ECE 2300
Digital Logic & Computer Organization

Fall 2016

More Combinational Building Blocks

Cornell University

Combinational Building Blocks

- More complex functions built from basic gates
  - Multiplexers
  - Decoders
  - Exclusive OR (XOR)
  - Comparators
  - Priority encoders
  - Tristate drivers
- Typically tens to hundreds of transistors
  - *Medium Scale Integration (MSI)*
- Building blocks for digital systems

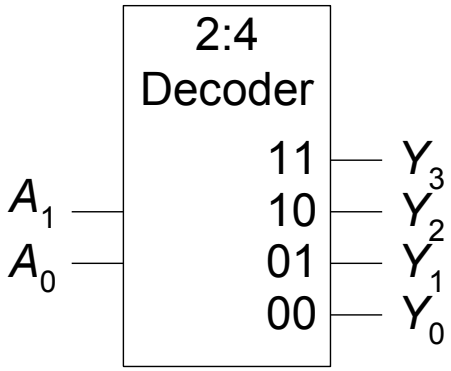
Multiplexer ("mux")

- Connects one of *n* inputs to the output
- Useful when multiple data sources need to be routed to a single destination
  - Example: select 1-of-n *registers* as input to the *ALU*

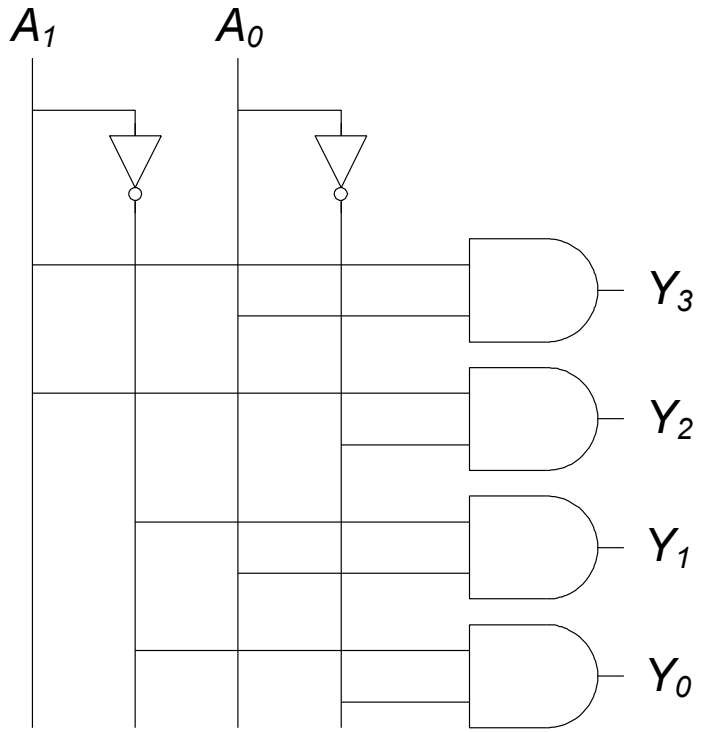
Decoder

- n inputs, 2^n outputs
- Each output corresponds to a unique input value
- At most one output asserted at a time

