

Digital Logic

Abstract

The basics of Boolean algebra will be introduced, starting from simple combinational logic functions and basic sequential logic circuits such as latches and flip-flops. Based on these fundamentals, more complex logic structures such as decoders, adders, registers, and memories will be constructed and their performance analyzed. FET Transistor basics will be reviewed and circuits introduced that demonstrate implementations of the basic logic elements in CMOS technology. The logic functions used as the architectural basis for programmable devices used in spaceborne electronics will be examined and analyzed.

Some common design problems found in flight circuits will be introduced.

Logic Outline

- Introduction
- Basic Logic
- Hardware
- Combinational Logic
- Sequential Logic

Basic Digital Logic

Logic Values

- For real binary hardware
 - Either a ‘1’ or a ‘0’
 - “Most of the time”
- Other types of hardware can be have more than two values
- Simulation can use a multi-value logic system
 - VHDL Std_Logic uses a nine-valued logic

Binary Hardware Logic Values

- Logic '1'
- Logic '0'
- Undefined value

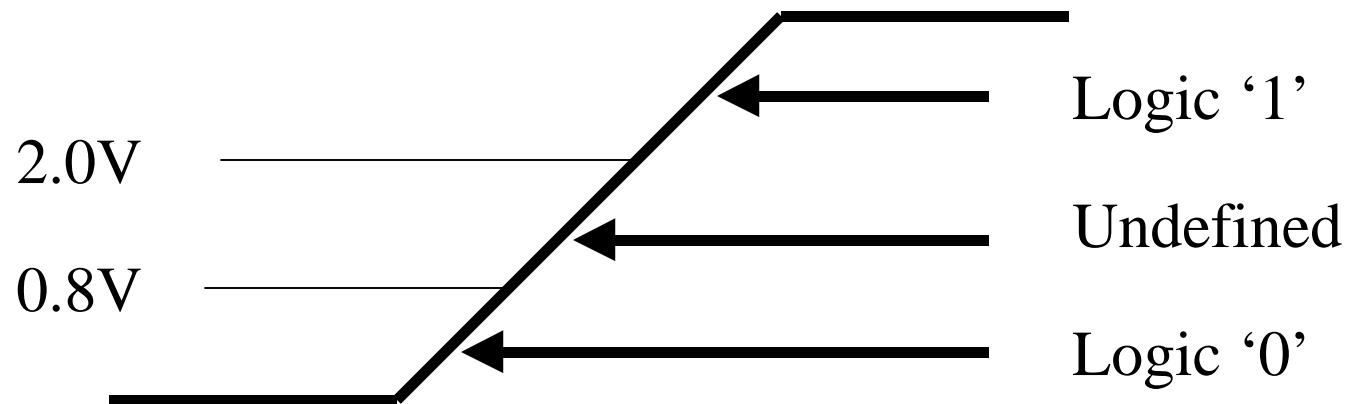
TTL Logic Levels

- Logic '1' threshold - V_{IH}
 - $\geq 2.0 \text{ V}$
- Logic '0' threshold - V_{IL}
 - $\leq 0.8 \text{ V}$
- Undefined
 - $0.8 \text{ V} < v < 2.0 \text{ V}$

TTL Logic Levels

Examples of Undefined Values

- Floating signals may take on illegal values
- What happens during a signal transition?



MIL-STD-1553

- Three physical values
 - $> +V_{TH}$
 - $< -V_{TH}$
 - $-V_{TH} < v < +V_{TH}$
- All values have meaning in MIL-STD-1553

Intel Flash Memory

- Intel® StrataFlash™ memory components store two bits per cell for high density and low cost.
- Smaller margins for radiation.

Physical Logic Values

- Magnetic field (core memory, plated wire)
- Voltage (static CMOS logic)
- Charge (DRAM)
- Position (contacts open/closed)
- Sound (tones for a modem)
- Light (fiber optics - MIL-STD-1773)
- Etc.

