Abstraction & Digital Logic: From Transistors to Gates

Lab 1 – DUE TUESDAY 1/19 11:59pm

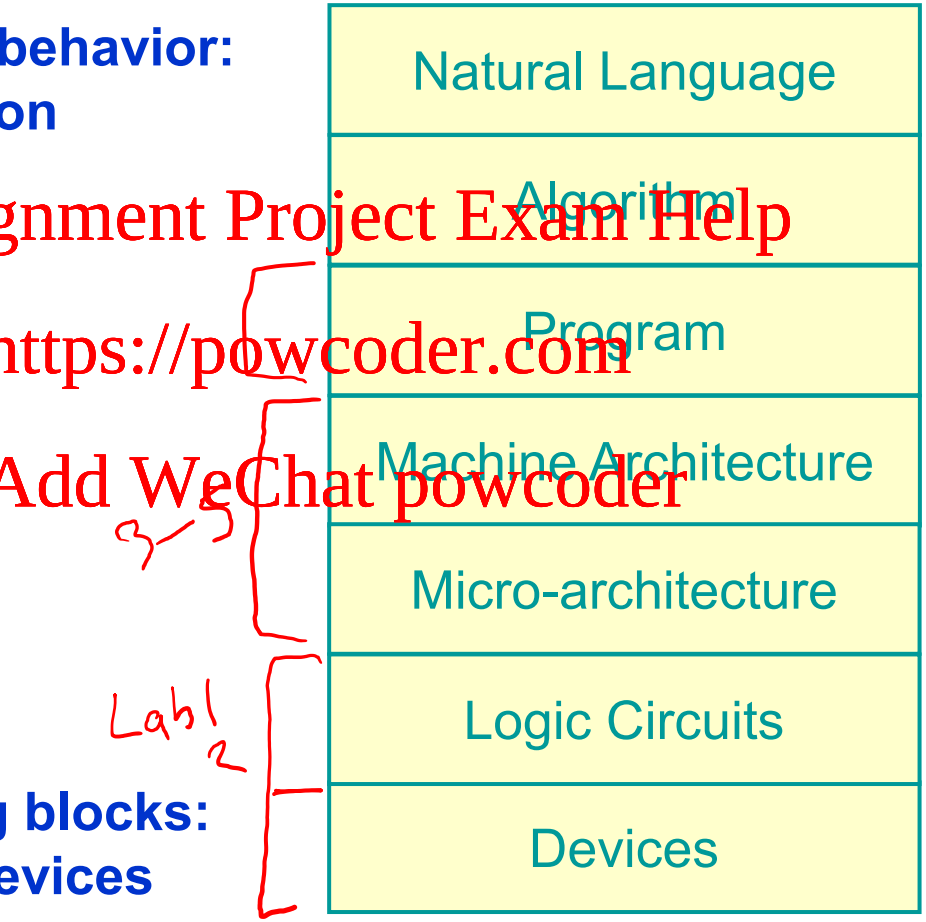
- Lab 1 due in 1.5 weeks, released tomorrow
  1. Must commit/push to git server
  2. **Submit Commit ID in Google form**
- Uses MML (MultiMedia Logic)
  - We will talk about transistors to build gates today
  - Lab will use gates to build logic functions
  - You should be reading Section 3 of the ZyBook (or equivalent of other books) as on schedule

Start going through Zybooks textbook

- We will first cover a little bit of Chapter 3
- Today's lecture will mirror 3.1,3.2
- Start doing the participation activities!

