CENG 311
Computer Architecture Lab

Logic Design Review

2023-2024

Fall

Digital Electronics

- Digital electronics operate on 2 voltage levels: high and low.
- Binary systems: 1 or 0 , true or false, asserted or deasserted
- Values 0 and 1 are called as **complement** or **inverses** of one another.

Number Conversions

- **Decimal** -> Base 10;   1,56,798,…
- **Binary** -> Base 2;   0000, 001001,…
- **Octal** -> Base 8;   01,02,…
- **Hexadecimal (Hex)** -> Base 16; 0A, ABCD, FFFF,…

Binary- Hex Conversion

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Example:

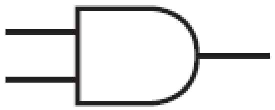
$$(0001001010111111)_2 = \underbrace{0001}_{1}\ \underbrace{0010}_{2}\ \underbrace{1011}_{B}\ \underbrace{1111}_{F}$$

$$(0001001010111111)_2 = (12BF)_{16}$$

Logic Functions and Gates

- **AND Operator:** A * B, logical product

AND Gate:

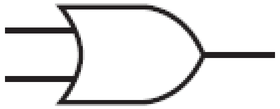


AND Truth Table

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

- **OR Operator**: A + B, logical sum

OR Gate:

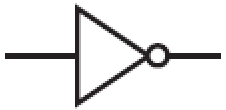


OR Truth Table

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

NOT operator: $\overline{A}$

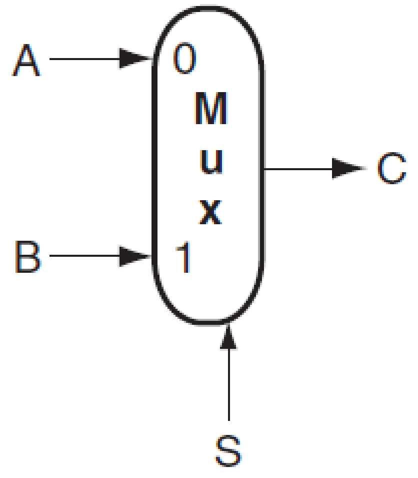
NOT Gate:



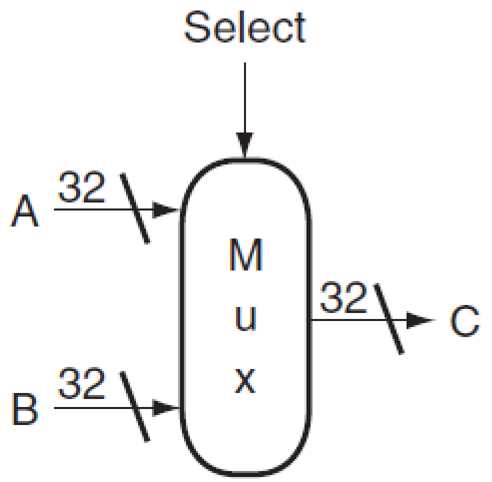
NOT Truth Table

A	B
0	1
1	0

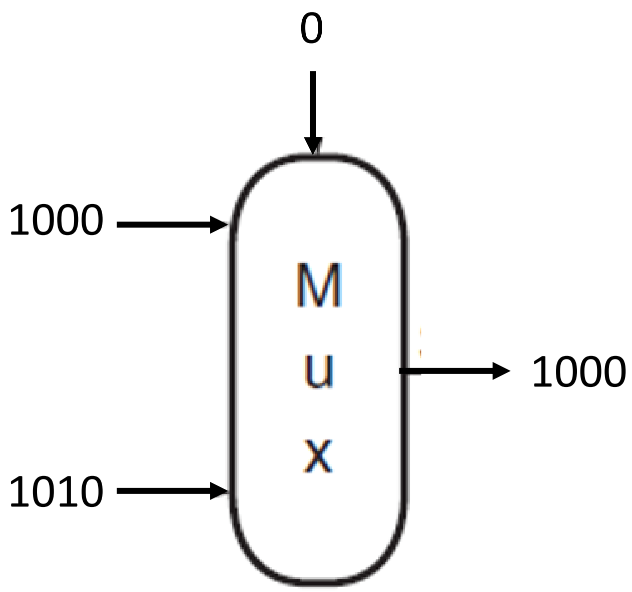
Multiplexer (Mux)



1-bit 2x1 Mux

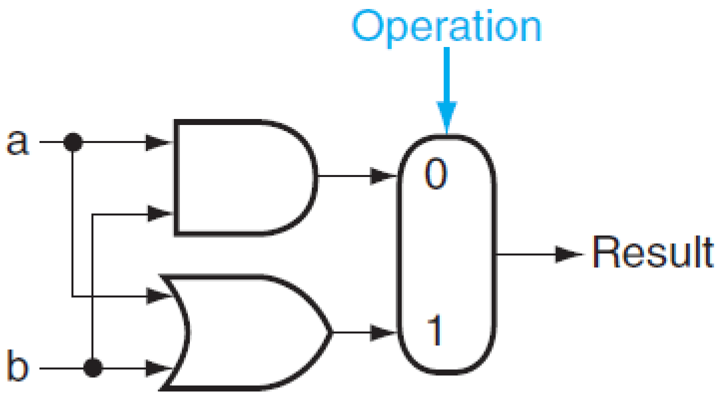


32-bits 2x1 Mux



4-bits 2x1 Mux

Basic Arithmetic and Logic Unit (ALU)



