Rensselaer

why not change the world?®

Intro to Digital Design & Computer Architecture

Guest Lecture | Spring 2023

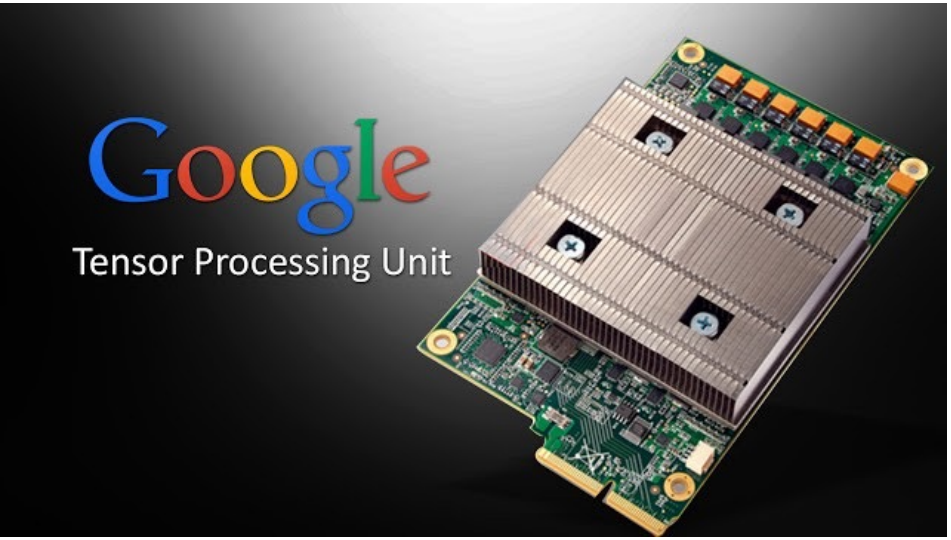
Liu Liu,
https://faculty.rpi.edu/liu-liu

Modern Computing Platforms

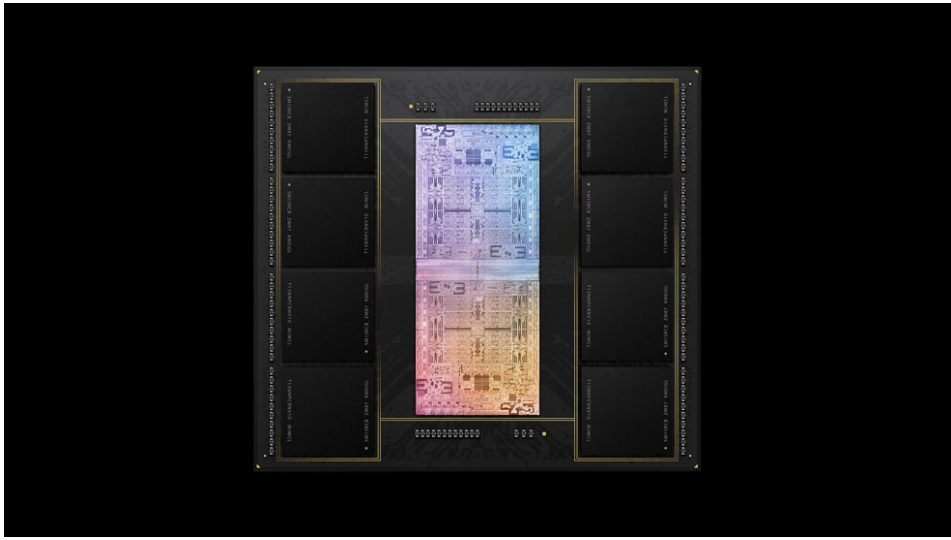
NVIDIA Hopper Architecture

The New Engine for the World's AI Infrastructure Makes Order-of-Magnitude Performance Leap.

https://www.nvidia.com/en-us/technologies/hopper-architecture/



https://medium.com/gdg-vit/simplifying-machine-learning-googles-tensor-processing-unit-tpu-7515dcfe3fc0



