EECE 200

Computer Hardware

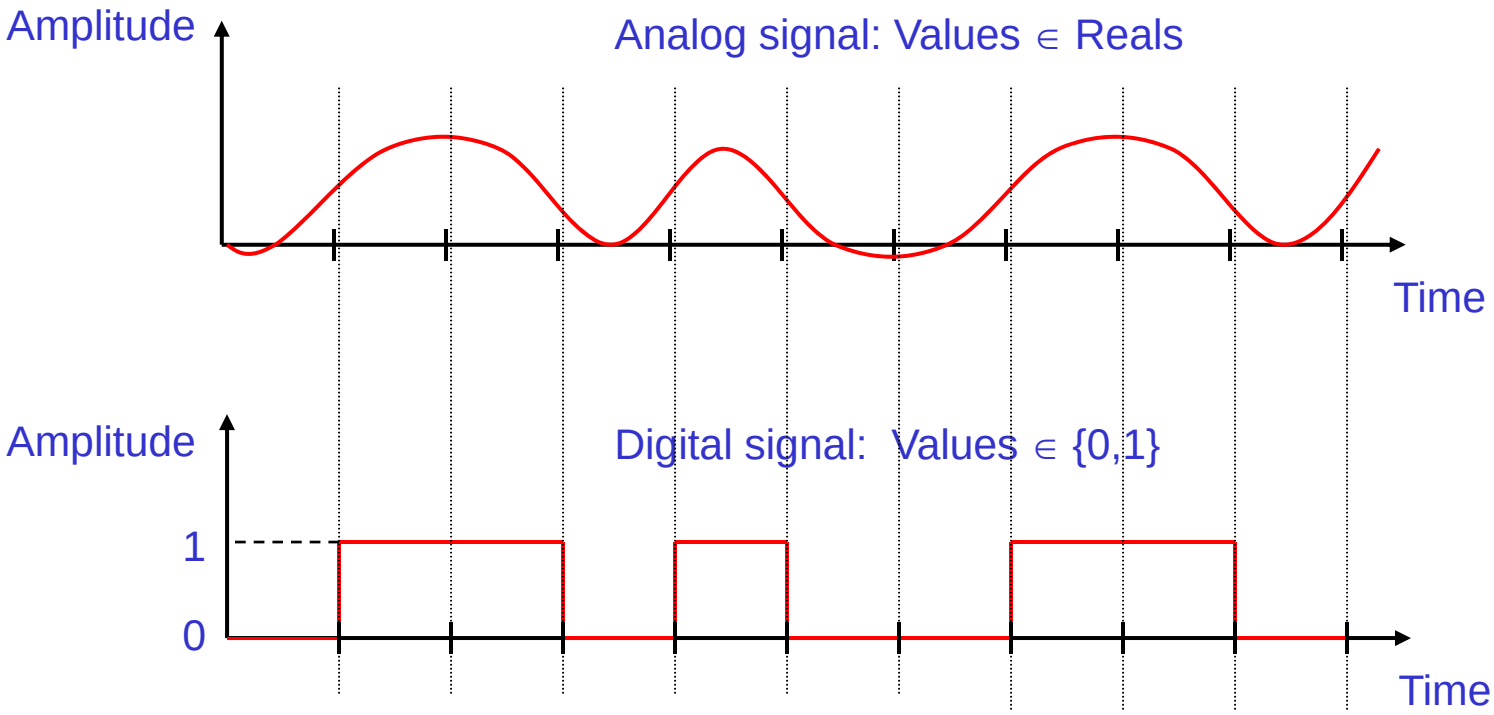
Haitham Akkary

Outline

- The Digital Revolution
- Digital Systems and Circuits
- Computer Organization
- Hardware Courses in ECE
- Career Opportunities

Digital vs. Analog

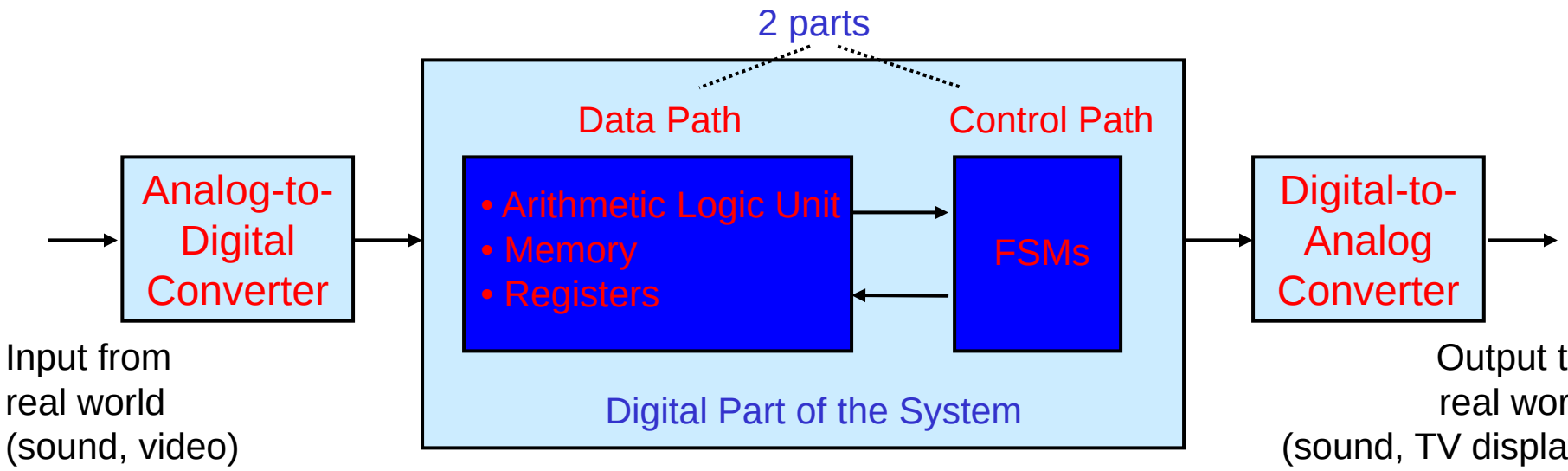
- Two basic ways of representing information:
  - Analog: Take any value across a continuous range (voltage,current)
  - Digital: Take only values from a discrete set
    - Decimal: {0,1,…,9}, Binary: {0,1} or {T,F} or {Low,High}



Waveform is represented by a series of numbers rather than a voltage or current, as in analog systems.

