



13.02.20 21:58

# CSCI 150

## Introduction to Digital and Computer System Design

### Lecture 3: Combinational Logic Design V



Jetic Gū  
2020 Winter Semester (S1)

# Overview

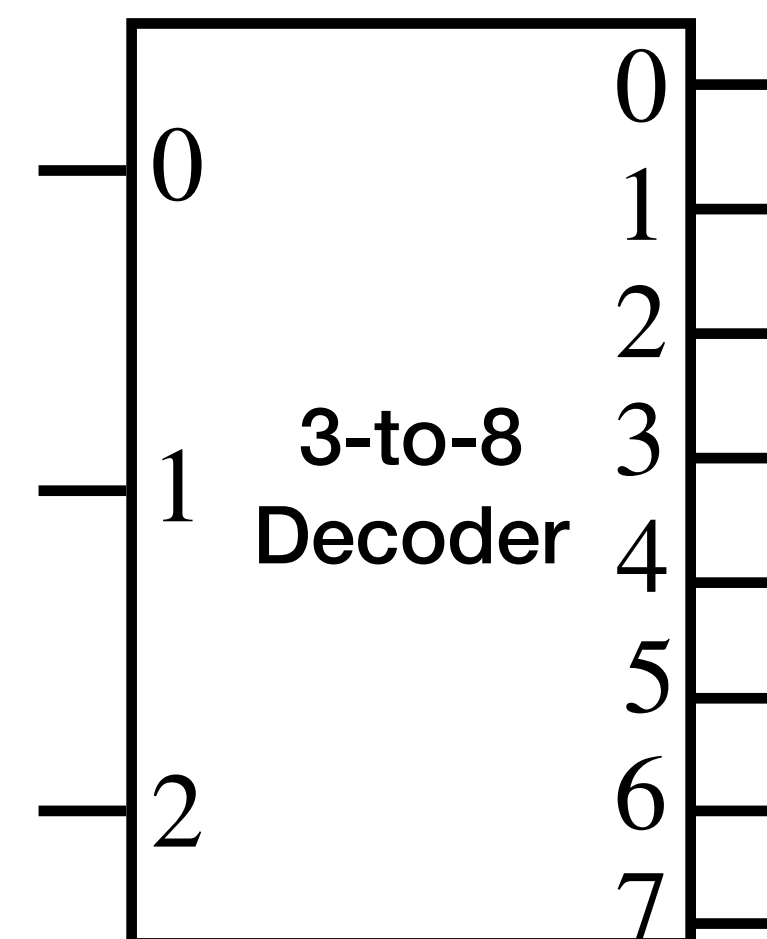
- Focus: Arithmetic Functional Blocks
- Architecture: Combinatory Logical Circuits
- Textbook v4: Ch4 4.2; v5: Ch3 3.9
- Core Ideas:
  1. Binary Adder

# Systematic Design Procedures

1. **Specification:** Write a specification for the circuit
2. **Formulation:** Derive relationship between inputs and outputs of the system  
e.g. using truth table or Boolean expressions
3. **Optimisation:** Apply optimisation, minimise the number of logic gates and literals required
4. **Technology Mapping:** Transform design to new diagram using available implementation technology
5. **Verification:** Verify the correctness of the final design in meeting the specifications

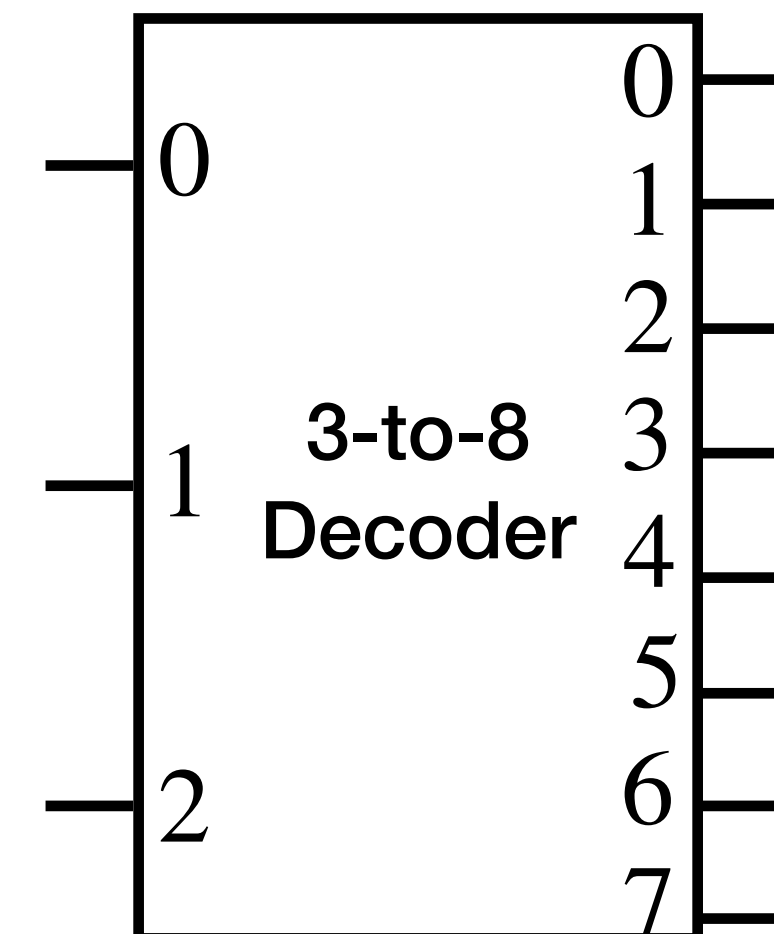
# Functional Components (1)

- Value-Fixing, Transferring, Inverting, Enabler
- Decoder
  - Input:  $A_0A_1 \dots A_{n-1}$
  - Output:  $D_0D_1 \dots D_{2^n-1}$ ,  $D_i = m_i$



# Functional Components (2)

- Encoder
  - Input:  $m_0, \dots, m_{2^n-1}$  with only one positive value
  - Output:  $A_0, \dots, A_{n-1}$
  - Priority Encoder: validity, priority output
- Multiplexer
  - Switching between multiple input channels