1 bit ALU

Binary Addition- Half Adder

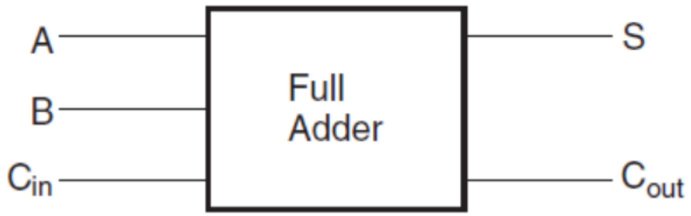
Inputs		Outputs	
A	**B**	**Sum**	**Carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

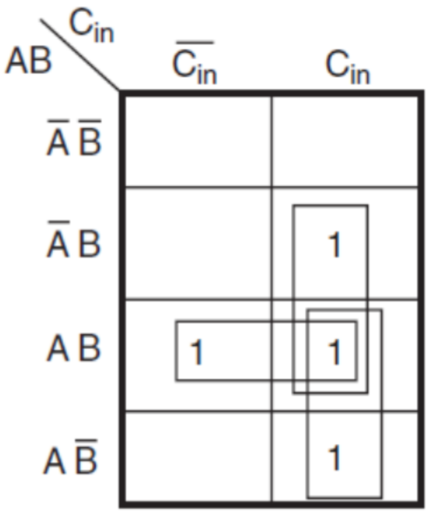
Binary Addition- Full Adder

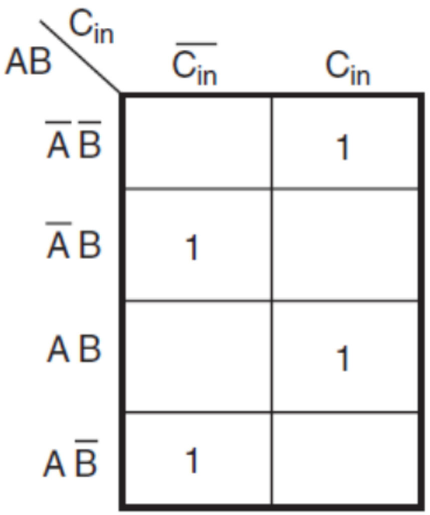
Inputs			Outputs		Comments
a	b	CarryIn	CarryOut	Sum	
0	0	0	0	0	$0 + 0 + 0 = 00_{two}$
0	0	1	0	1	$0 + 0 + 1 = 01_{two}$
0	1	0	0	1	$0 + 1 + 0 = 01_{two}$
0	1	1	1	0	$0 + 1 + 1 = 10_{two}$
1	0	0	0	1	$1 + 0 + 0 = 01_{two}$
1	0	1	1	0	$1 + 0 + 1 = 10_{two}$
1	1	0	1	0	$1 + 1 + 0 = 10_{two}$
1	1	1	1	1	$1 + 1 + 1 = 11_{two}$