The Digital Revolution

- Digital systems have inputs and outputs that are represented by Binary digits (Bits) or groups of bits.



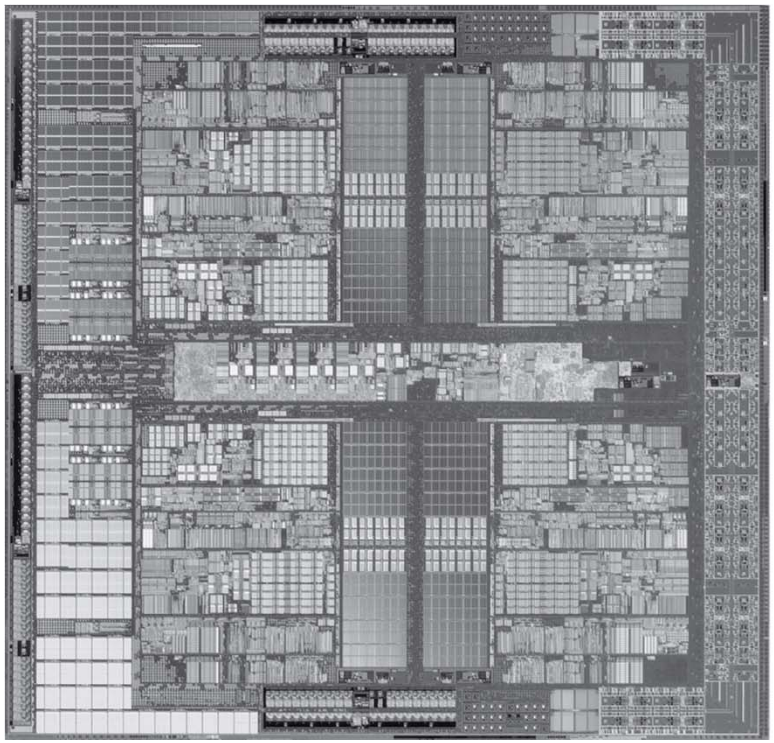
- Examples: General-purpose digital computers, digital cameras, digital versatile disks (DVDs), digital telephones, digital television, personal data assistants (PDAs)
- Applications: communication, business, traffic control, space, science, medicine, Internet, education, entertainment, weather …

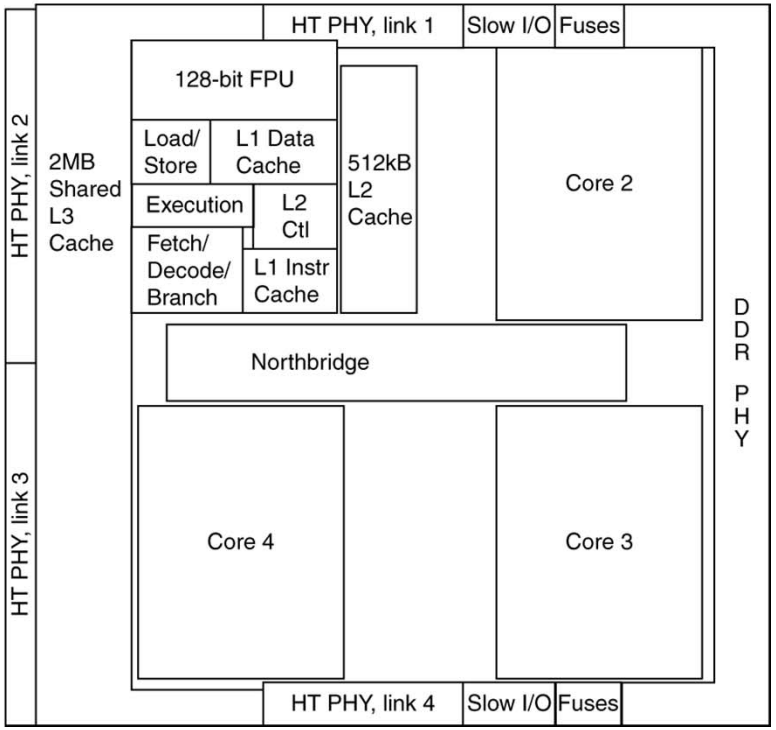
Behind the Digital Revolution

- Economy
  - Commonly used circuits can be integrated into chips and mass-produced at low cost and used in different products such as calculators, digital watches, …
- Steadily advancing technology
  - Moores Law: chip capacity doubles every two years
  - Digital designers try to accommodate advances in technology while designing. An example is multi-core processors