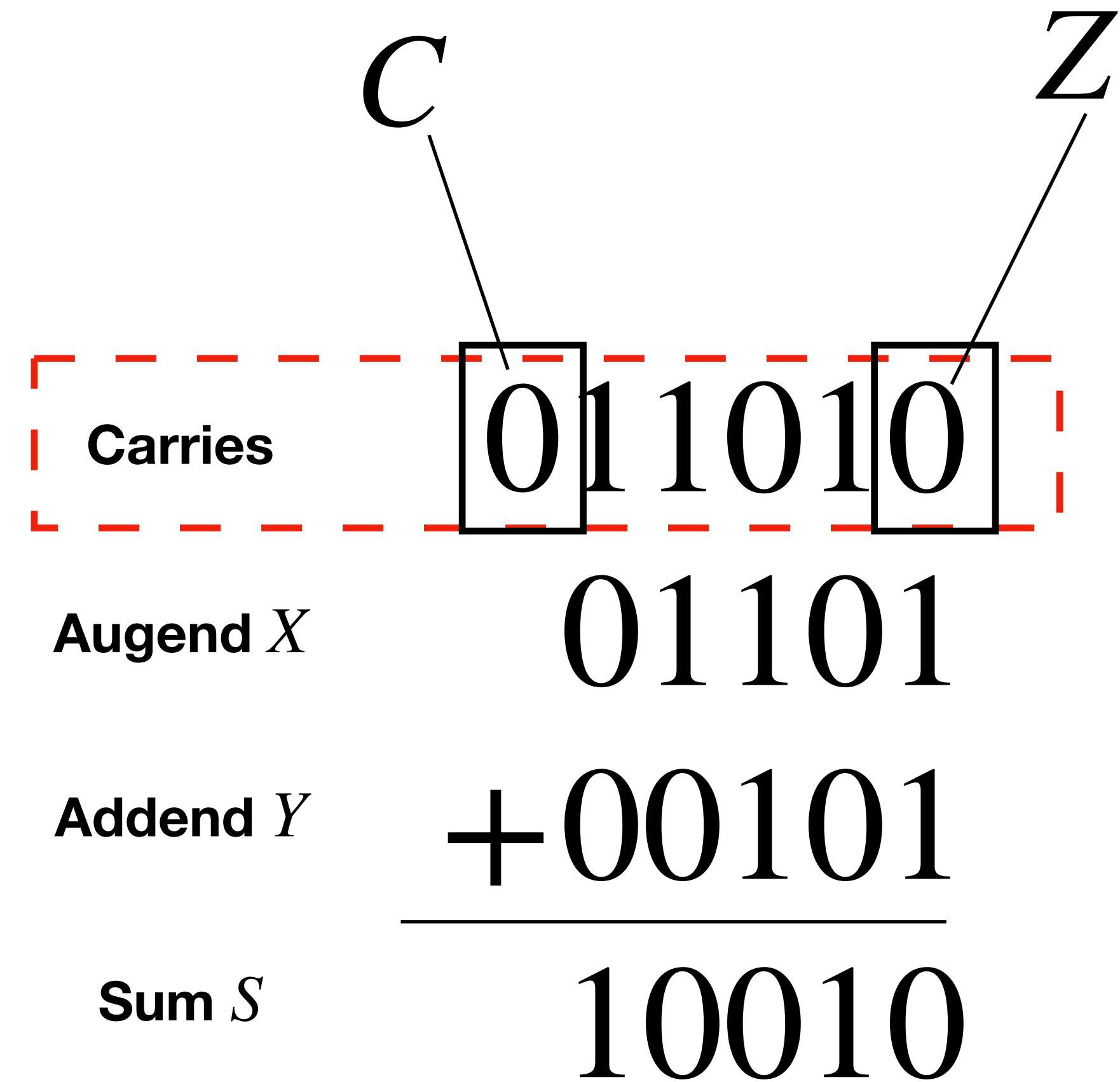


# Binary Adder

The good stuff begins

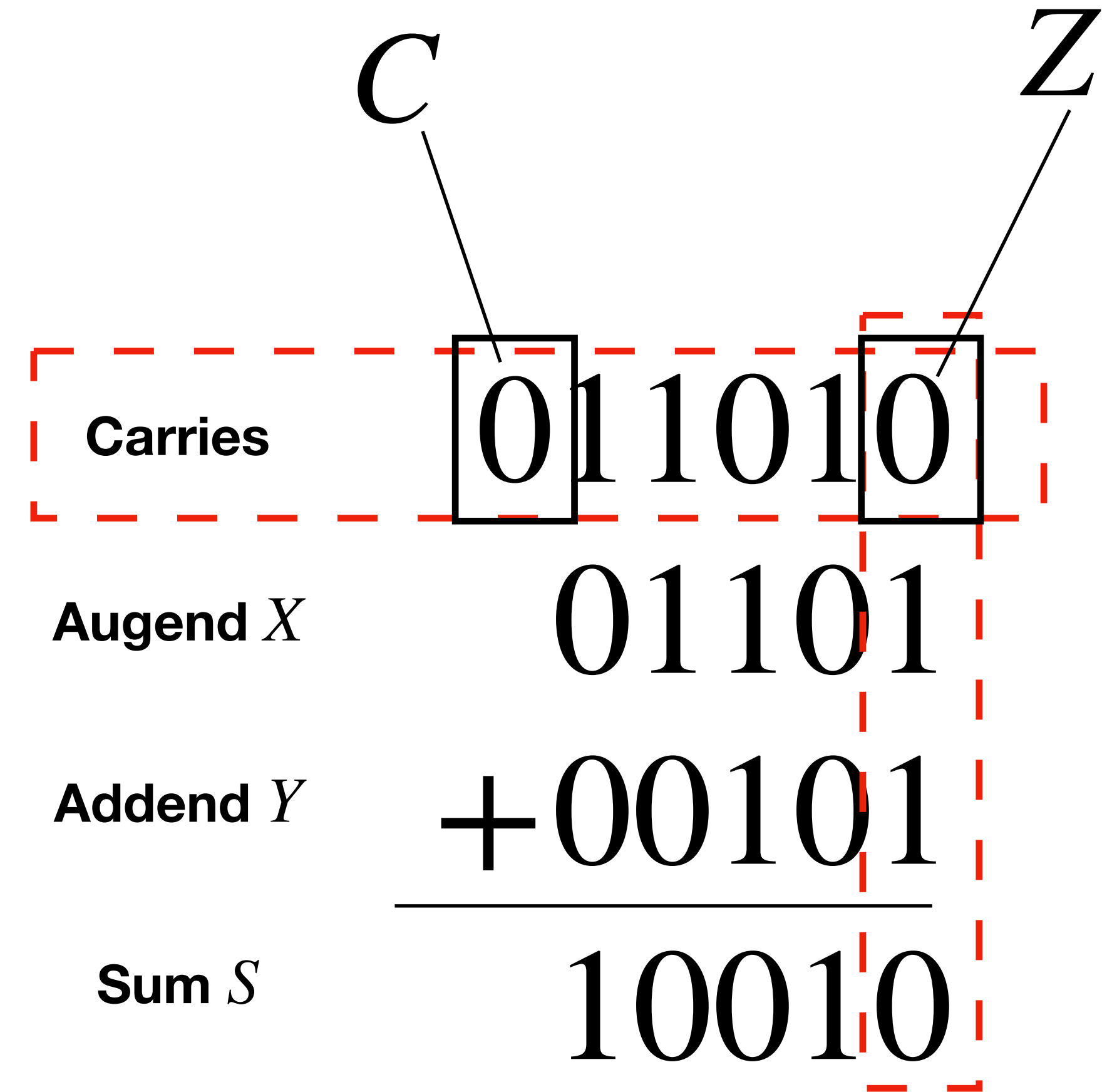
# Binary Adder

- Perform binary addition
- 1-bit Half adder  
input  $X, Y$ ;  
output  $S, C$
- Full adder  
input vectors  $X, Y$ , and single-bit  $Z$ ;  
output vector  $S$  and single-bit  $C$
- Remember what we did before?



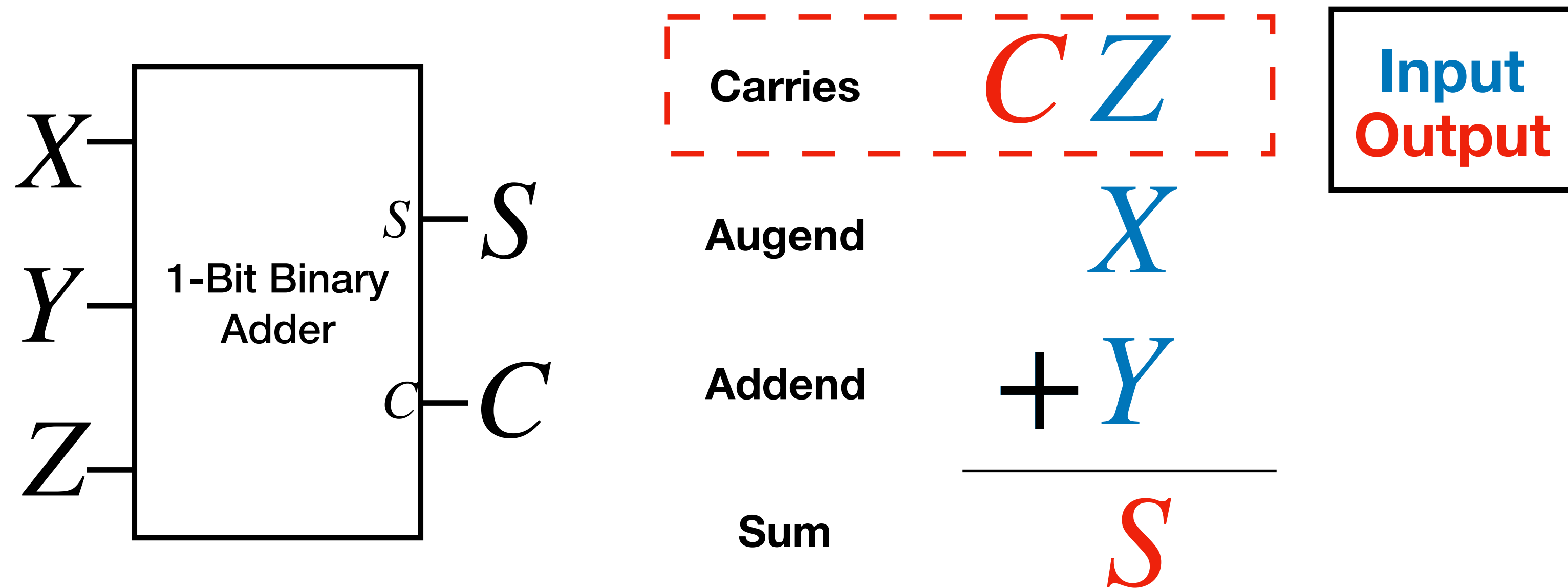
# Binary Adder

- Perform binary addition
- 1-bit Half adder  
input  $X, Y$ ;  
output  $S, C$
- Full adder  
input vectors  $X, Y$ , and single-bit  $Z$ ;  
output vector  $S$  and single-bit  $C$
- Remember what we did before?





