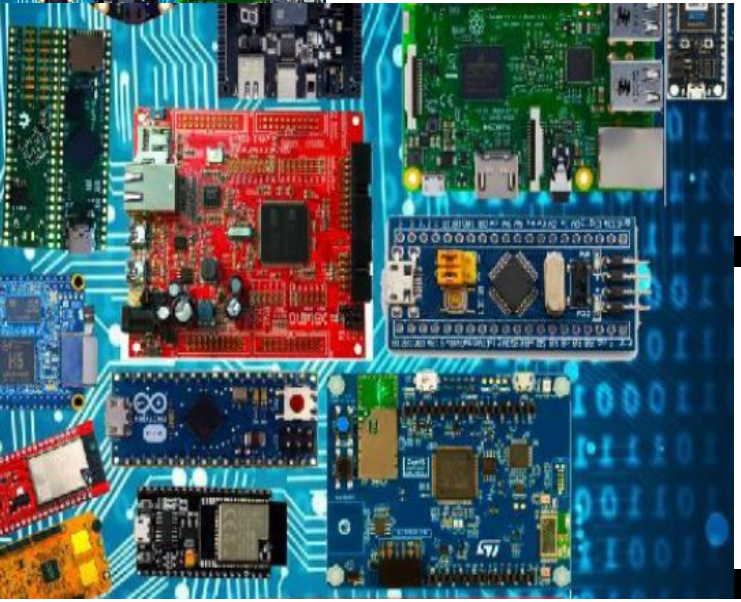


Computer Architecture & Microprocessor System

COMBINATIONAL LOGIC DESIGN

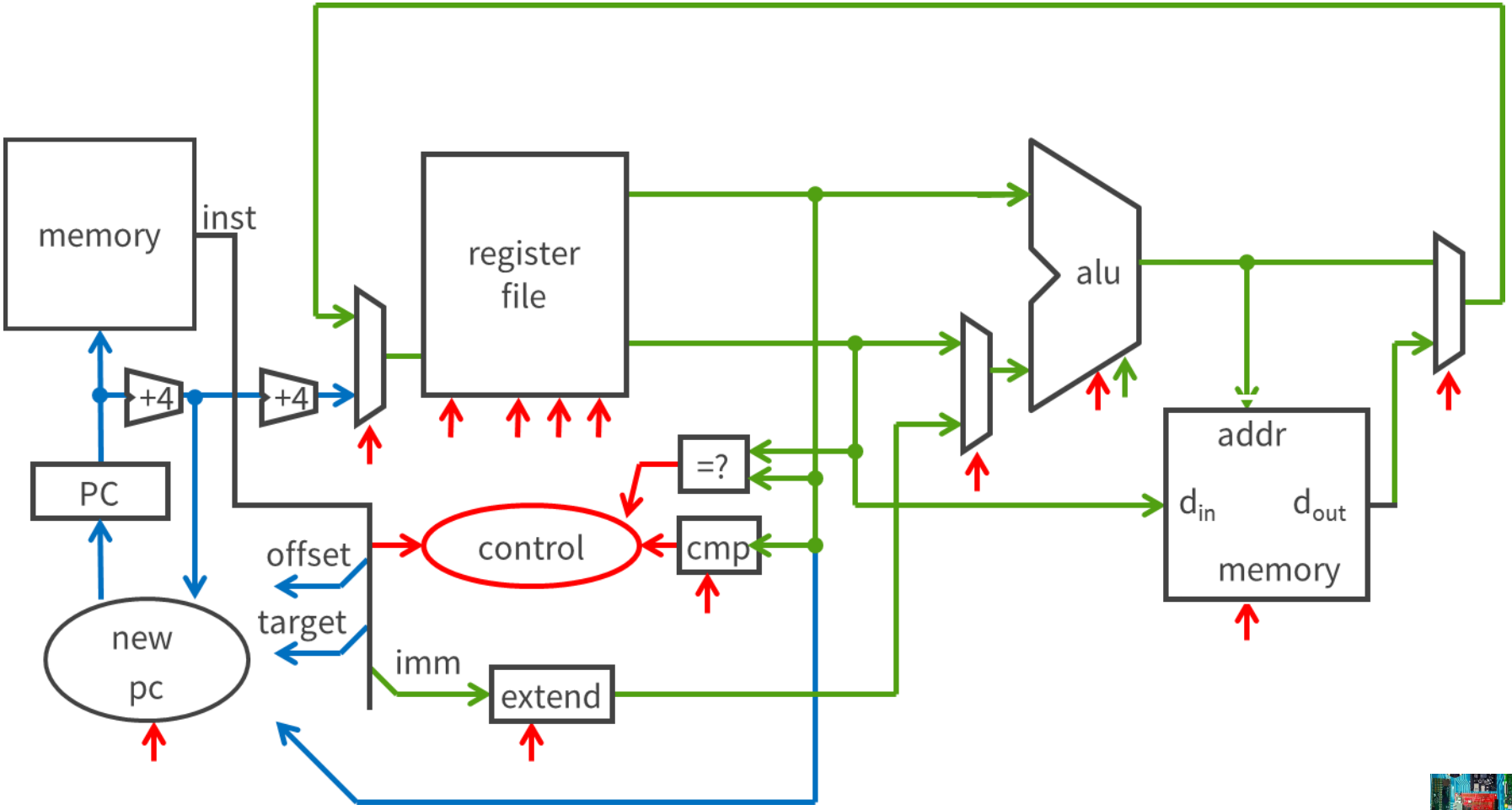
Dennis A. N. Gookyi

CONTENTS

- Combinational Logic Design

BIG PICTURE: BUILDING A PROCESSOR

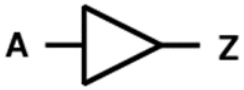
- Single cycle processor



COMMON LOGIC GATES

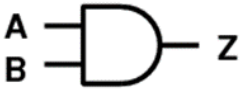
- Basic Logic gates

Buffer



A	Z
0	0
1	1

AND



A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

XOR

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

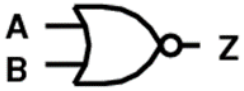