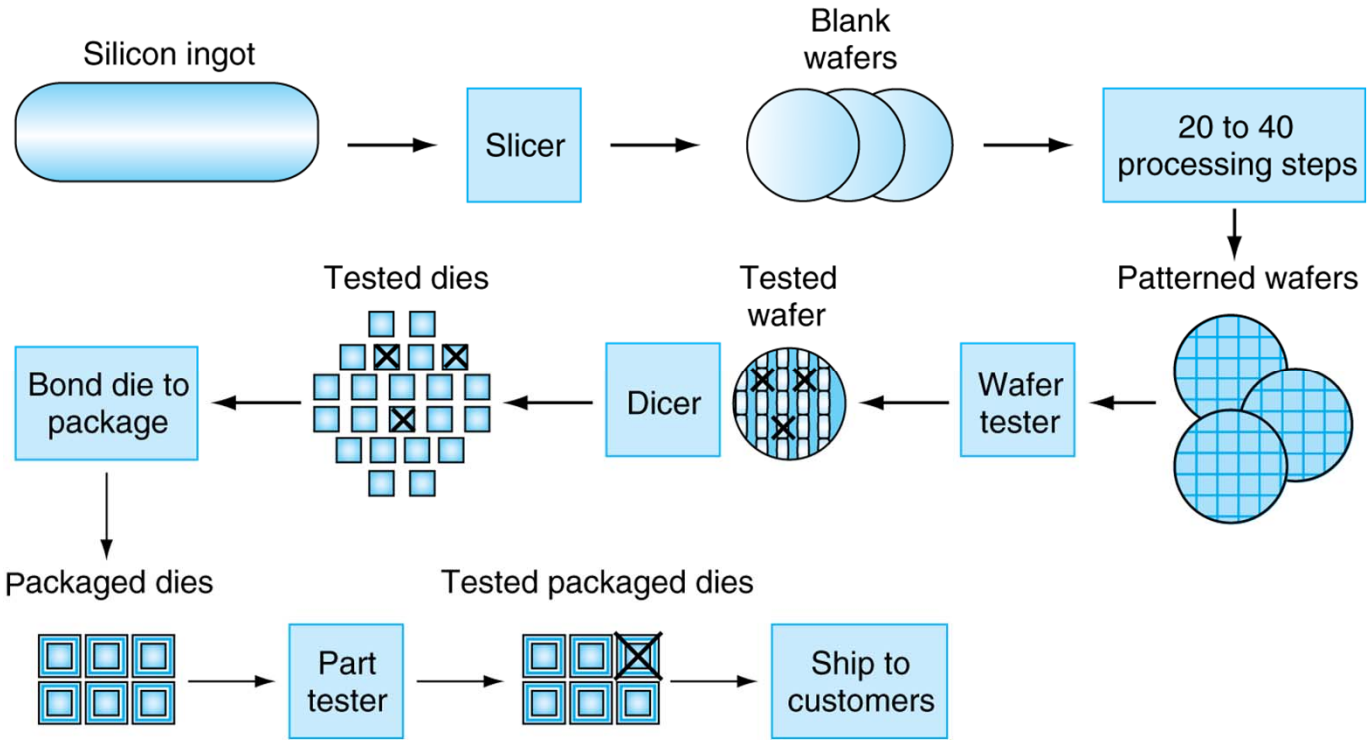
Inside the Processor

- AMD Barcelona: 4 processor cores



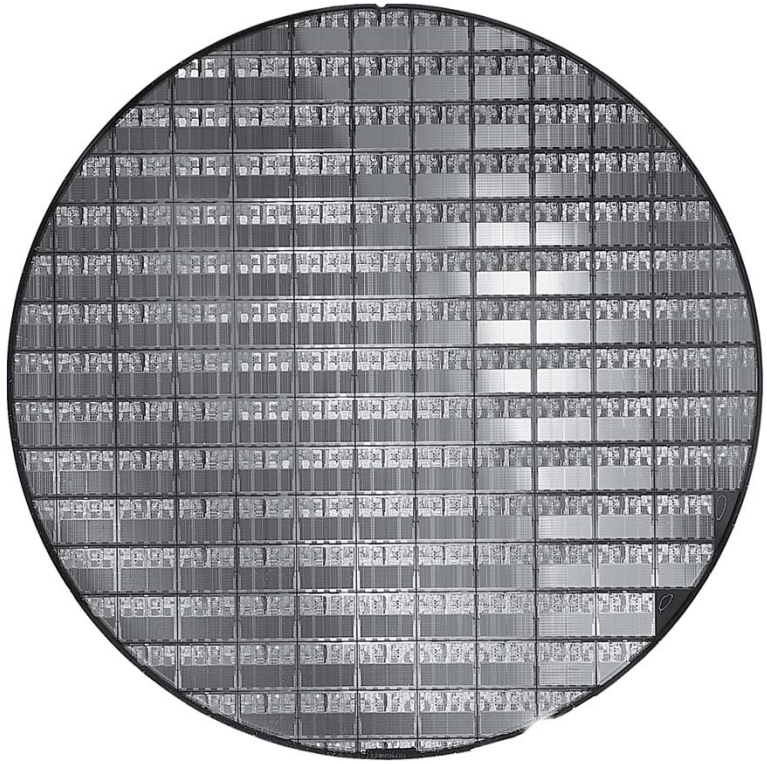


Manufacturing ICs



- Yield: proportion of working dies per wafer

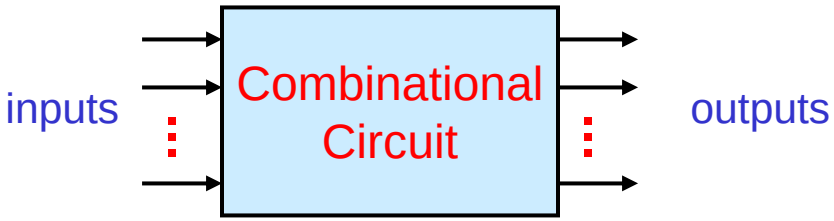
AMD Opteron X2 Wafer



- X2: 300mm wafer, 117 chips, 90nm technology
- X4: 45nm technology

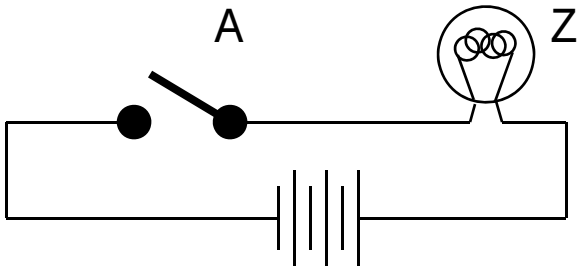
Combinational Circuits

- Take binary inputs, process them, and produce binary outputs
- Some typical combinational circuits:
  - Adders/Subtractors
  - Multipliers
  - Decoders/Encoders
  - Multiplexers/Demultiplexers

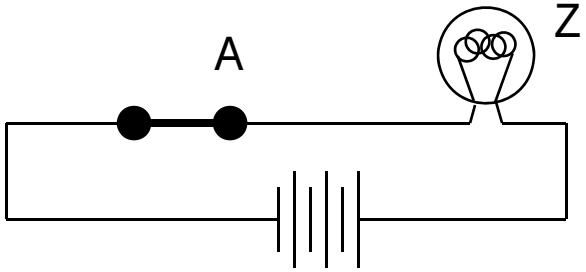


Switches: Basic Elements of Physical Implementations

- Implementation of a simple circuit:



open switch (if *A* is "0" or unasserted)
and turn off light bulb (Z)