Logic Values and Basic Functions

Logic Values

- Levels
 - '0' or '1'
- Pulses
 - Presence or absence of pulses
 - Pulse widths
 - Number of pulses
- Zero crossings

A Multi-valued Logic System

```
TYPE std_ulogic IS (  
    'U', -- Uninitialized  
    'X', -- Forcing Unknown  
    '0', -- Forcing 0  
    '1', -- Forcing 1  
    'Z', -- High Impedance  
    'W', -- Weak Unknown  
    'L', -- Weak '0'  
    'H', -- Weak '1'  
    '- ' -- Don't Care  
);
```

Switching Algebra

- Two-valued variable from $\{0,1\}$
- Two binary operations
 - And (\cdot)
 - Or ($+$)
- One unary operation
 - Not ($'$)

Unary Operation - Not

A	Y
0	1
1	0

Binary Operation - And

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Binary Operation - [Inclusive] Or

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Binary Functions

- How many functions are there of two binary variables?
 - Basic combinatorics
 - 4 positions or rows: 00, 01, 10, 11
 - For each position you can select one of two values {0, 1}
 - Order counts since it is a function
 - Number of functions = $2 \times 2 \times 2 \times 2 = 2^4 = 16$

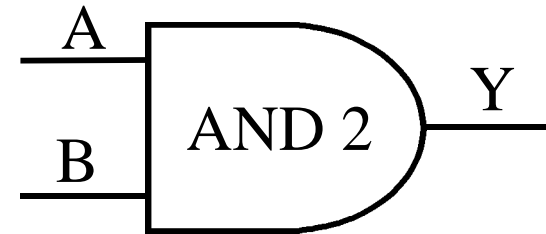
Binary Functions - cont'd

- How many functions are there of n binary variables?
 - 2^n positions or rows
 - for $n = 2$: 00, 01, 10, 11
 - for $n = 3$: 000, 001, 010, 011, 100, 101, 110, 111
 - For each position or row you can select one of two values $\{0, 1\}$
 - Number of functions = 2^{2^n}
 - for $n = 2$: $2^2 = 4$; $2 \times 2 \times 2 \times 2 = 2^4 = 16$
 - for $n = 3$: $2^3 = 8$; $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 64$
 - for $n = 4$: $2^4 = 16$; $2^{16} = 65536$

