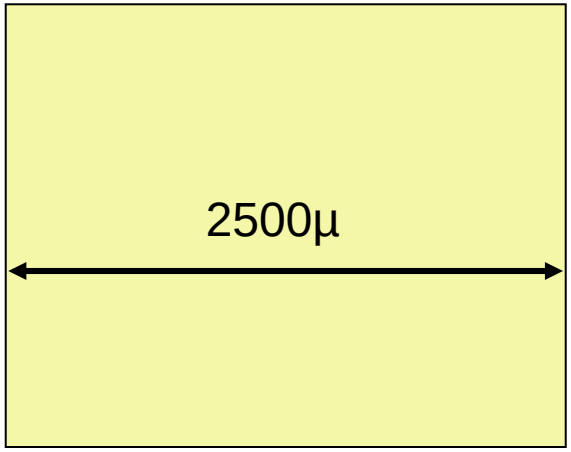
EE 434
Lecture 4

Digital Systems – A preview

Quiz 2

A die is 2500u on a side, comes from an 8” wafer that costs $1200 in a process with a defect density of 1.2/cm^2. Determine the cost per good die as limited by the hard faults in processing.



And the number is ….

1

8

3

5

4

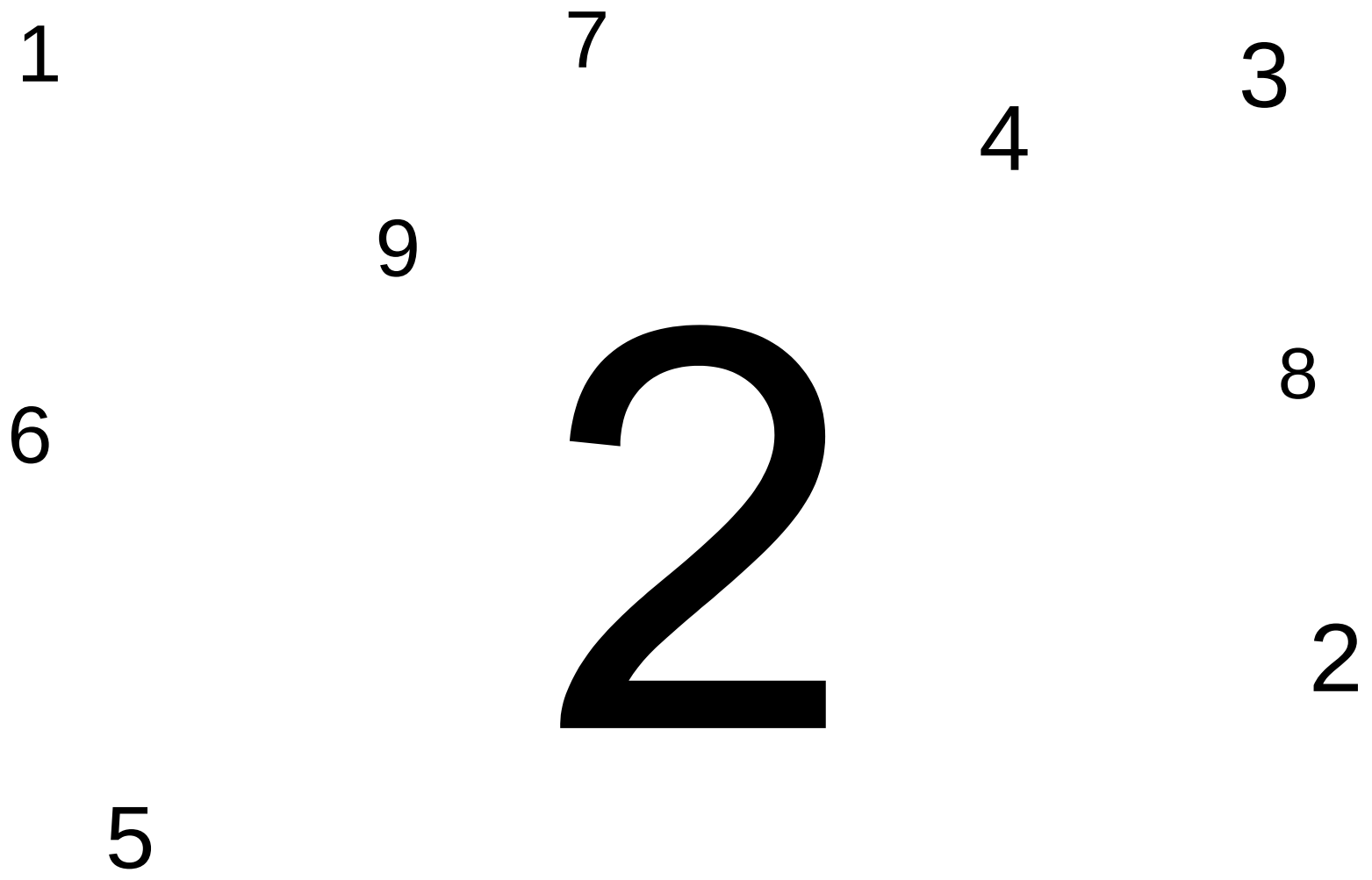
6

9

7

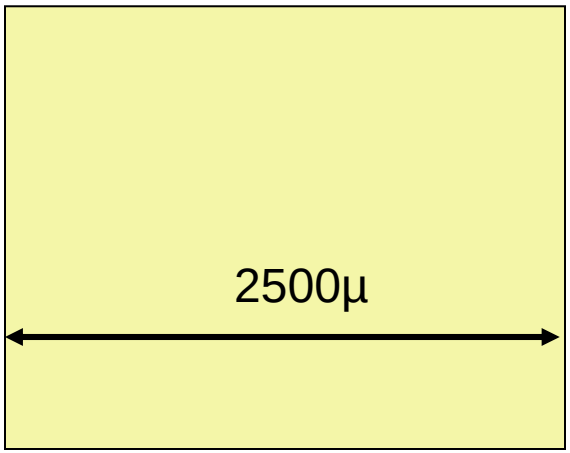
2

And the number is ….



Quiz 2

A die is 2500u on a side, comes from an 8” wafer that costs $1200 in a process with a defect density of 1.2/cm^2. Determine the cost per good die as limited by the hard faults in processing.



Solution:

$$C_{GOOD} = \frac{C_{FAB}}{Y_H}$$

$$C_{FAB} = C_{WAFER}\frac{A_{DIE}}{A_{WAFER}}$$

$$Y_H = e^{-A_{DIE}d}$$

$$C_{GOOD} = C_{WAFER}\frac{A_{DIE}}{A_{WAFER}}e^{A_{DIE}d}$$

Quiz 2

A die is 2500u on a side, comes from an 8” wafer that costs $1200 in a process with a defect density of 1.2/cm^2. Determine the cost per good die as limited by the hard faults in processing.

Solution:

$$C_{GOOD} = C_{WAFER} \frac{A_{DIE}}{A_{WAFER}} e^{A_{DIE} d} = \$1200 \frac{(2500\mu)^2}{\pi 16 in^2} e^{(2500\mu)^2 \frac{1.2}{cm^2}}$$

$$C_{GOOD} = \frac{\$0.231}{.928} = \$0.249$$

Review from Last Time

- Device Geometries Limited by Size of Atoms
- Hard Yield Decreases with Area

$$Y_H = e^{-Ad}$$

  - Major factor that limits die size

- Soft yield generally increases with die area

$$Y = Y_H Y_S$$

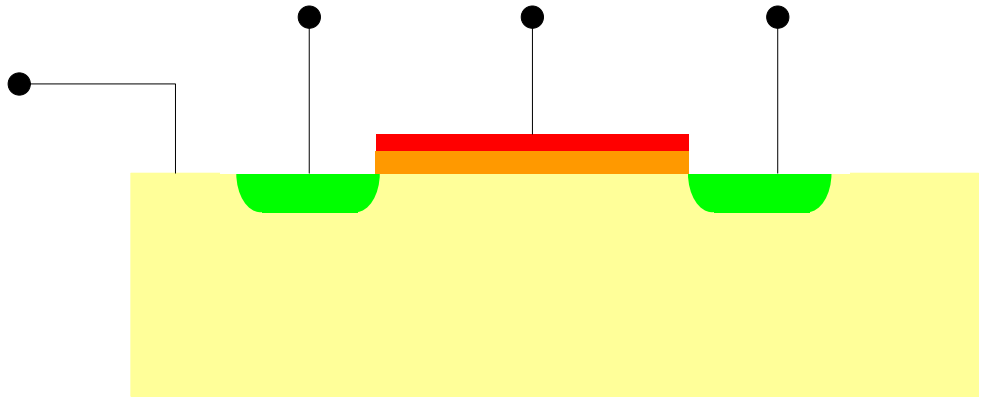
- 6-Sigma Challenge More of a Process (Religion) than Actual Procedure
  - 6-sigma too stringent for many requirements
  - 6-sigmal too lax for many others
  - Cost/good die ultimately of primary interest

