19.10.20 12:30

CSCI 150
Introduction to Digital and Computer System Design
Lecture 3: Combinational Logic Design VII

Jetic Gū

2020 Fall Semester (S3)

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

P0
Review

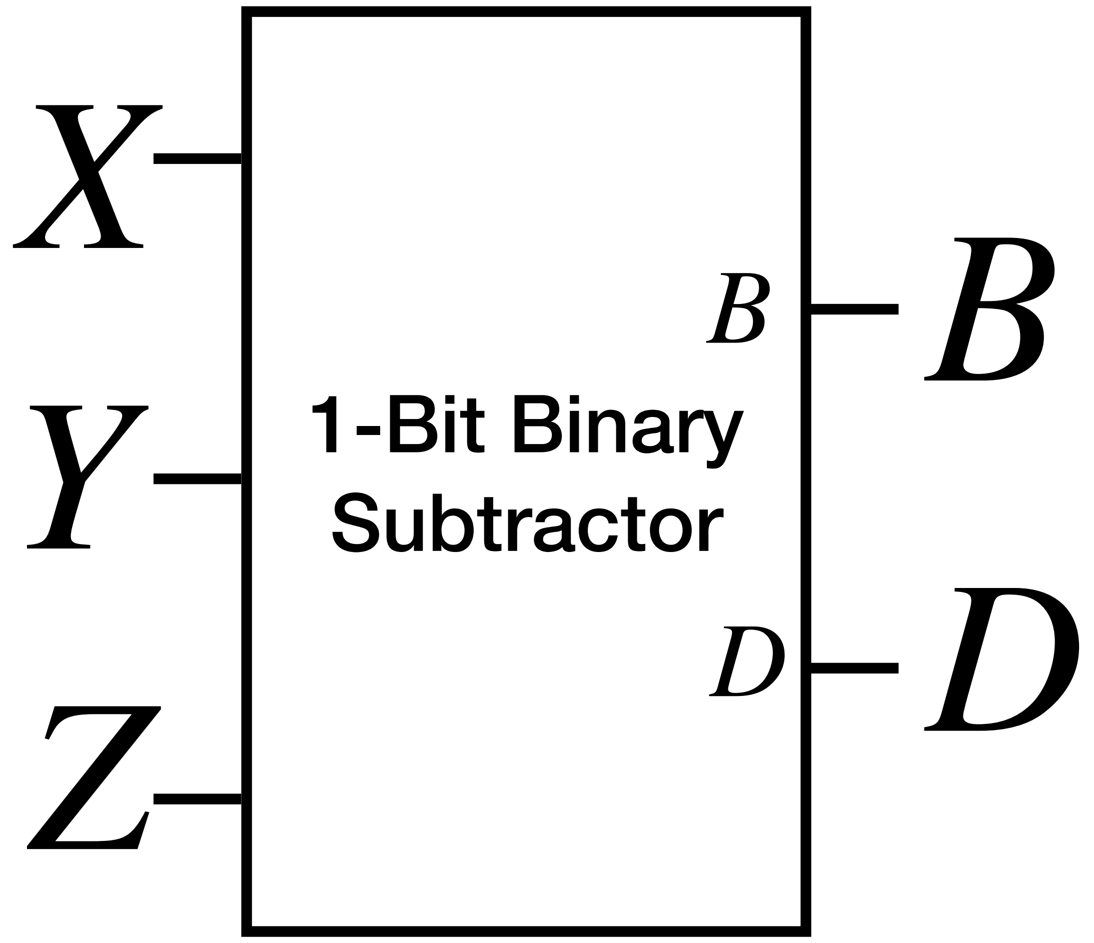
Unsigned Binary Subtraction I

Review

Summary

Unsigned 1-bit Binary Subtraction

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

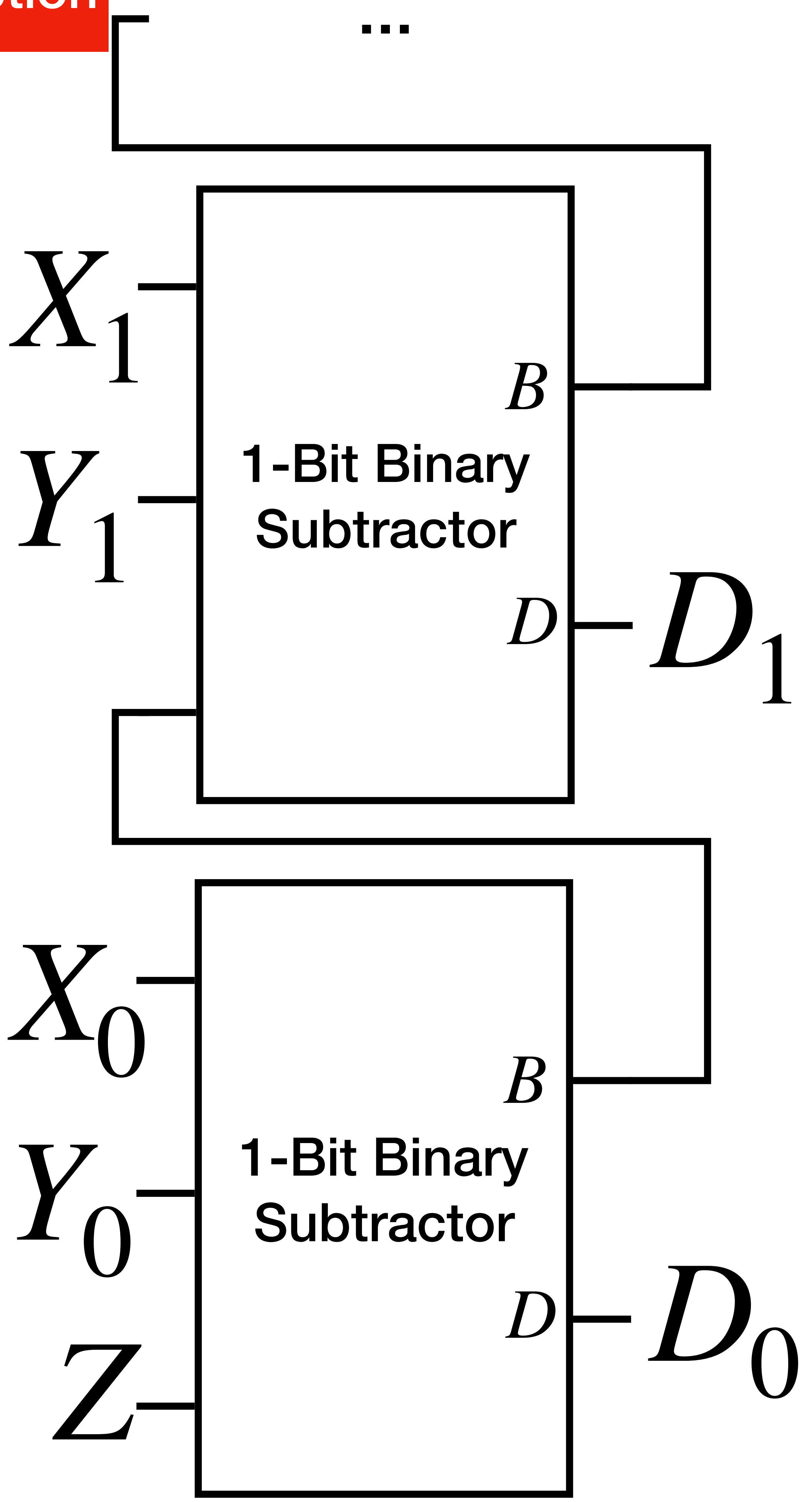


Review

P1
Subtraction

Unsigned Binary Subtraction

Technology
- 1 bit Unsigned Subtractor



	B $\qquad\qquad$ Z
Borrows	000110
Minuend $X_{0:n-1}$	10110
Subtrahend $Y_{0:n-1}$	− 10011
Difference $D_{0:n-1}$	00011

Input
Output

Review

Unsigned Binary Subtraction II

$X - Y$ when $Y > X$

P1
Subtraction

What we have so far

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

Review

P1 Subtraction

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. $X < Y, F = X - Y$?