Inverter

A	Z
0	1
1	0

NAND

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

NOR



A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

XNOR

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

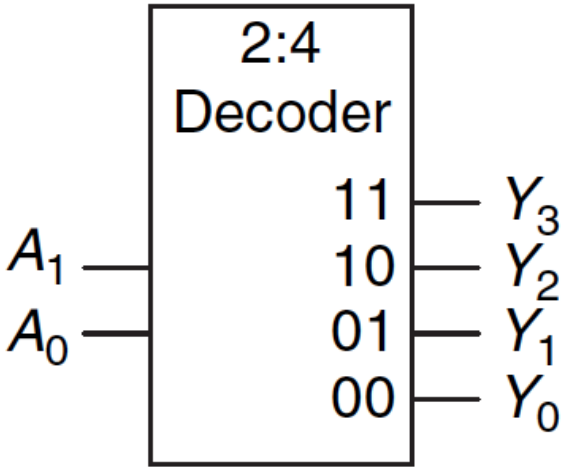
COMBINATIONAL BUILDING BLOCKS

- Combinational logic is often grouped into larger building blocks to build more complex systems
- Hides the unnecessary gate-level details to emphasize the function of the building block
- We now examine:
  - Decoder
  - Multiplexer
  - Full adder
  - PLA (Programmable Logic Array)