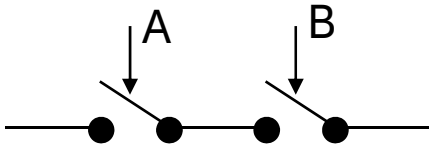
close switch (if *A* is "1" or asserted)
and turn on light bulb (Z)

$$Z \equiv A$$

Computing with Switches

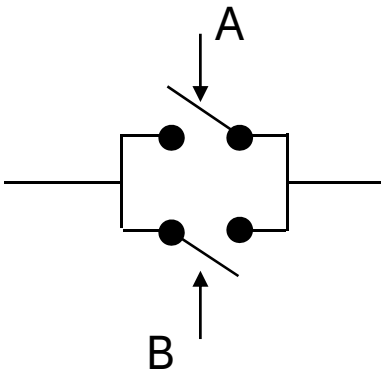
- Compose switches into more complex (Boolean) functions:

AND



Z ≡ A and B

OR

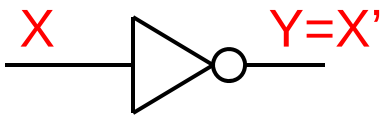


Z ≡ A or B

Two fundamental structures: series (AND) and parallel (OR)

Logic Gates

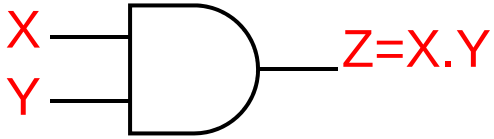
- Inverter: Output is opposite of input
  - Y = X'



Truth Table

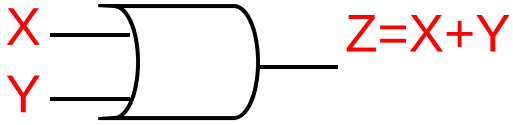
X	Y
0	1
1	0

- AND: Output is 1 iff all inputs are 1



X	Y	X.Y
0	0	0
0	1	0
1	0	0
1	1	1

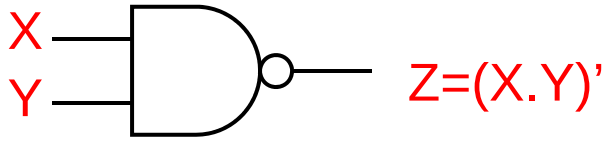
- OR: Output is 1 iff at least one input is 1



X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

Logic Gates

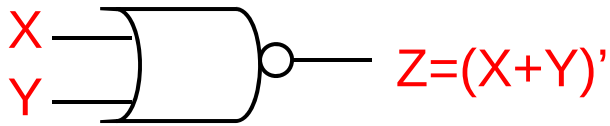
- NAND: Output is 0 iff all inputs are 1



Truth Table

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

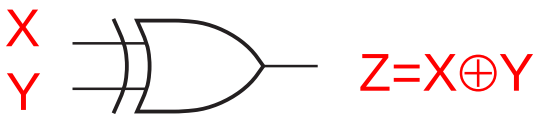
- NOR: Output is 0 iff at least one input is 1



X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Logic Gates

- XOR: Output is 1 iff one of the inputs is 1 but not both



X	Y	X⊕Y
0	0	0
0	1	1
1	0	1
1	1	0

Number Systems and Codes

- A digital designer must establish a correspondence between binary digits and real-life numbers, events and conditions

Binary Number Systems

- Binary: radix = 2
  - Used to represent numbers in a digital system
  - Reliable since only 2 values need to be distinguished
  - EX: 110.01 = 1x4 + 1x2 + 0x1 + 0x0.5 + 1x0.25 = 6.25_{10}
  - In general the value is given by:

$$B = \sum_{i=-n}^{p-1} b_i 2^i$$

Example 1: Binary Addition

- Similar to decimal addition:
  - 0+0=0, with a carry of 0
  - 0+1=1+0=1, with a carry of 0
  - 1+1=10 = 0, with a carry of 1
- Example:

carry:	1	0	1	1	1	1
augend:		1	0	1	1	0	1
addend:	+1	0	0	1	1	1
sum:	1	0	1	0	1	0	0

- Basic operation: Adding binary numbers is done by simply adding bits from right to left, while rippling the carry:
  - (Sum, Carry_OUT) = A plus B plus Carry_IN

Example 1: 1-Bit Binary Adder

- Truth table for a 1-bit adder:

