

# COMPUTER ORGANIZATION AND DESIGN

The Hardware/Software Interface

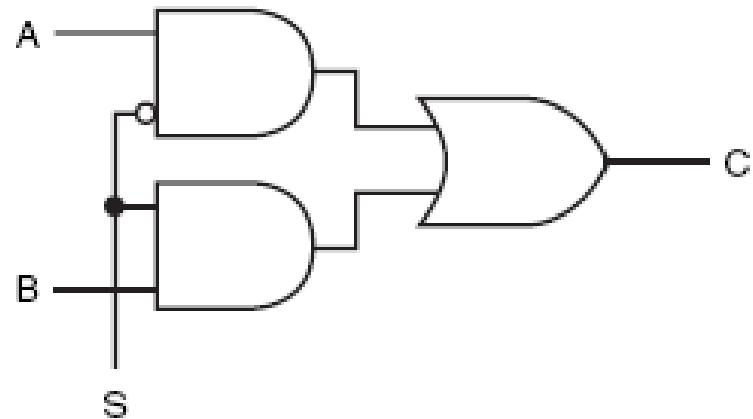
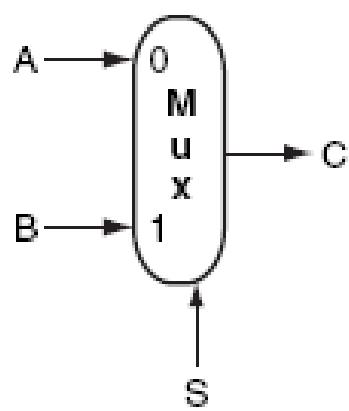
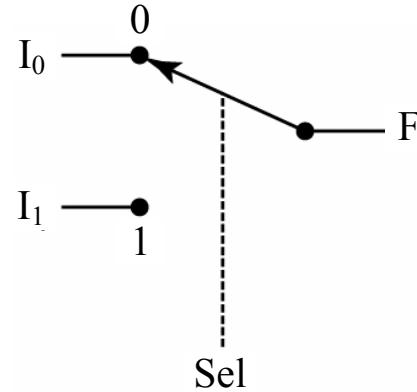
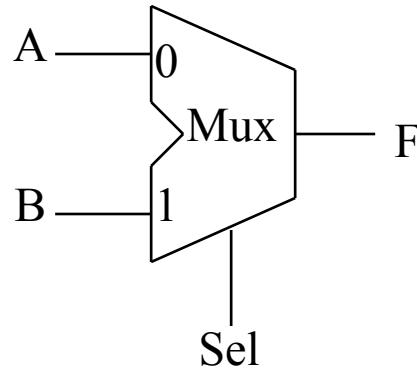
## Topic 5

### Review of Digital Logic

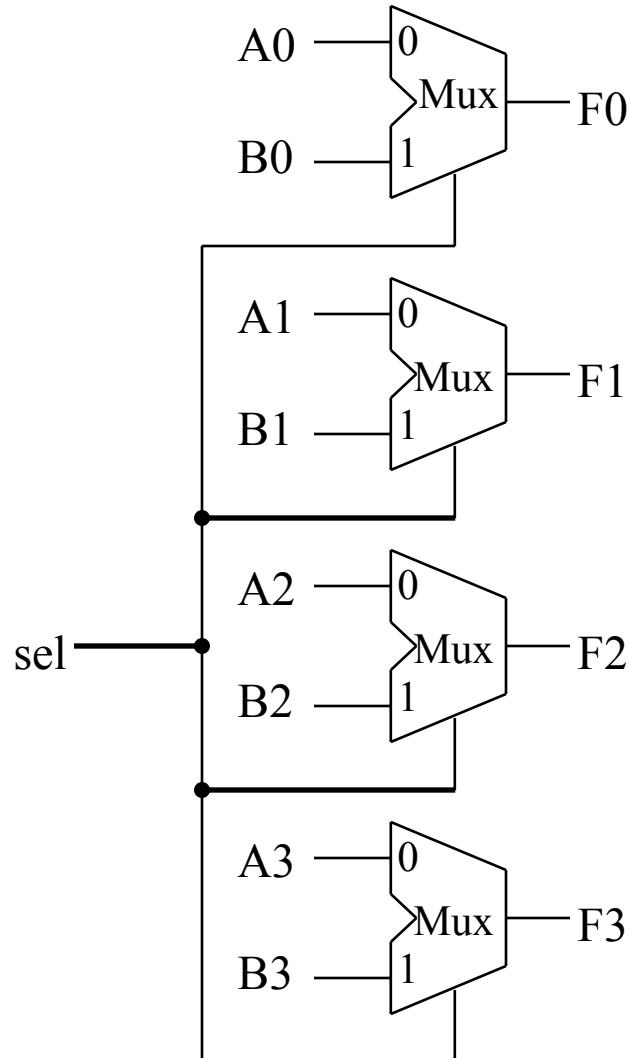
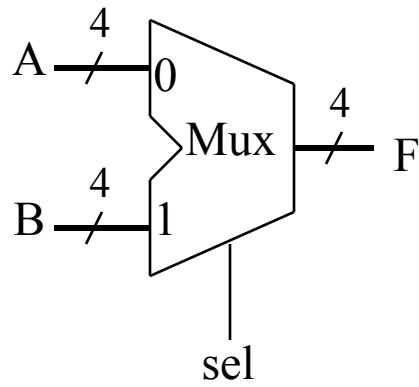
# Digital System Building Blocks

- Multiplexer (MUX)
- Decoder
- Register
- Register File
- Controller (Finite State Machine)
- Tri-State Buffer
- Memory
- ALU