Concept

Unsigned Binary Subtraction

Borrows	
Minuend	10011
Subtrahend	-00110
Difference	

Borrows	
Minuend	00110
Subtrahend	-10011
Difference	

Concept

Unsigned Binary Subtraction

Borrows		**Borrows**	
Minuend	10011	**Minuend**	00110
Subtrahend	-00110	**Subtrahend**	-10011
Difference		**Difference**	

Concept

P1 Subtraction

Unsigned Binary Subtraction

Borrows	011000
Minuend	10011
Subtrahend	-00110
Difference	01101

Borrows	100110
Minuend	00110
Subtrahend	-10011
Difference	10011

Concept

Unsigned Binary Subtraction

Borrows	011000
Minuend	10011
Subtrahend	−00110
Difference	01101

This is correct

Borrows	100110
Minuend	00110
Subtrahend	−10011
Difference	10011

Concept

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

Concept

Unsigned Binary Subtraction

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

Borrows	011000
Minuend	10011
Subtrahend	$-$00110
Difference	01101

This is correct

Borrows	100110
Minuend	00110
Subtrahend	$-$10011
Difference	10011

This is incorrect

Concept

P1 Subtraction

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

P1 Subtraction

Unsigned Binary Subtraction

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

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

Borrows	$\color{red}{1}00110$
Minuend X	00110
Subtrahend Y	-10011
Difference D	10011
Corrected $\color{red}{D'}$	$\color{red}{-01101}$

(2's compliment)

This is incorrect

Concept

P1 Subtraction

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'
$$= -(Y - X)$$
$$= -(2^n - D)$$
$$= -(\overline{D} + 1)$$ **(2's compliment)**

Borrows	100110
Minuend X	00110
Subtrahend Y	-10011
Difference D	10011
Corrected D'	-01101

This is incorrect

Concept

P1
Subtraction

2s compliment

- Given binary unsigned integer of n bits D, its 2s compliment $2^n - D = \overline{D} + 1$
- Proof
  - Biggest number represented in n bit: $(11\ldots1)_2 = 2^n - 1$
  - $2^n - D = [(11\ldots1)_2 + 1] - D = (11\ldots1)_2 - D + 1 = \overline{D} + 1$
- Implementation
  - Inversion and plus 1, easily doable as complementer

Concept

Subtraction

Example

Subtraction

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

Example

Subtraction

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

- $20 = (010100)_2, 15 = (001111)_2$

Example

Subtraction

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

- $20 = (010100)_2$, $15 = (001111)_2$

$$
\begin{array}{rr}
\textbf{Borrows} & {\color{red}0}011110 \\
 & 010100 \\
 & -001111 \\
\hline
 & 000101
\end{array}
$$

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

$$
\begin{array}{r}
{\color{red}0}011110 \\
010100 \\
-001111 \\
\hline
000101
\end{array}
$$

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

$$\begin{array}{lr} \textbf{Borrows} & {\color{red}0}011110 \\ & 010100 \\ & -001111 \\ \hline & 000101 \end{array}$$

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
   - Correction: $(111011)_2$ 2s complement: $(000101)_2$