Problem Transformation - levels of abstraction

The desired behavior: the application

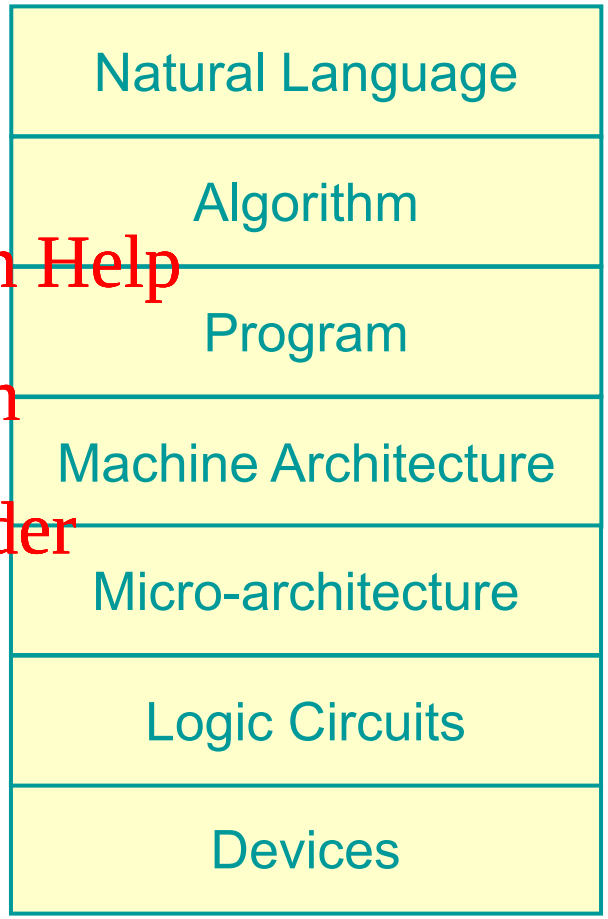


The building blocks: electronic devices

Physics Materials

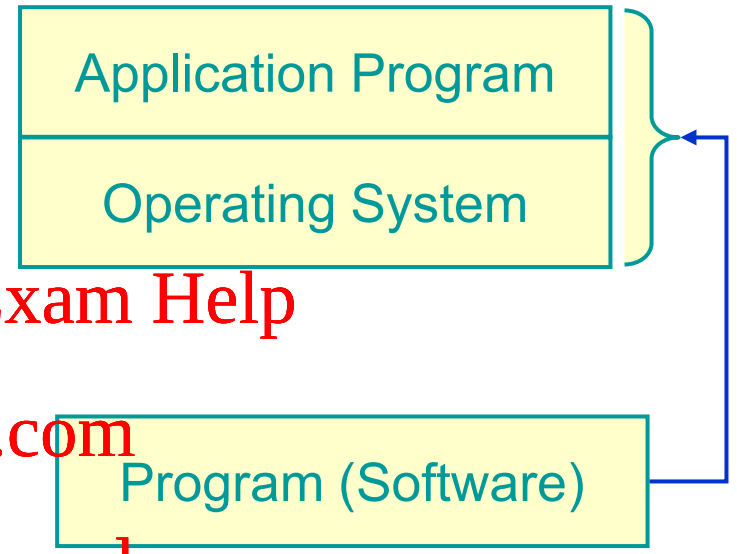
Levels of Abstraction

- These levels do not necessarily correspond to discrete components, but to well defined standard interfaces.
- Standard interfaces provide
  - portability
  - third party software/hardware
  - wider usage
- These levels are to some extent arbitrary - there are other ways to draw the lines.



The Program Level

- Most computers run a management program called the *operating system (OS).*
- Application programs interface to the machine architecture via the OS.



This lecture	Data
PowerPoint	Application Program
Windows 10	Operating System

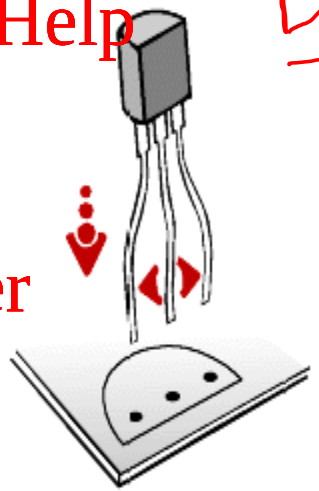
Course Outline - What is Next?

- The building blocks of computers: logic gates
- How to represent information
- The basic algorithm: the von Neumann model
- MIPS structure
- Programming the machine: assembly language
- A higher-level language: (Some) C

The Transistor: building block of computers

- Microprocessors contain many transistors
  - 15-Core Xeon Ivy Bridge-EX (2014): 4310 million
  - 10-Core Xeon Westmere-EX (2011): 2600 million
  - Six-Core Core i7 (2010): 1170 million
  - AMD 6-core Opteron (2009): 904 million
  - Intel Core i7 Quad (2008): 731 million
  - Intel Core 2 Duo (2006): 291 million
  - IBM/Apple PowerPC G5 (2003): 58 million
  - Intel Pentium 4 (2000): 42 million
  - IBM PowerPC 750FX (2002): 38 million
  - Intel 4004 (1971): 2300

The Transistor: Past and Present



Buy

Integrated Circuit

What Is a Transistor?

- A switch, which can close between the source and the drain
- Changing the voltage of the gate lets you change the current flow between the source and drain (closing or opening the switch)

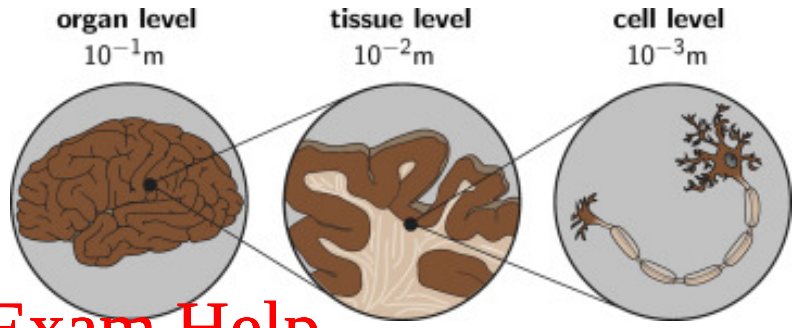
Neuron- Transistor analogy

100 billion neurons

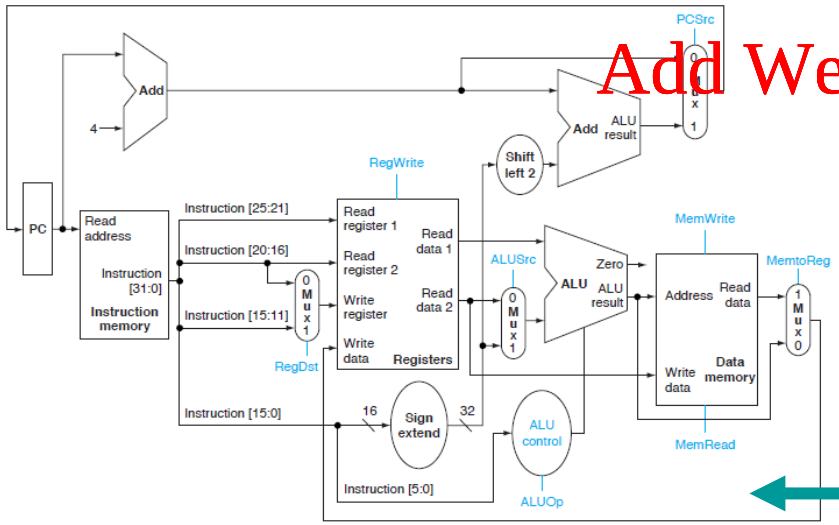
Neuroscientists have become used to a number of "facts" about the human brain: It has **100 billion neurons** and 10- to 50-fold more glial cells; it is the largest-than-expected for its body among primates and mammals in general, and therefore the most cognitively able; it consumes an outstanding 20% of the total body ... Jun 26, 2012