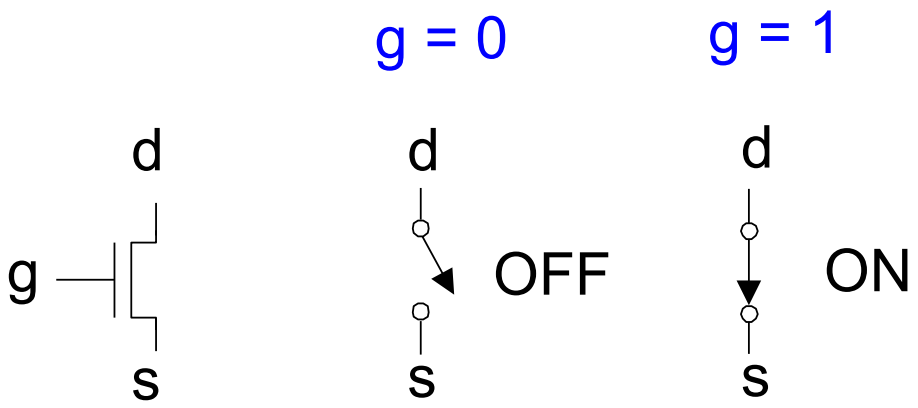
https://www.apple.com/newsroom/2022/03/apple-unveils-m1-ultra-the-worlds-most-powerful-chip-for-a-personal-computer/

Digital Abstraction

- CPUs/GPUs/TPUs/… are **digital systems**
- **Electronic** systems
- Use **discrete** values to represent and process information
  - Mostly, **binary** representation
- Typically, using **digital logic gates**
- **Boolean Logic**
  - George Boole, 1815-1864

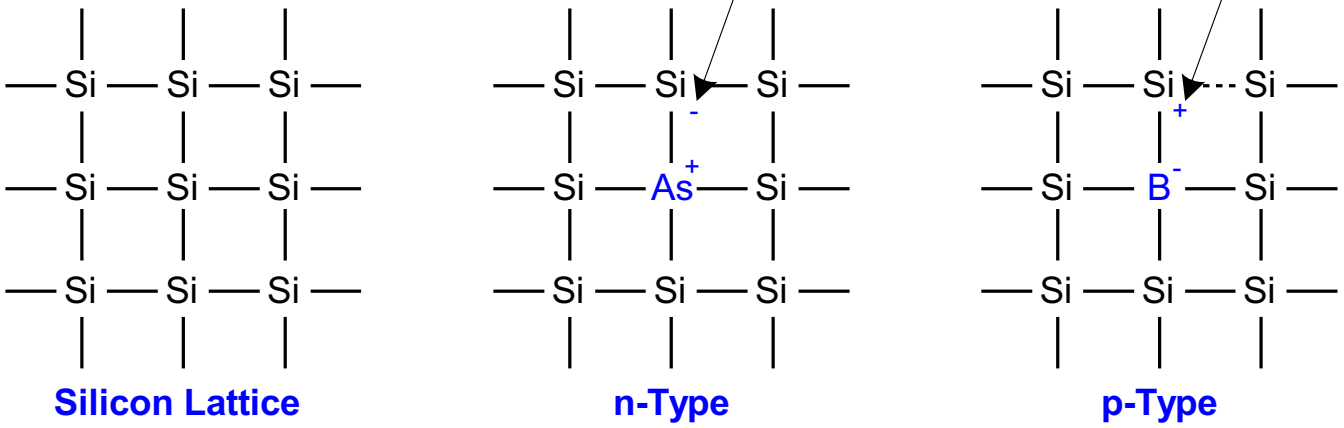
Transistors

- Logic gates built from transistors
- 3-ported voltage-controlled switch
  - 2 ports connected depending on voltage of 3rd
  - d and s are connected (ON) when g is 1



