



20.02.20 07:39

CSCI 150

Introduction to Digital and Computer System Design

Lecture 3: Combinational Logic Design VI



Jetic Gū
2020 Winter Semester (S1)

Overview

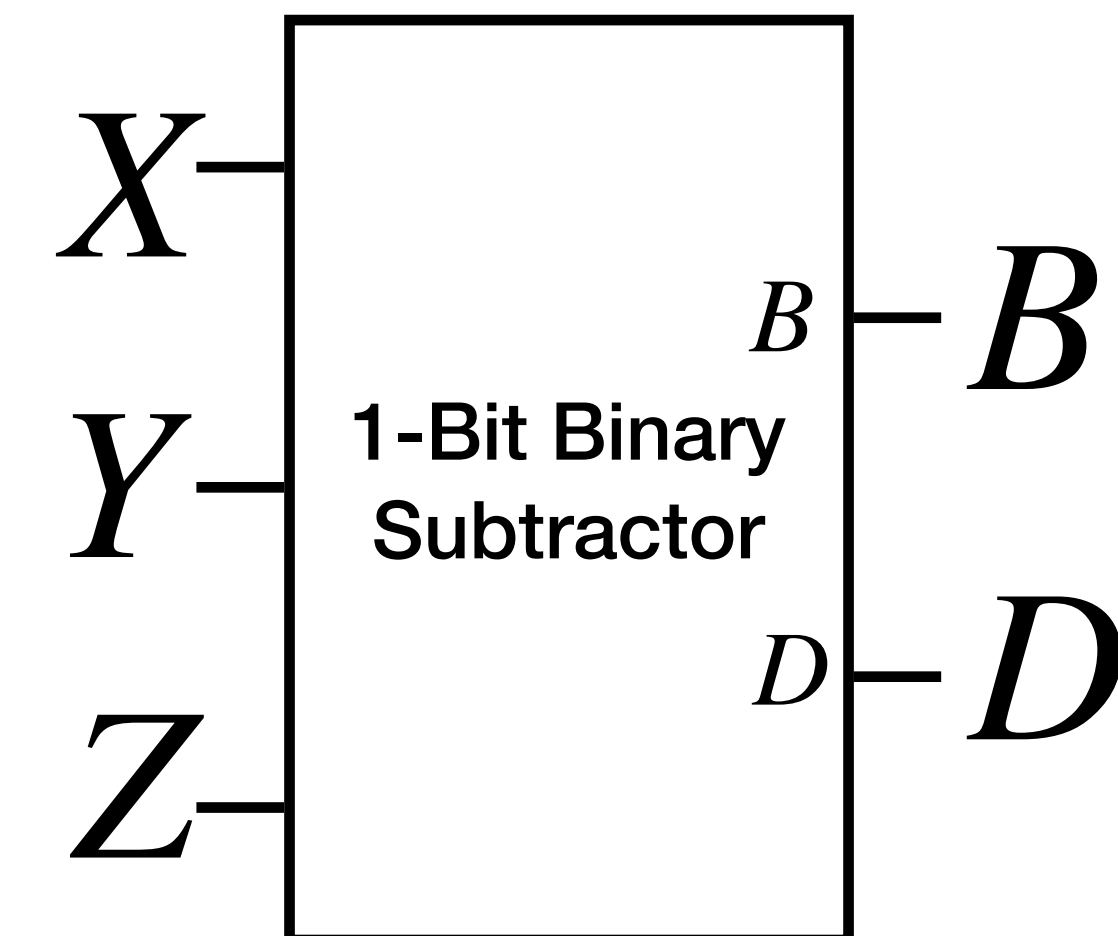
- Focus: Arithmetic Functional Blocks
- Architecture: Combinatory Logical Circuits
- Textbook v4: Ch4 4.3, 4.4, 4.7; v5: Ch2 2.9, Ch3 3.10, 3.11
- Core Ideas:
 1. Subtraction II
 2. Subtraction III
 3. VHDL

Unsigned Binary Subtraction I

Review

Unsigned 1-bit Binary Subtraction

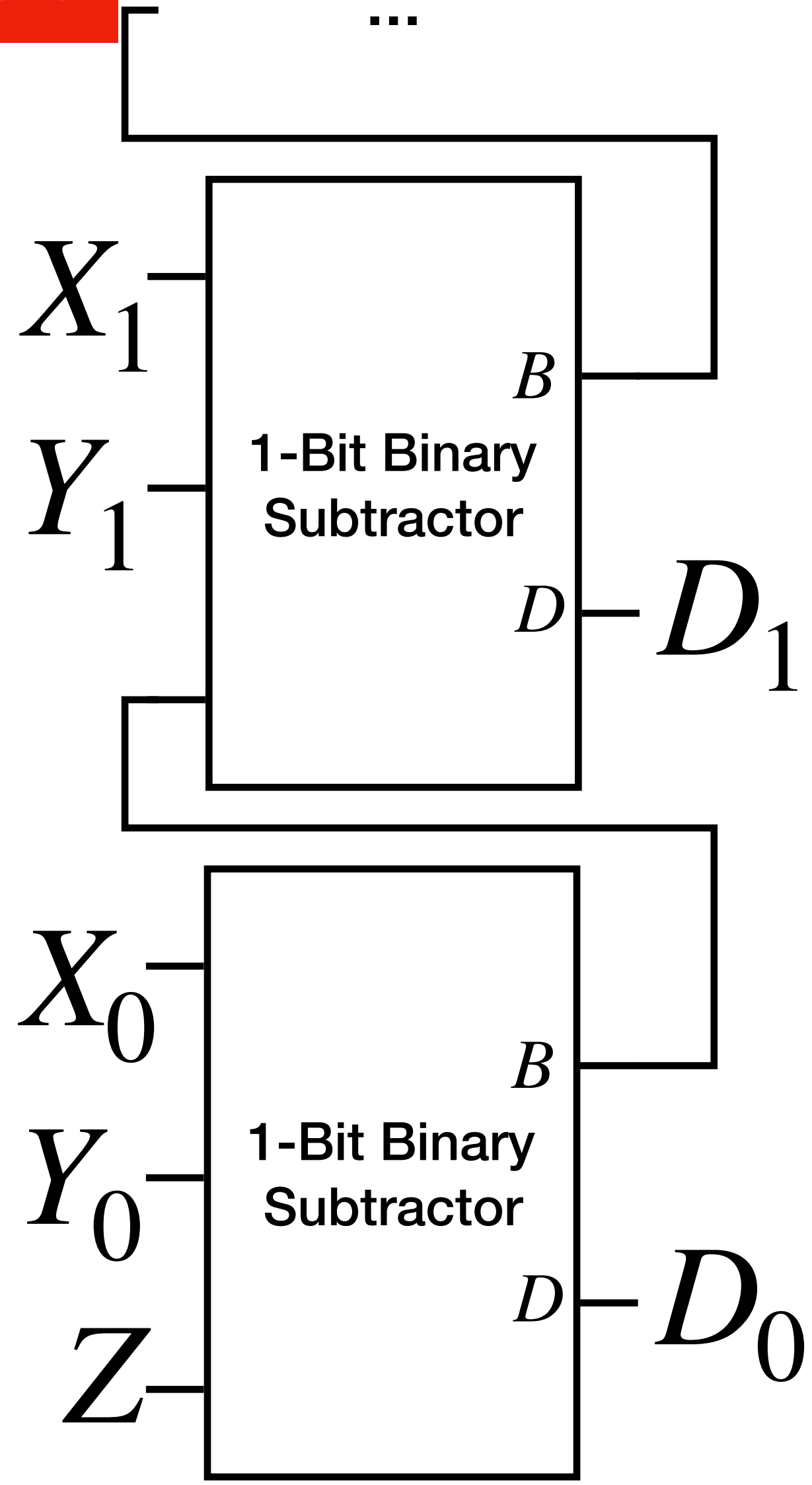
- Implementation using 3-to-8 Decoder
 - $B = \Sigma m(1,2,3,7)$
 - $D = \Sigma m(1,2,4,7)$



Unsigned Binary Subtraction

Technology

- 1 bit Unsigned Subtractor



Borrows	<i>B</i>	<i>Z</i>
	0	0
Minuend $X_{0:n-1}$	0011	0
Subtrahend $Y_{0:n-1}$	1011	1
	—	—
Difference $D_{0:n-1}$	0001	1

Input
Output

Unsigned Binary Subtraction II

$X - Y$ when $Y > X$

What we have so far

- Binary Adder
- Binary Subtractor ($X - Y, X \geq Y$)

Unsigned Binary Subtraction

$$X > Y, F = X - Y$$

- We learned to perform subtraction, by subtracting the smaller number from the greater number
- What if it's the opposite? i.e. $X < Y, F = X - Y$?