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



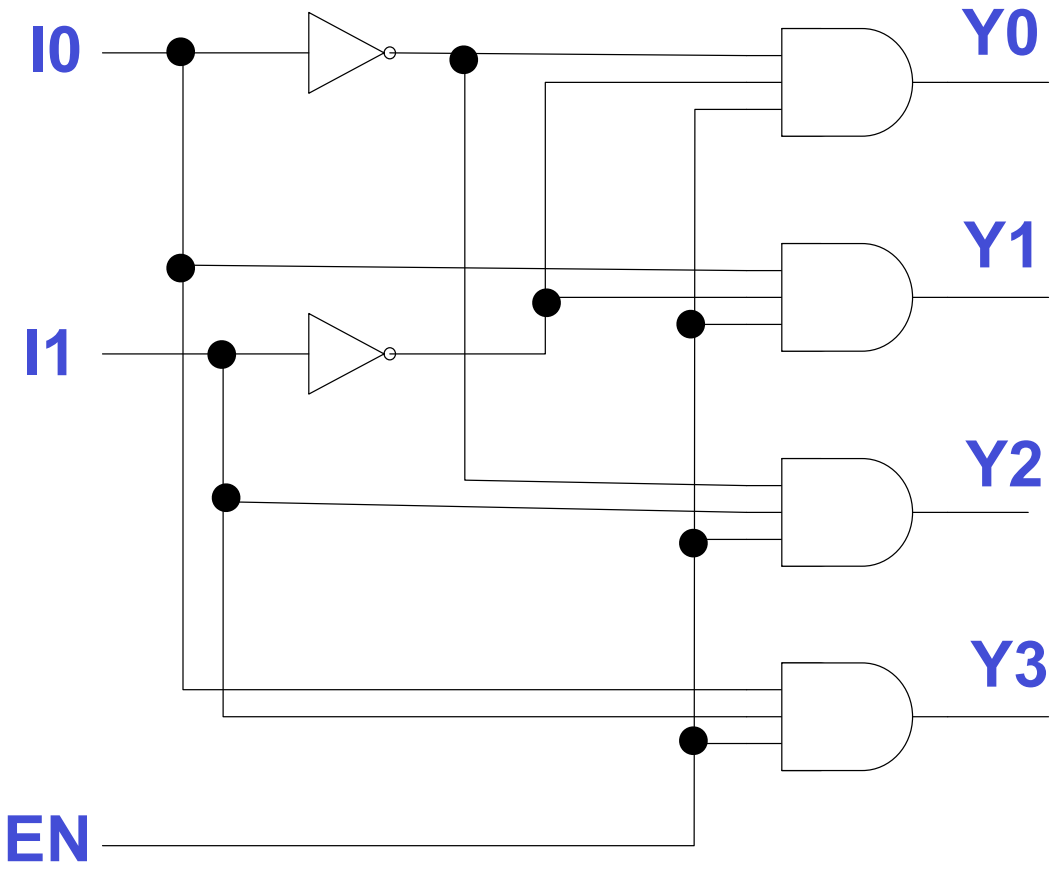
$Y_0 = A_1' \bullet A_0'$

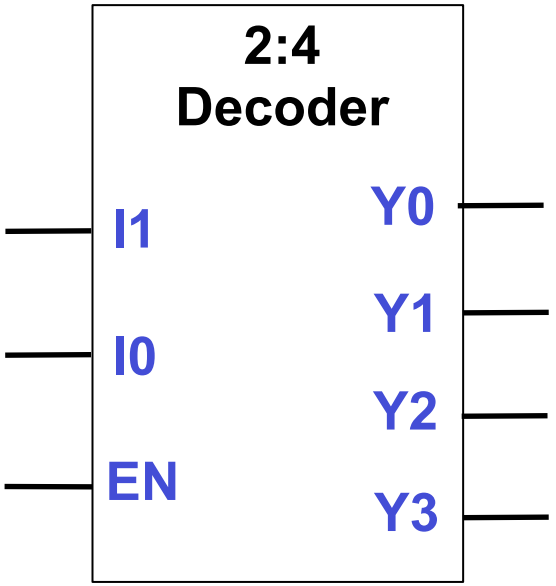
$Y_1 = A_1' \bullet A_0$

$Y_2 = A_1 \bullet A_0'$

$Y_3 = A_1 \bullet A_0$

2-to-4 Decoder with Enable



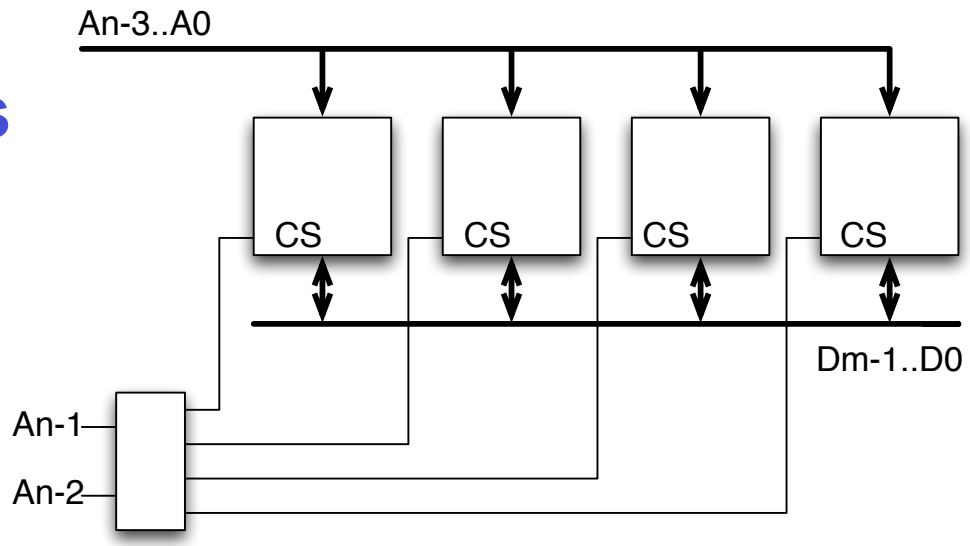


Example Decoder Applications

- **Microprocessor instruction decoding**
  - Select the appropriate arithmetic operation depending on the decoded instruction type
- **Memories**
  - Select one of n banks of memory chips to read/write
  - Select one of n rows of memory cells within a memory chip to read/write
- **Input/output systems**
  - Select one of n I/O devices to read/write

