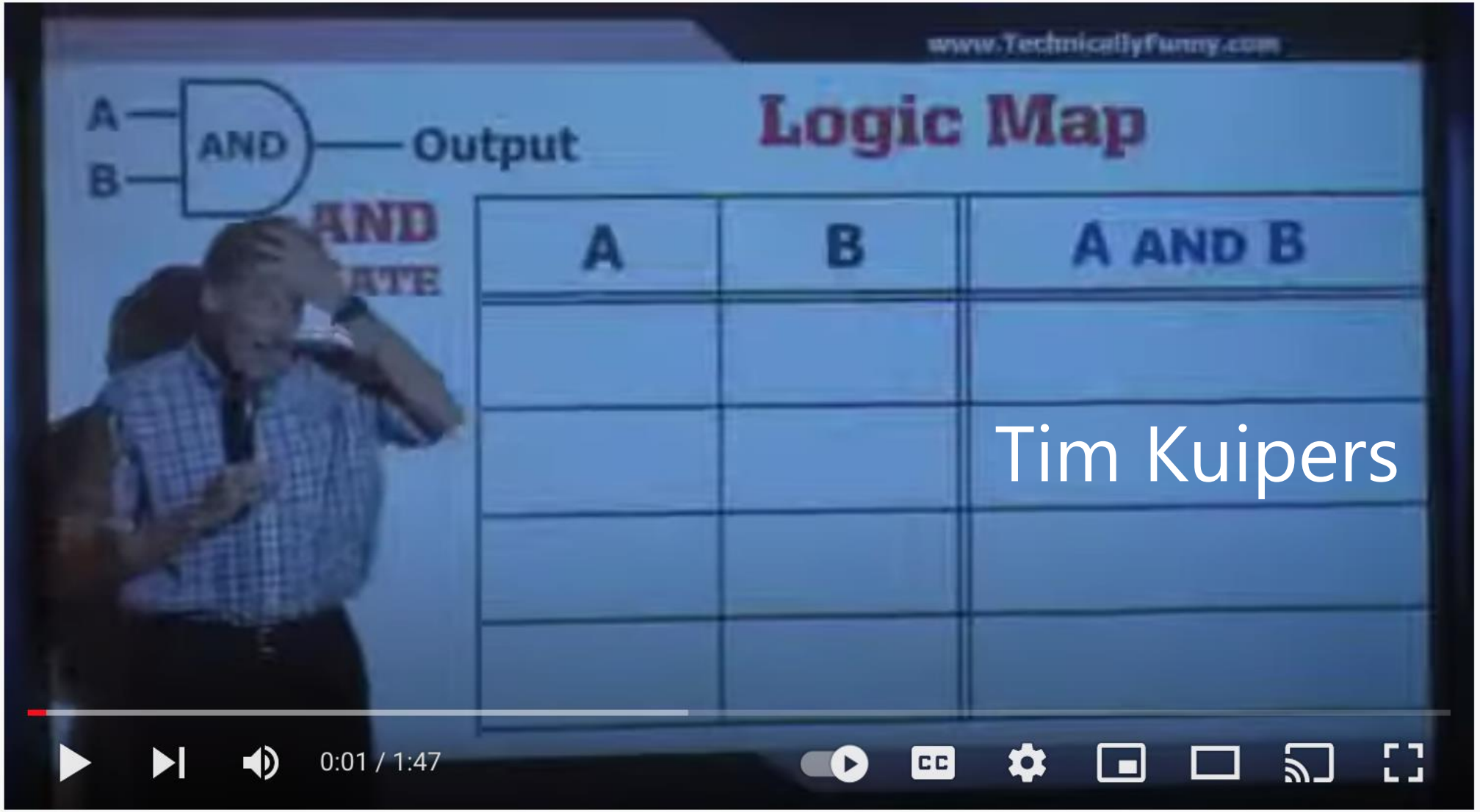
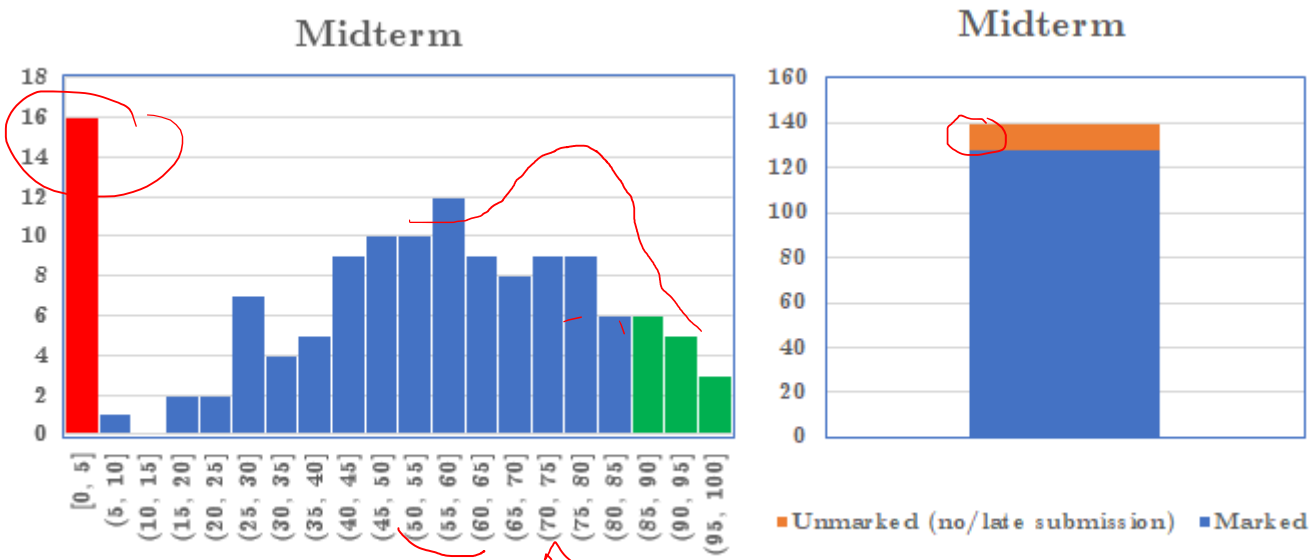


Marriage Logic Map of "SHE" is always "RIGHT"

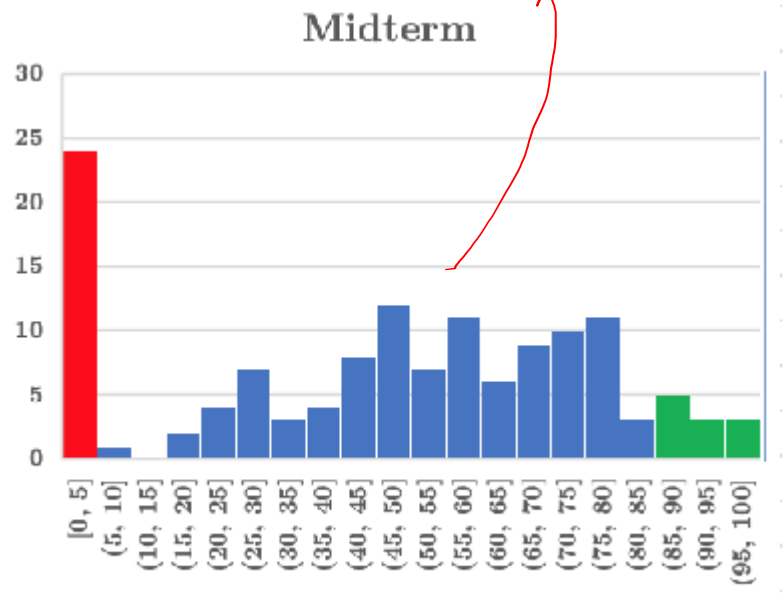
28,308 views • Sep 10, 2020

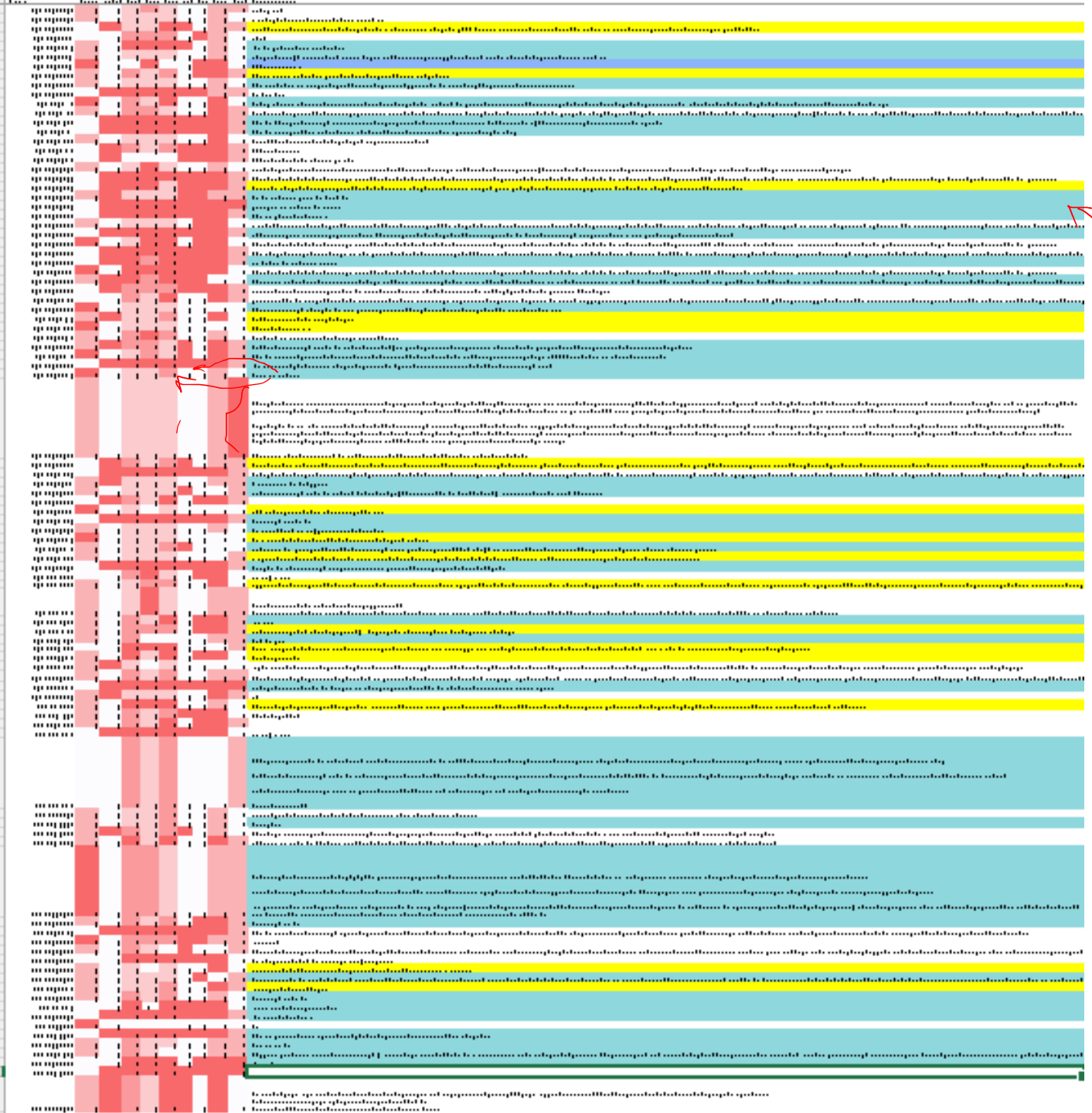
723 DISLIKE SHARE DOWNLOAD SAVE ...

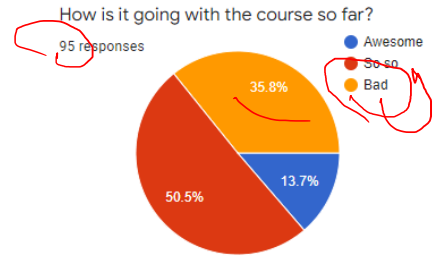
After

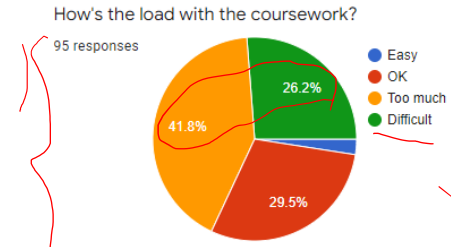


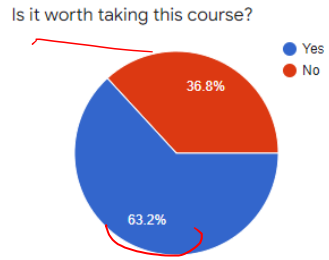
Before

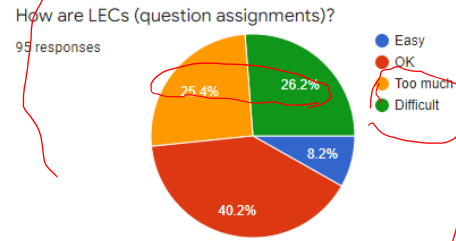


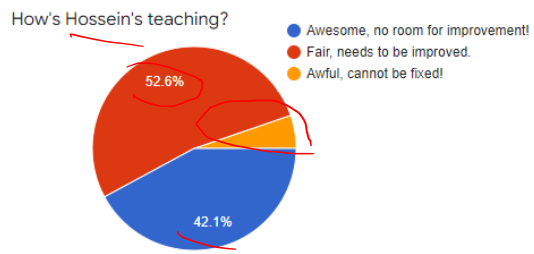


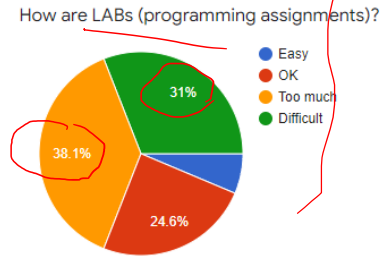

How is it going with the course so far?
95 responses
Awesome
So so
Bad
35.8%
50.5%
13.7%

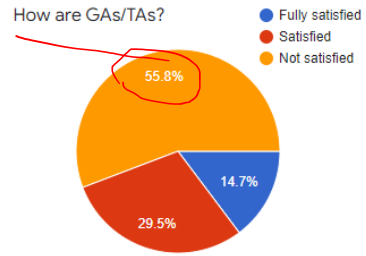

How's the load with the coursework?
95 responses
Easy
OK
Too much
Difficult
26.2%
41.8%
29.5%

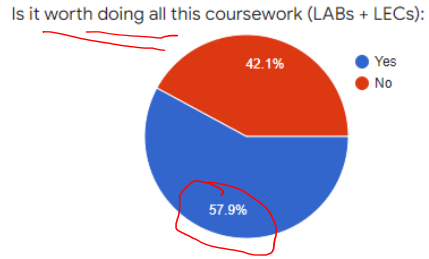

Is it worth taking this course?
Yes
No
36.8%
63.2%

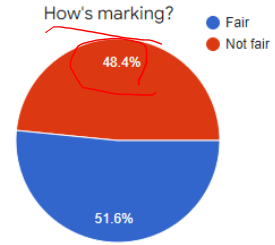

How are LECs (question assignments)?
95 responses
Easy
OK
Too much
Difficult
25.4%
26.2%
8.2%
40.2%


How's Hossein's teaching?
Awesome, no room for improvement!
Fair, needs to be improved.
Awful, cannot be fixed!
52.6%
42.1%


How are LABs (programming assignments)?
Easy
OK
Too much
Difficult
31%
38.1%
24.6%


How are GAs/TAs?
Fully satisfied
Satisfied
Not satisfied
55.8%
14.7%
29.5%


Is it worth doing all this coursework (LABs + LECs):
Yes
No
42.1%
57.9%


How's marking?
Fair
Not fair
48.4%
51.6%

Midterm Exam

More Time
Totally Agree
Sorry!
Because of Class Duration
Will Double or More in Final Exam

Midterm Exam

Difficult

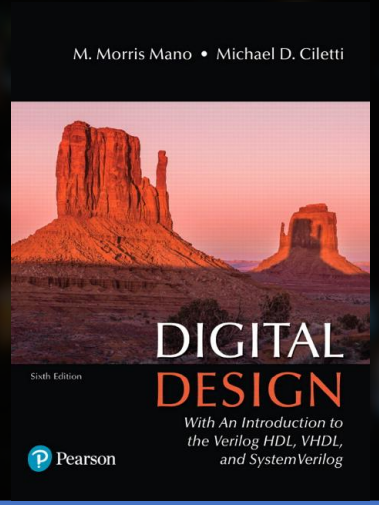
Topics not or Little Covered

→Not Agree

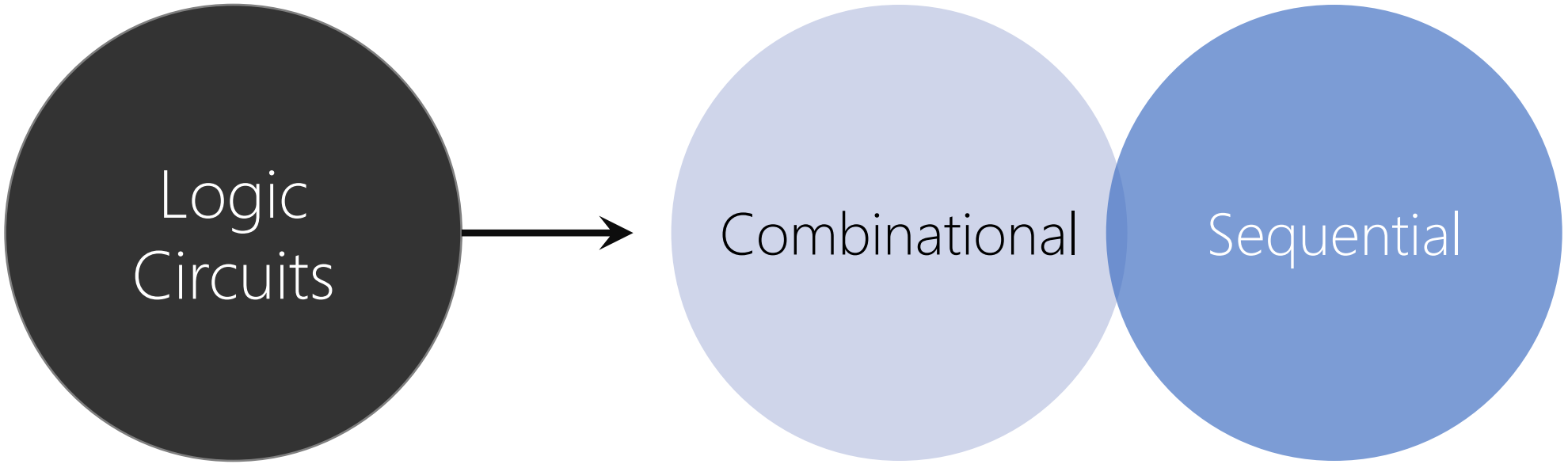
→Study More!

