ECE 2300
Digital Logic & Computer Organization

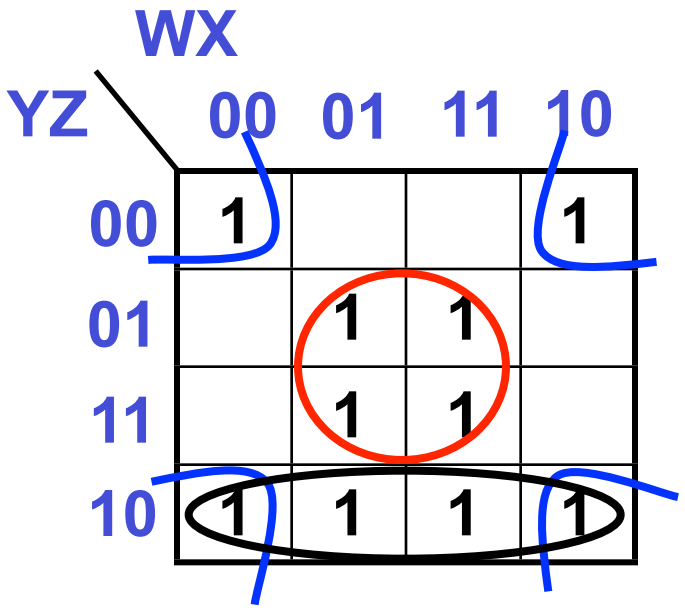
Fall 2016

CMOS Logic
Combinational Building Blocks

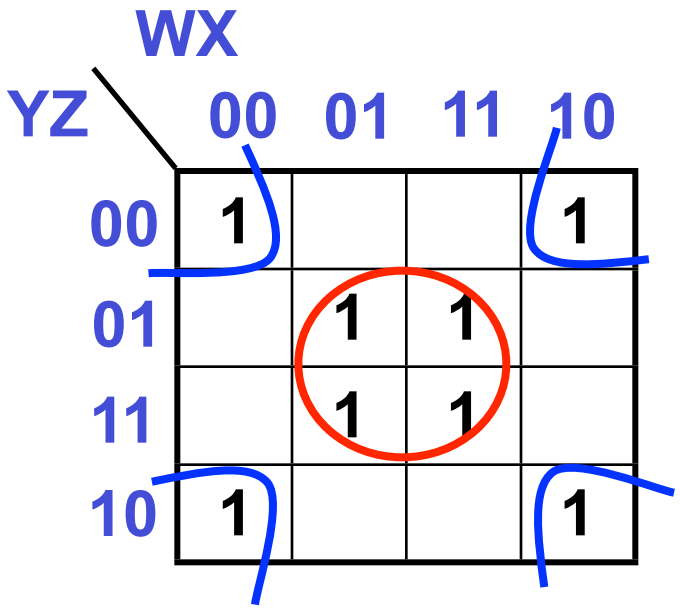
Cornell University

5 Variable Karnaugh Maps

$F=\Sigma_{VWXYZ}(0,2,5,6,7,8,10,13,14,15,16,18,21,23,24,26,29,31)$



V=0

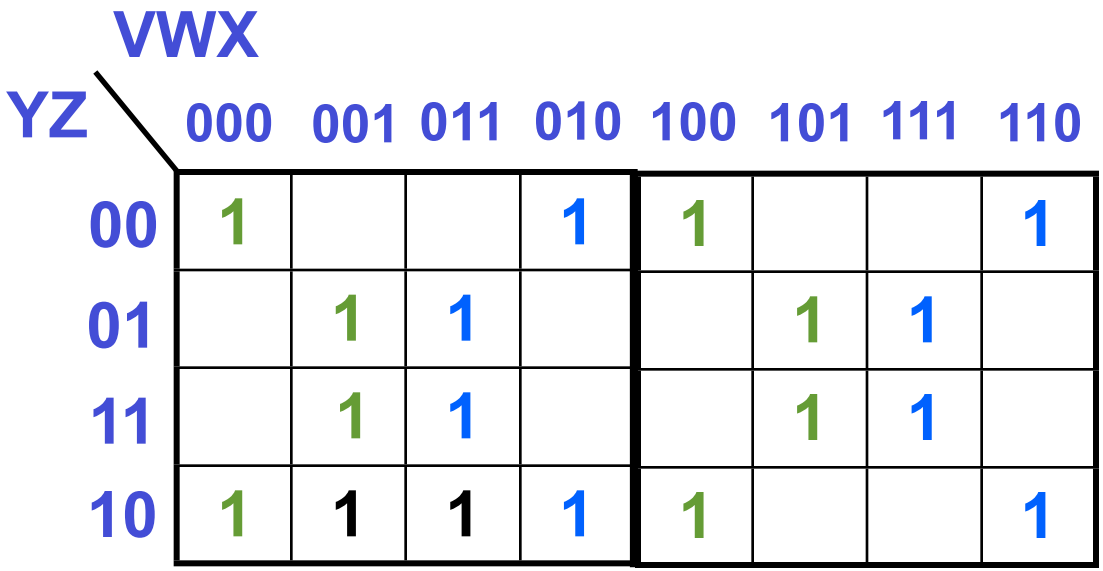


V=1

$F=X'\bullet Z' + X\bullet Z + V'\bullet Y\bullet Z'$

5 Variable *Overlaid* Karnaugh Map

$F=\Sigma_{VWXYZ}(0,2,5,6,7,8,10,13,14,15,16,18,21,23,24,26,29,31)$

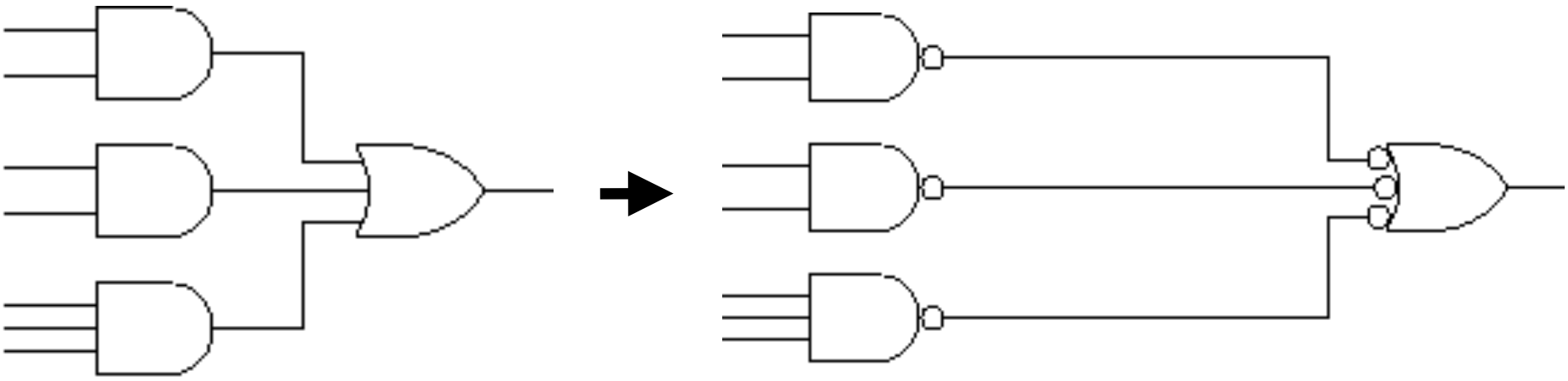


5 Variable *Mirrored* Karnaugh Map

$F=\Sigma_{VWXYZ}(0,2,5,6,7,8,10,13,14,15,16,18,21,23,24,26,29,31)$

YZ \ VWX	000	001	011	010	110	111	101	100
00	1			1	1			1
01		1	1			1	1	
11		1	1			1	1	
10	1	1	1	1	1			1

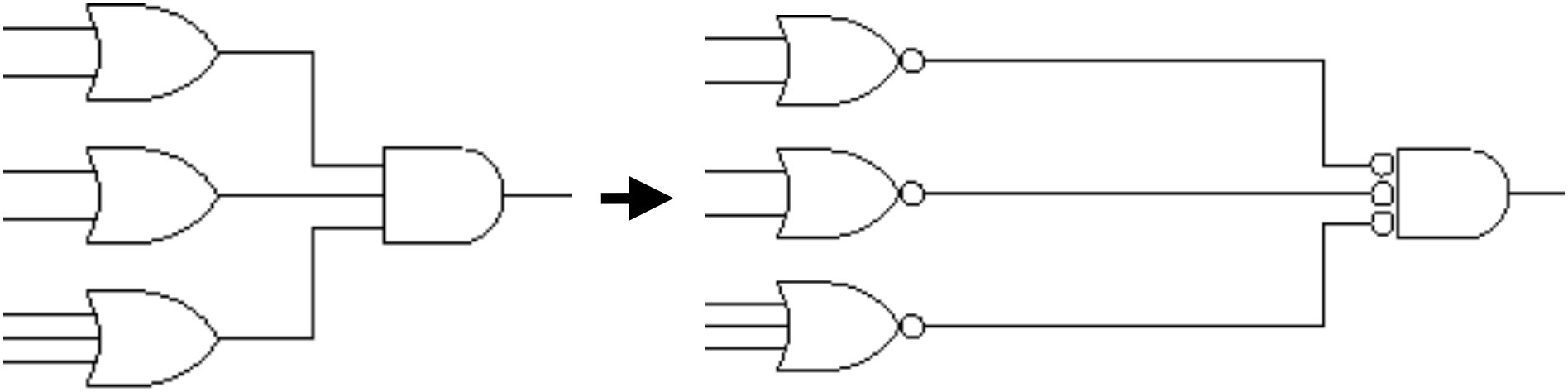