Binary Addition- Full Adder

A	B	C_{in}	SUM (S)	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



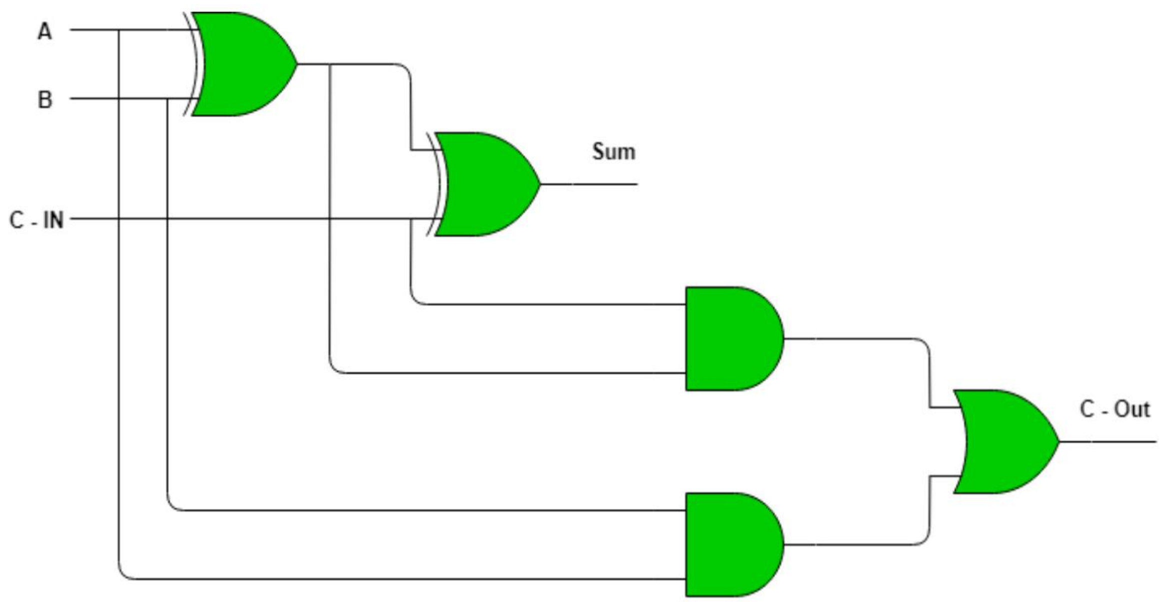


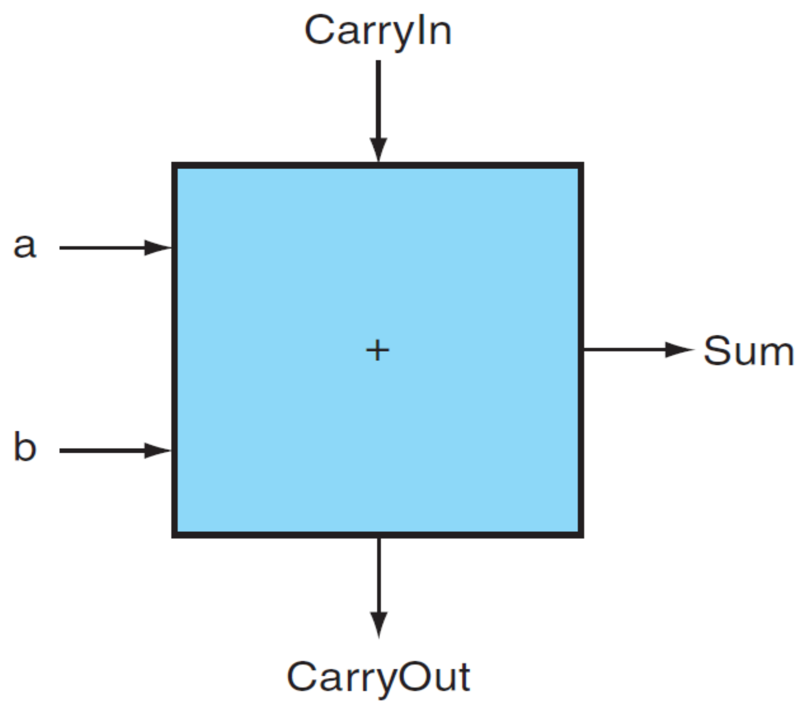


$$C_{out} = \overline{A}.B.C_{in} + A.\overline{B}.C_{in} + A.B.\overline{C}_{in} + A.B.C_{in}$$

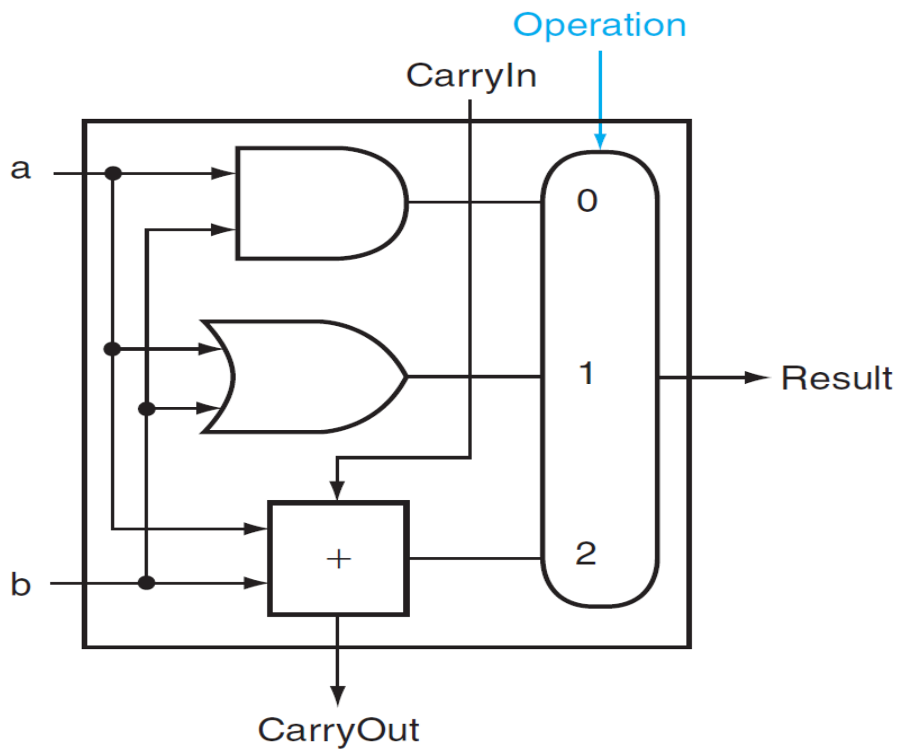
$$S = \overline{A}.\overline{B}.C_{in} + \overline{A}.B.\overline{C}_{in} + A.\overline{B}.\overline{C}_{in} + A.B.C_{in}$$

