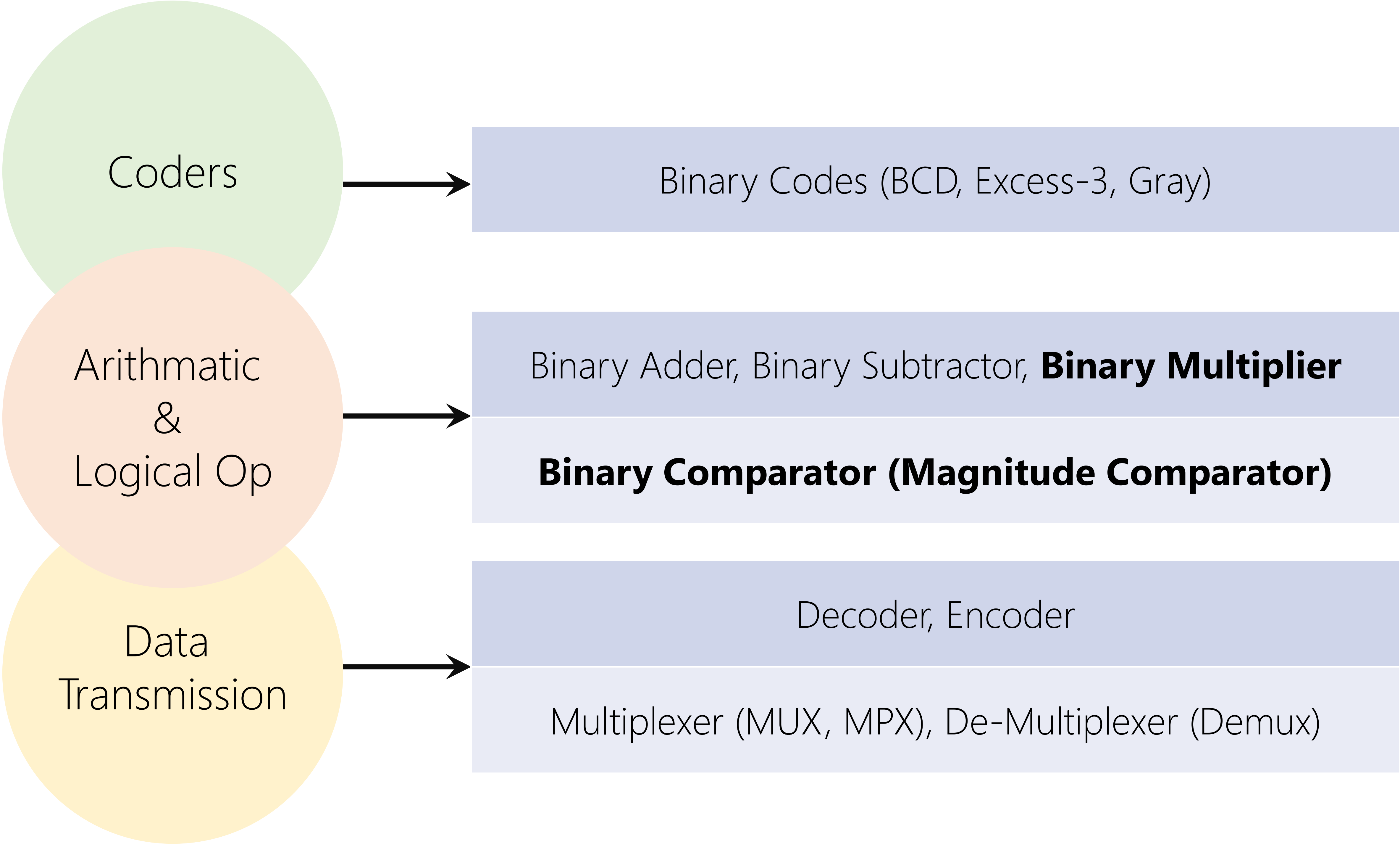


A=0 → Adder

A=1 → Subtractor

Binary Adder | Subtractor | Overflow

Unsigned?