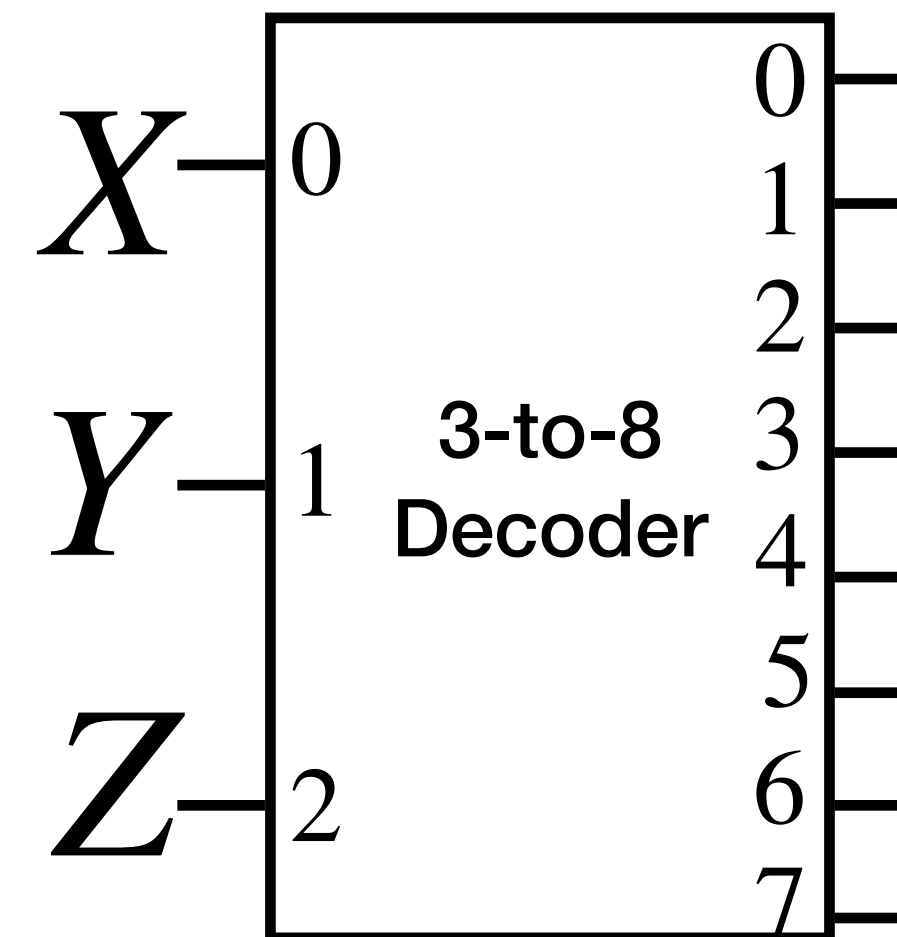
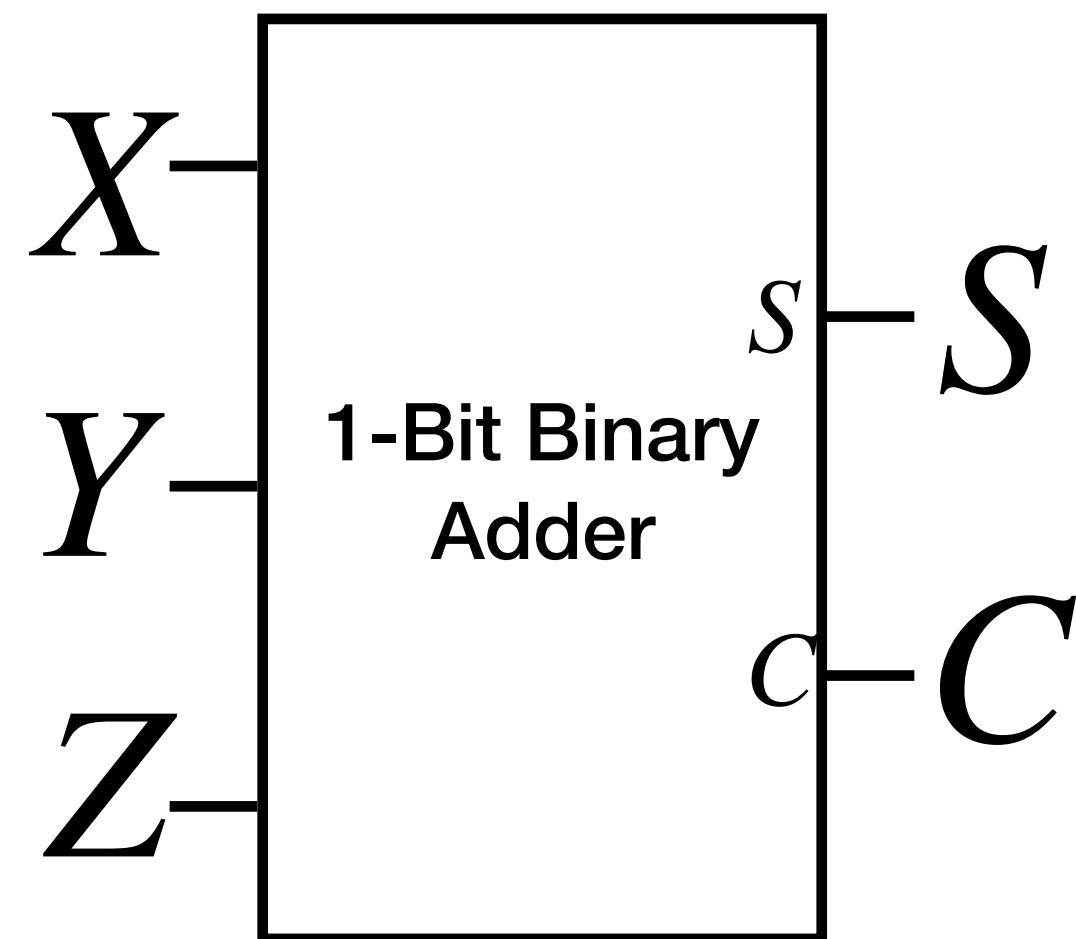
# 1-bit Binary Adder



Full Adder

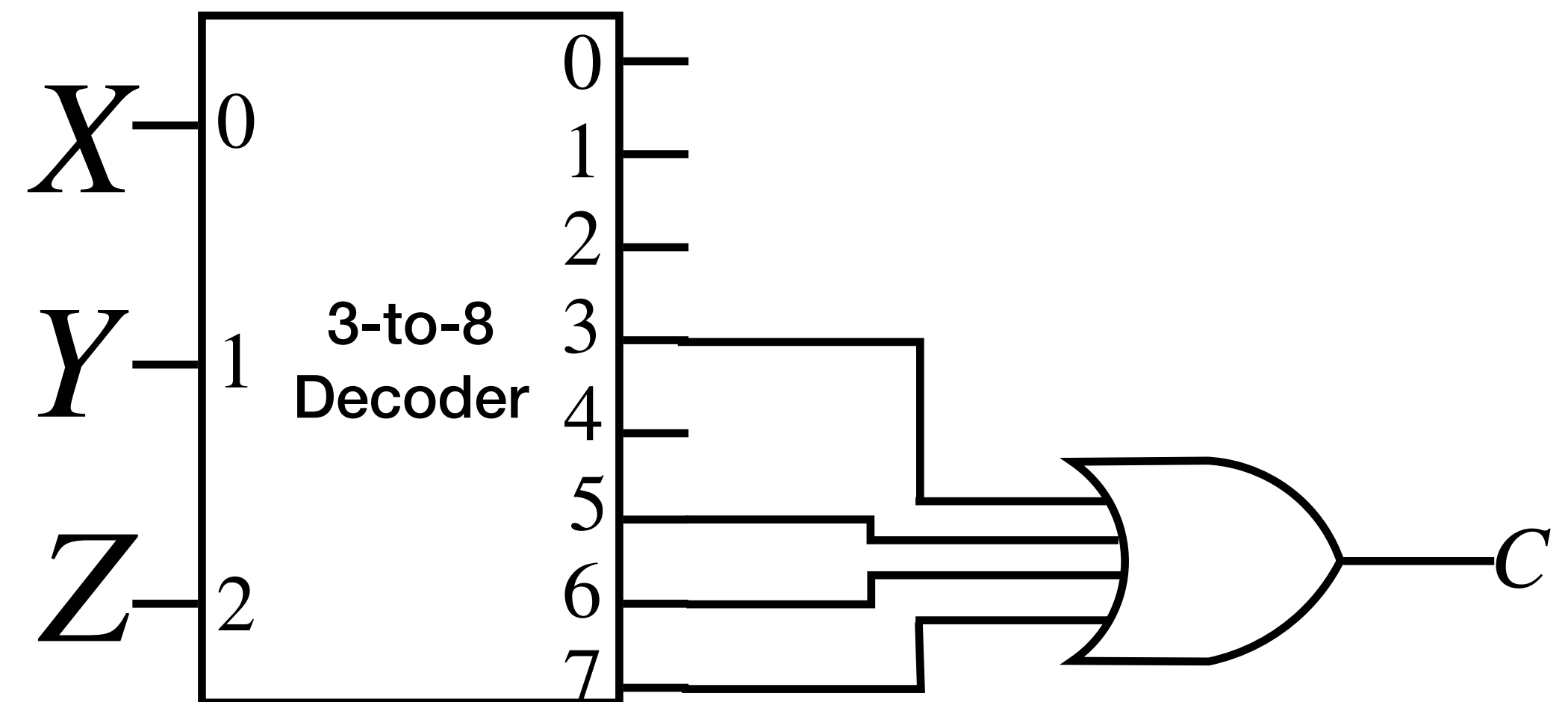
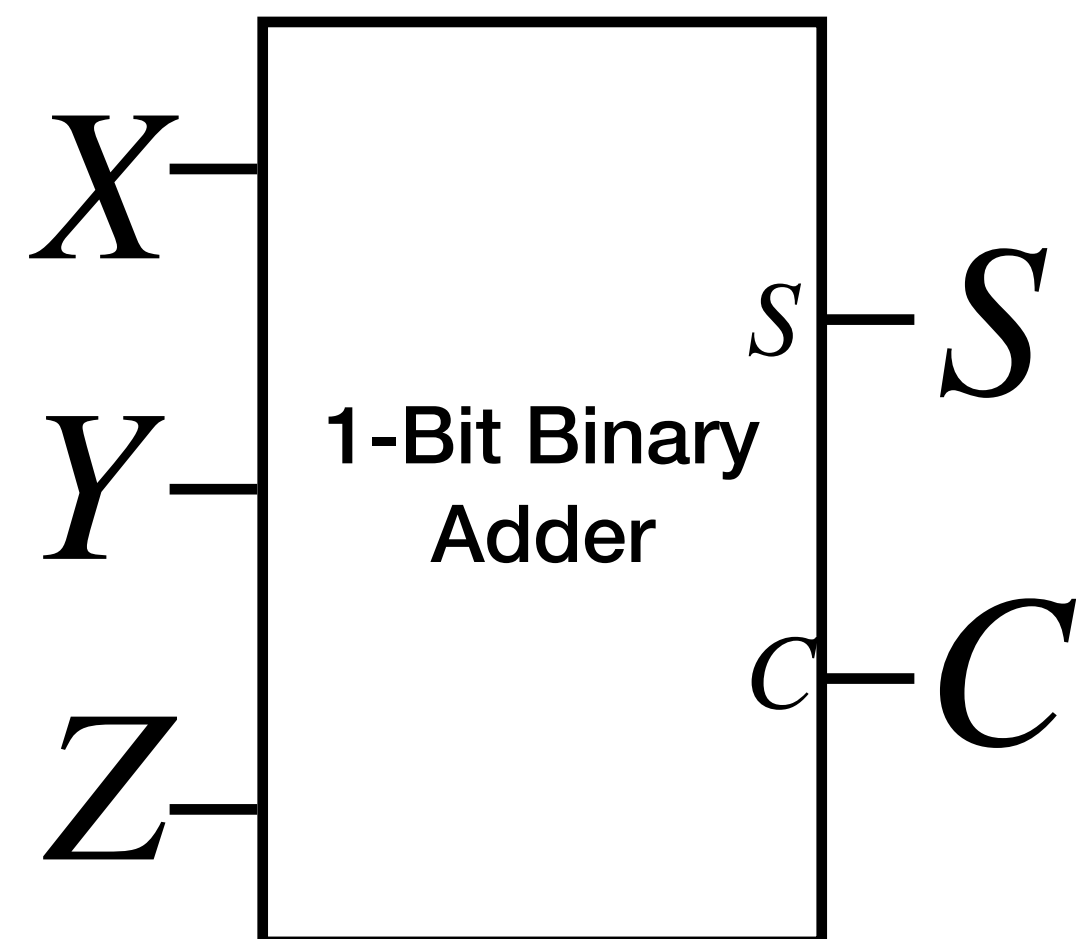
Review