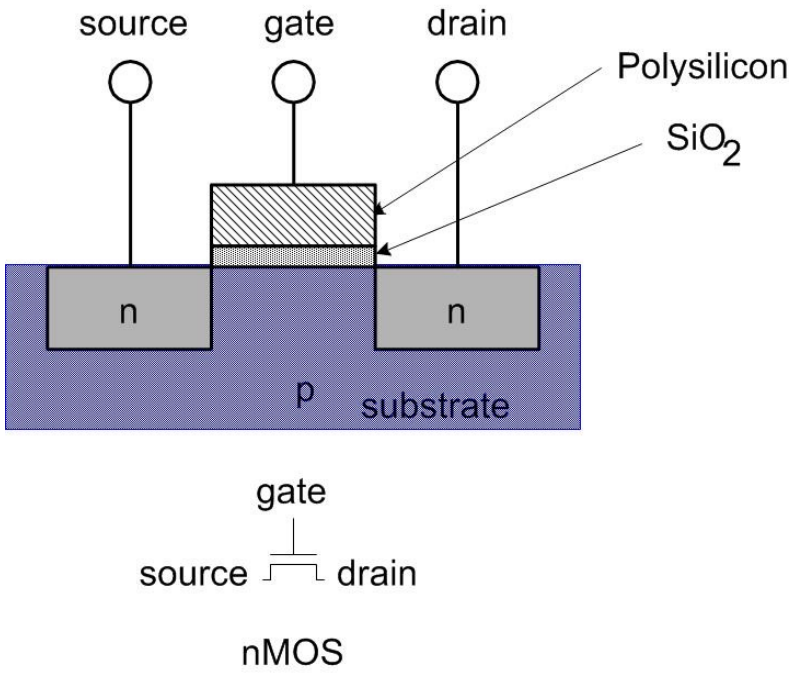
Silicon

- Transistors built from silicon, a semiconductor
- Pure silicon is a poor conductor (no free charges)
- Doped silicon is a good conductor (free charges)
  - n-type (free ***n***egative charges, electrons)
  - p-type (free ***p***ositive charges, holes)