# 2-to-1-Line Multiplexer (Mux)

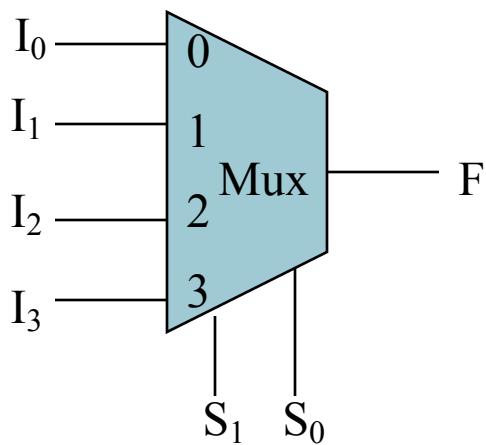


# Quad 2-to-1-Line MUX

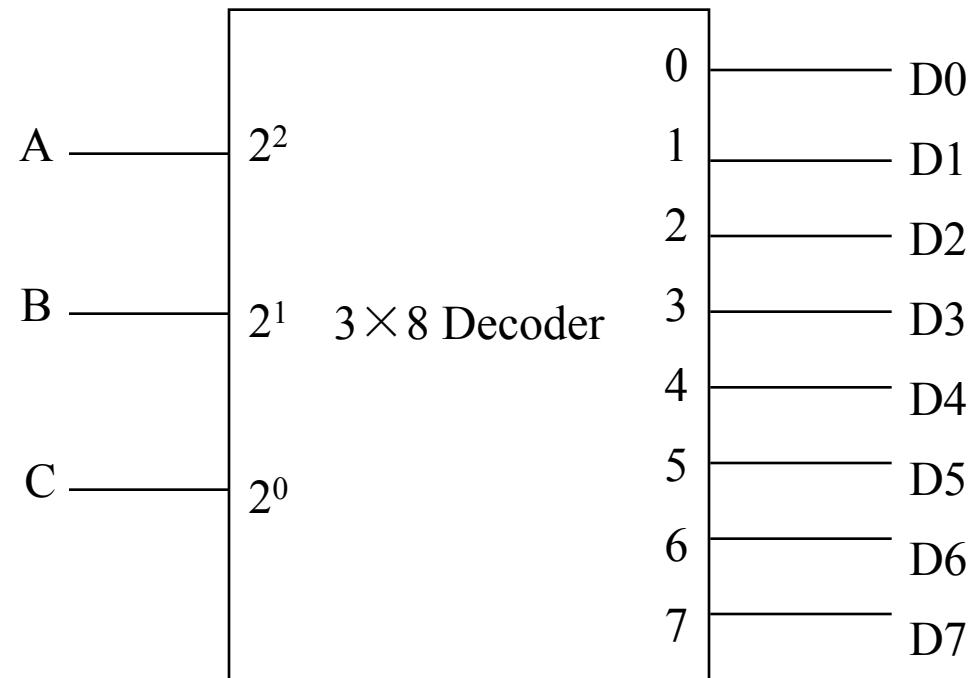


# 4-to-1-Line Multiplexer

- Selects one as output from 4 inputs, needs two select signals ( $4 = 2^2$ )



# **3×8 Decoder**



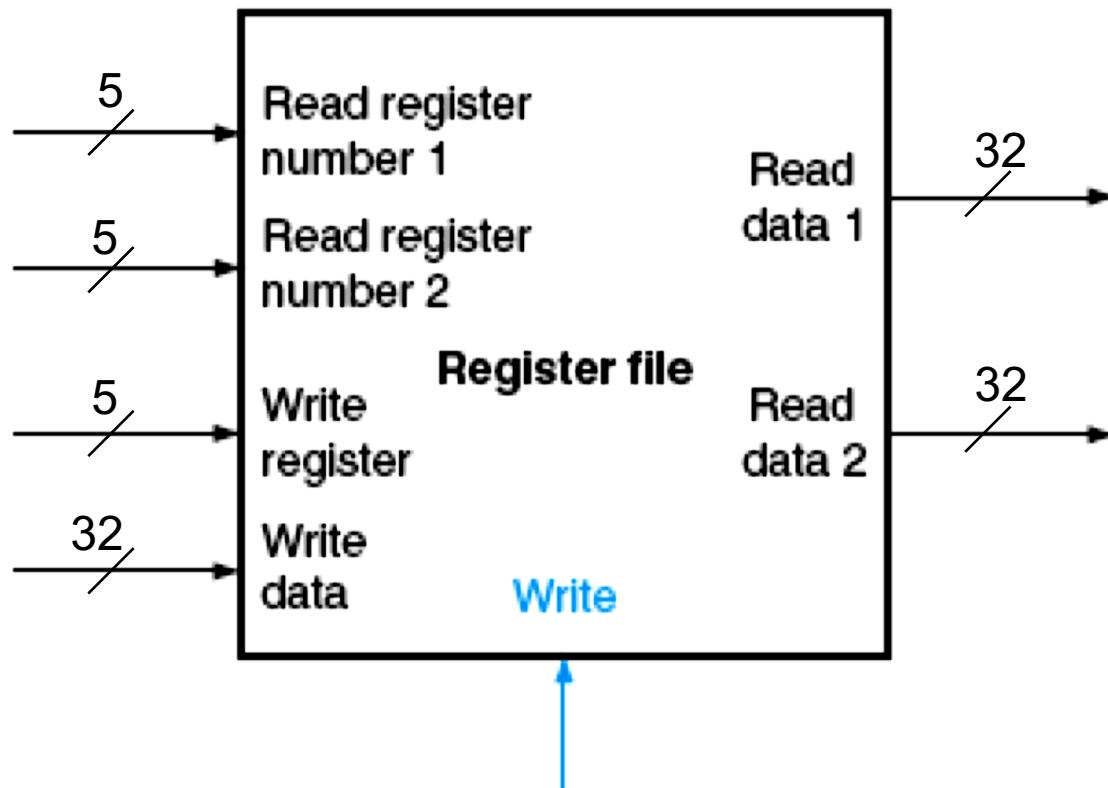
# Truth Table of $3 \times 8$ Decoder

- For any input combination, only one of the outputs is turned on

Inputs			Outputs							
A	B	C	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

