

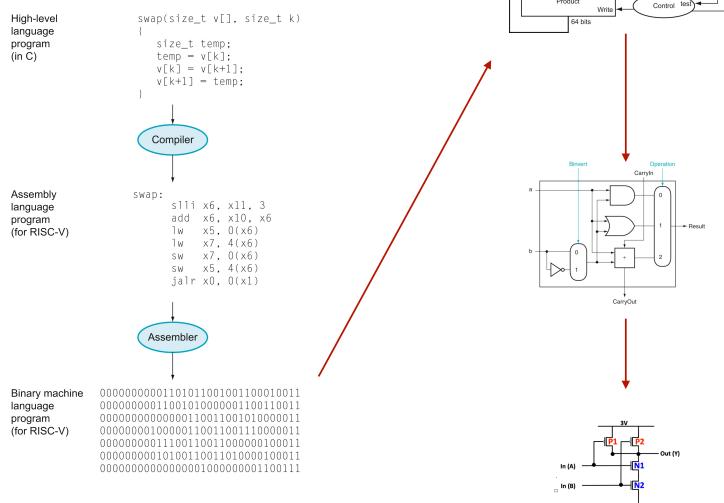
CSC 411

Computer Organization (Spring 2024)

Lecture 19: Introduction to logic design

Prof. Marco Alvarez, University of Rhode Island

Context

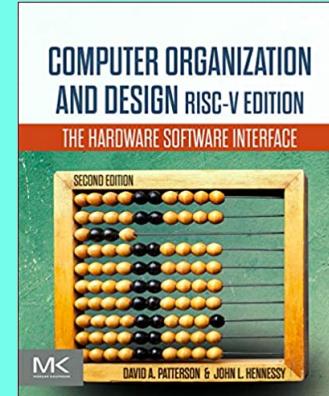


Disclaimer

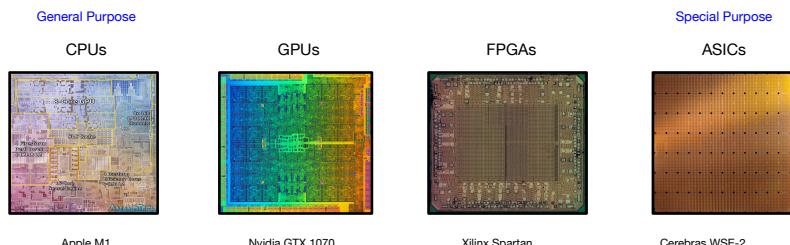
Some figures and slides are adapted from:

Computer Organization and Design (Patterson and Hennessy)

The Hardware/Software Interface



Computing systems today



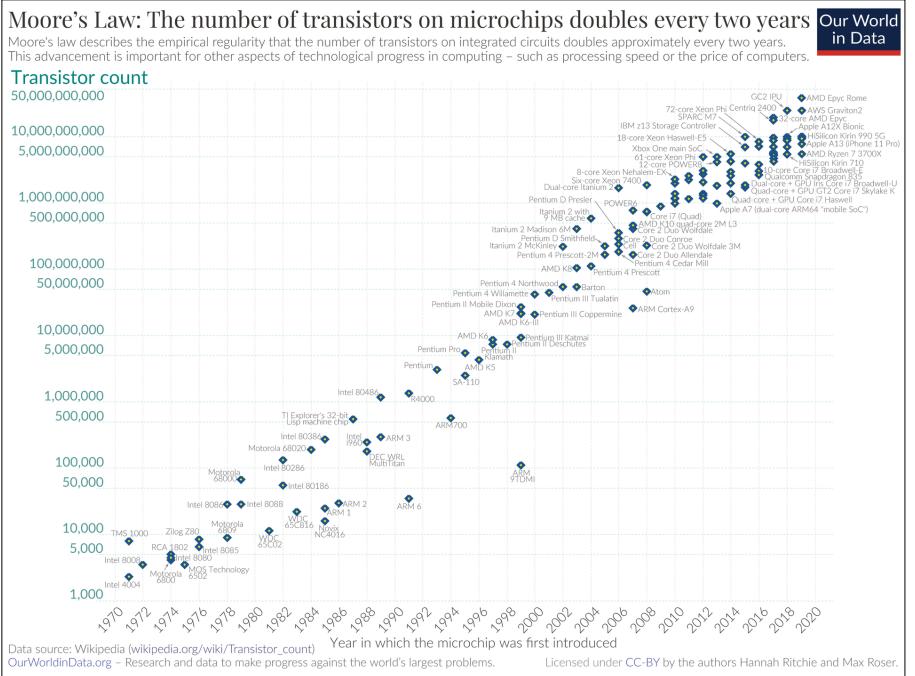
Flexible: Can execute any program
Easy to program & use