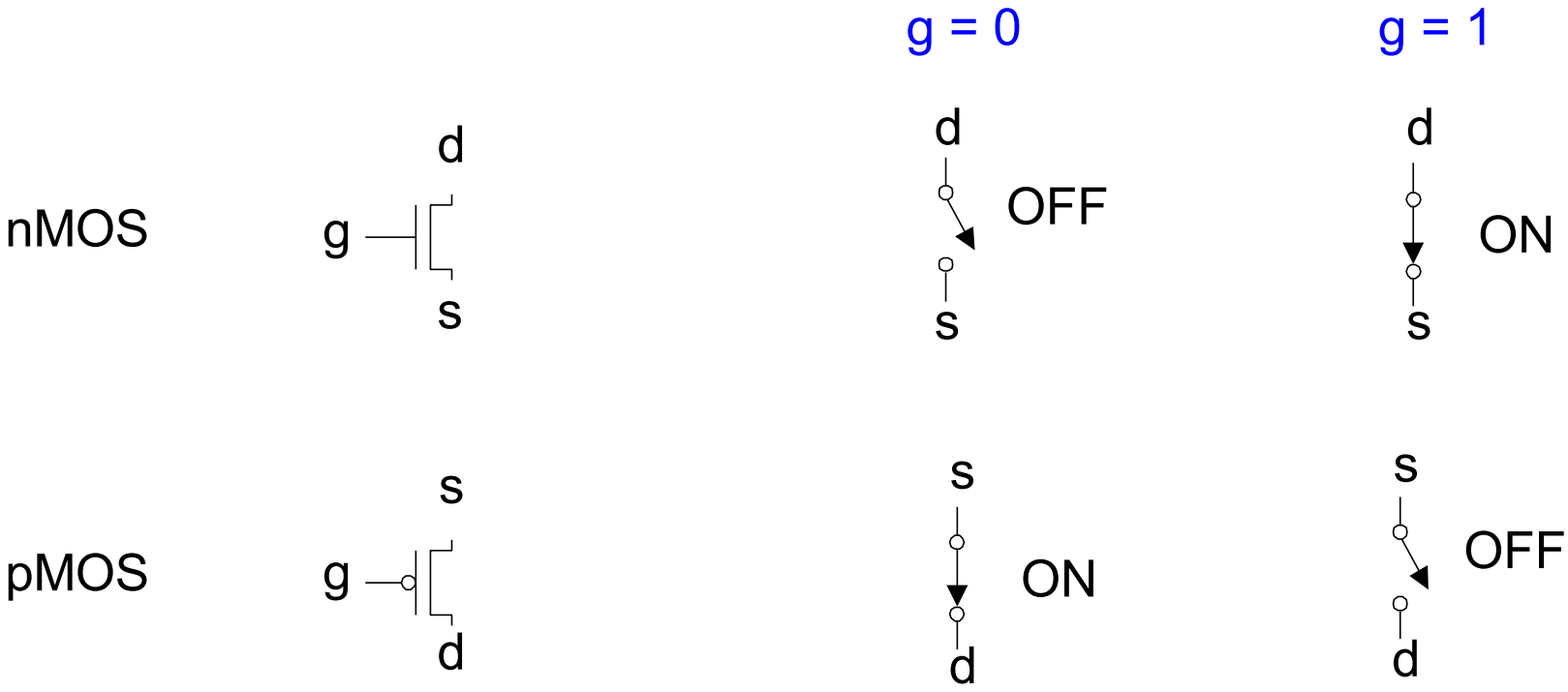


MOS Transistors

- **Metal oxide silicon (MOS) transistors:**
  - Polysilicon (used to be **metal**) gate
  - **Oxide** (silicon dioxide) insulator
  - Doped **silicon**

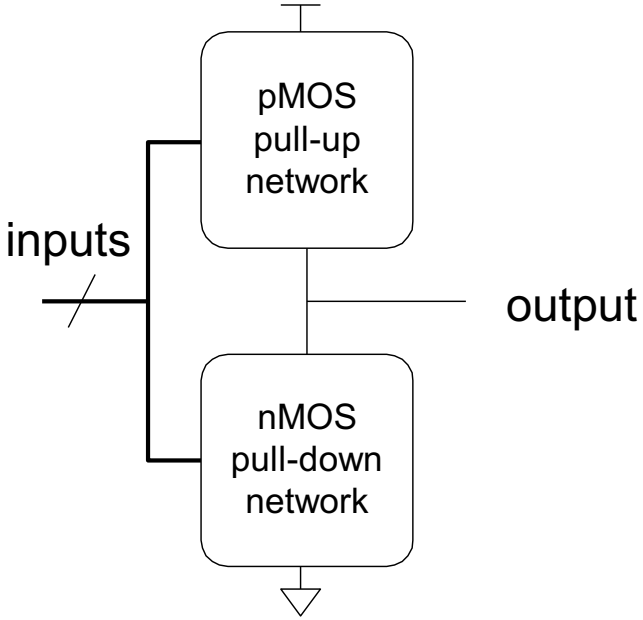


Transistor Function



CMOS Transistors

- **General Form of Inverting Logic Gates**
- **nMOS**: pass good **0**'s, so connect source to GND (pull-down)
- **pMOS**: pass good **1**'s, so connect source to V_{DD} *(pull-up)*
- **Complementary**

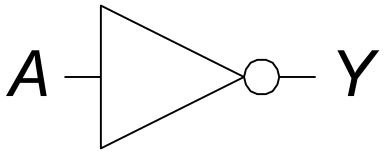


Logic Gates

- Perform operations on binary variables
- **Logic functions:**
  - inversion (NOT), AND, OR, NAND, NOR, etc.
- **Single-input:**
  - NOT gate, buffer
- **Two-input:**
  - AND, OR, XOR, NAND, NOR, XNOR
- **Multiple-input**