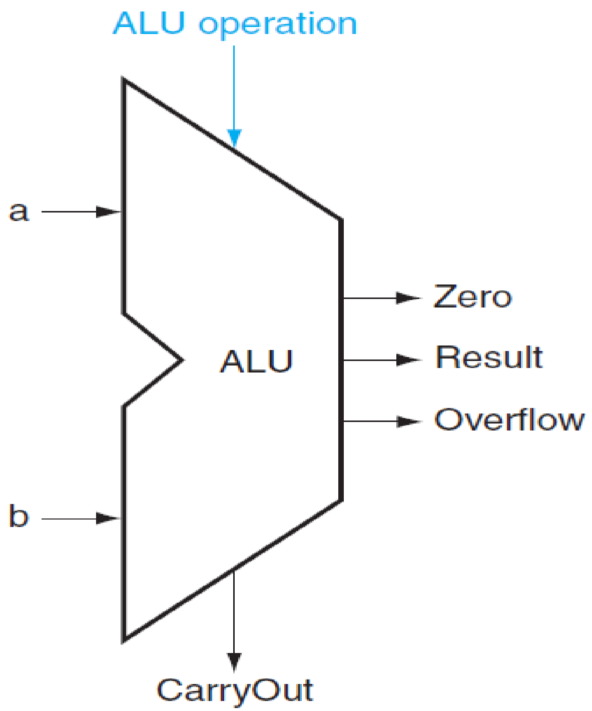
Binary Addition- Full Adder





1-bit ALU Unit with Adder Circuit





Binary Subtraction

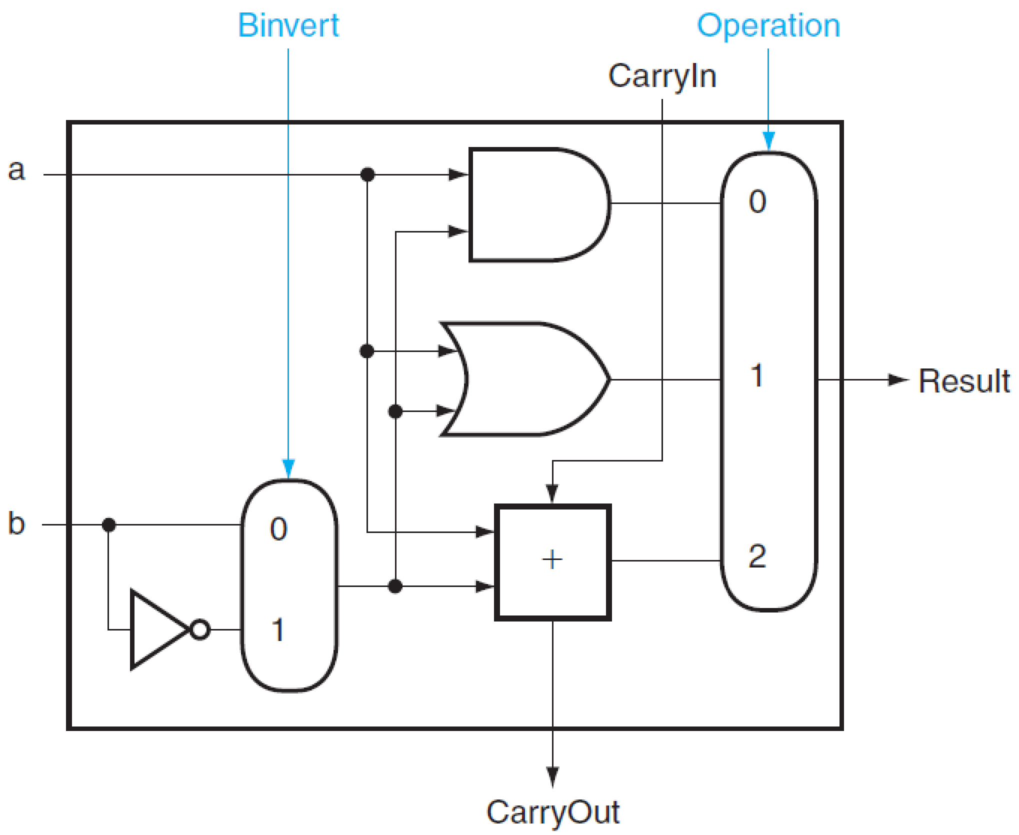
- Use 2's complement format:
    1. Invert each bits (take the 1's complement)
    2. Add 1.