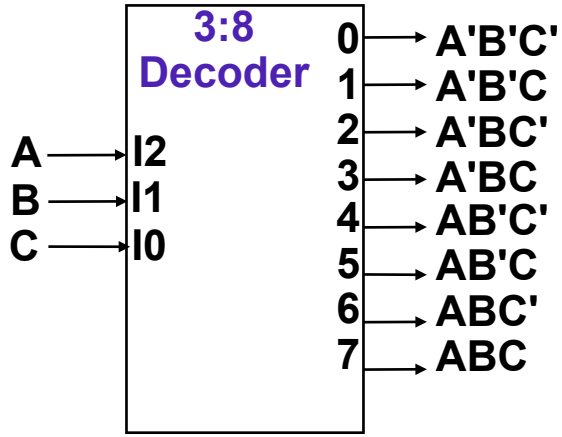
Example: Memory Decoding

- 2^n memory locations spread across 4 chips
  - 2^{n-2} locations per chip
- 2 address bits select the chip
  - Chip Select (CS)
  - Outputs of chips that are not selected are in Hi-Z state (later)
- n-2 address bits select row within the chip



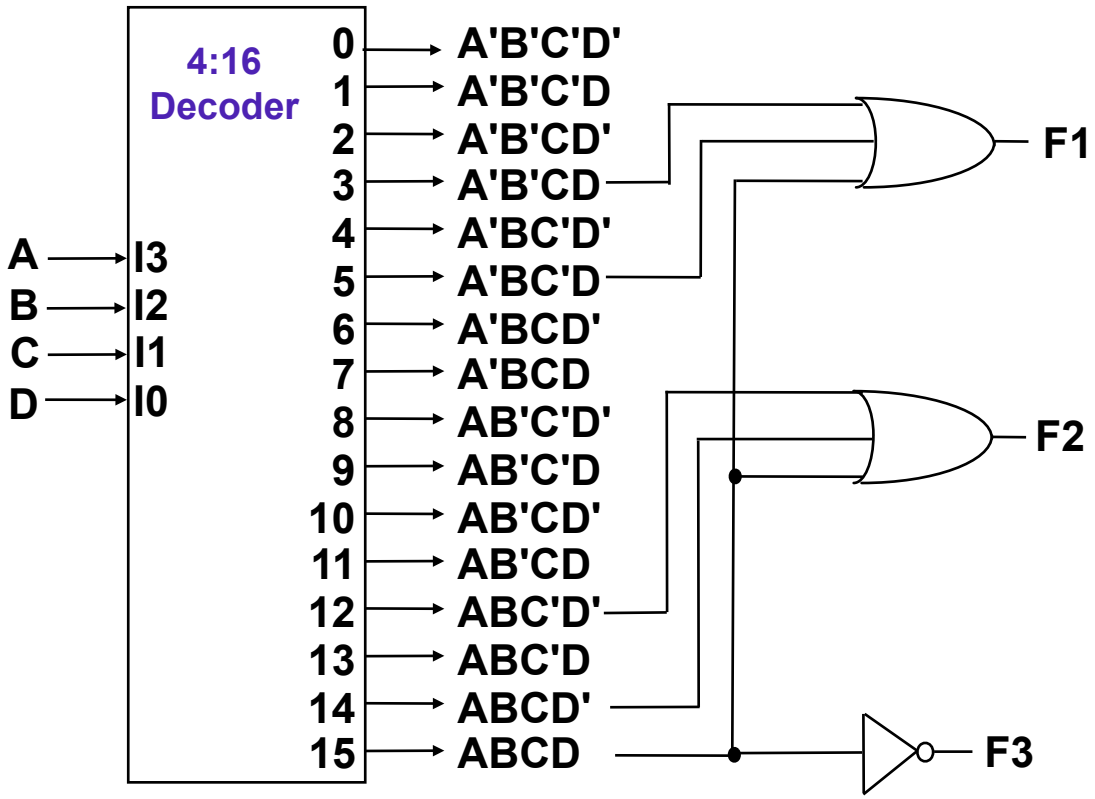
Logic Functions Using Decoders

- n:2^n decoder can be used to implement any function of n variables
  - Connect variables to inputs
  - Appropriate minterms summed using extra gates to form the function



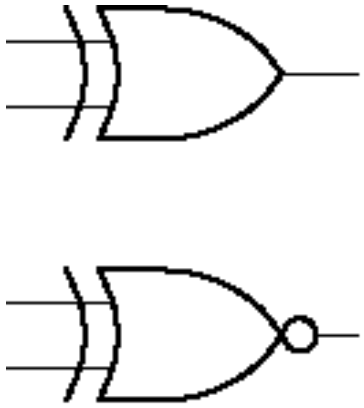
Logic Functions Using Decoders

- F1 = A’B’CD + A’BC’D + ABCD
- F2 = ABC’D’ + ABC
- F3 = A’+B’+C’+D’