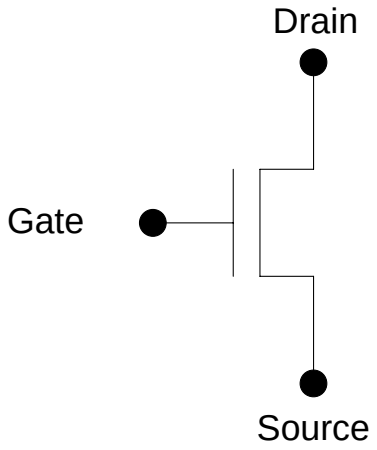
Basic Logic Circuits

- Will present a brief description of logic circuits based upon simple models and qualitative description of processes
- Will discuss process technology needed to develop better models
- Will provide more in-depth discussion of logic circuits based upon better device models

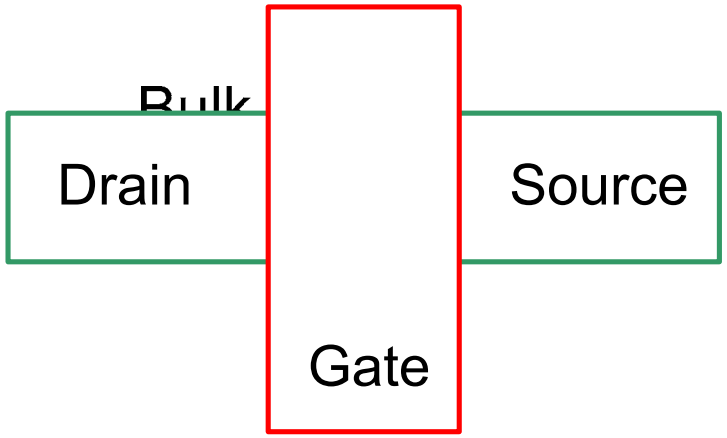
MOS Transistor

Qualitative Discussion of n-channel Operation





Source

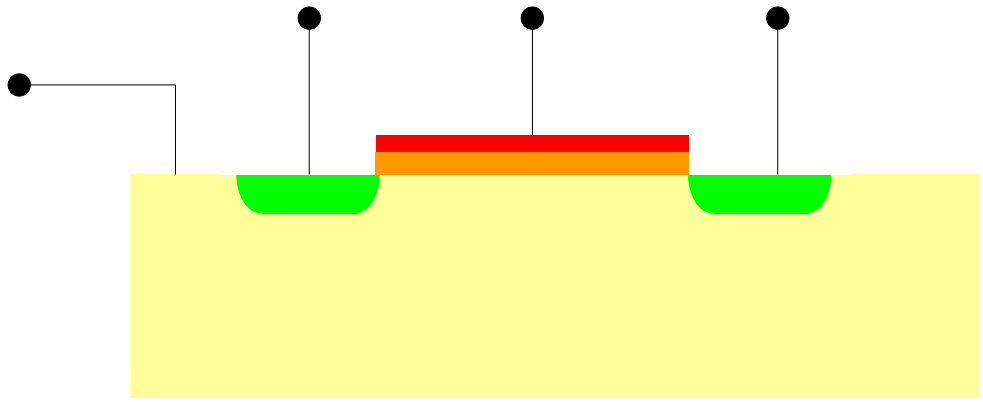


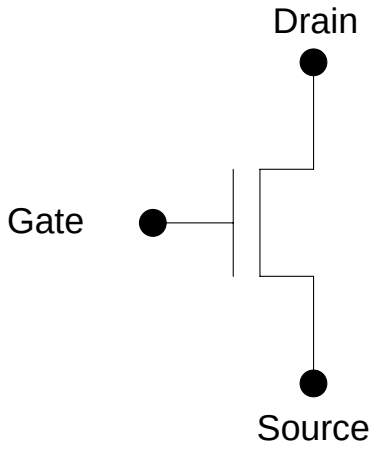
Gate

Drain

MOS Transistor

Qualitative Discussion of n-channel Operation





Gate

Drain

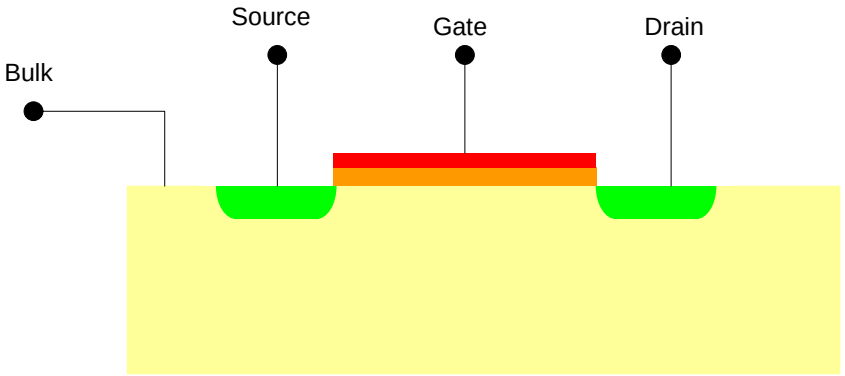
Behavioral Description of Basic Operation

If V_{GS} is large, short circuit exists between drain and source

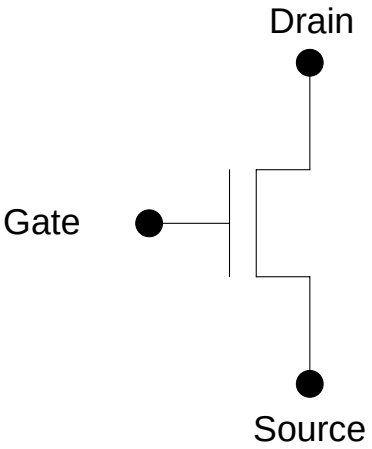
If V_{GS} is small, open circuit exists between drain and source

MOS Transistor

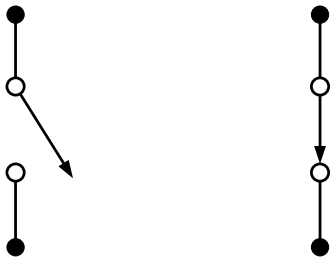
Qualitative Discussion of n-channel Operation