Sum-of-products



AND-OR

NAND-NAND

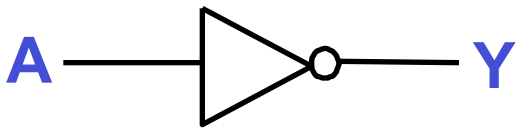
Product-of-sums



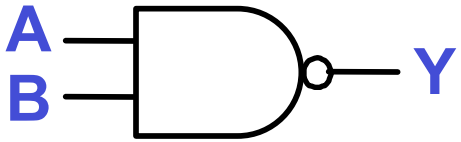
OR-AND

NOR-NOR

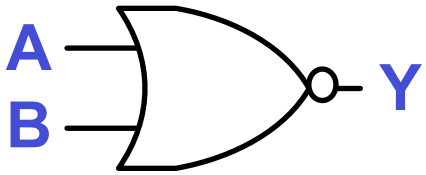
NOT, NAND, and NOR



A	Y
0	1
1	0



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



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

NOT Gate Input & Output Voltages

A	Y
0	1
1	0

→

A	Y
L	H
H	L

→

A	Y
0V	5V
5V	0V

- When the input voltage is *low* (0), the output should be connected to the voltage supply (1)
- When the input voltage is *high* (1), the output should be connected to ground (0)
- Can build a NOT using two types of switches
  - Type 1: Closed when input = 0, open when input = 1
  - Type 2: Closed when input = 1, open when input = 0