X	Y	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

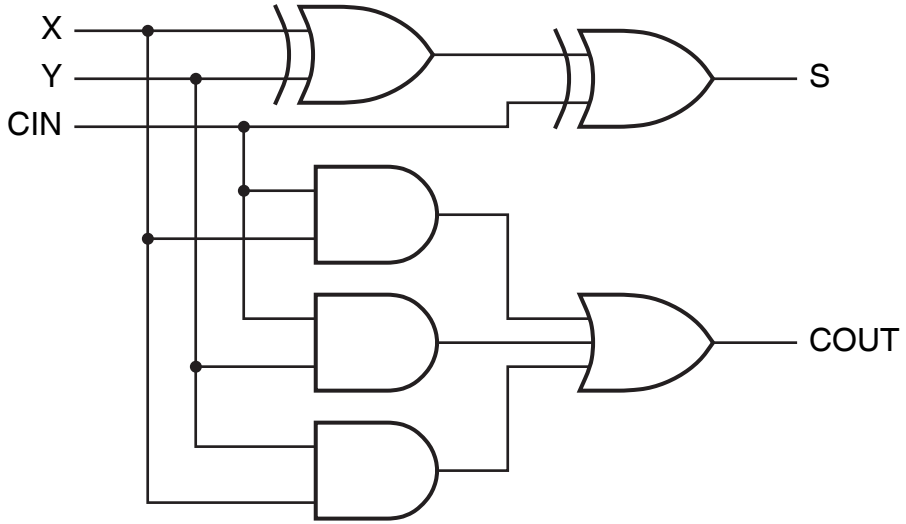
$$S = X \oplus Y \oplus CIN$$

$$COUT = X \cdot Y + X \cdot CIN + Y \cdot CIN$$

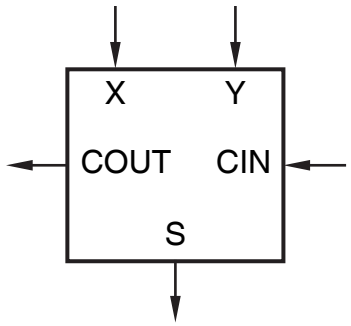
Example 1: 1-Bit Adder Circuit

$S = X \oplus Y \oplus CIN$

$COUT = X \cdot Y + X \cdot CIN + Y \cdot CIN$

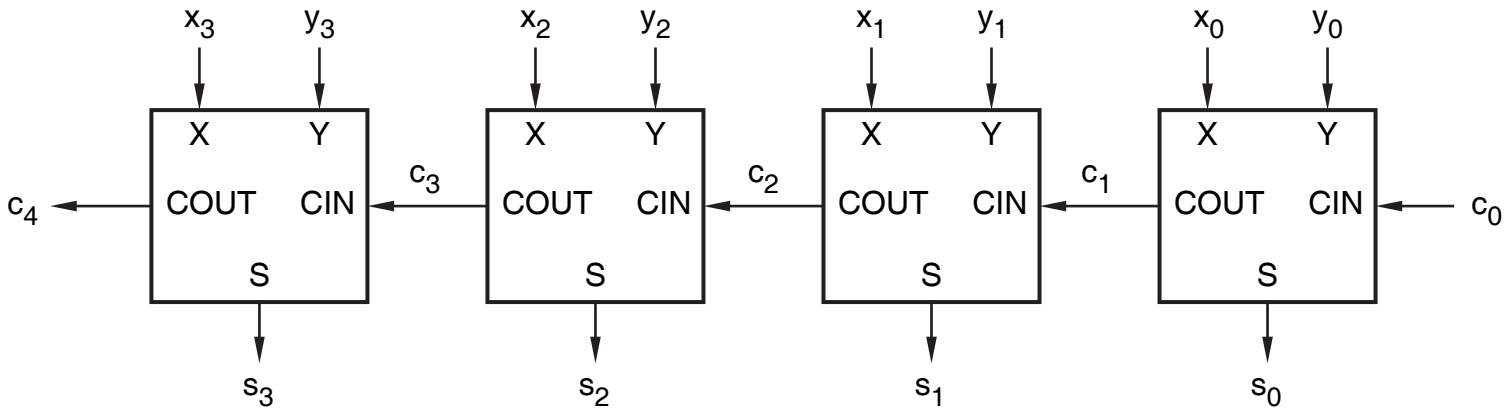


Symbol



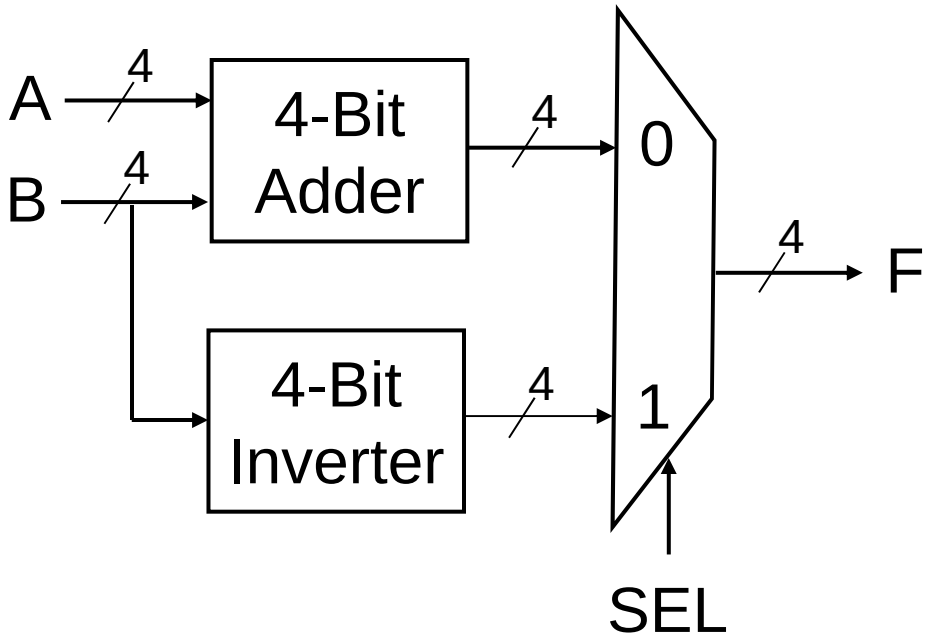
Example 1 : 4-Bit Adder Circuit

Simply build a 1-bit adder and replicate it 4 times.



Example 2: A Simple ALU

If SEL=0, F = A + B
Else if SEL=1, F = not(B)



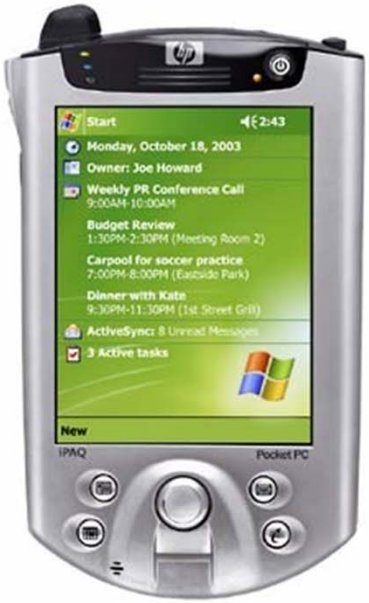
What is a Computer?

- A digital system that processes information according to a sequence of internally stored machine instructions called a *program*.
  - Both the information to be processed and the instructions used to process them are represented as *binary* data.