Example: Write -5 in binary format.

$(5)_{10} = 0101$

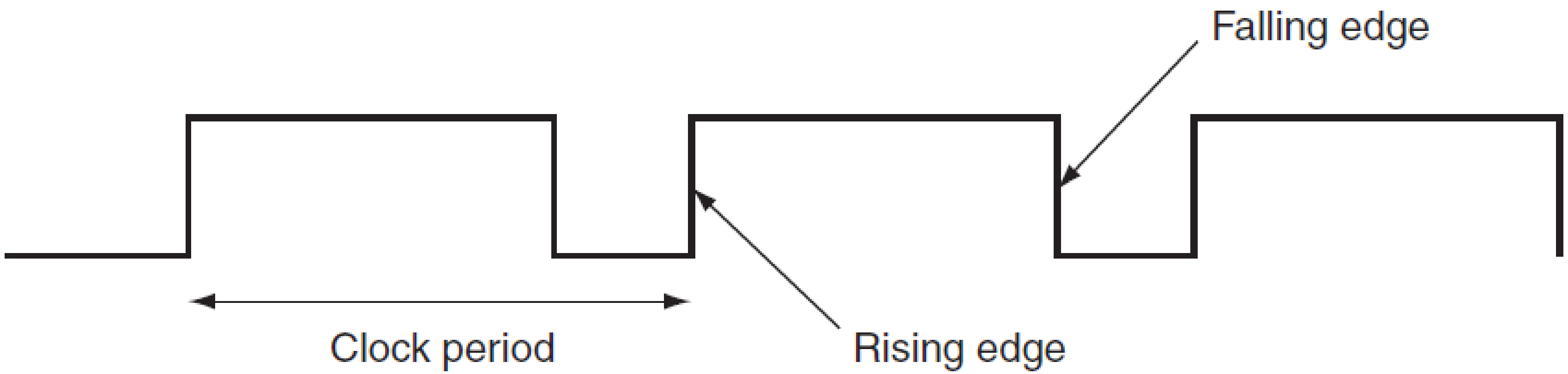
1. Take 1's complement: 1010
2. Add 1 : $(1011)_2 = (-5)_{10}$

ALU with binary subtraction



Clocks

- Clocks are used in sequential logic in order to decide when an element that contains **state** should be updated.
- **Edge triggered clocking:** state changes occurs on clock edges.