NOT Using Switches

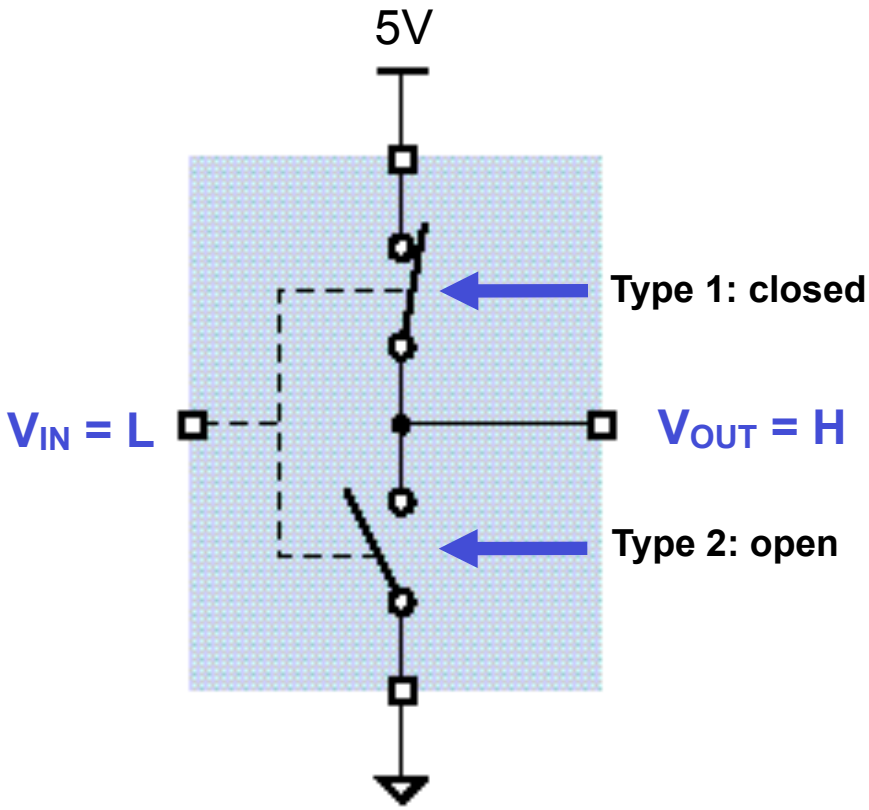
A	Y
0	1
1	0

→

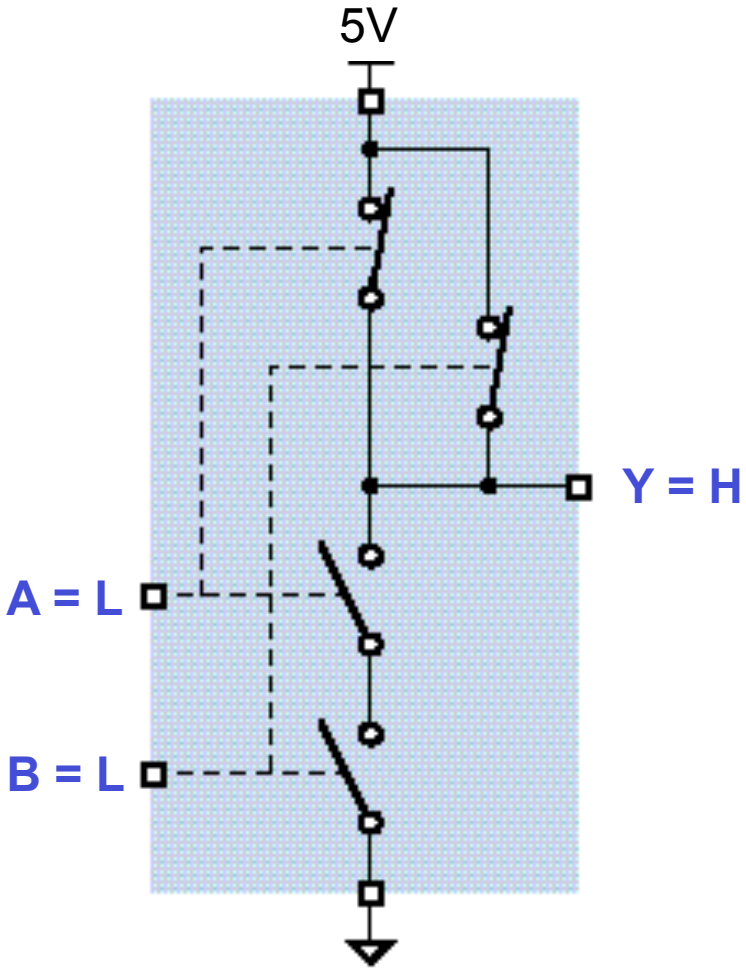
A	Y
L	H
H	L

→

A	Y
0V	5V
5V	0V

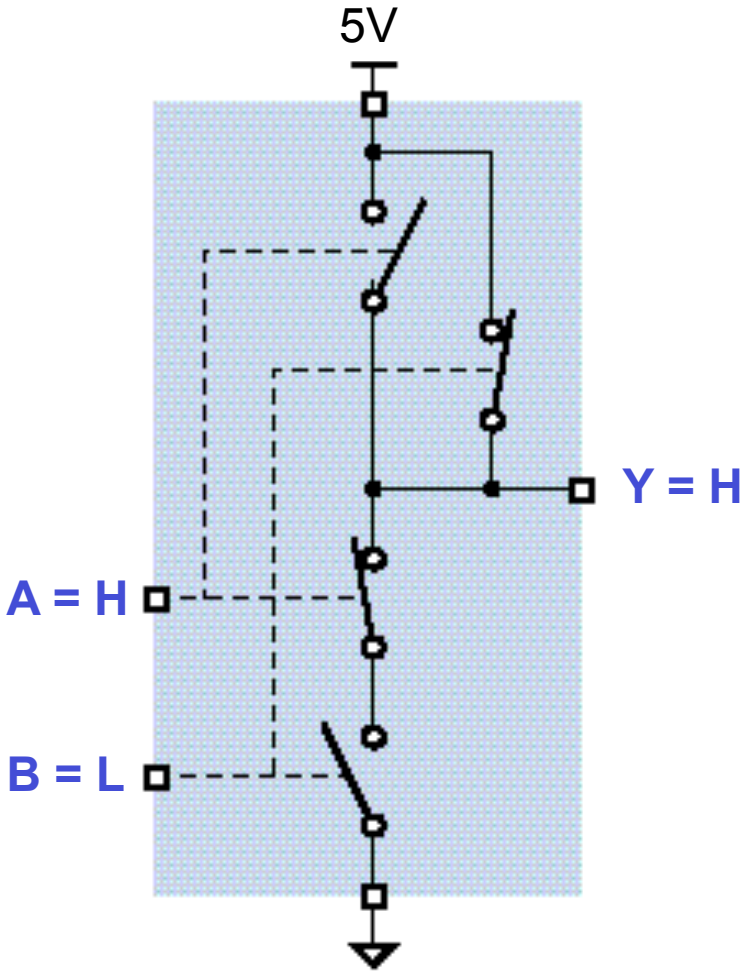


NAND Using Switches



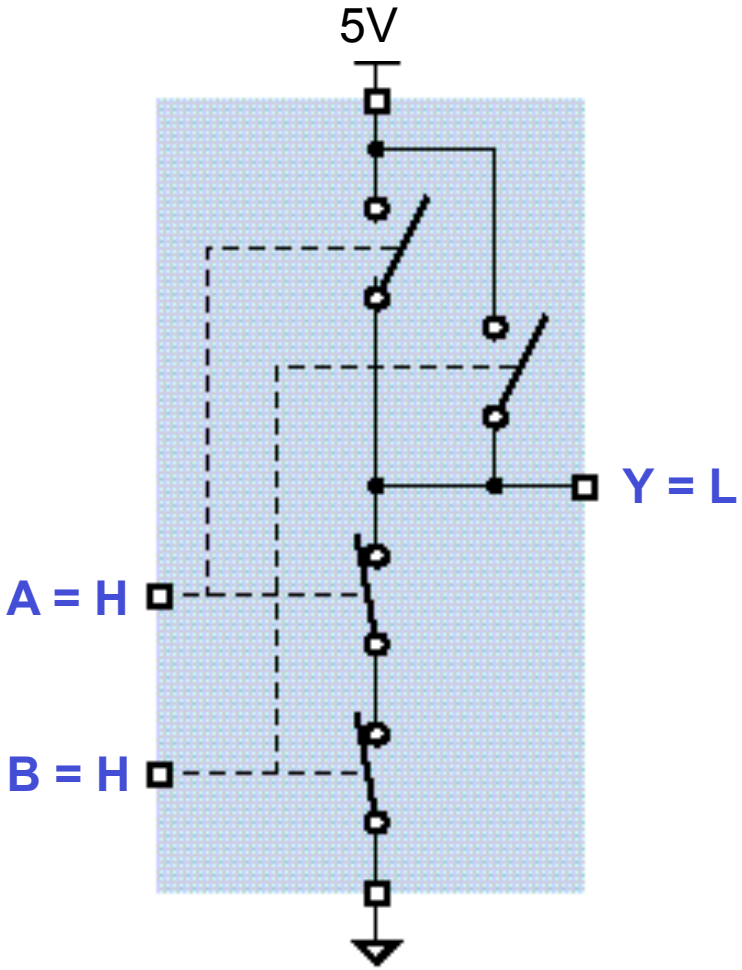
Inputs		Output
A	B	Y
L	L	H

NAND Using Switches



Inputs		Output
A	B	Y
L	L	H
H	L	H

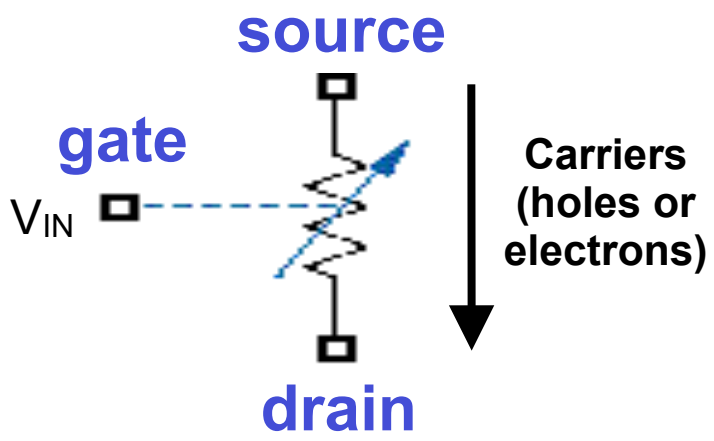
NAND Using Switches



Inputs		Output