www.pnas.org › content › Supplement_1

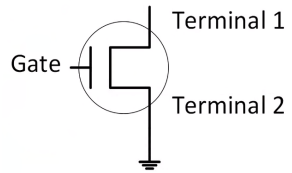
The remarkable, yet not extraordinary, human brain as a ...



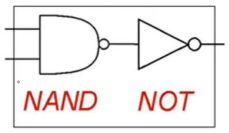
Register transfer level (RTL)

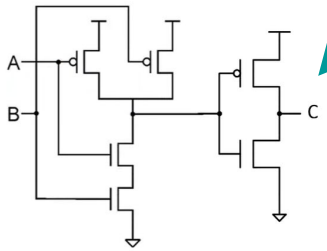


Transistor level



Gate level



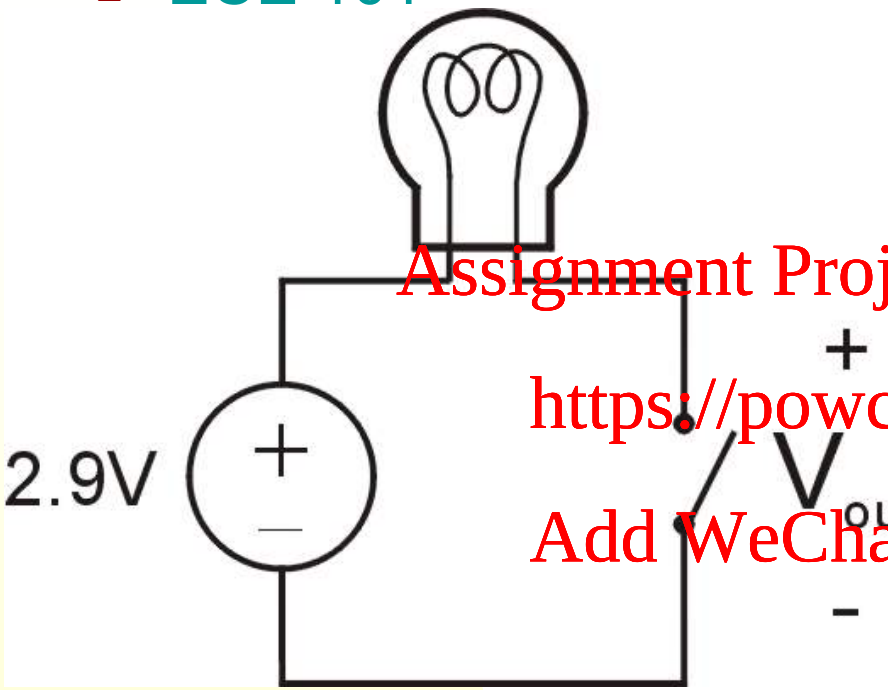


What is a transistor?

- Logically, each transistor is used as a switch
- Combined to implement logic functions
  - AND, OR, NOT
- Combined to build higher-level structures
  - Adder, multiplexer, decoder, register, …
- Combined to build a processor
  - ARM, Core 2 Duo, i7 9700k, etc

Simple switch circuit

- ECE 101



Switch-based circuits can easily represent two states: on/off, open/closed, voltage/no voltage.

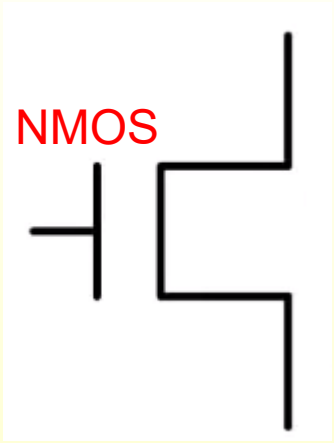
Switch open:

- No current through circuit
- Light is off
- V_{out} is +2.9V

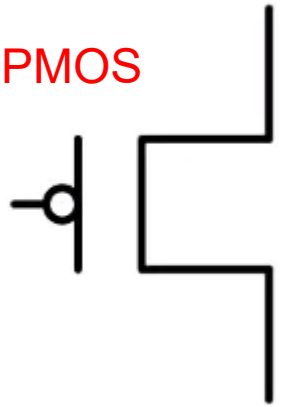
Switch closed:

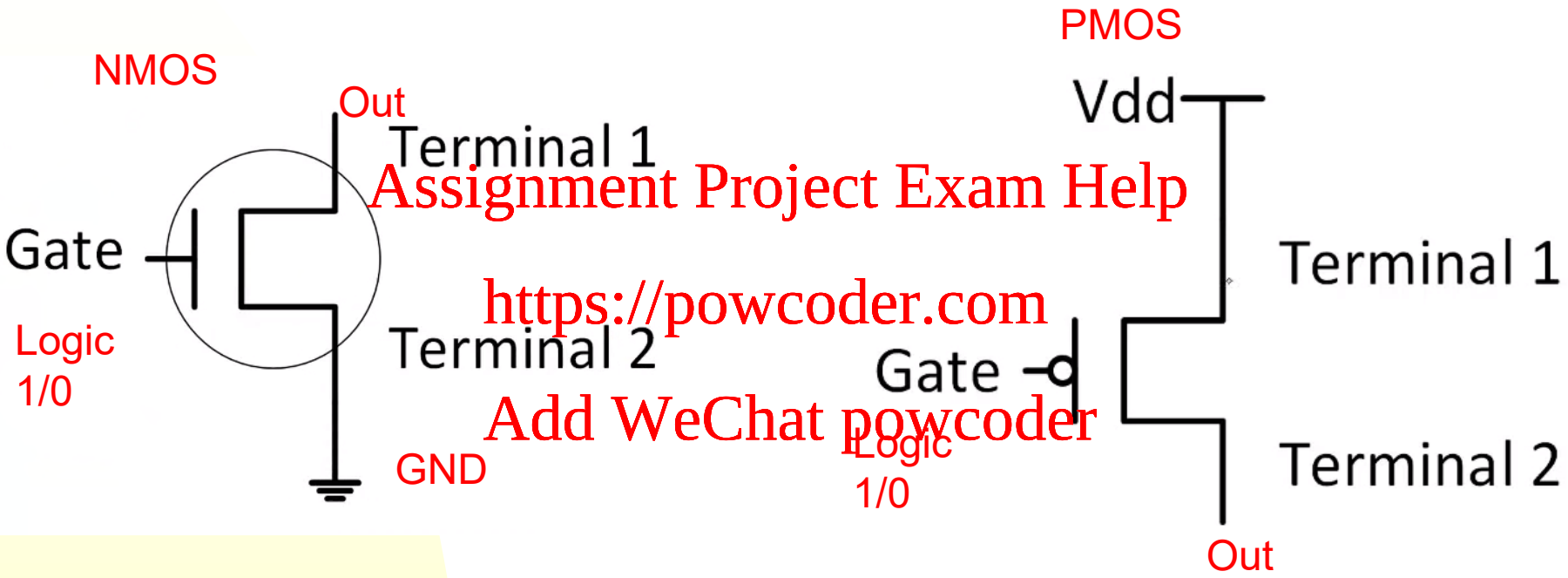
- Short circuit across switch
- Current flows
- Light is on
- V_{out} is 0V