XOR Gate

X	Y	X⊕Y	(X⊕Y)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

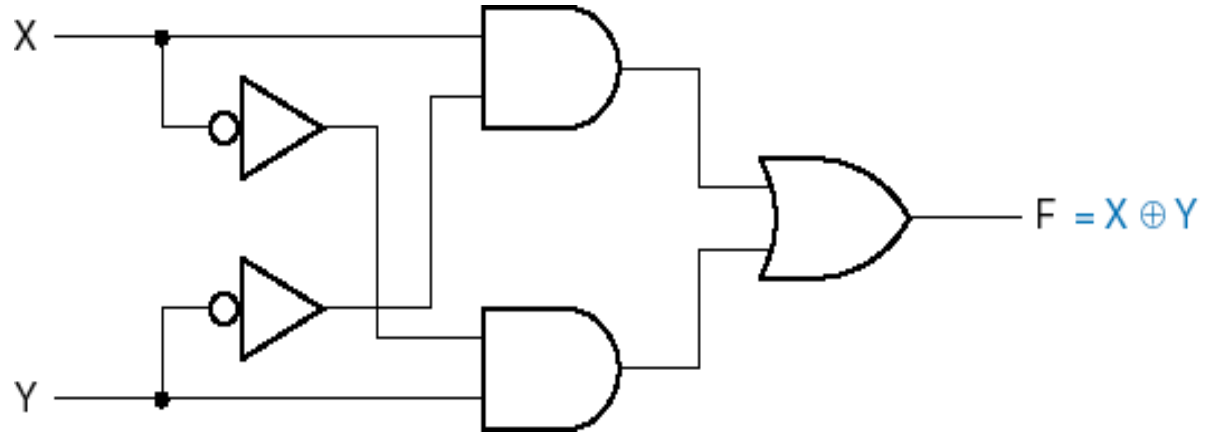


- Same as an OR gate, except when inputs are 1
- Used for comparisons, error checking

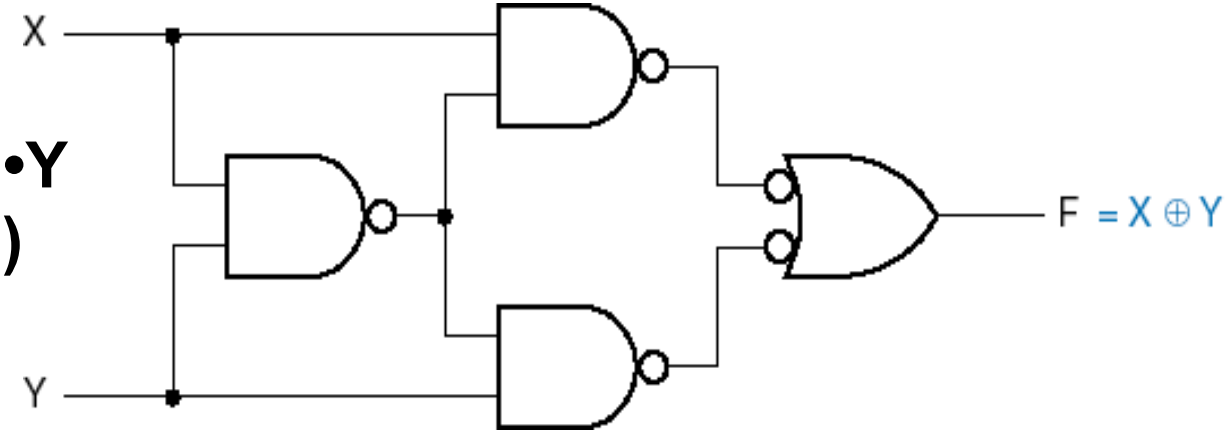
- XNOR: complemented version of XOR

XOR Gate Implementations

$F = X \cdot Y' + X' \cdot Y$

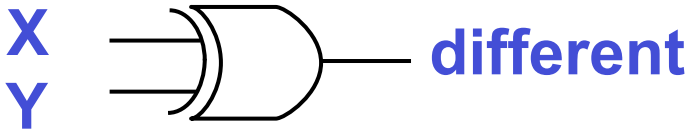


$F = X \cdot Y' + X' \cdot Y$
$= X \cdot X' + X \cdot Y' + Y \cdot Y' + X' \cdot Y$
$= X \cdot (X' + Y') + Y \cdot (X' + Y')$
$= X \cdot (X \cdot Y)' + Y \cdot (X \cdot Y)'$