Unsigned Binary Subtraction

Borrows	
Minuend	10011
Subtrahend	—00110
Difference	

Borrows	
Minuend	00110
Subtrahend	—10011
Difference	

Unsigned Binary Subtraction

Borrows	
Minuend	10011
Subtrahend	— 00110
Difference	

Borrows	
Minuend	00110
Subtrahend	— 10011
Difference	

Unsigned Binary Subtraction

$$\begin{array}{r} \text{Borrows} \quad 011000 \\ \text{Minuend} \quad 10011 \\ \text{Subtrahend} \quad -00110 \\ \hline \text{Difference} \quad 01101 \end{array}$$

$$\begin{array}{r} \text{Borrows} \quad 100110 \\ \text{Minuend} \quad 00110 \\ \text{Subtrahend} \quad -10011 \\ \hline \text{Difference} \quad 10011 \end{array}$$

Unsigned Binary Subtraction

Borrows	011000
Minuend	10011
Subtrahend	—00110
	—————
Difference	01101

This is correct

Borrows	100110
Minuend	00110
Subtrahend	—10011
	—————
Difference	10011

Unsigned Binary Subtraction

Borrows	011000
Minuend	10011
Subtrahend	—00110
	—————
Difference	01101

This is correct

Borrows	100110
Minuend	00110
Subtrahend	—10011
	—————
Difference	10011

This is incorrect

Unsigned Binary Subtraction

- Standard subtraction module works if the Minuend is bigger than the Subtrahend

Borrows	011000
Minuend	10011
Subtrahend	-00110
<hr/>	
Difference	01101

This is correct

Borrows	100110
Minuend	00110
Subtrahend	-10011
<hr/>	
Difference	10011

This is incorrect

Unsigned Binary Subtraction

- Standard subtraction module works if the Minuend is bigger than the Subtrahend

Borrows	100110
Minuend X	00110
Subtrahend Y	- 10011

Difference D	10011
Corrected D'	- 01101

(2's compliment)

This is incorrect

Concept

Unsigned Binary Subtraction

- Standard subtraction module works if the Minuend is bigger than the Subtrahend

- Incorrect output D
 $= 2^n + X - Y$

(2's compliment)

Borrows	100110
Minuend X	00110
Subtrahend Y	- 10011

Difference D	10011
Corrected D'	- 01101

This is incorrect

Unsigned Binary Subtraction

- Standard subtraction module works if the Minuend is bigger than the Subtrahend

- Incorrect output D
 $= 2^n + X - Y$

- Correct output D'
 $= -(Y - X)$
 $= -(2^n - D)$
 $= -(\bar{D} + 1)$ (2's compliment)

Borrows	100110
Minuend X	00110
Subtrahend Y	- 10011

Difference D	10011
Corrected D'	- 01101

This is incorrect

Concept

2s compliment

- Given binary unsigned integer of n bits D , its 2s compliment $2^n - D = \bar{D} + 1$
- Proof
 - Biggest number represented in n bit: $(11\dots1)_2 = 2^n - 1$
 - $2^n - D = [(11\dots1)_2 + 1] - D = (11\dots1)_2 - D + 1 = \bar{D} + 1$
- Implementation
 - Inversion and plus 1, easily doable as complementer

Subtraction

Subtraction

1. Compute $20-15=5$ using 6-bit binary

Subtraction

1. Compute $20-15=5$ using 6-bit binary
 - $20 = (010100)_2$, $15 = (001111)_2$

Subtraction

1. Compute $20 - 15 = 5$ using 6-bit binary
 - $20 = (010100)_2$, $15 = (001111)_2$

Borrows

$$\begin{array}{r} 0011110 \\ 010100 \\ -001111 \\ \hline 000101 \end{array}$$

Subtraction

1. Compute $20-15=5$ using 6-bit binary
 - $20 = (010100)_2$, $15 = (001111)_2$
2. Compute $15-20=-5$ using 6-bit binary and 2s complement

Borrows

$$\begin{array}{r} 0011110 \\ 010100 \\ -001111 \\ \hline 000101 \end{array}$$

Subtraction

1. Compute $20-15=5$ using 6-bit binary
 - $20 = (010100)_2$, $15 = (001111)_2$
2. Compute $15-20=-5$ using 6-bit binary and 2s complement

Borrows

$$\begin{array}{r} 0011110 \\ 010100 \\ -001111 \\ \hline 000101 \end{array}$$

Subtraction

1. Compute $20-15=5$ using 6-bit binary
 - $20 = (010100)_2$, $15 = (001111)_2$
2. Compute $15-20=-5$ using 6-bit binary and 2s complement
 - Correction: $(111011)_2$ 2s complement: $(000101)_2$

Borrows

$$\begin{array}{r} 1\ 100000 \\ \ 001111 \\ -\ 010100 \\ \hline \ 111011 \\ \ 000101 \end{array}$$

Example

Subtraction

1. Compute $20-15=5$ using 6-bit binary
 - $20 = (010100)_2$, $15 = (001111)_2$
2. Compute $15-20=-5$ using 6-bit binary and 2s complement
 -