Lectures (Slides + Talk)

Next Session



Chapter 4
Combinational Logic


Logic
Circuits
Combinational
Sequential

Chapter 4 Combinational Logic

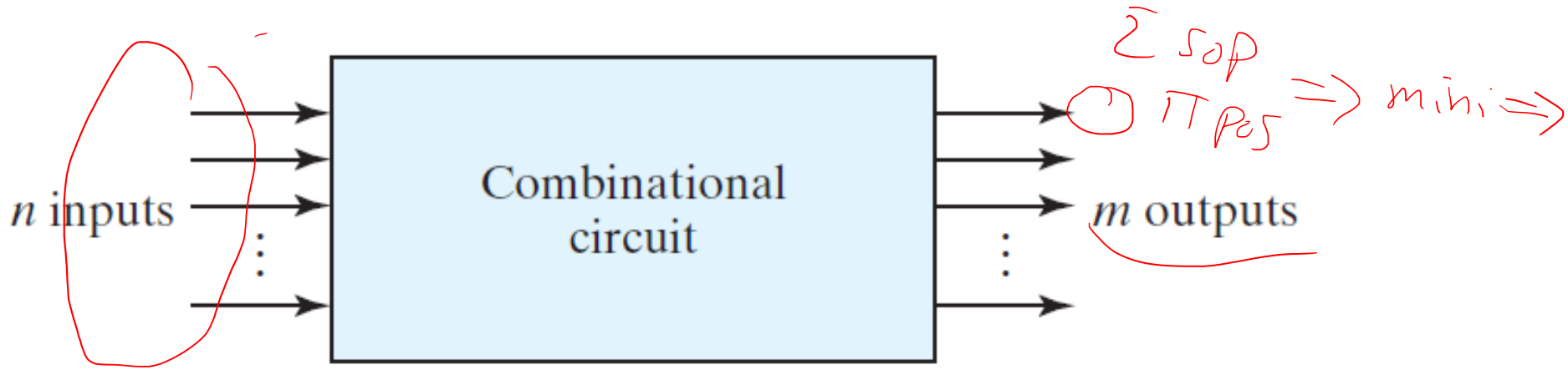
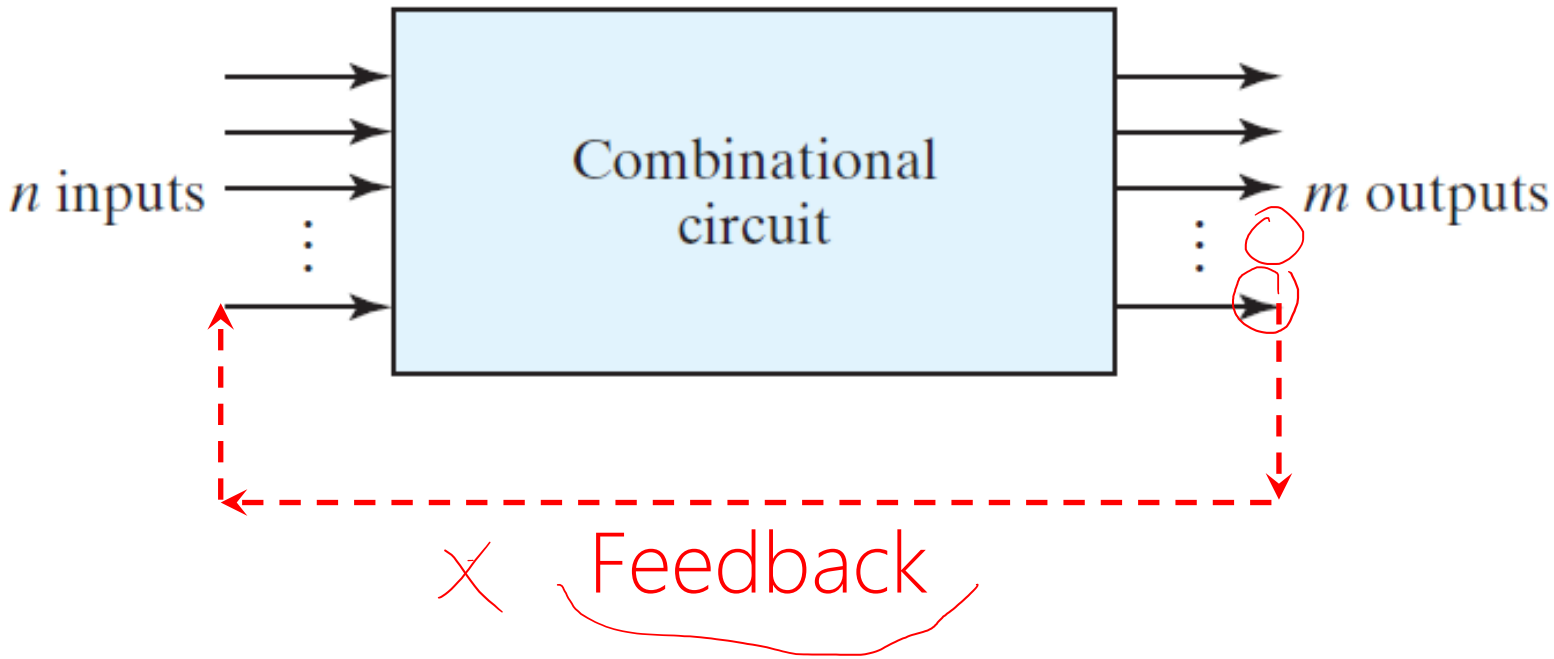


FIGURE 4.1
Block diagram of combinational circuit

Sequential Logic

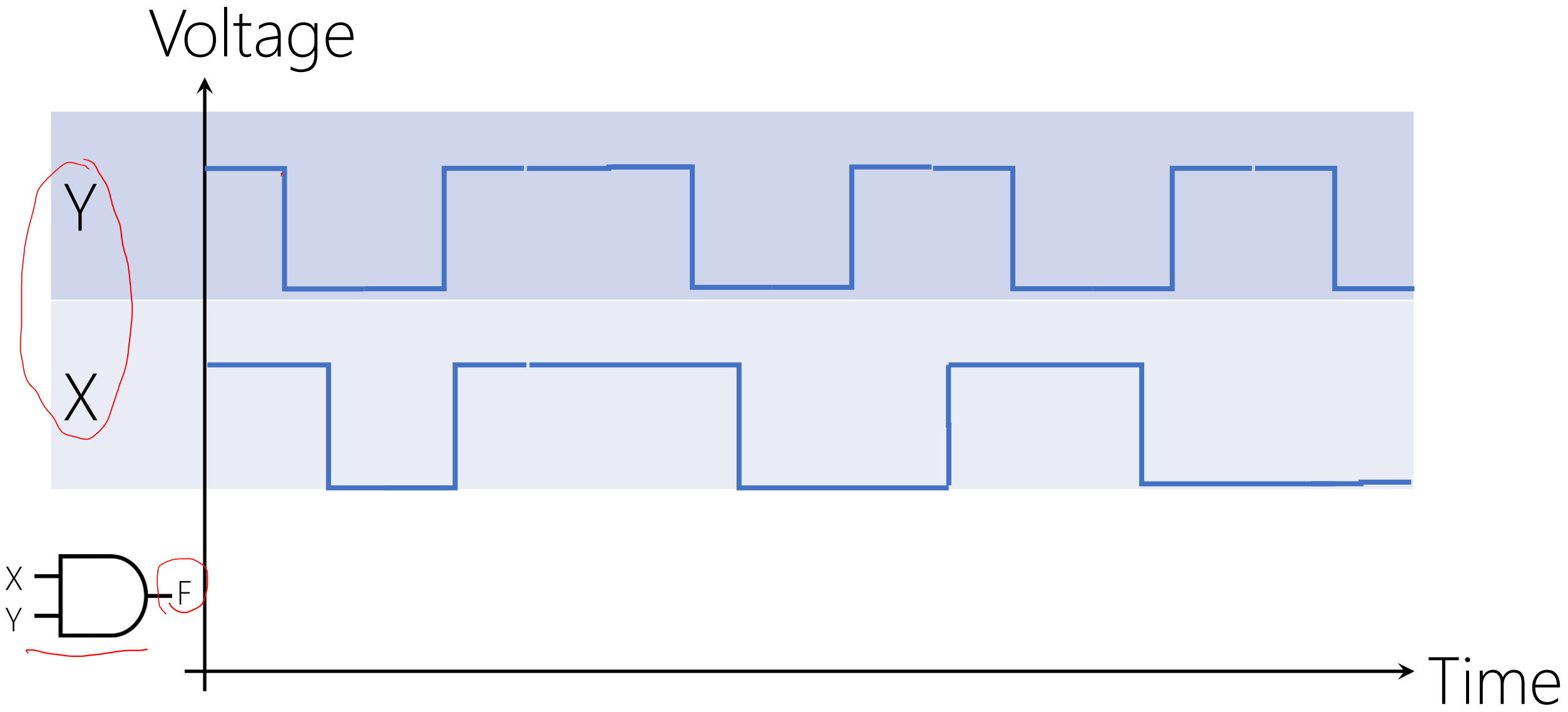


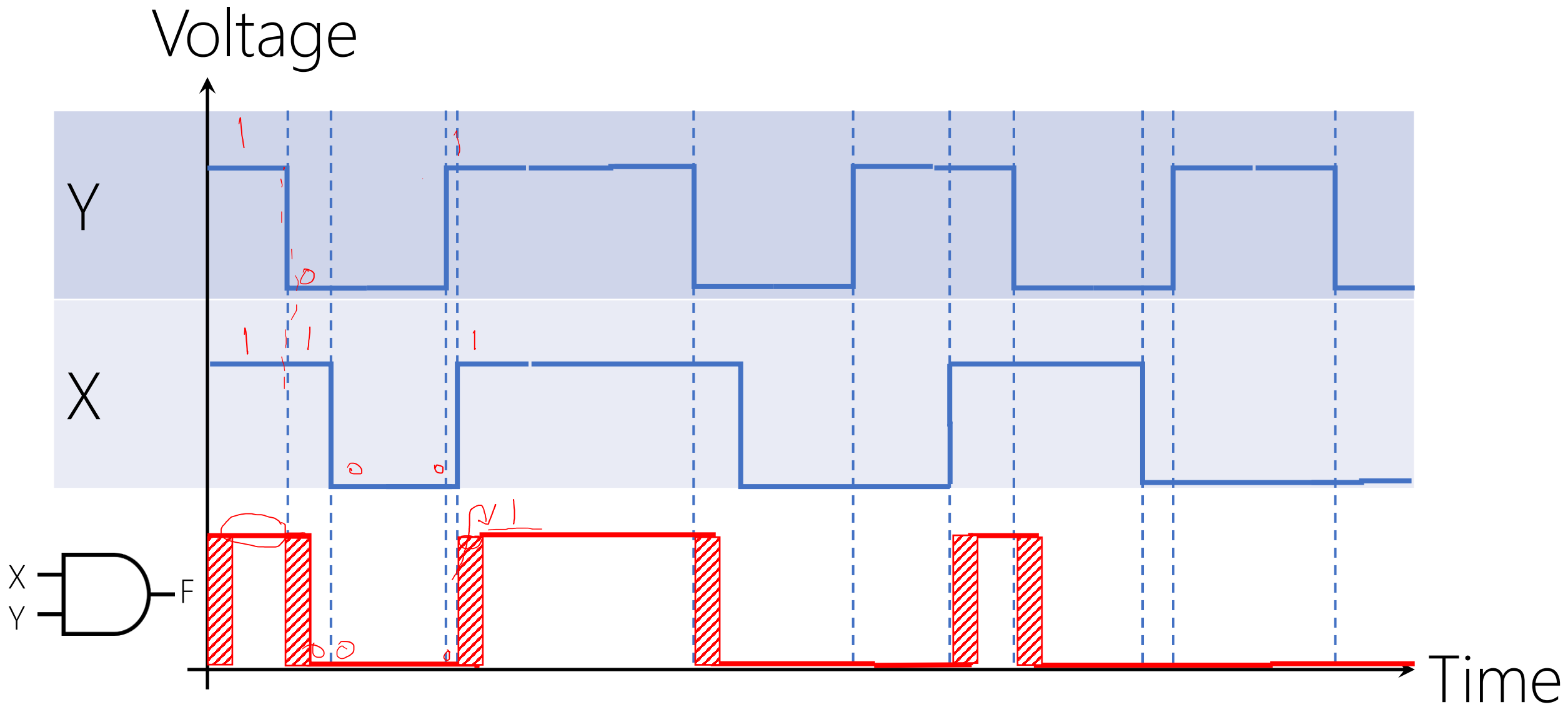
Combinational Logic

aka. Combinational Circuit

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

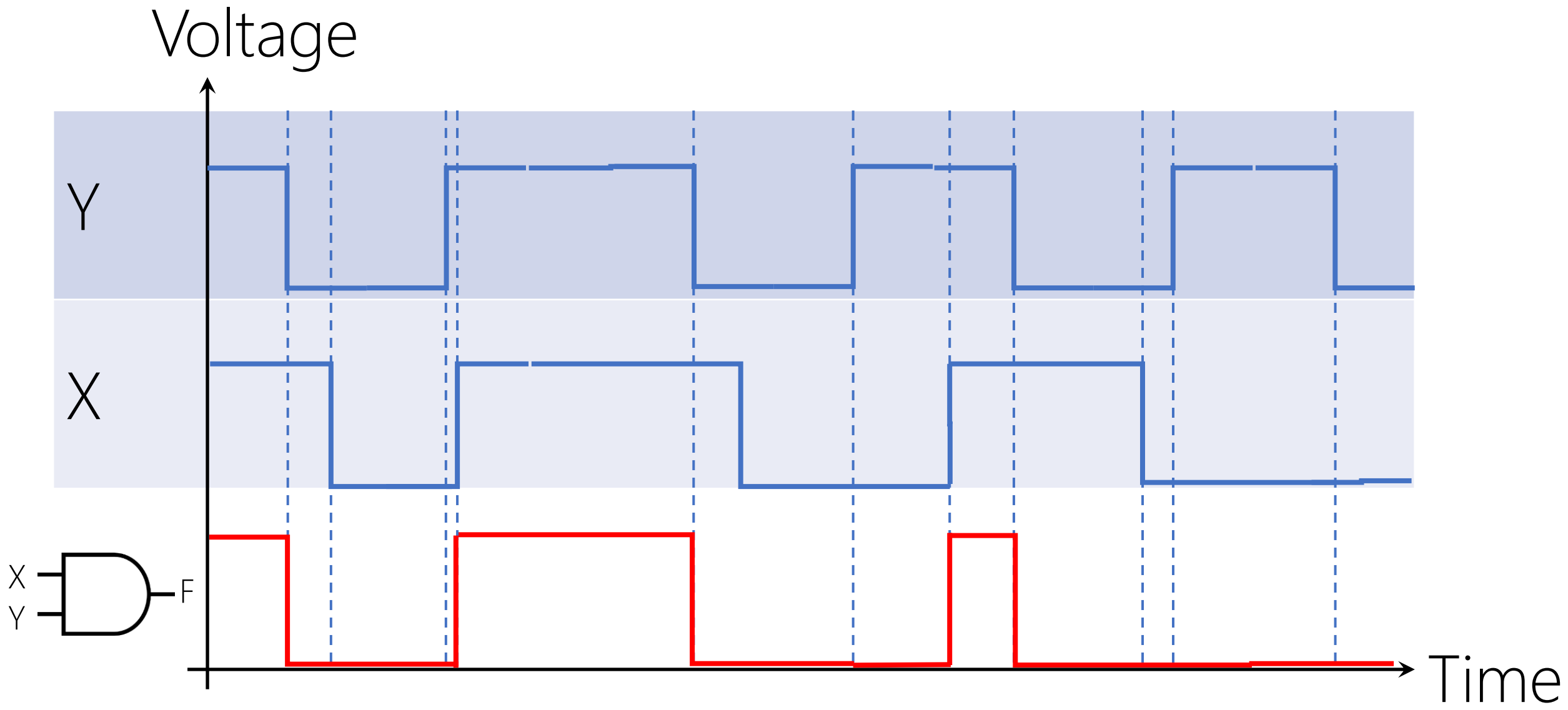
A combinational circuit performs an operation that can be specified logically by a set of Boolean functions.


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

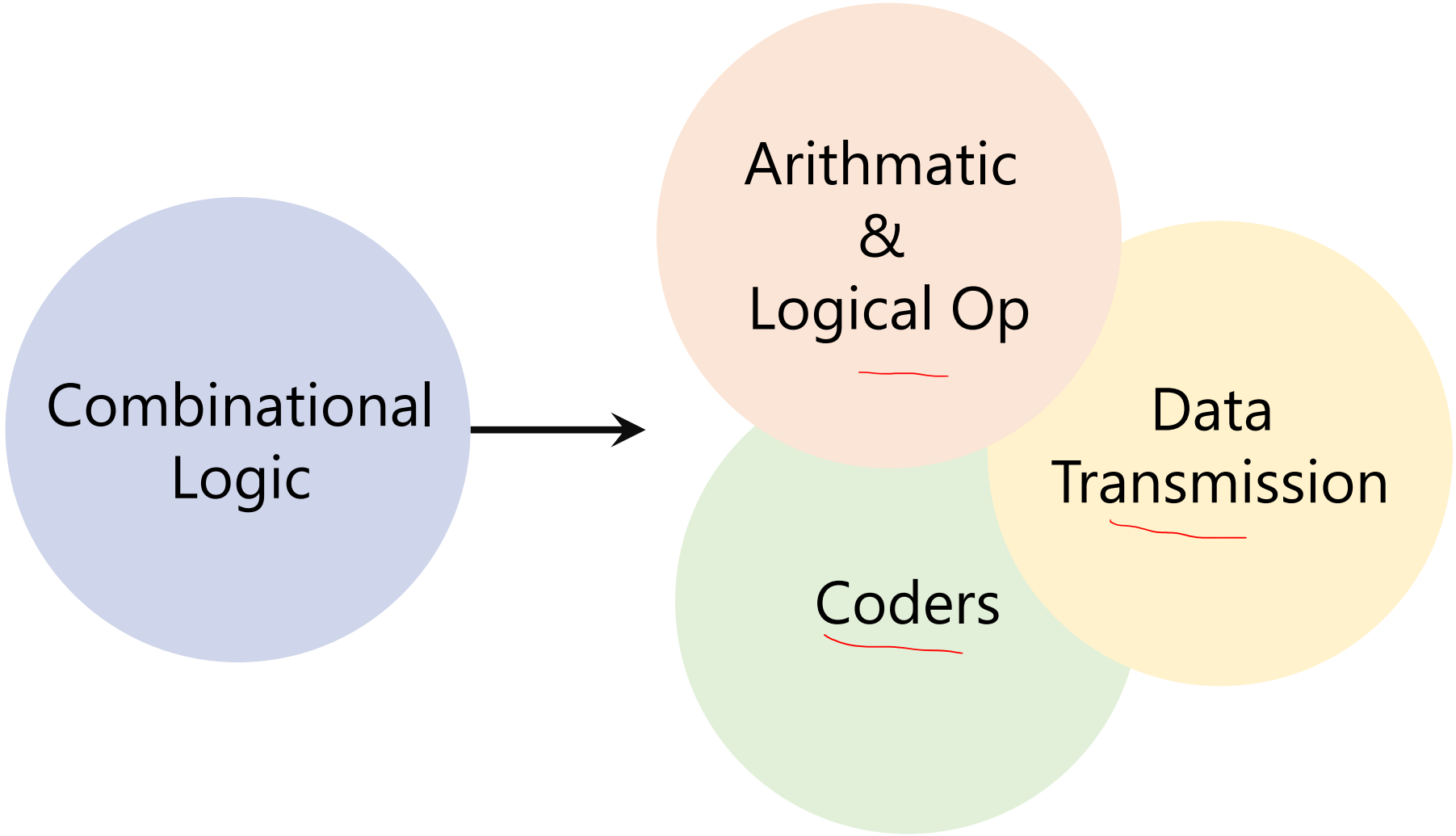
What we've done so far

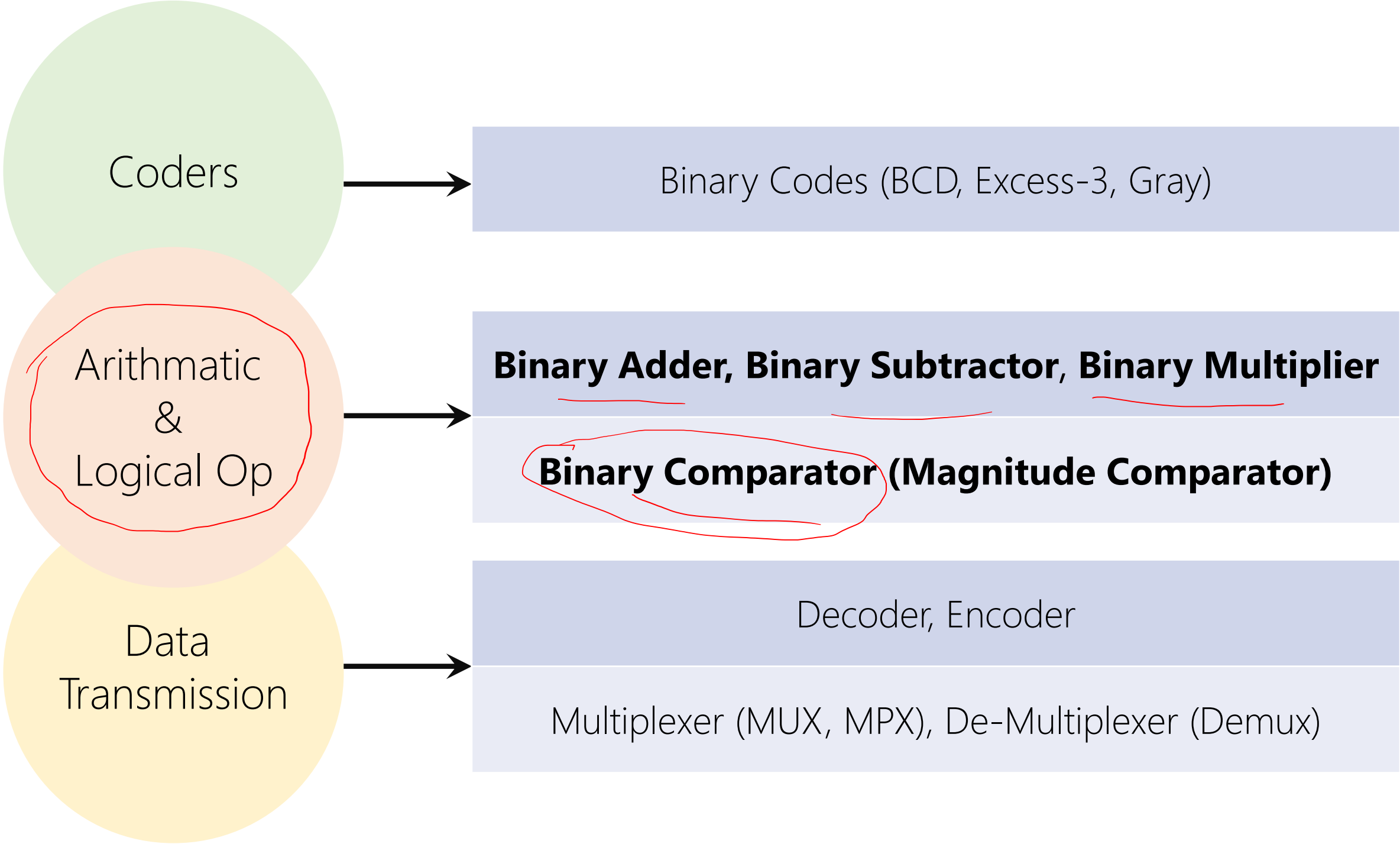
Combinational Logic aka. Combinational Circuit

Design a combinational logic circuit:

1. Truth Table (Inputs, Outputs)
2. Output Boolean Functions (SoP: ∑m | PoS: ∏M)
3. Minimization
   - Algebraically | K-Map | Quine-McCluskey

4. Logic Diagram | Circuit


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)

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?