Single-Input Logic Gates

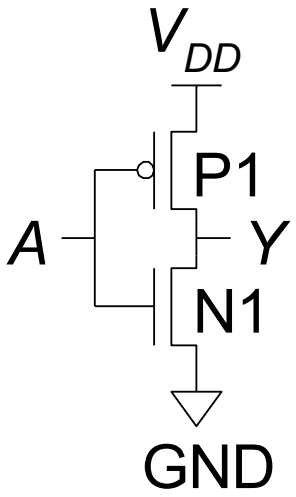
NOT



$Y = \overline{A}$

A	Y
0	1
1	0

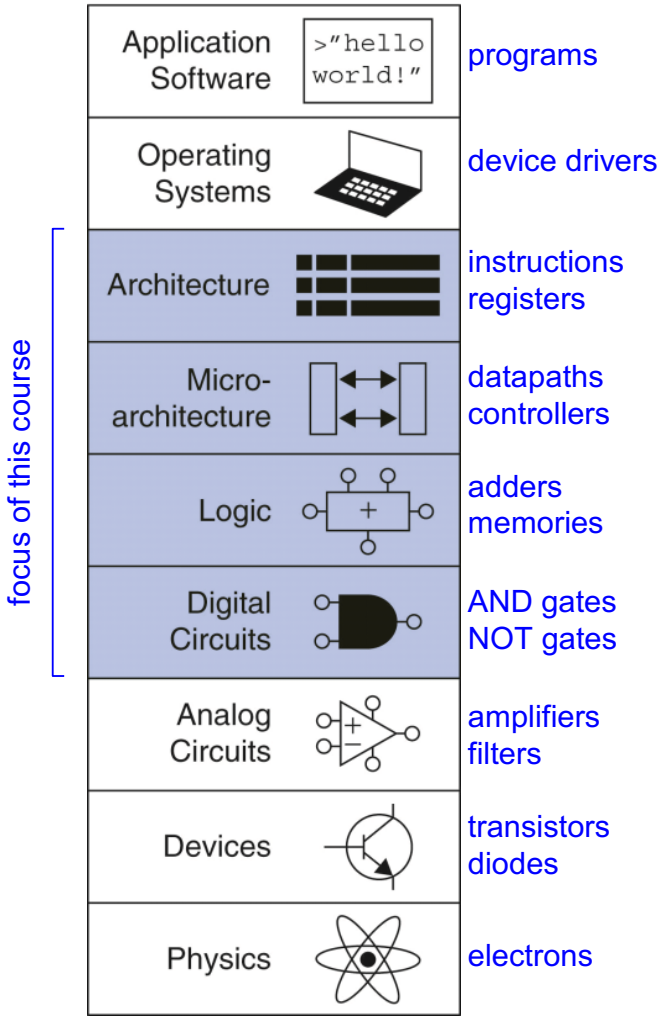
CMOS Implementation



Managing Complexity

- Abstraction
- Digital Discipline
- Design Principles:
  - Hierarchy
  - Modularity
  - Regularity

Layers of Abstraction



- Related courses
  - Data Structures & Algorithms
  - Computer Systems
  - Computer Architecture
  - Components & Operations
  - VLSI Design
  - Digital Electronics
  - IC Process & Design