- Different types of computers
  - Servers
  - General purpose desktop and laptop computers
  - Ultra-mobile internet devices (smart phones, e.g., iphone)
  - Special purpose or embedded computers

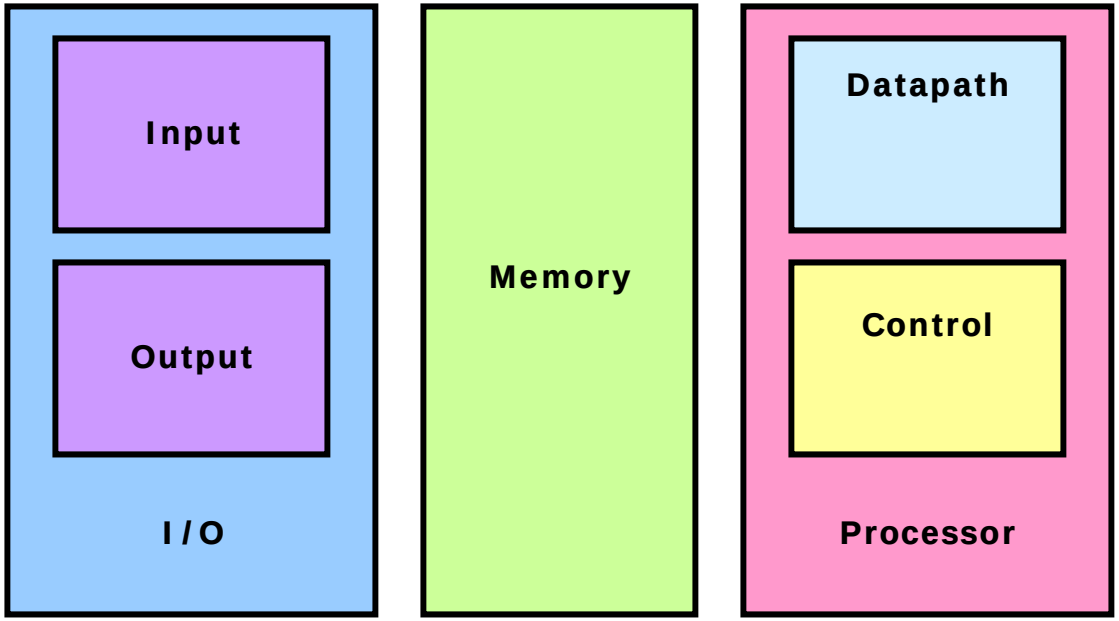
General Purpose Computers

Embedded Computers



Airplanes – Computers that Fly!

The Five Components of *Every* Computer



Input and Output Devices

- Input devices read data from the outside environment
  - Keyboard, mouse, joystick, electronic sensor
- Output devices send processed results to the external environment
  - Display monitor, LCD, printer, transducer
- Some devices provide input and output functions
  - Modems, network adapters

Memory

- Memory is used to store instructions (application programs) and data both inside and outside the computer.

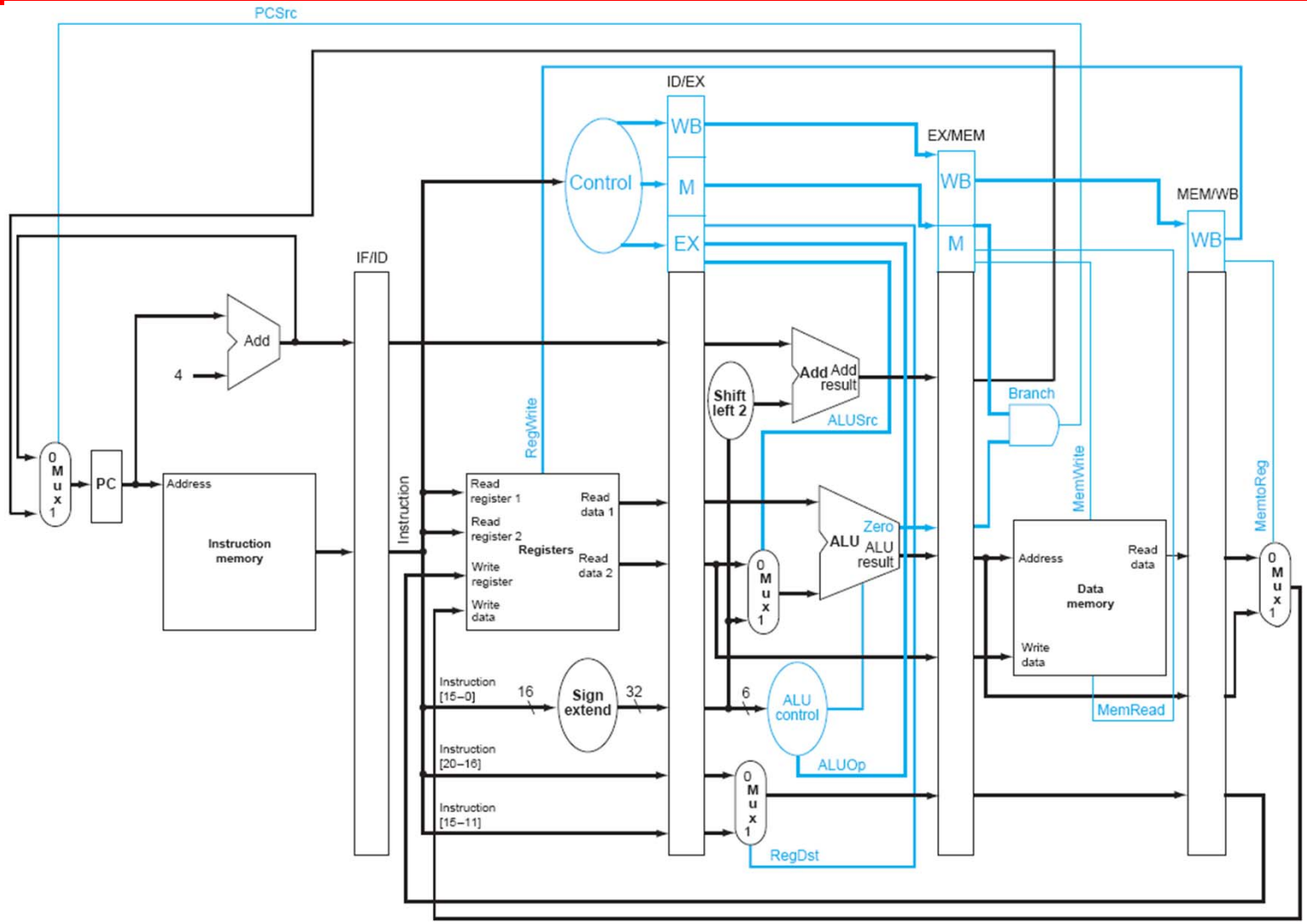
- Primary memory
  - Implemented using silicon technology
  - Main memory (DRAM, ROM)
  - Cache memory (SRAM)