Borrows $1\,100000$

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

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

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

Example

Subtraction

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

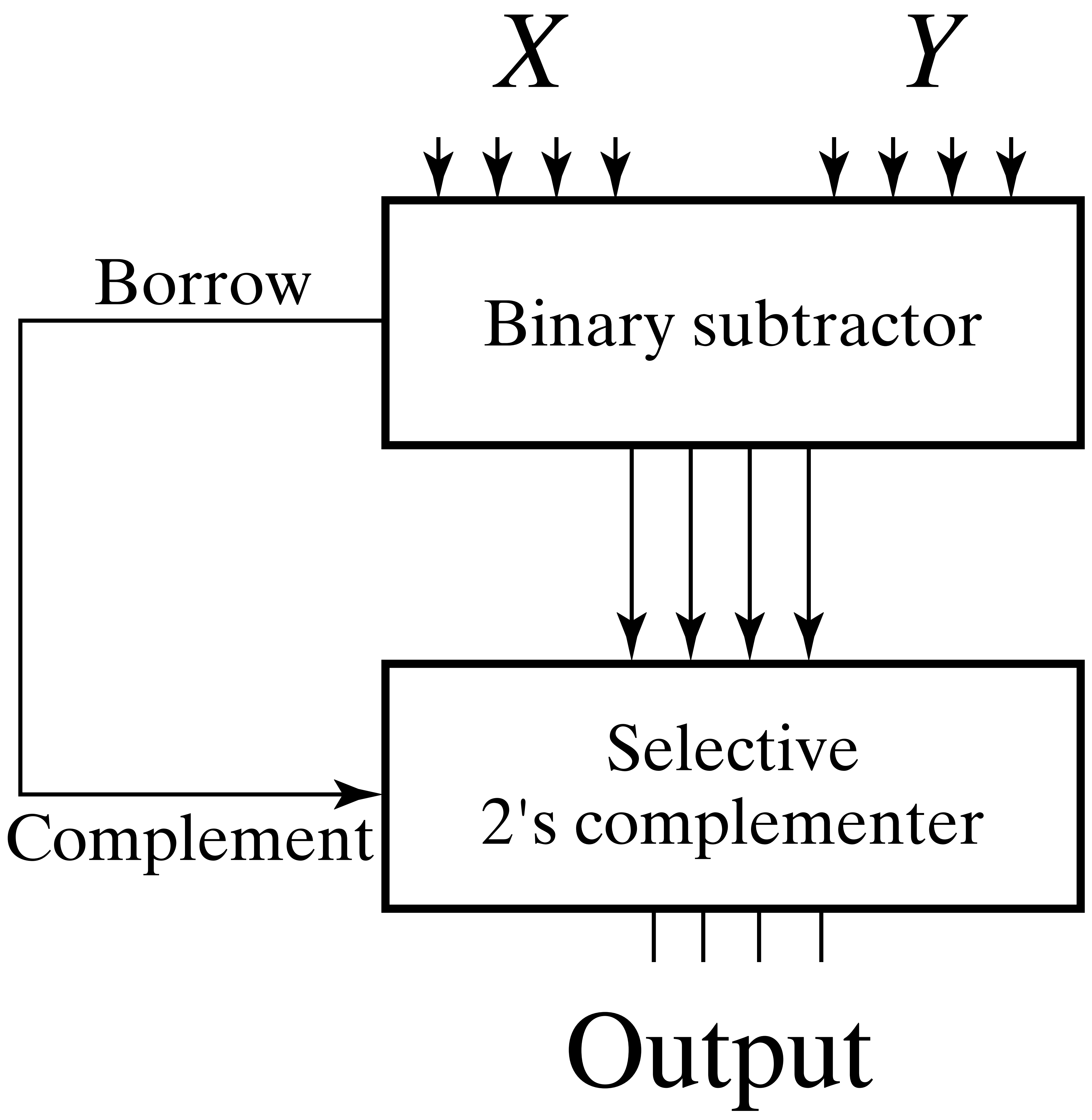
Example

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

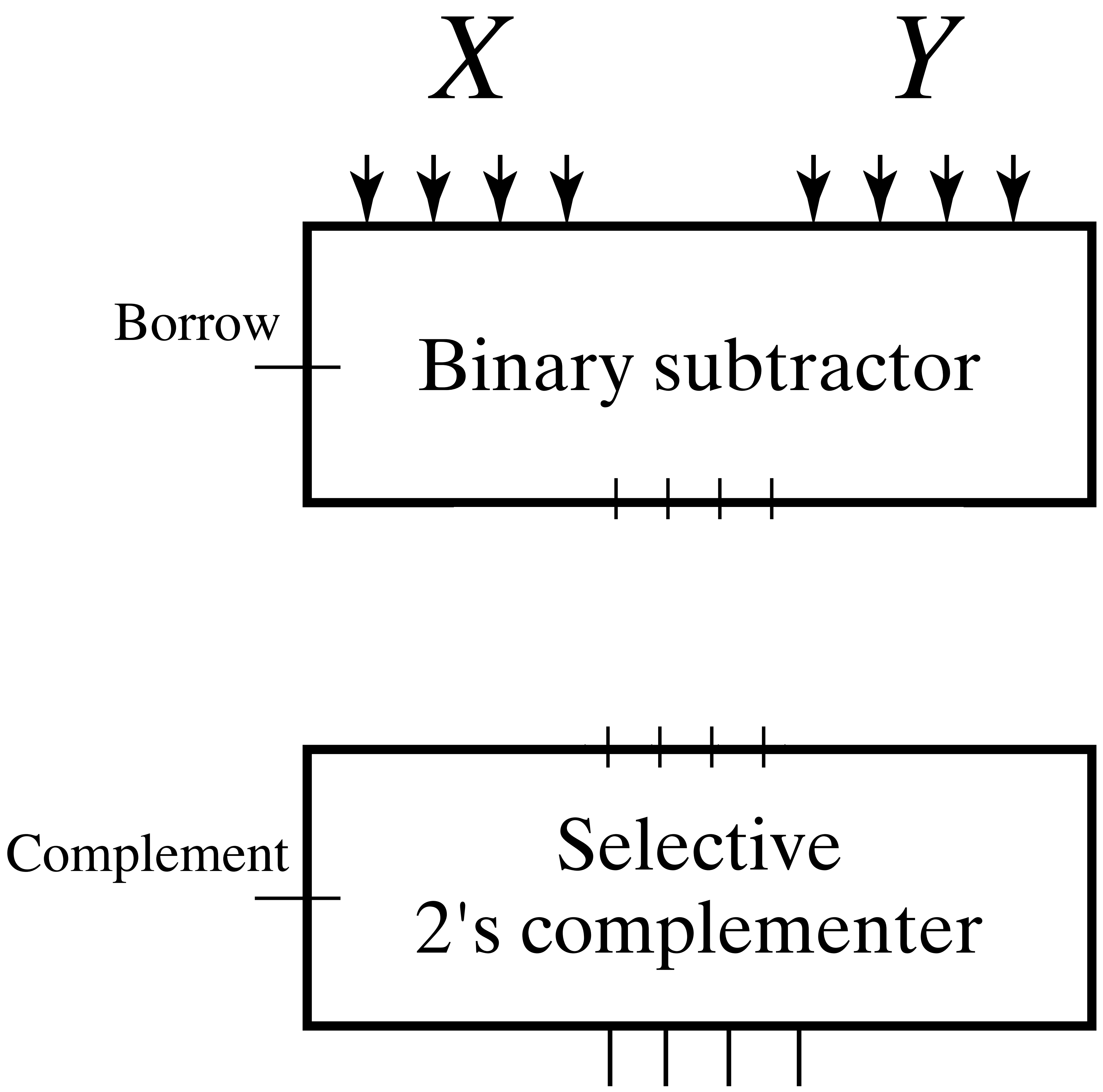
Concept

Full Unsigned Subtraction



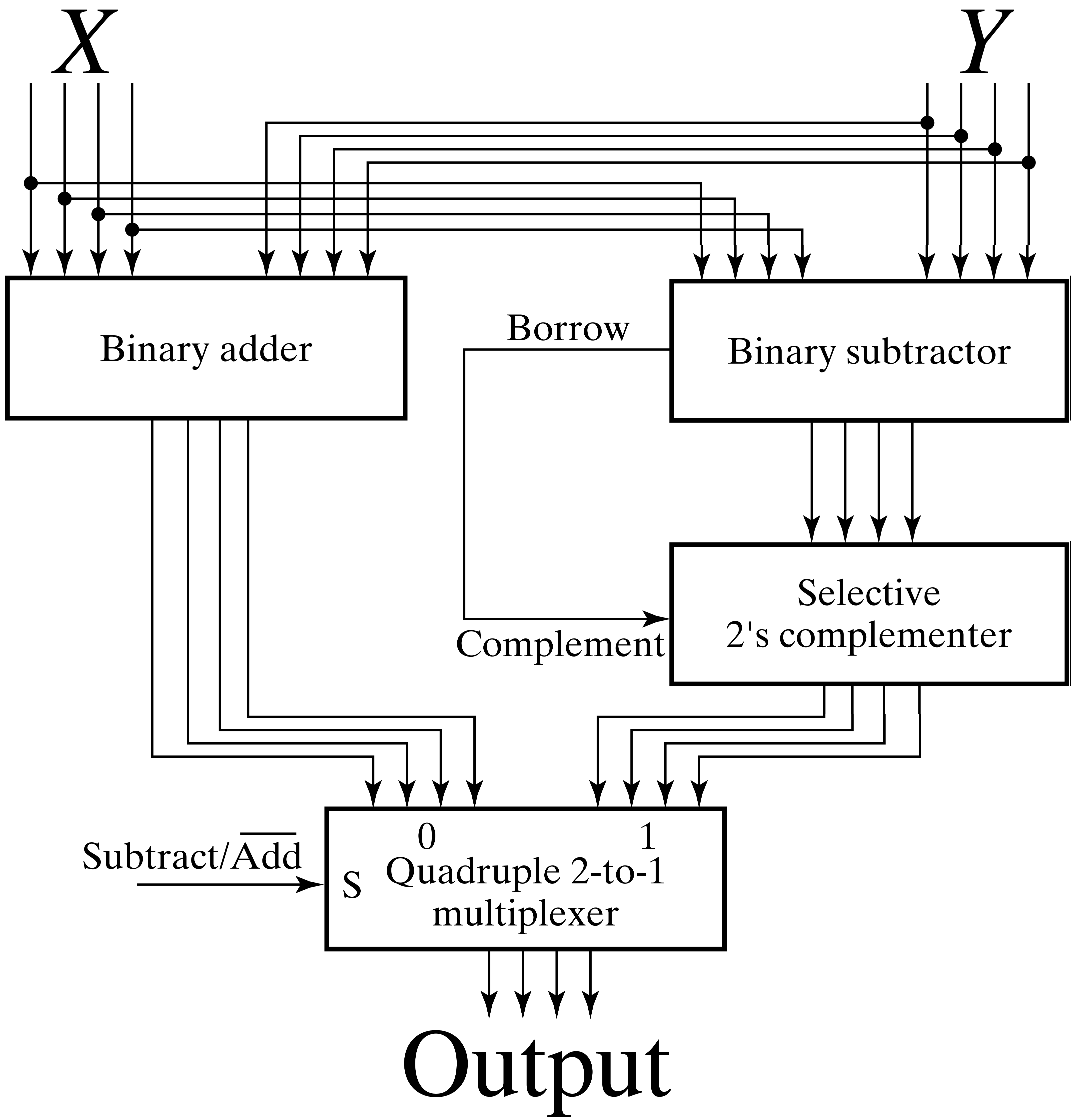
Concept

Full Unsigned Subtraction



P1
Subtraction

Adder-Subtractor



Concept

Unsigned Binary Subtraction III

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

2s compliment

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

P2 Subtraction

Subtraction using 2s Complement

- Input: X and Y
  - Y: 2s complement $Y' = \overline{Y} + 1$ ($\overline{Y} + 1 = 2^n - Y$)
  - $X - Y = X + (2^n - Y) - 2^n = X + Y' - 2^n$
  - Since $2^n = (10...0)_2$, and we only output n-bits, it can be discarded

Concept

P2
Subtraction

Subtraction using 2s Complement

- 84 - 67 (10bit)