Clock Frequency = 1 / clock period

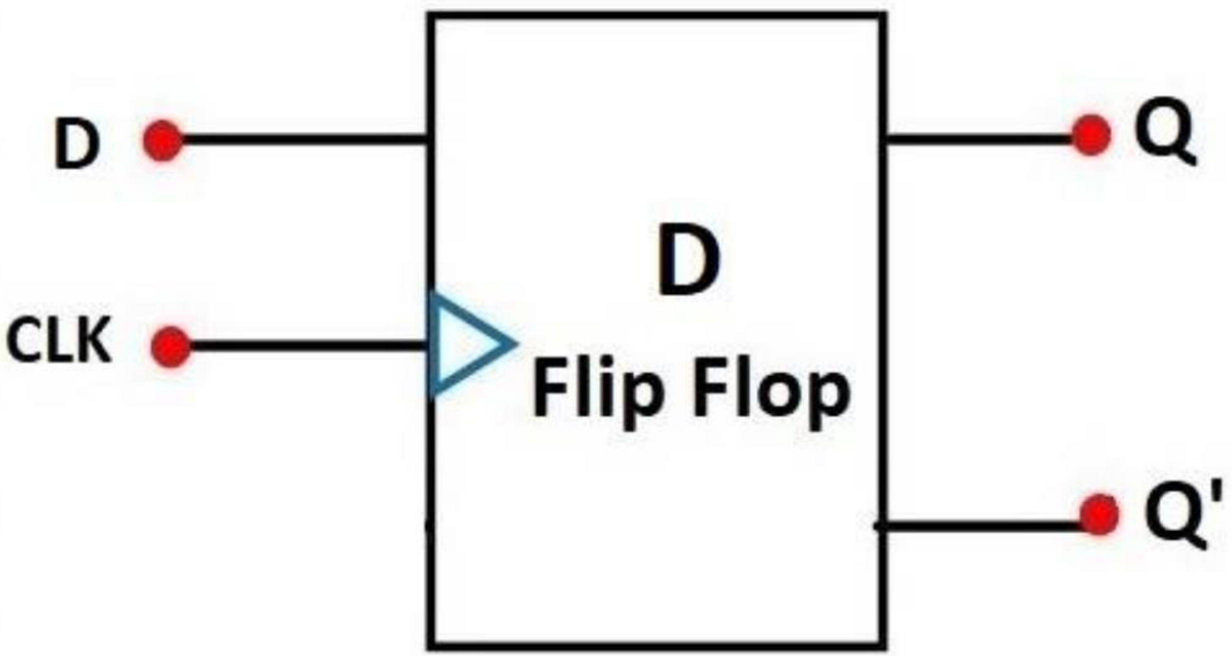
Flip-Flops – D Flip-Flop

- A kind of memory component that can hold 1 bit data

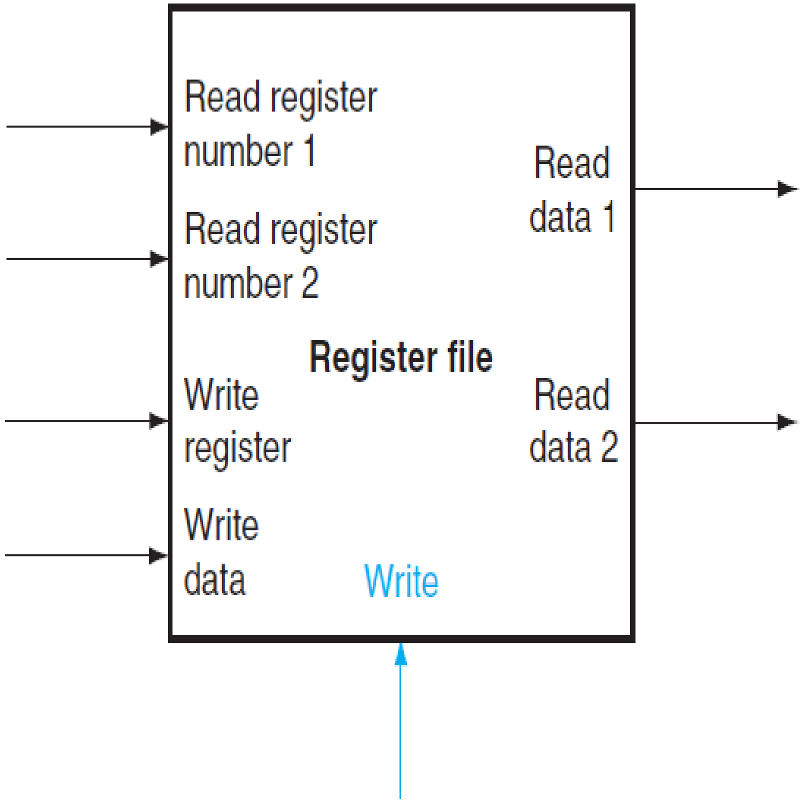