- Secondary memory
  - Implemented using magnetic, optical, or silicon technologies
    - Hard disks
    - Magnetic tapes
    - DVDs
    - Flash drives/memory sticks

Processor

- The “brains” of the computer
  - Microprocessors
  - Microcontrollers
  - Special-purpose processors (e.g. GPUs, NPUs, DSPs)
- Datapath
  - The part of the processor responsible for decoding and executing instructions
  - Consists of arithmetic and logic units and temporary storage elements called *registers*
- Control
  - The part of the processor that coordinates the fetching and execution of instructions
  - Also coordinates the operation of input, output, memory, and arithmetic/logic units

MIPS R2000 Datapath



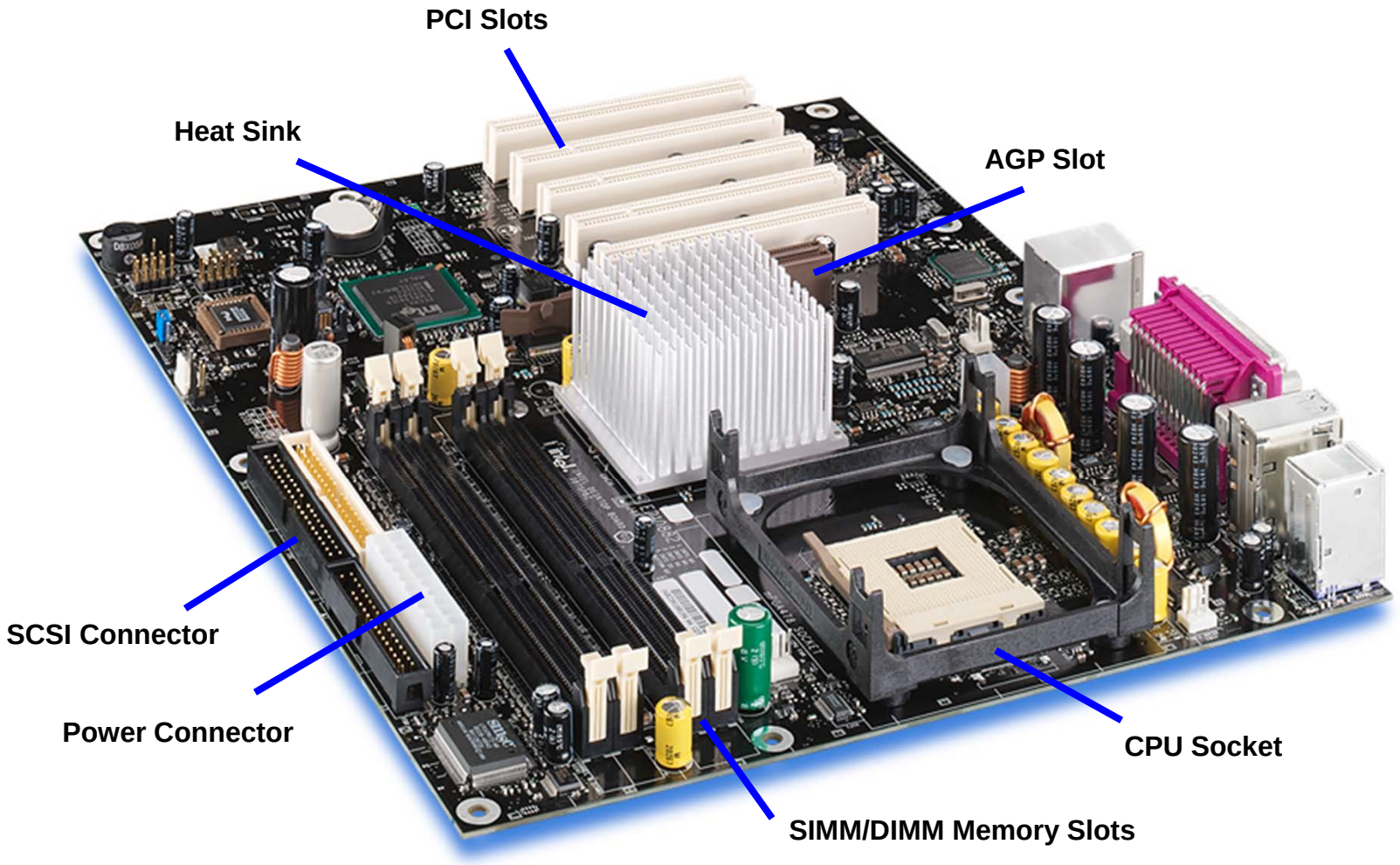
Instruction Set Architecture

- The interface between the processor hardware and the lowest level software
  - Includes all the information needed to write a *machine language* program (e.g. instructions, registers, support for memory and I/O access, etc…)
  - **`add $t0,$s0,$s1`** **=** 00000010001100100100000000100000