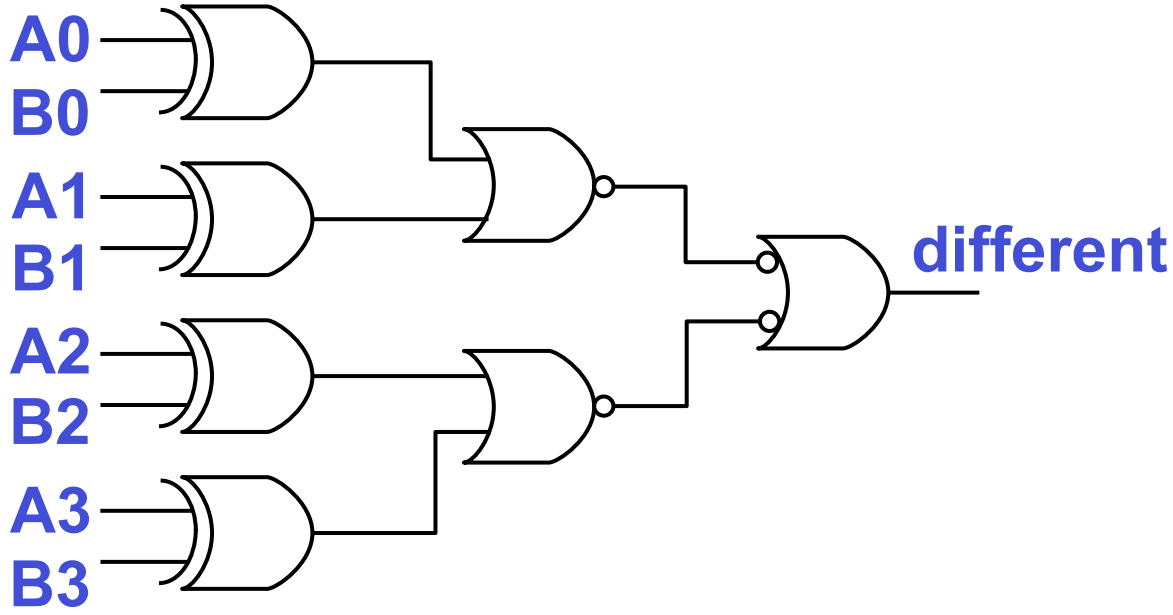


Equality Comparators Using XOR

1-bit comparator

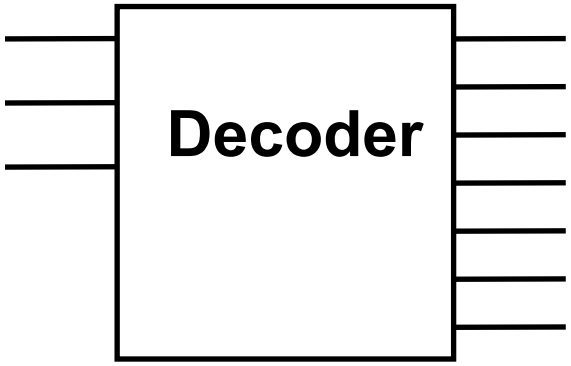


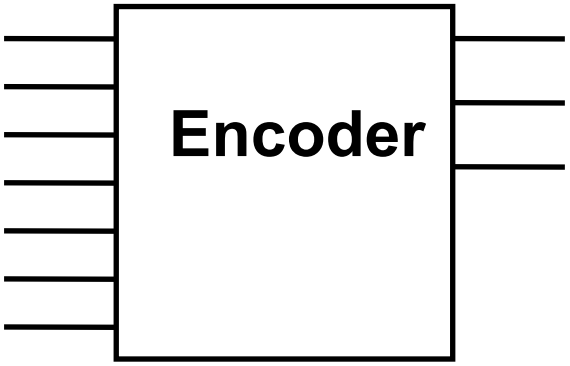
4-bit comparator



Encoders

- Opposite of decoders
- 2^n inputs and n outputs



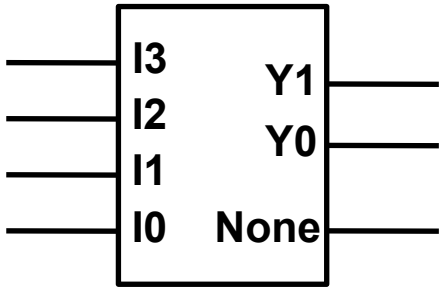


What if multiple inputs are asserted?