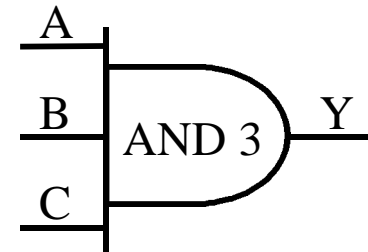
Fan-In

- Number of inputs to a logic gate
- Examples for And function

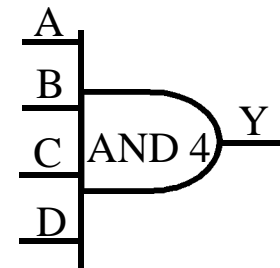
- $n = 2$: $Y = A \cdot B$



- $n = 3$: $Y = A \cdot B \cdot C$

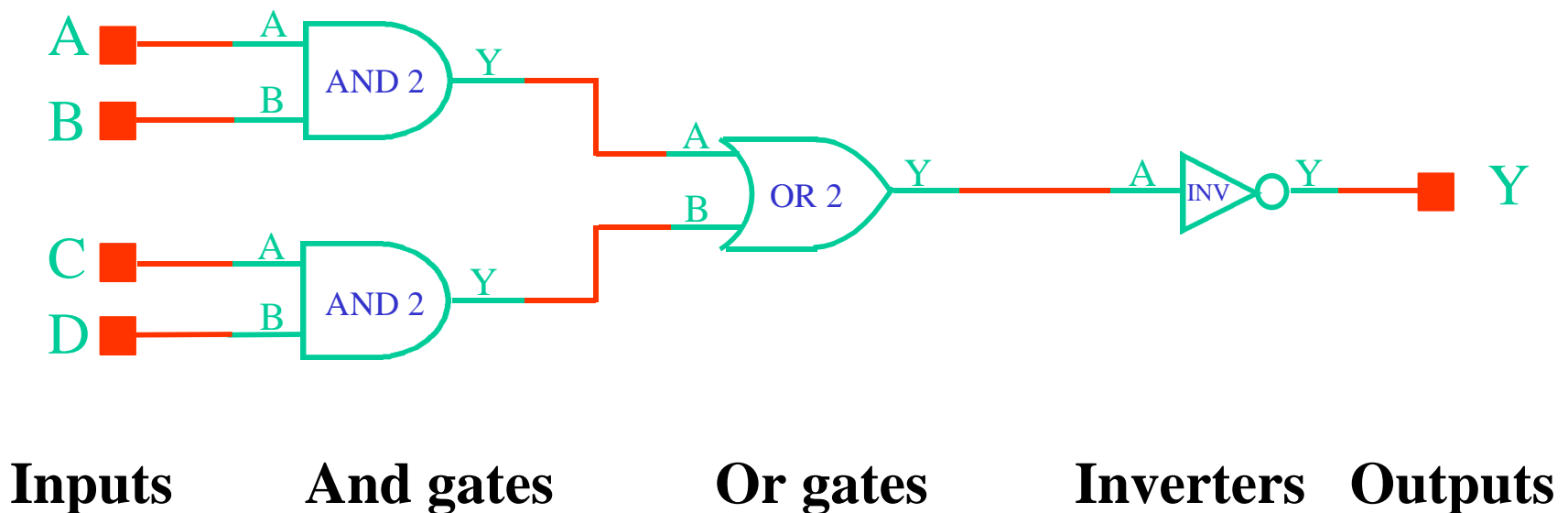


- $n = 4$: $Y = A \cdot B \cdot C \cdot D$



Logic Functions

- Any logic function can be implemented with And, Or, and Not.
- One standard form is the sum of products
 - Example: $Y = (A \cdot B + C \cdot D)'$



Universal Logic Element

Universal Logic Gate

Introduction

- Some other logic functions

- NOR ::= Negative OR

$$Y = (A + B)'$$

- NAND ::= Negative AND

$$Y = (A \cdot B)'$$

- Multiplexor

$$Y = A \cdot S + B \cdot S'$$

- Look up table (LUT)