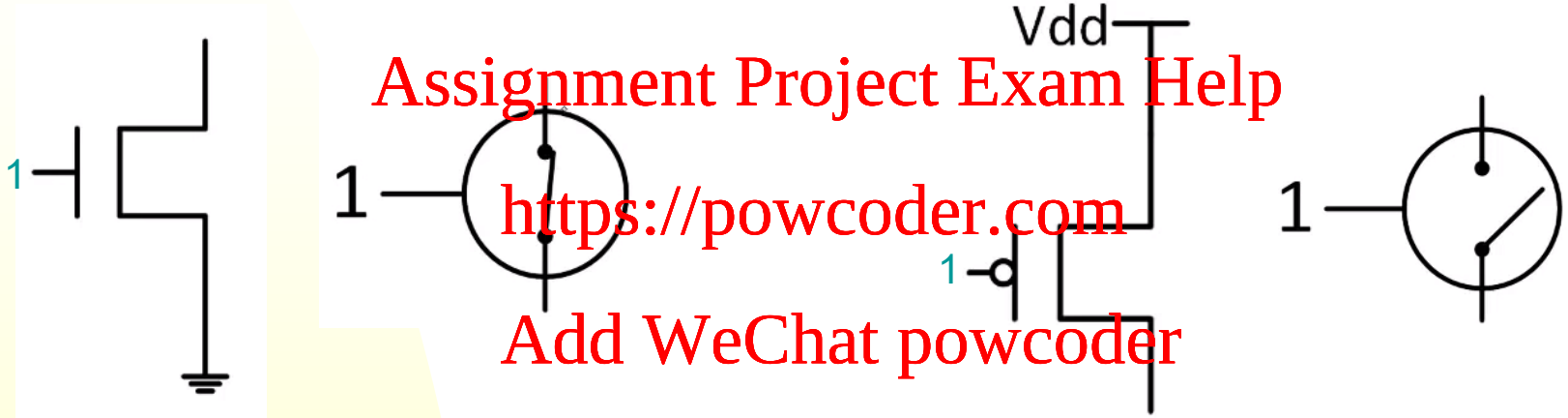
1. CMOS Logic Operation

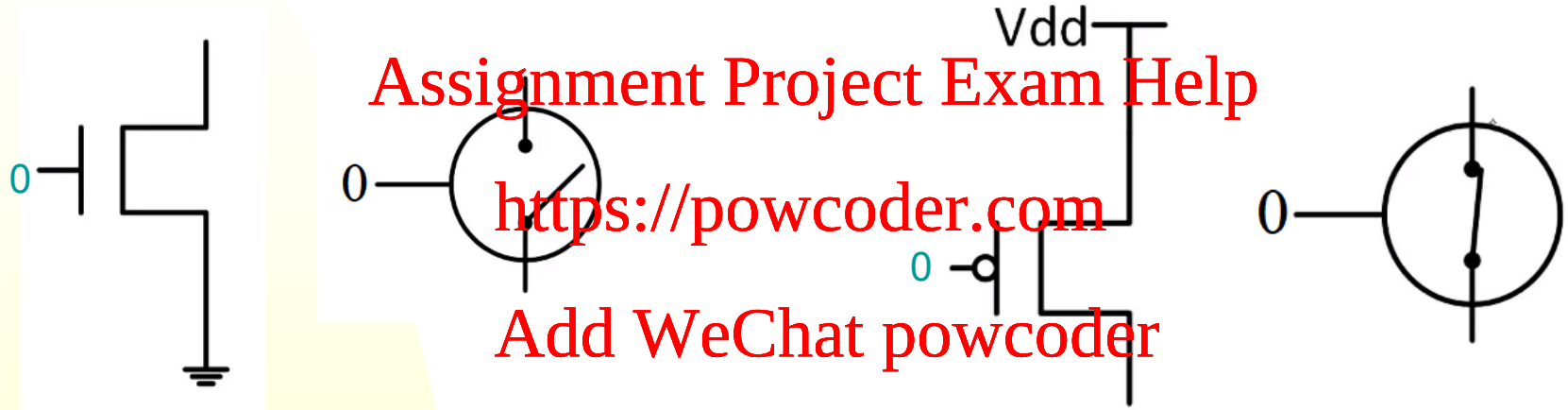


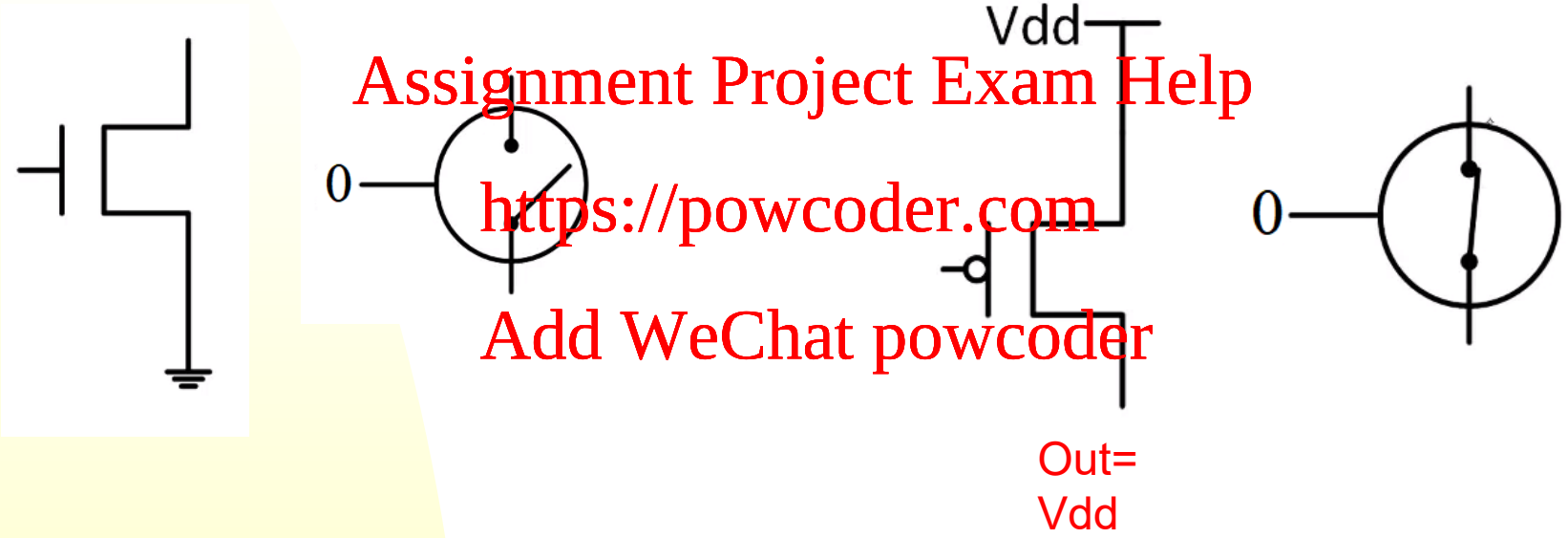
Task - Problem Statement
Algorithm
Program
Instruction Set Architecture
Microarchitecture
Logic Gates
Devices



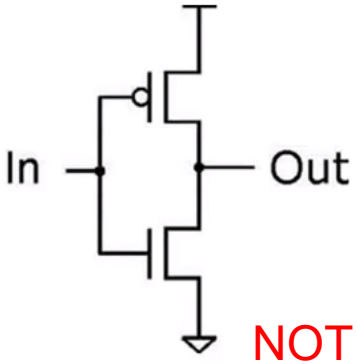


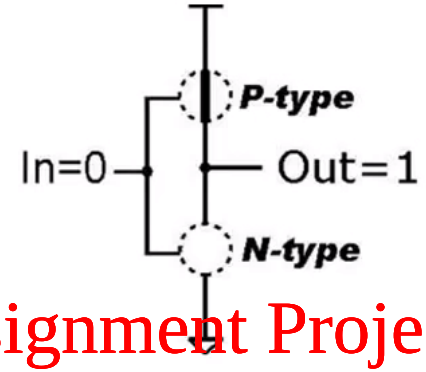


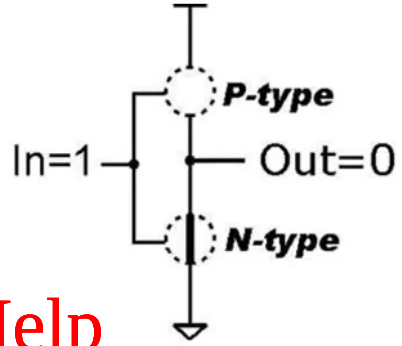




Complementary MOS (CMOS)