A	B	Y
L	L	H
H	L	H
H	H	L

MOS Transistors

- **Metal-Oxide Semiconductor Field-Effect Transistors (MOSFETS)**



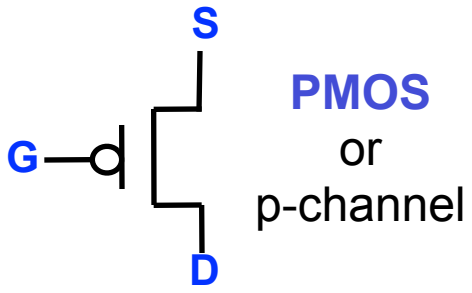
Voltage-controlled resistance

- **Extreme changes in resistance (0 to ∞) make transistors act like switches**

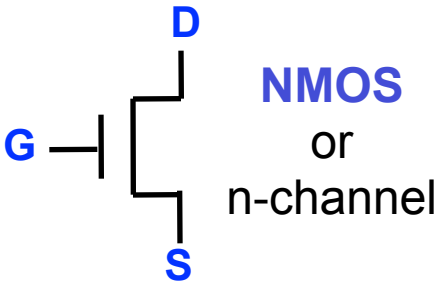
MOS Transistors

- Current flows when ON (*conducting*)
- No current flows when OFF (*not conducting*)