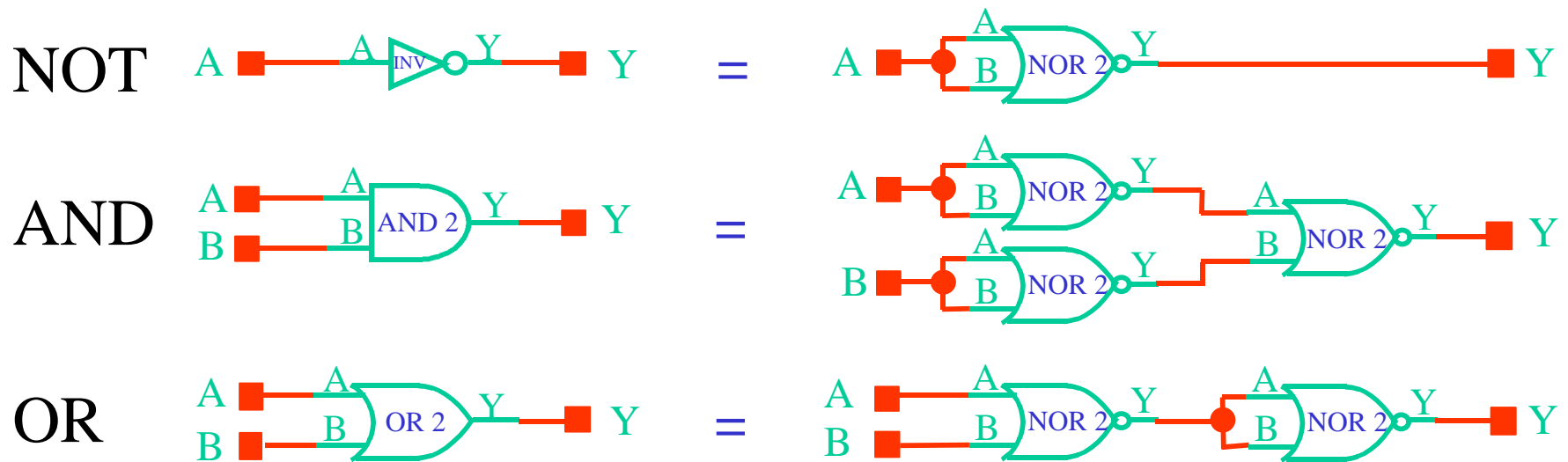
Small memory

Universal Logic Gate

NOR Function

- NOR ::= Negative OR

$$Y = (A + B)'$$

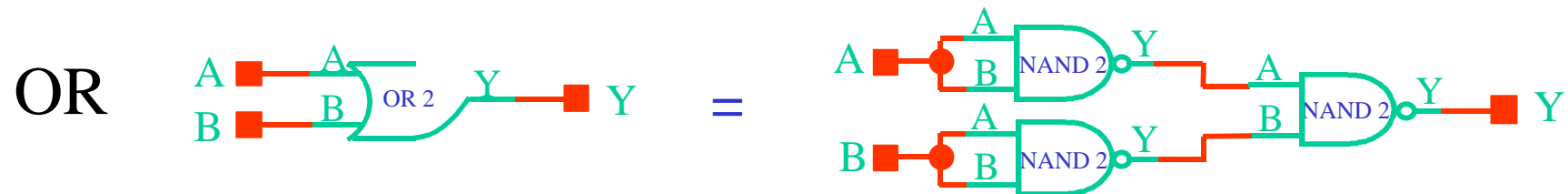


Universal Logic Gate

NAND Function

- NAND ::= Negative AND

$$Y = (A \cdot B)'$$

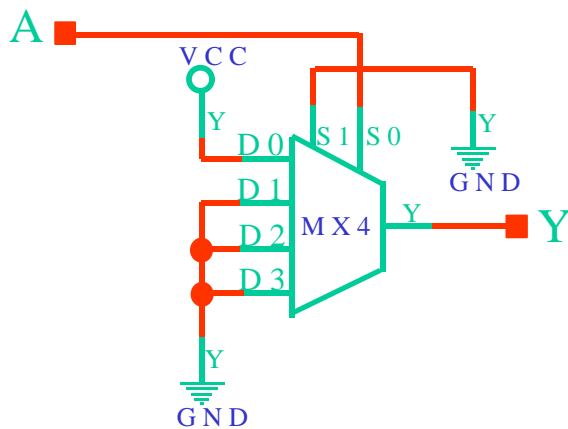


Universal Logic Gate Multiplexor Function

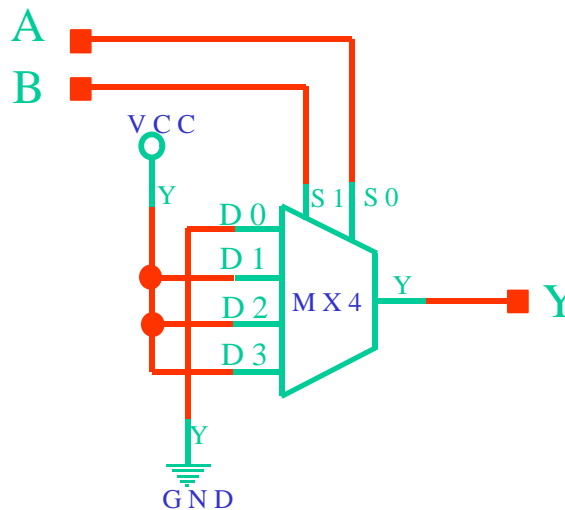
- Multiplexor

$$Y = A \cdot S + B \cdot S'$$

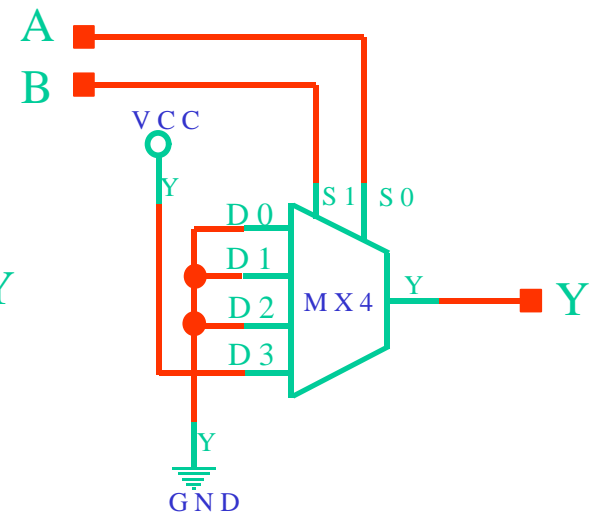
NOT



OR



AND



Universal Logic Gate

Look up table (LUT)