# 1-bit Binary Adder



Full Adder

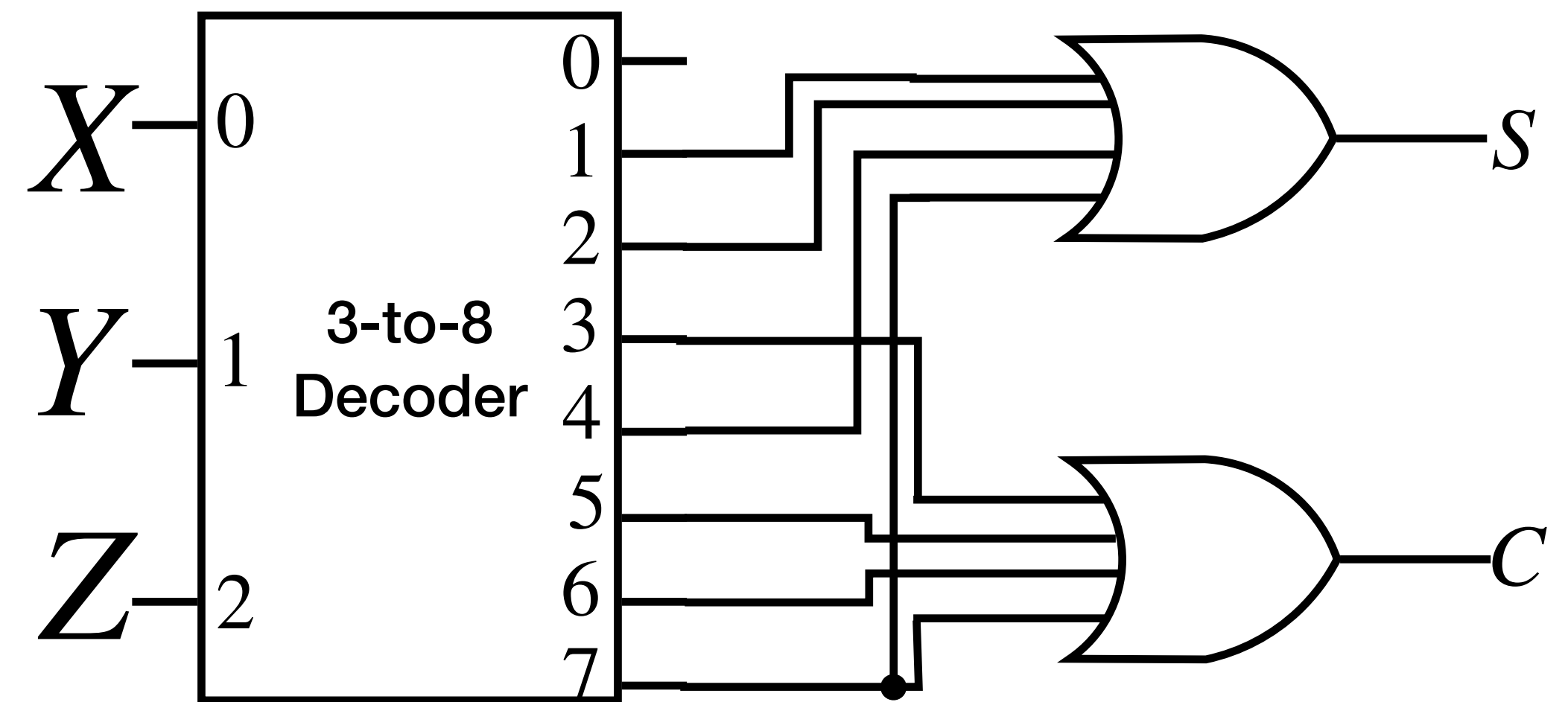
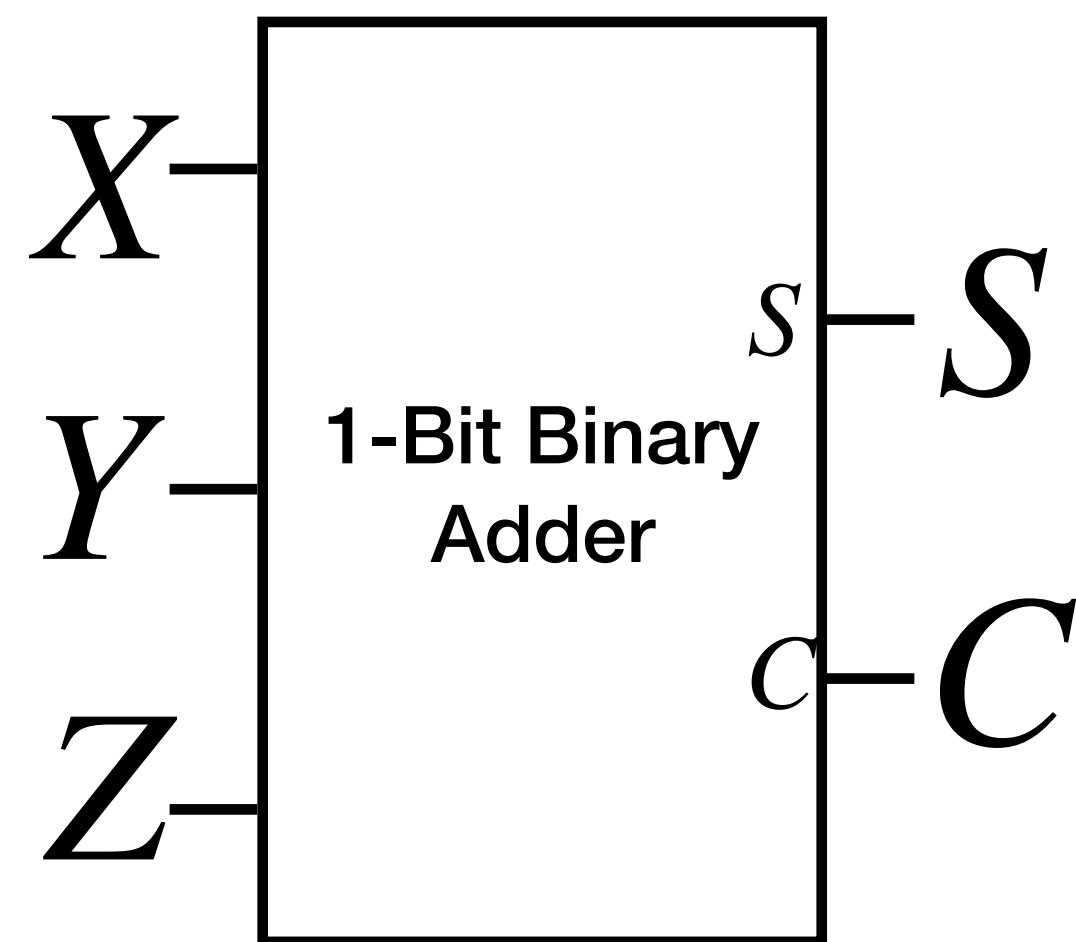
# 1-bit Binary Adder