Example

P2
Subtraction

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$

Example

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$
$$Y = 0001000011$$

Example

P2
Subtraction

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$
$$Y = 0001000011$$

2s complement $$Y' = 1110111101$$

Example

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$

$$Y = 0001000011$$

2s complement

$$Y' = 1110111101$$

$$X + Y' = 10000010001$$

Example

P2
Subtraction

Subtraction using 2s Complement

- 84 - 67 (10bit)

$$X = 0001010100$$

$$Y = 0001000011$$

2s complement $$Y' = 1110111101$$

$$X + Y' = 10000010001$$

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

Example

P2
Subtraction

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

Example

P2 Subtraction

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

Example

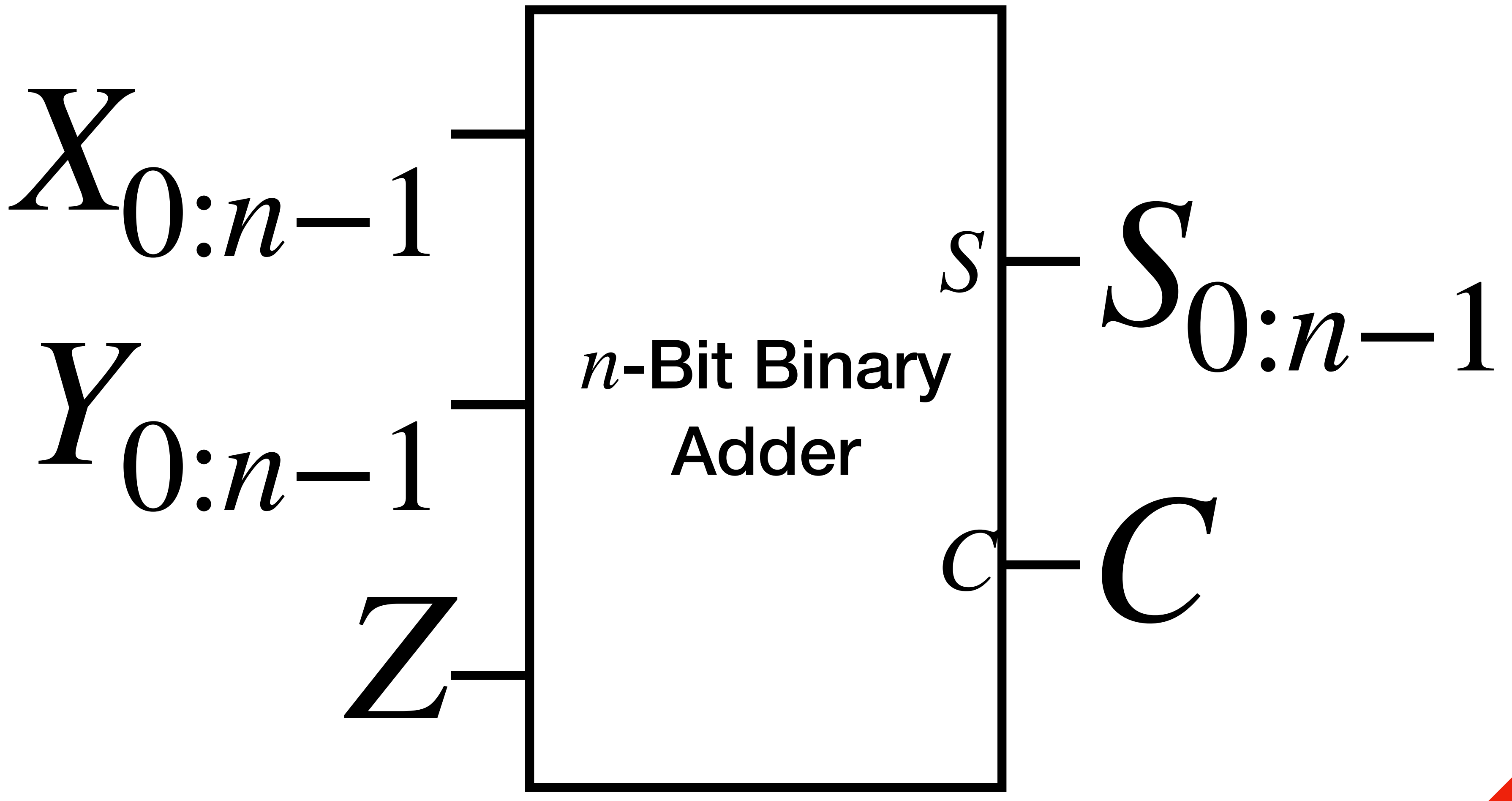
Adder-Subtractor Unit

Concept

P2
Subtraction

Adder-Subtractor Unit

1. Adder

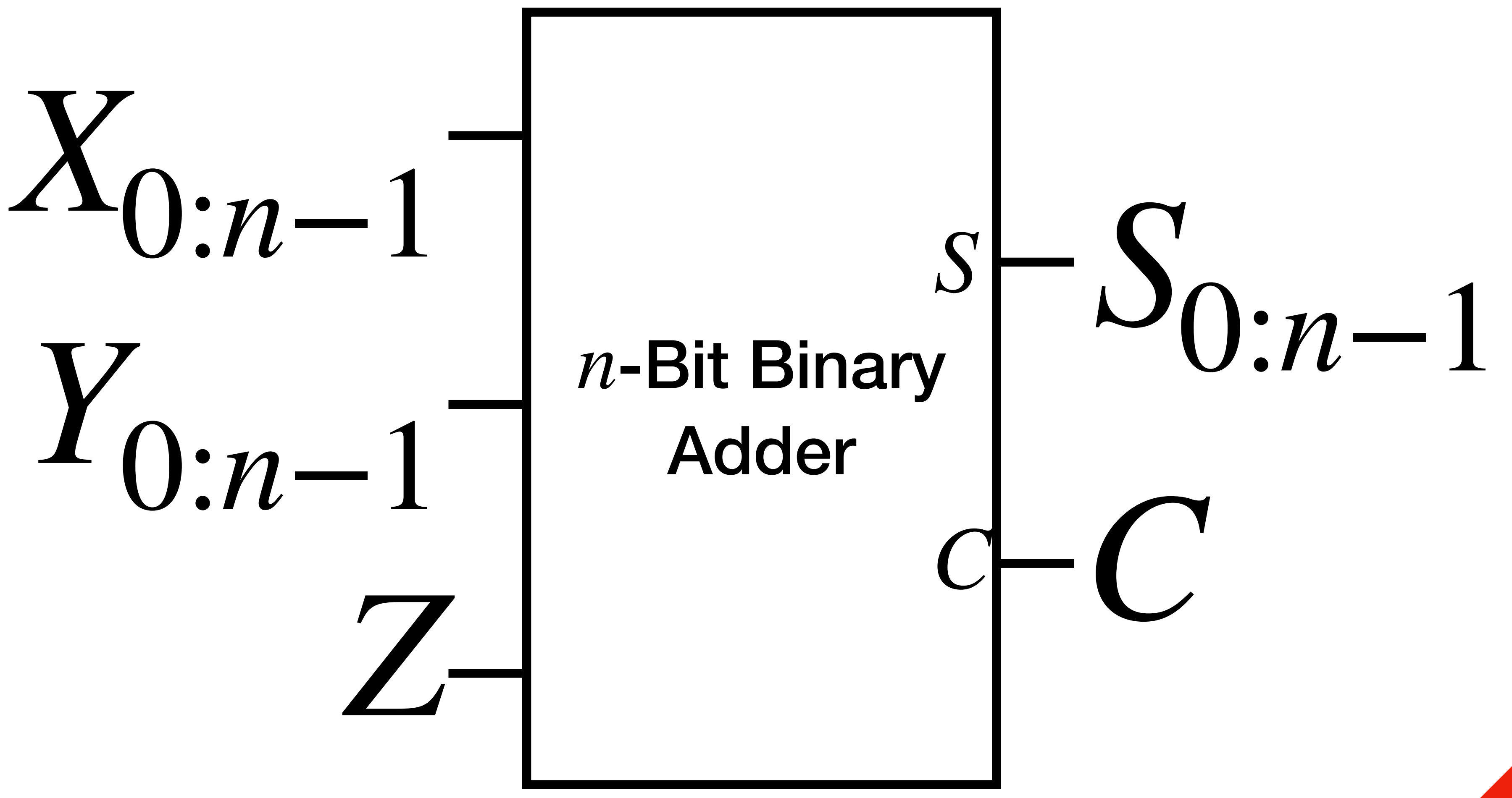


Concept

P2
Subtraction

Adder-Subtractor Unit

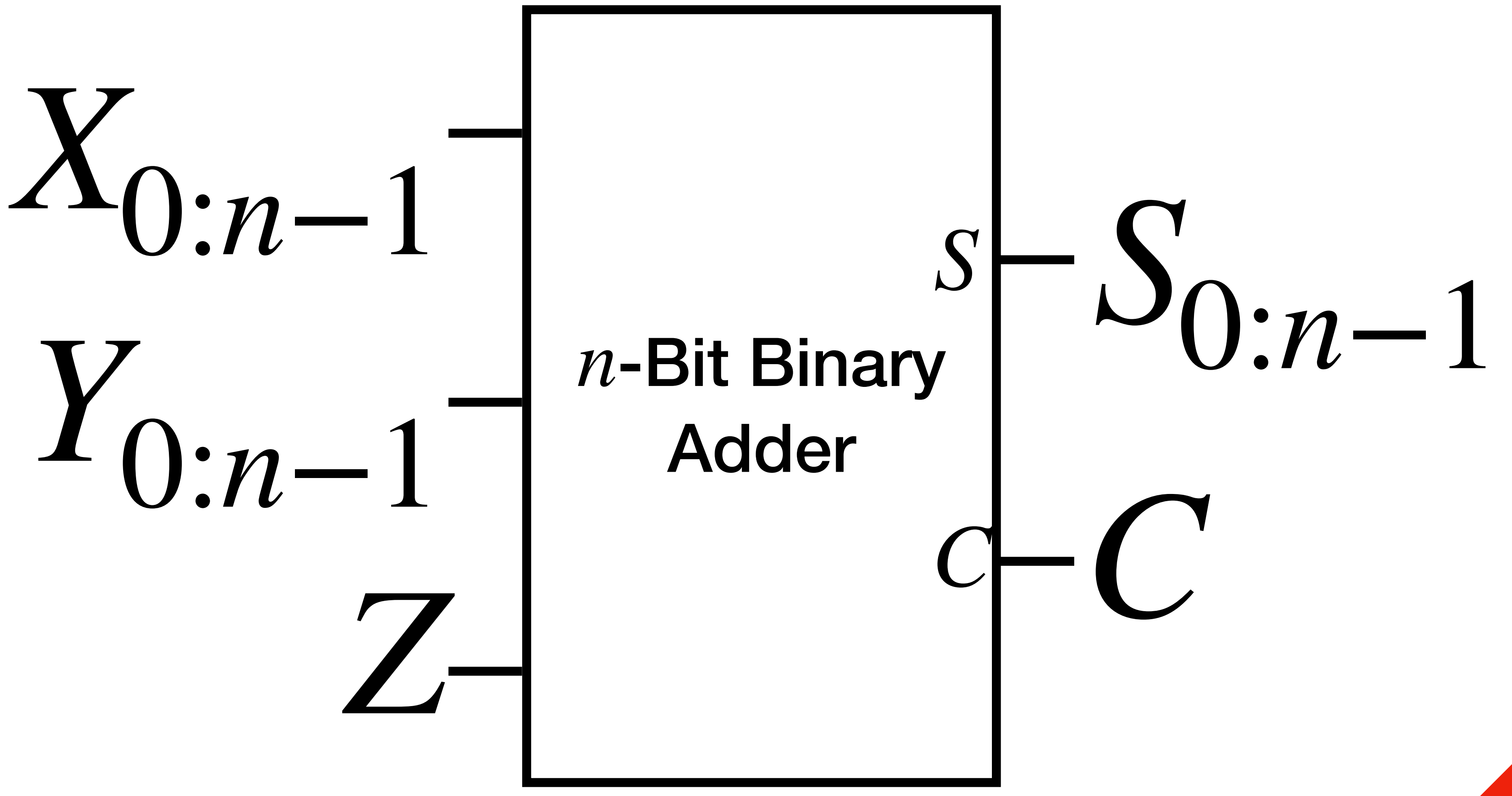
1. Adder
2. Complementer (Inverting and add 1)