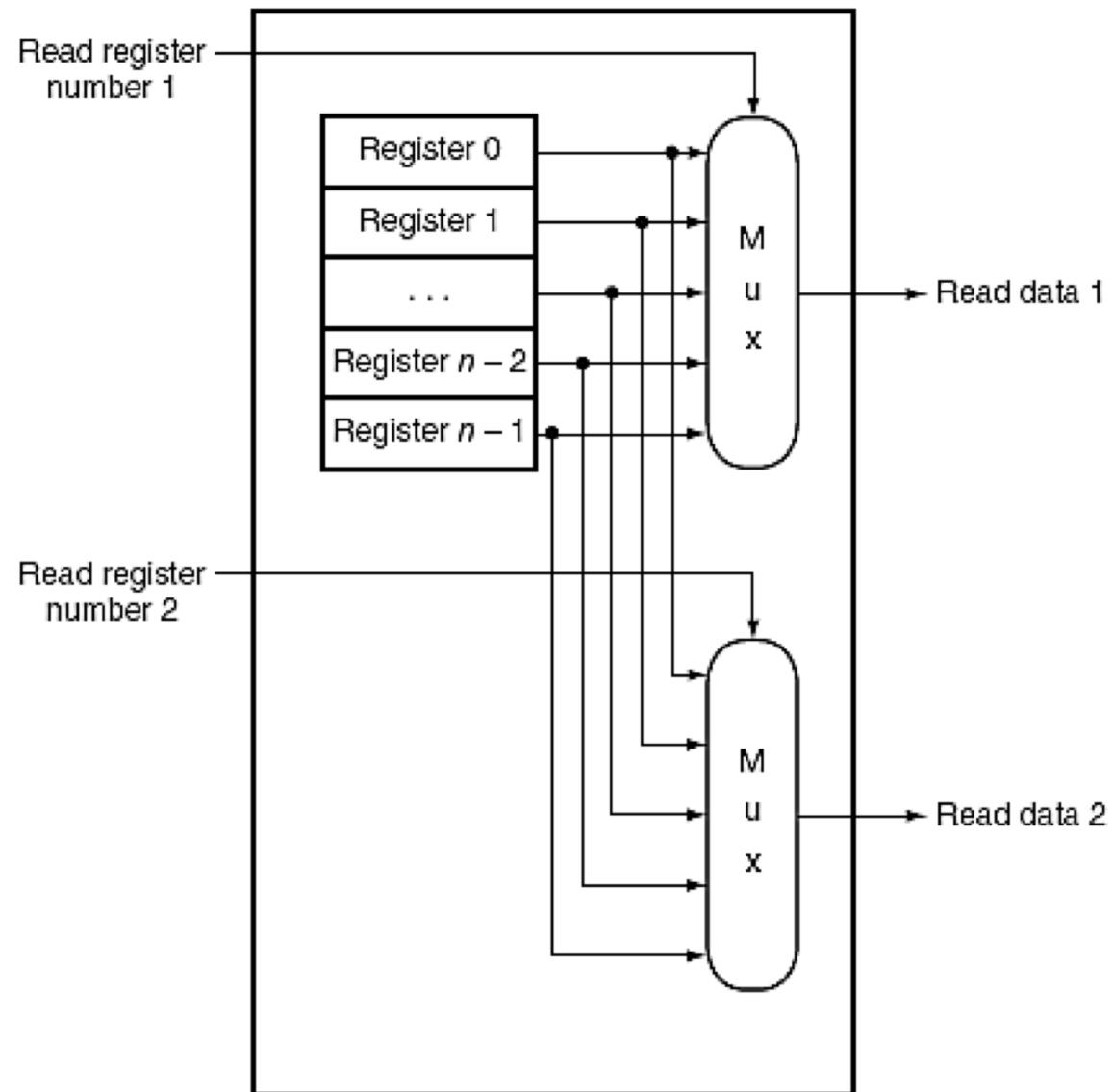
# Register File

- A bank of registers



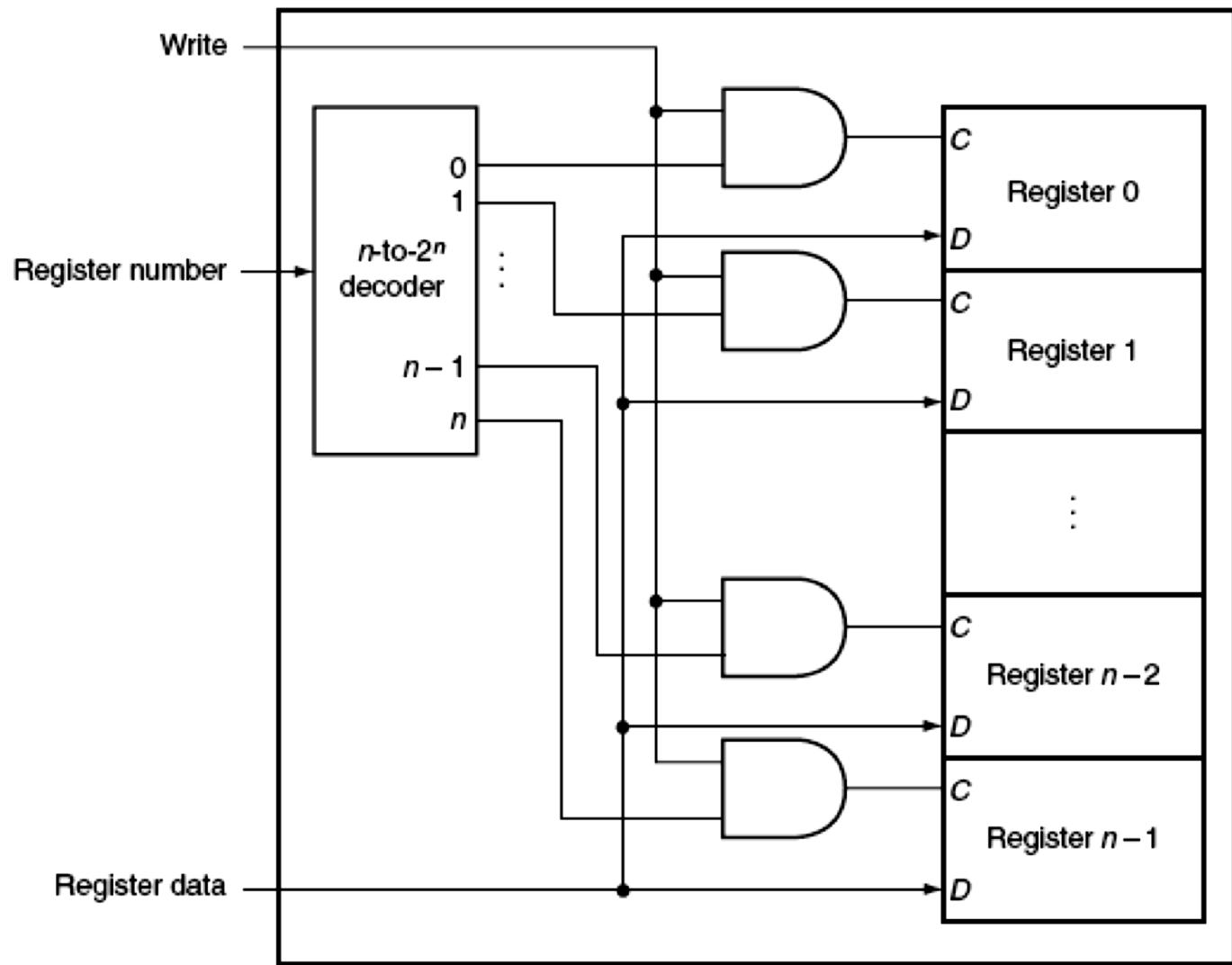
# Register File – Read Operation

- Reading register file



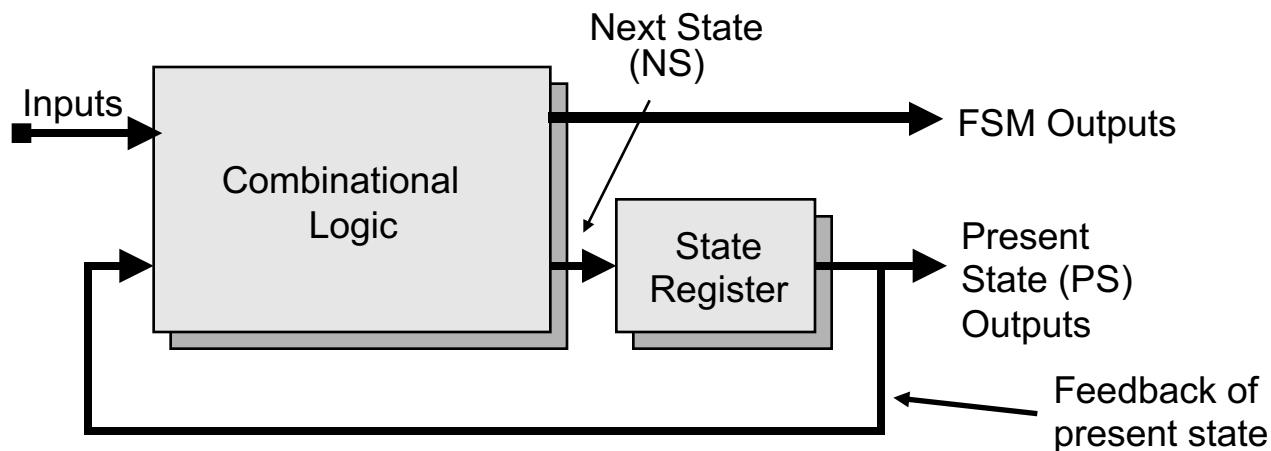
# Register File – Write Operation

- Writing register file



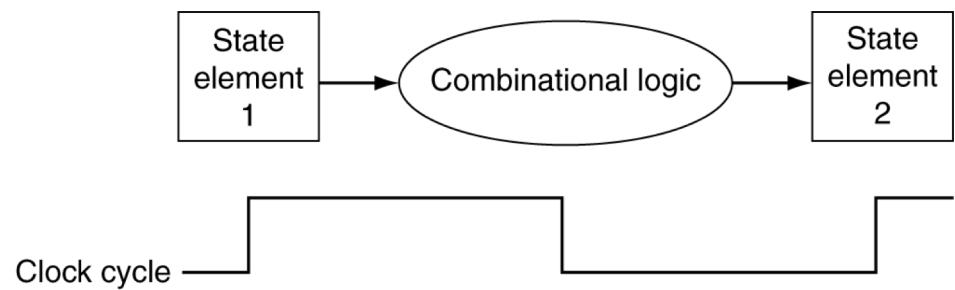
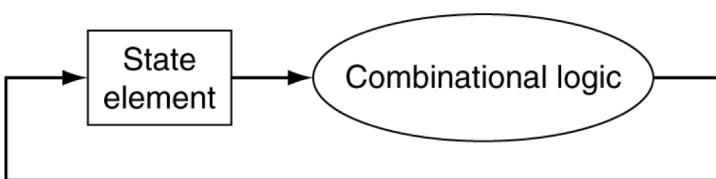
# State Machine

- Finite State Machine

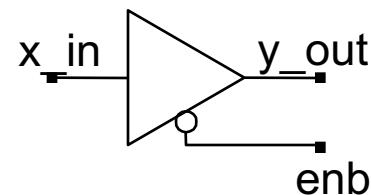
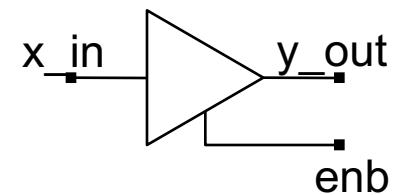


# Clocking Methodology for FSM

- Combinational logic transforms data during clock cycles
  - Between clock edges
- Clock cycles should be
  - Long enough to allow combinational logic completes computation
    - Longest delay determines clock period
  - Short enough to ensure acceptable performance and to capture small changes on external inputs