Borrows

$$\begin{array}{r} 1\ 100000 \\ 001111 \\ -010100 \\ \hline 111011 \end{array}$$

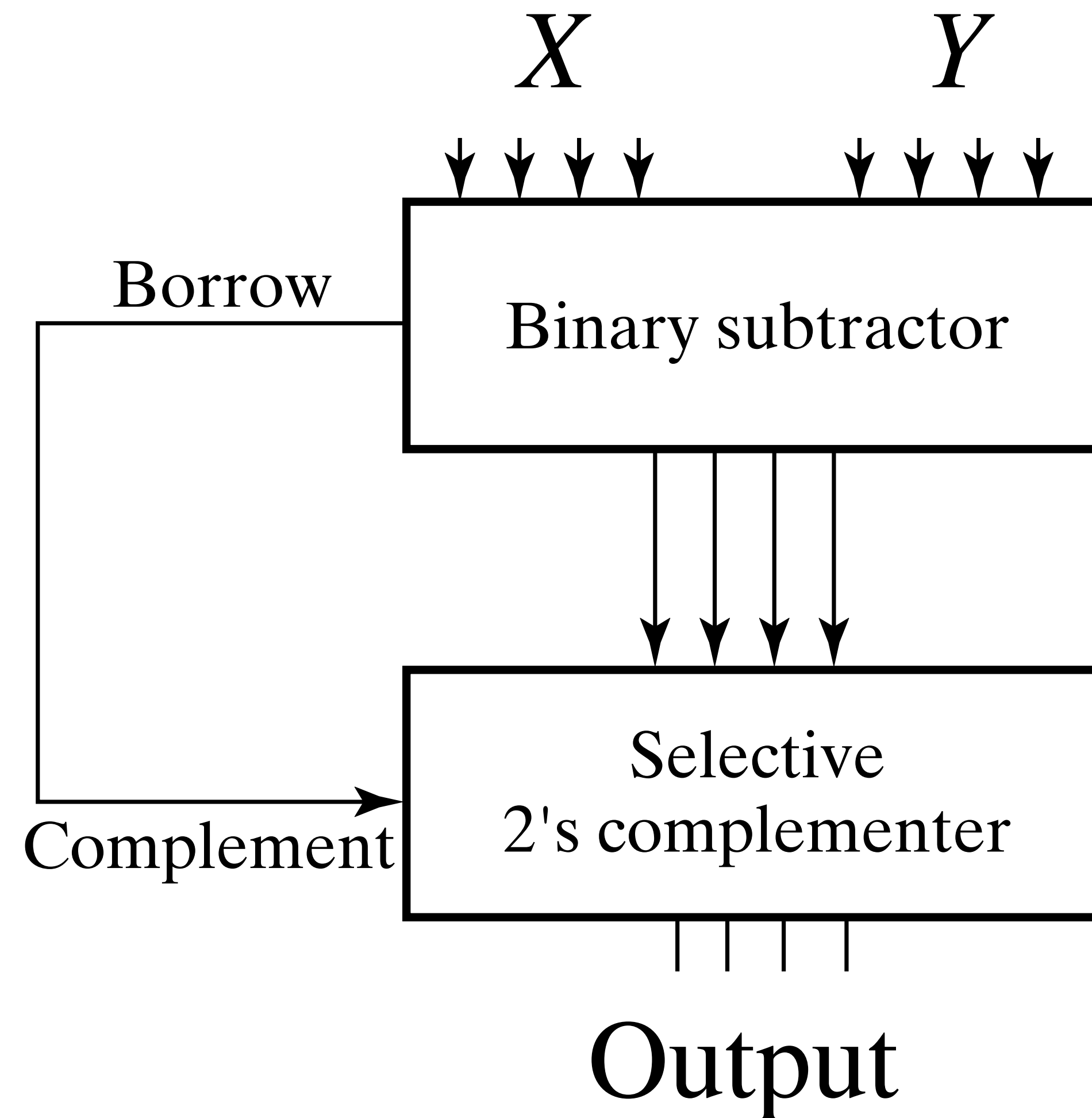
Subtraction

1. Compute $7-15=-8$ using 6-bit binary and 2s complement

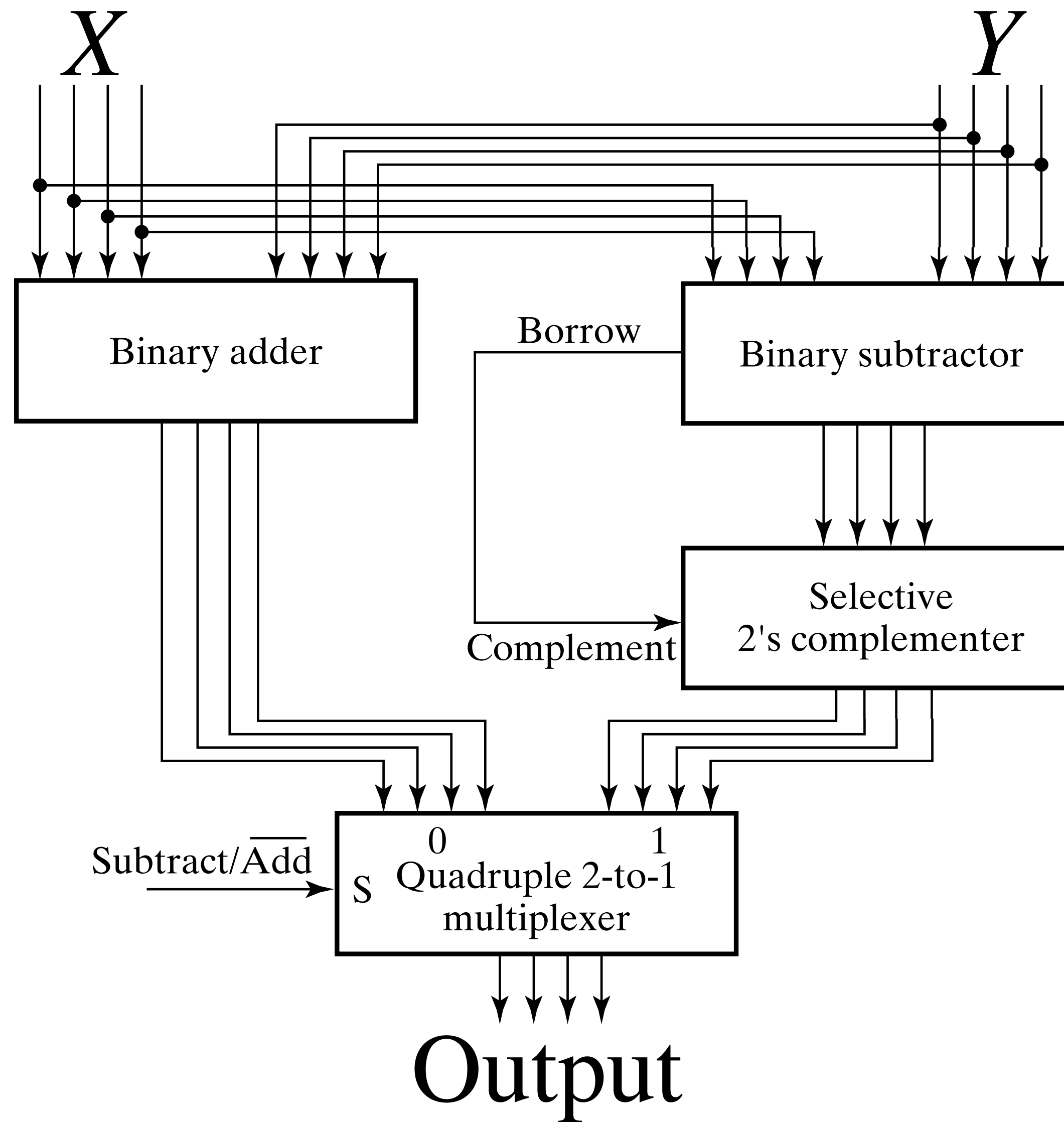
Full Unsigned Subtraction

- Solution 1
 - Compare the Minuend and Subtrahend, switch places if the Subtrahend is greater, then add negative sign
- Solution 2
 - Use 2s compliment

Full Unsigned Subtraction



Adder-Subtractor



Unsigned Binary Subtraction III

What do you mean we can do subtraction using an adder?

2s compliment

$$2^n - D = \bar{D} + 1$$

Subtraction using 2s Complement

- Input: X and Y
- Y : 2s complement $Y' = \bar{Y} + 1$ ($\bar{Y} + 1 = 2^n - Y$)
- $X - Y = X + (2^n - Y) - 2^n = X + Y' - 2^n$
- Since $2^n = (10\dots0)_2$, and we only output n -bits, it can be discarded

Subtraction using 2s Complement

- $84 - 67$ (10bit)

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$

$$Y = 0001000011$$

Subtraction using 2s Complement

- 84 - 67 (10bit)

2s complement

$$X = 0001010100$$

$$Y = 0001000011$$

$$Y' = 1110111101$$

Subtraction using 2s Complement

- 84 - 67 (10bit)

2s complement

$$X = 0001010100$$

$$Y = 0001000011$$

$$Y' = 1110111101$$

$$X + Y' = 10000010001$$

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$

$$Y = 0001000011$$

2s complement

$$Y' = 1110111101$$

$$X + Y' = 10000010001$$

discard carry

$$X + Y' - 2^{10} = 0000010001$$

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$

$$Y = 0001000011$$

2s complement

$$Y' = 1110111101$$

$$X + Y' = 10000010001$$

discard carry

$$X + Y' - 2^{10} = 0000010001$$

verify

$$(84 - 67)_{10} = 17_{10} = 10001 = 0000010001$$

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$

$$Y = 0001000011$$

2s complement

$$Y' = 1110111101$$

$$X + Y' = 10000010001$$

discard carry

$$X + Y' - 2^{10} = 0000010001$$

correct!