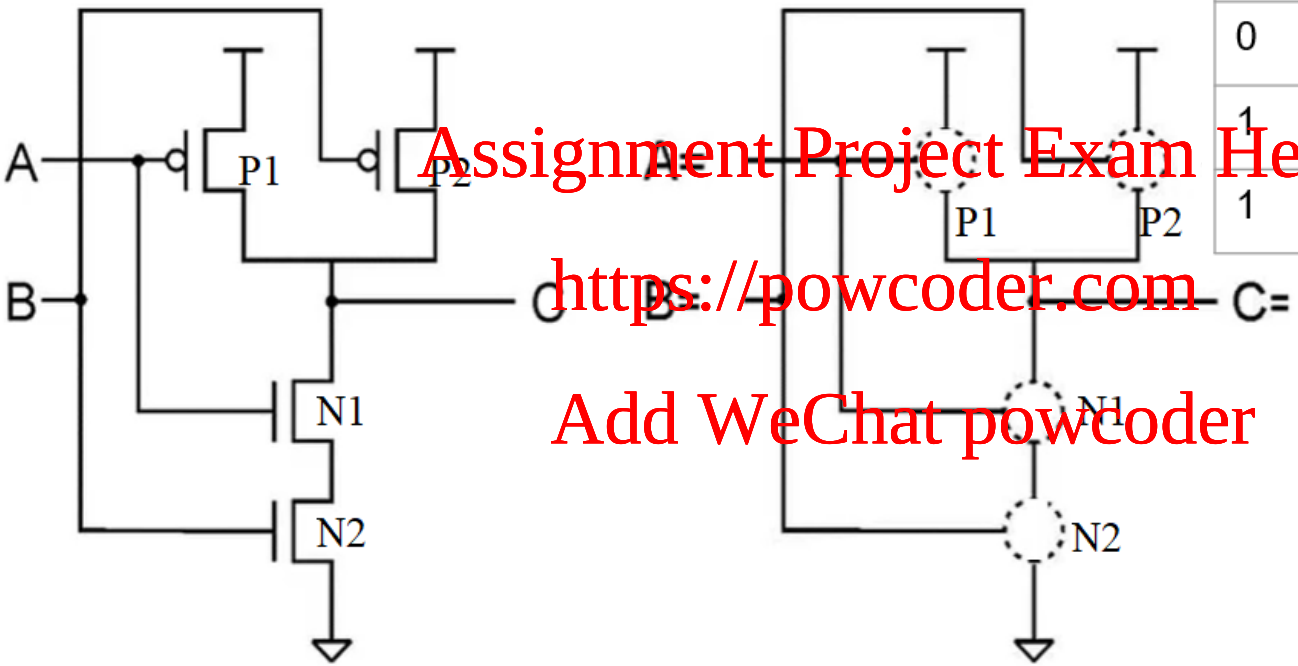






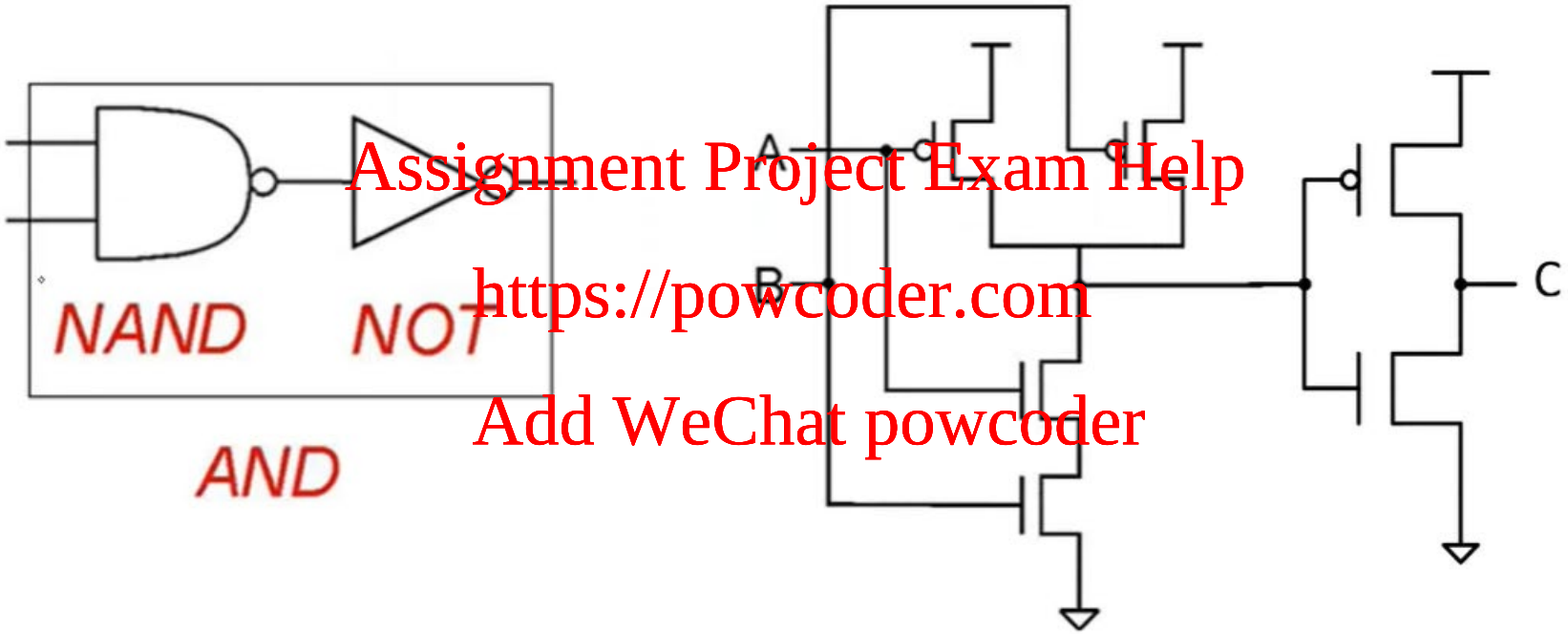
NOT gate (Inverter