n-channel MOSFET

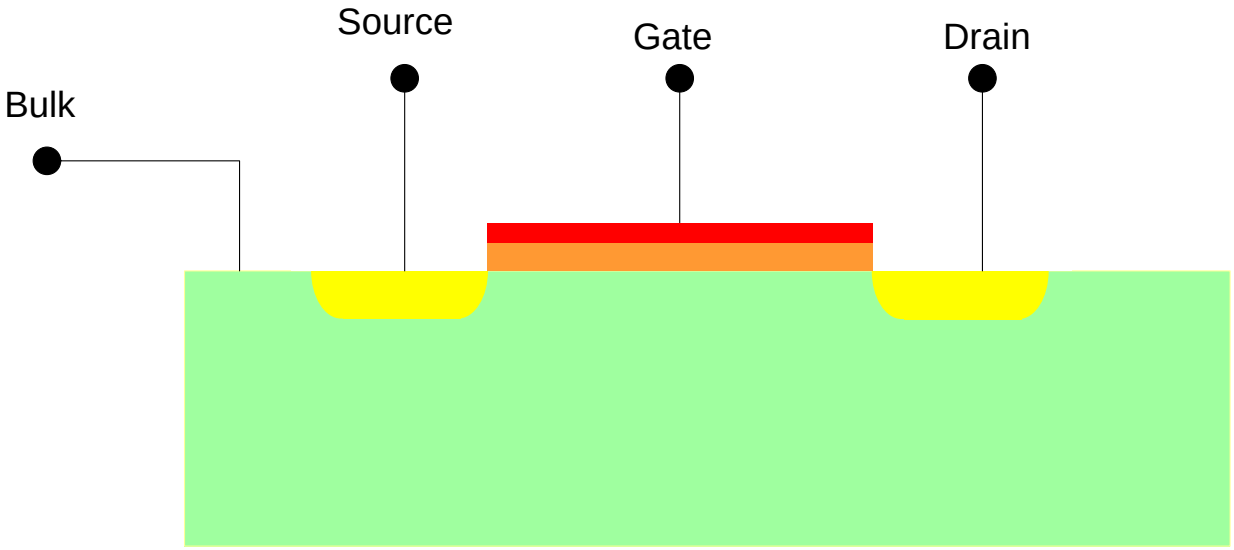


Equivalent Circuit for n-channel MOSFET

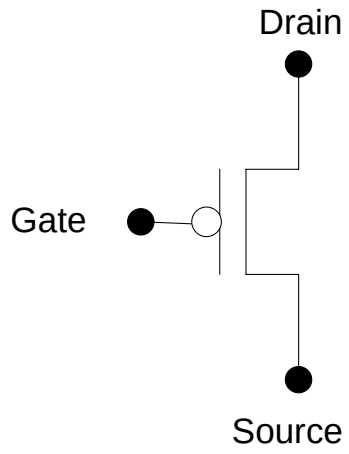


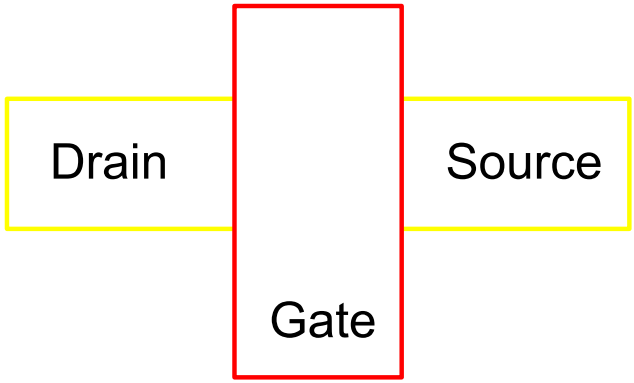
MOS Transistor

Qualitative Discussion of p-channel Operation



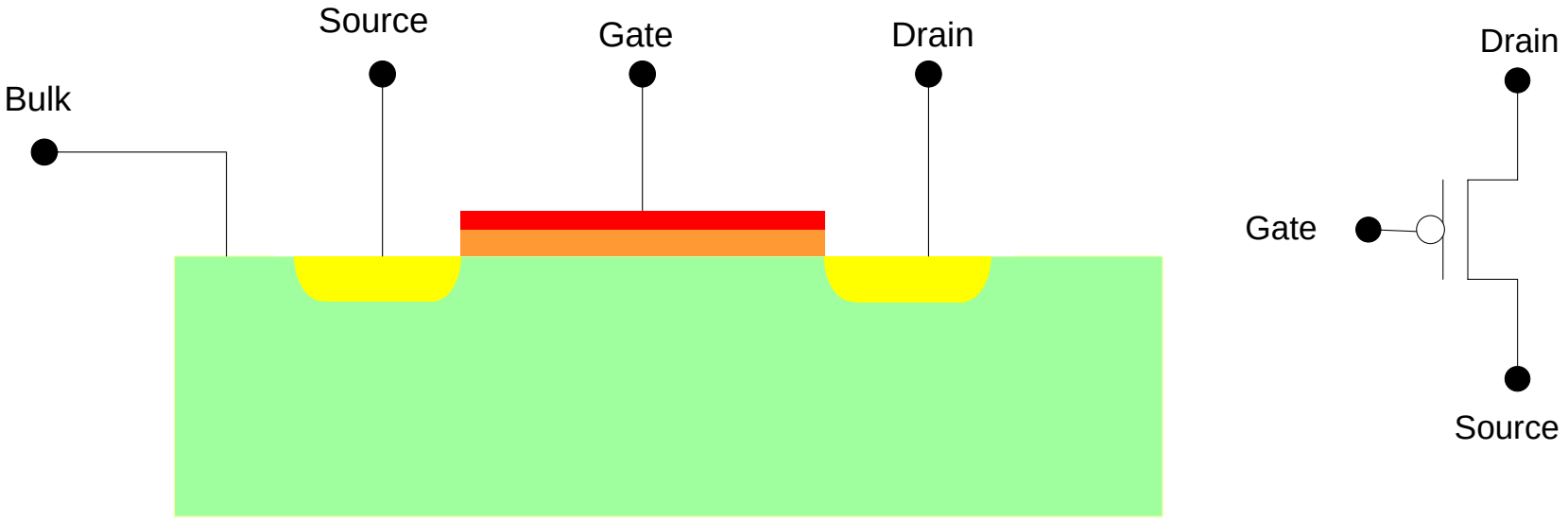
p-channel MOSFET





MOS Transistor

Qualitative Discussion of p-channel Operation



p-channel MOSFET

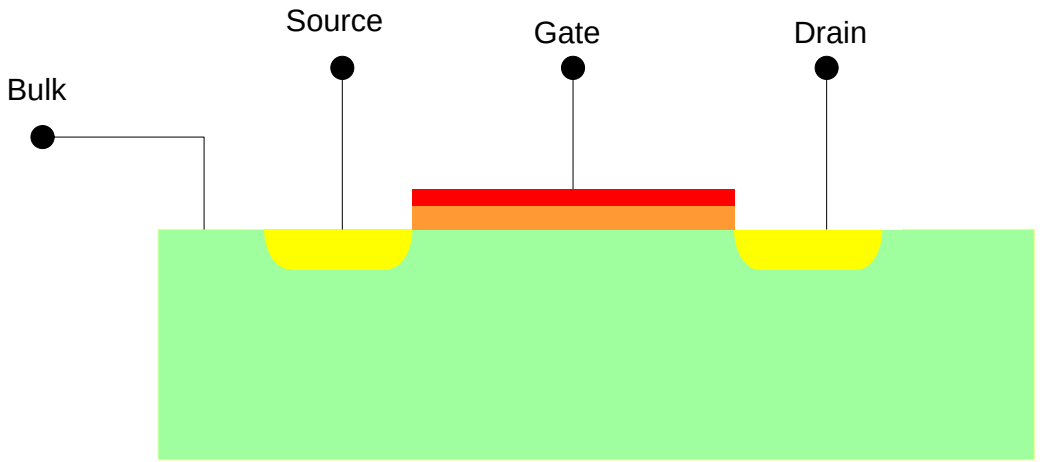
Behavioral Description of Basic Operation

If V_{GS} is small (negative), short circuit exists between drain and source

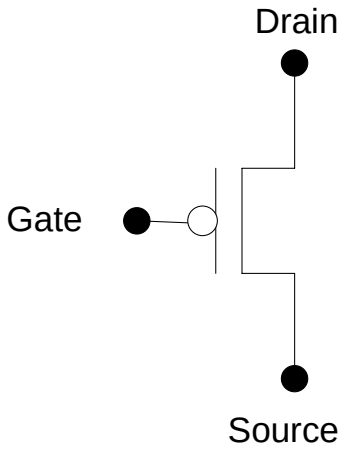
If V_{GS} is large (near 0), open circuit exists between drain and source

MOS Transistor

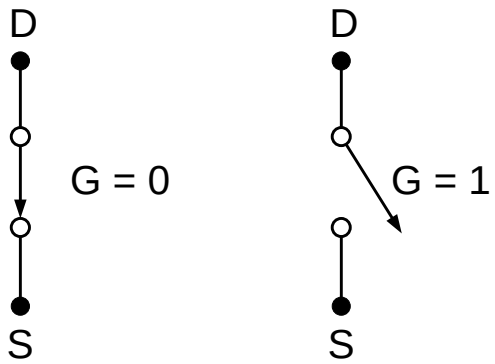
Qualitative Discussion of p-channel Operation