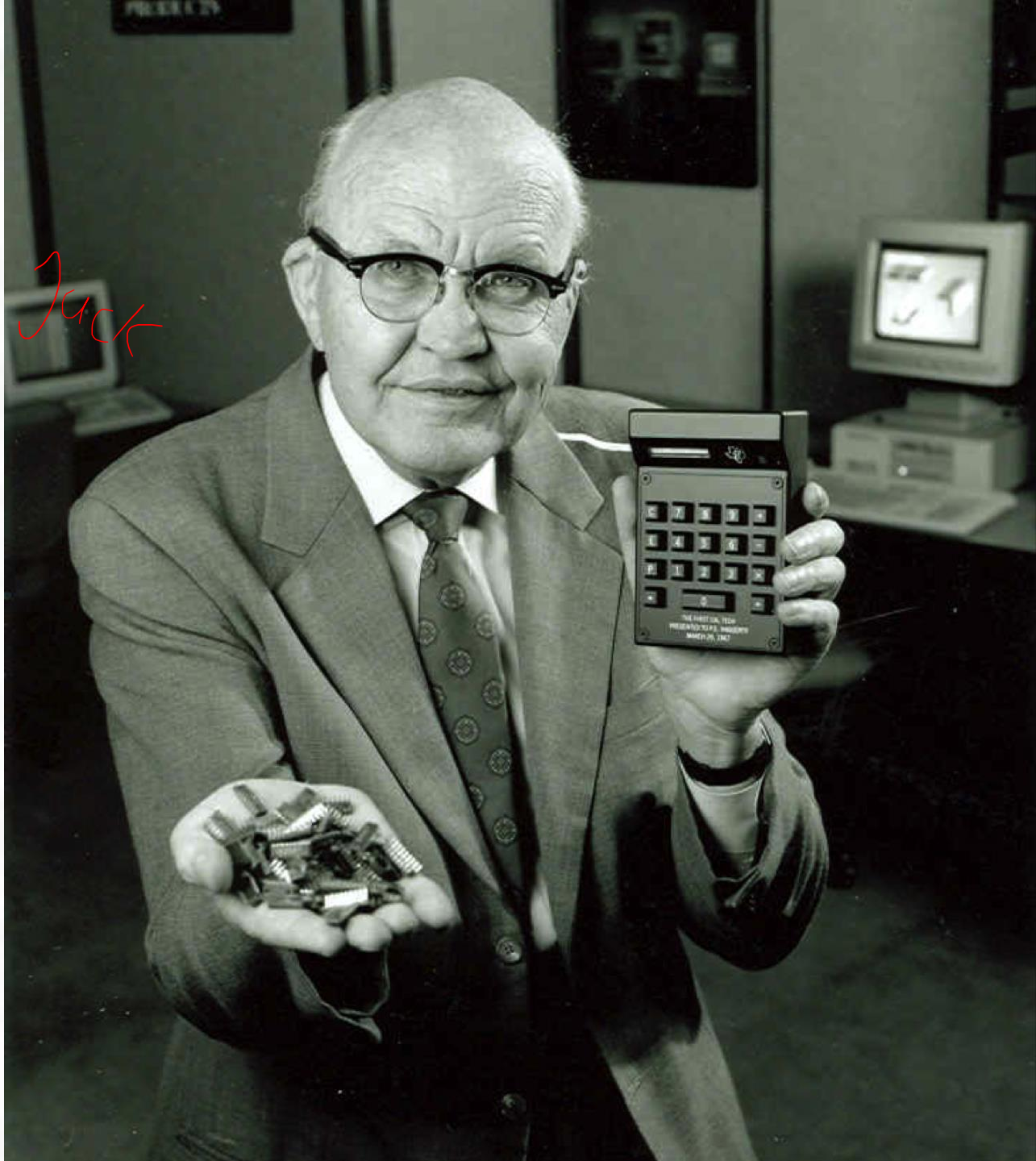
Bowmar was the first American company that made and sold their own line of portable electronic machines.

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



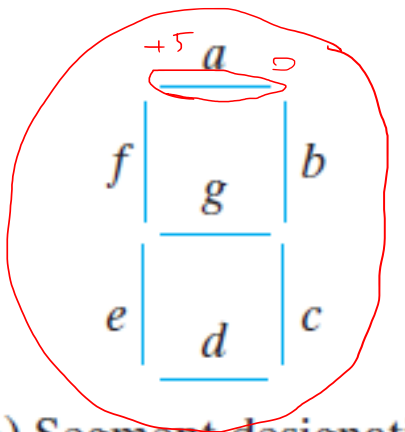
Calculator

Decimal Interface (External)

Binary Calculation (Internal)

Decimal vs. Binary Systems

4.9 An ABCD-to-seven-segment decoder is a combinational circuit that converts a decimal digit in BCD to an appropriate code for the selection of segments in an indicator used to display the decimal digit in a familiar form. The seven outputs of the decoder *(a, b, c, d, e, f, g)* select the corresponding segments in the display, as shown in Fig. P4.9(a). The numeric display chosen to represent the decimal digit is shown in Fig. P4.9(b). Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder using a minimum number of gates. The six invalid combinations should result in a blank display. (HDL—see Problem 4.51.)