DECODER

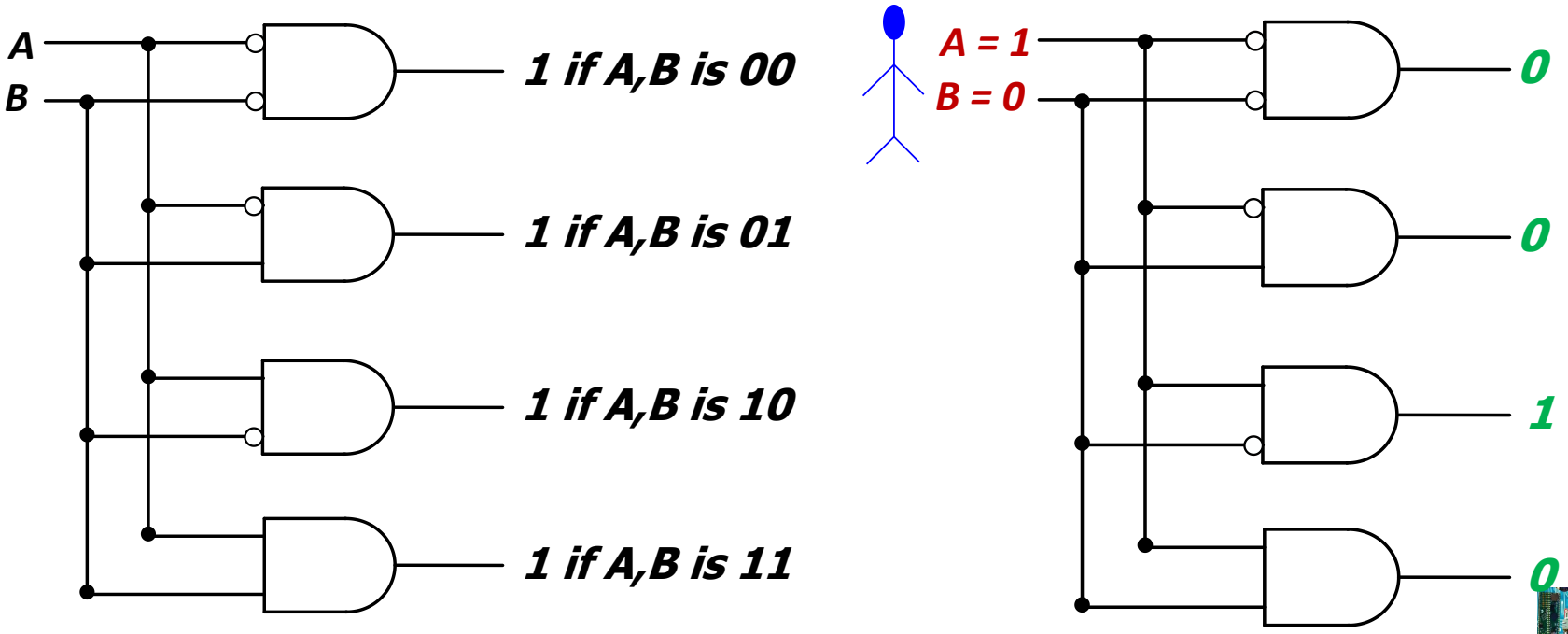
- "Input pattern detector"
- n inputs and 2^n outputs
- Exactly one of the outputs is 1 and all the rest are 0s
- The output that is logically 1 is the output corresponding to the input pattern that the logic circuit is expected to detect
- Example: 2-to-4 decoder