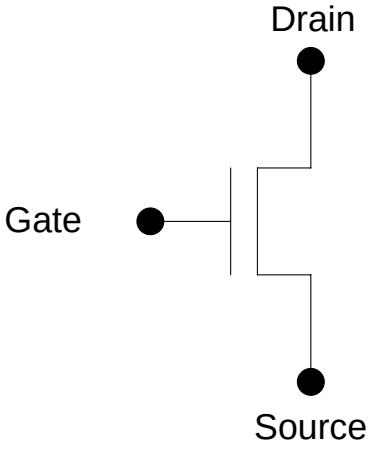
p-channel MOSFET

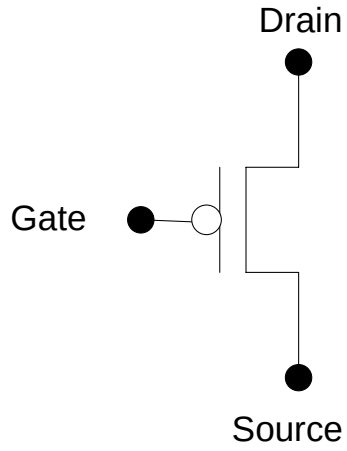


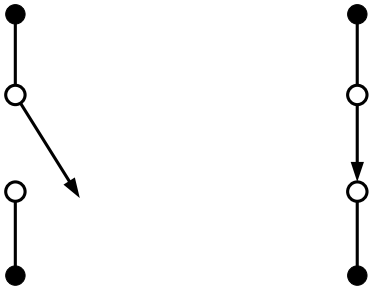
Equivalent Circuit for p-channel MOSFET

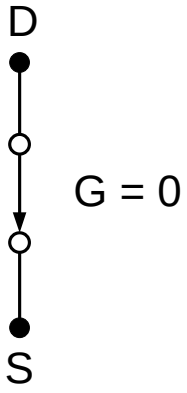


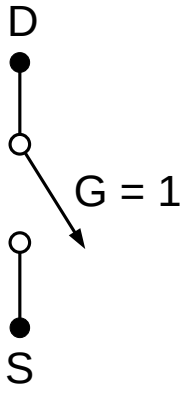
MOS Transistor







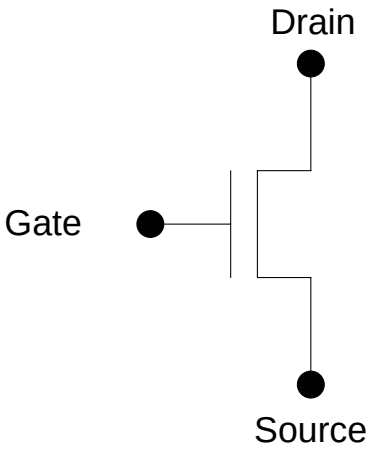


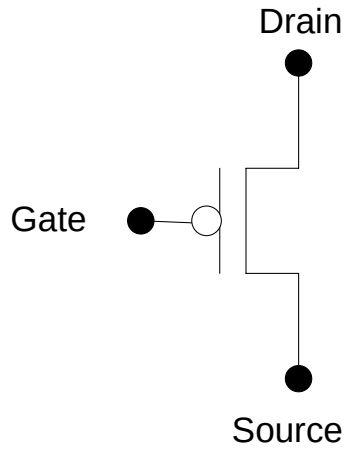


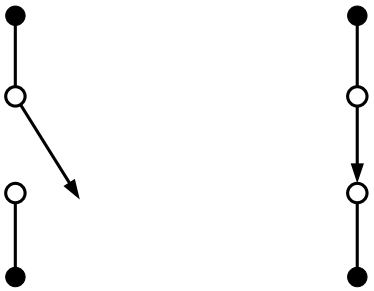
- These represent simple models for the MOS Transistor
- This simple model is adequate for some digital system design work
- More accurate models often needed for other aspects of digital design and for almost all analog design