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)

Arithmatic
&
Logical Op

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

Binary Comparator (Magnitude Comparator)

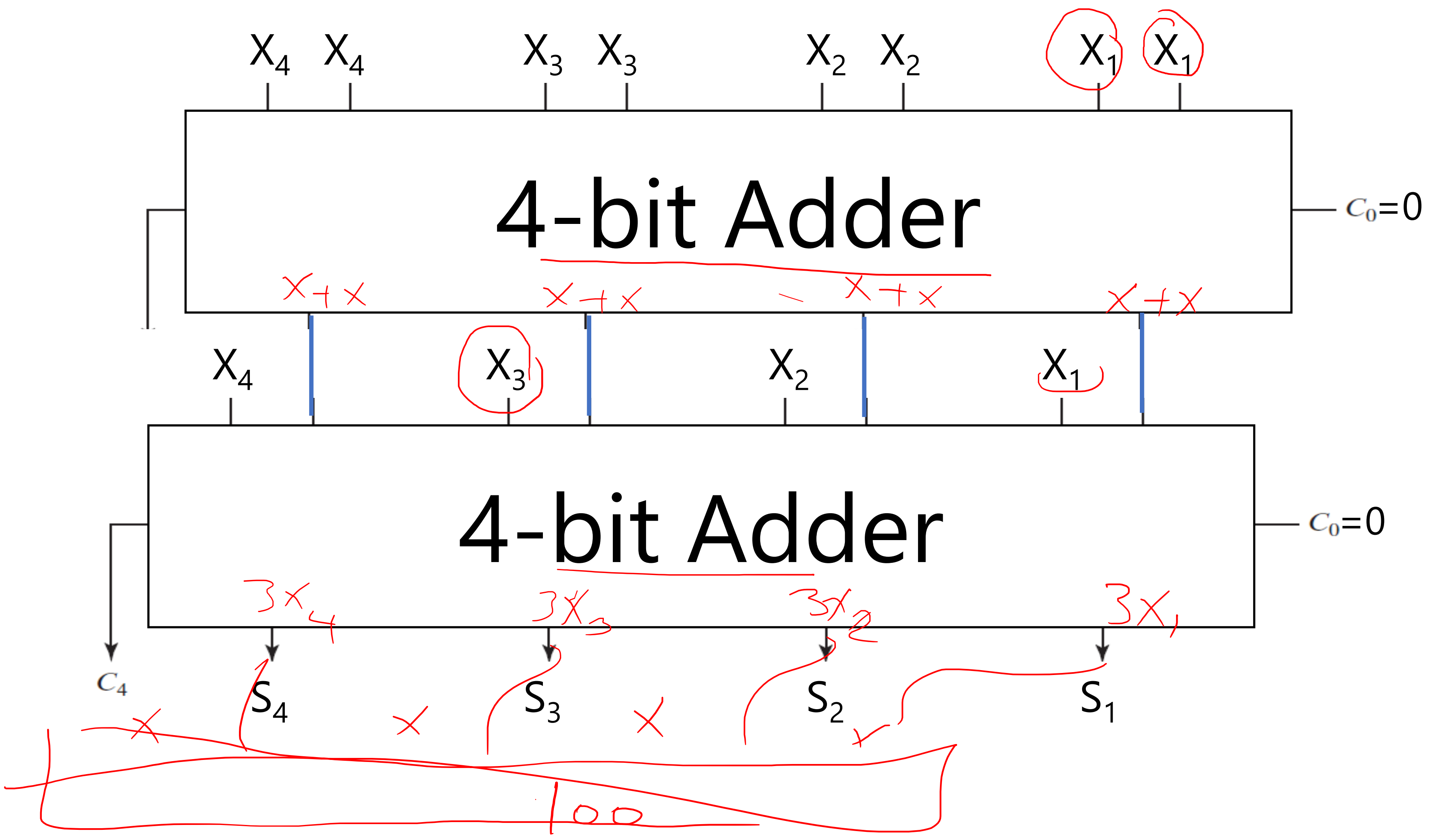
100
× 11 ⟹

100 +
100 +
100 +

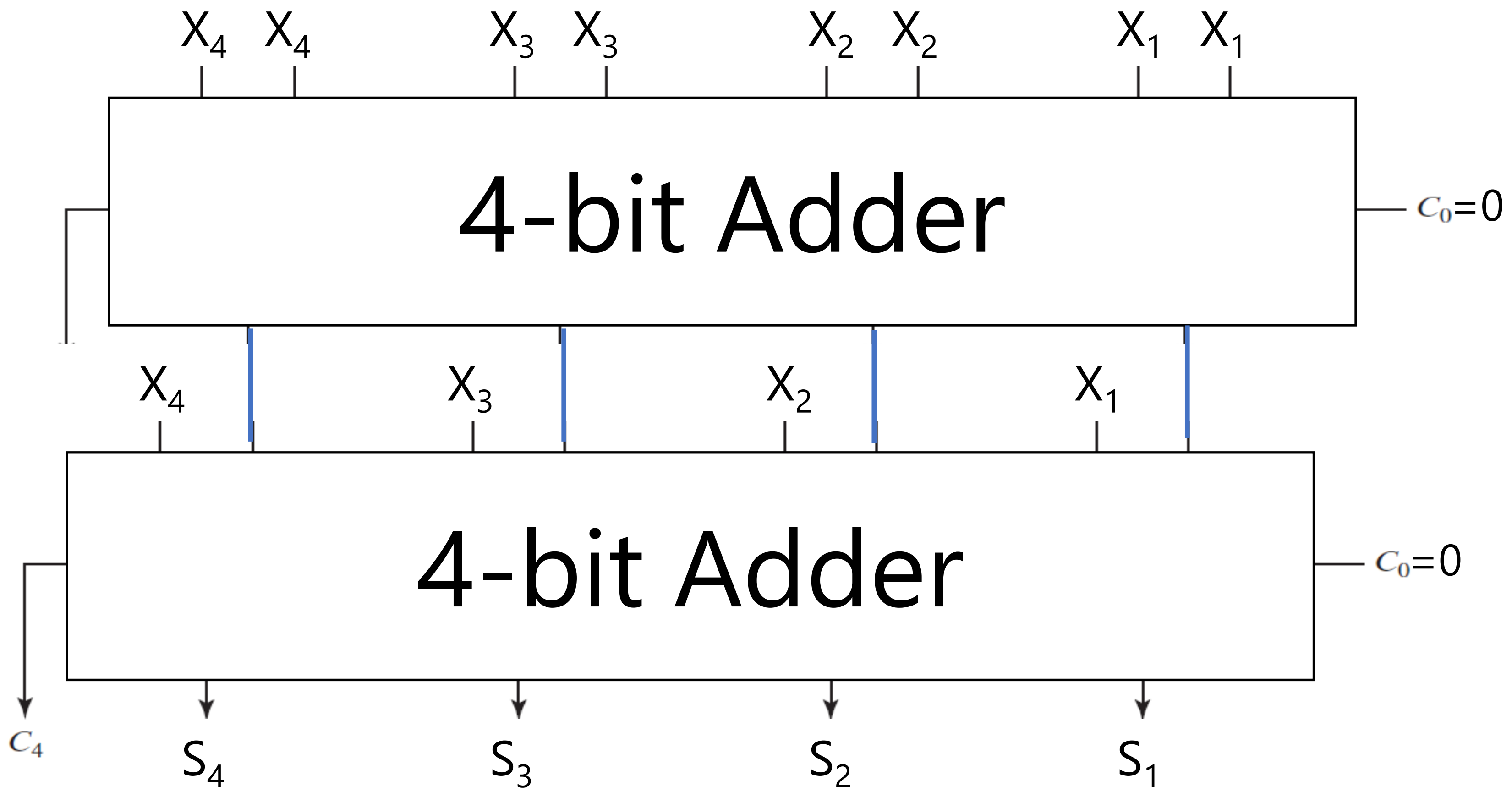