A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



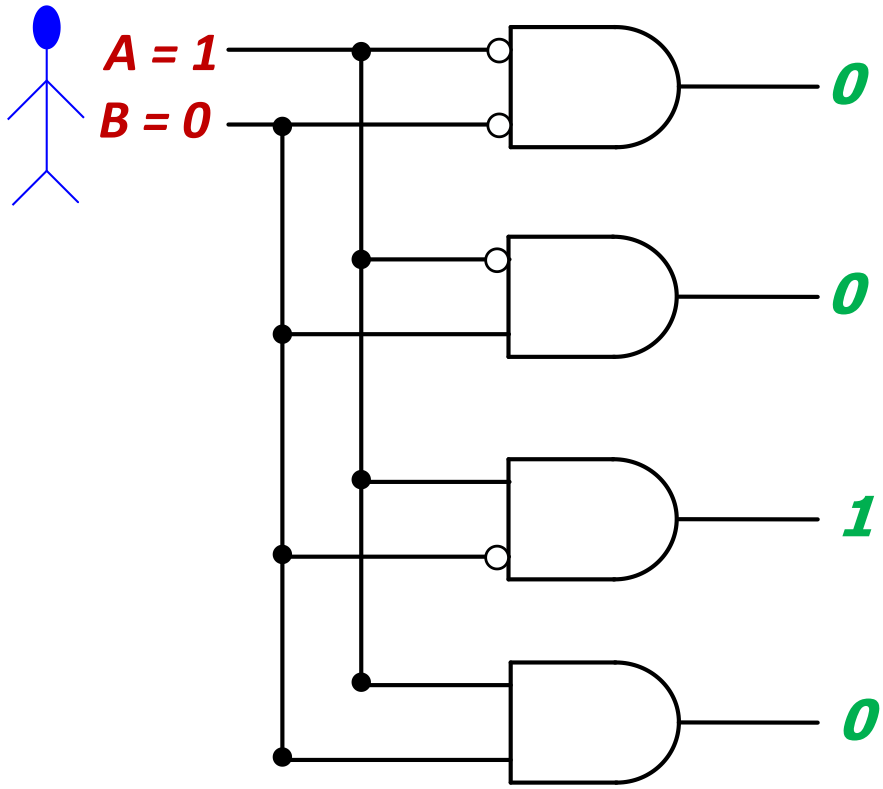
DECODER

- n inputs and 2^n outputs
- Exactly one of the outputs is 1 and all the rest are 0s
- The output that is logically 1 is the output corresponding to the input pattern that the logic circuit is expected to detect



DECODER

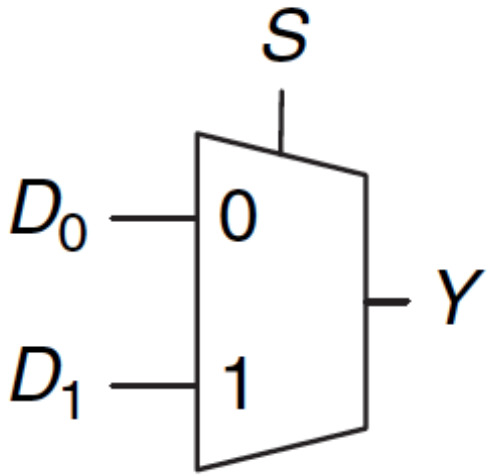
- The decoder is useful in determining how to interpret a bit pattern
  - **It could be the address of a location in memory, that the processor intends to read from**
  - **It could be an instruction in the program and the processor needs to decide what action to take (based on *instruction opcode*)**



MULTIPLEXER

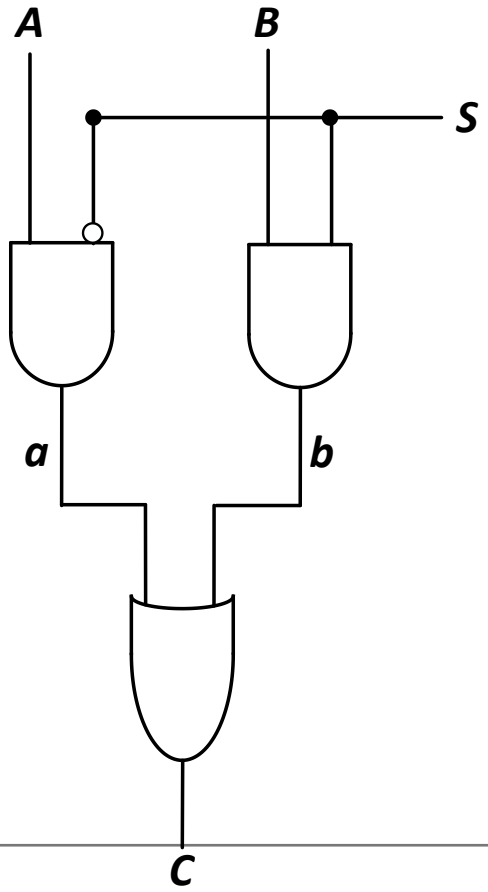
- Selects one of the N inputs to connect it to the output
  - Based on the value of a $\log_2$N-bit control input called select
- Example: 2-to-1 MUX