- Look up table (LUT)
Small memory

NOT

A	X	Y
0	0	1
0	1	1
1	0	0
1	1	0

OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Logic Assignments and Duality

Logic Gate and Voltage

A	B	Y
0V	0V	0V
0V	5V	0V
5V	0V	0V
5V	5V	5V

Logic Gate - Positive Logic

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

0 = 0V

1 = 5V

Logic Gate - Negative Logic

A	B	Y
1	1	1
1	0	1
0	1	1
0	0	0

1 = 0V

0 = 5V

Reorder Rows



Logic Gate - Negative Logic

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

1 = 0V

0 = 5V

Adders

Half-Adder

Truth Table

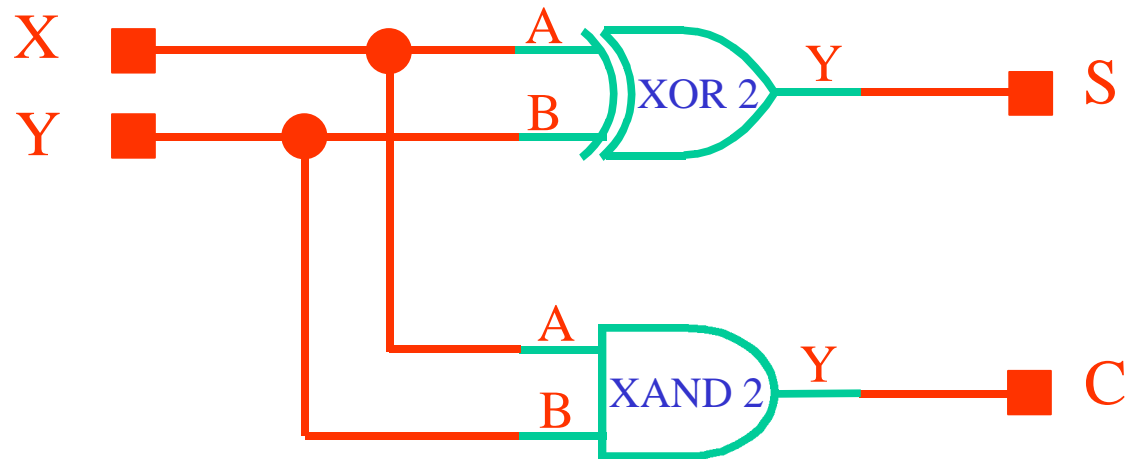
<u>x</u>	<u>y</u>	<u>C</u>	<u>S</u>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Logic Equations