Priority Encoder

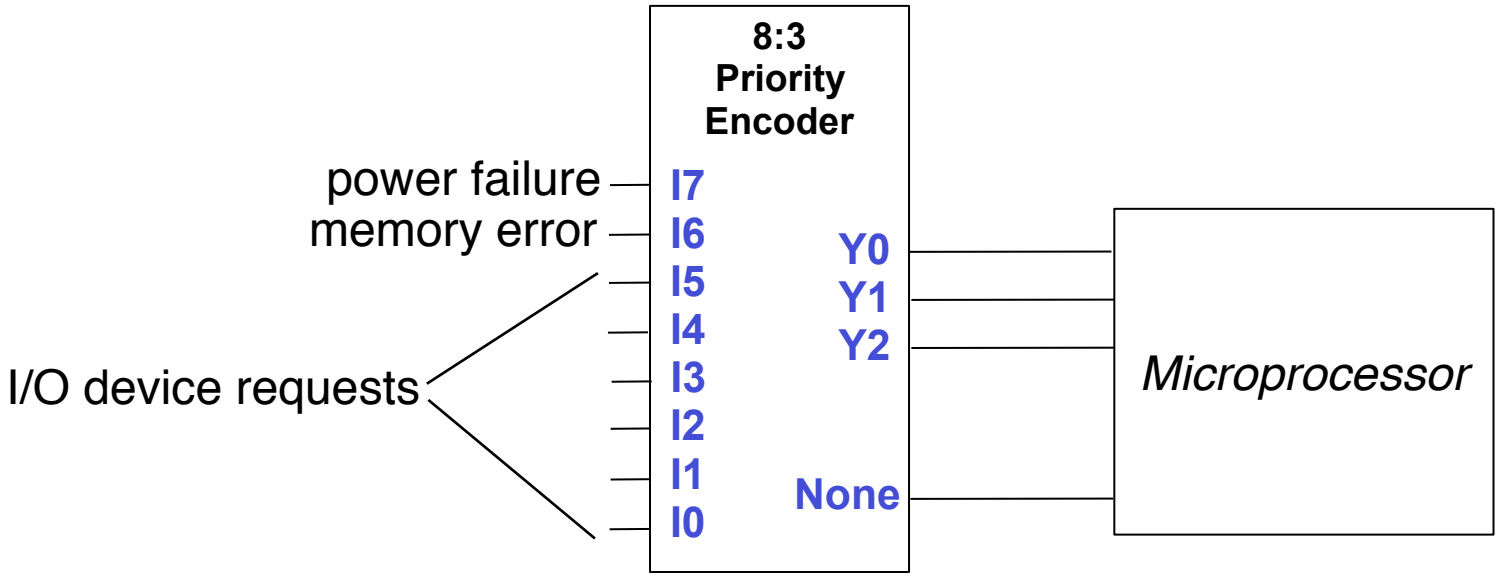
- Highest numbered inputs have priority
- Example: 4-to-2 priority encoder

I3	I2	I1	I0	Y1	Y0	None
1	X	X	X	1	1	0
0	1	X	X	1	0	0
0	0	1	X	0	1	0
0	0	0	1	0	0	0
0	0	0	0	0	0	1



Example: Microprocessor Interrupts

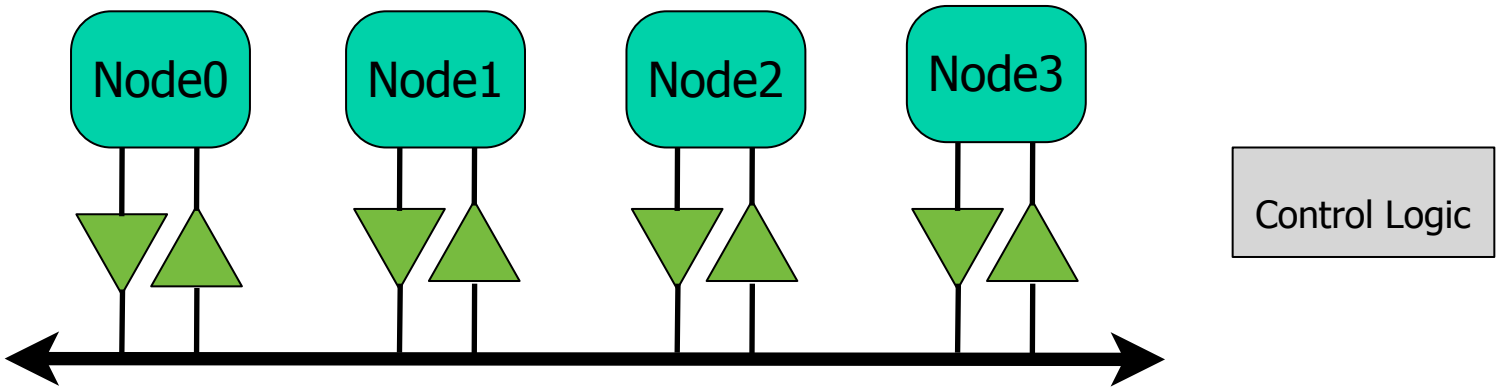
- In order for devices to get service, they *interrupt* the microprocessor
- Most important requests are given priority