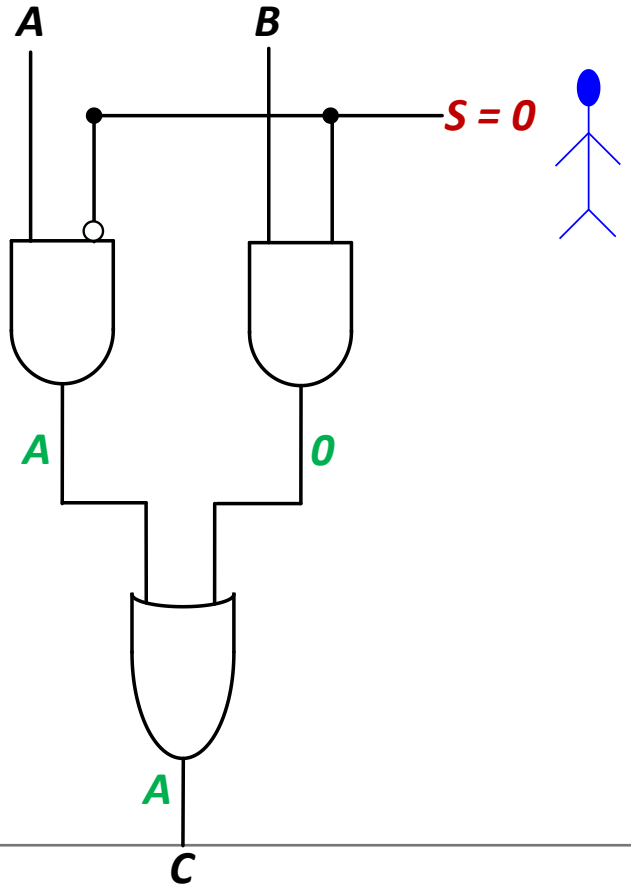
S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



MULTIPLEXER

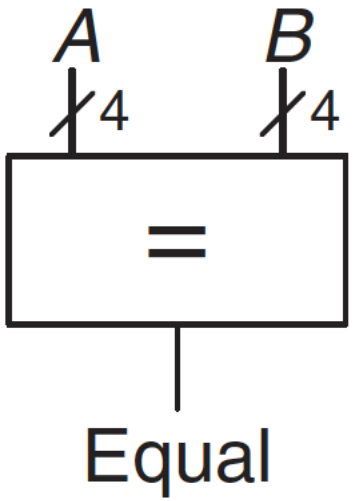
- Selects one of the N inputs to connect it to the output
  - Based on the value of a $\log_2$N-bit control input called select
- Example: 2-to-1 MUX

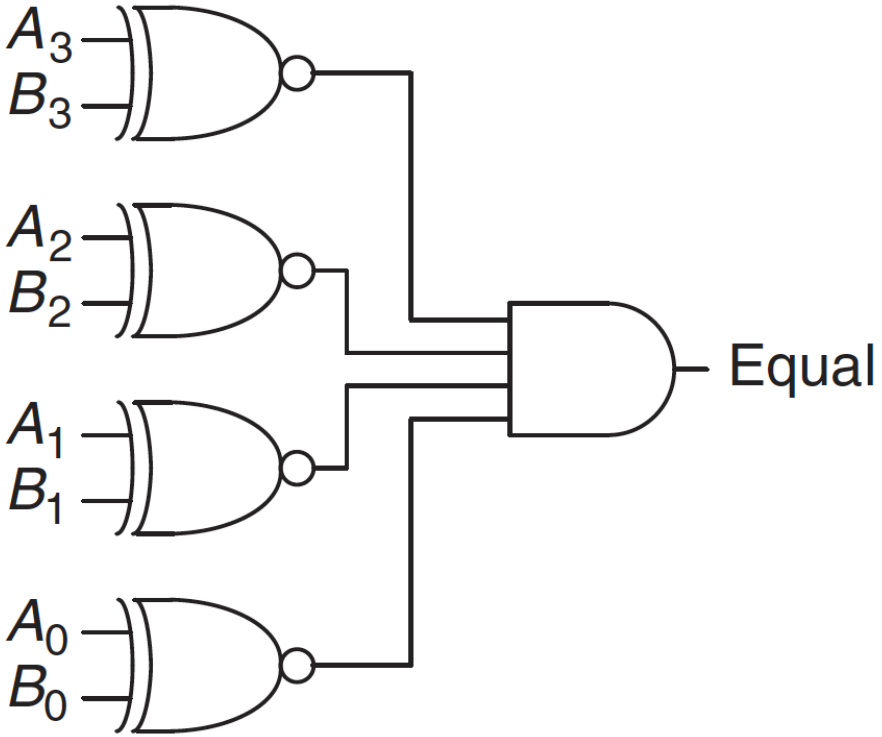




COMPARATOR

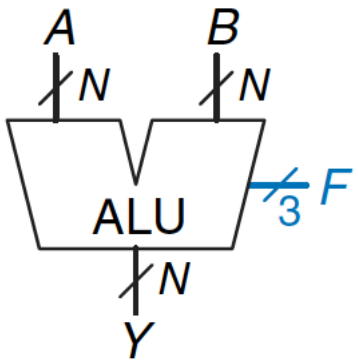
- Equality checker (compare if equal)
- Checks if two N-input values are exactly the same
- Example: 4-bit Comparator