Concept

Adder-Subtractor Unit

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

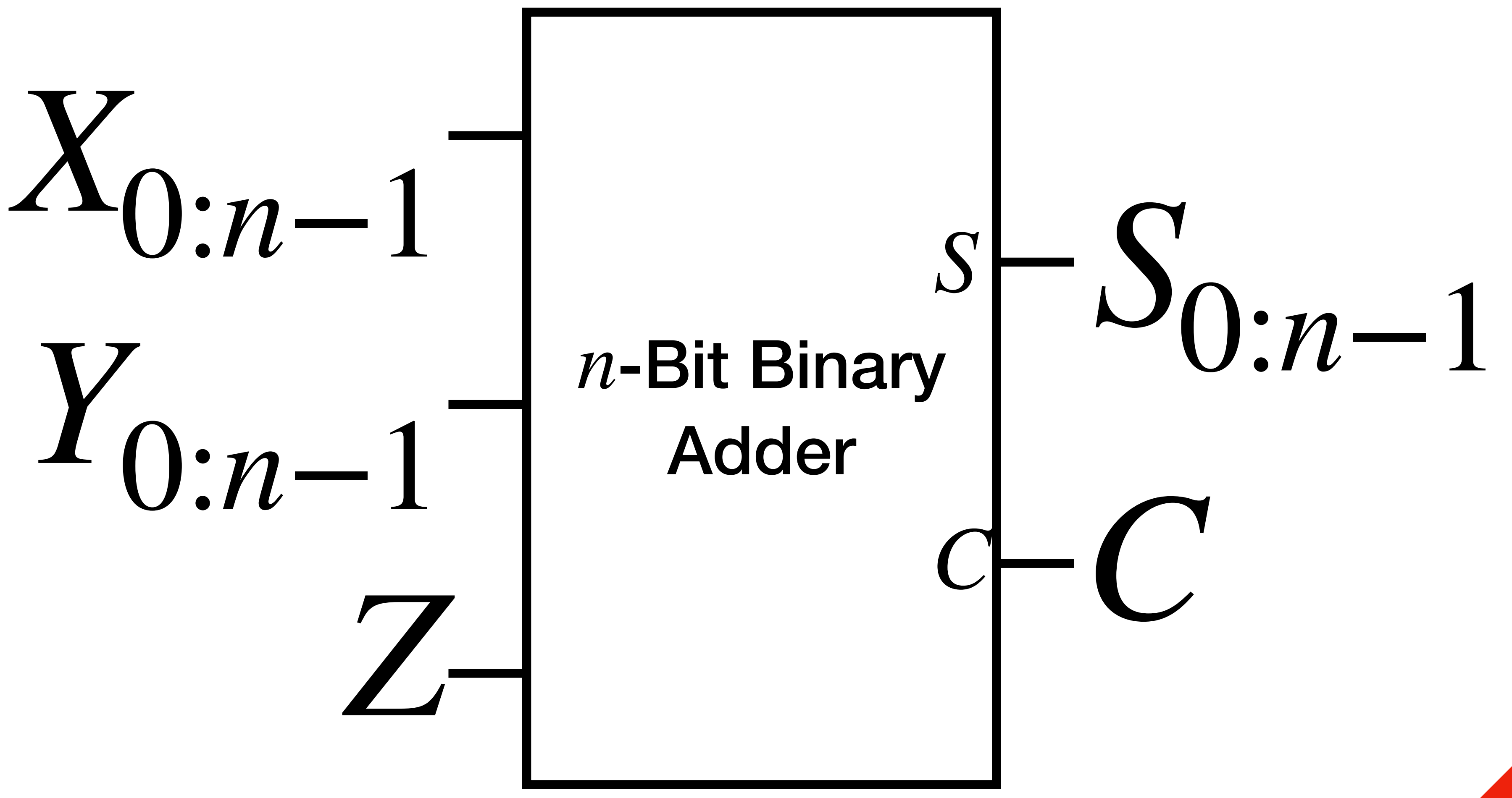


Concept

Adder-Subtractor Unit

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

- Addition: $X + Y$

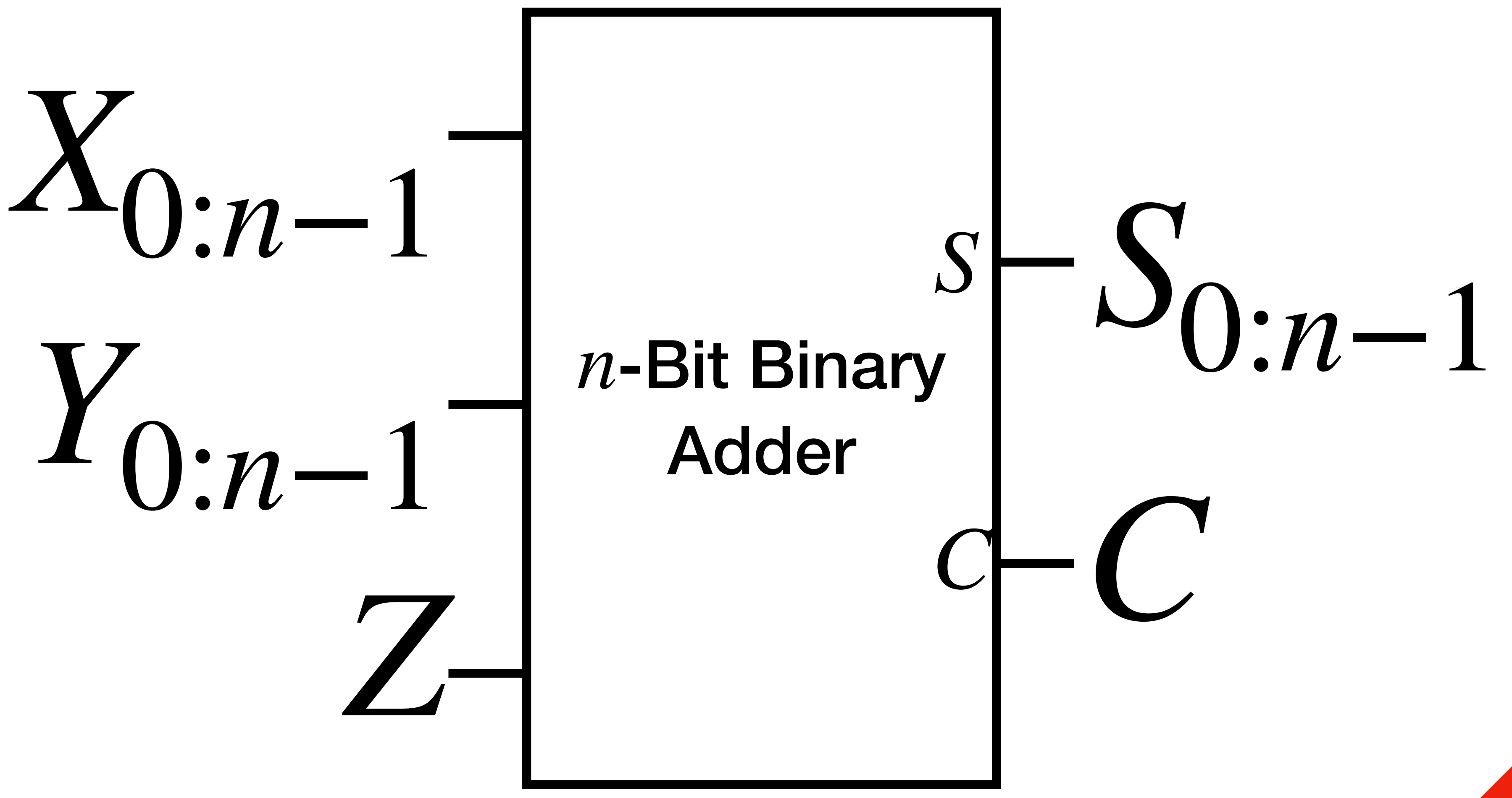


Concept

Adder-Subtractor Unit

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

- Addition: $X + Y$
- Subtraction: $X - Y = X + \overline{Y} + 1 - 2^n$

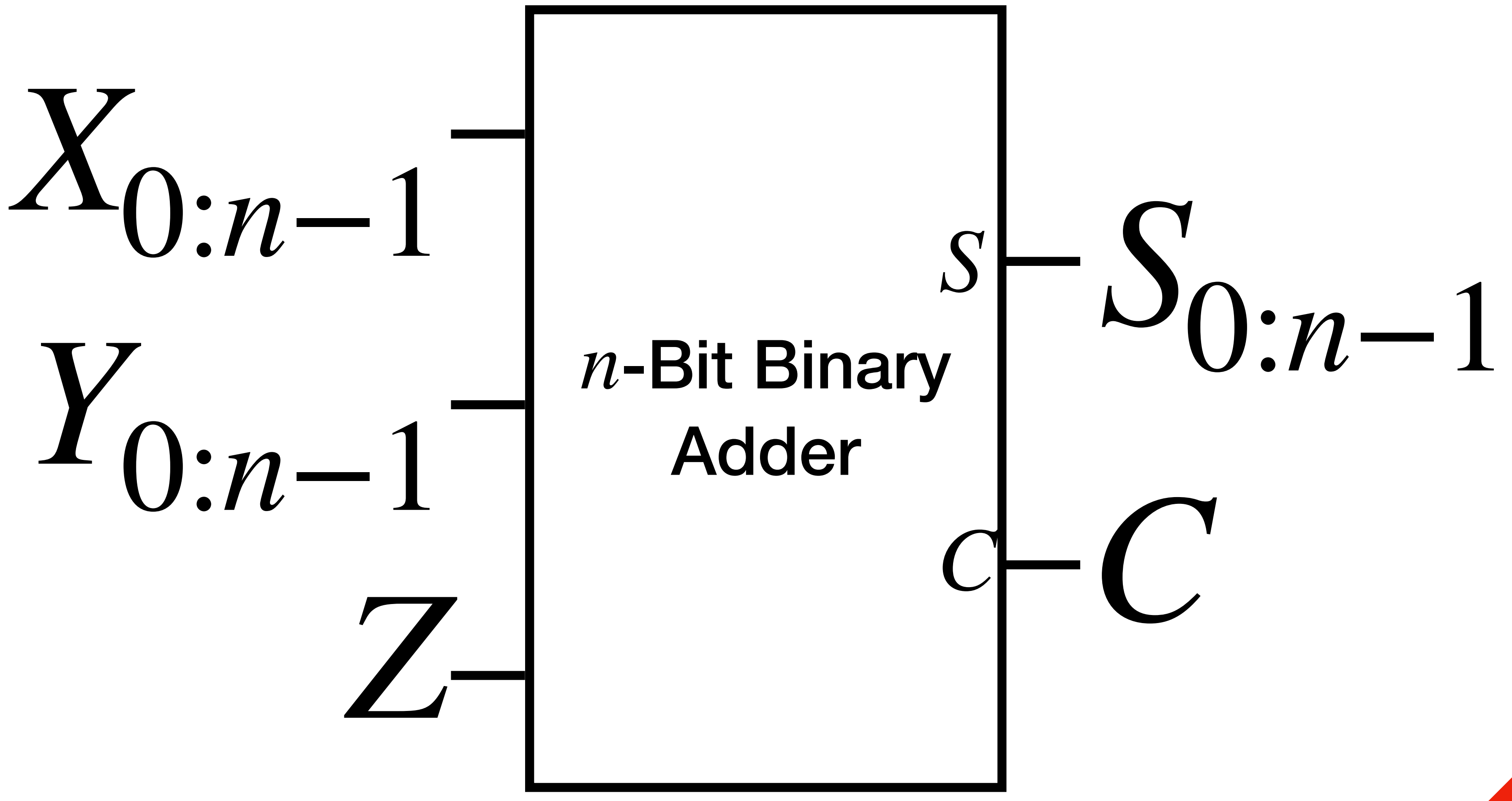


Concept

Adder-Subtractor Unit

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

- Addition: $X + Y$
- Subtraction: $X - Y = X + \overline{Y} + 1 - 2^n$
  - We are using n-bit adder, 2^n can be disregarded