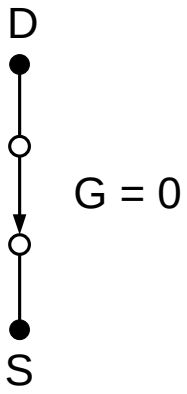
MOS Transistor

Comparison of Operation

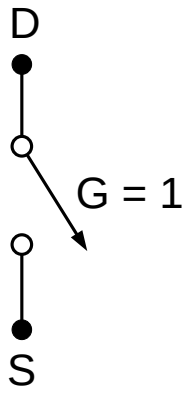






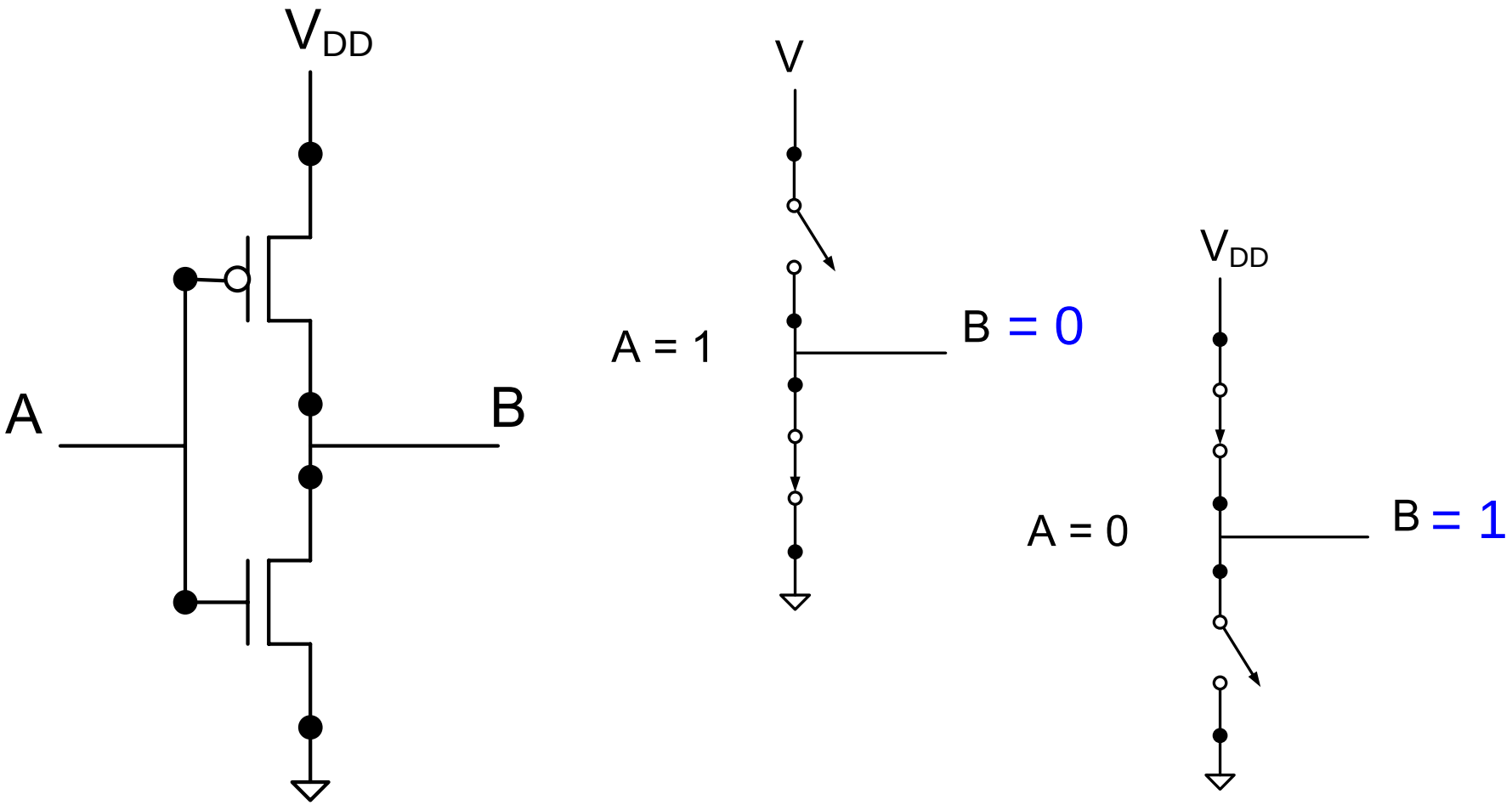


G = 0



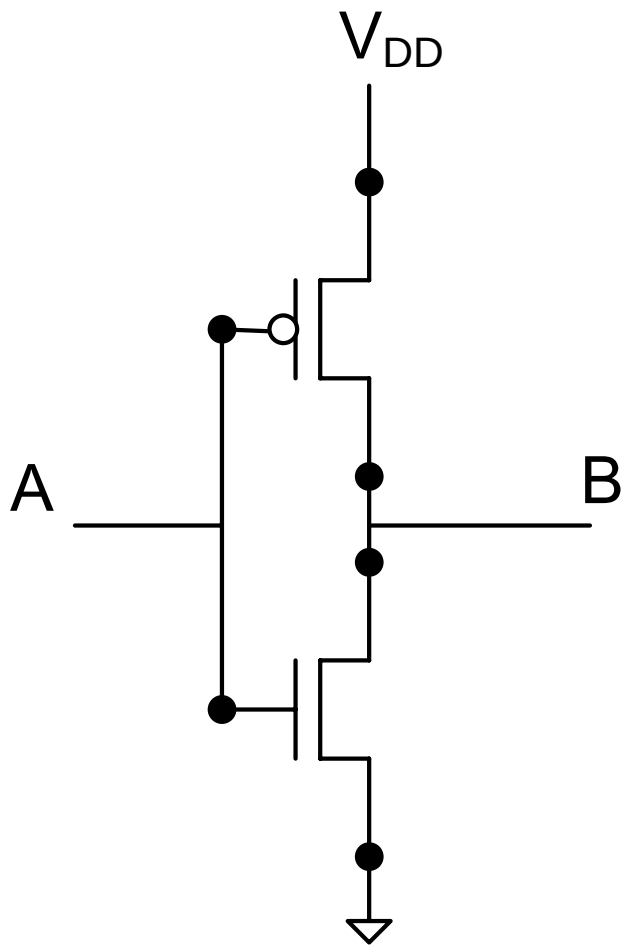
G = 1

Logic Circuits



Circuit Behaves as a Boolean Inverter

Logic Circuits

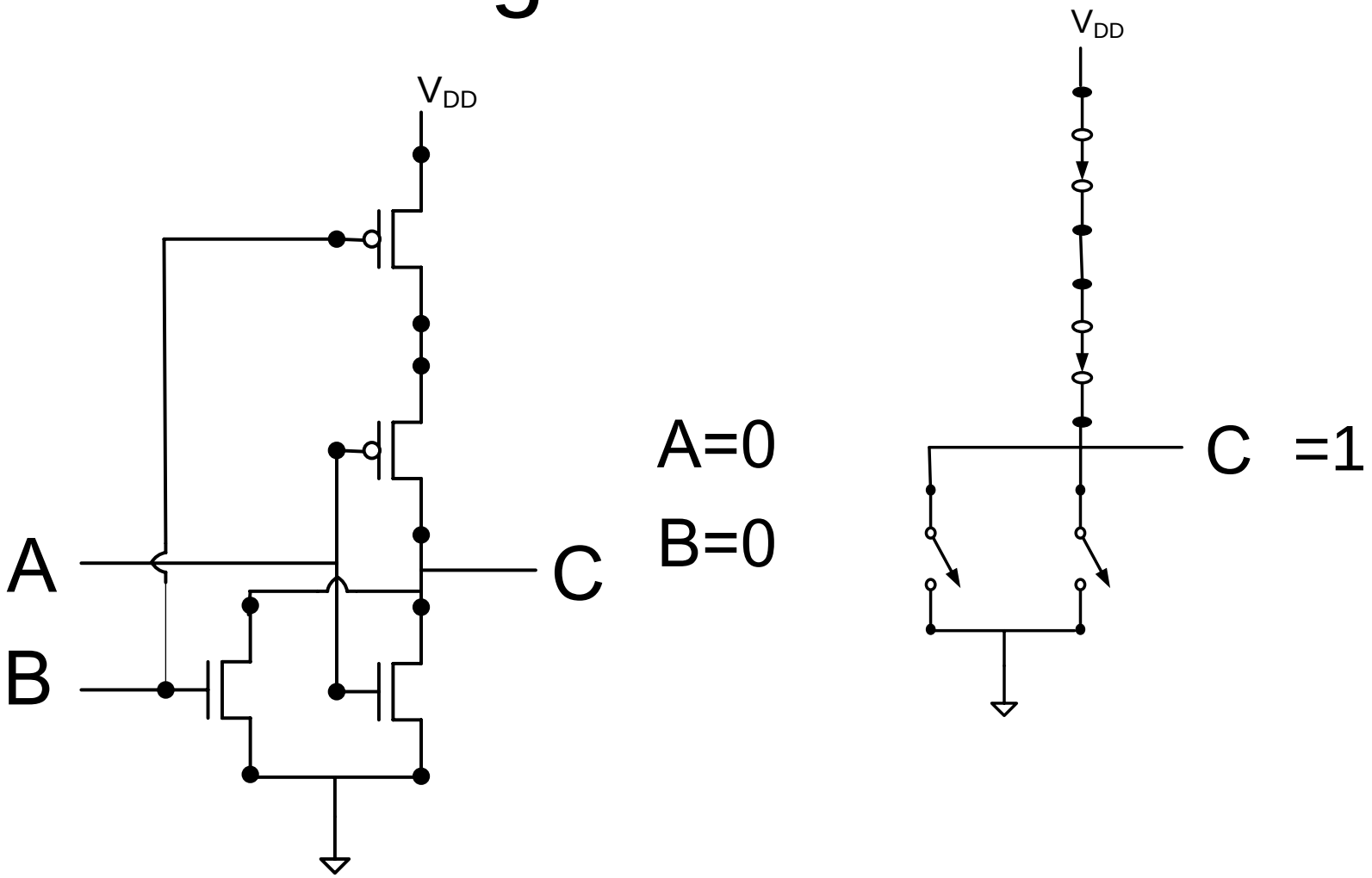


Inverter

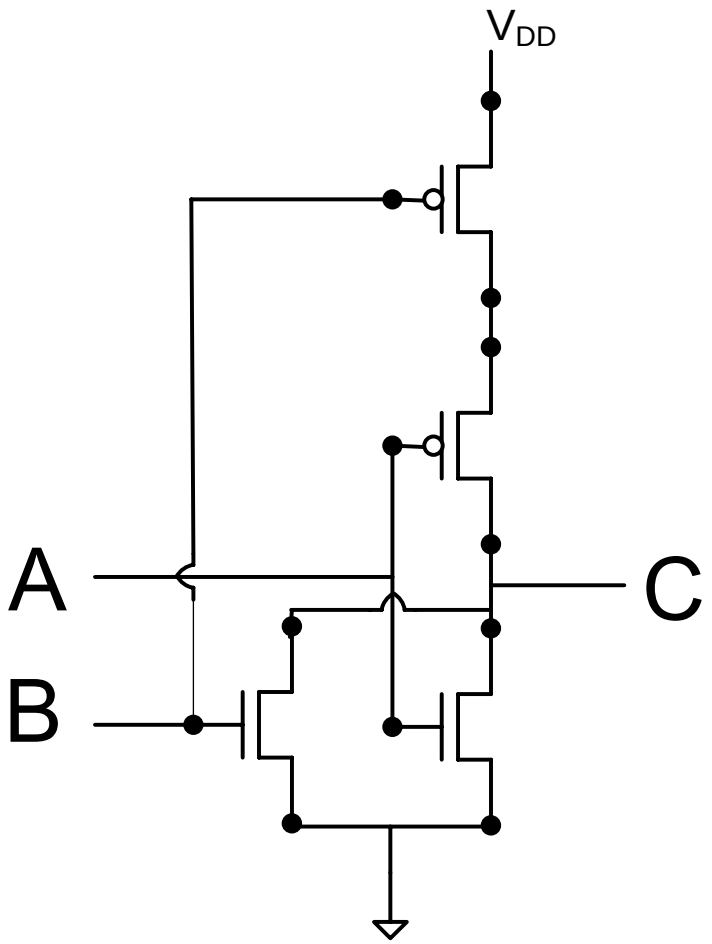