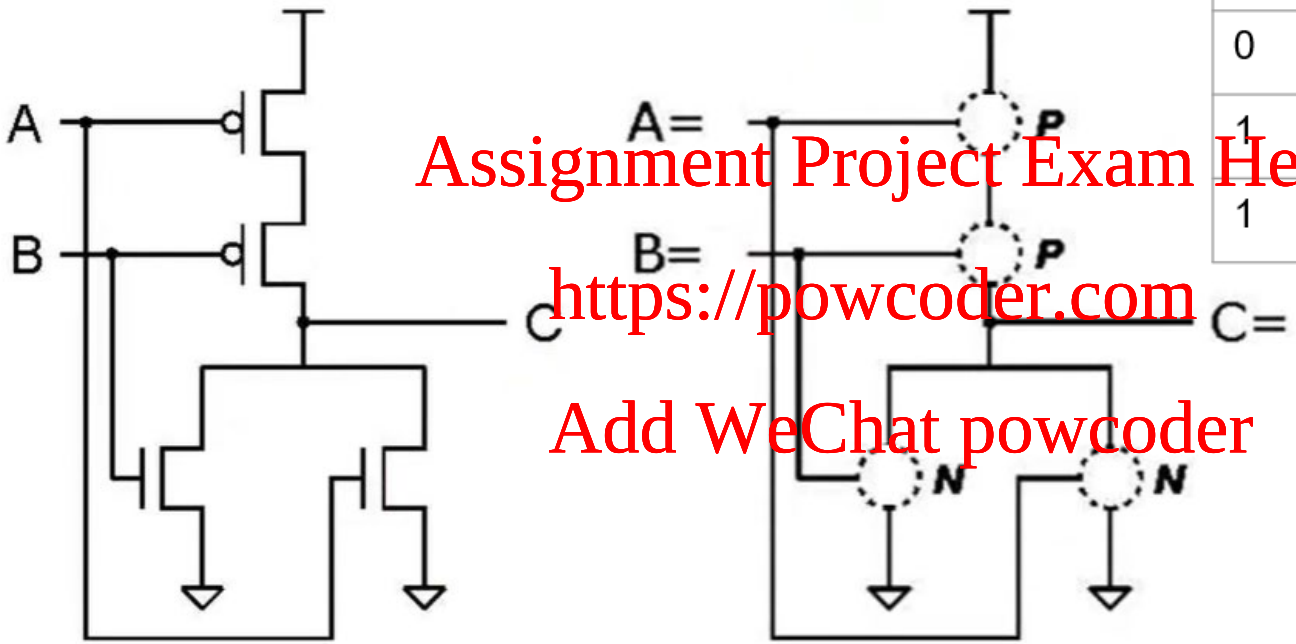
In	Out
0 V (Logic 0)	Vdd V (Logic 1)
Vdd V (Logic 1)	0 V (Logic 0)