Binary Multiplier

Unsigned

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

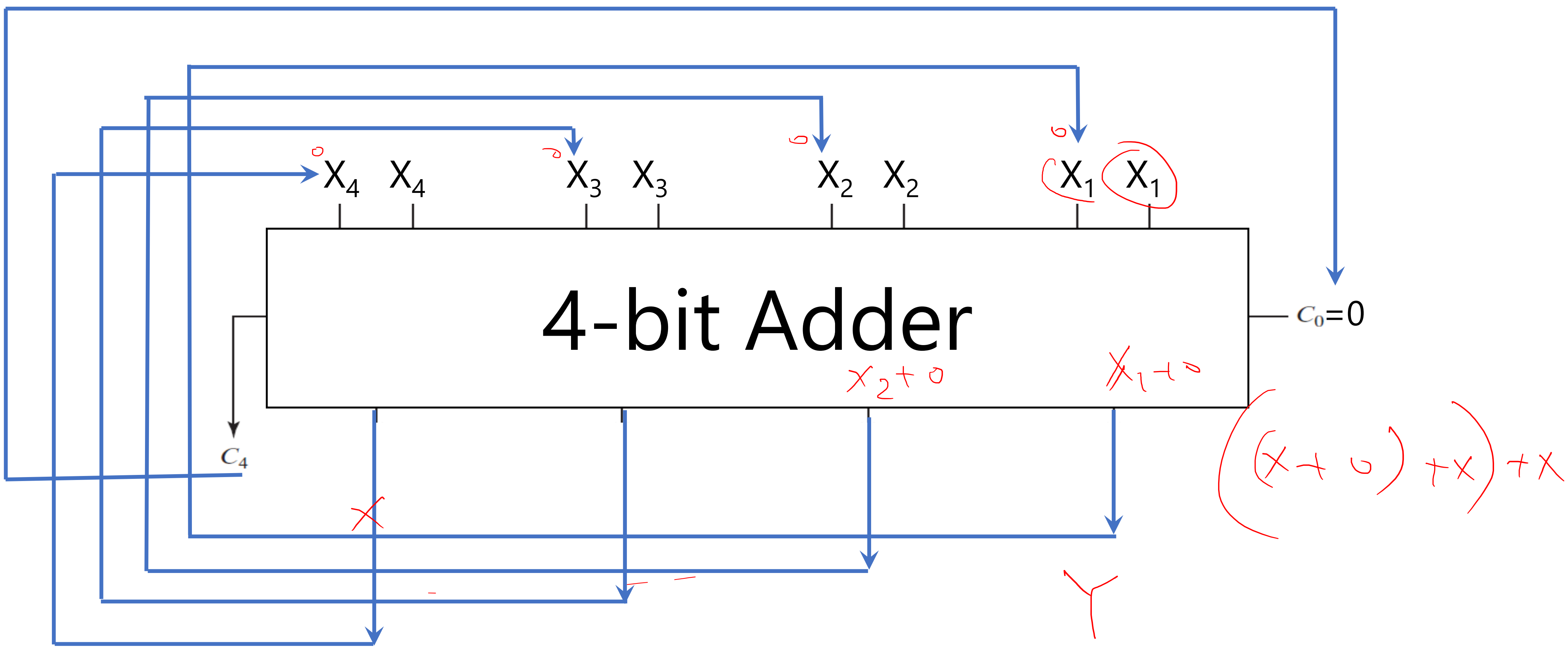


$$X \times (11)_2 = X+X+X$$



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

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



$X \times Y = X + \ldots + X \rightarrow$ When to stop?

Feedback $\rightarrow$ Sequential Logic

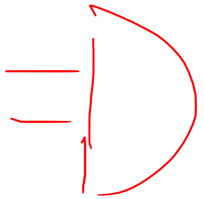
Binary Multiplier

Unsigned

X AND $0 = 0$

X AND $1 = X$

$$
\begin{array}{rr}
 & X_4X_3X_2X_1 \\
\times & 1\;Y_1 \\
\hline
 & M_4M_3M_2M_1
\end{array}
$$