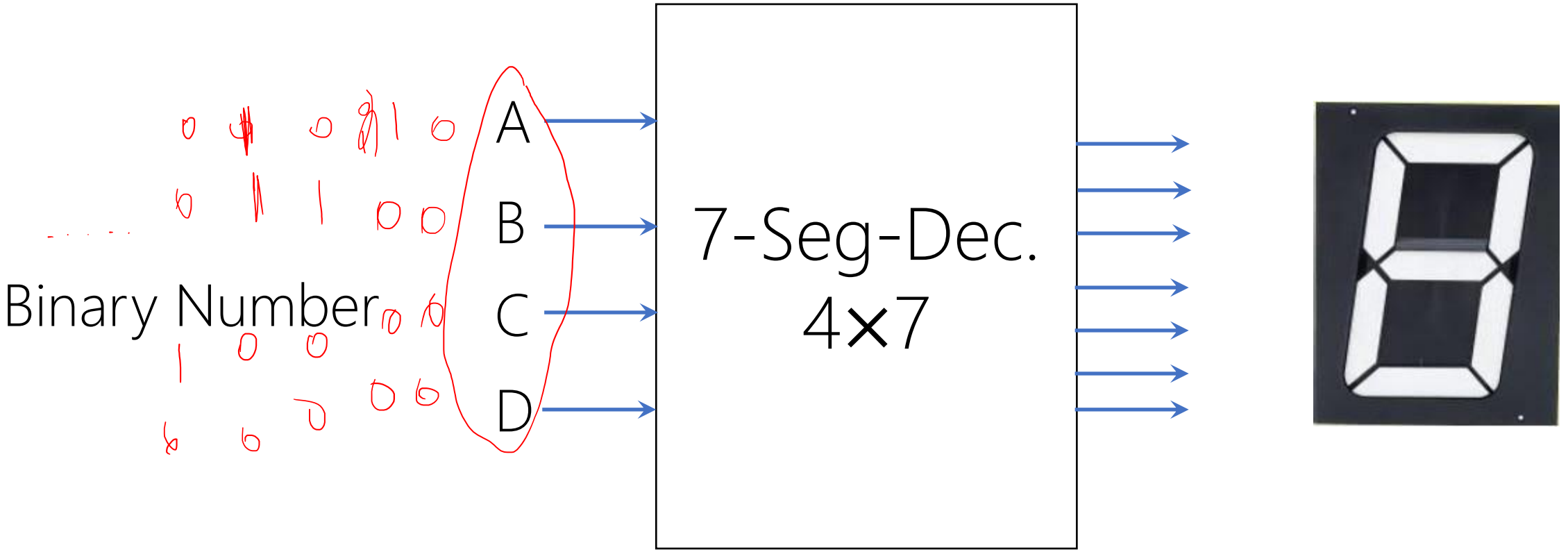
(a) Segment designation

(b) Numerical designation for display

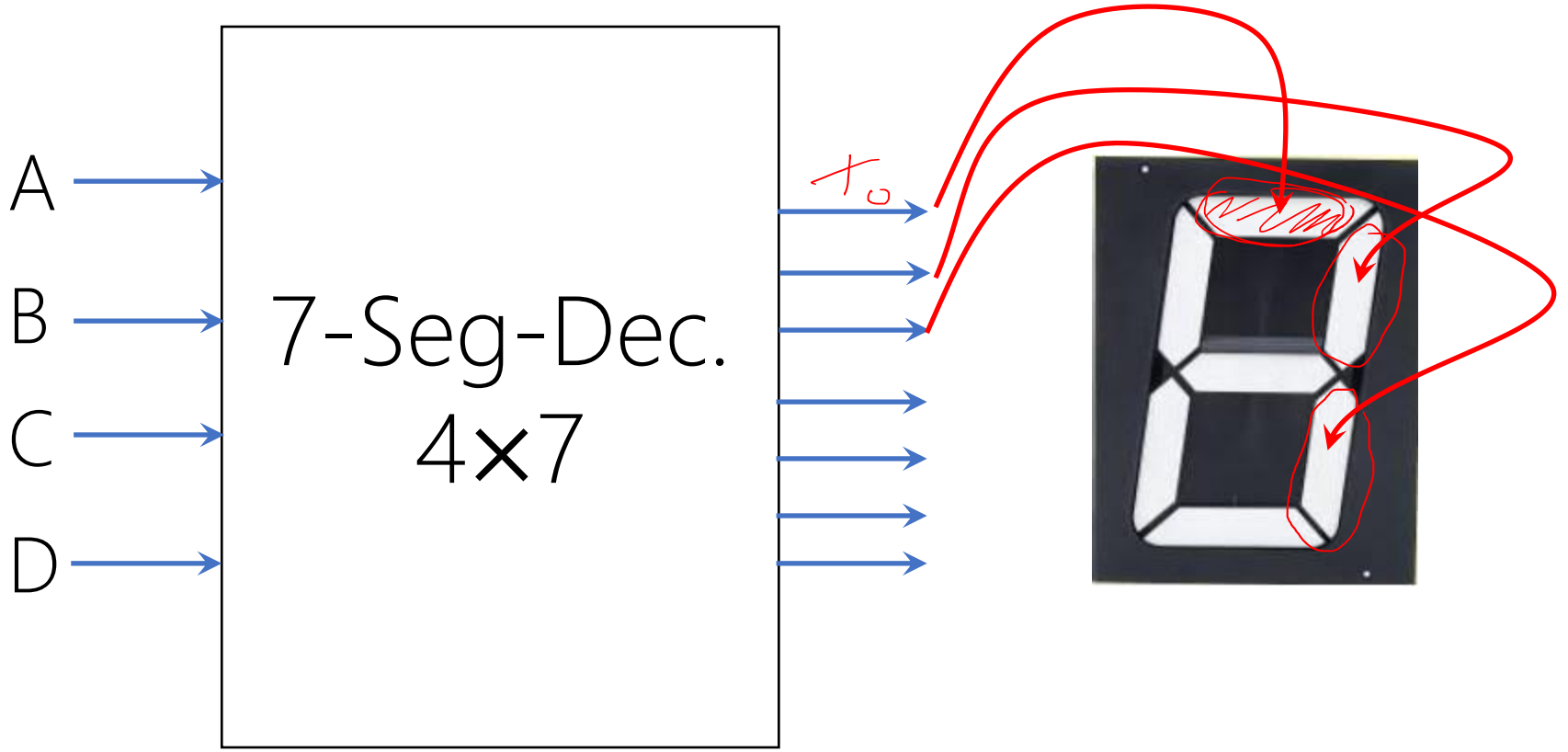
FIGURE P4.9

Combinational Logic

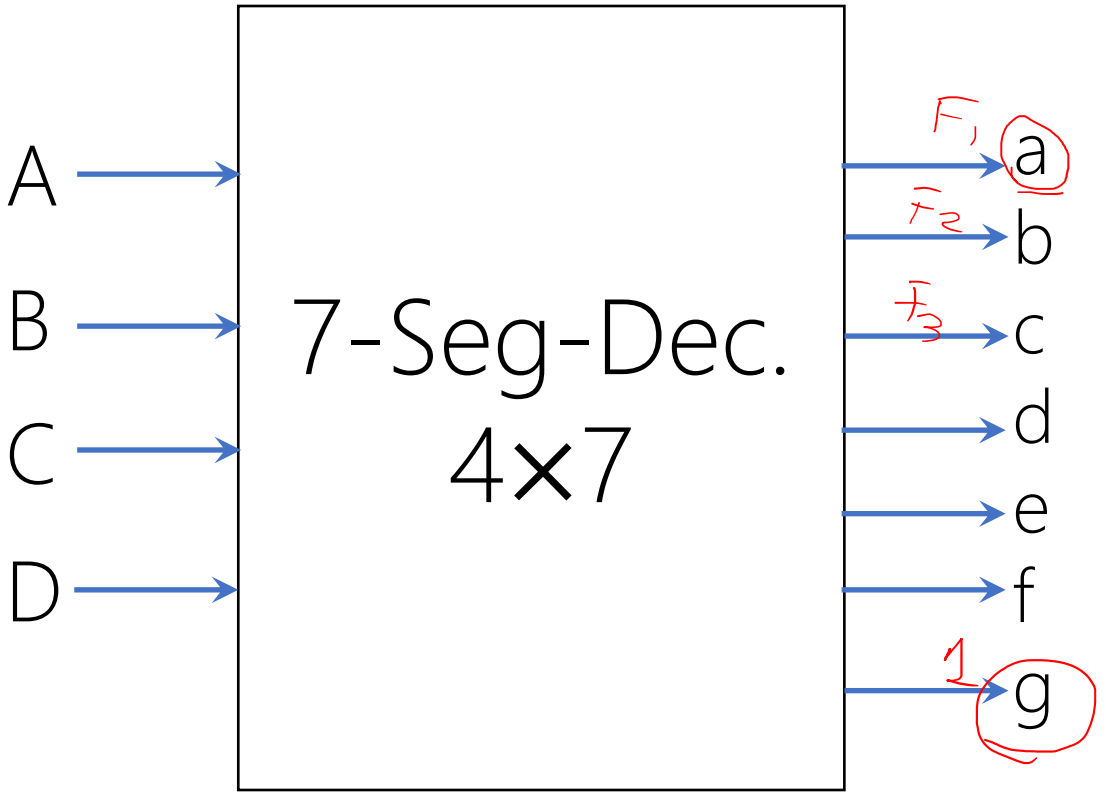
Display Decoder

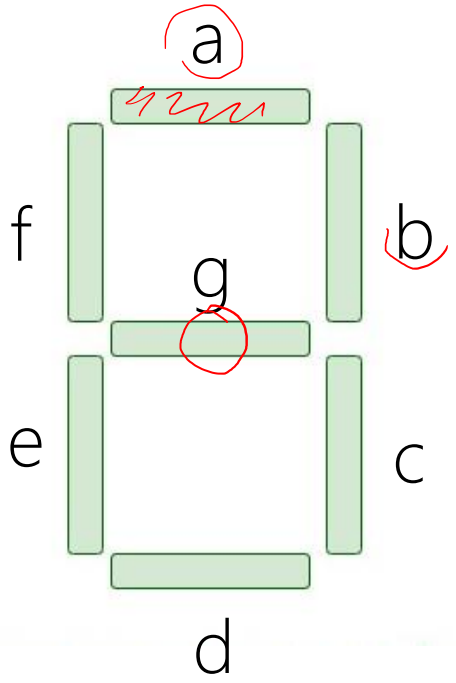

Binary Number
A
B
C
D
7-Seg-Dec.
4×7

Binary Number

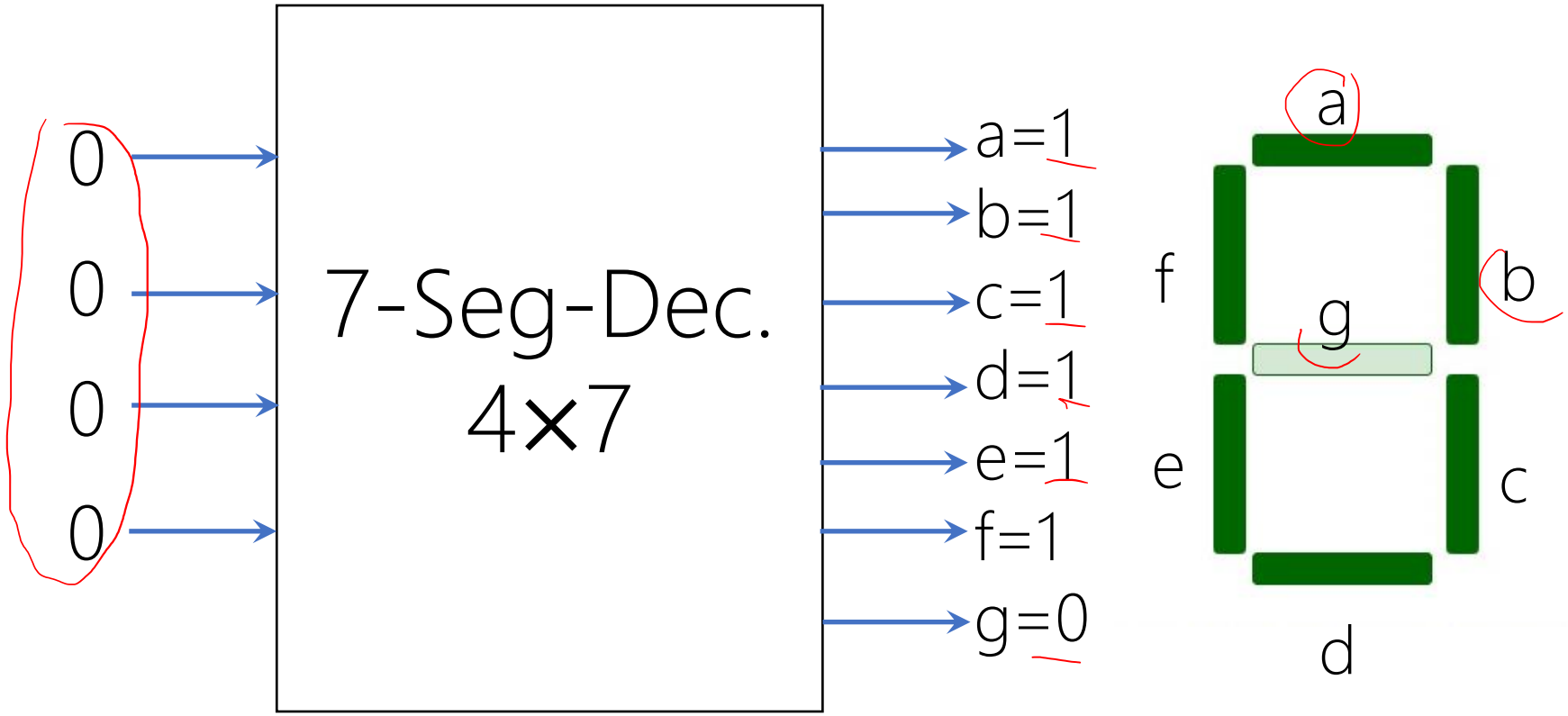


Binary Number

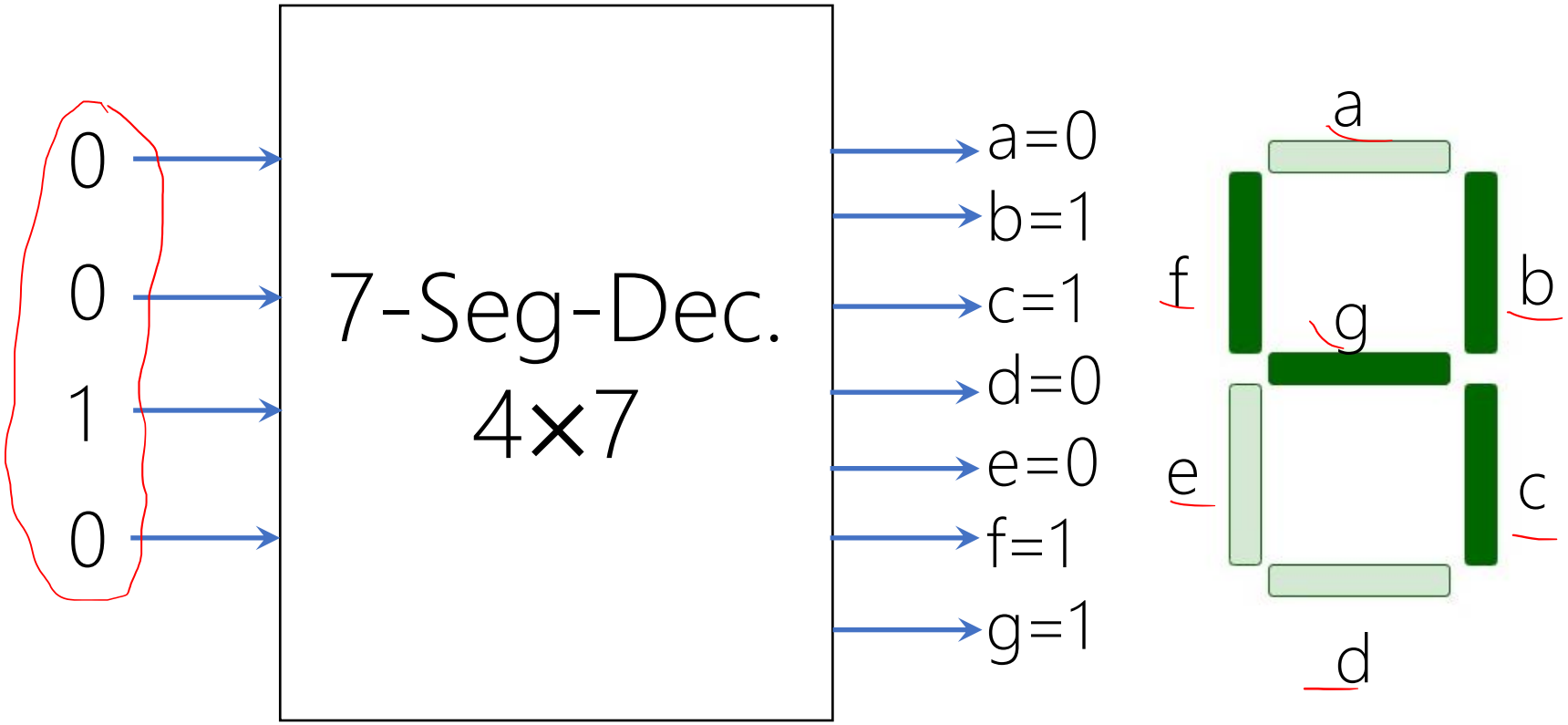




Binary Number



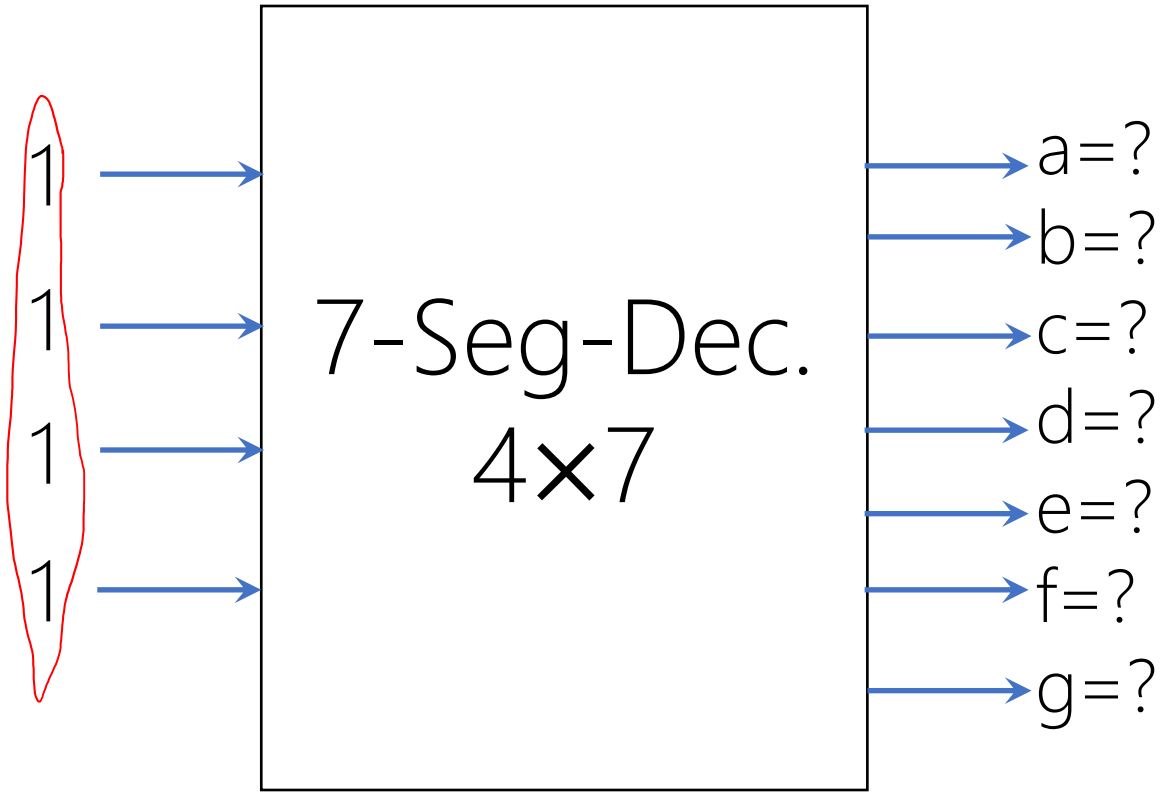
Binary Number



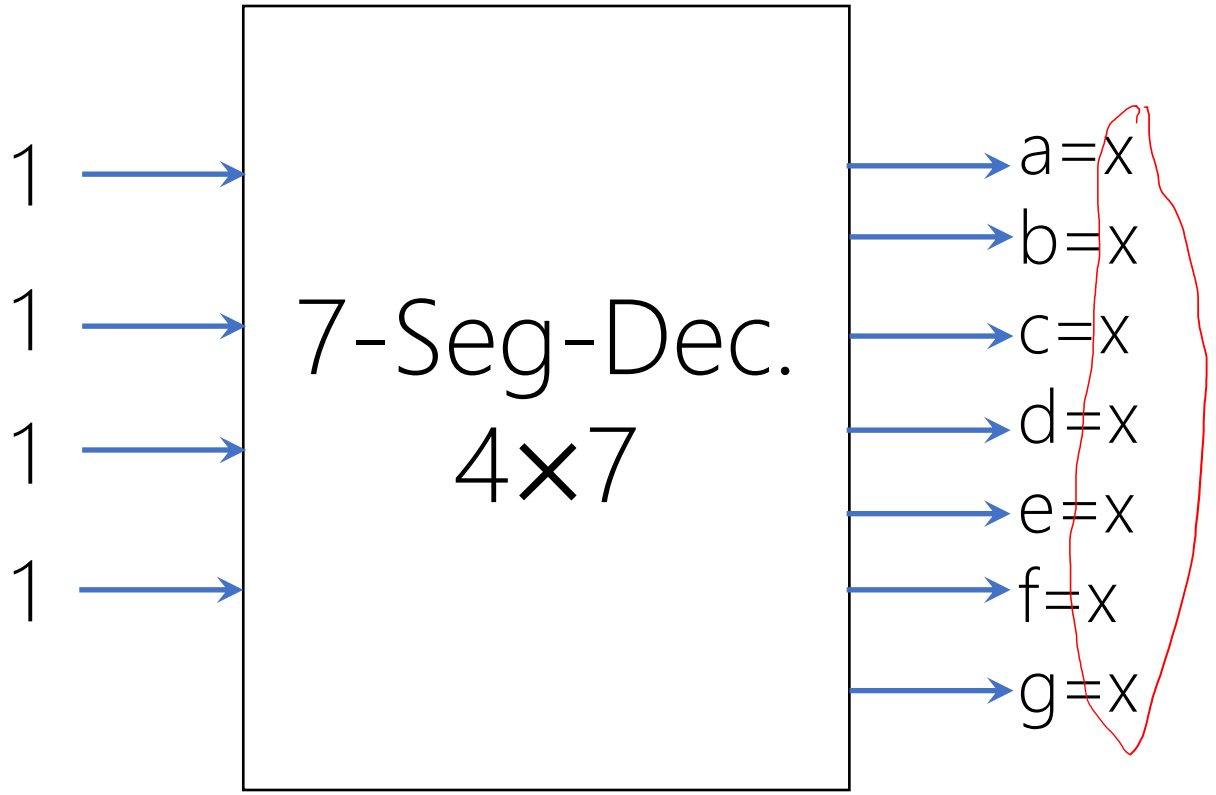
Binary Number

15

> 9



Binary Number



From 10 to 15, don't care conditions!

Truth Table

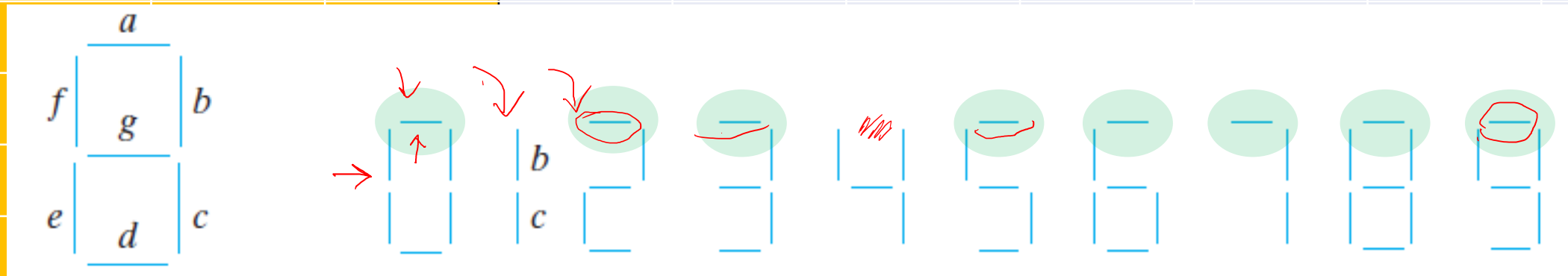
Display Decoder

D	C	B	A	a	b	c	d	e	f	g
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1						
0	0	0	1	0						
0	0	1	0	1						
0	0	1	1	1						
0	1	0	0	0						
0	1	0	1	1						
0	1	1	0	1						
0	1	1	1	1						
1	0	0	0	1						
1	0	0	1	1						
1	0	1	0	X	X	X	X	X	X	X
1										X
1										X
1										X
1										X
1	1	1	1	X	X	X	X	X	X	X

$$a_{SOP} = \sum(0,2,3,5,6,7,8,9) \Rightarrow$$

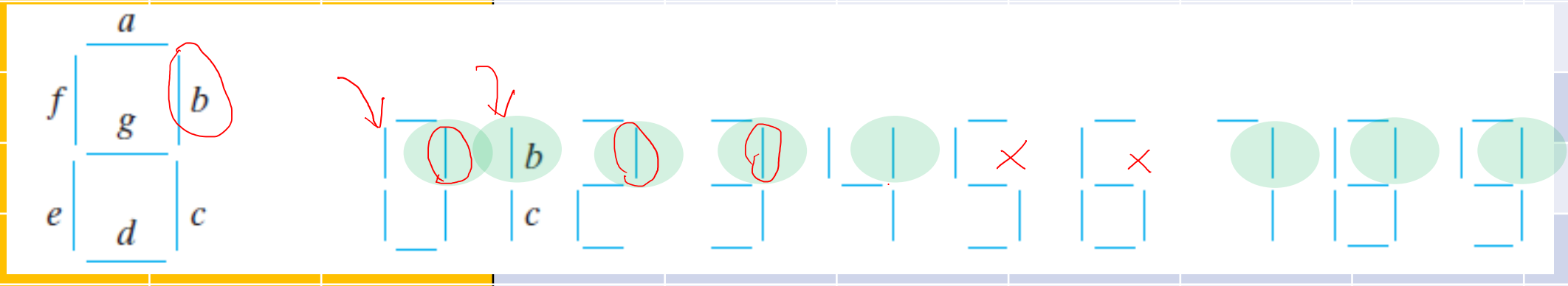
$$a_{POS} = \Pi(1,4)$$



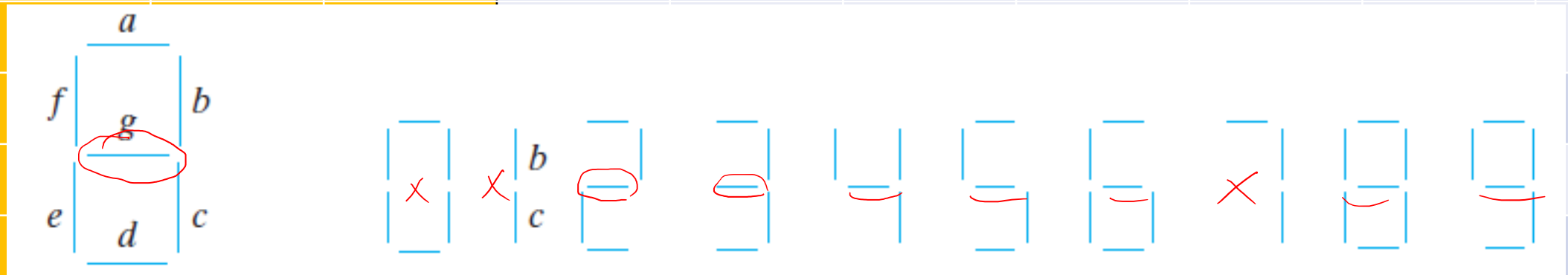
D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1					
0	0	0	1	0	1					
0	0	1	0	1	1					
0	0	1	1	1	1					
0	1	0	0	0	1					
0	1	0	1	1	0					
0	1	1	0	1	0					
0	1	1	1	1	1					
1	0	0	0	1	1					
1	0	0	1	1	1					
1	0	1	0	X	X	X	X	X	X	X
1										X
1										X
1										X
1										X
1	1	1	1	X	X	X	X	X	X	X

$b_{SOP} = \sum(0,1,2,3,4,7,8,9) + d(10,11,12,13)$

$b_{POS} = \prod M(5,6) + D(10,11,15)$



D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1										X
1										X
1										X
1										X
1	1	1	1	X	X	X	X	X	X	X



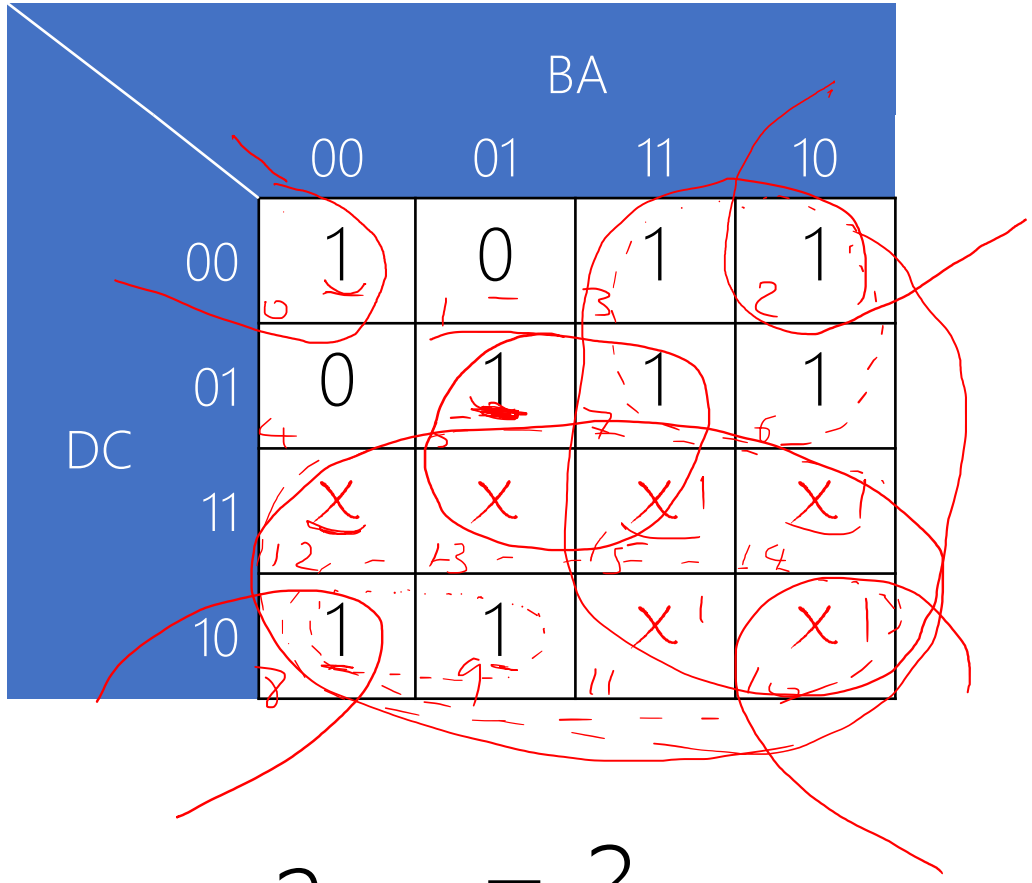
Boolean Functions ☹

Display Decoder

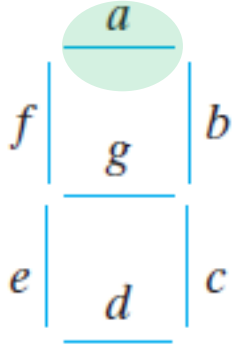
? × ?-Variable K-Map

Display Decoder

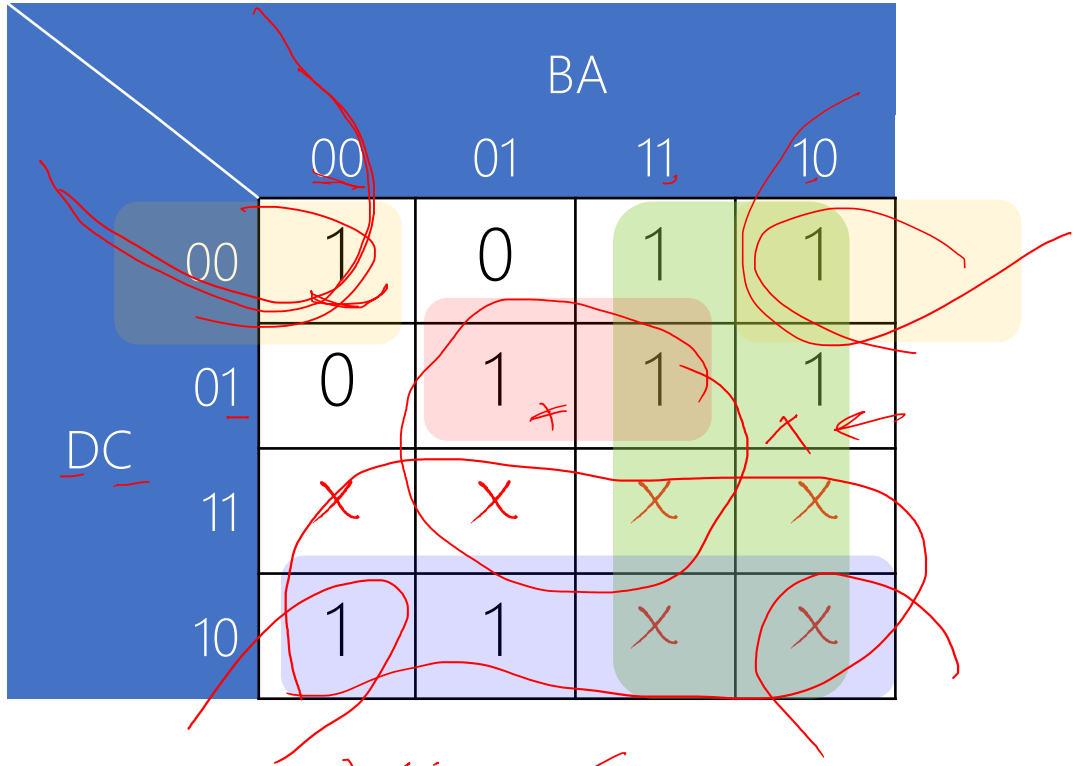
D	C	B	A	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X



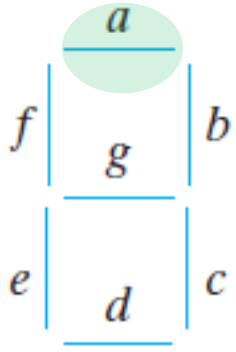
$a_{sop} = ?$



D	C	B	A	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X



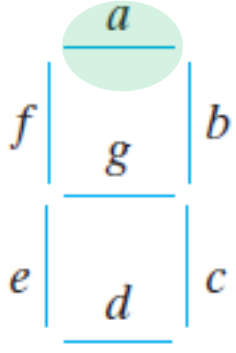
$$a_{sop} = B + DC' + D'CA + D'C'A'$$



D	C	B	A	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

DC \ BA	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

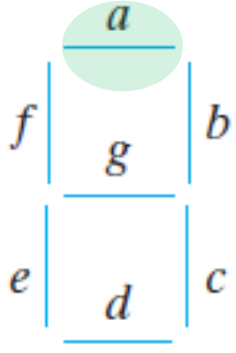
$$a_{sop} = B + D + CA + C'A'$$



D	C	B	A	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

DC \ BA	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

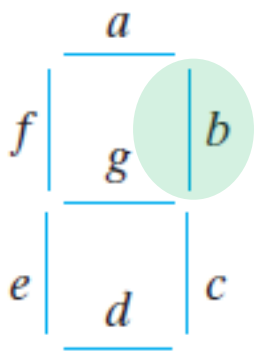
$$a_{pos} = (CB'A' + D'C'B'A)'$$
$$= (C' + B + A)(D + C + B + A')$$



D	C	B	A	b
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

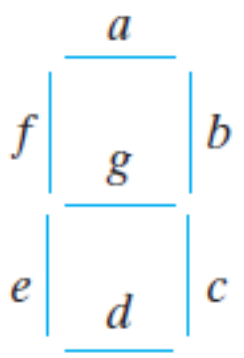
DC \ BA	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	X	X	X	X
10	1	1	X	X

$$b_{pos} = (CA)' = C' + A'$$

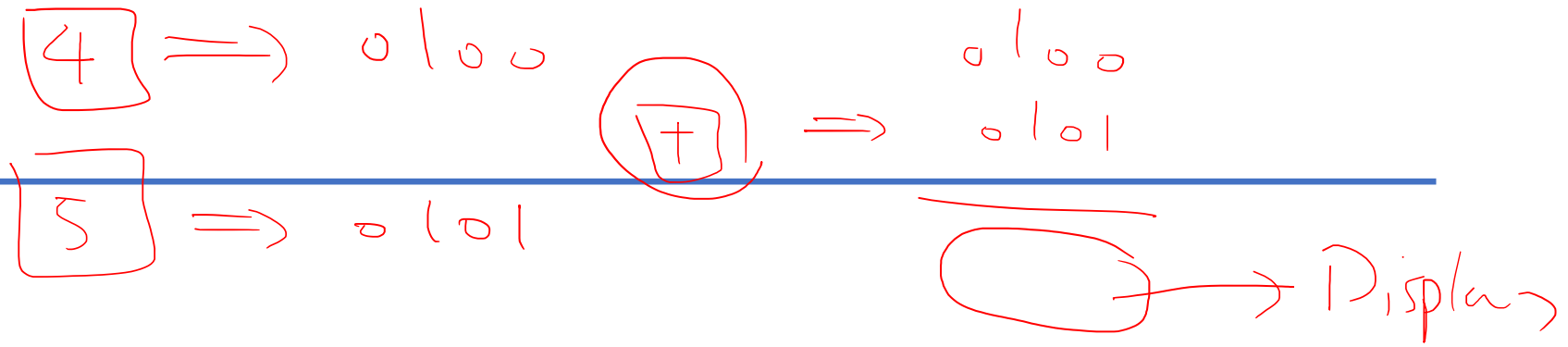


7 × 4-Variable K-Map

Display Decoder



At Home!



Binary Adder

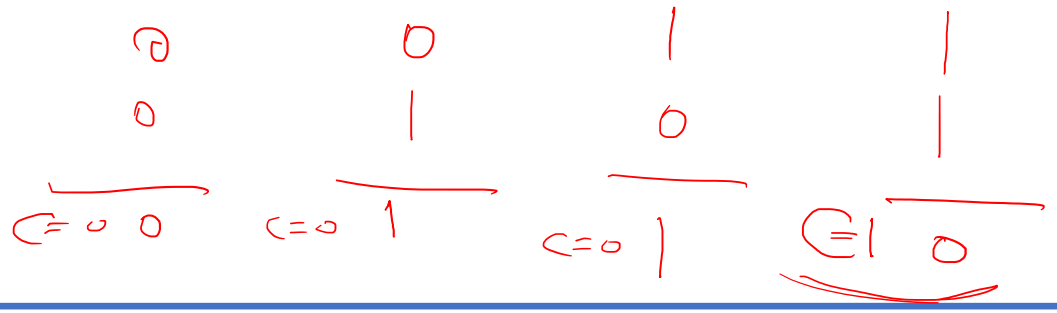
Design a logic circuit that adds two binary digits (bit).