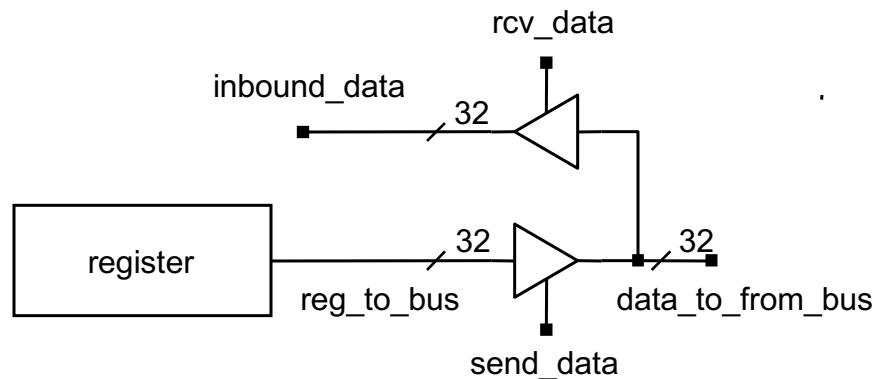
# Tri-state Buffers



x_in	enb	y_out
X	0	Z
0	1	0
1	1	1

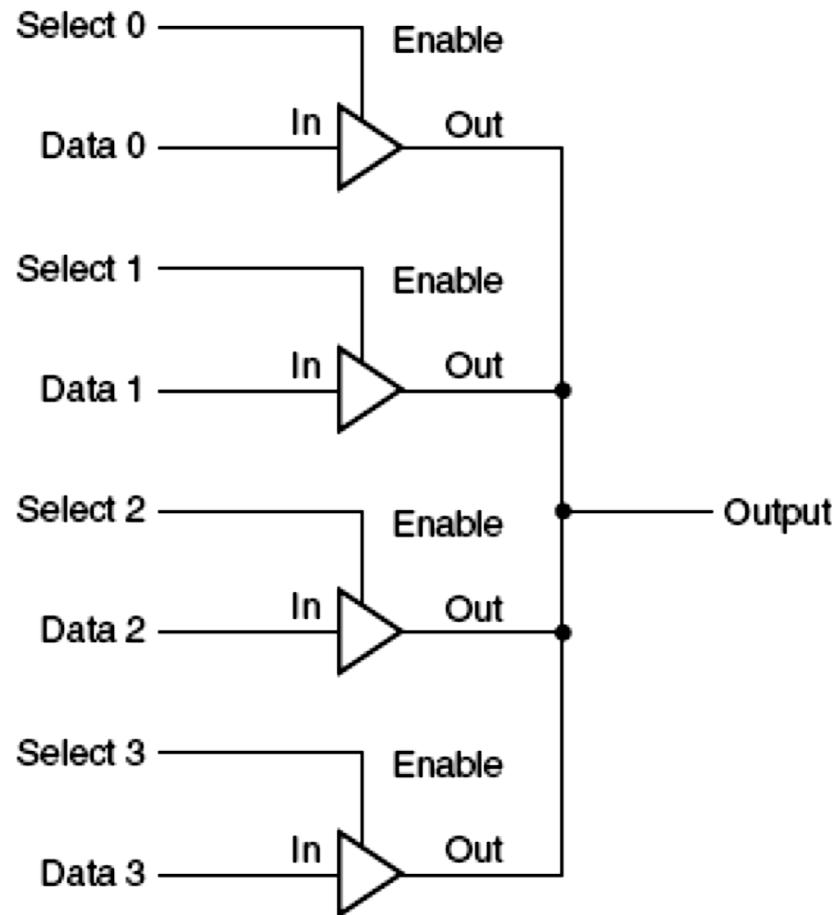
x_in	enb	y_out
X	1	Z
0	0	0
1	0	1

Typical applications: i/o pad and bus isolation.



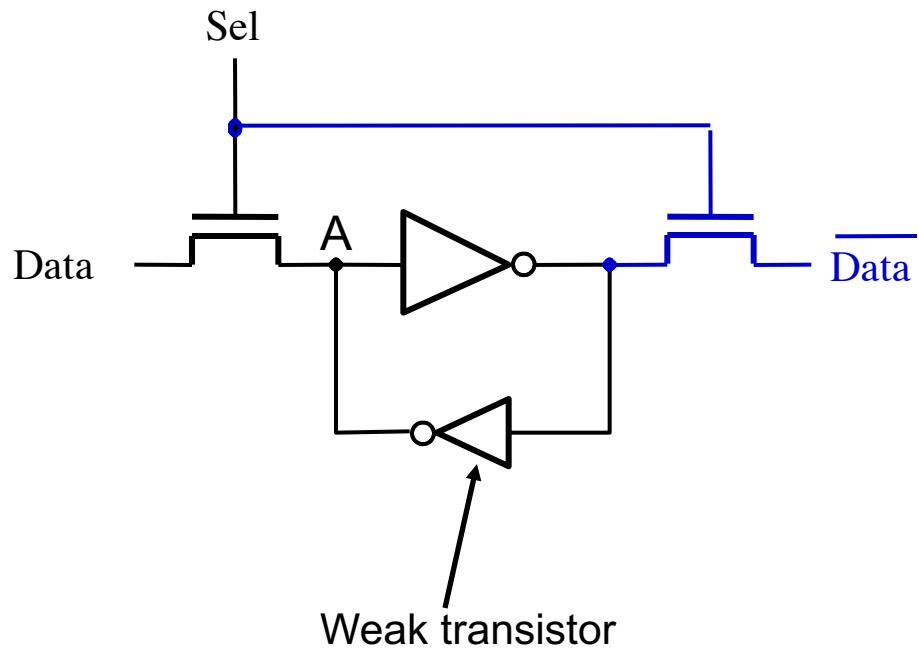
# Replacement of Big MUXes

- Mux becomes impractical when bigger than 32 channels
  - Replaced with tri-state buffers



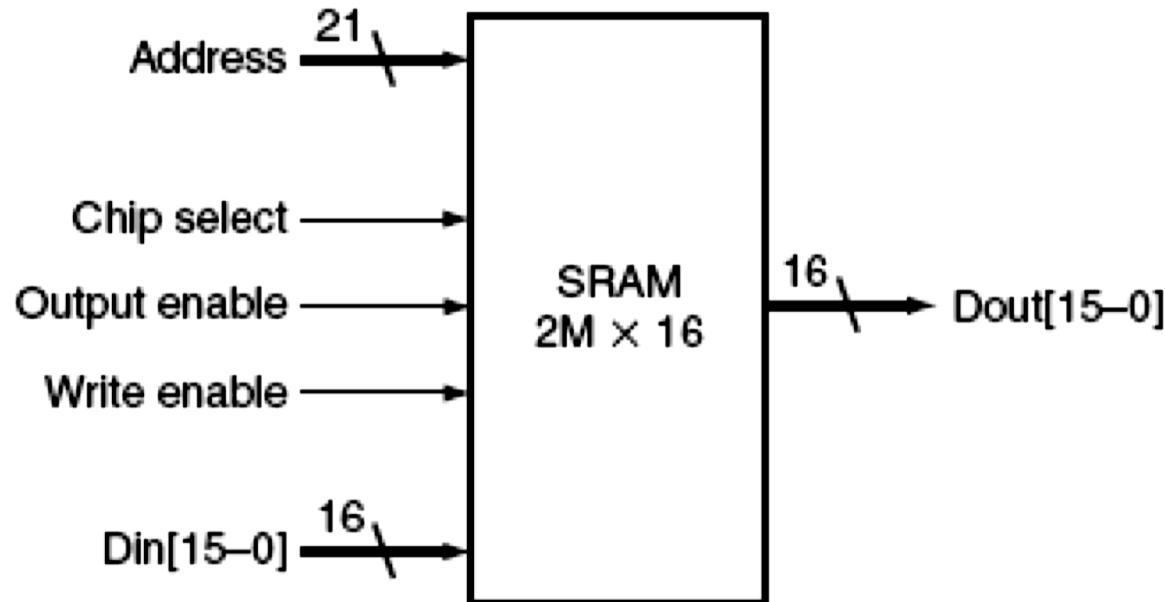
# Memory – Static RAM (SRAM)