$\cancel{0}$ 1 $\Rightarrow$ 

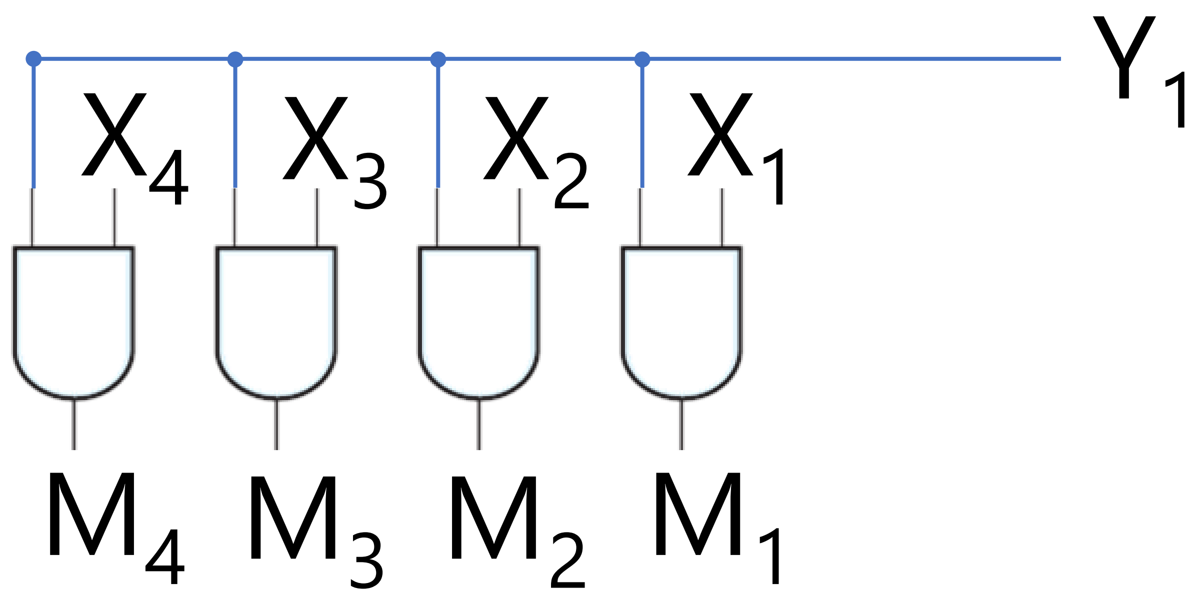
X_4 X_3 X_2 X_1

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

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

$$
\begin{array}{r}
X_4 X_3 X_2 X_1 \\
\times \qquad\qquad Y_1 \\
\hline
M_3 = Y_1 X_3
\end{array}
$$

$$
\begin{array}{r}
X_4X_3X_2X_1 \\
\times \qquad\qquad\qquad\qquad Y_1 \\
\hline
M_4=Y_1X_4
\end{array}
$$



Y1
X4
X3
X2
X1
M4
M3