Range of inputs:
2 bits

$x \in \{0,1\}$

$y \in \{0,1\}^+$

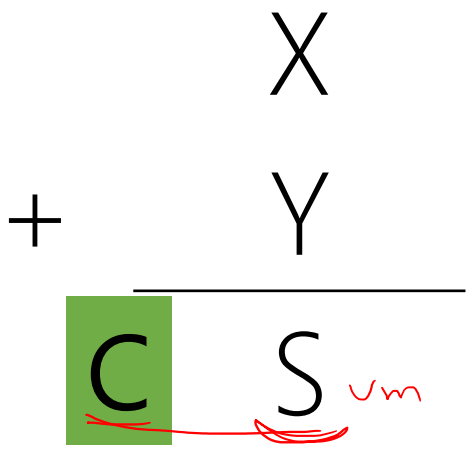
Input binary variables: X and Y



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

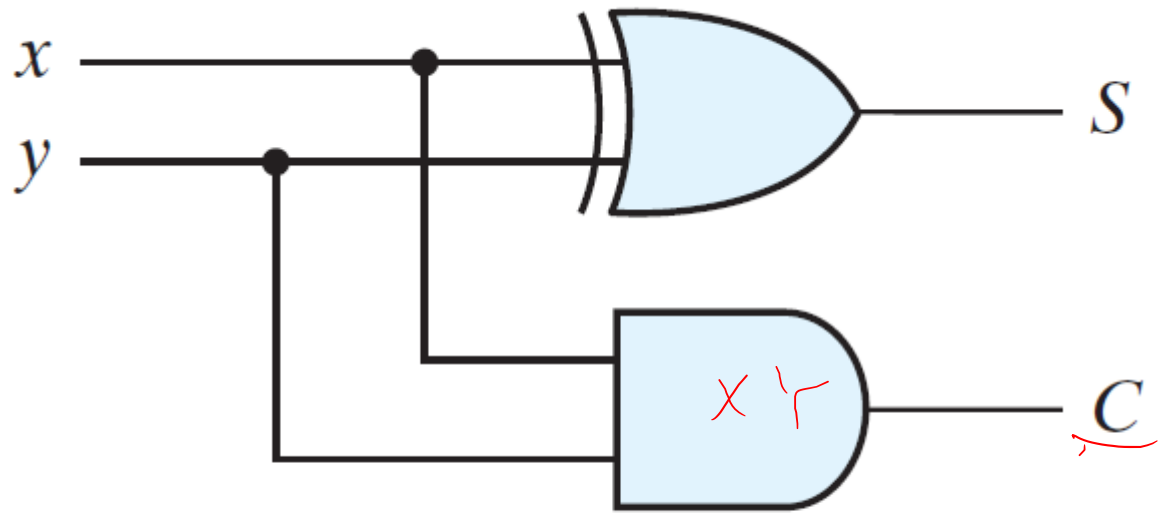

X
+ Y
C S
Um

Output binary variables: Carry and Sum

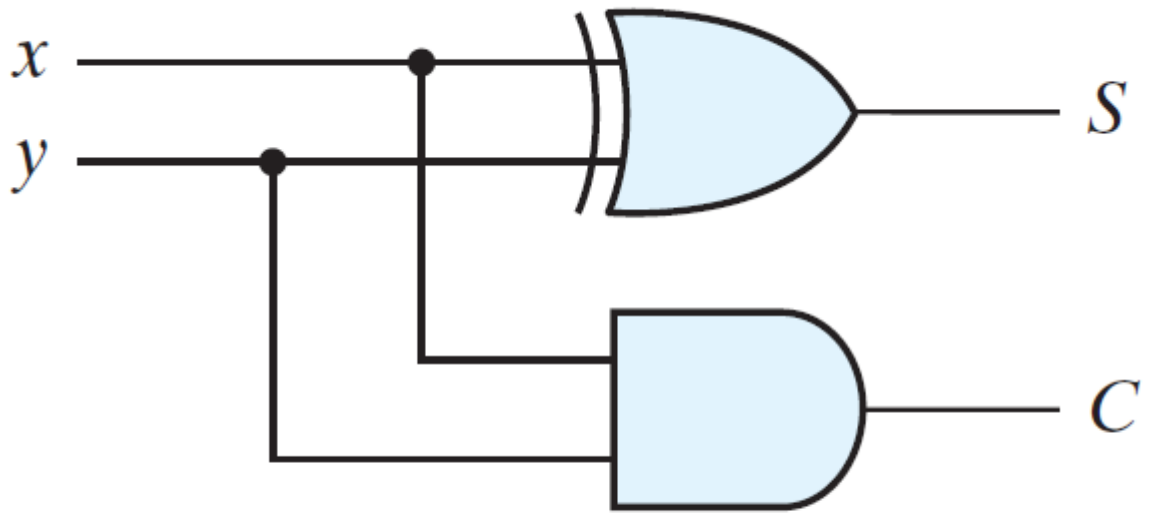
F_1 F_2

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

x+y ⇒ x⊕y

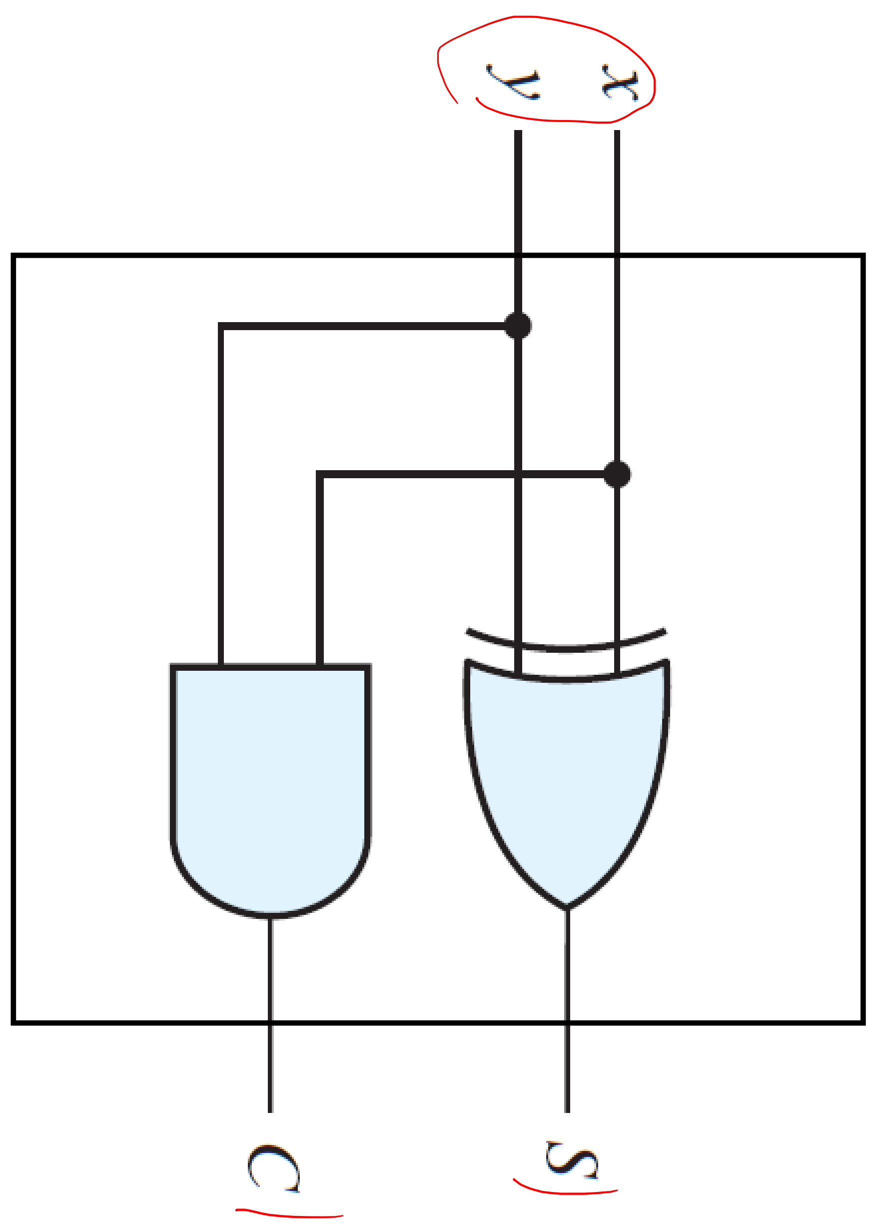


$$S = x \oplus y$$
$$C = xy$$

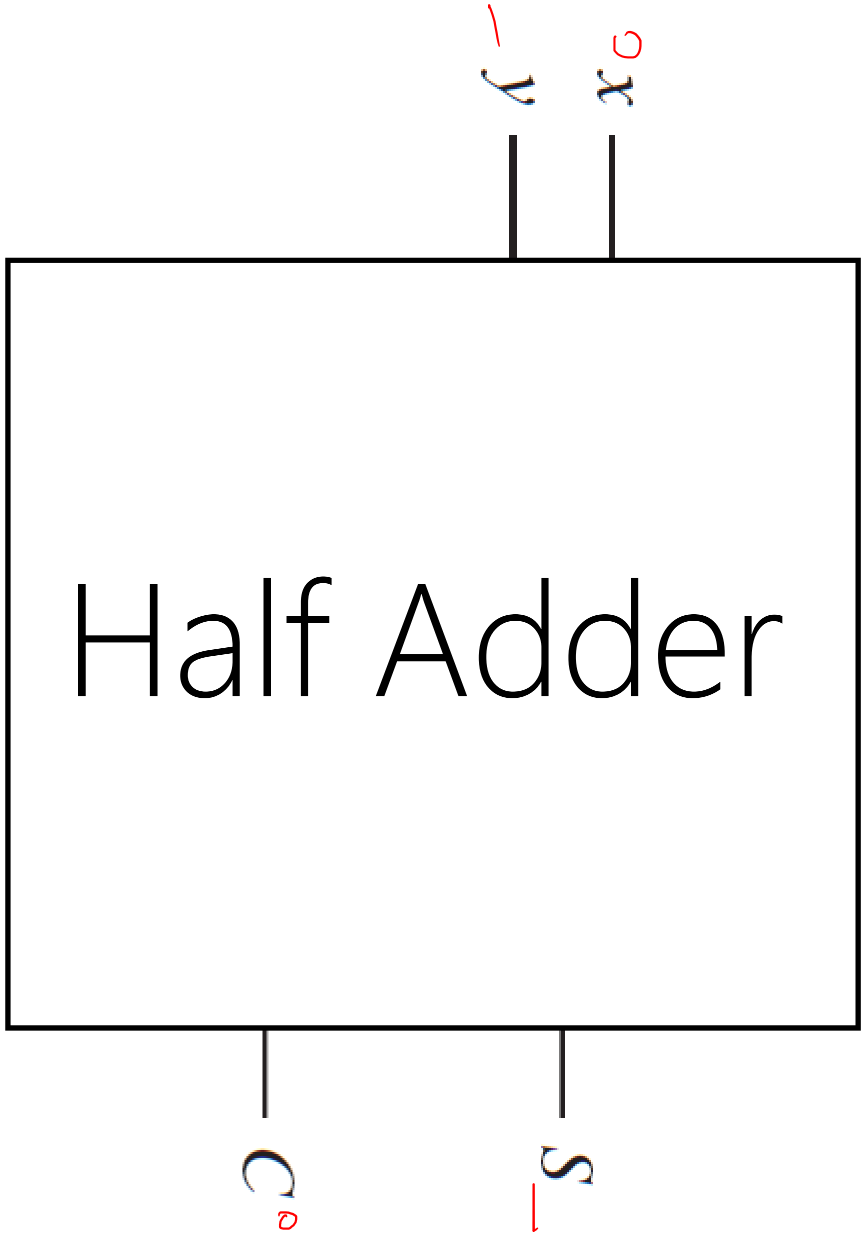


$$S = x \oplus y$$
$$C = xy$$

Half Adder: Just 2 bits: X+Y

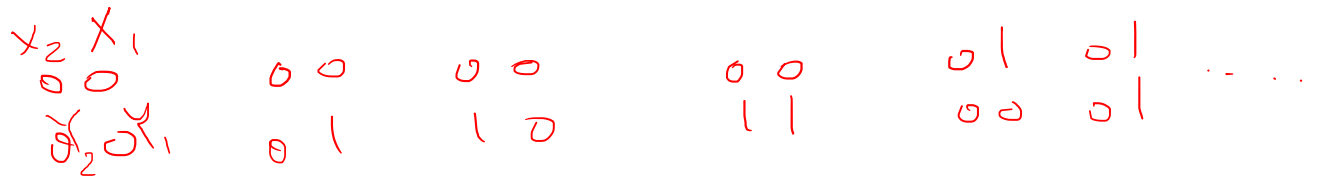


y
x
C
S


y
x
Half Adder
C
S

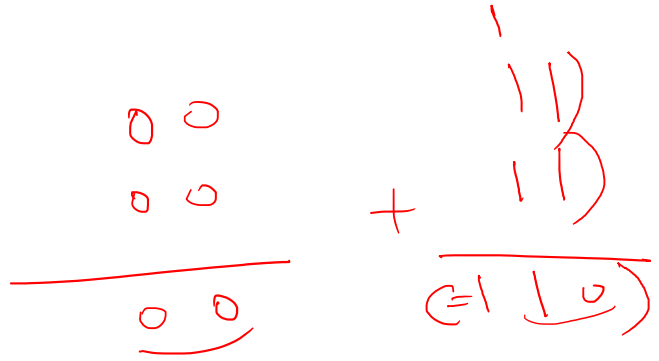
0100 + 0101

Design a logic circuit that adds two binary numbers!



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

Input binary variables: $X = X_2 X_1$ and $Y = Y_2 Y_1$



Range of outputs?

16

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

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

Range of outputs?
Carry, S_2, S_1

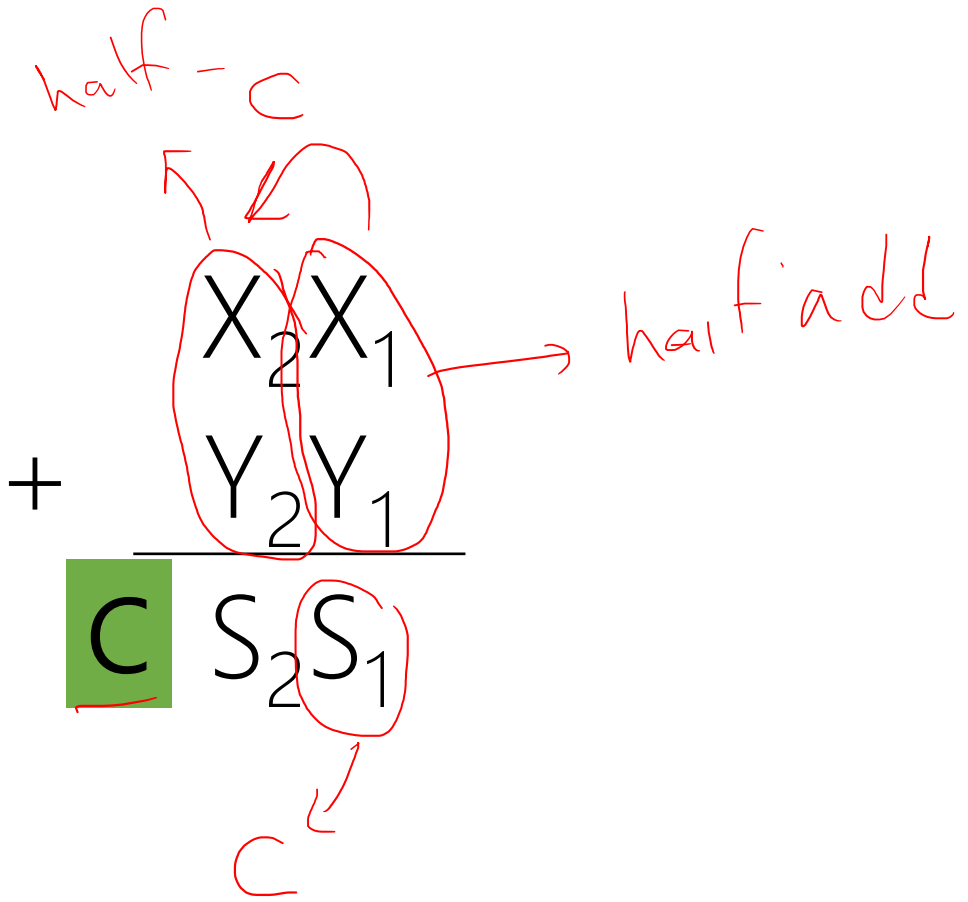
Y_2	Y_1	X_2	X_1	$C(Y_1,Y_2,X_2,X_1)$	$S_2(Y_1,Y_2,X_2,X_1)$	$S_1(Y_1,Y_2,X_2,X_1)$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1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

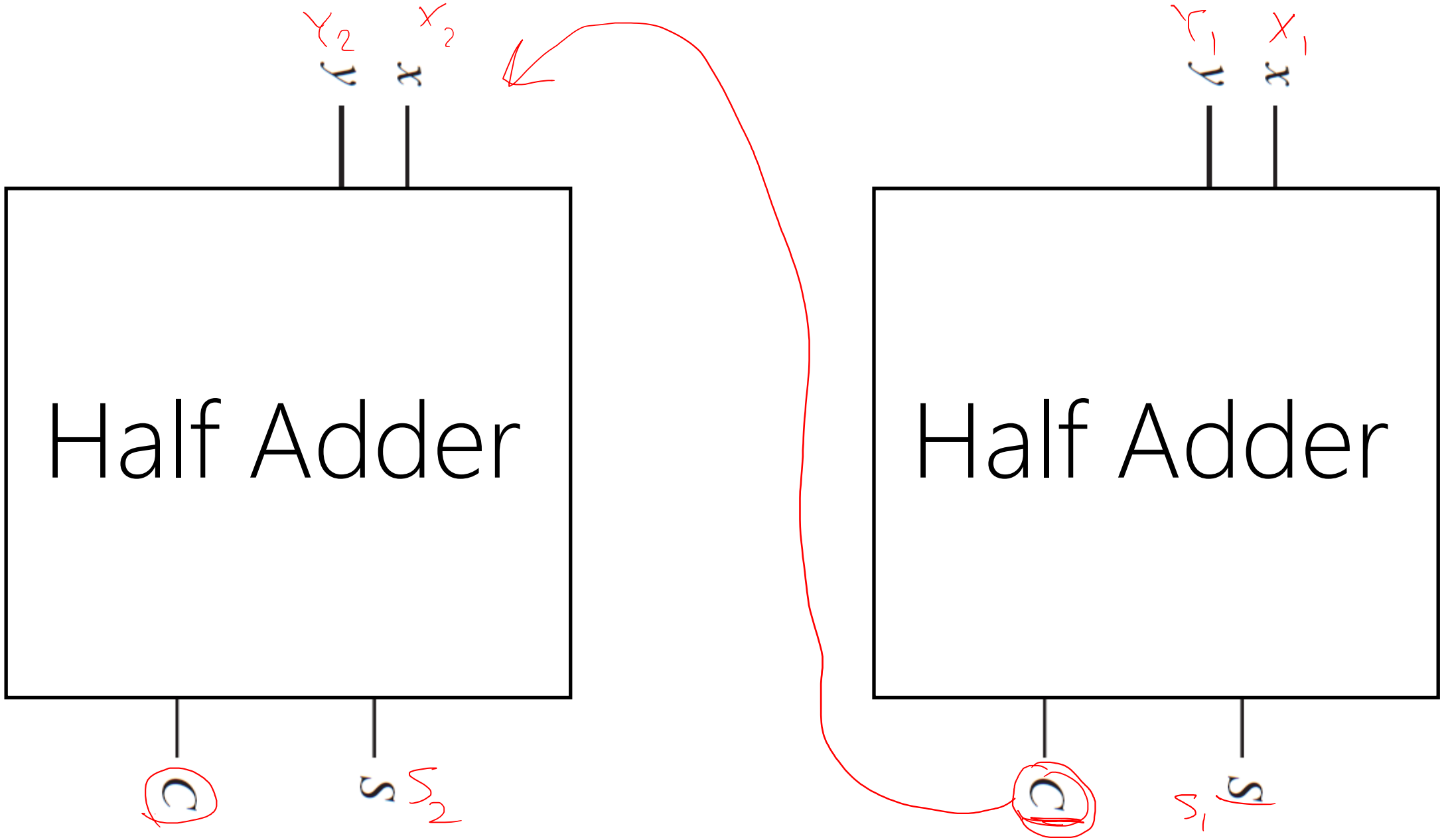
Y_2	Y_1	X_2	X_1	$C(Y_1,Y_2,X_2,X_1)$	$S_2(Y_1,Y_2,X_2,X_1)$	$S_1(Y_1,Y_2,X_2,X_1)$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	0	1	0	0	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1
1	1	1	1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

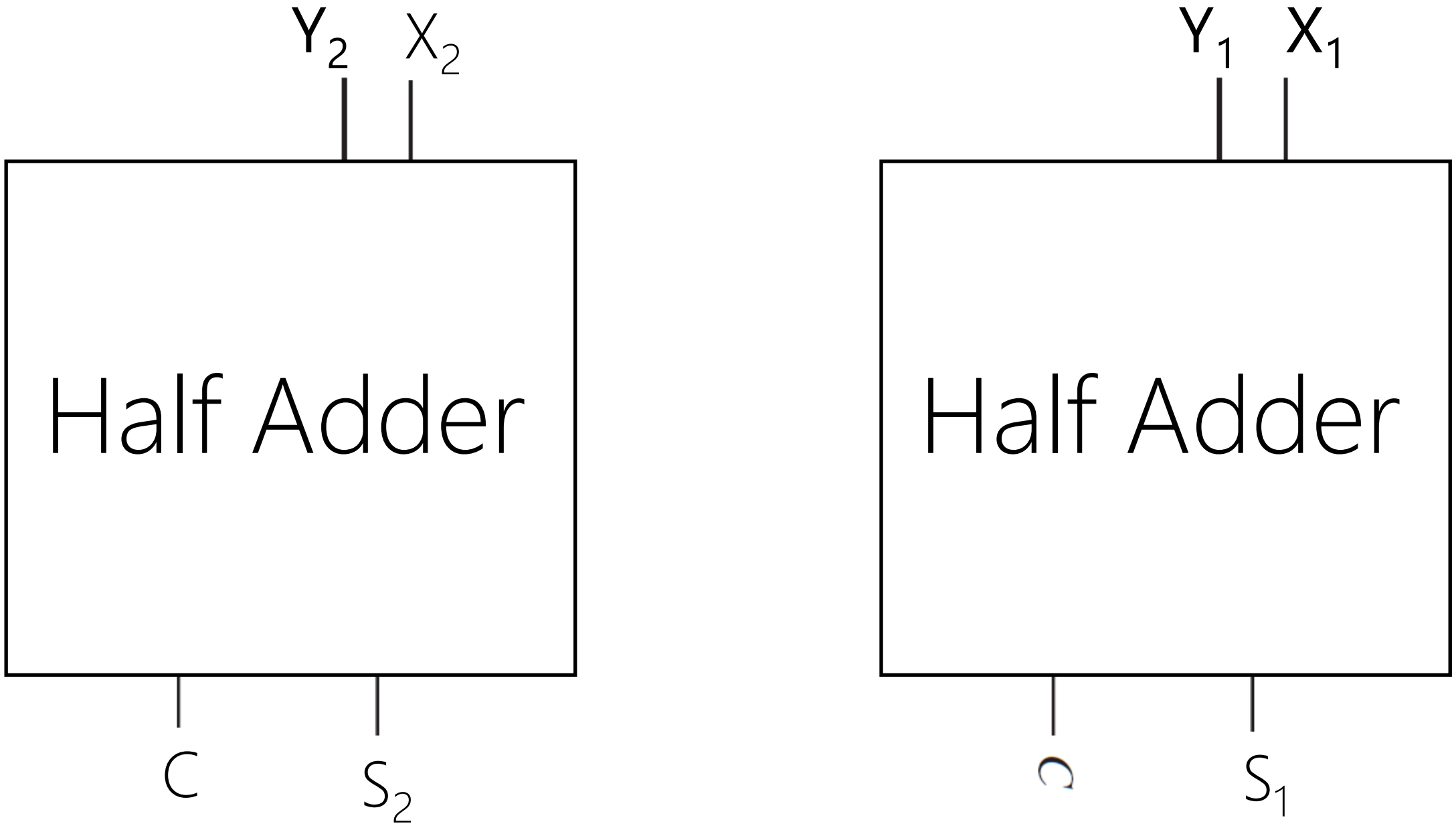
Wait a sec!