Truth Table

A	B
0	1
1	0

Logic Circuits

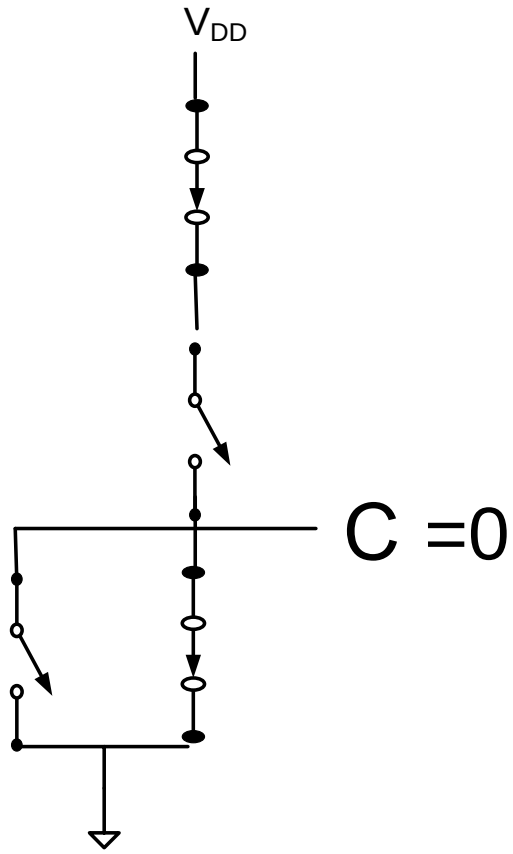


Logic Circuits

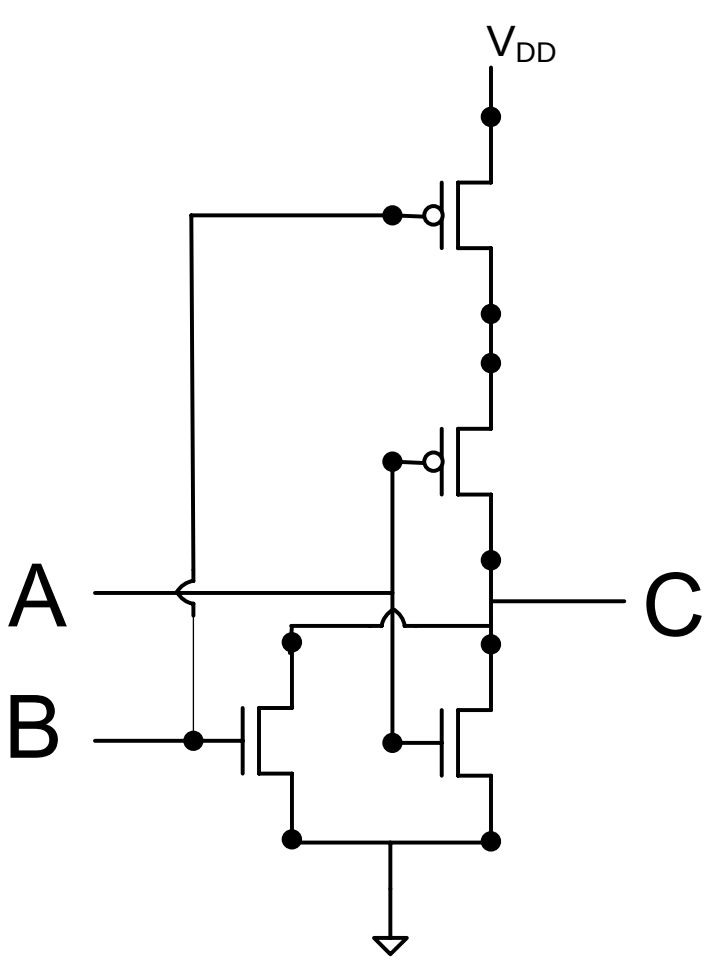


A=1

B=0

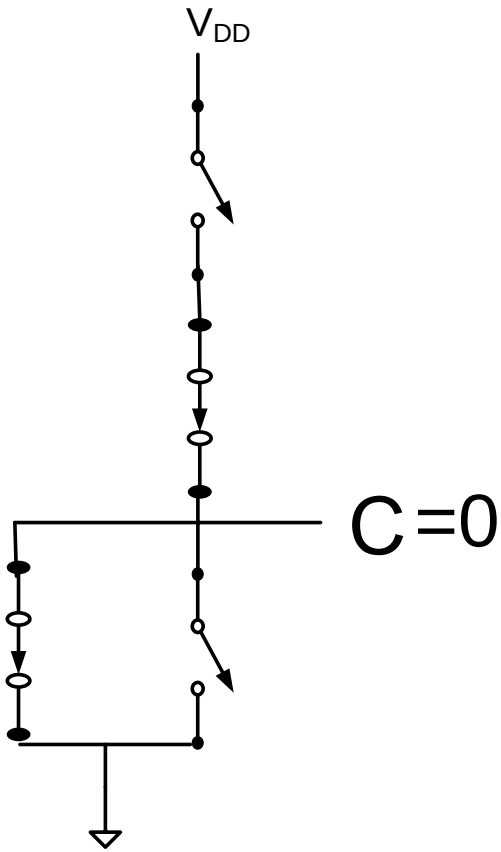


Logic Circuits

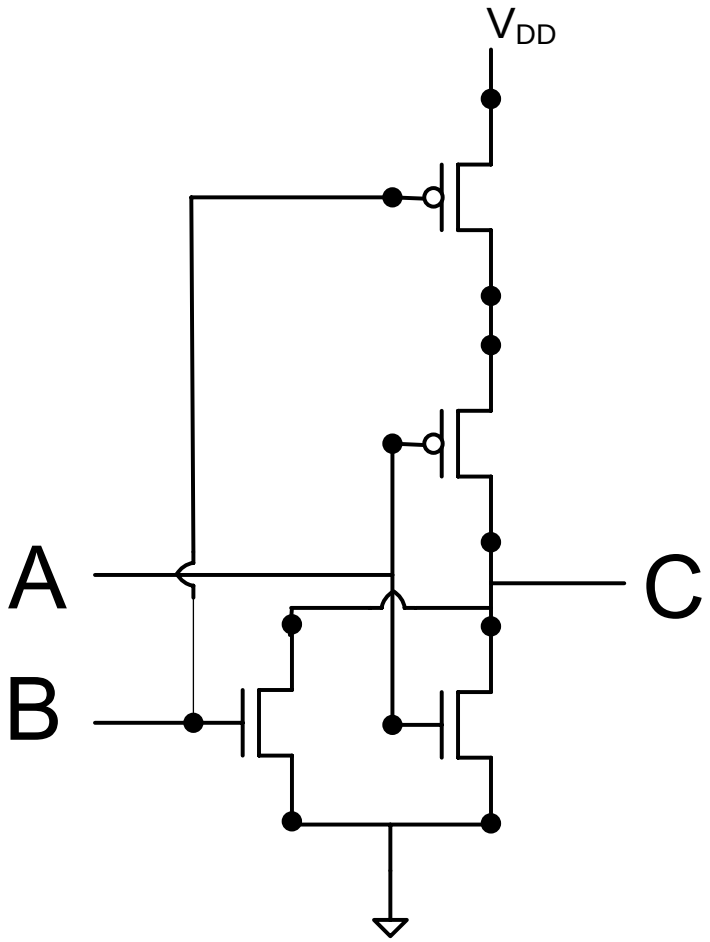


A=0

B=1

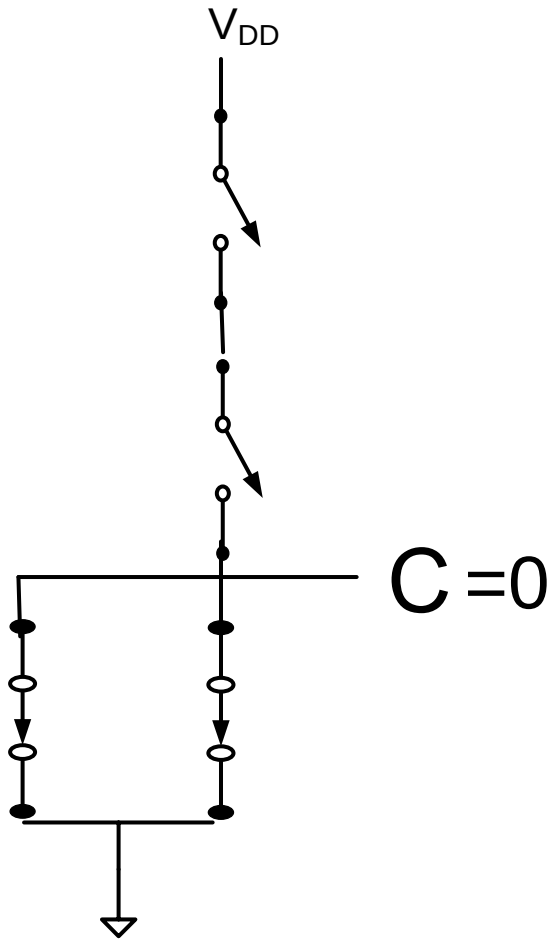


Logic Circuits

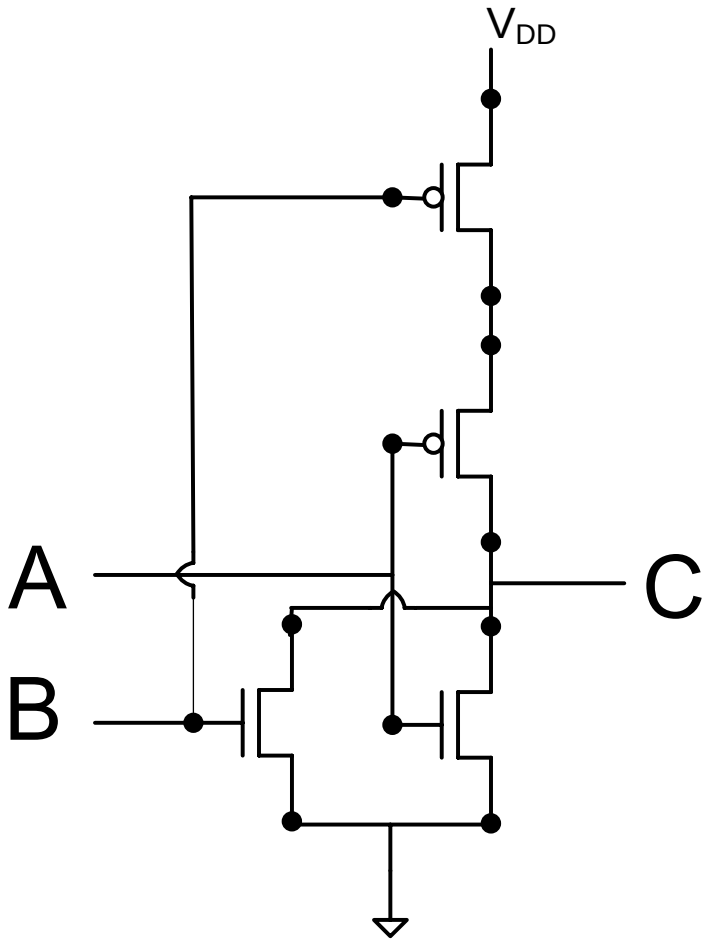


A=1

B=1



Logic Circuits