M2
M1

$$\begin{array}{lrrrrr}
 & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_2 & Y_1 \\
\hline
 & & M_4 & M_3 & M_2 & M_1 \\
+ & Y_2X_4 & Y_2X_3 & Y_2X_2 & Y_2X_1 & 0 \\
\hline
 & & & & M_2 & M_1
\end{array}$$

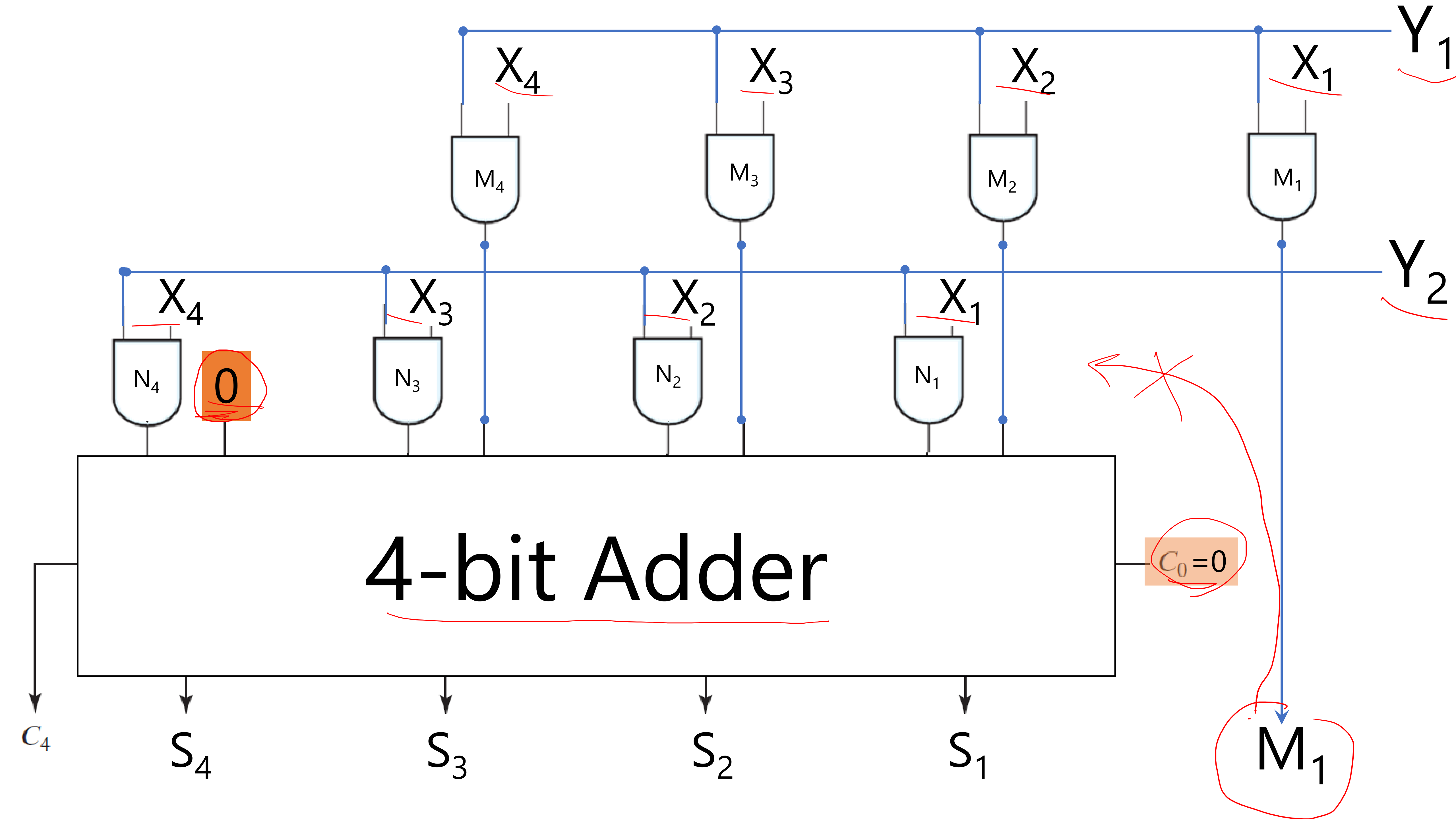
$$
\begin{array}{rccccc}
 & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_2 & Y_1 \\
\hline
 & & M_4 & M_3 & M_2 & M_1 \\
 & N_4 & N_3 & N_2 & N_1 & 0
\end{array}
$$