- When Sel = 1, Data is stored and retained in the SRAM cell by the feedback loop
- When Sel = 0, Data can be read out on the same port
- Point A is driven by both the Data transistor and the smaller inverter, but the Data transistor wins because the inverter is implemented using a weak transistor



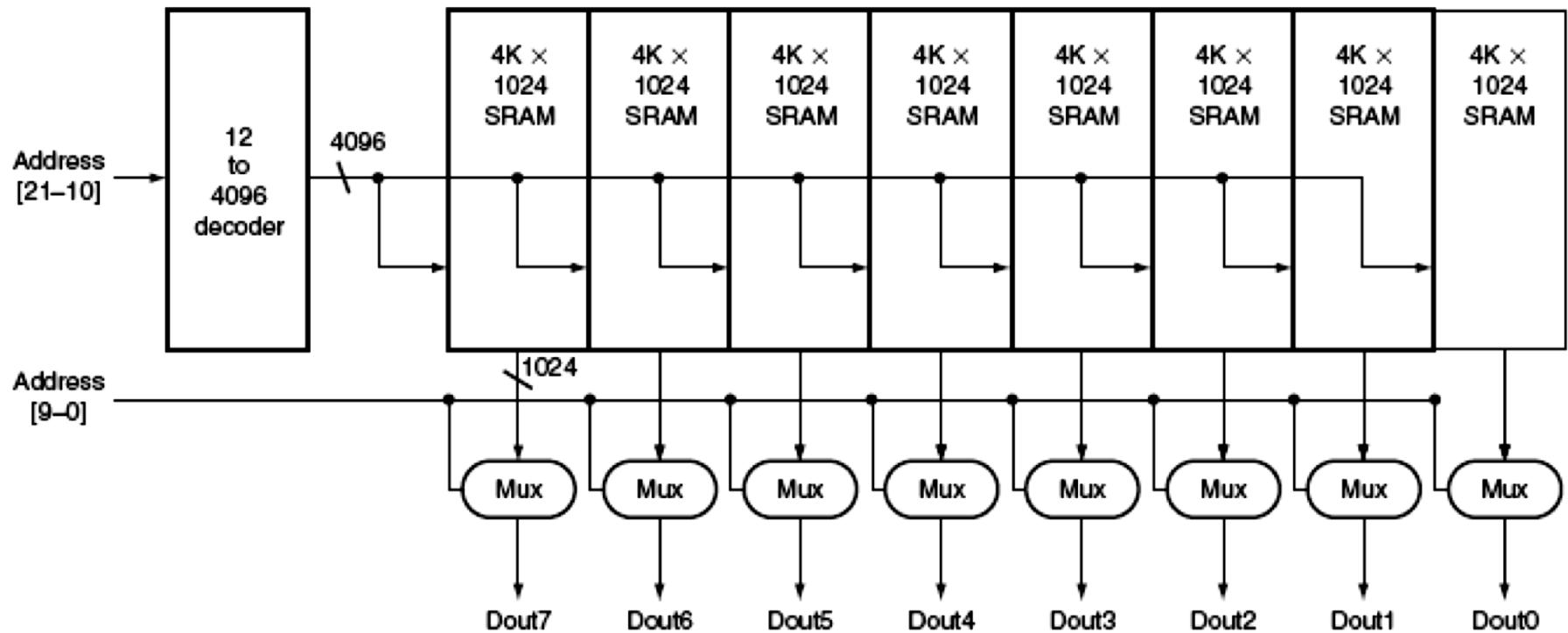
# Memory – SRAM

- Typical SRAM block



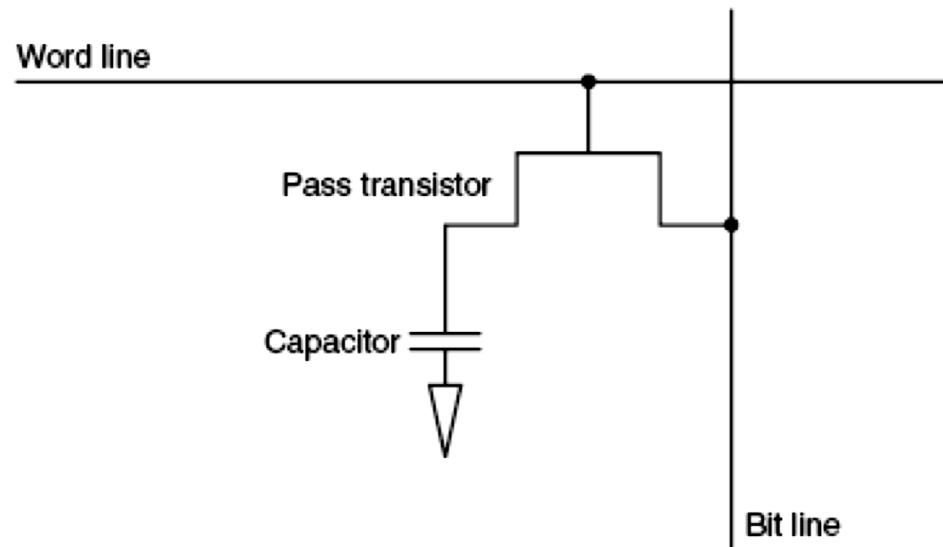
# Memory – SRAM

- Typical memory organization
    - Typical access time: < 20 ns



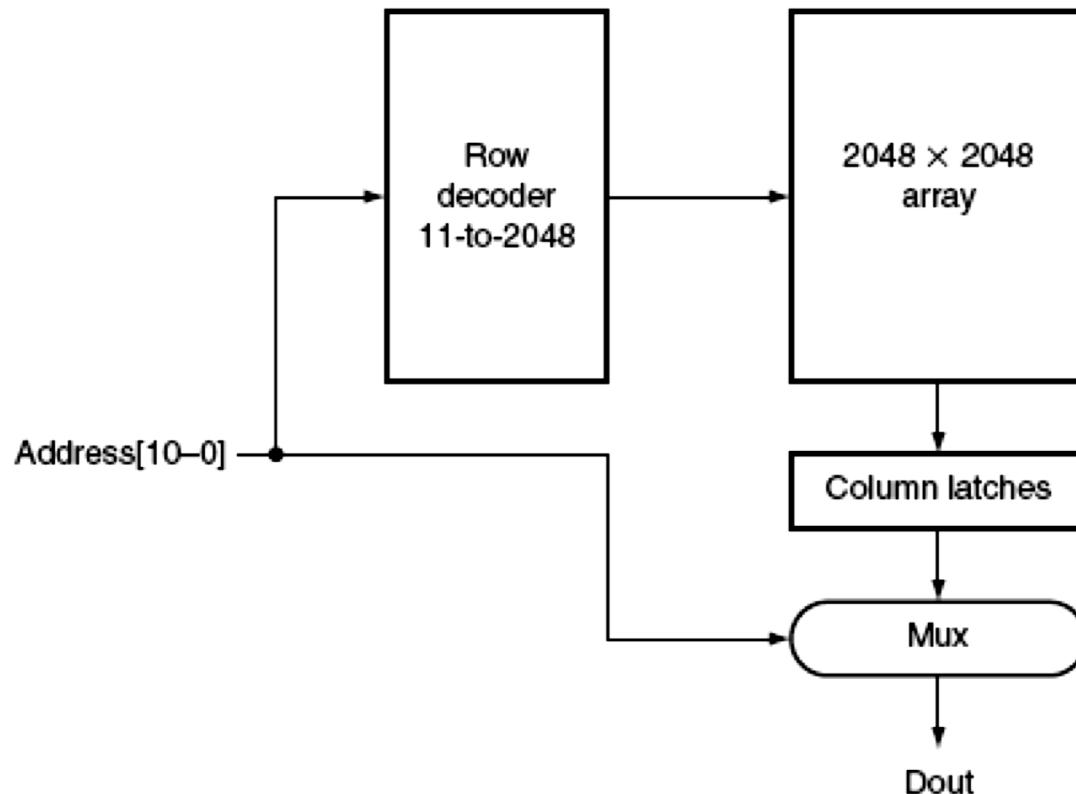
# Memory – Dynamic RAM (DRAM)

- Write: turn on word line, charge capacitor through pass transistor by bit line
- Read: charge bit line halfway between high and low, turn on word line, then sense the voltage change on bit line
  - 1 if voltage increases
  - 0 if voltage decreases