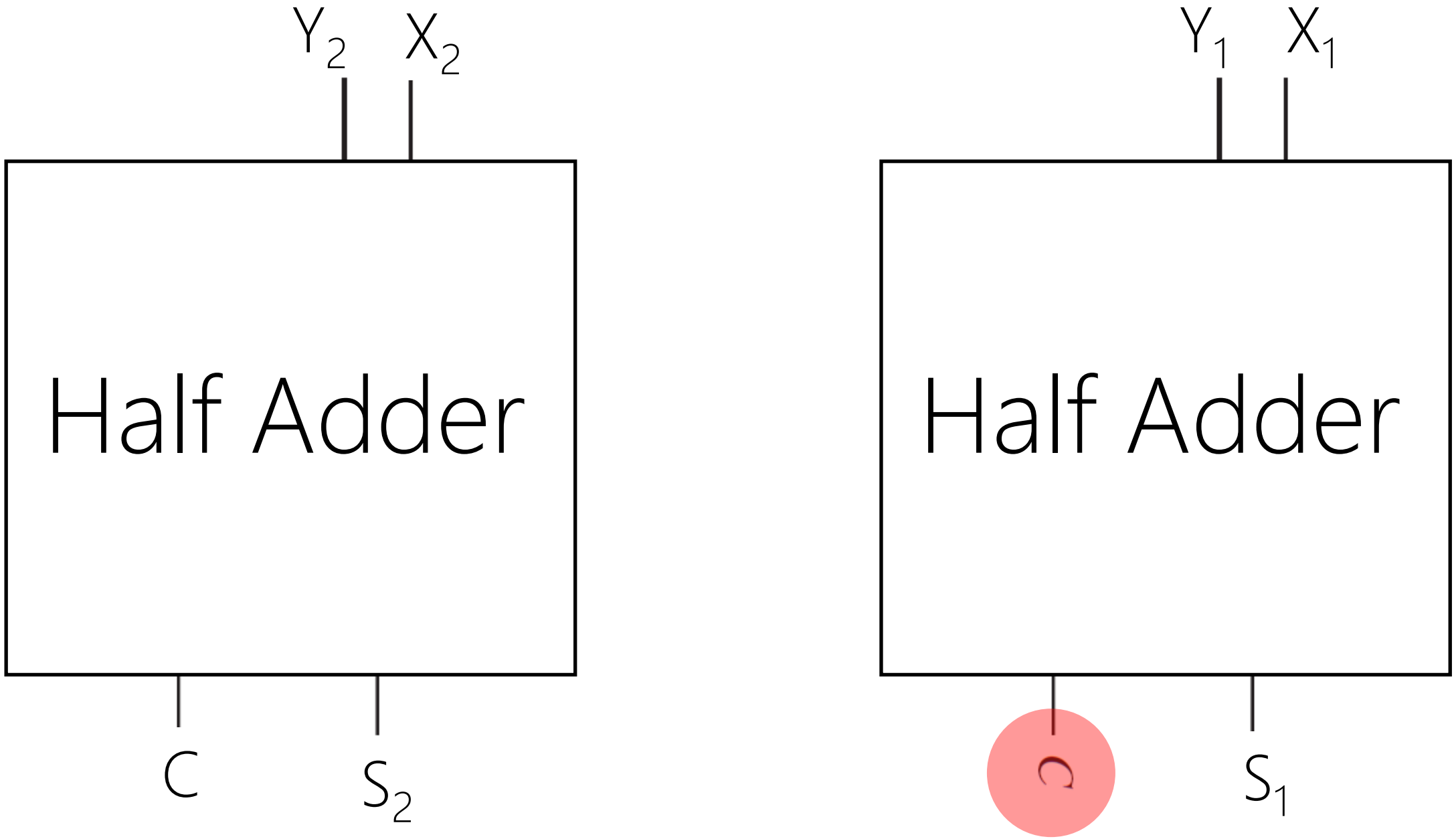
Can we re-use the half adder?

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


half - c
X_2X_1
+ Y_2Y_1
C S_2S_1
half add
C

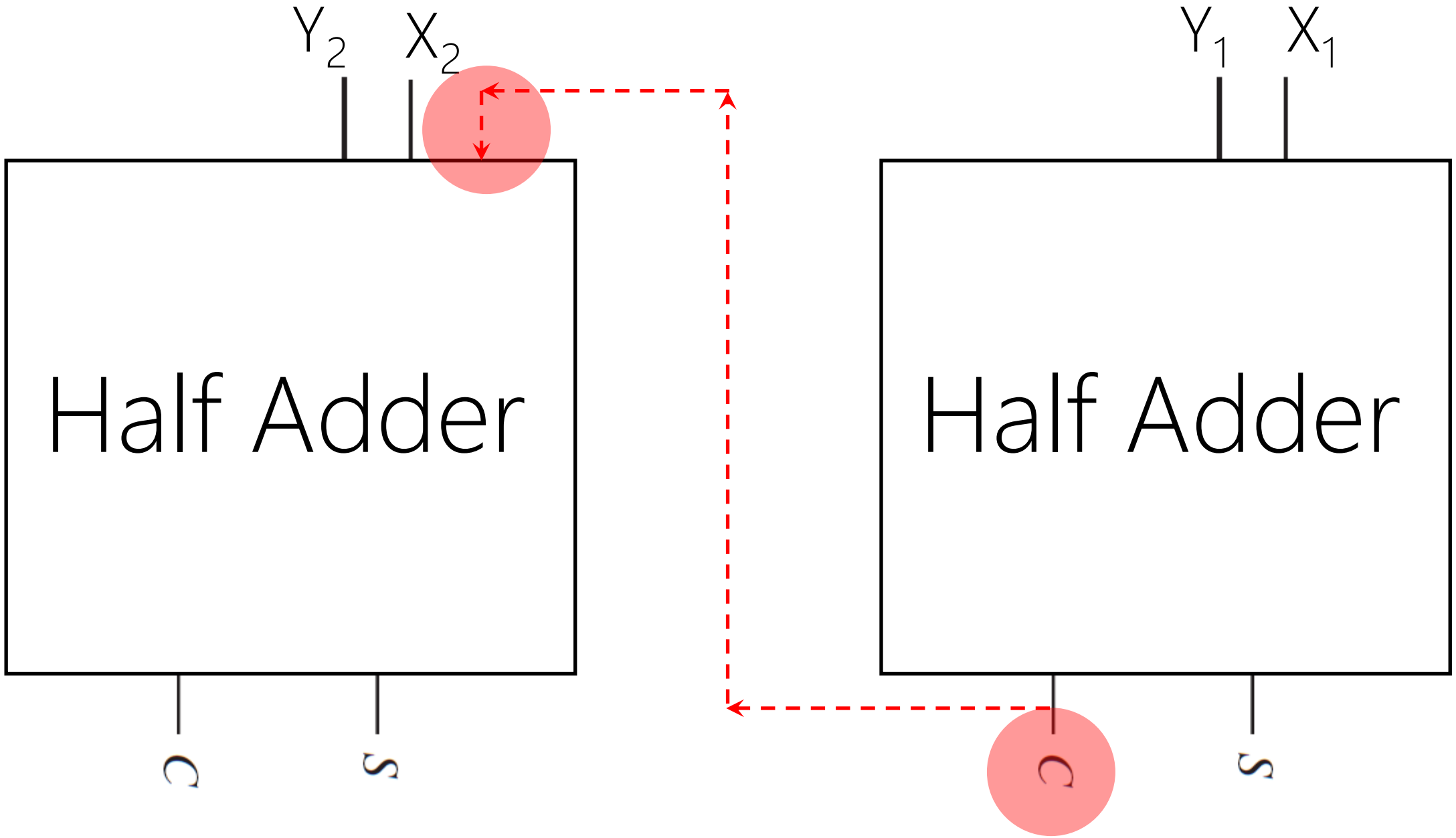

Y2
X2
y
x
Half Adder
C
S
S2
Y1
X1
y
x
Half Adder
C
S
S1

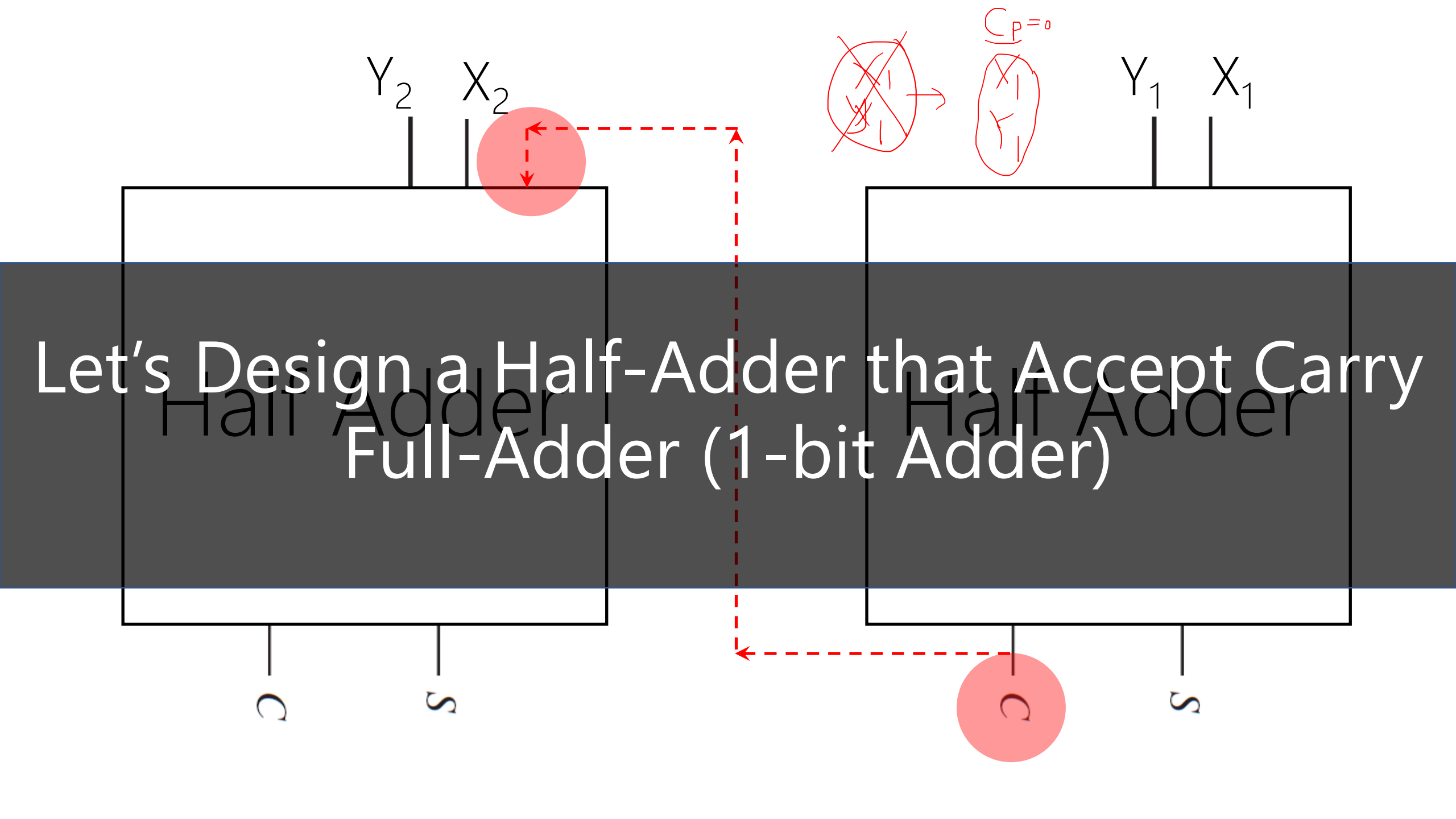

Y_2
X_2
Half Adder
C
S_2
Y_1
X_1
Half Adder
S_1


Y_2
X_2
Half Adder
C
S_2
Y_1
X_1
Half Adder
S_1

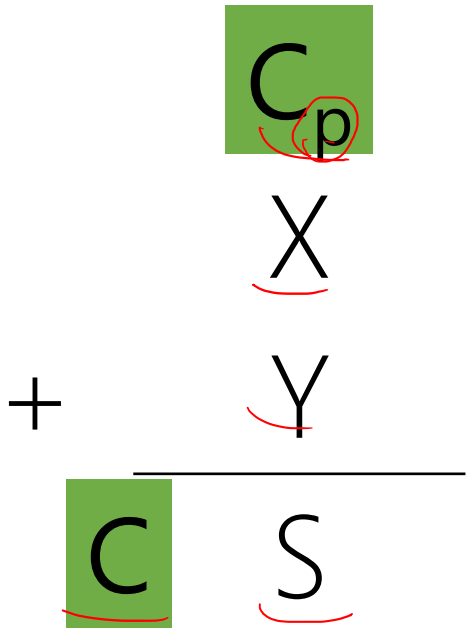
C=1

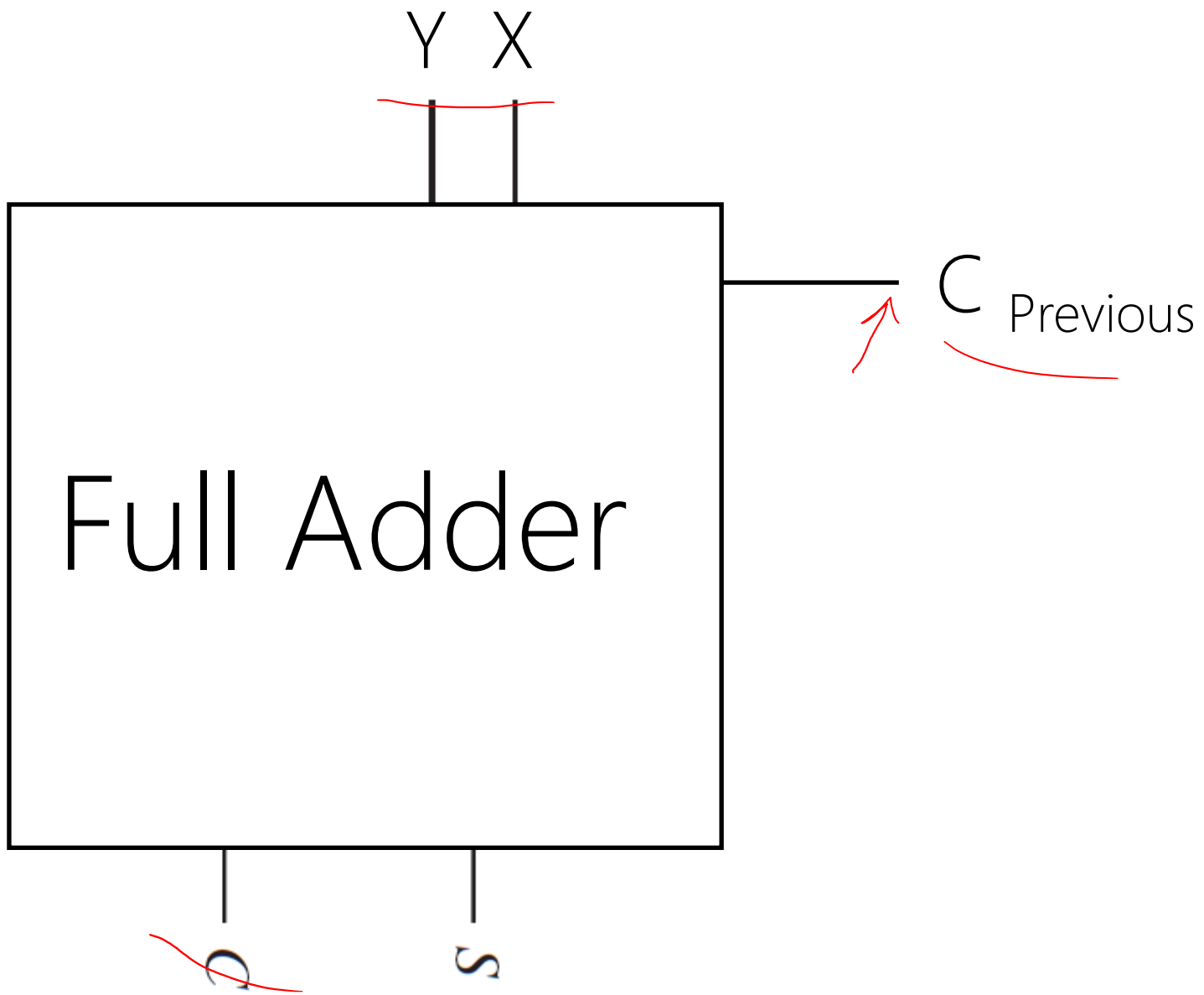
```
    0 1
+   0 1
-------
C=0 1 0
```


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



Let's Design a Half-Adder that Accept Carry
Full-Adder (1-bit Adder)

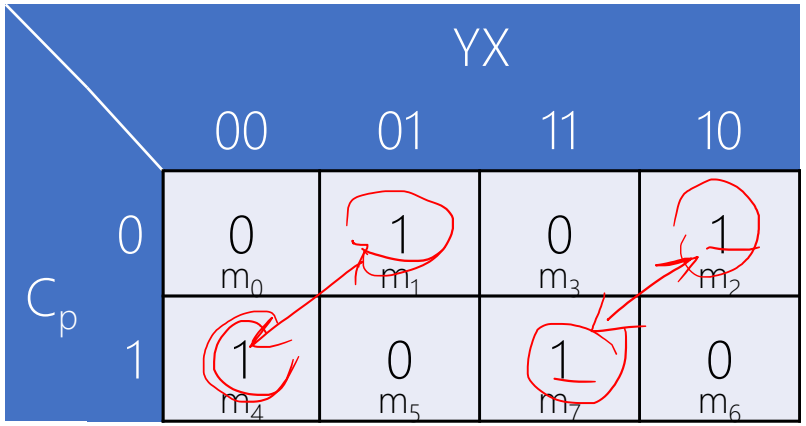

C_p
X
+ Y
C S


Y X
Full Adder
C Previous
C
S

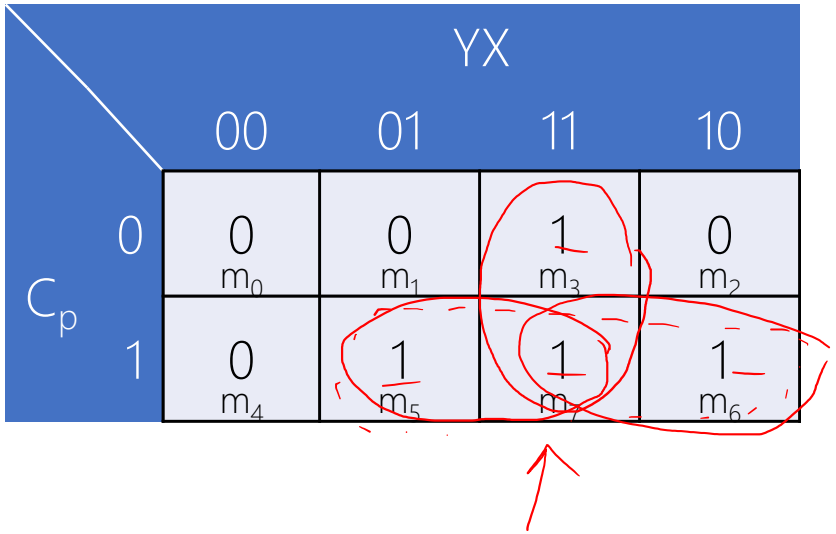
Design a logic circuit that adds two binary digits (bit) and a carry bit.