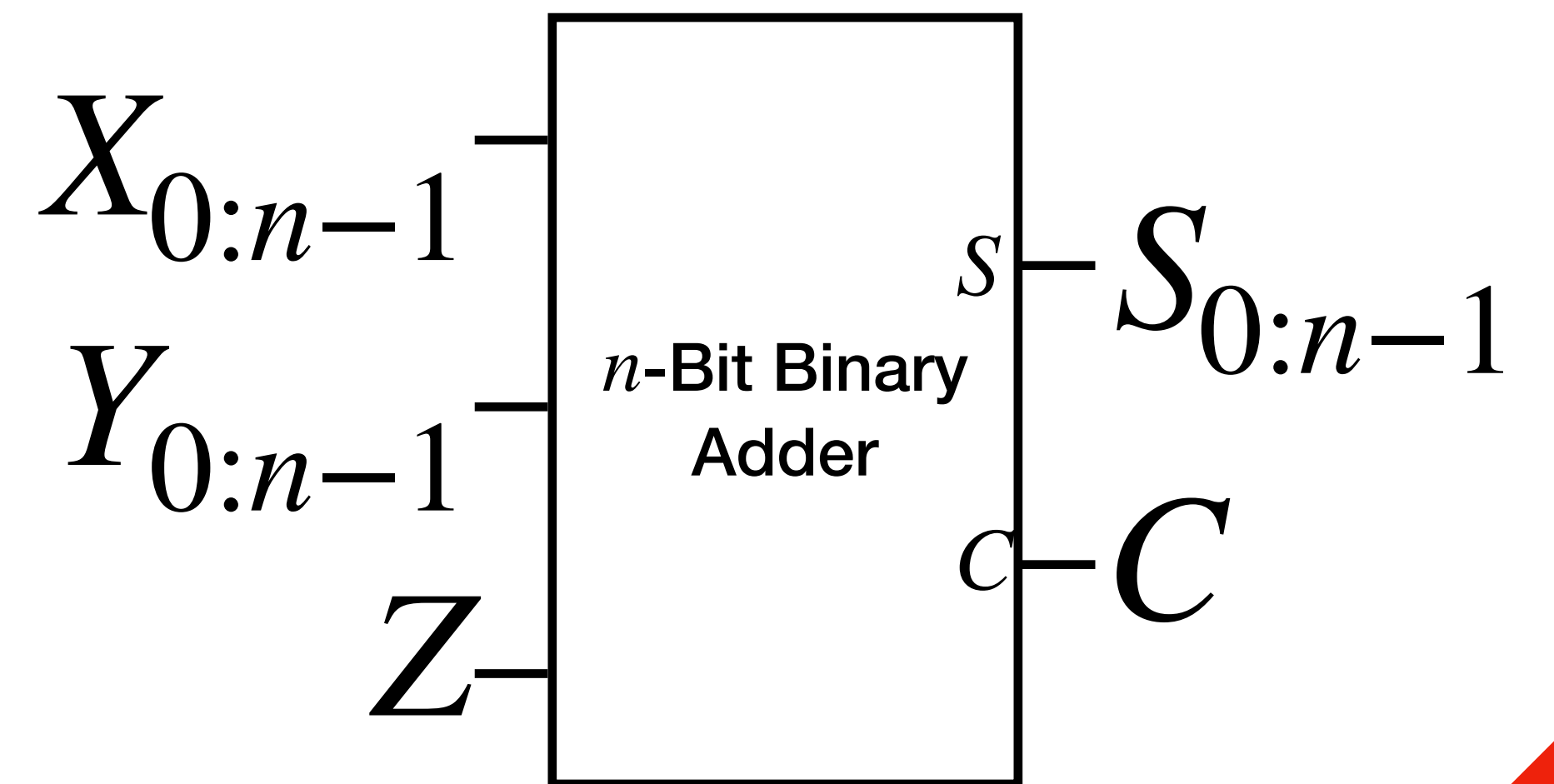
verify

$$(84 - 67)_{10} = 17_{10} = 10001 = 0000010001$$

Adder-Subtractor Unit

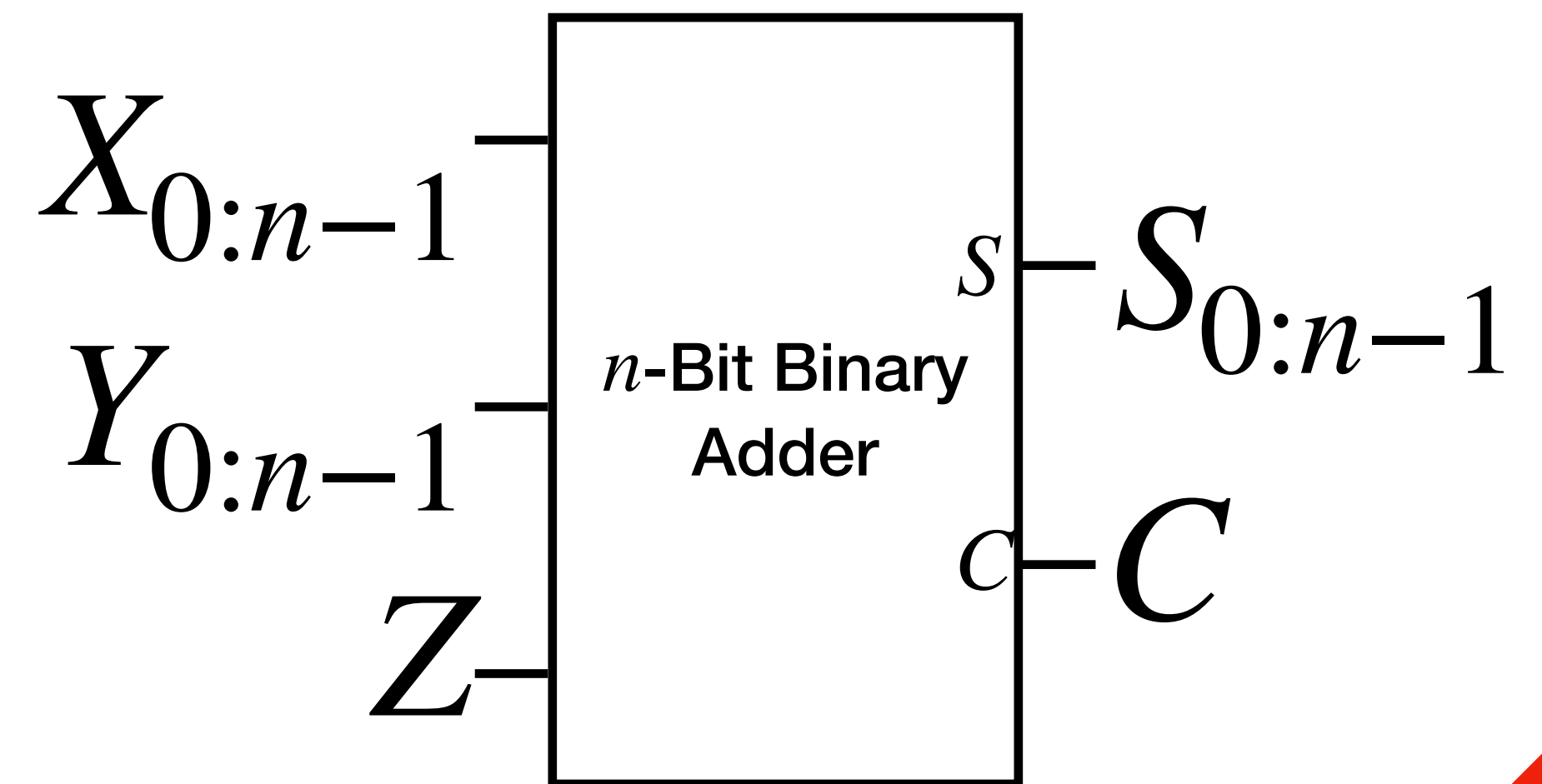
Adder-Subtractor Unit

1. Adder



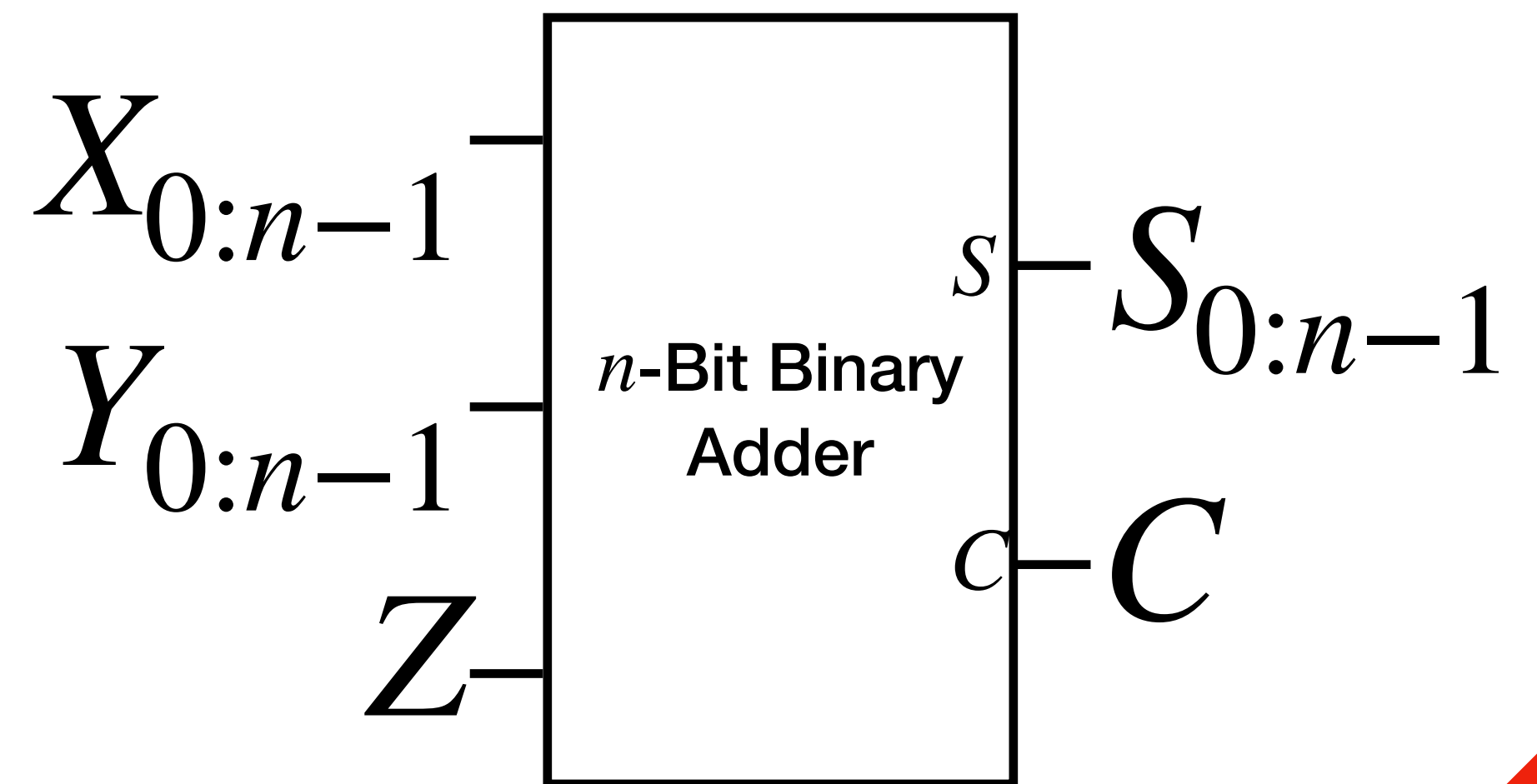
Adder-Subtractor Unit

1. Adder