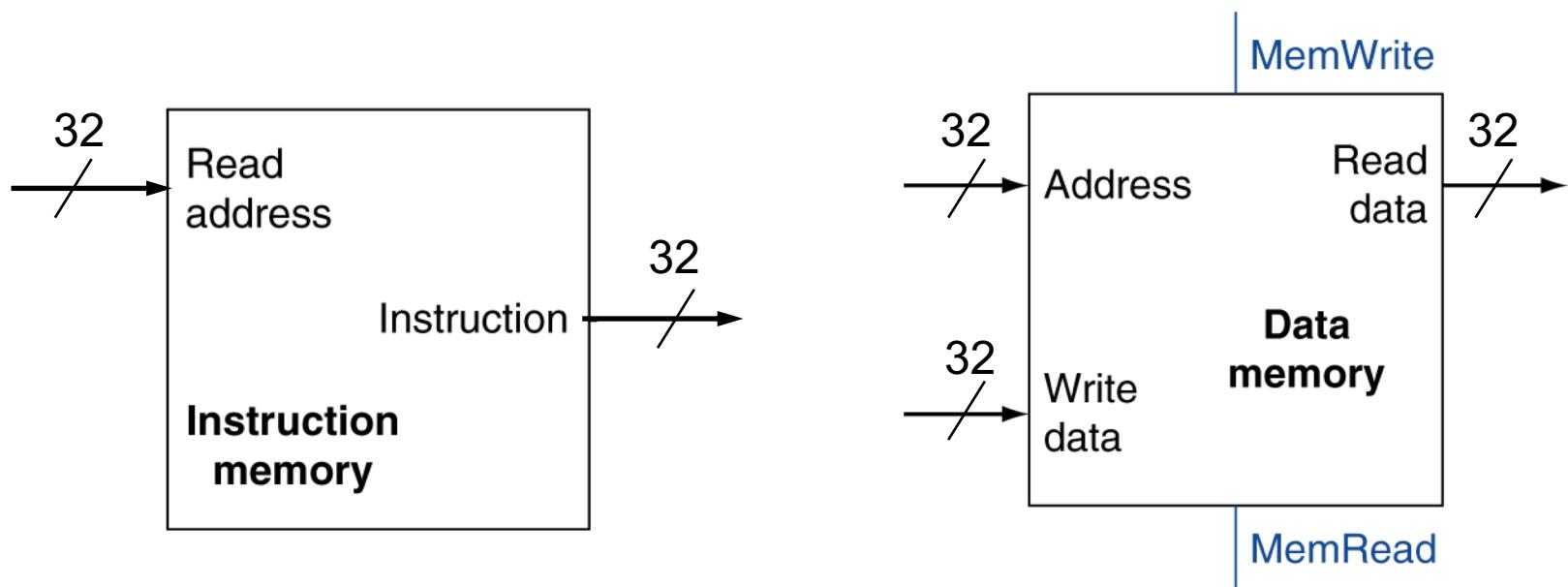


# Memory – DRAM

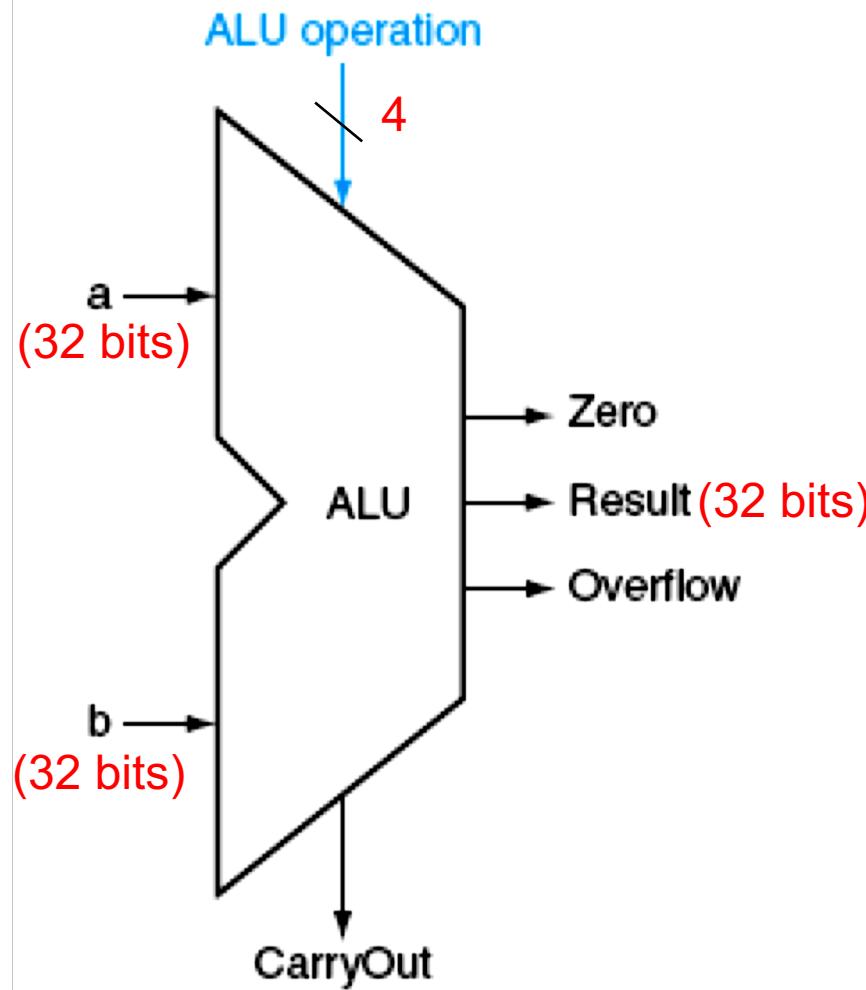
- Typical memory organization
  - Typical access time: 5 – 10 times more than SRAM