- Type 1 and Type 2 switches



Type 1

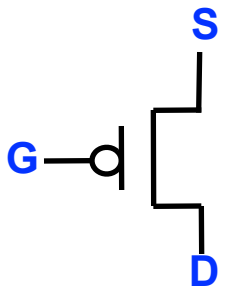


Type 2

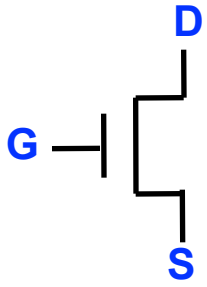
○ = LOW closes the switch

MOS Transistors

- **PMOS**
  - Closed when input is low
  - Open when input is high



- **NMOS**
  - Closed when input is high
  - Open when input is low

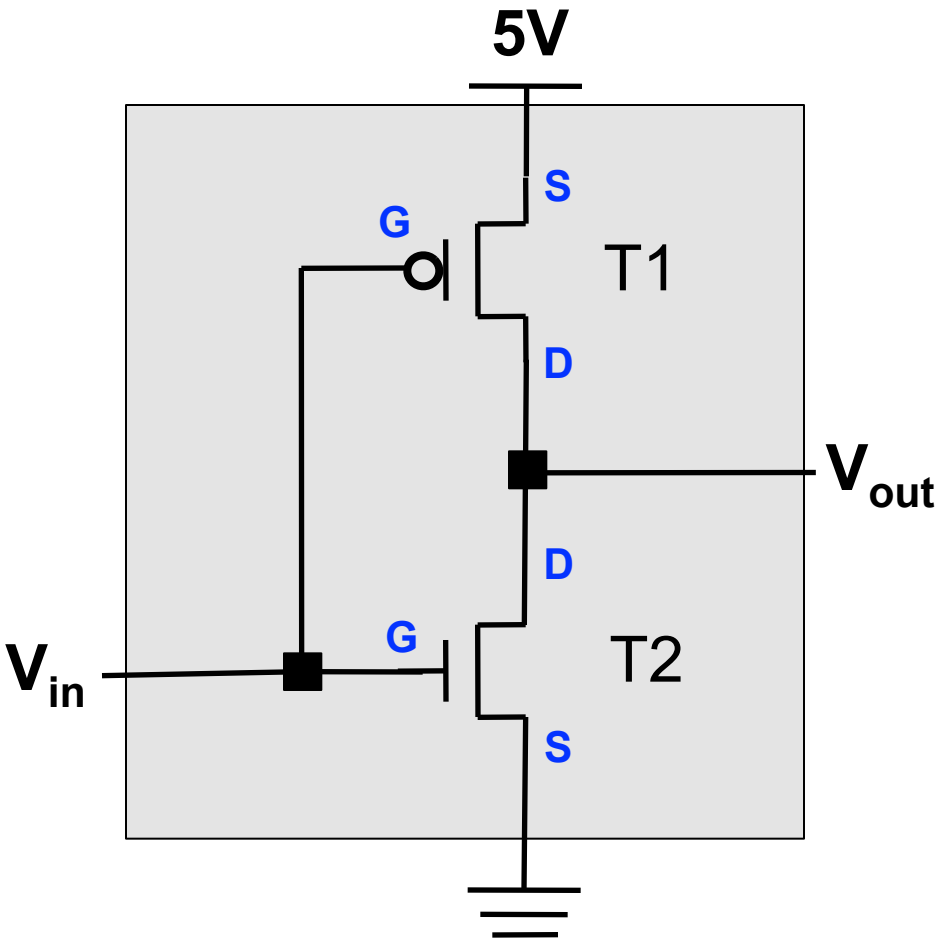


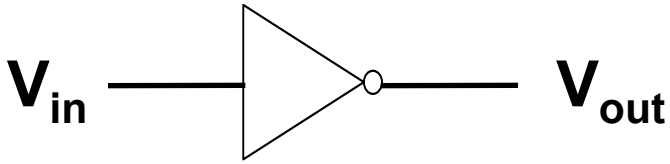
PMOS and NMOS have *complementary* properties

CMOS Logic Gates

- Complementary MOS
- Uses both NMOS and PMOS devices such that there is no direct supply-ground path
  - Dissipates little power when the inputs don't change
- CMOS dominates the digital IC market
- Our focus: Static CMOS gates
  - Other types as well (pseudo-NMOS, domino, ...)