2. Complementer (Inverting and add 1)



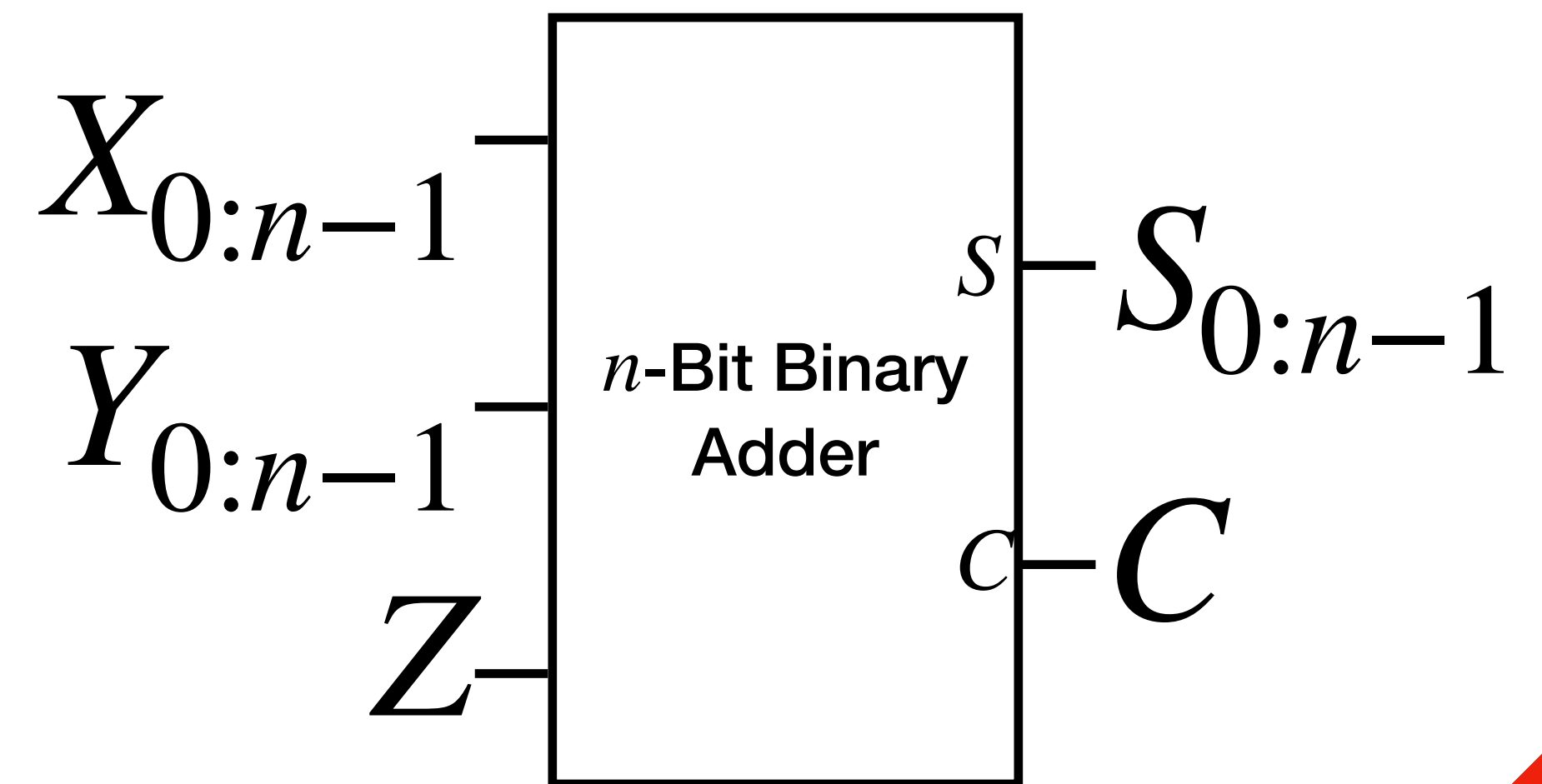
Adder-Subtractor Unit

1. Adder
2. Complementer (Inverting and add 1)
 - Or just inverting, and then plus one



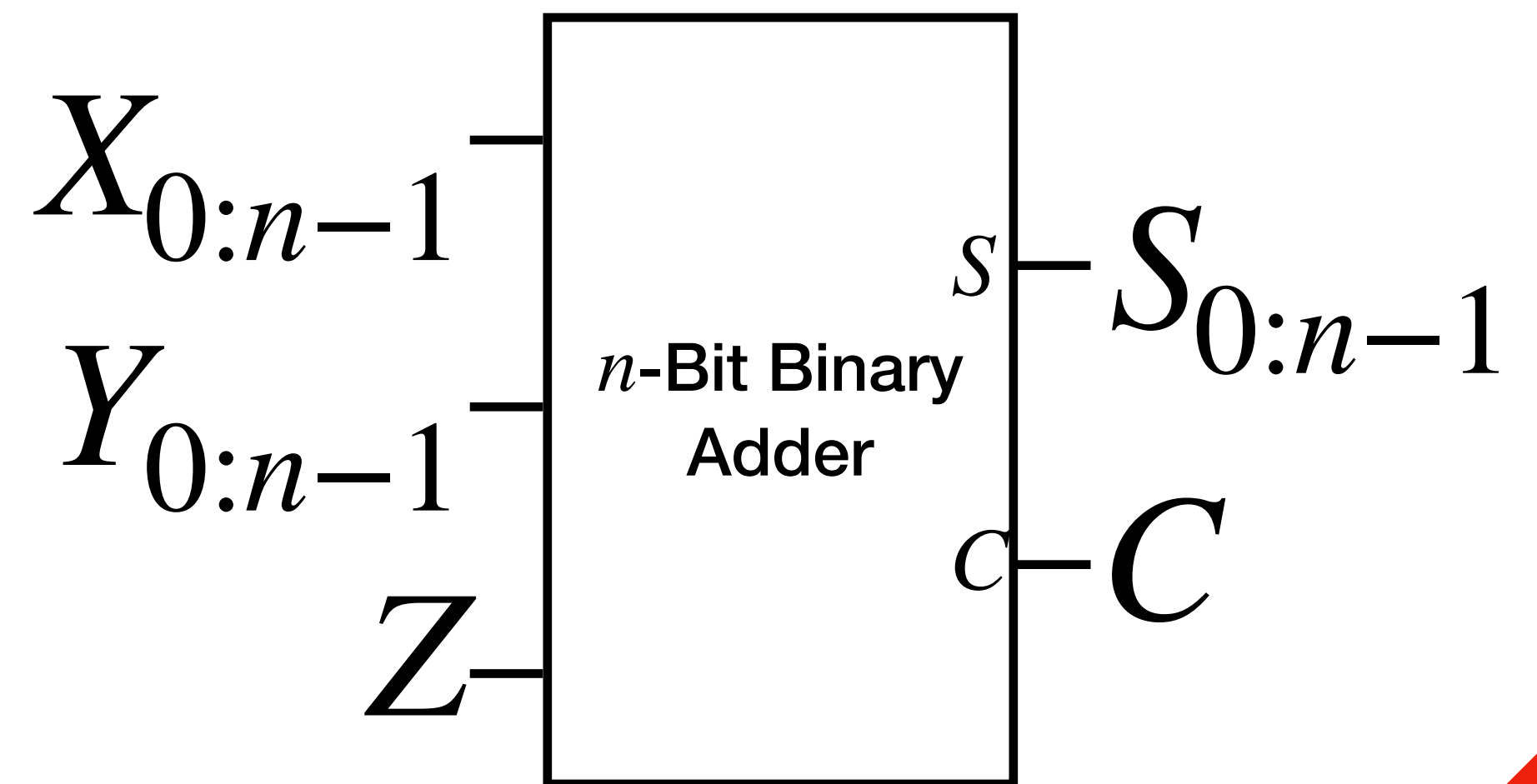
Adder-Subtractor Unit

1. Adder
 2. Complementer (Inverting and add 1)