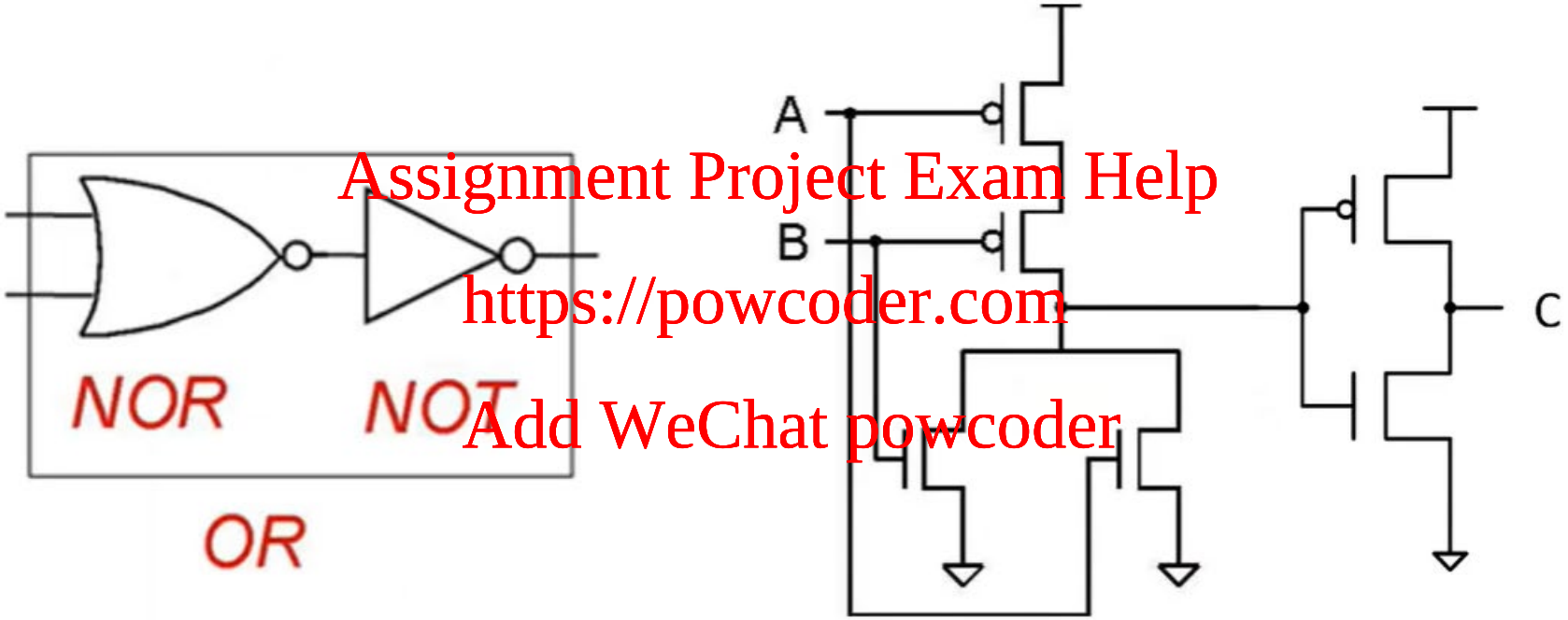
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

NAND gate

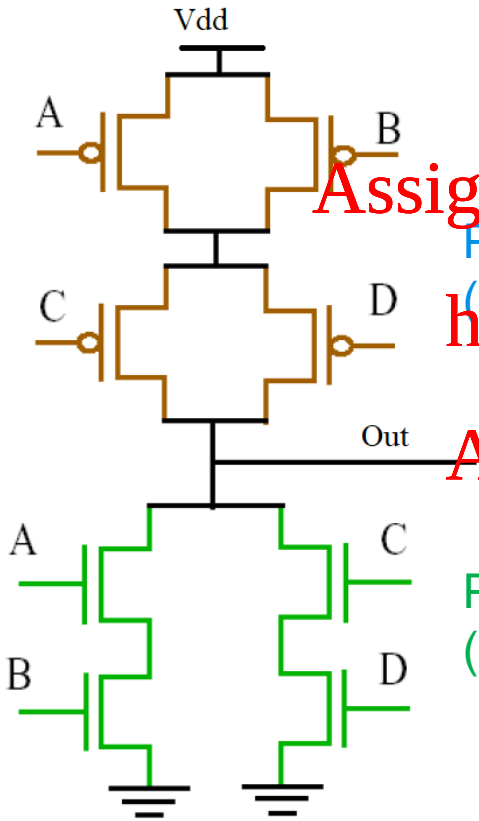




A	B	C
0	0	1
0	1	0
1	0	0
1	1	0



A general CMOS circuit



Pull Up Network (PUN)

Pull Down Network (PDN)

Q. How to determine Out without solving for entire truth table???

A. Follow Output path through either the PUN (towards Vdd) or the PDN (towards gnd)

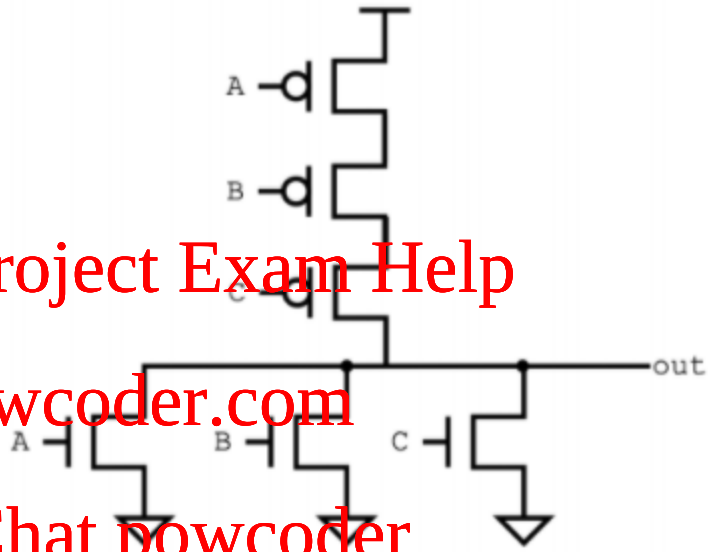
PDN approach: Under what conditions does Out have a direct path to gnd??

Out = 0, (when A AND B ON) OR (when C AND D ON)

$$Out=(AB+CD)'$$

Combinational Logic

7. The following CMOS diagram represents which logic gate?



- ○ A. NAND
- ○ B. NOR
- ○ C. AND
- ○ D. OR
- ○ E. Answer not listed.