Not the best performance & efficiency

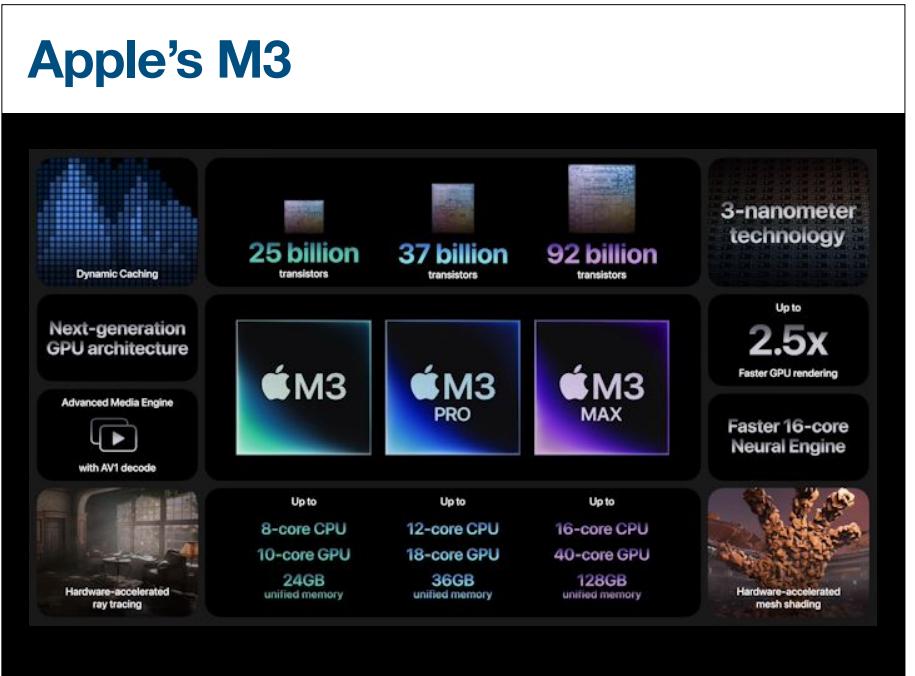
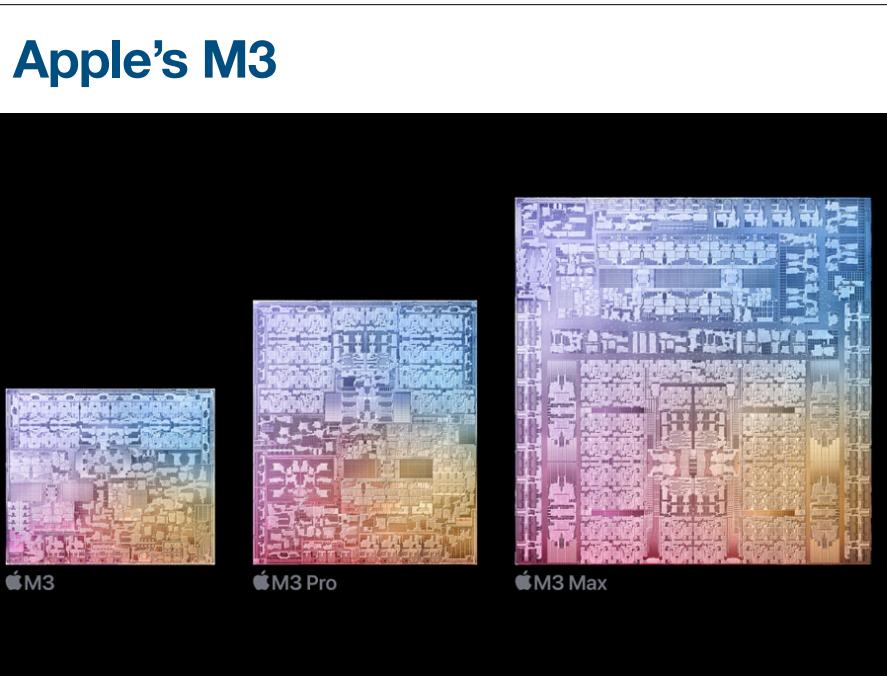
Efficient & High performance
(Usually) Difficult to program & use
Inflexible: Limited set of programs