 - Or just inverting, and then plus one
- Addition: $X + Y$



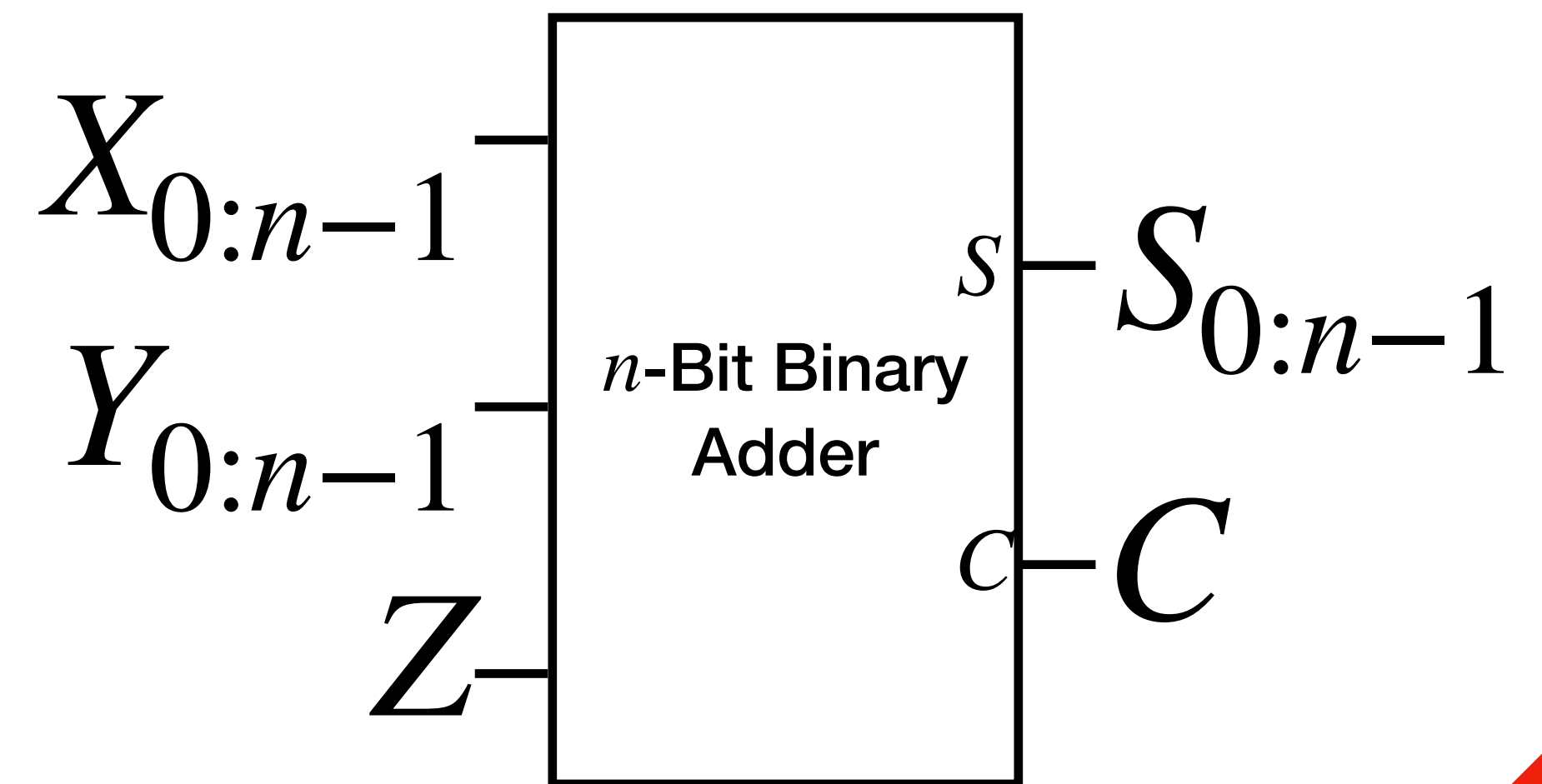
Adder-Subtractor Unit

1. Adder
 2. Complementer (Inverting and add 1)
 - Or just inverting, and then plus one
- Addition: $X + Y$
 - Subtraction: $X - Y = X + \bar{Y} + 1 - 2^n$