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

$$
\begin{array}{rcccccc}
 & & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & & Y_2 & Y_1 \\
\hline
 & & 0 & M_4 & M_3 & M_2 & M_1 \\
+ & & N_4 & N_3 & N_2 & N_1 & 0 \\
\hline
 & \boxed{C4} & S_4 & S_3 & S_2 & S_1 & M_1
\end{array}
$$

4 bit adder

$$
\begin{array}{cccccc}
 & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_2 & Y_1 \\
\hline
 & 0 & M_4 & M_3 & M_2 & M_1 \\
+ & N_4 & N_3 & N_2 & N_1 & 0 \\
\hline
C4 & S_4 & S_3 & S_2 & S_1 & M_1
\end{array}
$$



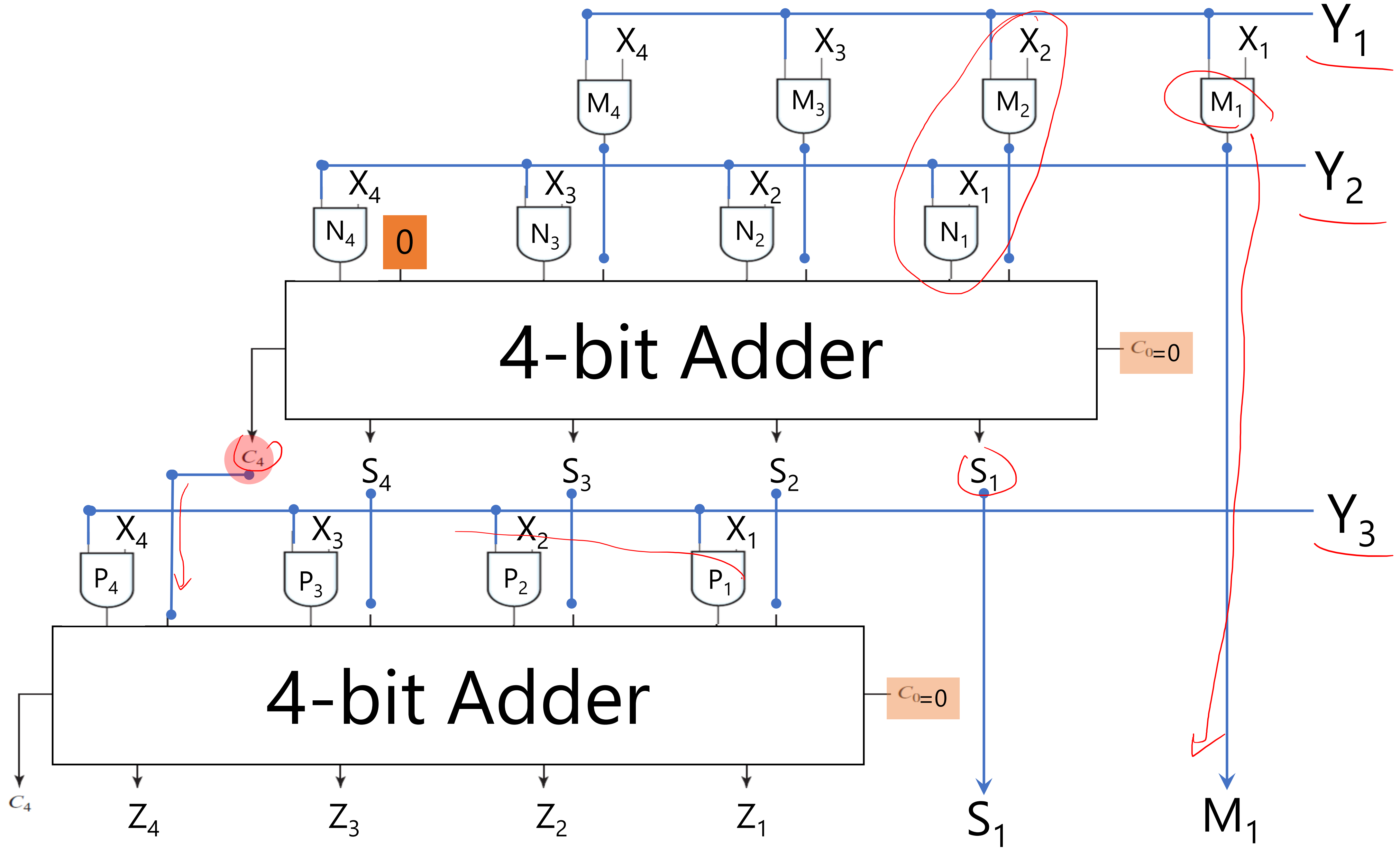
Y1
X4
X3
X2
X1
M4
M3
M2
M1
Y2
X4
X3
X2
X1
N4
0
N3
N2
N1
4-bit Adder
C0 =0
C4
S4
S3
S2
S1
M1

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

$$
\begin{array}{rcccccc}
 & & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_3 & Y_2 & Y_1 \\
\hline
+ & \boxed{C4} & S_4 & S_3 & S_2 & S_1 & M_1 \\
 & P_4 & P_3 & P_2 & P_1 & 0 & 0 \\
\hline
\end{array}
$$

$$
\begin{array}{r@{\quad}cccccc}
 & & & X_4 & X_3 & X_2 & X_1 \\
\times & & & & Y_3 & Y_2 & Y_1 \\
\hline
+ & C4 & S_4 & S_3 & S_2 & S_1 & M_1 \\
 & P_4 & P_3 & P_2 & P_1 & 0 & 0 \\
\hline
 & Z_4 & Z_3 & Z_2 & Z_1 & S_1 & M_1
\end{array}
$$

$$
\begin{array}{r r}
 & X_4X_3X_2X_1 \\
\times & Y_3Y_2Y_1 \\
\hline
+ & \boxed{C4}\ S_4\ S_3\ S_2\ S_1\ M_1 \\
 & P_4\ P_3\ P_2\ P_1\ 0\ \ 0 \\
\hline
 & Z_4\ Z_3\ Z_2\ Z_1\ S_1\ M_1
\end{array}
$$



Y1
Y2
Y3
X4
X3
X2
X1
M4
M3
M2
M1
N4
N3
N2
N1
0
4-bit Adder
C0=0
C4
S4
S3
S2
S1
P4
P3
P2
P1
4-bit Adder
C0=0
C4
Z4
Z3
Z2
Z1
S1
M1

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

Signed?

n-bit $X \times m$-bit Y

Arithmatic
&
Logical Op

Binary Adder, Binary Subtractor, Binary Multiplier

Binary Comparator (Magnitude Comparator)