All Computers are built upon the same building blocks

Transistors



Apple's M3



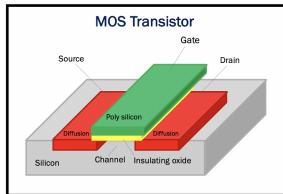
Transistors

‣ MOS transistors

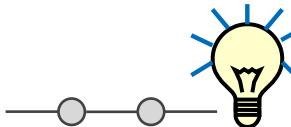
- Metal-Oxide-Semiconductor transistors
- **fundamental building block** of computers

‣ Structure

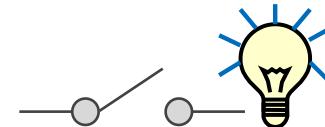
- **substrate**: semiconductor region that forms the base of the transistor (typically made of silicon)
- **gate**: thin metallic layer insulated from the substrate by a thin layer of oxide
- **source**: a doped region on one side of the substrate where current enters the transistor
- **drain**: a doped region on the other side of the substrate where current exits the transistor
- the type of doping (p-type or n-type) determines whether the transistor is an N-type P-type transistor



Switches



conductor
(closed)



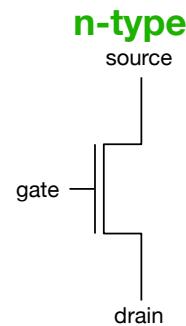
insulator
(open)

Transistors act as switches and can be combined to implement logic gates

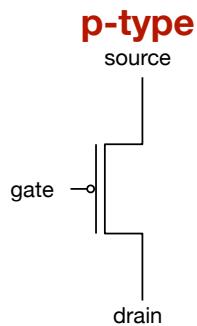
Types of MOS transistors

‣ Two types of MOS transistors

- operate as switches



If the gate is supplied with a high voltage, the connection from source to drain acts like a piece of wire



If the gate is supplied with zero voltage, the connection from source to drain acts like a piece of wire

CMOS transistors

- **CMOS (complementary MOS)**
 - combines two types of transistors: **n-type** and **p-type**
 - low power consumption, high noise immunity, scalability for higher integration densities
- **CMOS technology in modern processors**
 - CMOS logic gates are used to implement various functional units (e.g., ALU, control unit)
- **CMOS technology in memory chips**
 - CMOS technology is used in static RAM (SRAM) and dynamic RAM (DRAM) chips