CMOS Inverter

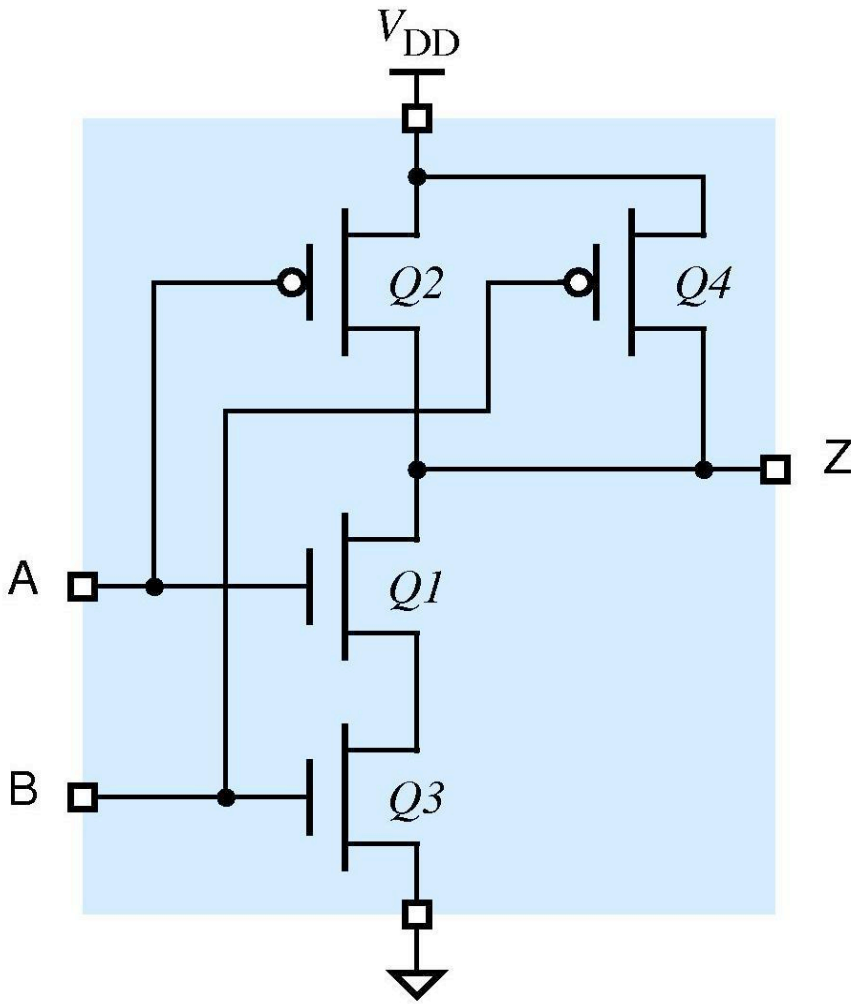




V_{in} is high
T1 is off
T2 is on
V_{out} is low

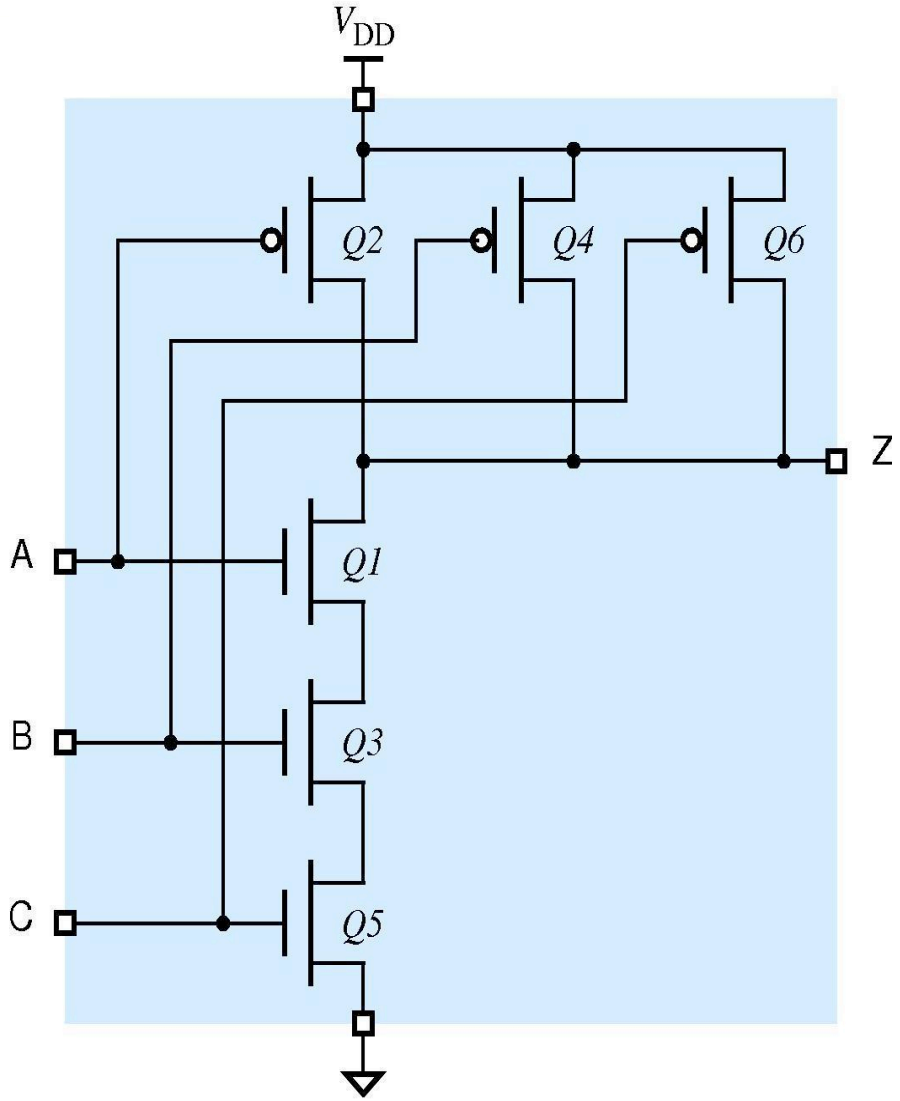
V_{in} is low
T1 is on
T2 is off
V_{out} is high

CMOS NAND Gate

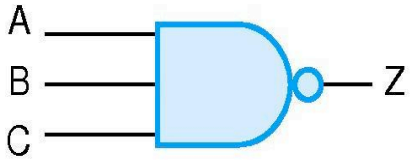


A	B	*Q1*	*Q2*	*Q3*	*Q4*	Z
L	L	off	on	off	on	
L	H	off	on	on	off	
H	L	on	off	off	on	
H	H	on	off	on	off	

3 Input CMOS NAND Gate

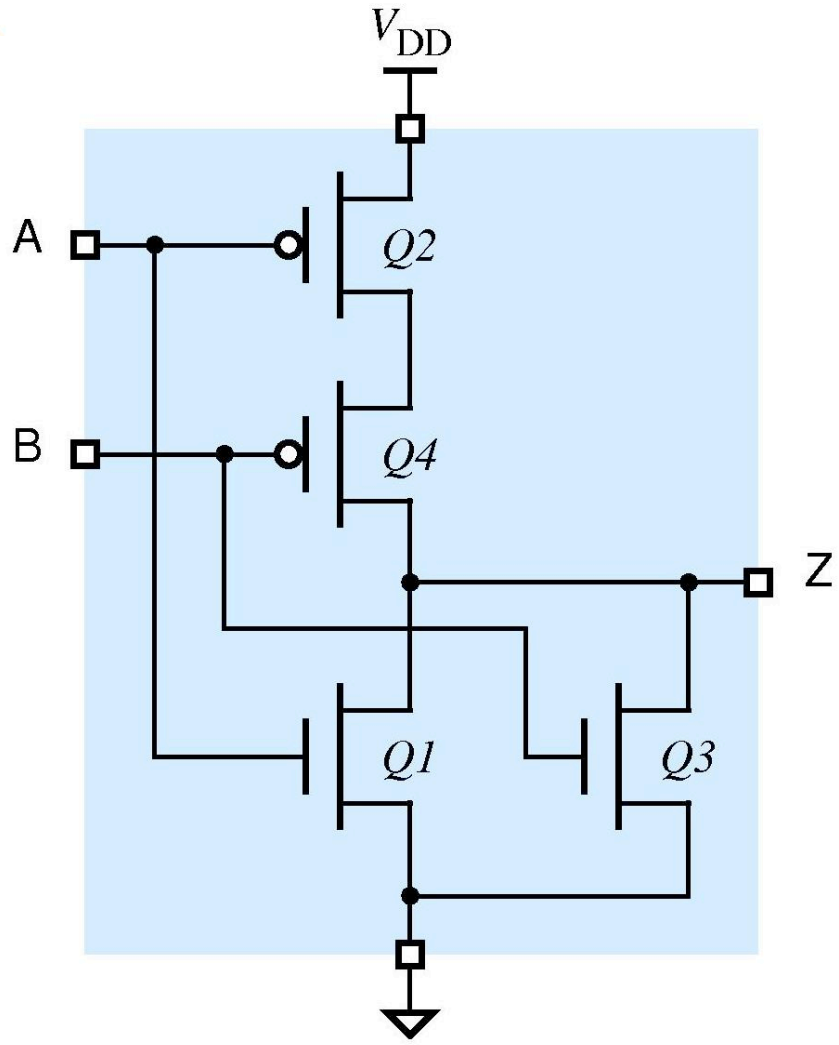


A	B	C	Q1	Q2	Q3	Q4	Q5	Q6	Z
L	L	L	off	on	off	on	off	on	
L	L	H	off	on	off	on	on	off	
L	H	L	off	on	on	off	off	on	
L	H	H	off	on	on	off	on	off	
H	L	L	on	off	off	on	off	on	
H	L	H	on	off	off	on	on	off	
H	H	L	on	off	on	off	off	on	
H	H	H	on	off	on	off	on	off	

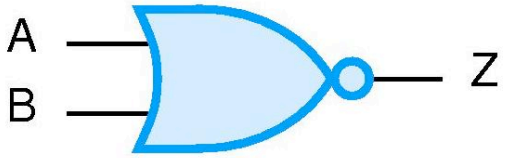


An *n* input gate uses 2*n* transistors

CMOS NOR Gate



A	B	*Q1*	*Q2*	*Q3*	*Q4*	Z
L	L	off	on	off	on	
L	H	off	on	on	off	
H	L	on	off	off	on	
H	H	on	off	on	off	