Multi-Drop Buses

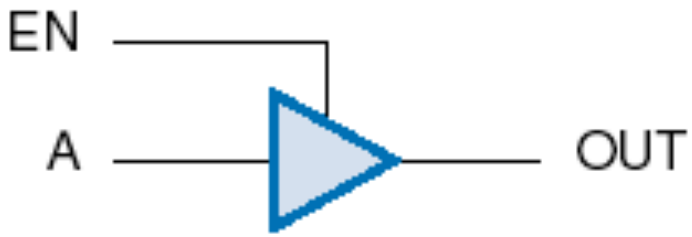
- Multi-drop buses are a low-cost way to communicate data among digital devices

- Components of a multi-drop bus
  - Set of wires to connect the nodes
  - Special circuits for sending and receiving bus data
  - Control logic that decides who is allowed to send



Tri-State Drivers

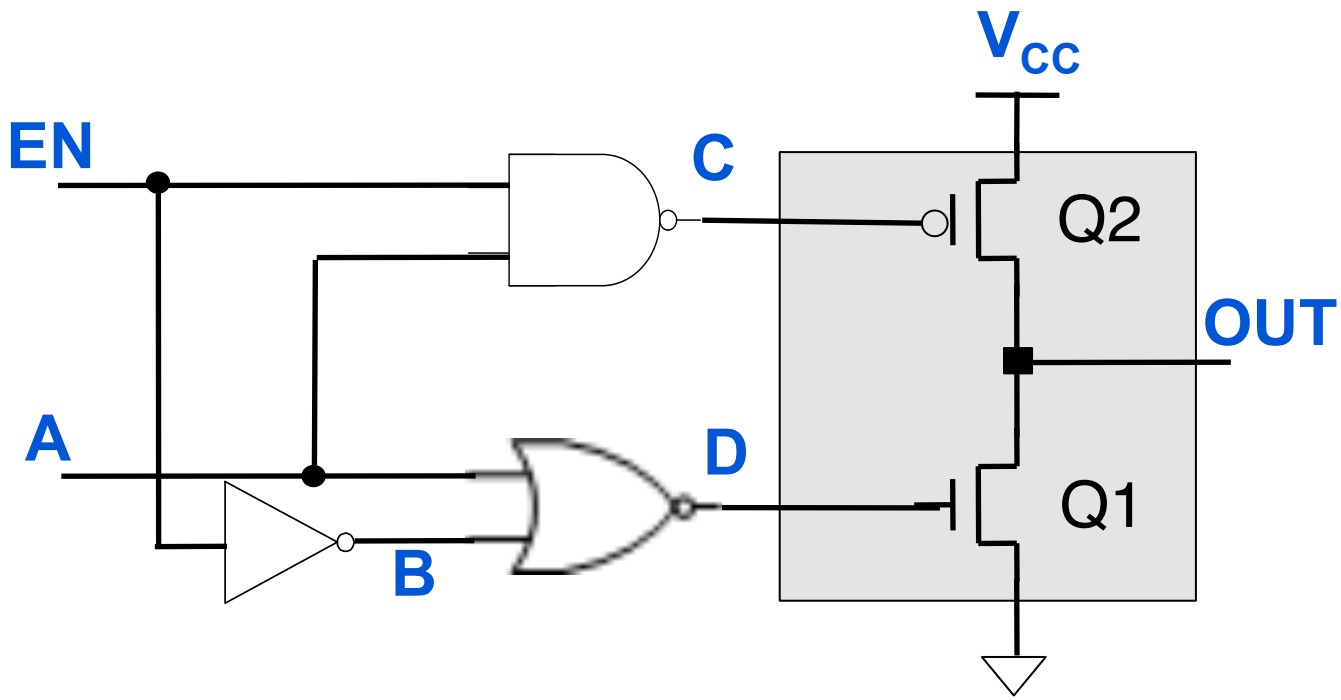
- **Along with 0 and 1, there is a third *Hi-Z* output**
  - **Output "floats" - no connection to supply or ground**