ARITHMETIC LOGIC UNIT (ALU)

- Combines a variety of arithmetic and logical operations into a single unit (that performs only one function at a time)
- Usually denoted with this symbol:



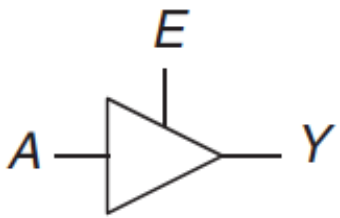
ALU Symbol

ALU Operations

$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\overline{B}$
101	A OR $\overline{B}$
110	A – B
111	SLT

TRI-STATE BUFFER

- A tri-state buffer enables gating of different signals onto a wire
- A tri-state buffer acts like a switch



E	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

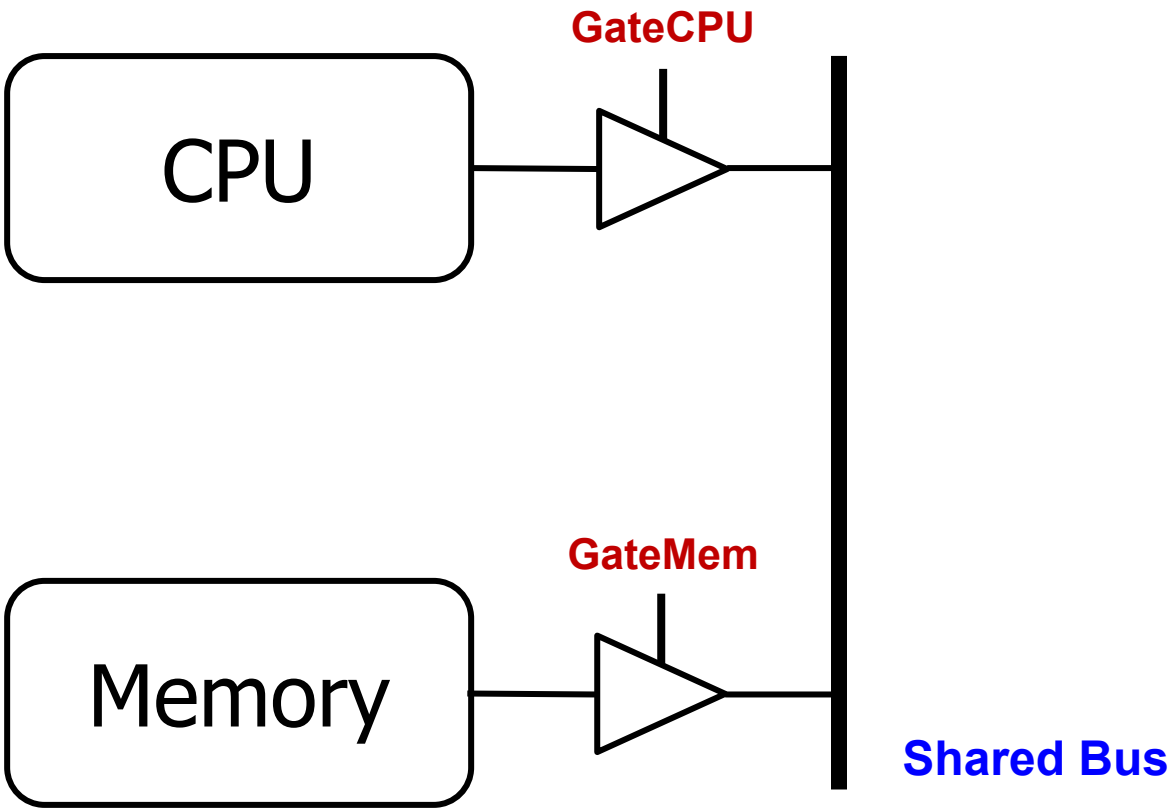
- Floating signal (Z): Signal that is not driven by any circuit
  - Open circuit, floating wire