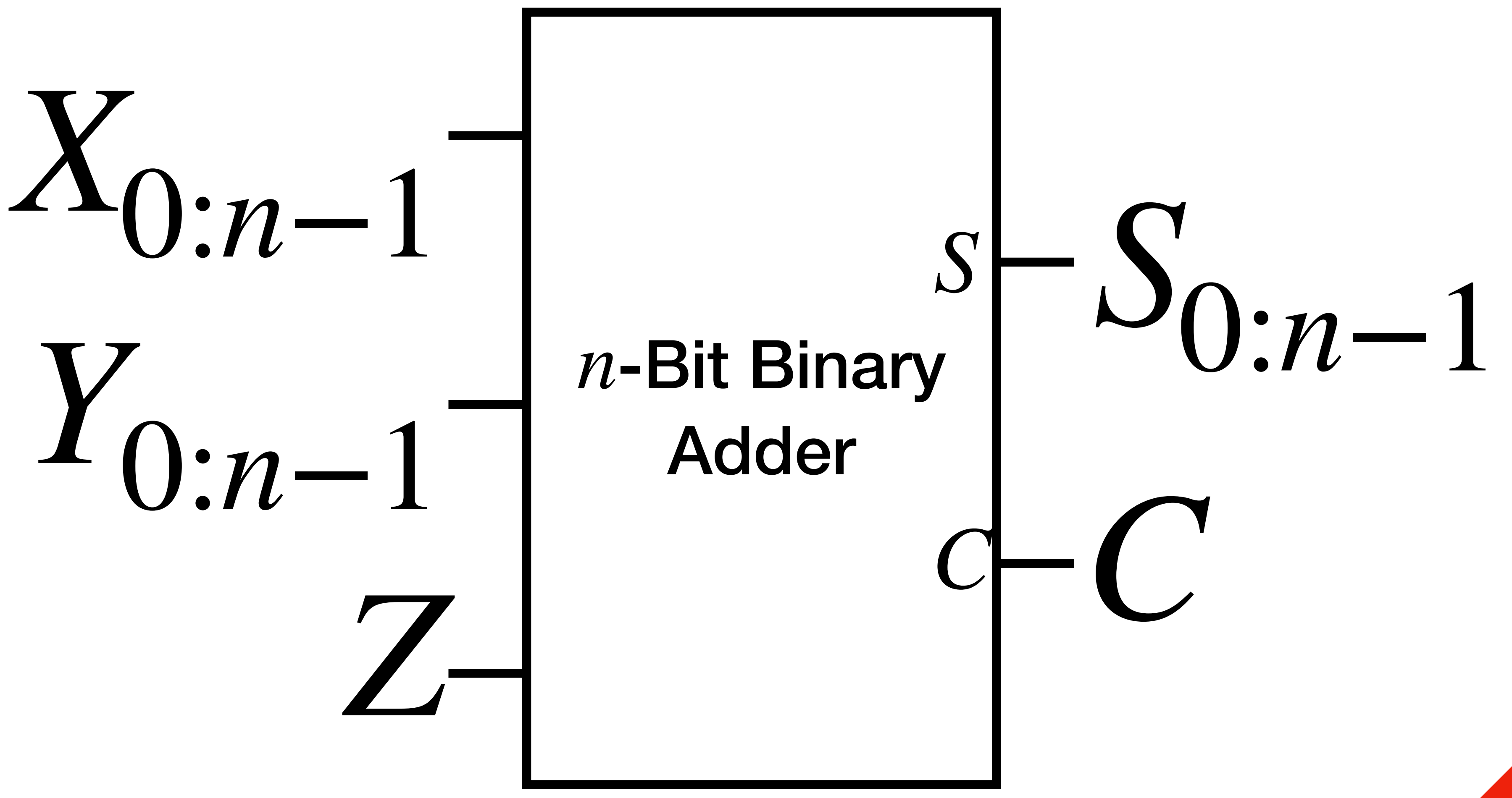


Concept

Adder-Subtractor Unit

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

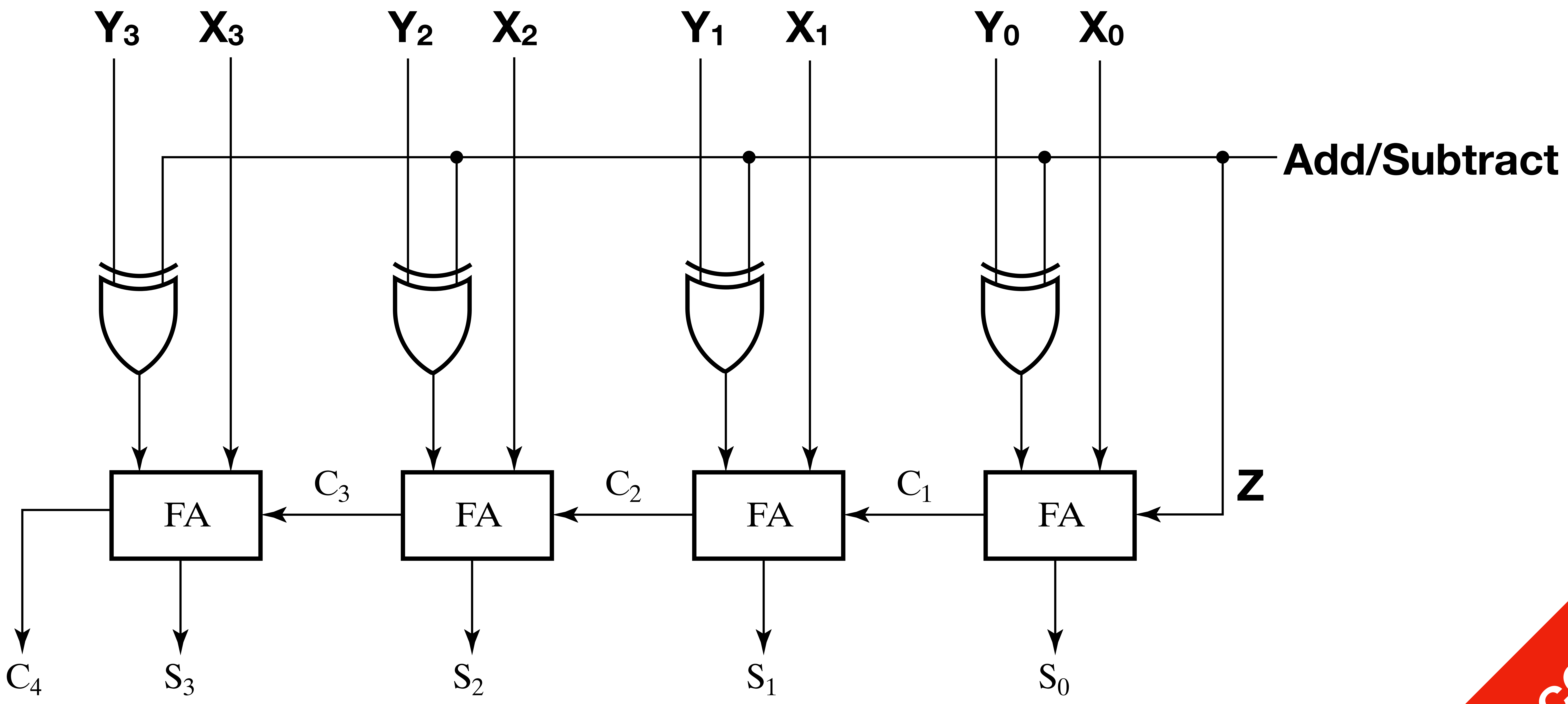
- Addition: $X + Y$
- Subtraction: $X - Y = X + \overline{Y} + 1 - 2^n$
    - We are using n-bit adder, 2^n can be disregarded
    - The plus 1 here can be Z input to the adder