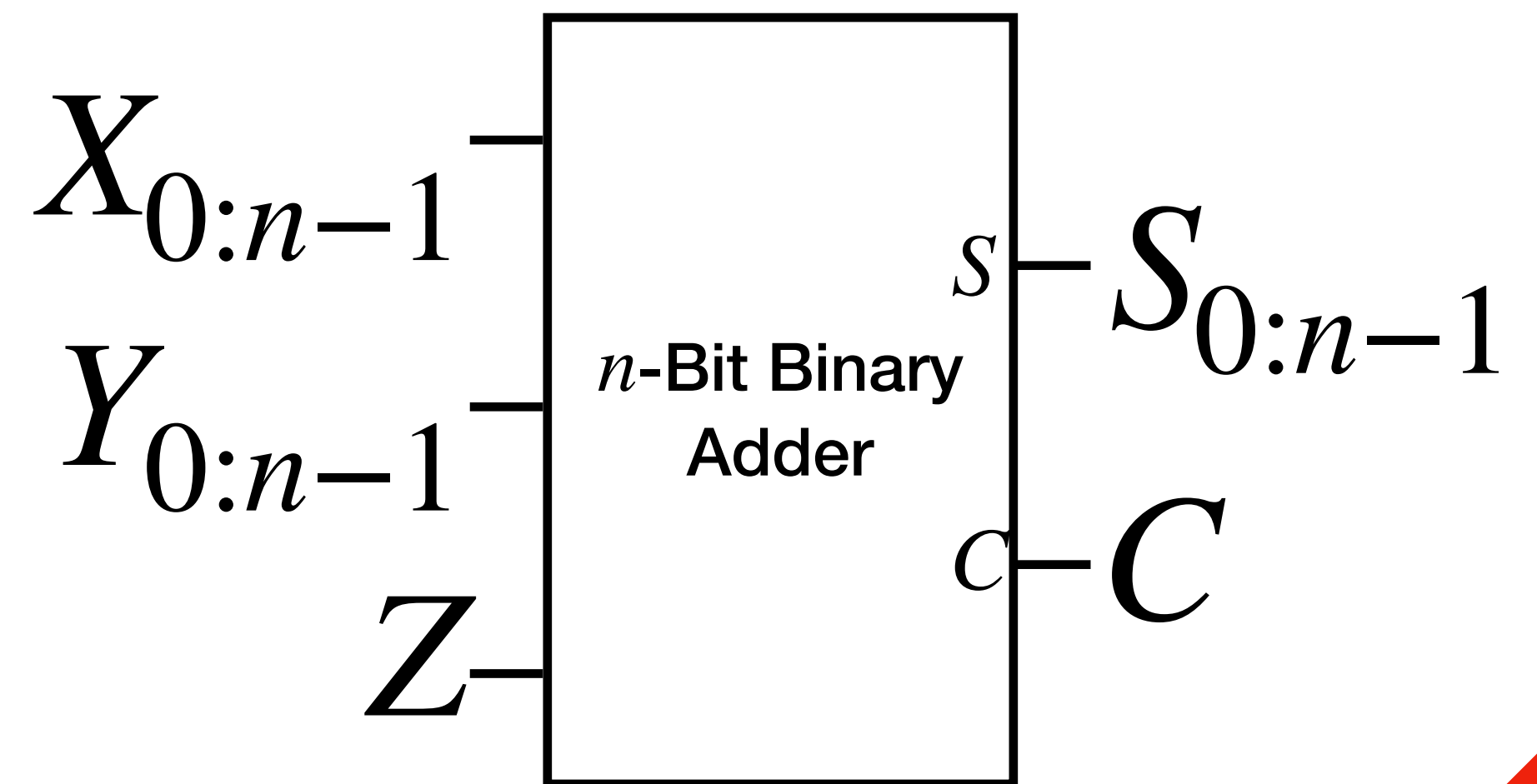
Adder-Subtractor Unit

1. Adder
 2. Complementer (Inverting and add 1)
 - Or just inverting, and then plus one
- Addition: $X + Y$
 - Subtraction: $X - Y = X + \bar{Y} + 1 - 2^n$
 - We are using n -bit adder, 2^n can be disregarded

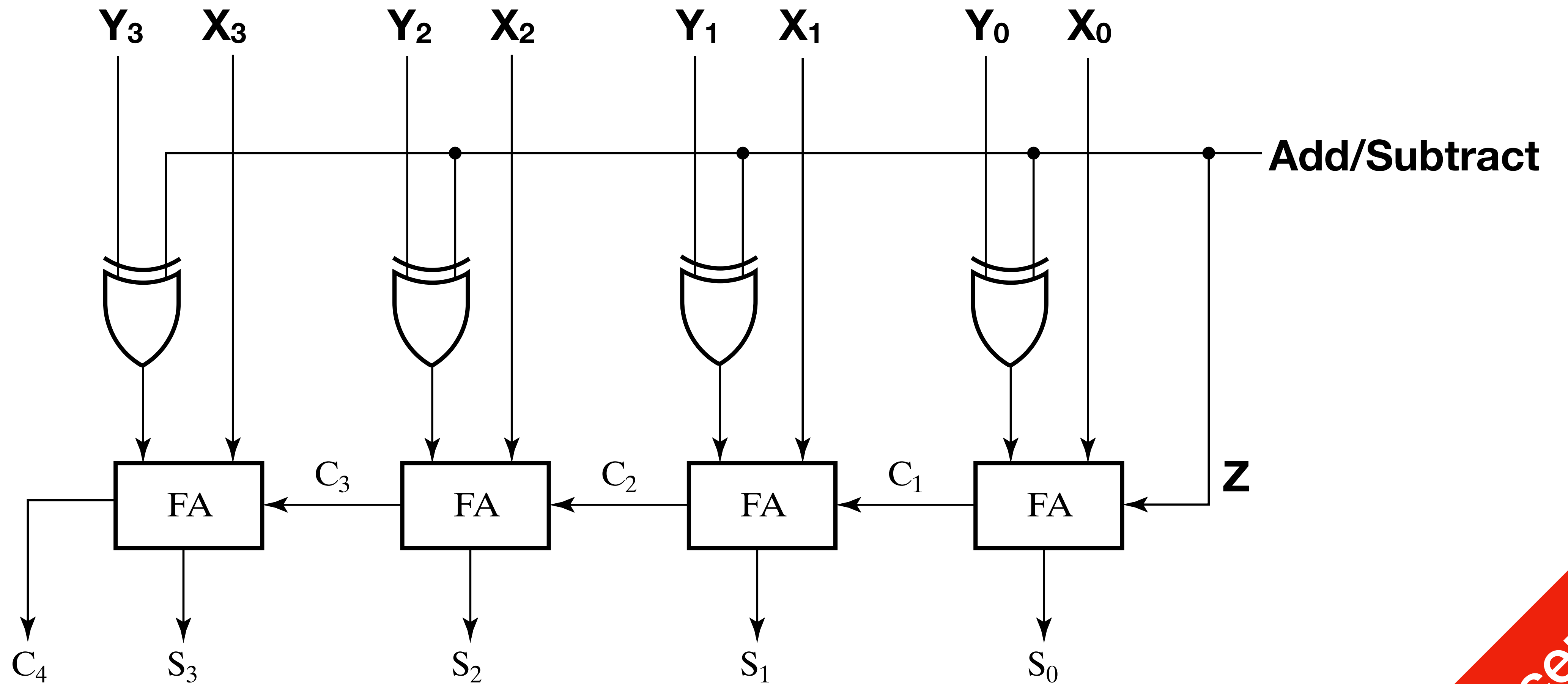


Adder-Subtractor Unit

1. Adder
 2. Complementer (Inverting and add 1)
 - Or just inverting, and then plus one
- Addition: $X + Y$
 - Subtraction: $X - Y = X + \bar{Y} + 1 - 2^n$
 - We are using n -bit adder, 2^n can be disregarded
 - The plus 1 here can be Z input to the adder



Adder-Subtractor Unit



Adder Subtractor Units (Unsigned)

- Binary Adder
- Binary Subtractor
- Binary Adder-Subtractor Unit, using Adder, Subtractor, Complementer and Multiplexer
- Binary Adder-Subtractor Unit using Adder and XOR