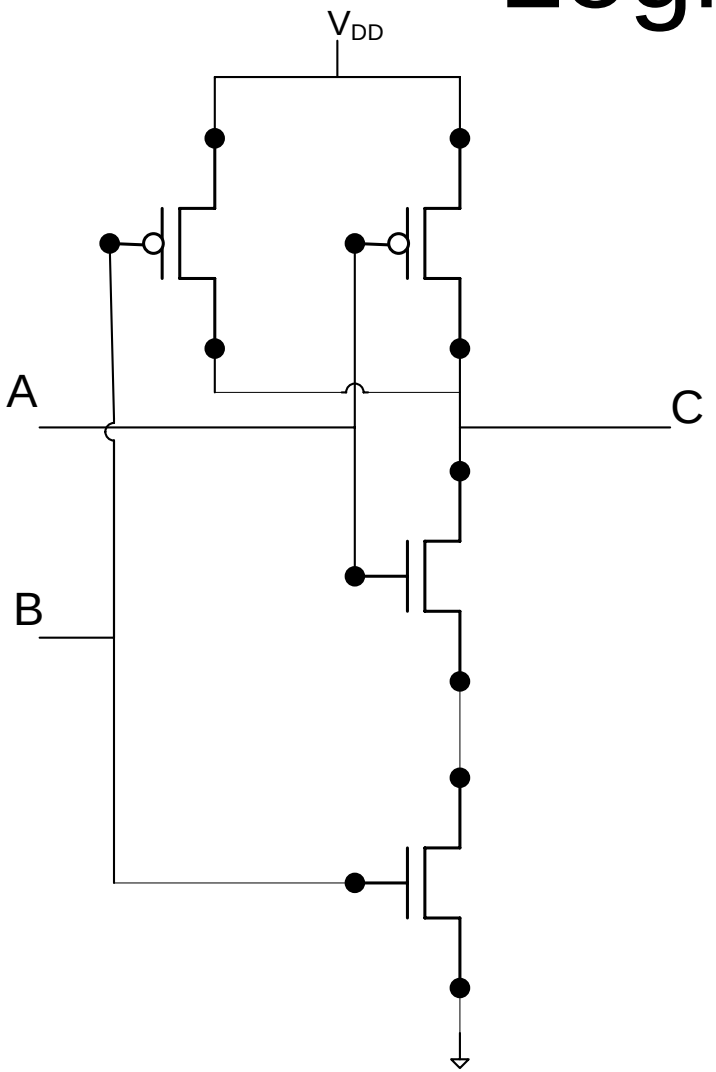


NOR Gate

Truth Table

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

Logic Circuits

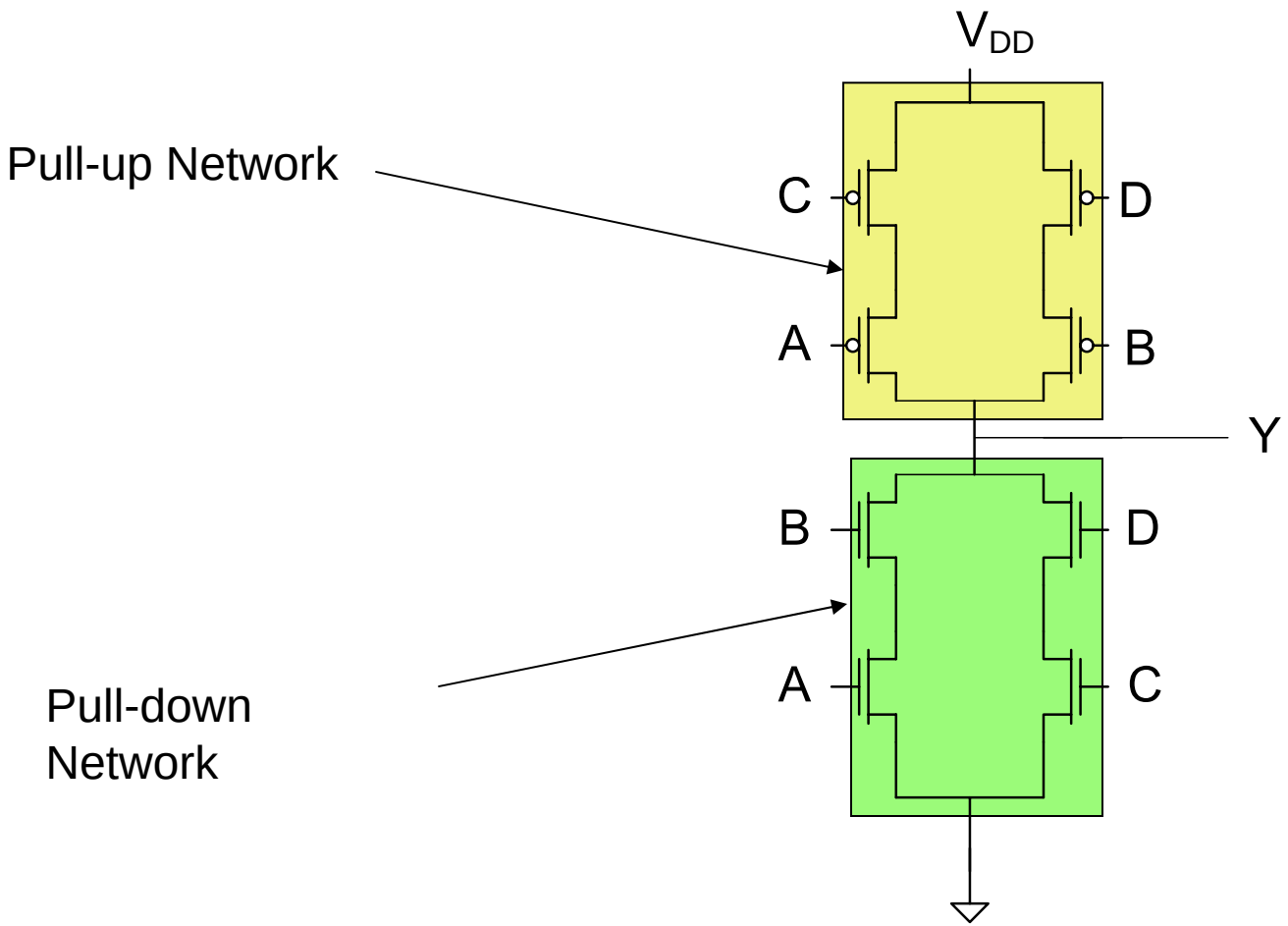


NAND Gate

Truth Table

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Complex Gates

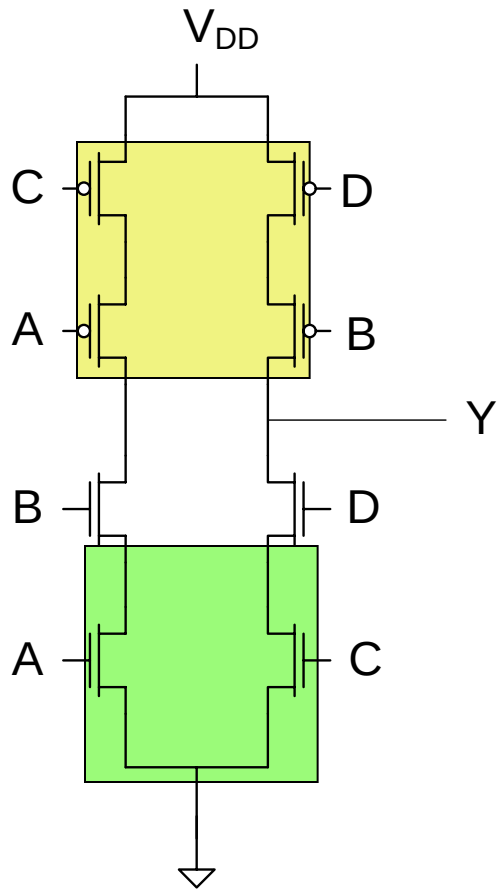


$$\mathbf{Y} = \overline{(\mathbf{A} \bullet \mathbf{B}) + (\mathbf{C} \bullet \mathbf{D})}$$