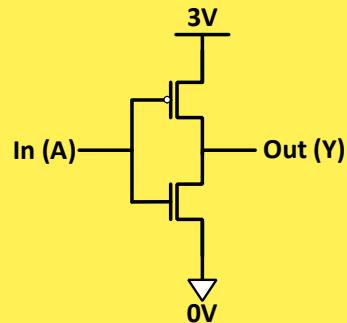
Logic gates

Logic design basics

- Information encoded in binary (basis of logic design)
 - low voltage, represented as false or 0
 - high voltage, represented as true or 1
 - all other (voltage) values are temporary and occur while transitioning between the low/high voltages
- Logic gates
 - implement simple boolean functions
 - can be built using CMOS transistors
- CMOS technology
 - complementary MOS (use both n-type and p-type transistors)

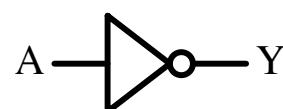
Practice

- How many transistors are used?
- What are their types?
- What does this circuit do?



NOT gate

- The NOT gate is also called an **inverter**
 - output zero voltage if input is high voltage
 - output high voltage if input is zero voltage

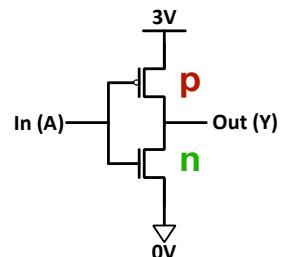


Truth table

- shows the output for all possible inputs (using binary notation)
- for n inputs, the truth table contains 2^n entries (all possible combinations of input values)

$$Y = \bar{A}$$

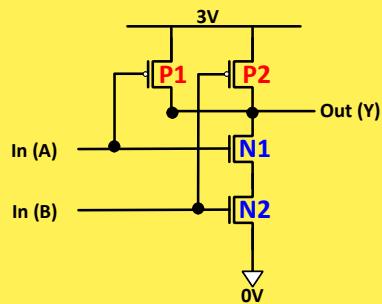
In	Out
0	1
1	0



Practice

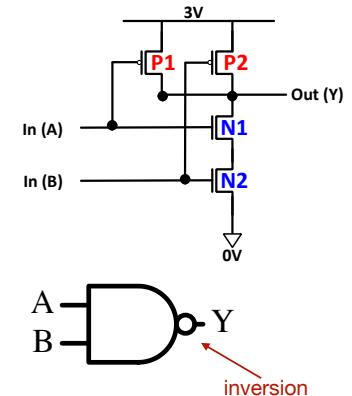
- How many transistors are used?
- Complete the truth table
- What does this circuit do?

A	B	P1	P2	N1	N2	Y



NAND gate

- P-type transistors in parallel
- only one must be "closed" (conducting) for the output to be high
- N-type transistors connected in series
- both transistors must be "closed" (conducting) to pull the output low

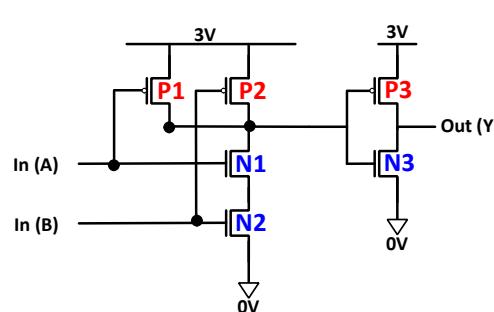


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

$$Y = \overline{AB}$$

AND gate

- Combining a NAND gate with a NOT gate

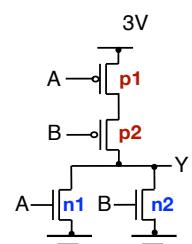


$$Y = AB$$

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

NOR gate

- P-type transistors are connected in series
- N-type transistors in parallel



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

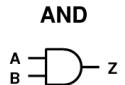
$$Y = \overline{A + B}$$

Logic gates (notation)

Buffer

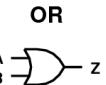


AND



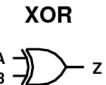
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

OR



A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

XOR



A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

Inverter



NAND



A	B	Z
0	1	0
1	0	0
1	1	0

NOR



A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

XNOR



A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

NAND and NOR are universal. Can implement any function with NAND or just NOR gates.

intel.