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

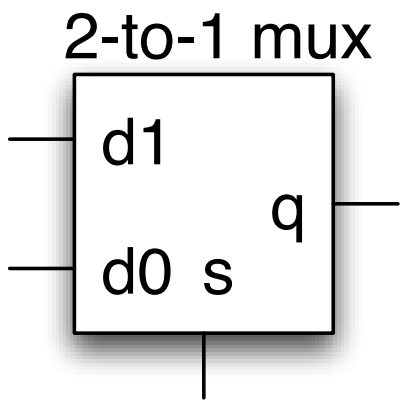
Multiplexer ("mux")

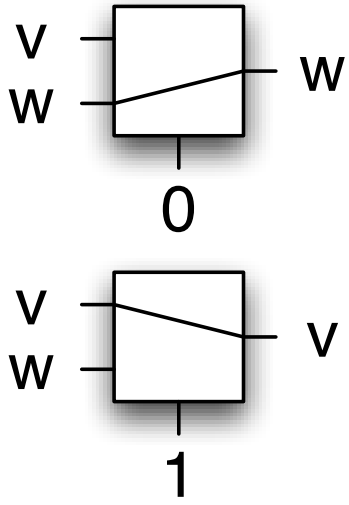
- **Connects one of *n* inputs to the output**
  - ***select* control inputs pick one of the *n* sources**
  - **$\lceil \log_2 n \rceil$ *select* bits**

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

2-to-1 Mux

- **Selects one of two inputs to appear at the output**
- **q = s'•d0 + s•d1**

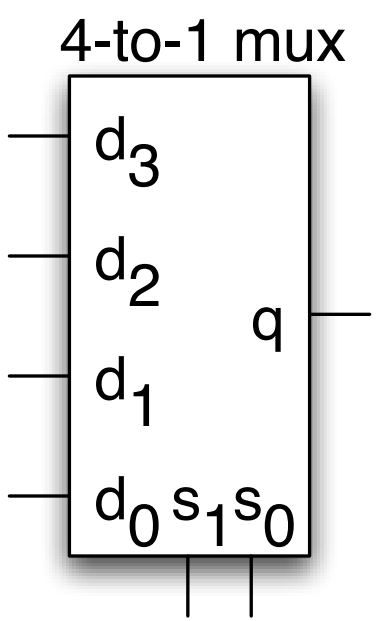


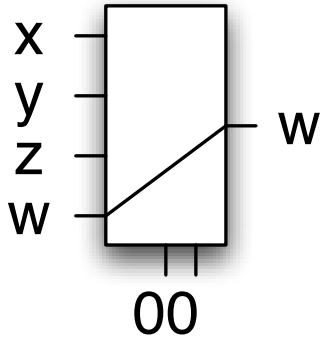


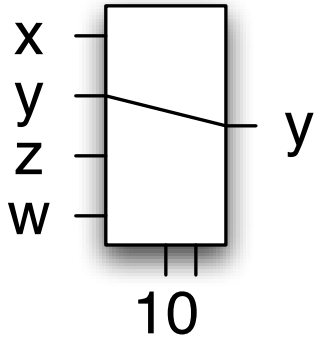
4-to-1 Mux

- Selects one of four inputs to appear at the output

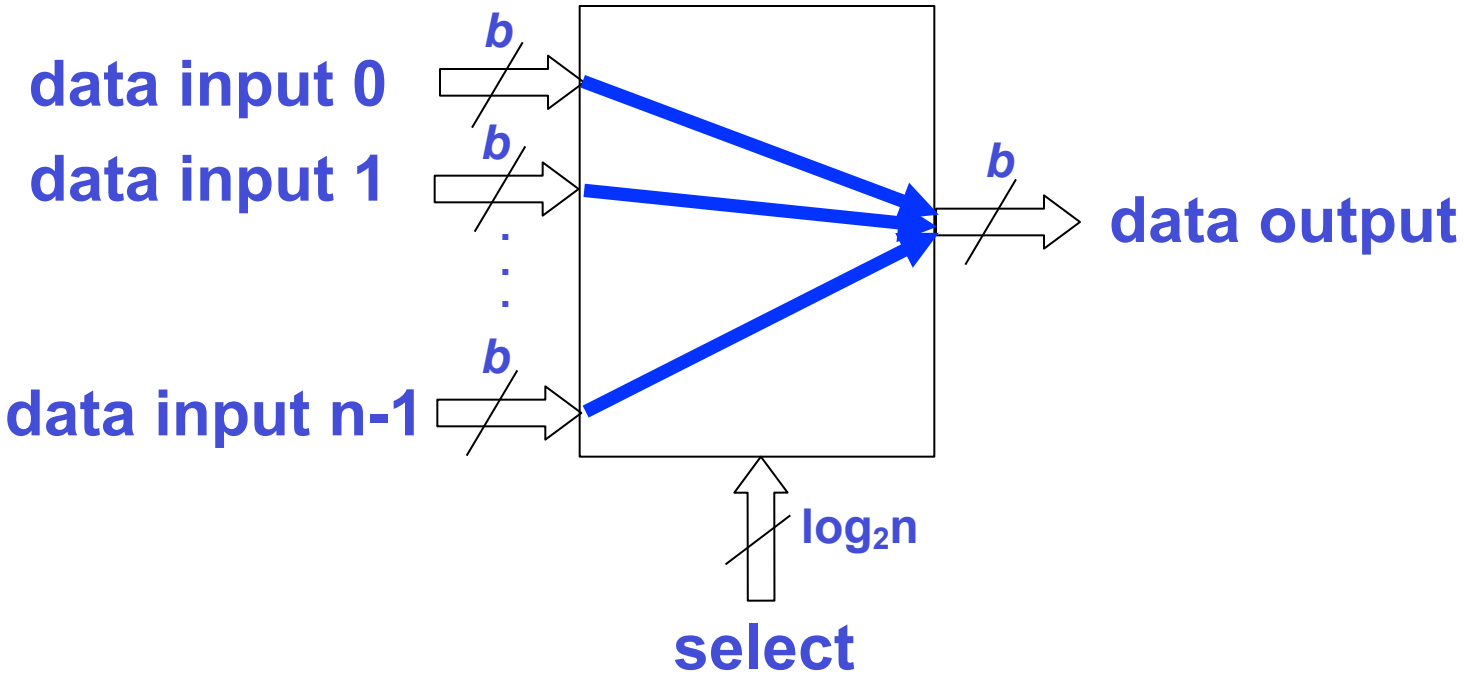
- $q = s_1' \cdot s_0' \cdot d_0 + s_1' \cdot s_0 \cdot d_1 + s_1 \cdot s_0' \cdot d_2 + s_1 \cdot s_0 \cdot d_3$