$$C = x \cdot y$$

$$S = x \oplus y$$

Schematic



Full-Adder

Truth Table

x	y	z	C	S
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

Logic Equations

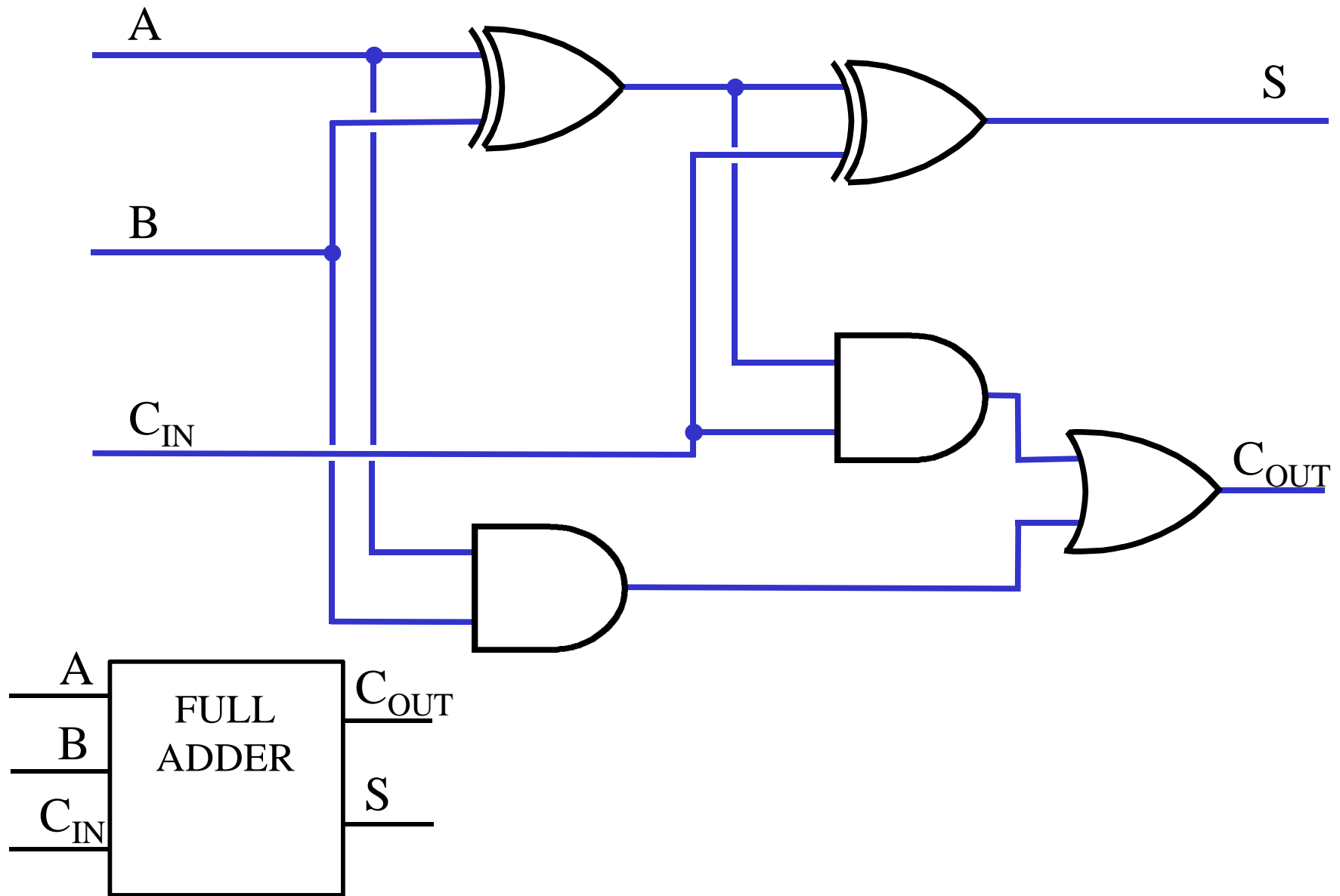
$$\begin{aligned}
 C &= x'y z + x y' z + x y z' + x y z \\
 &= z \cdot (x'y + x y') + x y \cdot (z + z') \\
 &= z \cdot (x \oplus y) + x \cdot y \\
 &= \text{MAJ}(x, y, z)
 \end{aligned}$$

$$\begin{aligned}
 S &= x'y'z + x'yz' + xy'z' + xyz \\
 &= x'y z' + x y' z' + x'y'z + xyz \\
 &= z'(x'y + x y') + z(x'y' + xy) \\
 &= z'(x \oplus y) + z(x \oplus y)' \\
 &= (x \oplus y) \oplus z \\
 &= x \oplus y \oplus z
 \end{aligned}$$