Digital Discipline

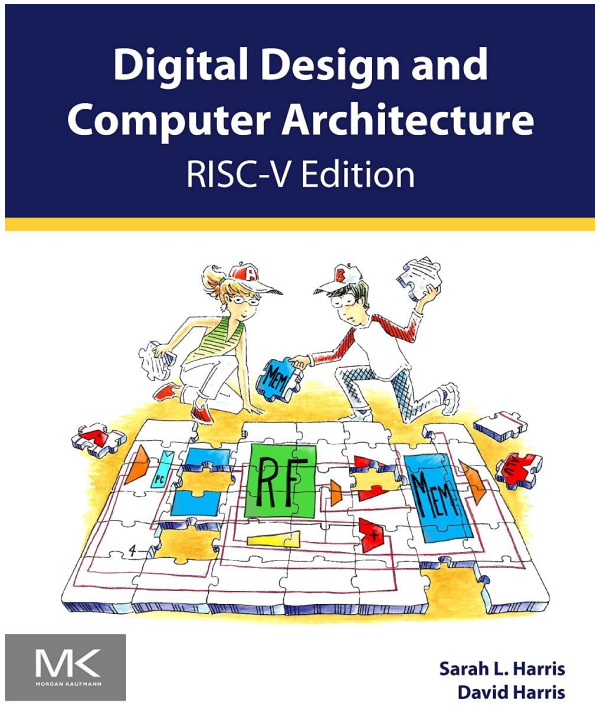
- Design constraints
  - Performance, Power, Area, and Cost (PPAC)
- Digital abstraction
  - Dealing with binary digit: 1's and 0's
  - True vs. false; high vs. low
  - Digital circuits use voltage levels
    - 0: low voltage (GND)
    - 1: high voltage (VDD)

Design Principles

- Hierarchy
  - Dividing and further subdividing
- Modularity
  - Well-defined functions and interface
- Regularity
  - Interchangeable
  - Industrial standards
  - Reusable and replaceable

Topics in Digital Design and Comp. Arch.

- Digital Design
  - Combinational Logic
  - Sequential Logic
  - Finite-State Machine
- Instruction Set Architecture
  - Assembly Language
  - Functions in Assembly
- Microarchitecture
  - RISC-V Processor



Courses: ECSE 4770 - Fall Semester

- Computer Hardware Design
  - Historically named
    - Should be **intro to computer architecture**
  - Processor architecture / core microarchitecture
- Digital Design
  - Reviewing basic building blocks
- Computer Architecture
  - Implementing a RISC-V microprocessor
  - Using hardware description language (**HDL**): SystemVerilog

Courses: ECSE 4780 / 6780 - Spring

- Advanced Computer Architecture
  - Modern microprocessor design
  - Domain-specific and accelerator architectures
- Topics include:
  - Pipelining
  - Cache coherence
  - Memory systems
  - GPGPU
  - AI/Machine Learning Processors
  - FPGA prototyping

HDL vs. Software Programming

- **Hardware Description Language (HDL):**
  - Specifies logic function only
  - Computer-aided design (CAD) tool produces or *synthesizes* the optimized gates
- Most commercial designs built using HDLs
- Two leading HDLs:
  - **SystemVerilog**
    - Developed in 1984 by Gateway Design Automation
    - IEEE standard (1364) in 1995
    - Extended in 2005 (IEEE STD 1800-2009)
  - **VHDL 2008**
    - Developed in 1981 by the Department of Defense
    - IEEE standard (1076) in 1987
    - Updated in 2008 (IEEE STD 1076-2008)

HDL to Gates

- **Simulation**
  - Inputs applied to circuit
  - Outputs checked for correctness
  - Millions of dollars saved by debugging in simulation instead of hardware
- **Synthesis**
  - Transforms HDL code into a *netlist* describing the hardware (i.e., a list of gates and the wires connecting them)

Why Study Computer Architecture?

- Next-gen computing platforms
- **Understand HW** to develop better SW
- **Understand SW** to design better HW
- Co-optimization
- Vertical integration of abstraction layers
- Working on SW or HW?

Research Topics in Comp. Arch. & Systems

- Computer Architecture
  - Domain-Specific Architectures
  - Algorithm-Architecture co-design
- AI / Machine Learning Systems
  - Specialized hardware and systems for ML
  - Hardware-efficient ML methods
  - Efficient model training, inference, and serving
  - Distributed and federated learning systems
  - Privacy and security for ML systems

Slides credit to:

Digital Design and Computer Architecture Lecture Notes