Full Adder

Review

# 1-bit Binary Adder

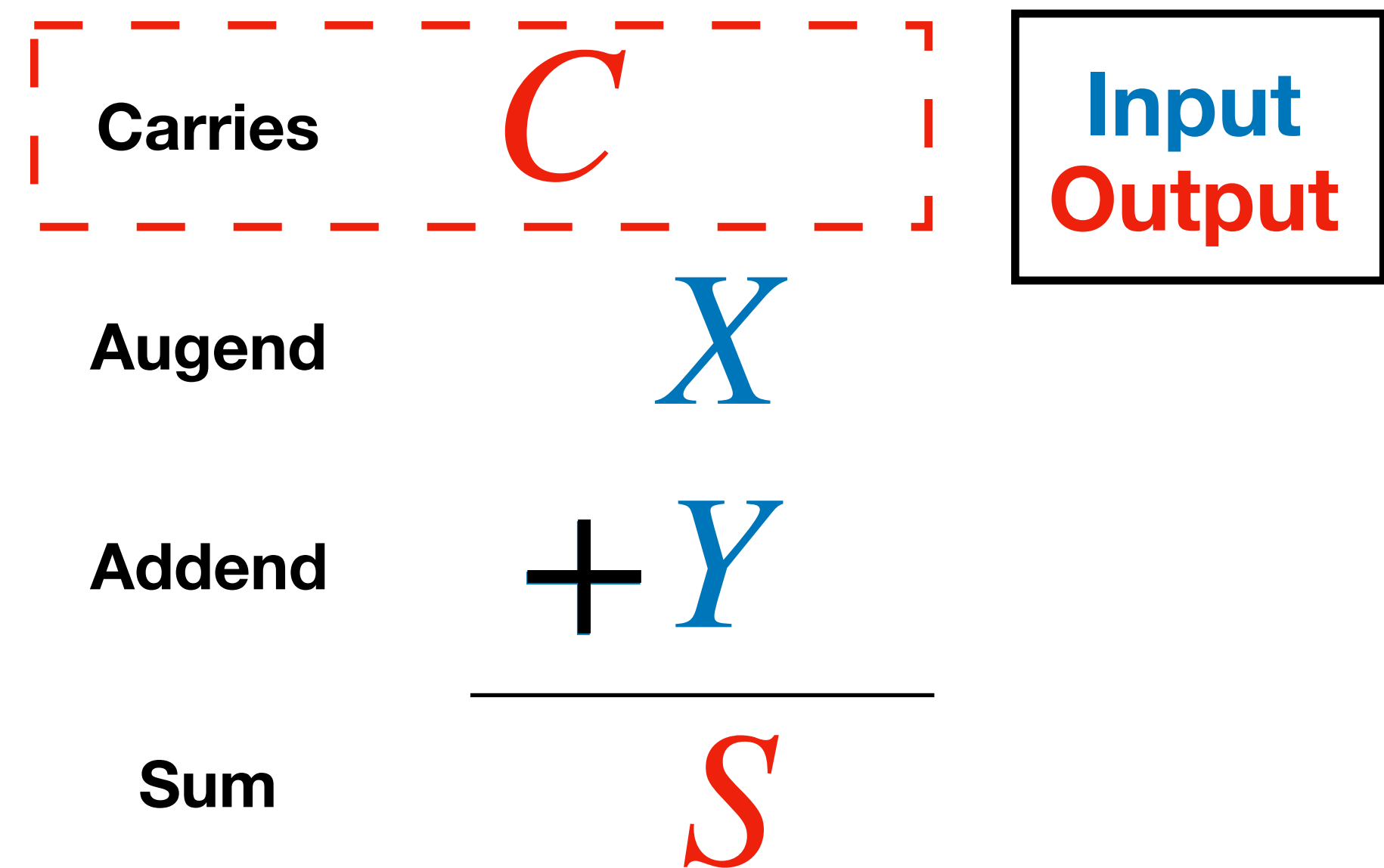


Full Adder

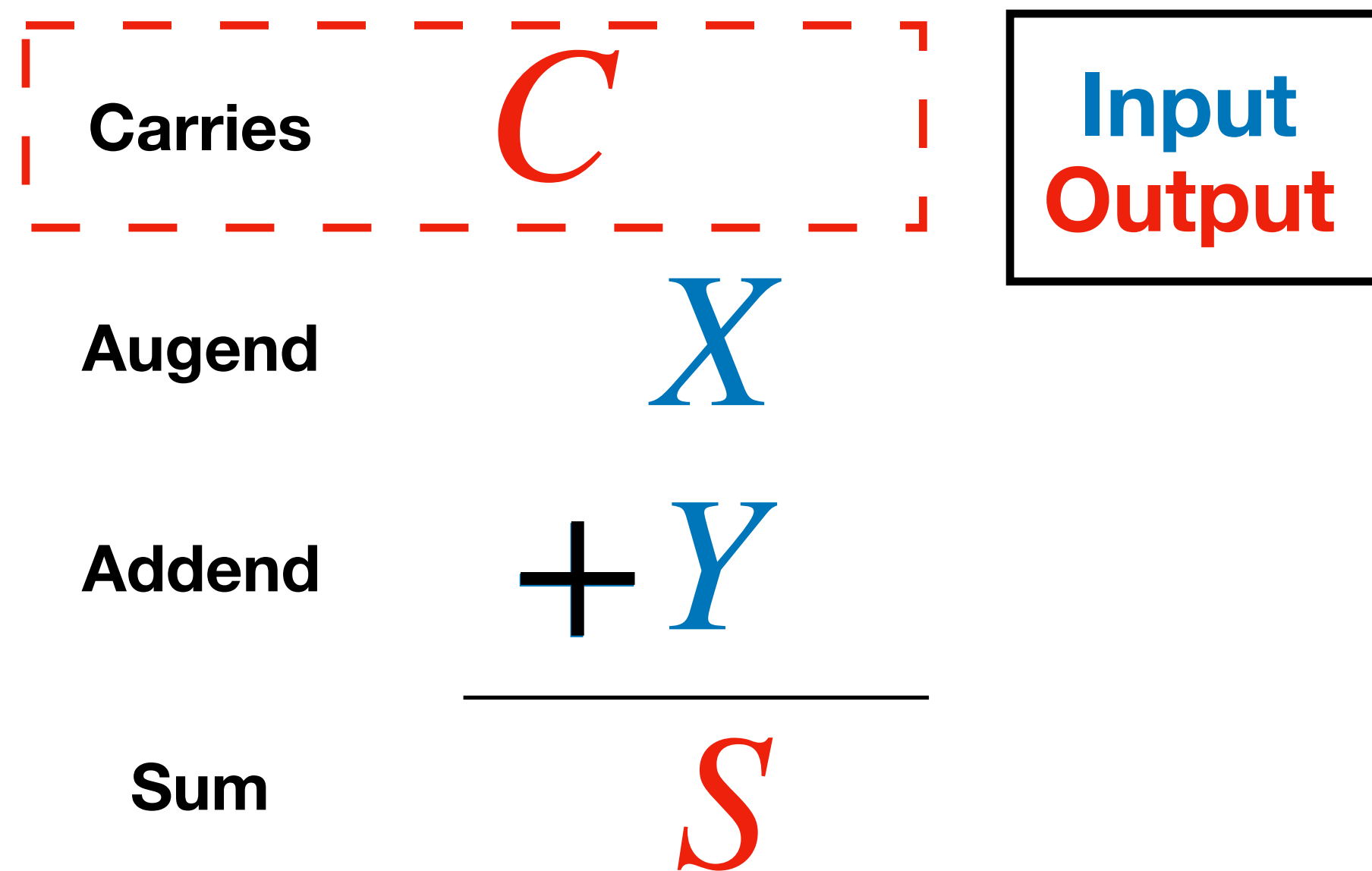
Review

# 1-bit Half Adder

- Half adder  
input  $X$ ,  $Y$   
output  $S$ ,  $C$



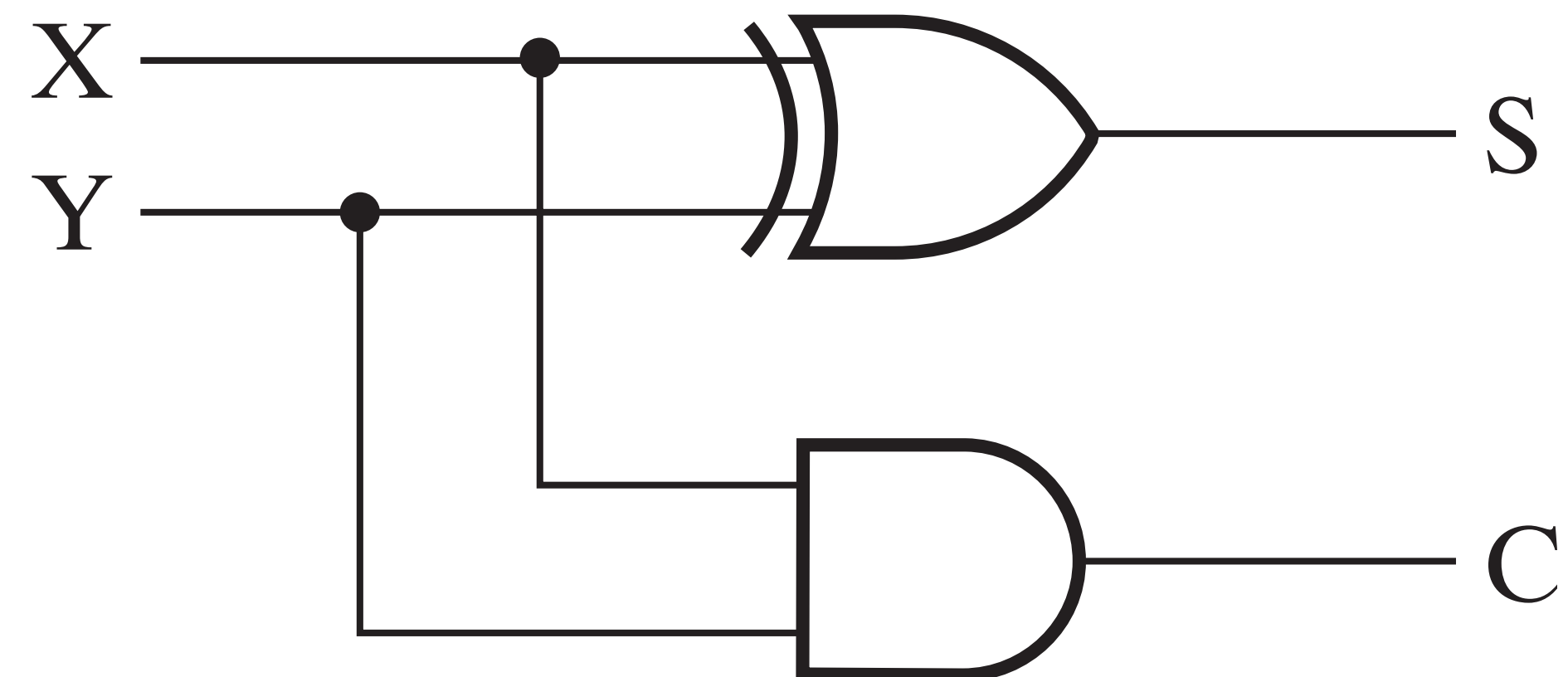
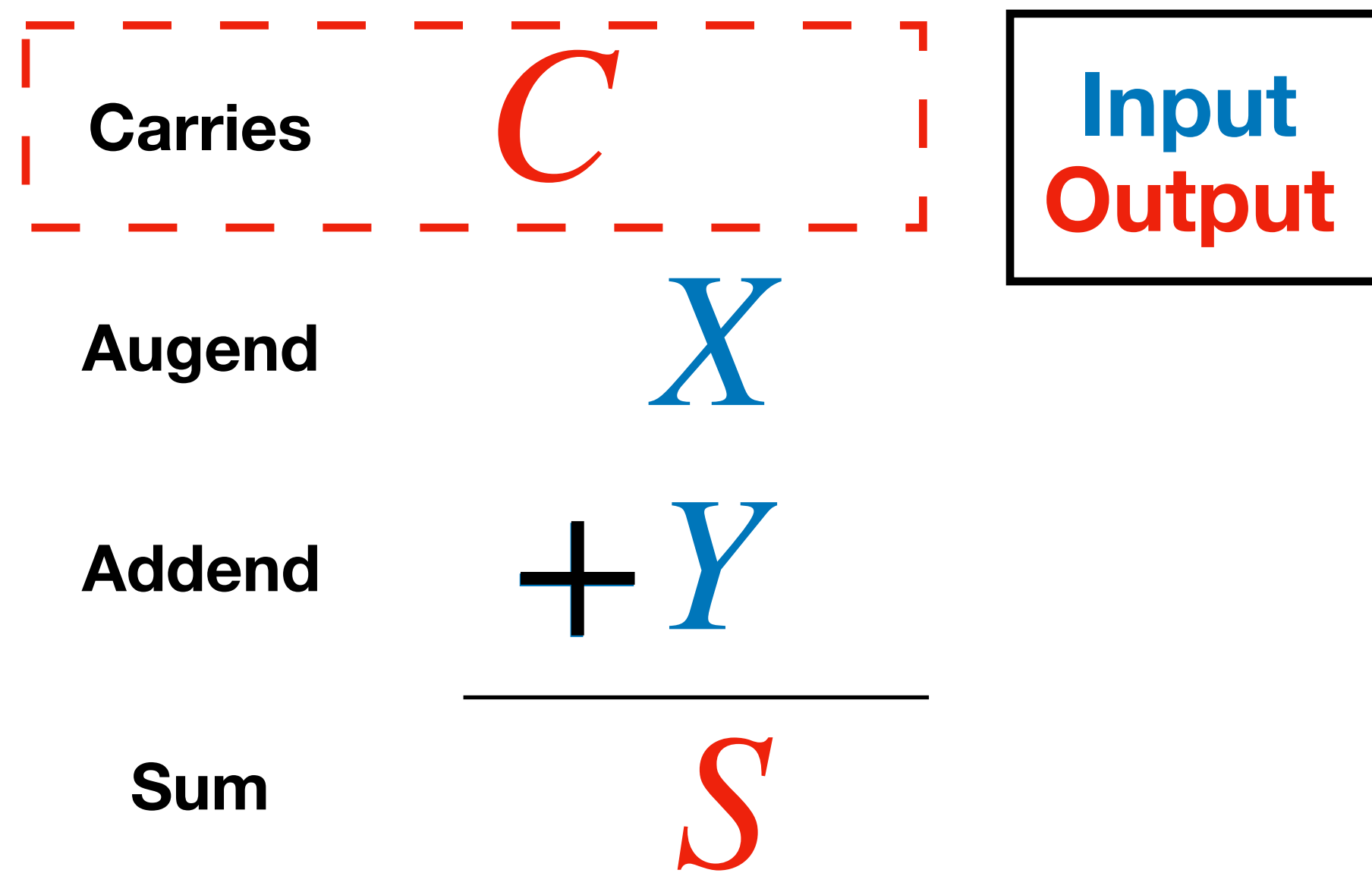
# 1-bit Half Adder