Excercises

2s Complement
Subtraction using 2s Complement

Subtraction

Subtraction

- Compute $10-7$ using 4-bit binary using Adder and 2s complement

Subtraction

Subtraction

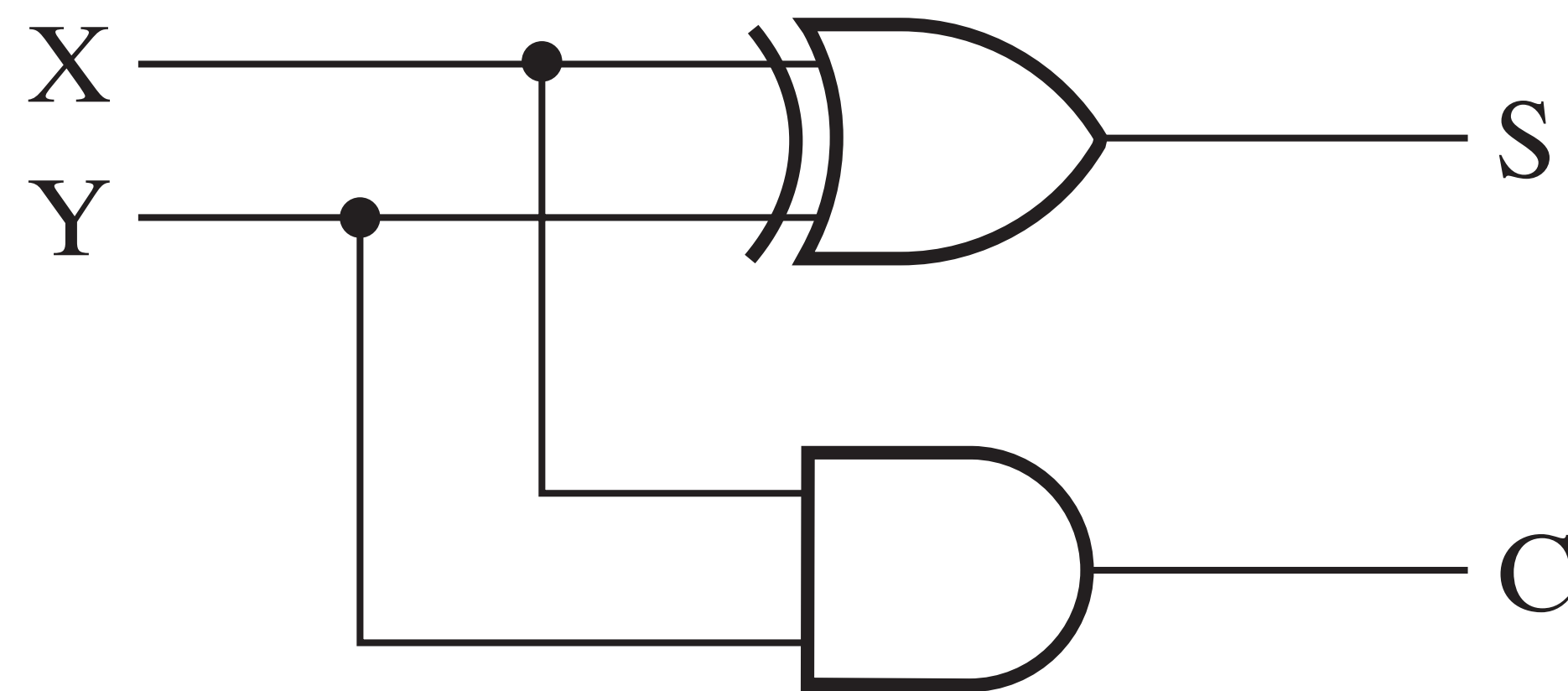
- Compute $24-17$ using 8-bit binary using Adder and 2s complement

Hardware Description Language

VHDL (VHSIC-HDL): Very High Speed Integrated
Circuit Hardware Description Language

Previous: 1-bit Half Adder

- Create a new component in VHDL called `HalfAdder1`
- Input: X, Y
- Output: S, C
- Don't use `AFTER`



Previous: 1-bit Half Adder

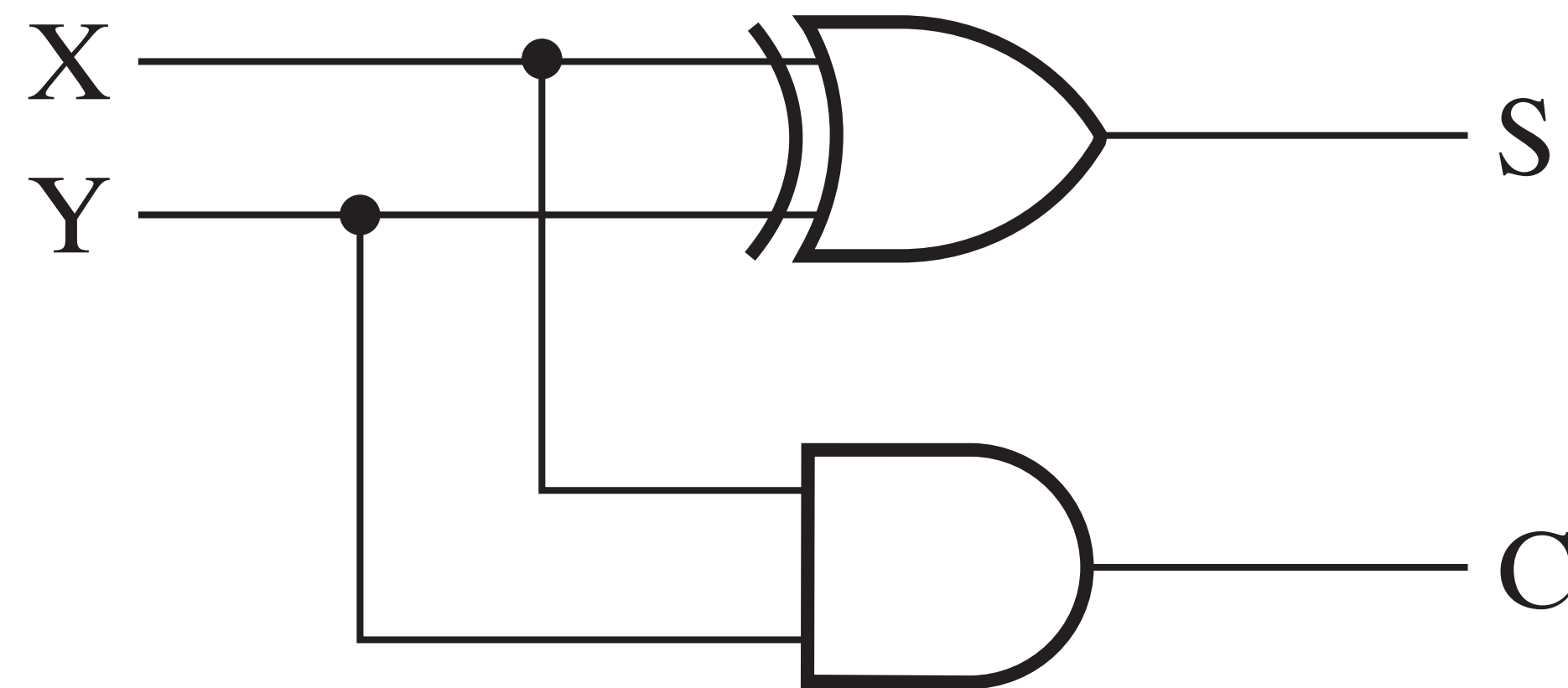
architecture arch1 of HalfAdder is

begin

```
S <= X XOR Y;
```

```
C <= X AND Y;
```

end arch1;



Today's Tasks

Today's Tasks

- 1-bit Half Adder

Today's Tasks

- 1-bit Half Adder
- 1-bit Full Adder using Schema Diagram (Logic Circuit Diagram)

Today's Tasks

- 1-bit Half Adder
- 1-bit Full Adder using Schema Diagram (Logic Circuit Diagram)
- 4-bit Full Adder using Schema Diagram (Logic Circuit Diagram)

Today's Tasks

- 1-bit Half Adder
- 1-bit Full Adder using Schema Diagram (Logic Circuit Diagram)
- 4-bit Full Adder using Schema Diagram (Logic Circuit Diagram)
- 4-bit Adder-Subtractor using Schema Diagram (Logic Circuit Diagram)