assembly code

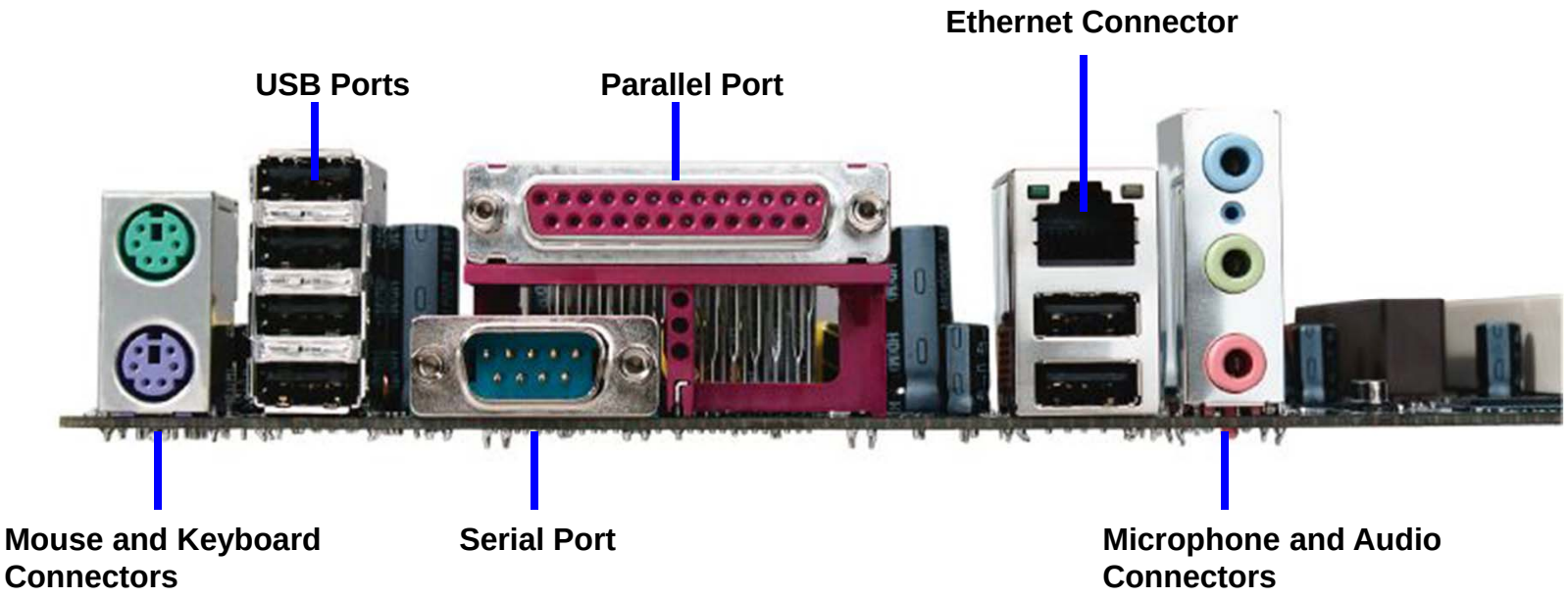
binary (machine) code

Inside the Personal Computer (PC)

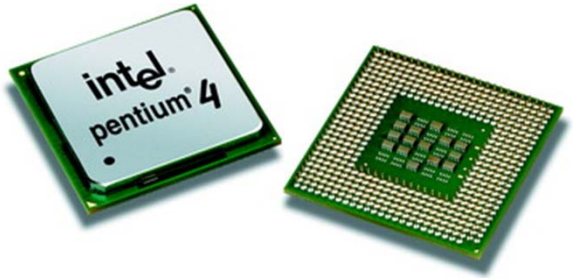


Intel D875PBZ motherboard. Picture courtesy of Intel Corporation.

The PC Motherboard – Back View



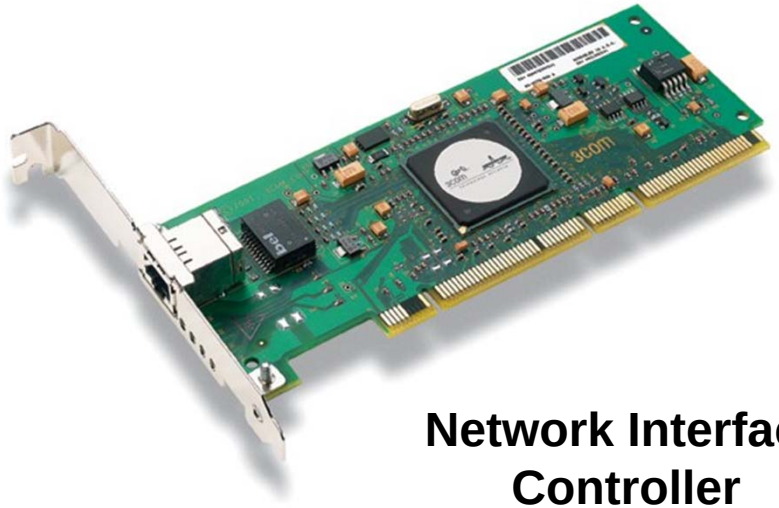
PC Hardware



Processor

Graphics Card

DRAM (Memory)



Network Interface Controller

PC Hardware (2)

Hard Disk Drive

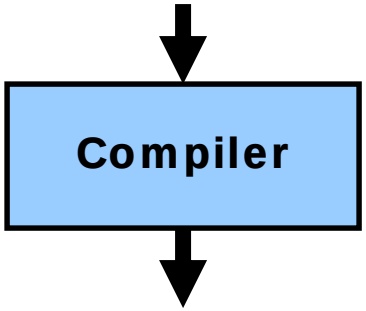
DVD/CD-ROM Drives

Computer = Hardware + Software

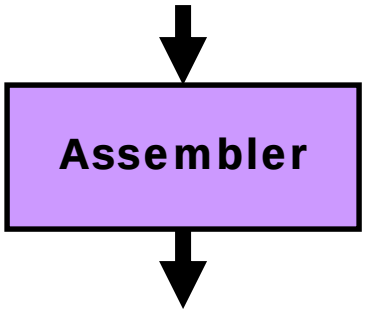
- Computers rely on specialized software programs to manage the hardware and simplify the task of programming the computer.
  - Operating systems
  - Compilers
  - Assemblers
  - Linkers
  - Loaders

From C++ to Machine Code

```
a = b + 5;
c = d * R[7];
```




```
ADDI    $R1,$R2,5
MUL     $R3,$R1,28($R4)
```




```
01101001101110101110101110001101
11101011101011100011010110100110
```

Hardware Courses at ECE

ECE/CCE Core

- EECE 320 Digital Systems Design
- EECE 321 Computer Organization
- EECE 321L Computer Organization Lab

Hardware Focus Area

- EECE 421 Computer Architecture
- EECE 422 Parallel Computer Architecture and Programming
- EECE 425 Embedded Systems Design

Technical Electives

- EECE 621 Advanced Computer Architecture
- EECE 623 Reconfigurable Computing
- EECE 624 Digital Systems Testing

Career Opportunities

- **Computer Architect**
  - Instruction set and microarchitecture design
  - Special purpose processor design (graphics processors, network processors, DSPs)
- **Hardware Engineer**
  - System- and board-level hardware designs using microcontrollers, CPLDs, or FPGAs
  - Integrated circuit component design
- **CAD Engineer**
  - Developing tools for logic synthesis, technology mapping, and placement and routing
- **System Software Developer**
  - Operating systems
  - Compilers
  - Software drivers and libraries for specialized processors (e.g. graphics, networking, etc…)