X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

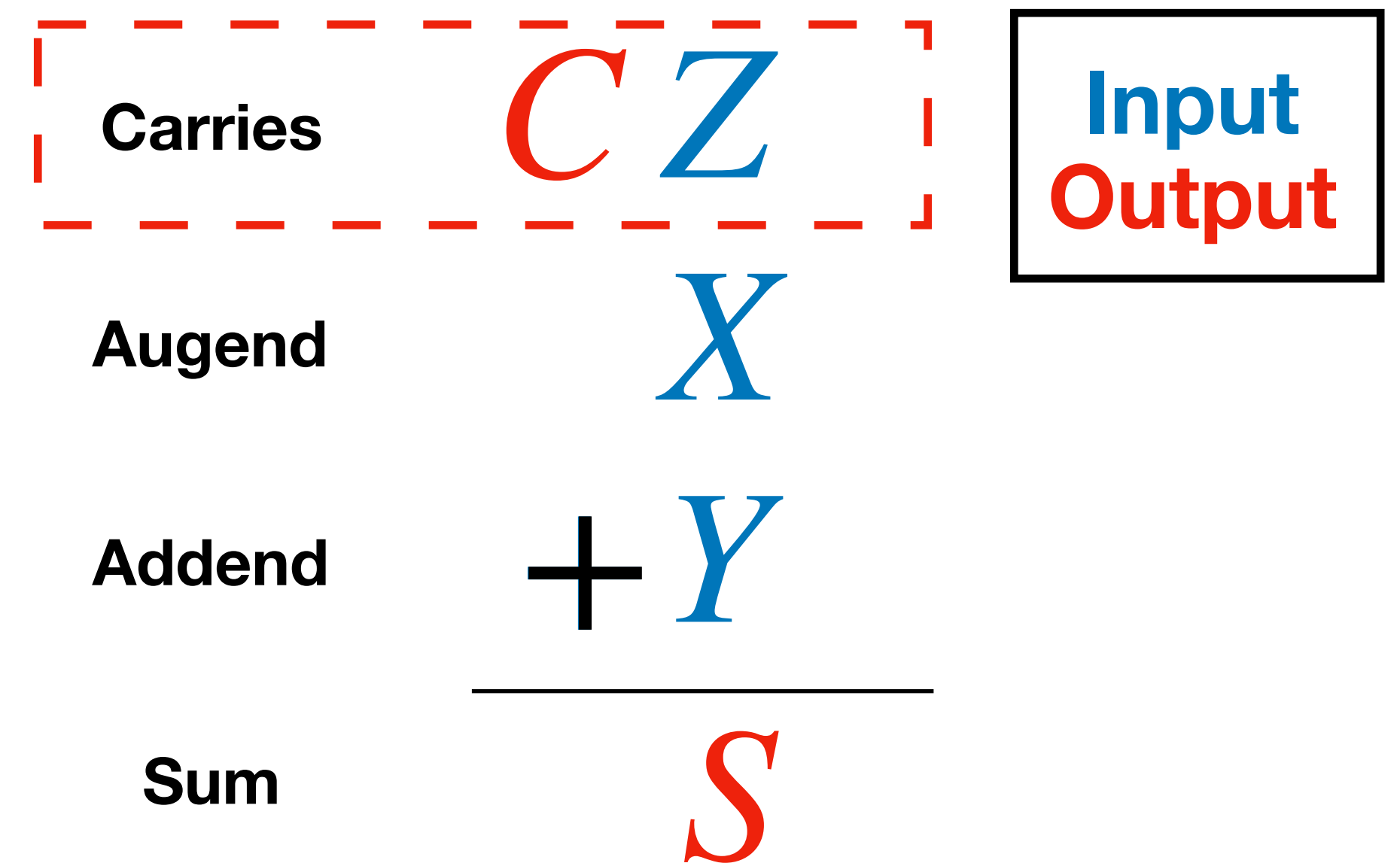
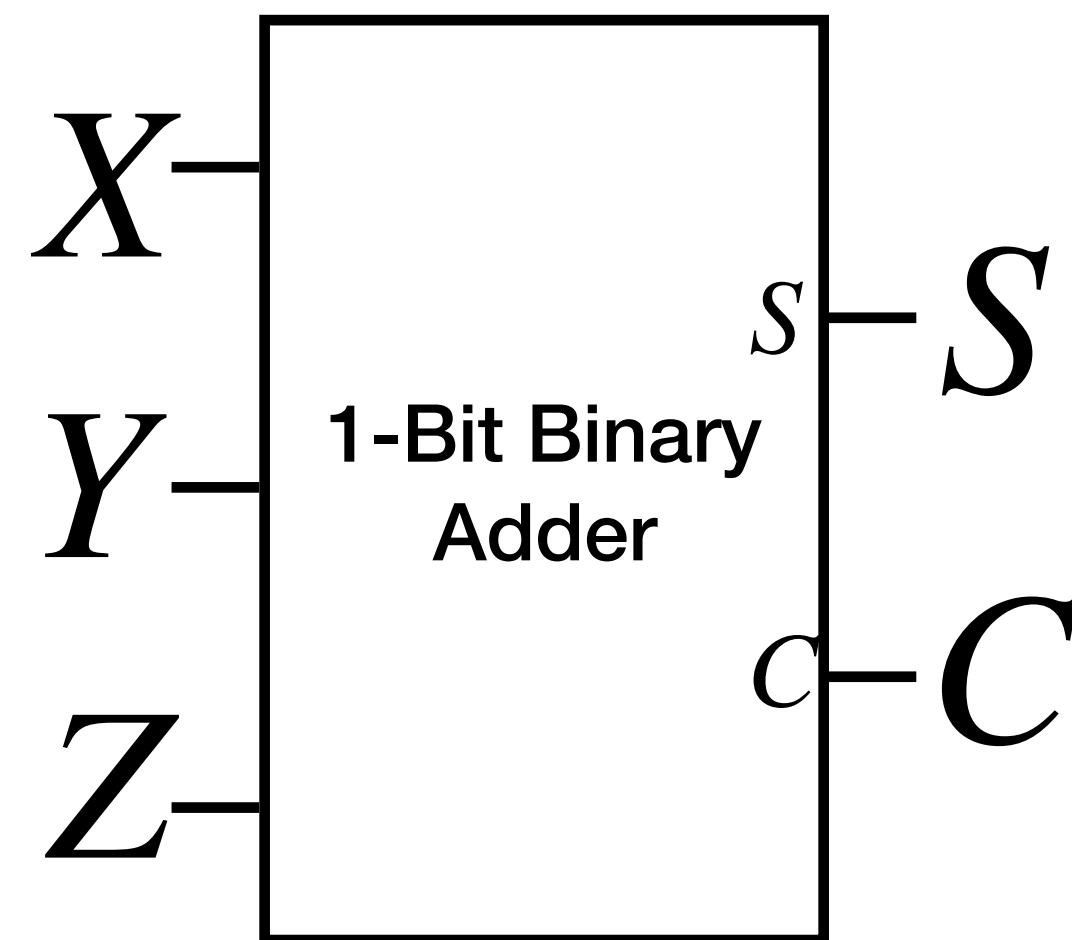
Concept