Complex Gates

Pull up and pull down network never both conducting

One of the two networks is always conducting

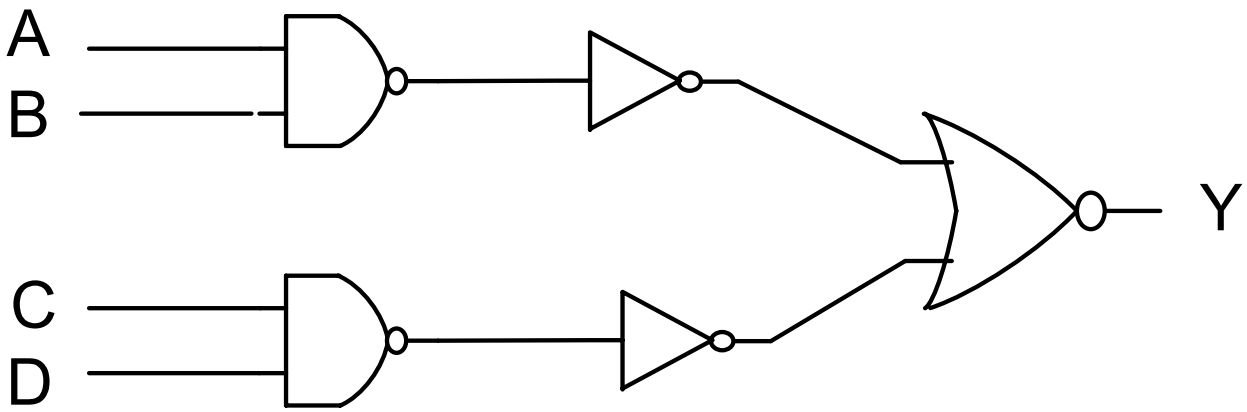


$$\mathbf{Y} = \overline{(\mathbf{A} \bullet \mathbf{B}) + (\mathbf{C} \bullet \mathbf{D})}$$

Consider

$$Y = \overline{(A \bullet B) + (C \bullet D)}$$

Alternate Implementation

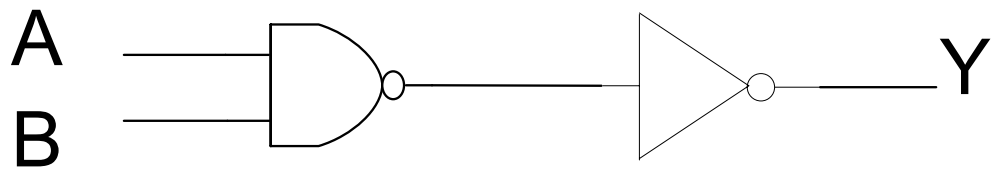


3 levels of Logic

16 Transistors if Basic CMOS Gates are Used

Consider $Y = A \bullet B$

Standard CMOS Implementation

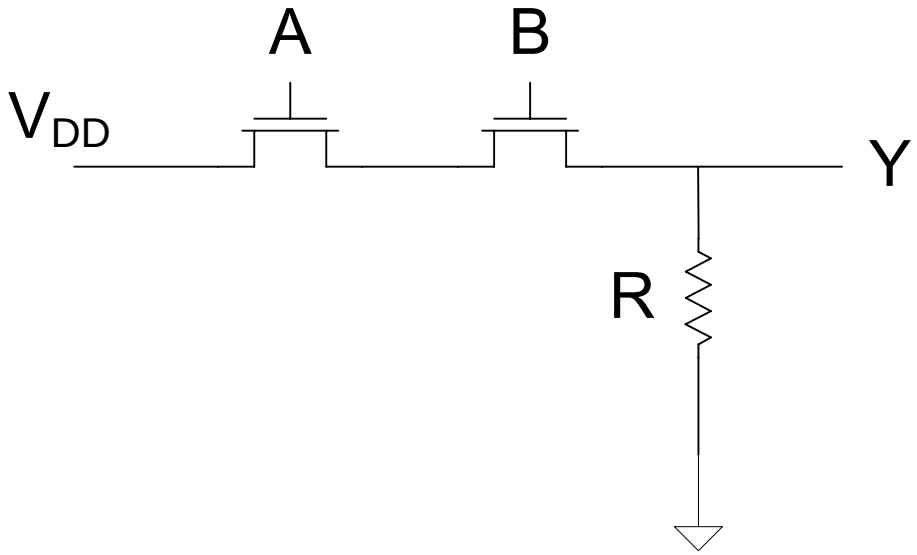


2 levels of Logic

6 Transistors if Basic CMOS Gates are Used

Basic noninverting functions generally require more complexity if basic CMOS gates are used for implementation

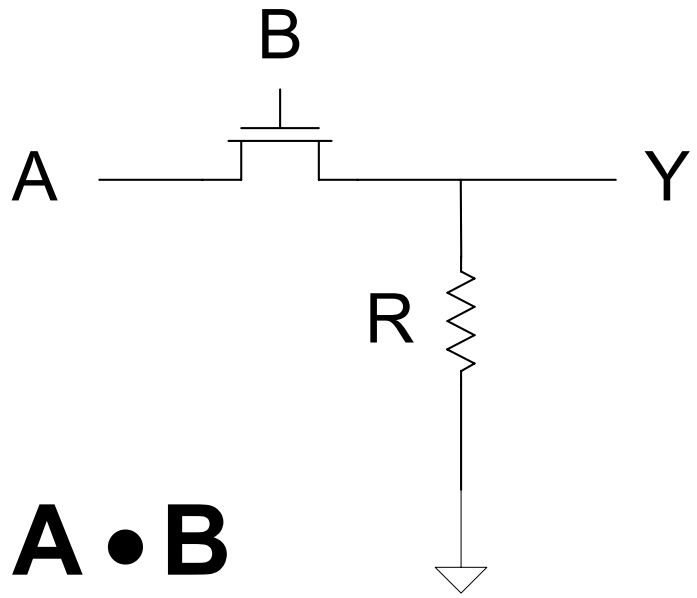
Pass Transistor Logic



$$Y = A \bullet B$$

Requires only 2 transistors rather than 6 for a standard CMOS gate (and a resistor).

Pass Transistor Logic

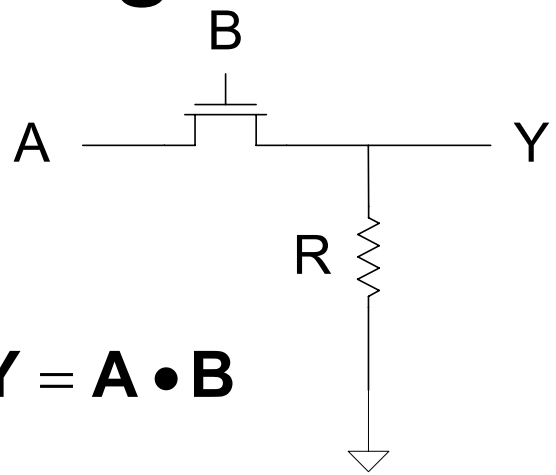


$$Y = A \bullet B$$

Even simpler pass transistor logic implementations are possible

Requires only 1 transistor (and a resistor).

Pass Transistor Logic



Requires only 1 transistor (and a resistor)

- **Resistor may require more area than several hundred or even several thousand transistors**
- **Signal levels may not go to V_{DD} or to 0V**
- **Static power dissipation may not be zero**
- **Signals may degrade unacceptably if multiple gates are cascaded**
- **"resistor" often implemented with a transistor to reduce area but signal swing and power dissipation problems still persist**
- **Pass transistor logic is widely used**

Logic Design Styles

- Several different logic design styles are often used throughout a given design
- The designer has complete control over what is placed on silicon and governed only by cost and performance
- New logic design strategies have been proposed recently and others will likely emerge in the future
- The digital designer needs to be familiar with the benefits and limitations of varying logic styles to come up with a good solution for given system requirements

End of Lecture 4