- D flip flops used in;
  - ❑Memory
  - ❑Registers
  - ❑Counters

Flip-Flops – D Flip-Flop

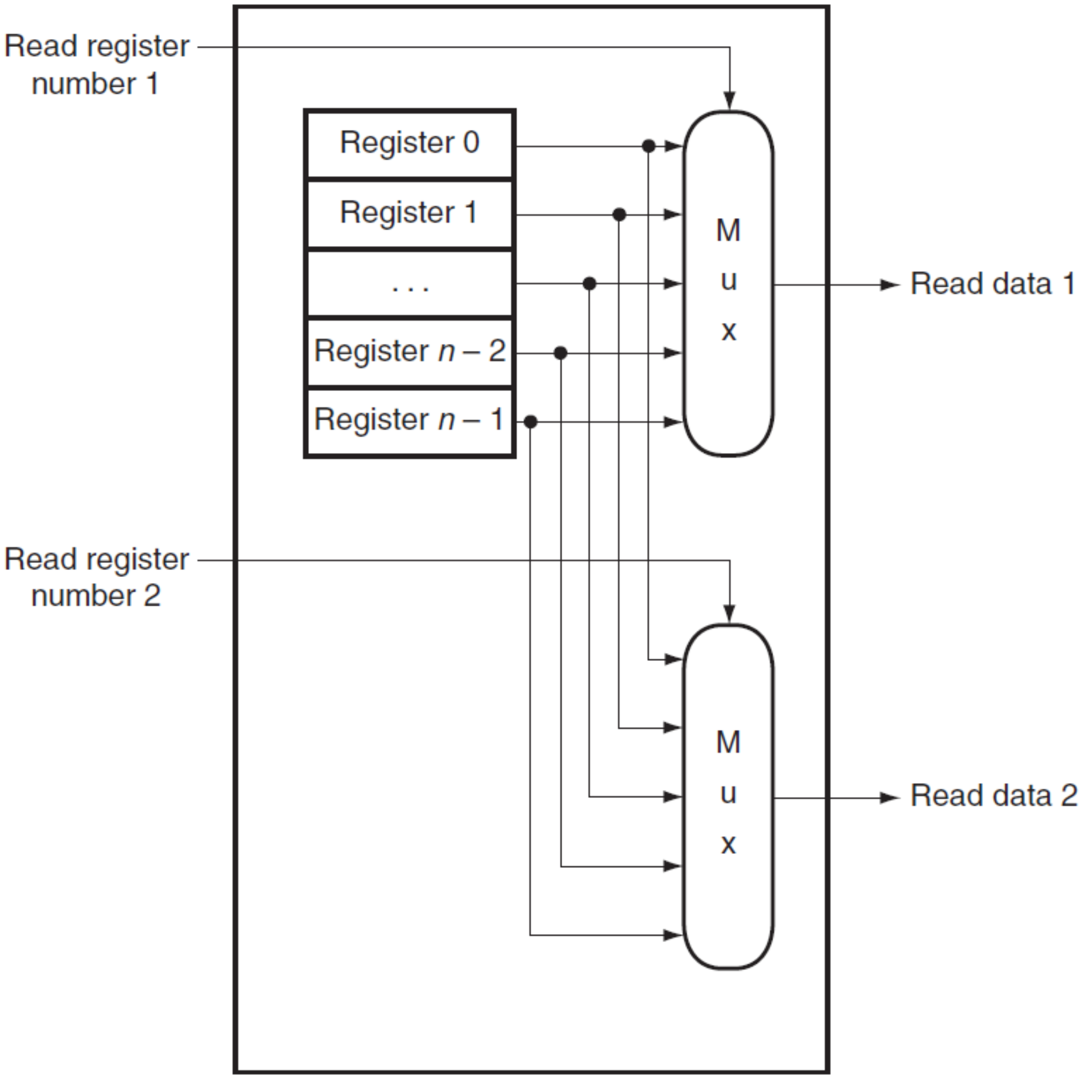
D	CLK	Q	Q`
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