# 1-bit Half Adder



# 1-bit Full Addder

- Full addder  
input  $X$ ,  $Y$ ,  $Z$ ;  
output  $S$ ,  $C$

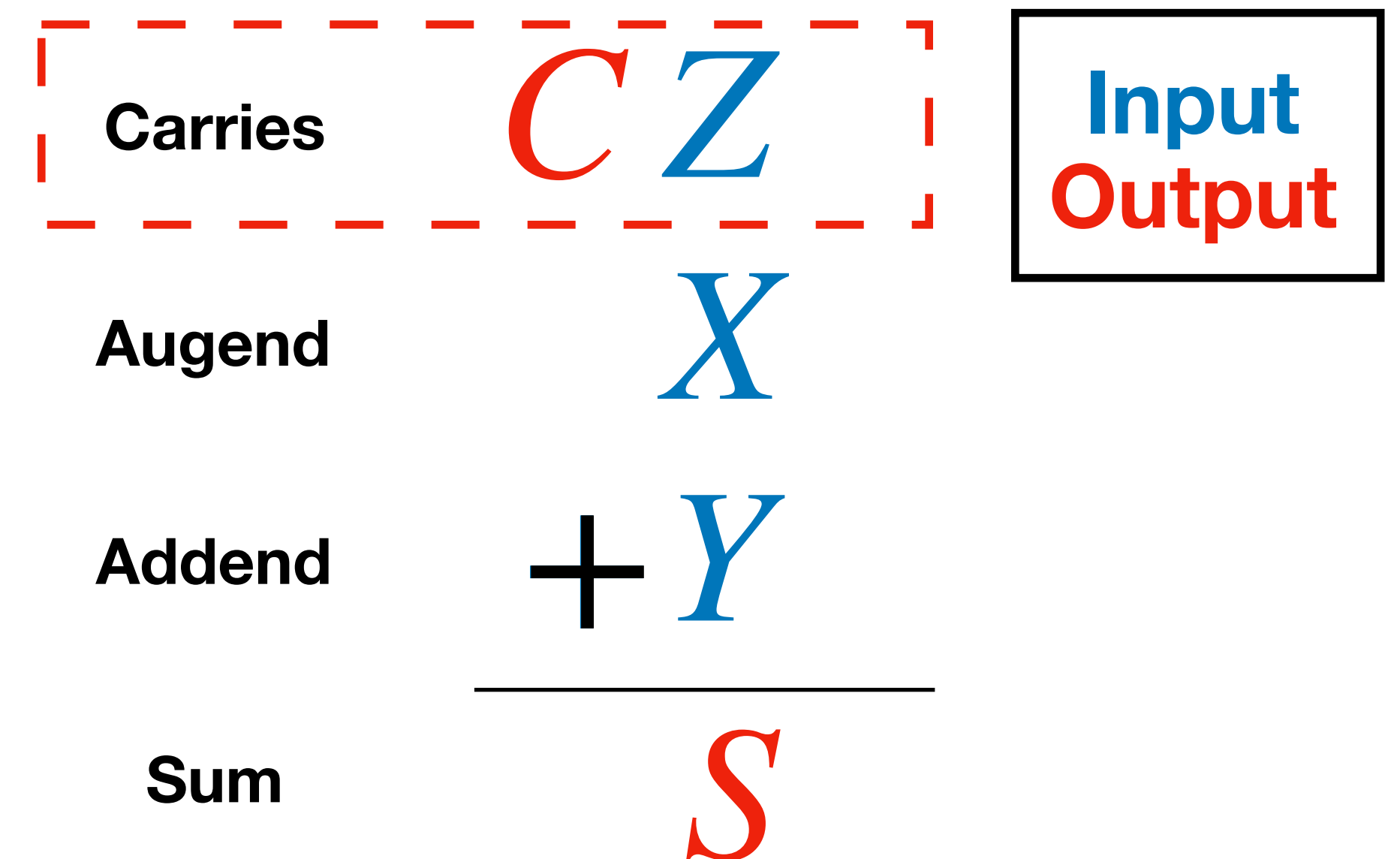




# 1-bit Full Addder

- Full addder  
input  $X, Y, Z$ ;  
output  $S, C$
- Half addder1  
input  $X, Y$   
output  $S', C'$
- Half addder2  
input  $S', Z$   
output  $S, C''$

$$C = C' + C''$$

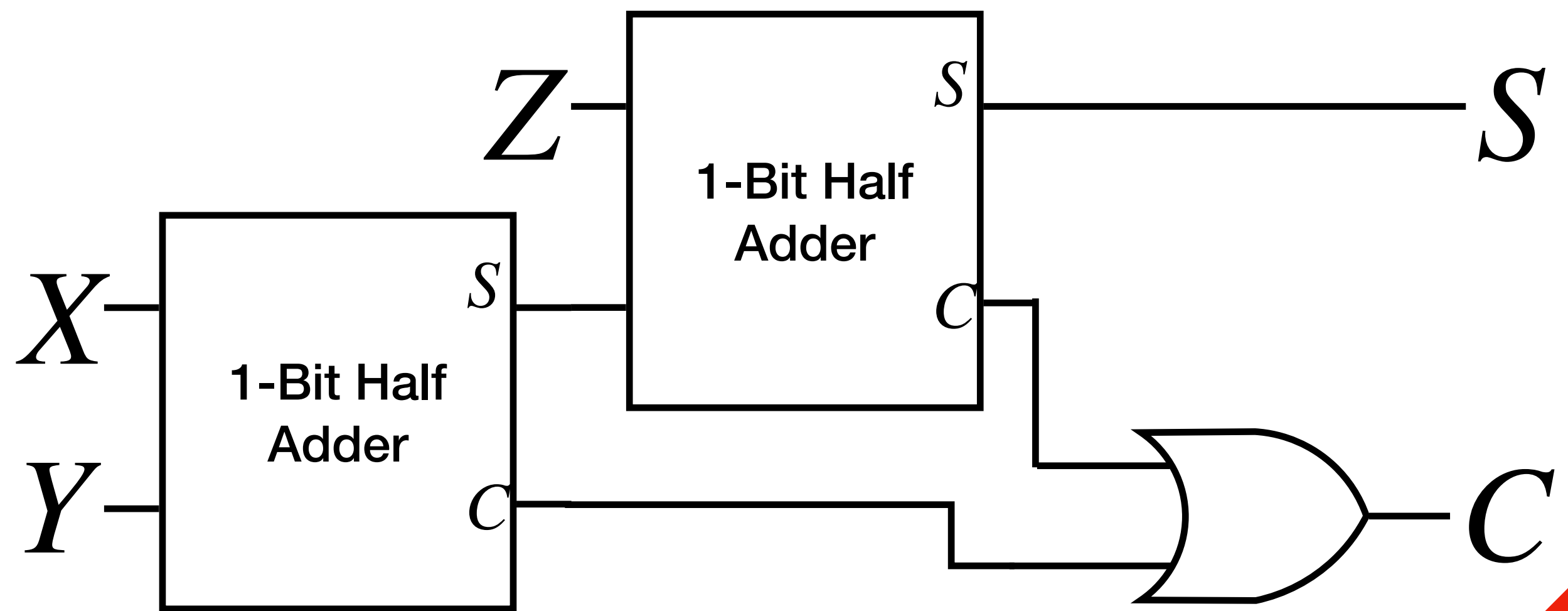


# 1-bit Full Adder

- Half adder1  
input  $X, Y$   
output  $S', C'$
- Half adder2  
input  $S', Z$   
output  $S, C''$

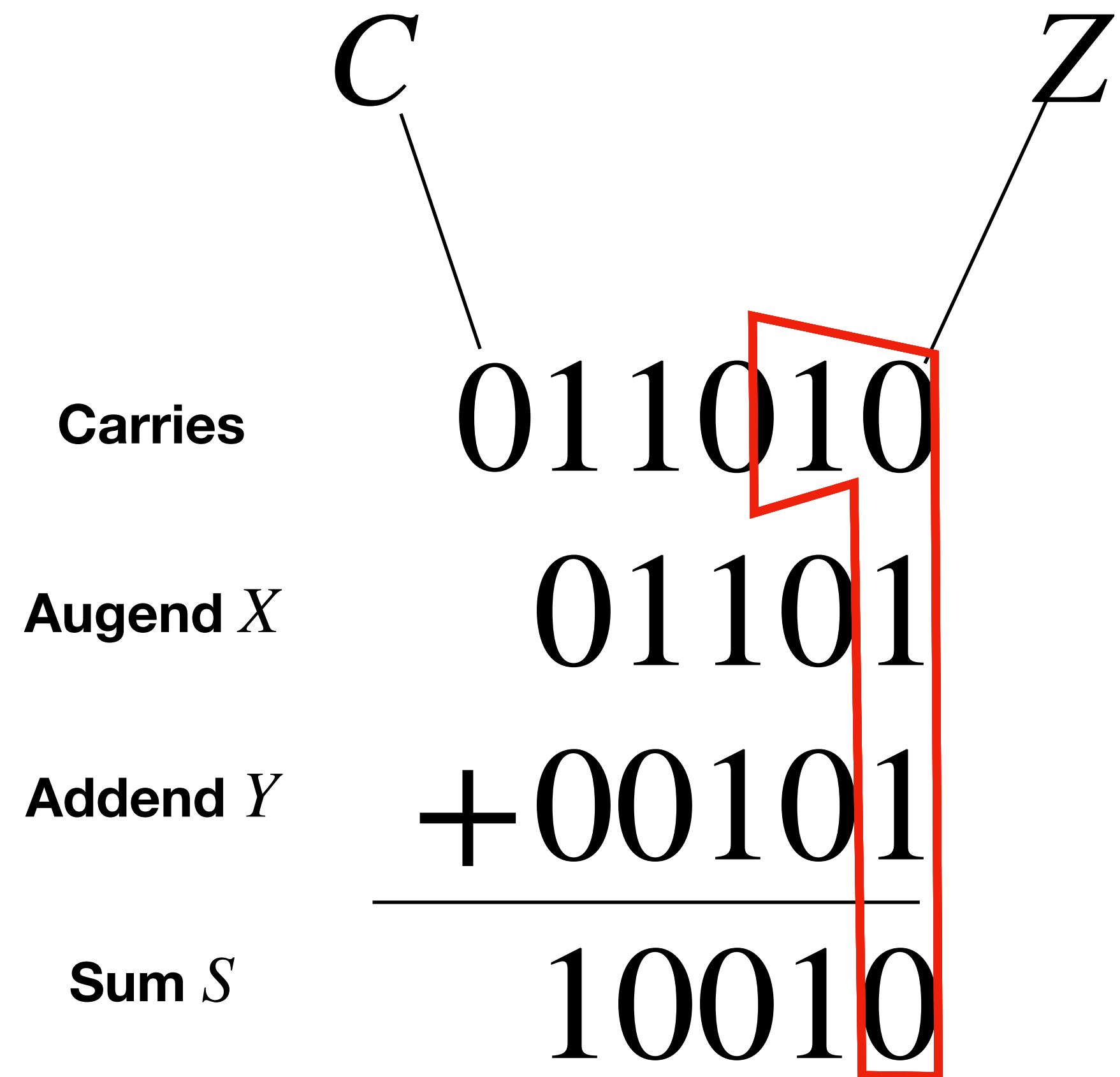
- Full adder  
input  $X, Y, Z$ ;  
output  $S, C$