TRI-STATE BUFFER

- Use of tri-state buffers
- Imagine a wire connecting the CPU and memory
  - At any time only the CPU or the memory can place a value on the wire, both not both
  - You can have two tri-state buffers: one driven by CPU, the other memory; and ensure at most one is enabled at any time

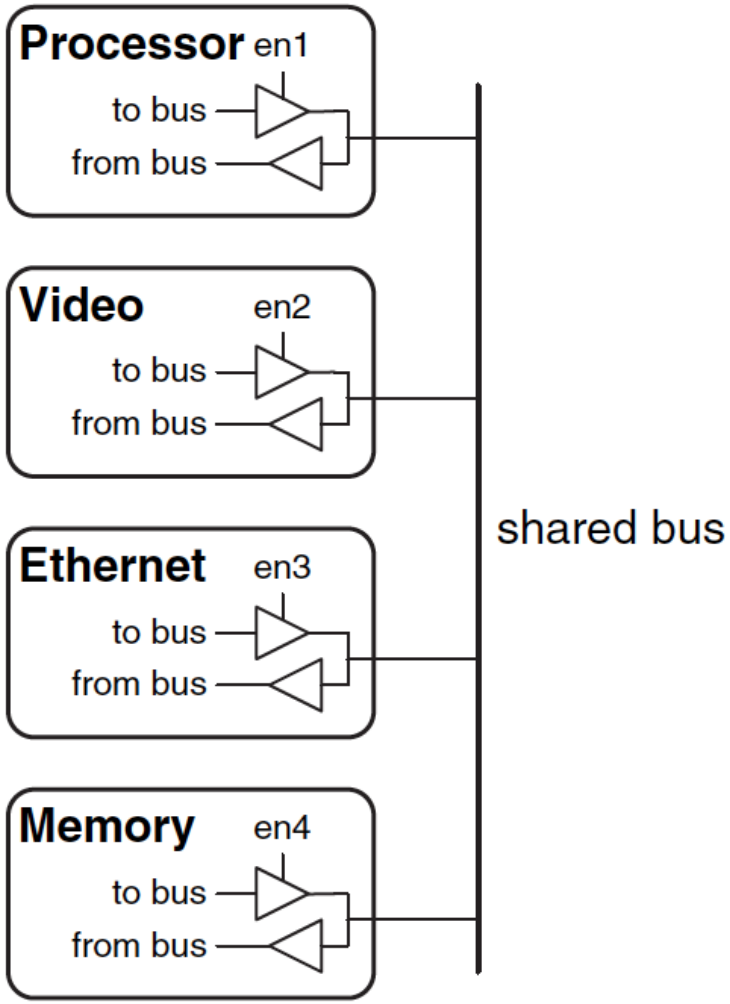
TRI-STATE BUFFER

- Use of tri-state buffers



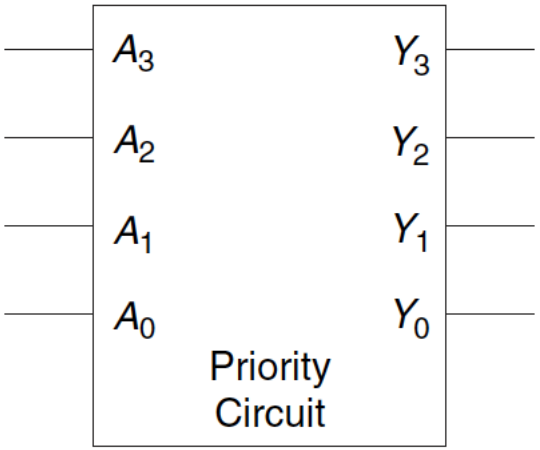
TRI-STATE BUFFER

- Use of tri-state buffers



PRIORITY CIRCUIT

- Inputs: "Requestors" with priority levels
- Outputs: "Grant" signal for each requestor
- Example 4-bit priority circuit
- Real life example: Imagine a bus requested by 4 processors



A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0