# Memory in MIPS

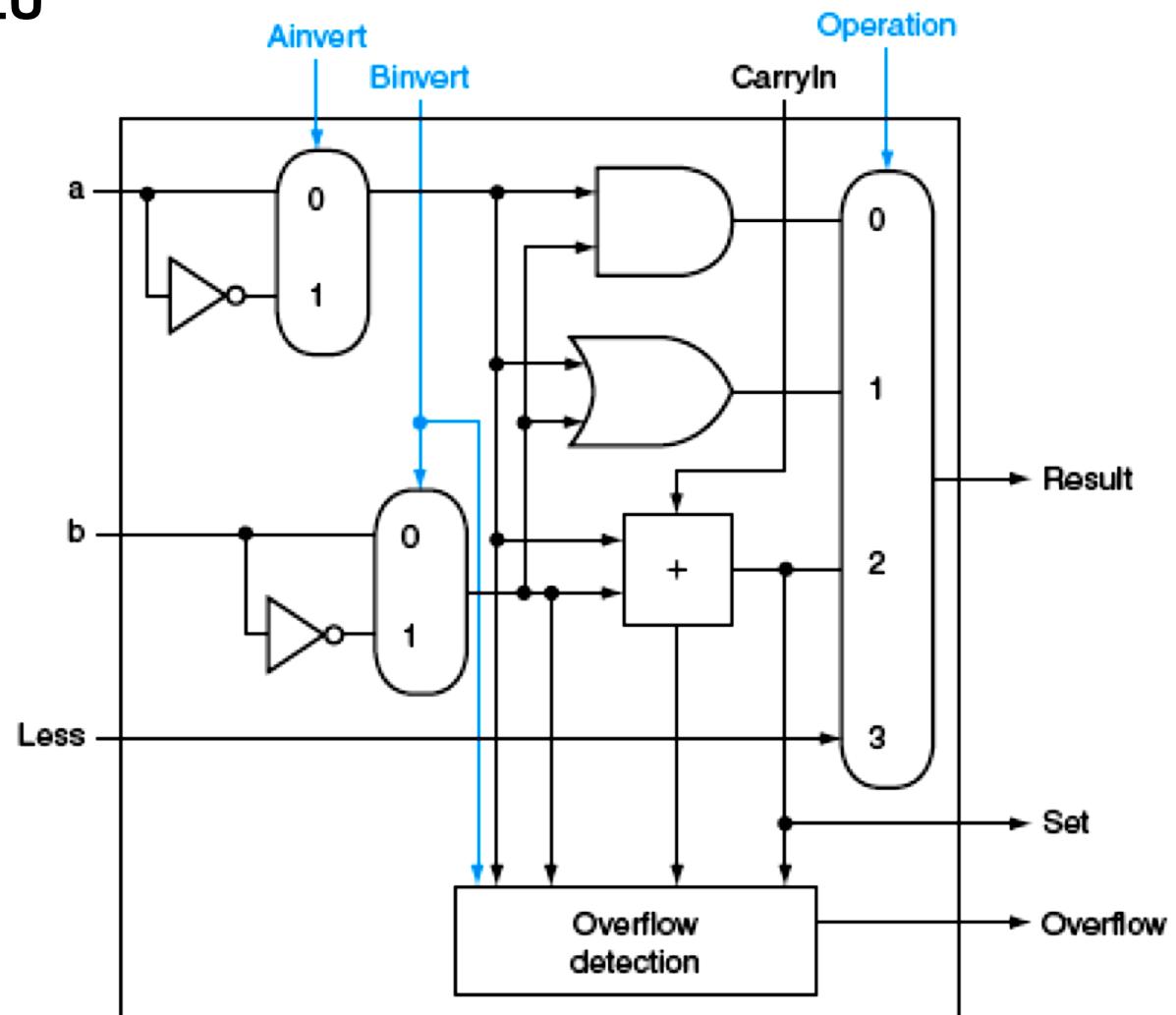


# Arithmetic Logic Unit (ALU)



# Arithmetic Logic Unit (ALU)

## One-bit of the ALU



# ALU in MIPS