$$C = C' + C''$$



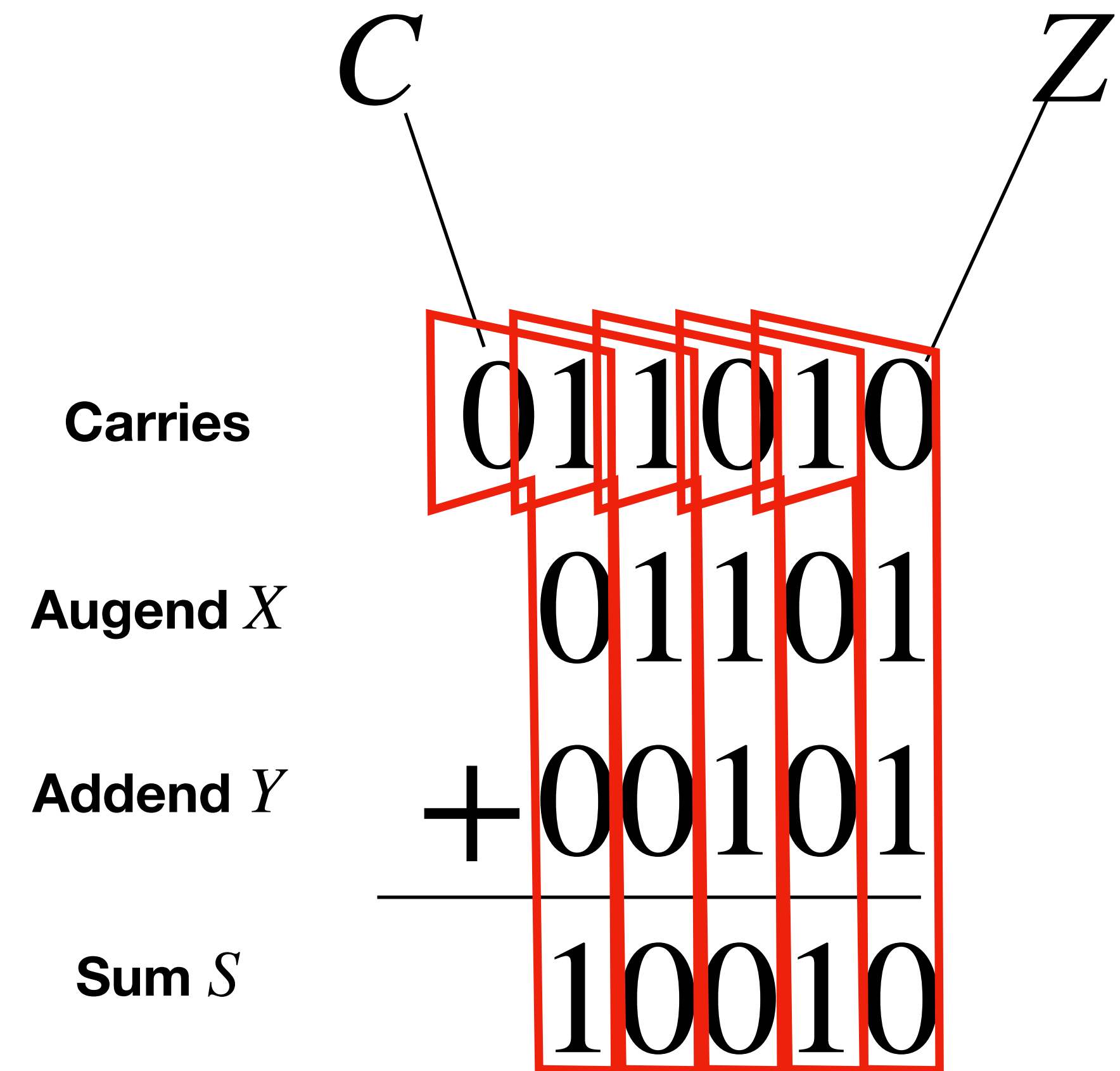
# n-bit Full Addder

- Full addder  
input vectors  $X$ ,  $Y$ , and single-bit  $Z$ ;  
output vector  $S$  and single-bit  $C$

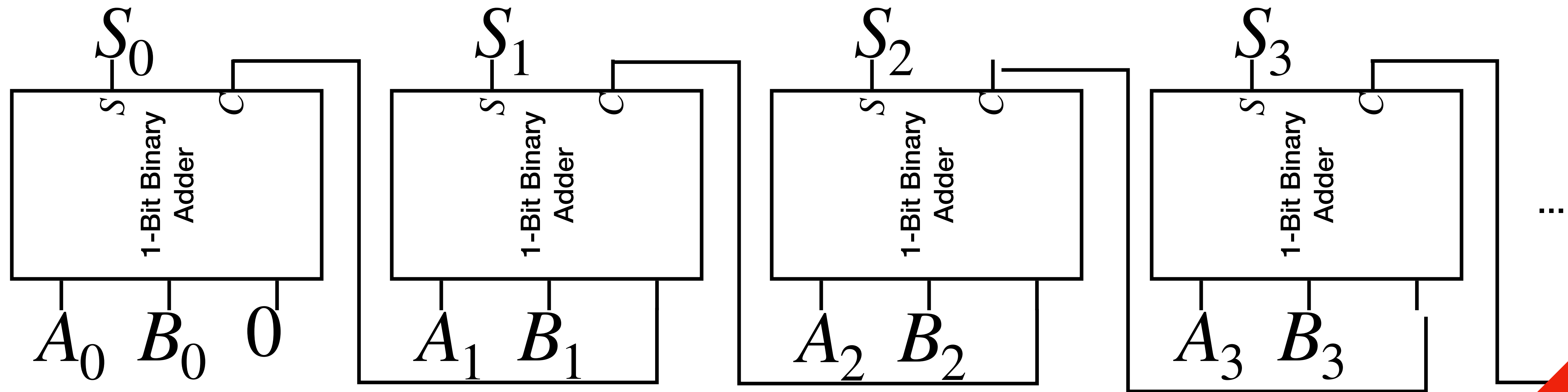


# n-bit Full Adder

- Full adder  
input vectors  $X$ ,  $Y$ , and single-bit  $Z$ ;  
output vector  $S$  and single-bit  $C$

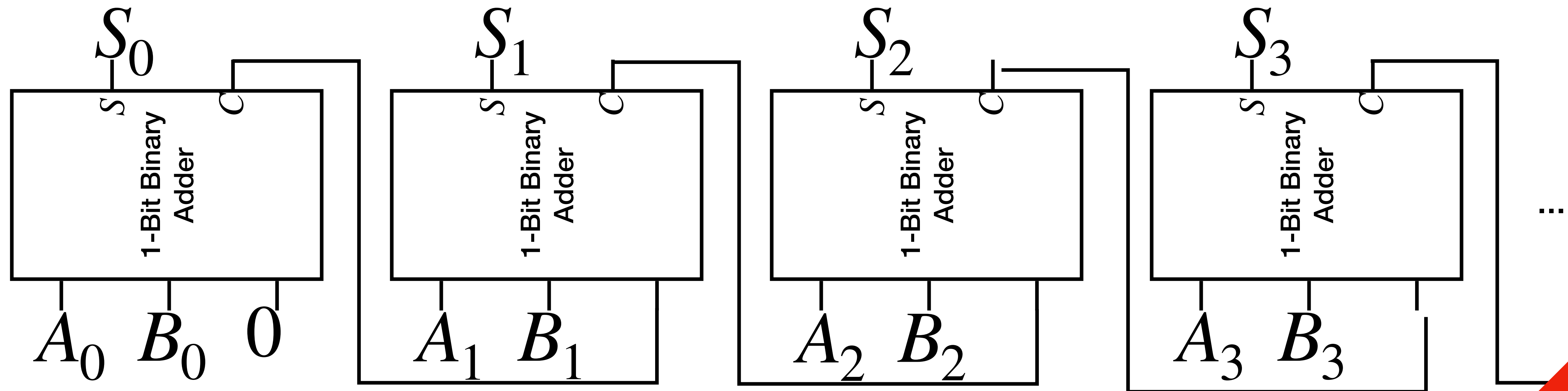


# n-bit Full Adder



# n-bit Full Adder

- Ripple Carry Adder



# Preview

- Binary Adder ✓
- Binary Subtraction (using complements)
- Adder-Subtractor Unit
- Contraction, Incrementing, Multiplication (by constant), Division (by constant)
- VHDL