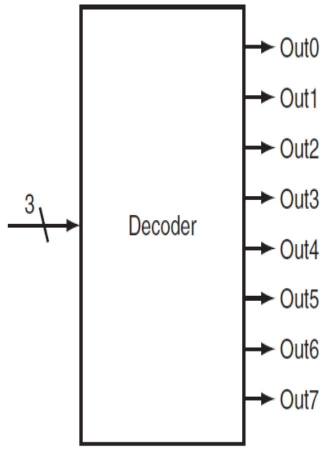
Register File



Read Register



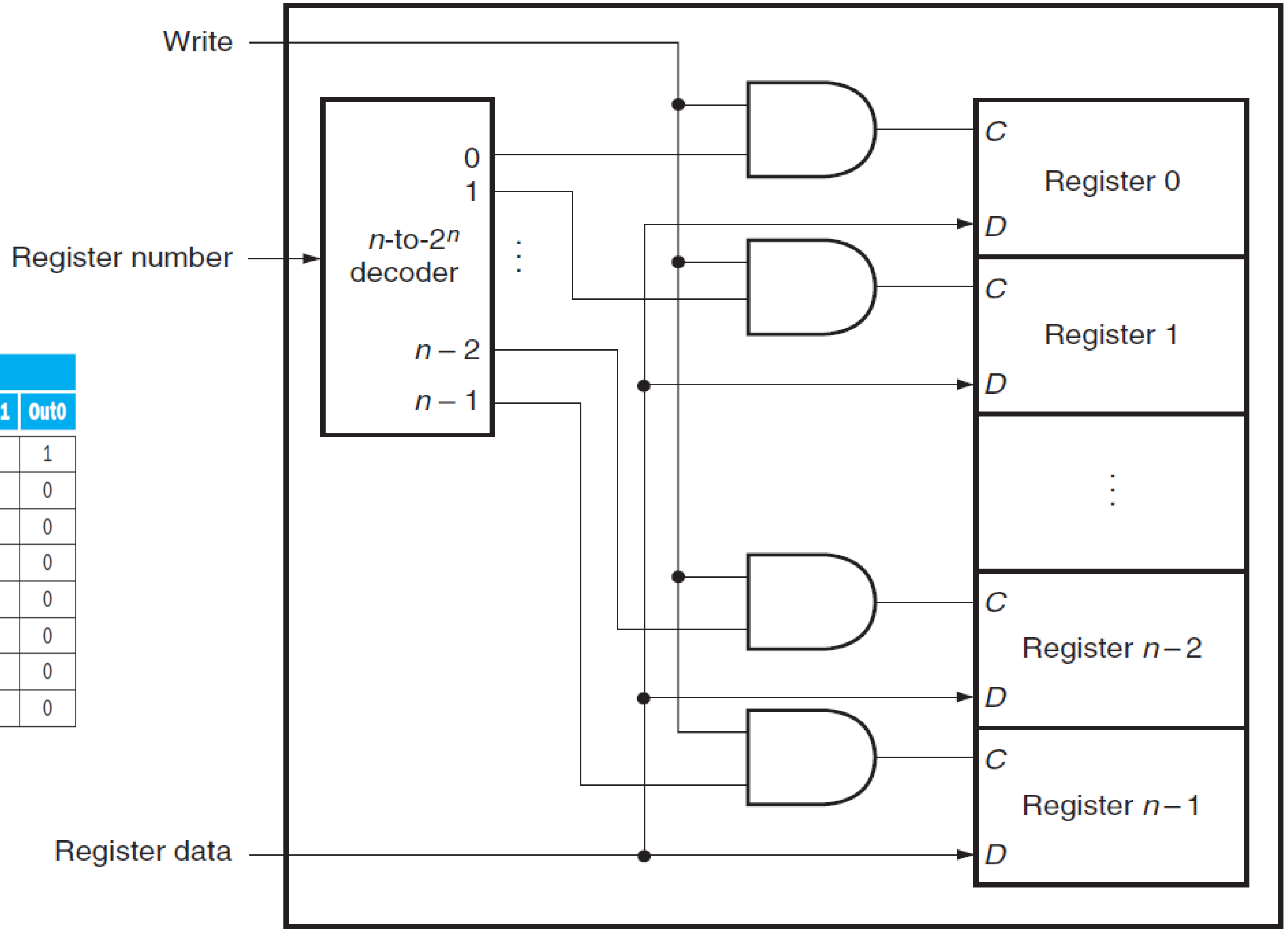
Write Register



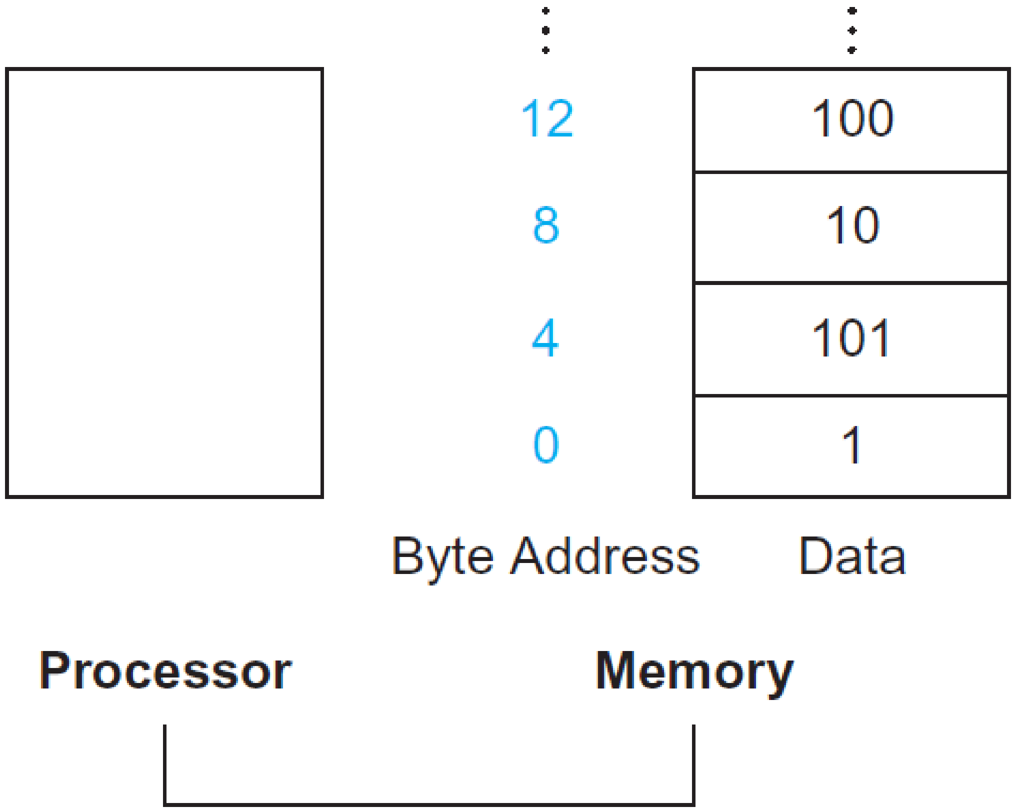
a. A 3-bit decoder

Inputs			Outputs							
12	11	10	Out7	Out6	Out5	Out4	Out3	Out2	Out1	Out0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

b. The truth table for a 3-bit decoder



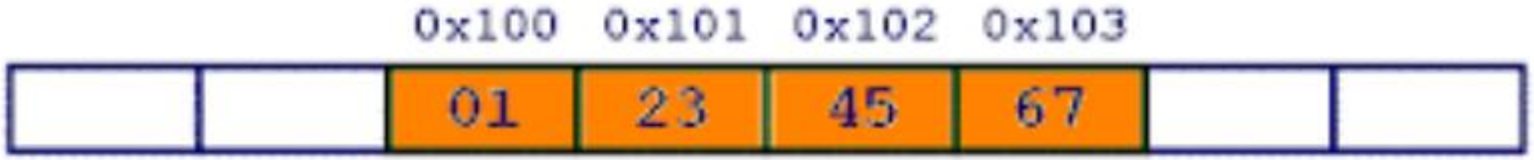
Memory address



1 Word = 4 bytes = 32 bits
1 byte = 8 bits

Little Endian- Big Endian

Data = 01234567



Big Endian

Little Endian