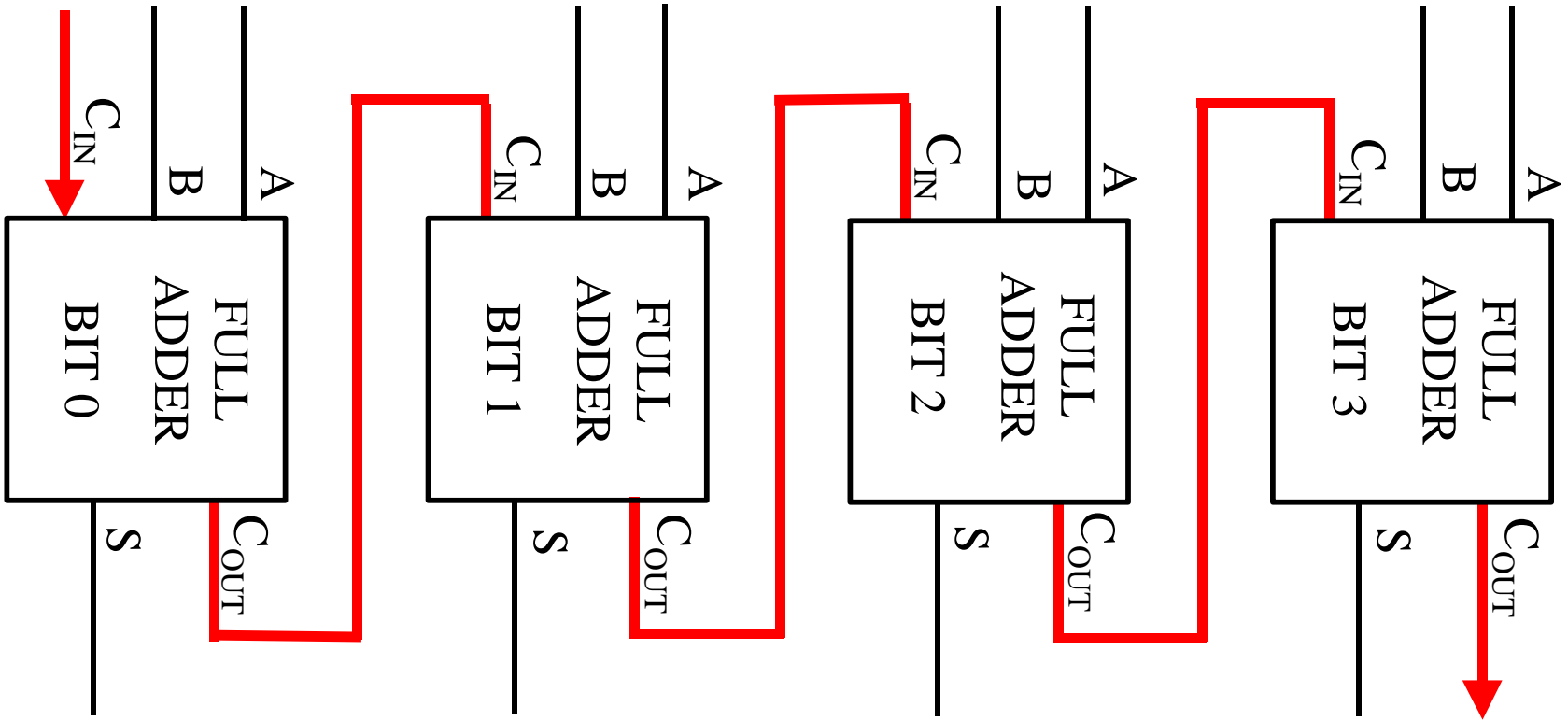
Adding Two Numbers

- Many Types of Adders: Some examples:
- Bit Serial Adder
 - Add time = $n \times f$
- Cascade Cascade Stages
 - Ripple carry adder
 - Add time = $n \times t_{PD}$
- Carry Look Ahead Adder
 - Generate carries in parallel
 - e.g., 4-bit AM2902. Can have “look ahead” of the “look ahead” units.