Concept

P2
Subtraction

Adder-Subtractor Unit



Concept

P0-2
Adder-Subtractor

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

Review

Excercises

2s Complement
Subtraction using 2s Complement

Subtraction

Exercise

Subtraction

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

Exercise

Subtraction

Exercise

Subtraction

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

Exercise

Hardware Description Language

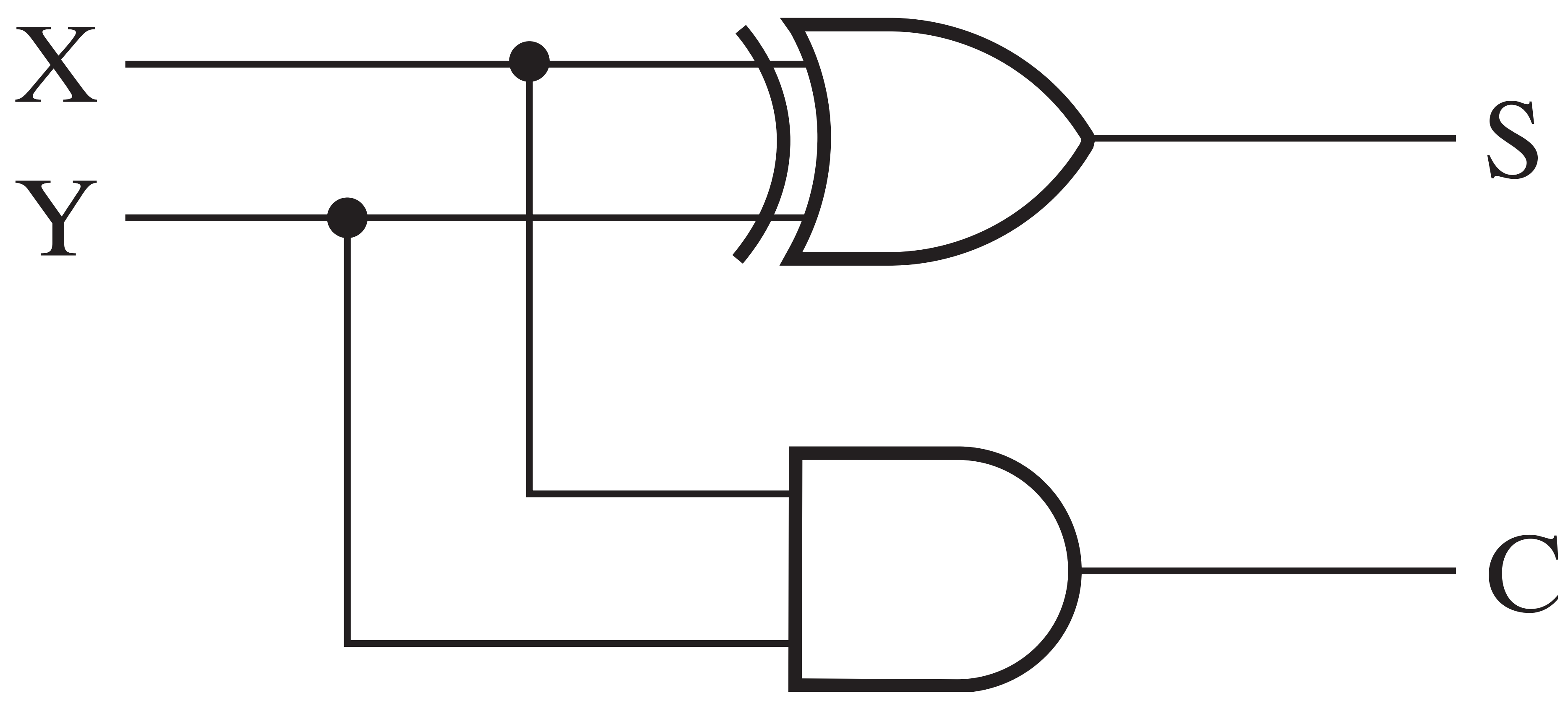
VHDL (VHSIC-HDL): Very High Speed Integrated Circuit Hardware Description Language

Summary

P4
VHDL

Previous: 1-bit Half Adder

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



Practice

Previous: 1-bit Half Adder

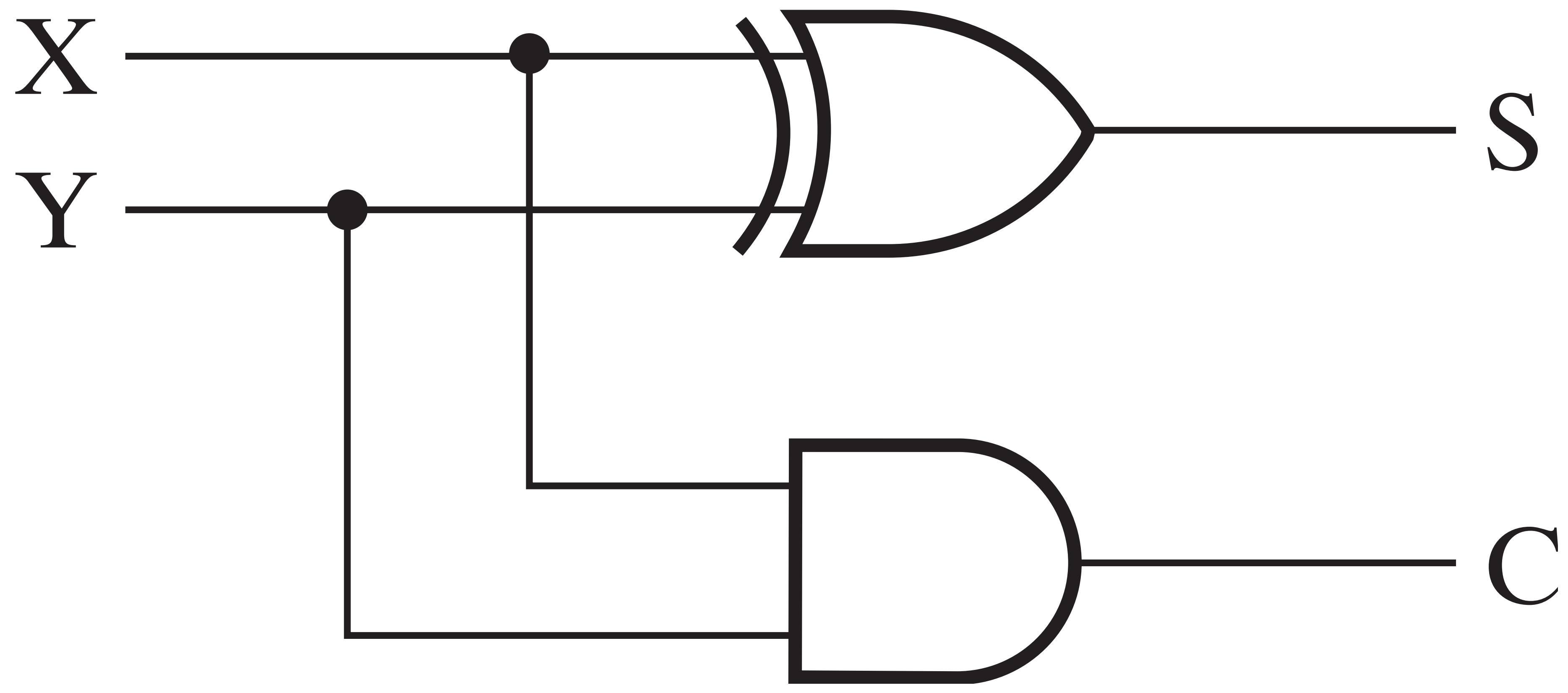
```
architecture arch1 of HalfAdder is

begin

    S <= X XOR Y;

    C <= X AND Y;

end arch1;
```



Practice

P4
VHDL

Today's Tasks

Exercise

P4
VHDL

Today’s Tasks

- 1-bit Half Adder

Exercise

P4
VHDL

Today's Tasks

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

Exercise

P4
VHDL

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)

Exercise

P4
VHDL

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)

Exercise