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

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



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

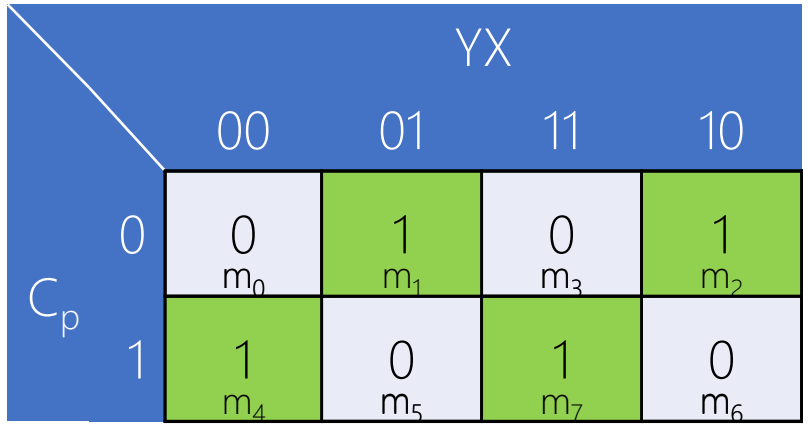


$$S=\sum m(1,2,4,7)$$

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

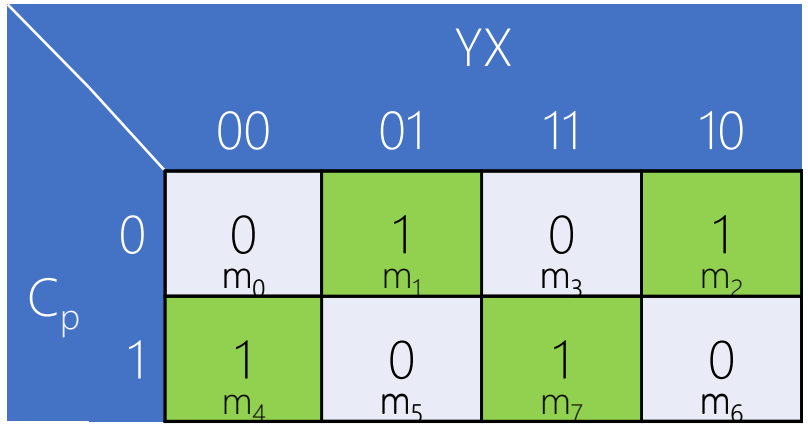
$$S=C'_pY'X+C'_pYX'+C_pY'X'+C_pYX$$

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



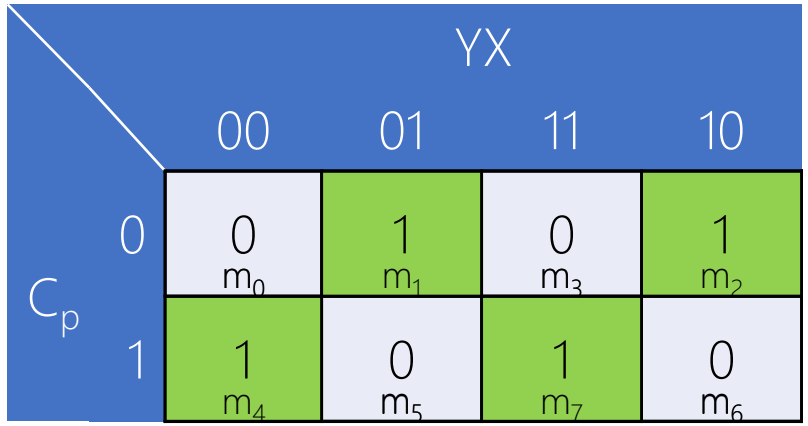
$$S=C'_pY'X+C'_pYX'+C_pY'X'+C_pYX$$
$$=C'_p(Y'X+YX')+C_p(Y'X'+YX)$$

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



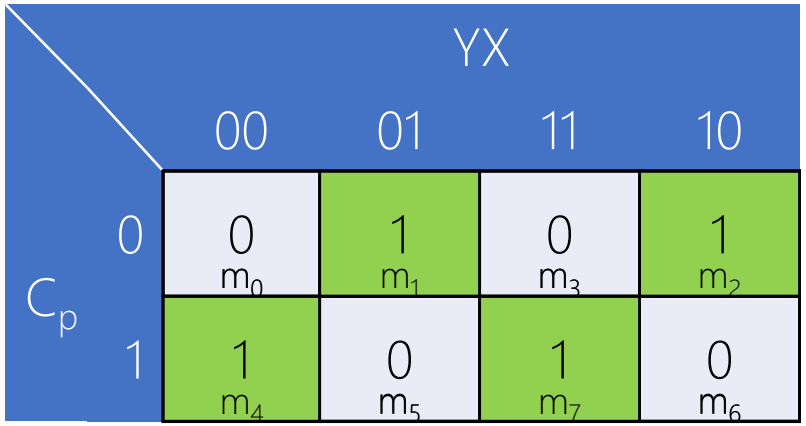
$$S=C'_pY'X+C'_pYX'+C_pY'X'+C_pYX$$
$$=C'_p(Y'X+YX')+C_p(Y'X'+YX)$$
$$=C'_p(X\oplus Y)+C_p(Y'X'+YX)$$

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



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

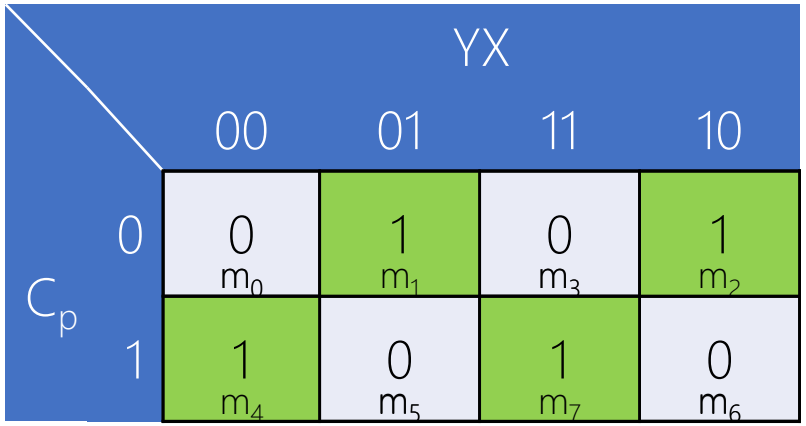
$$S=\sum m(1,2,4,7)$$



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

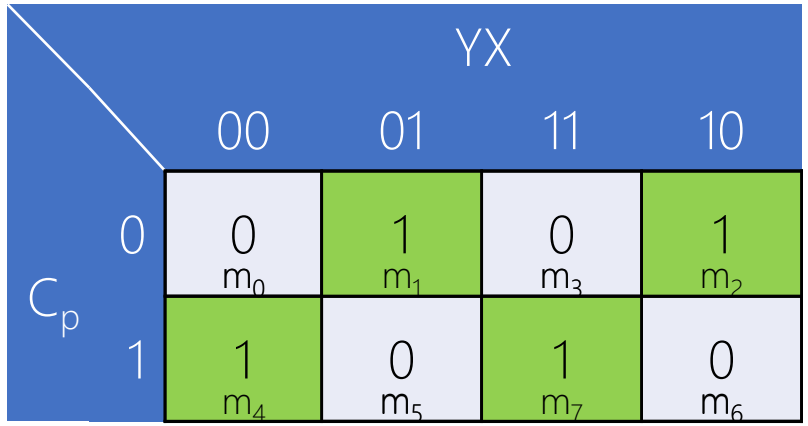
$$(X\oplus Y)' = (Y'X+YX')'$$
$$= (Y'X)'(YX')'$$
$$= (Y+X')(Y'+X)$$
$$= YY'+YX+X'Y'+X'X$$
$$= 0+YX+X'Y'+0$$
$$= YX+X'Y'$$
$$= Y\odot X$$

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



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

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

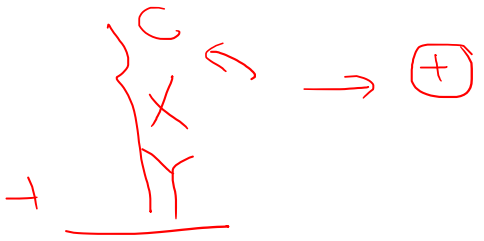


$$\begin{aligned}
S&=C'_pY'X+C'_pYX'+C_pY'X'+C_pYX\\
&=C'_p(Y'X+YX')+C_p(Y'X'+YX)\\
&=C'_p(X\oplus Y)+C_p(Y'X'+YX)\\
&=C'_p(X\oplus Y)+C_p(X\odot Y)\\
&=C'_p(X\oplus Y)+C_p(X\oplus Y)'\\
&=C'_p\alpha+C_p\alpha'\\
&=C_p\oplus\alpha
\end{aligned}$$

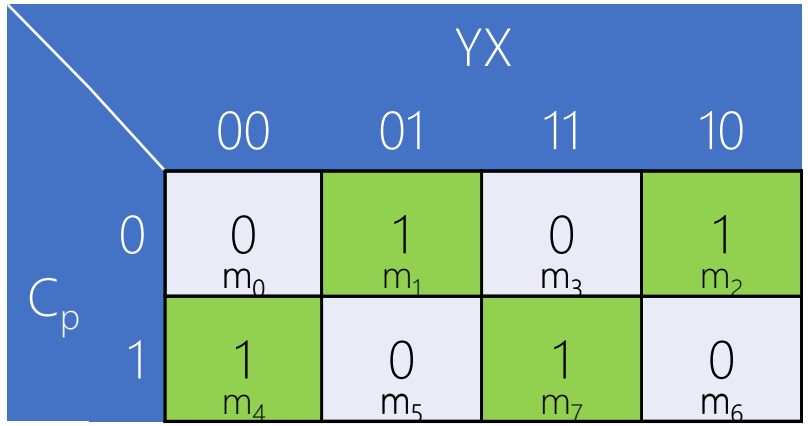
$$S=\sum m(1,2,4,7)$$

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

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

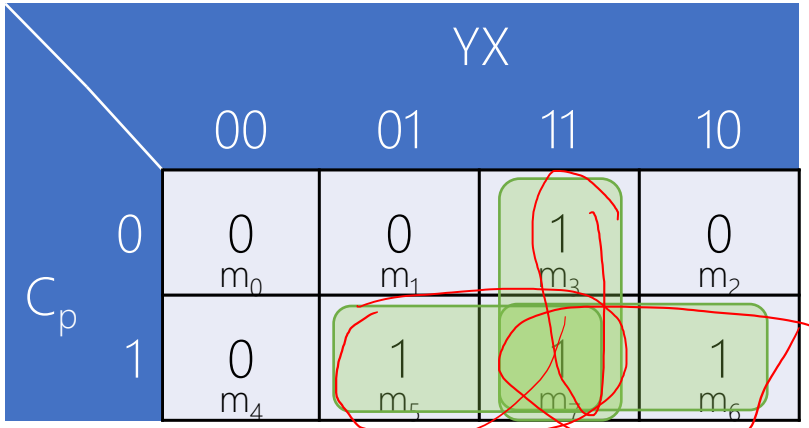


$$S = \sum m(1,2,4,7)$$
$$= C_p \oplus (X \oplus Y)$$

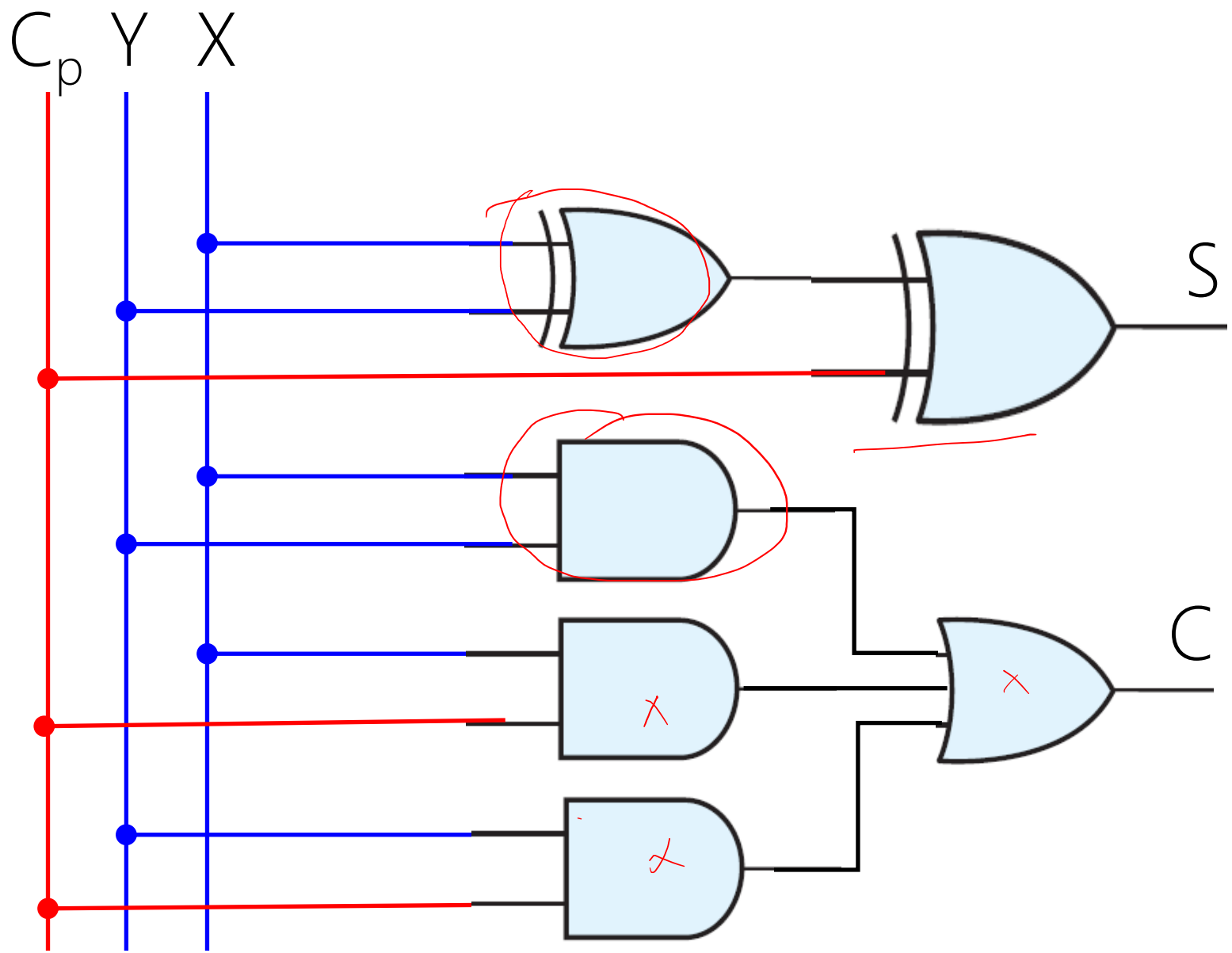


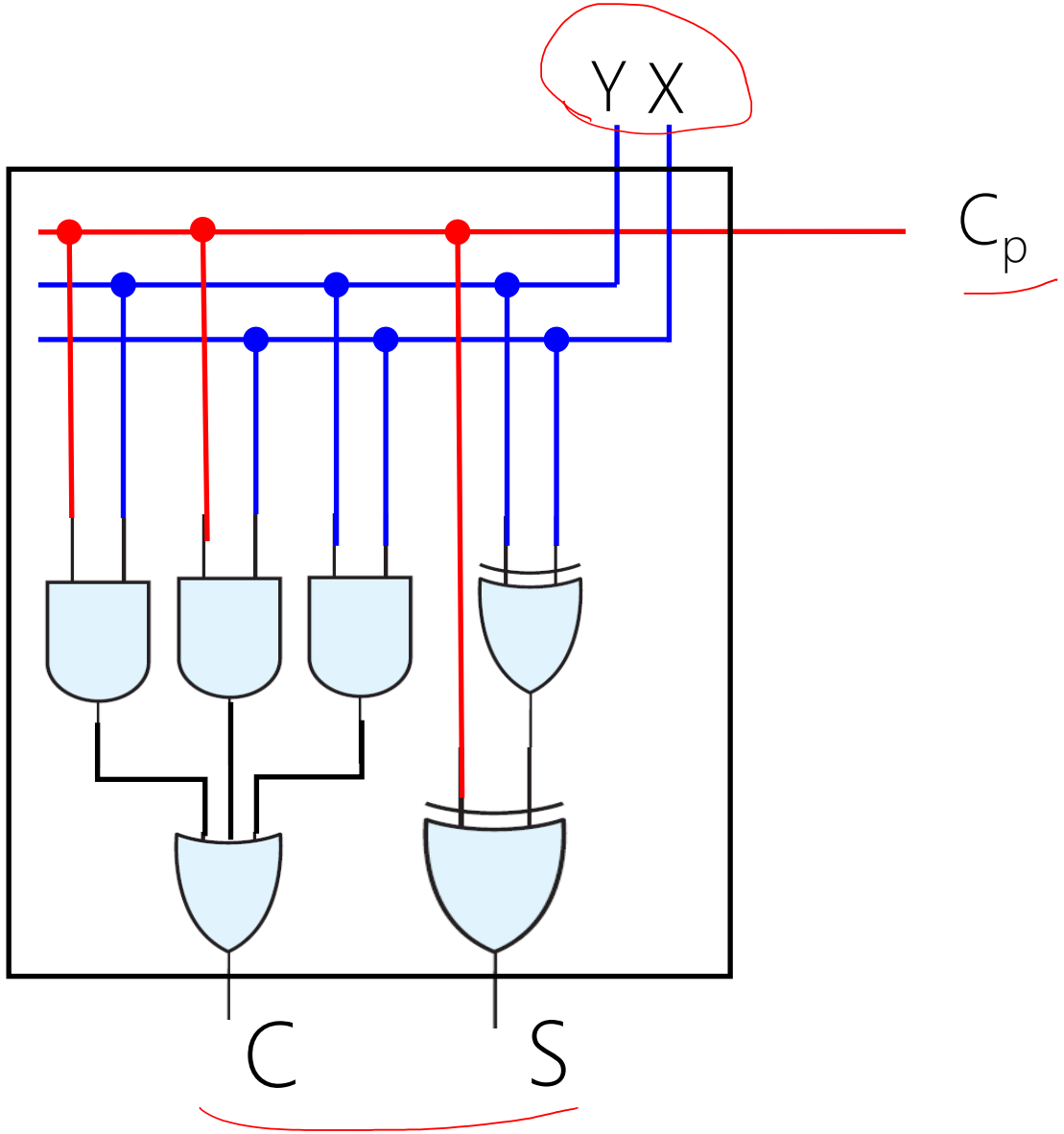
$\oplus$ is associative, we can drop ( ). But let's keep them!