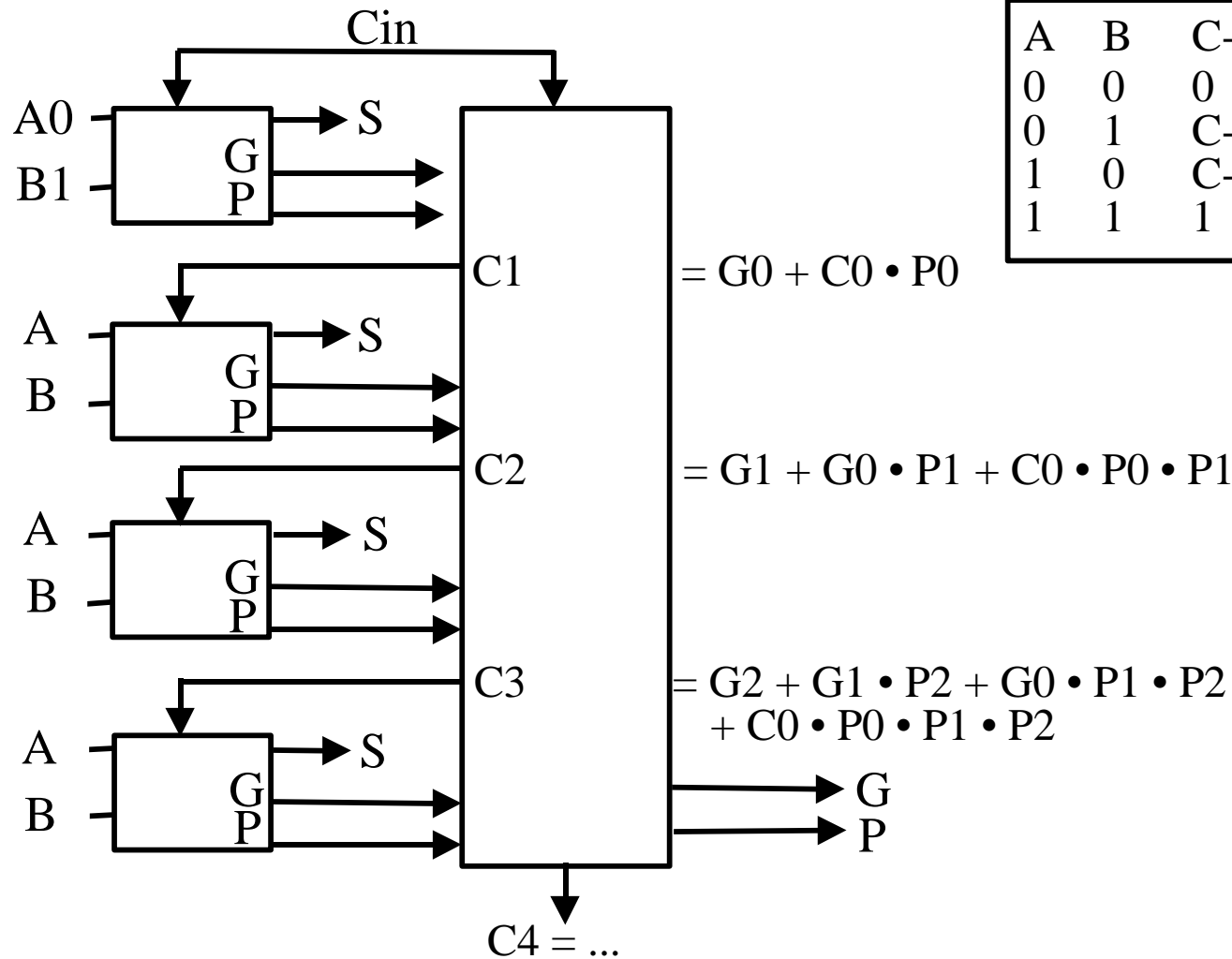
Full Adder Schematic



4-bit Ripple Carry Adder



Carry Lookahead Adder (CLA)



A	B	C-out	
0	0	0	“kill”
0	1	C-in	“propagate”
1	0	C-in	“propagate”
1	1	1	“generate”

Sample CLA Timing Analysis

- Generate g and p: 1 gate delay
 - Generate C: 2 gate delays
 - Generate sum: 3 gate delays
 - Total: 6 gate delays
- ⇒ Total is independent of word length

Negative Numbers and Subtraction

- Several different codes for negative numbers
 - 2's complement
 - 1's complement
 - signed magnitude
 - others
- For 2's complement, subtraction is implemented with the same hardware by negating the subtrahend. This is done by inverting each bit and adding one. The one can be added by setting the carry-in to the first stage equal to 1, saving an operation.