EN	A	OUT
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1

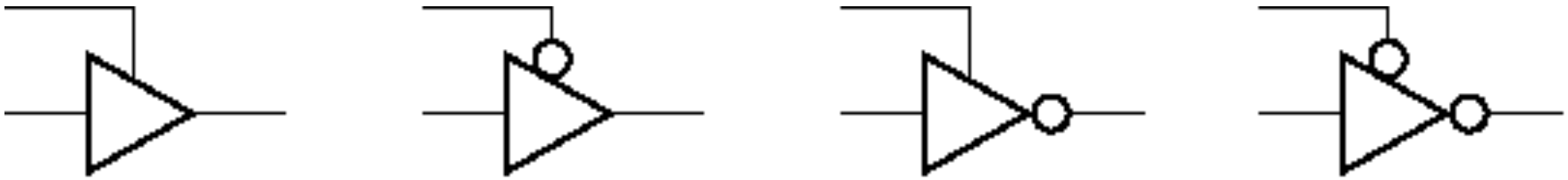
- **Used to drive a multi-drop bus**
  - **Only one node drives the bus, others in Hi-Z state**

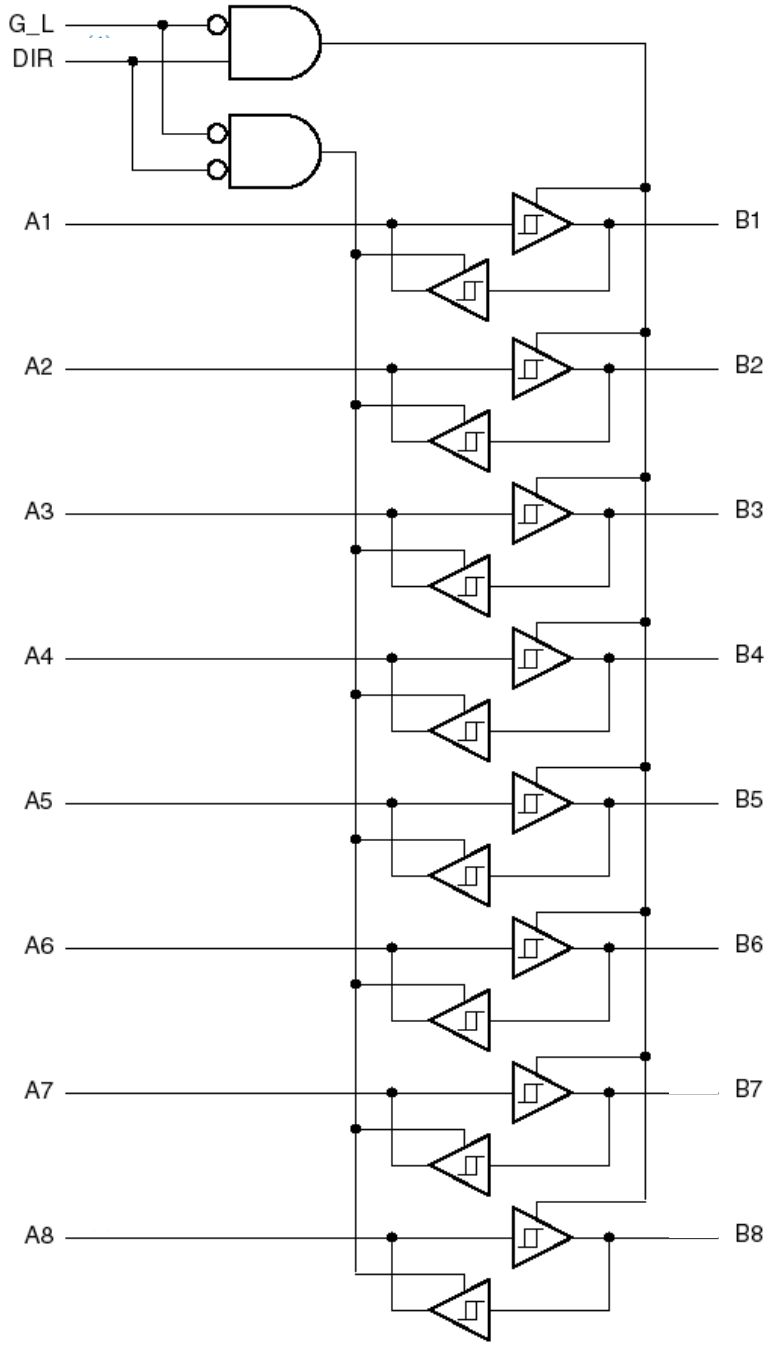
Tri-State Drivers



EN	A	B	C	D	Q1	Q2	OUT
L	L	H	H	L	off	off	Hi-Z
L	H	H	H	L	off	off	Hi-Z
H	L	L	H	H	on	off	L
H	H	L	L	L	off	on	H

Variations of Tri-State Drivers





Transceiver

- Data can flow in either direction

G_L	DIR	Action
0	0	A←B
0	1	A→B
1	X	A,B Hi-Z

Before Next Class

- H&H 3.1-3.2

Next Time

Sequential Logic: Clocks, Latches, Flip-Flops