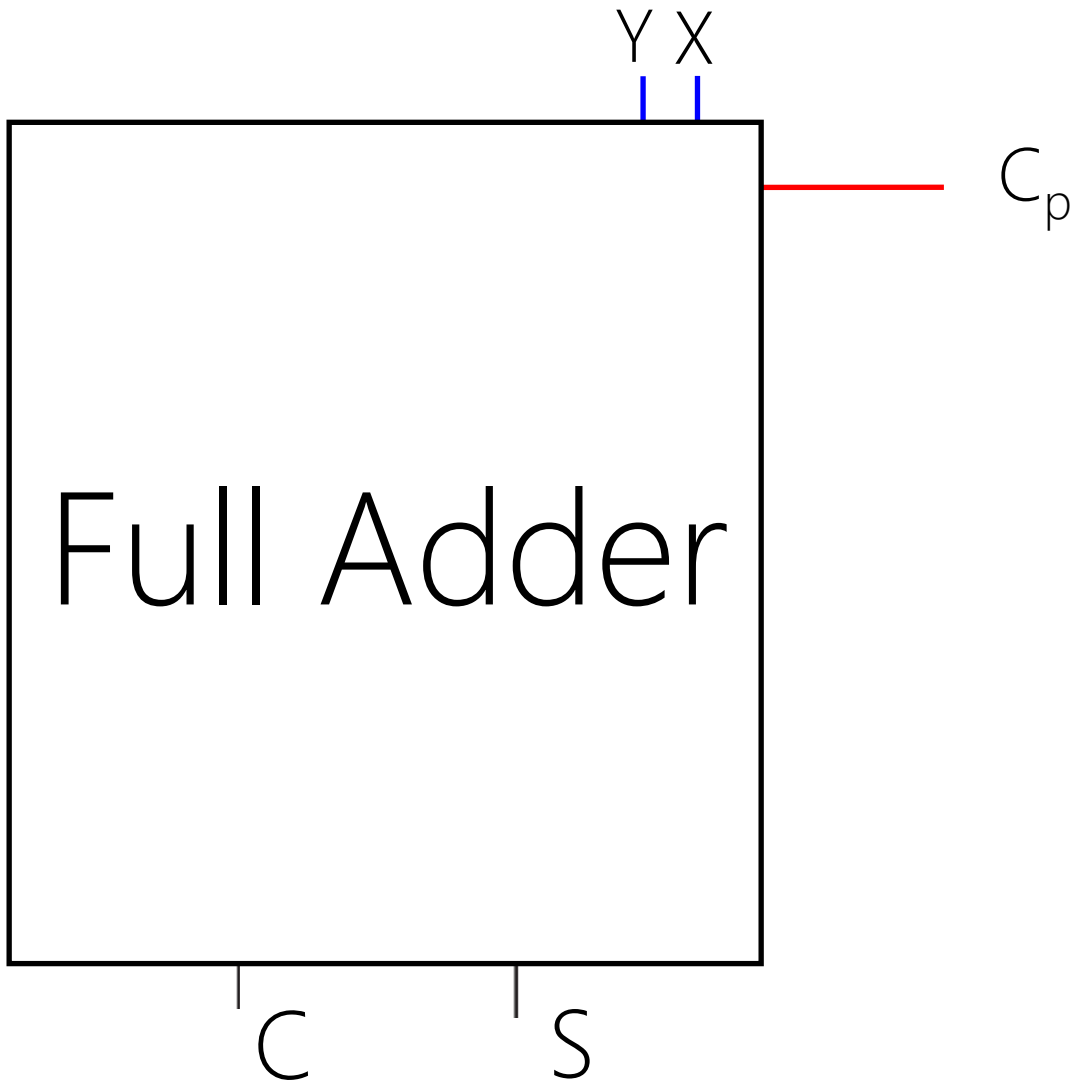
$$C = \sum m(3,5,6,7)$$

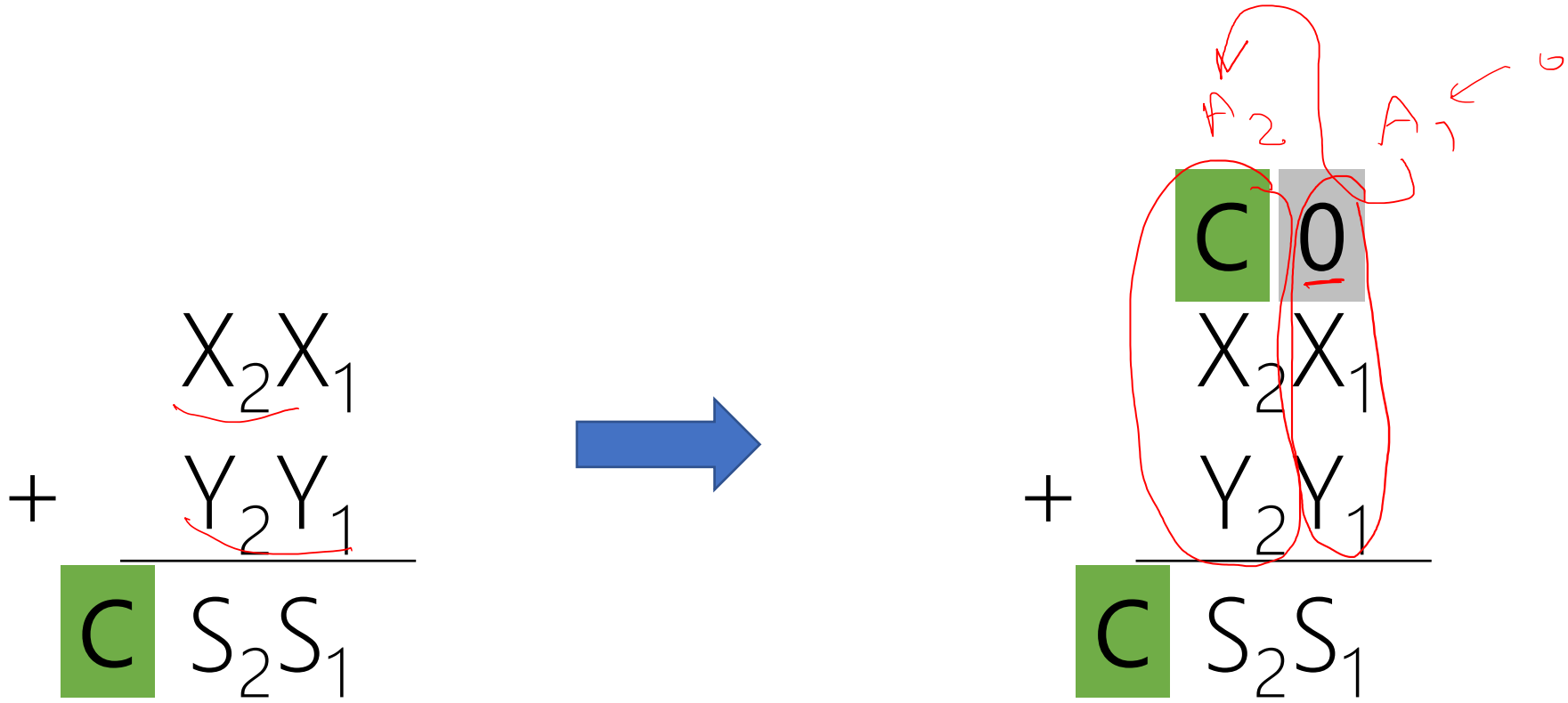


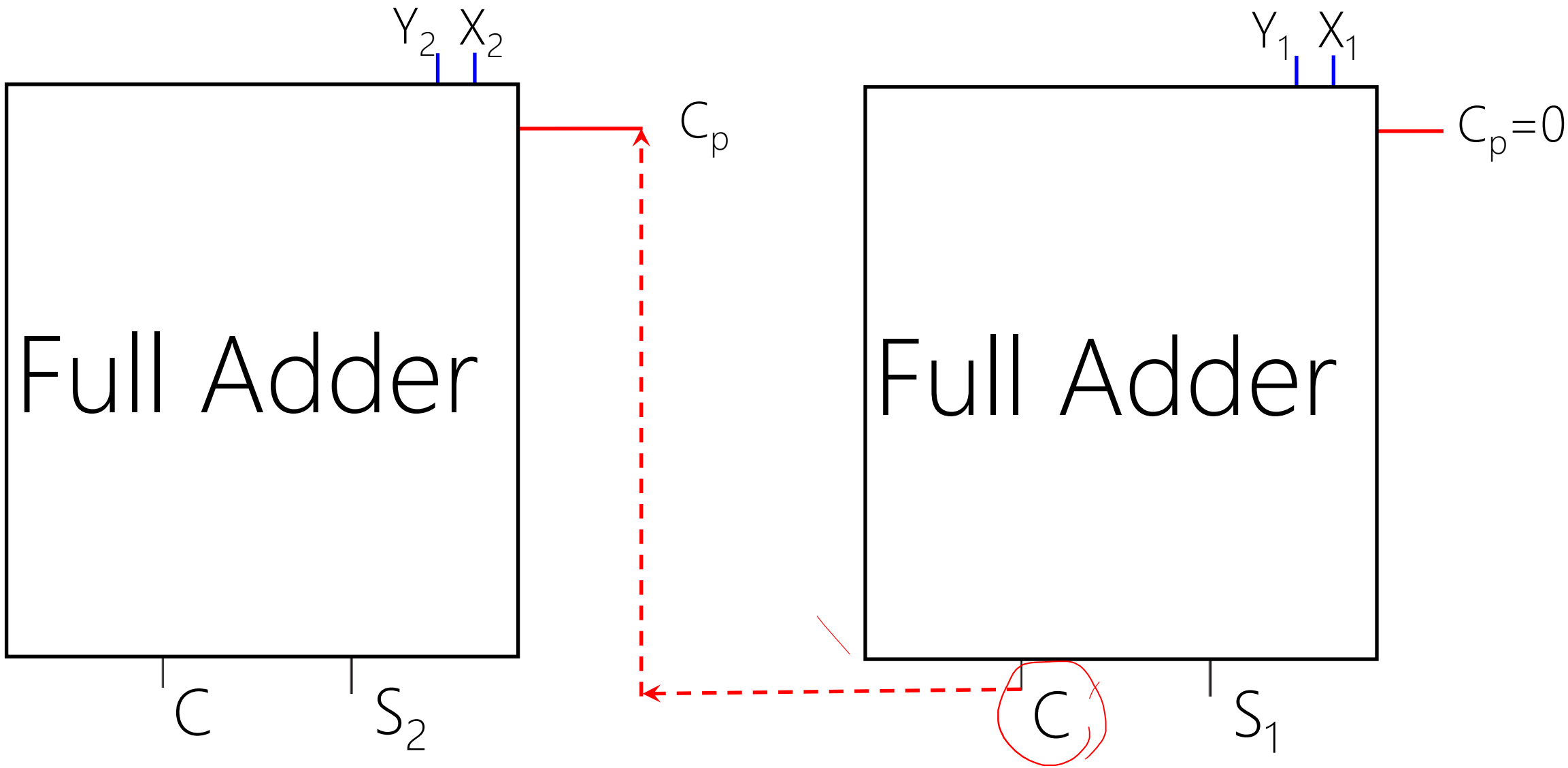
$$= YX + C_pX + C_pY$$


C_p Y X
S
C

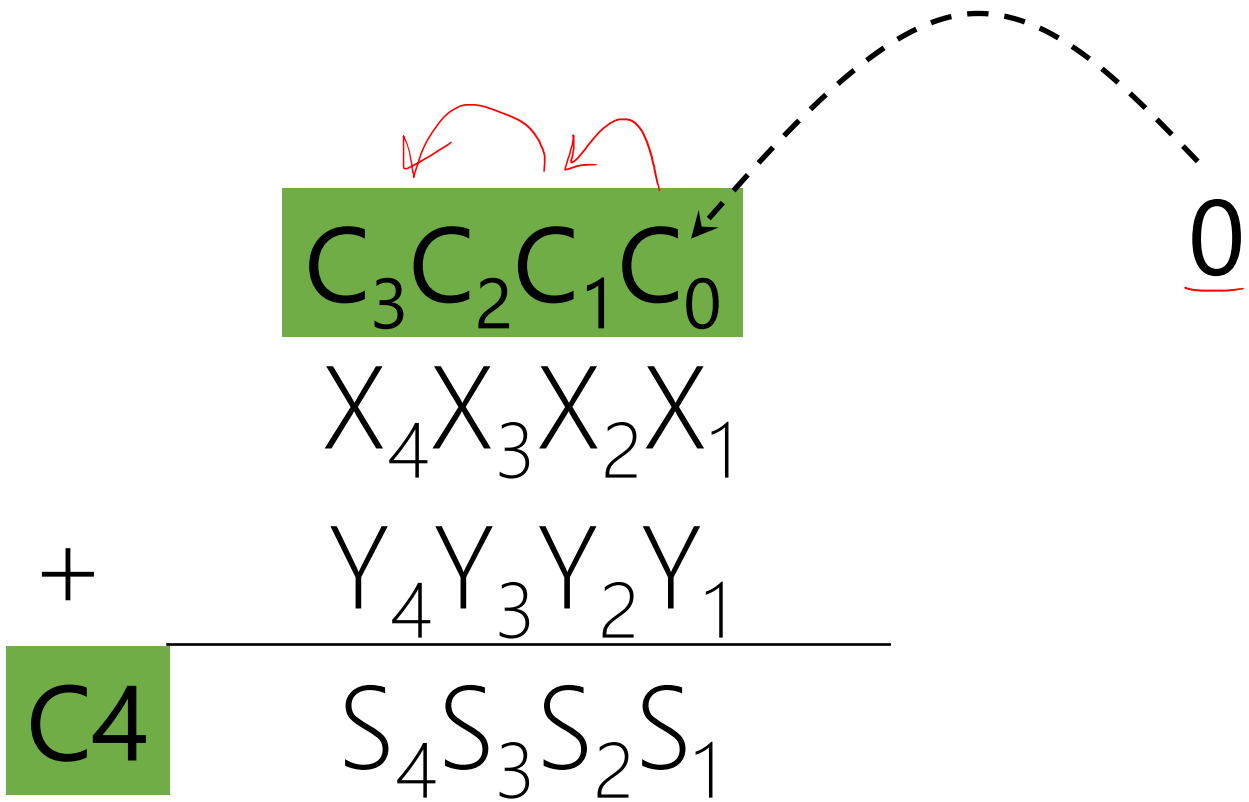

Y X
C_p
C
S

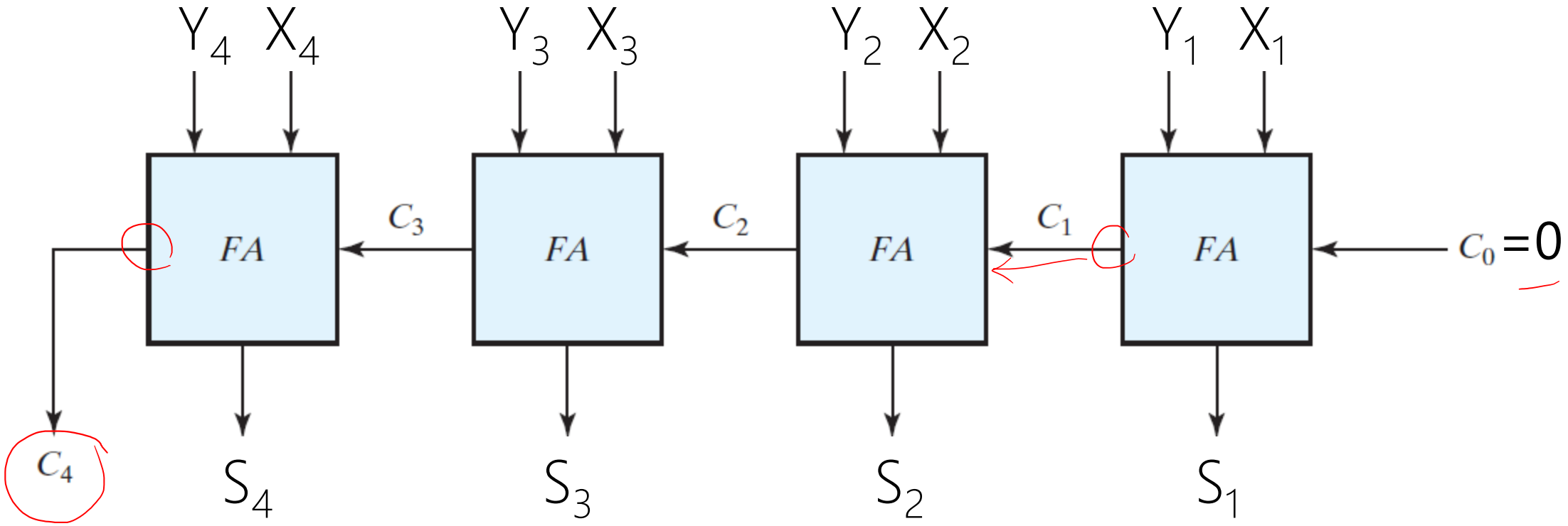

Y X
C_p
Full Adder
C
S

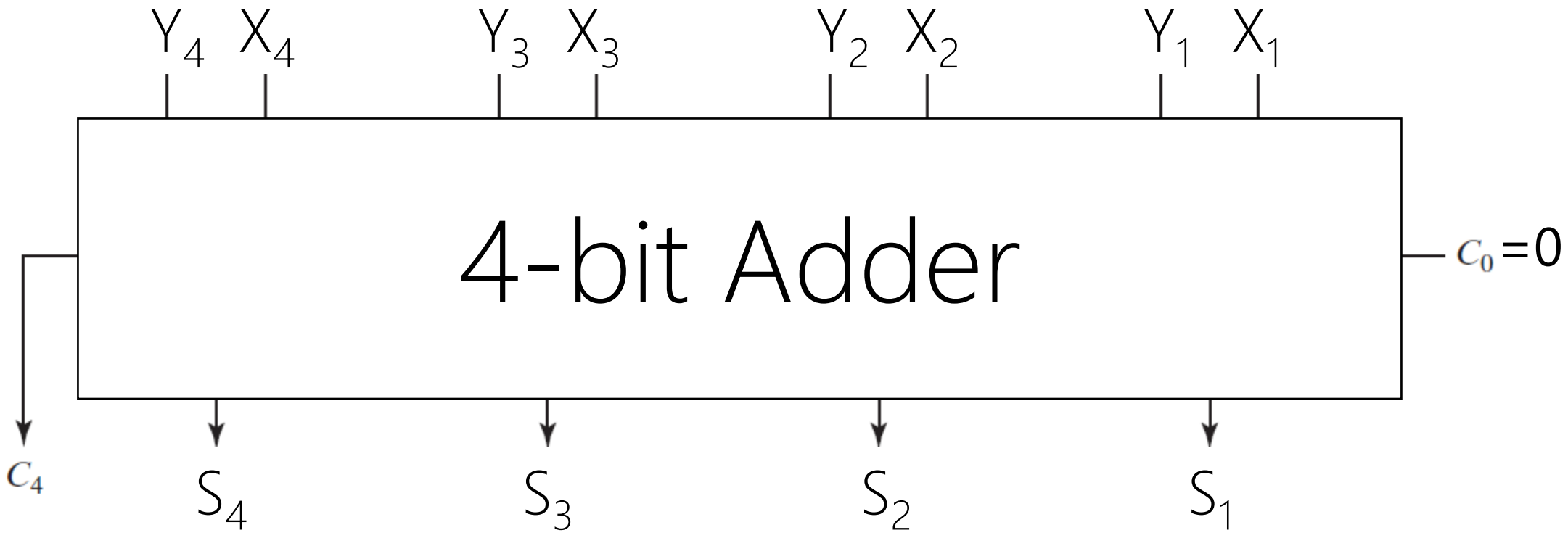

X_2X_1
$+ \quad Y_2Y_1$
$C \quad S_2S_1$
$C \quad 0$
X_2X_1
$+ \quad Y_2Y_1$
$C \quad S_2S_1$

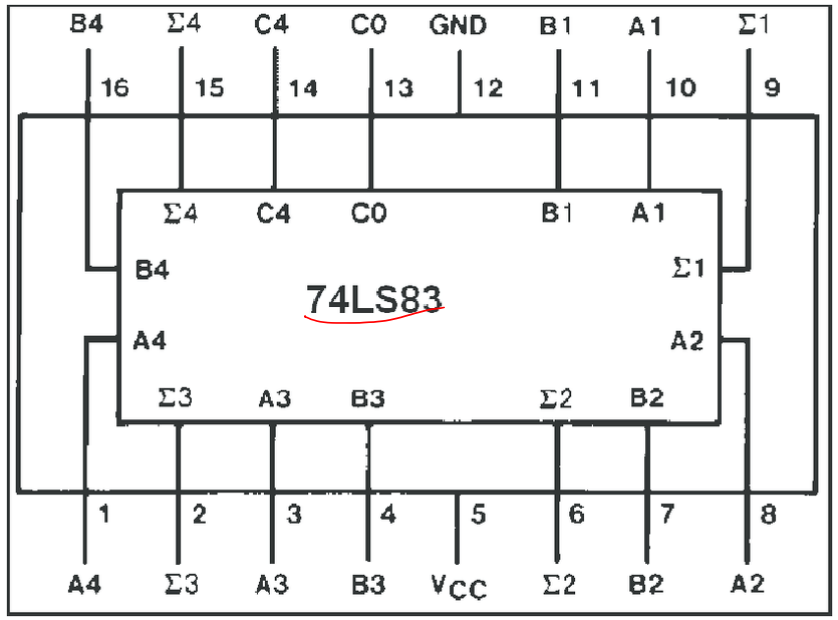

Y2 X2
Full Adder
Cp
C
S2
Y1 X1
Full Adder
Cp=0
C
S1

Design a logic circuit that adds two binary numbers!

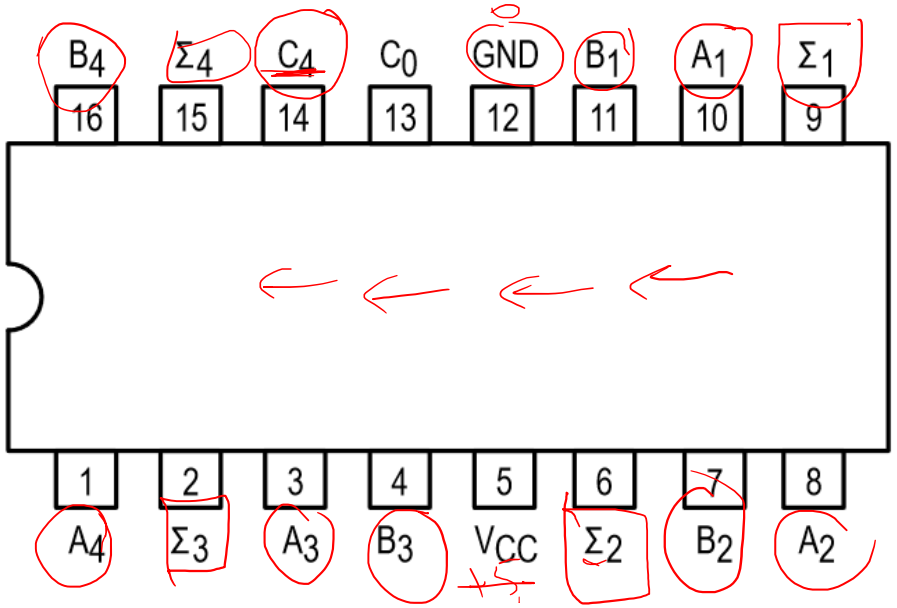

$C_3C_2C_1C_0$
0
$X_4X_3X_2X_1$
+
$Y_4Y_3Y_2Y_1$
C4
$S_4S_3S_2S_1$


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


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



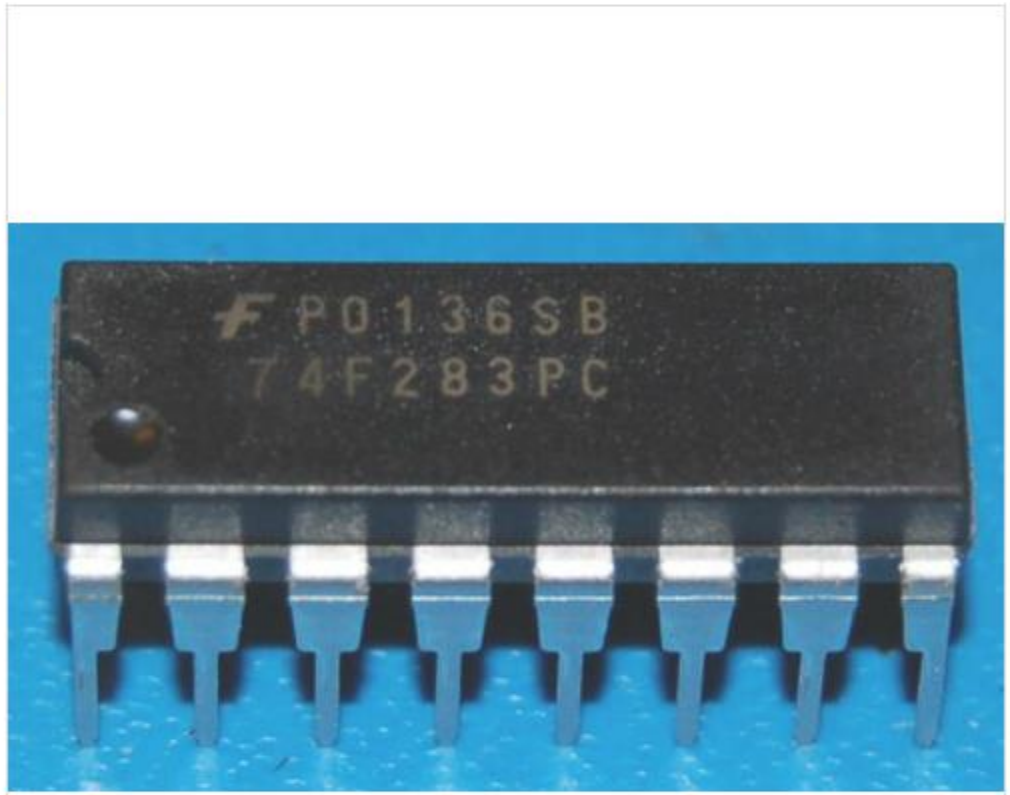
74LS83 pinout



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)

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