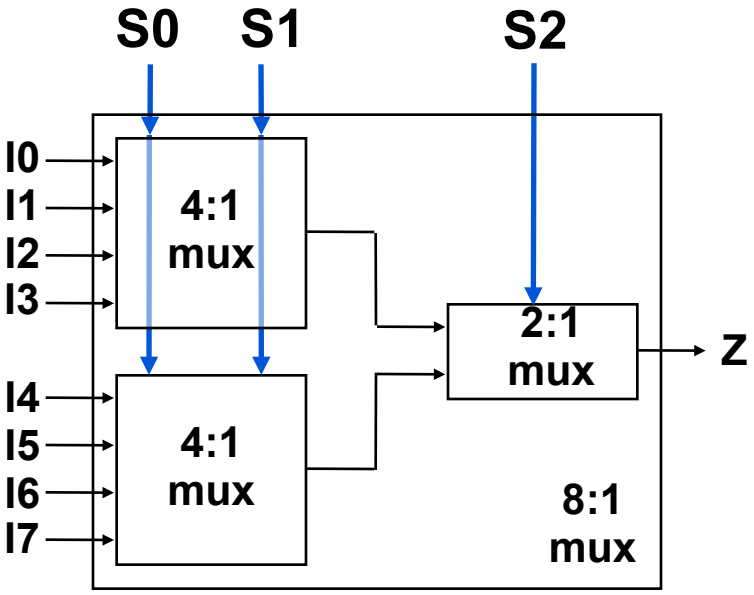
b-bit Mux Organization

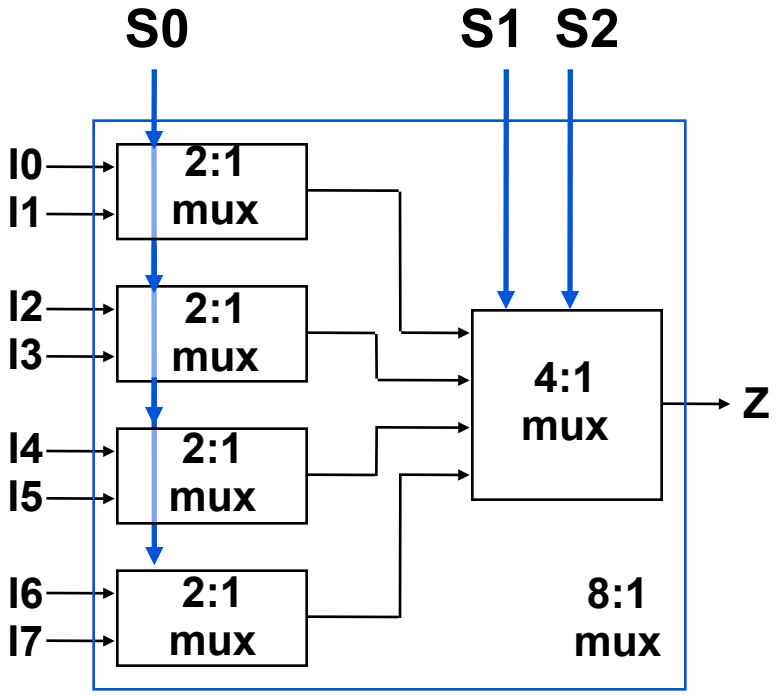


- b bits from the selected data input are connected to the data output
- Internally organized as b n-to-1 muxes
  - Each mux circuit handles 1 bit
  - Same *select* input(s) sent to each mux

Cascading Multiplexers

- **Large multiplexers can be implemented by cascading smaller ones**

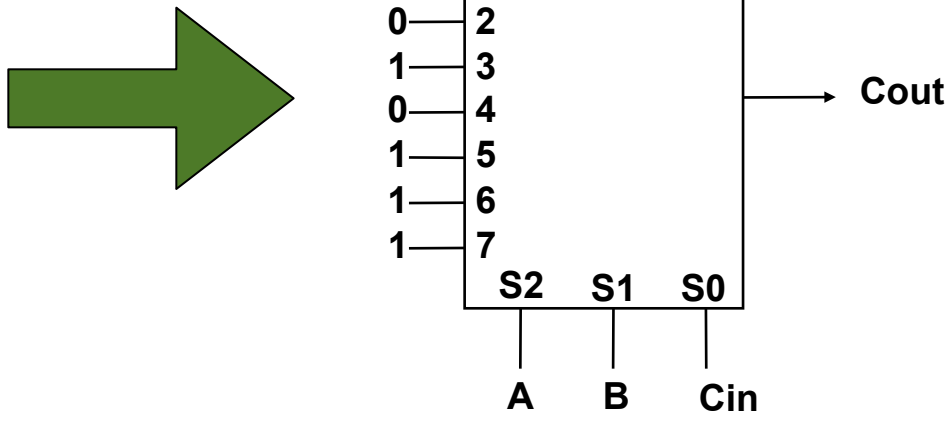




Logic Functions Using Muxes

- **Any function of n variables can be implemented with a 2^n:1 multiplexer**
  - **Input variables connected to *select* inputs**
  - **Data inputs tied to 0 or 1 according to truth table**

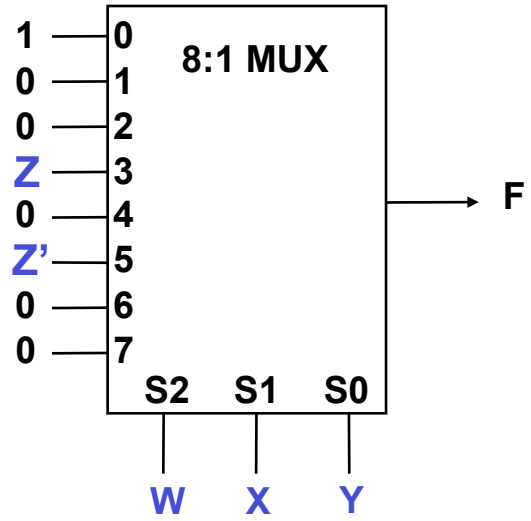
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Getting Away with a Smaller Mux

- Can use 2^{n-1}:1 multiplexer and at most one inverter
  - Connect n-1 input variables to *select* inputs
  - Data inputs tied to 0, 1, n^{th} variable, or inverted n^{th} variable

W	X	Y	Z	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



Next Time

More Combinational Building Blocks