Computer Architecture
1. Introduction

Lecturer: A.Prof.Dr. Hoàng Xuân Dậu

Email: dauhx@ptit.edu.vn

Faculty of Information Security

Posts & Telecommunications
Institute of Technology

Main Textbooks & References

- Hoàng Xuân Dậu, Bài giảng Kiến trúc máy tính, Học viện CNBCVT, 2010.
- Stallings W., Computer Organization and Architecture: Designing for Performance, 10th Edition, Prentice – Hall, 2016.
- Hennesy J.L. and Patterson D.A., Computer Architecture. A Quantitative Approach, Morgan Kaufmann, 6th Edition, 2017
- Trần Quang Vinh, *Cấu trúc máy vi tính*, Nhà xuất bản Giáo dục, 1999.
- Hồ Khánh Lâm, *Kỹ thuật vi xử lý*, Nhà xuất bản Bưu điện, 2005.

Subject Assessment

- Four component marks:
  - Class attendants: 10%
  - Exams: 10%
  - Group minor project: 20%
  - Final exam: 60%

Course Topics

1. Computer architecture – Introduction
2. The Central Processing Unit – CPU
3. Computer Instruction Sets
4. Introduction to CPU pipelines
5. Memory System
6. External Storage
7. Computer Peripherals
8. Computer Buses
9. Advanced Architectures and Technologies
10. Introduction to Cloud Computing

Computer architecture – Introduction

- Computer architecture versus organization
- Computer structure and functions of components
- Brief history of computers
- Von - Neumann architecture
- Harvard architecture
- Modern computer organization
- Data representation in computers

Architecture vs. organization

- Architecture vs. organization are two major concepts of computer engineering.
  - Computer organization is the science that studies computer components and their working methodologies.
  - Computer architecture is the science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals:
    - Performance: faster is better
    - Functionality: more functions is better
    - Costs: cheaper is better

Architecture vs. organization (cont.)

- Three components of computer architecture:
  - Instruction set architecture, or ISA, is the abstract image of a computing system at the machine language (or assembly language) level, including:
    - The instruction set;
    - Memory addressing modes
    - Processor registers
    - Address and data formats.

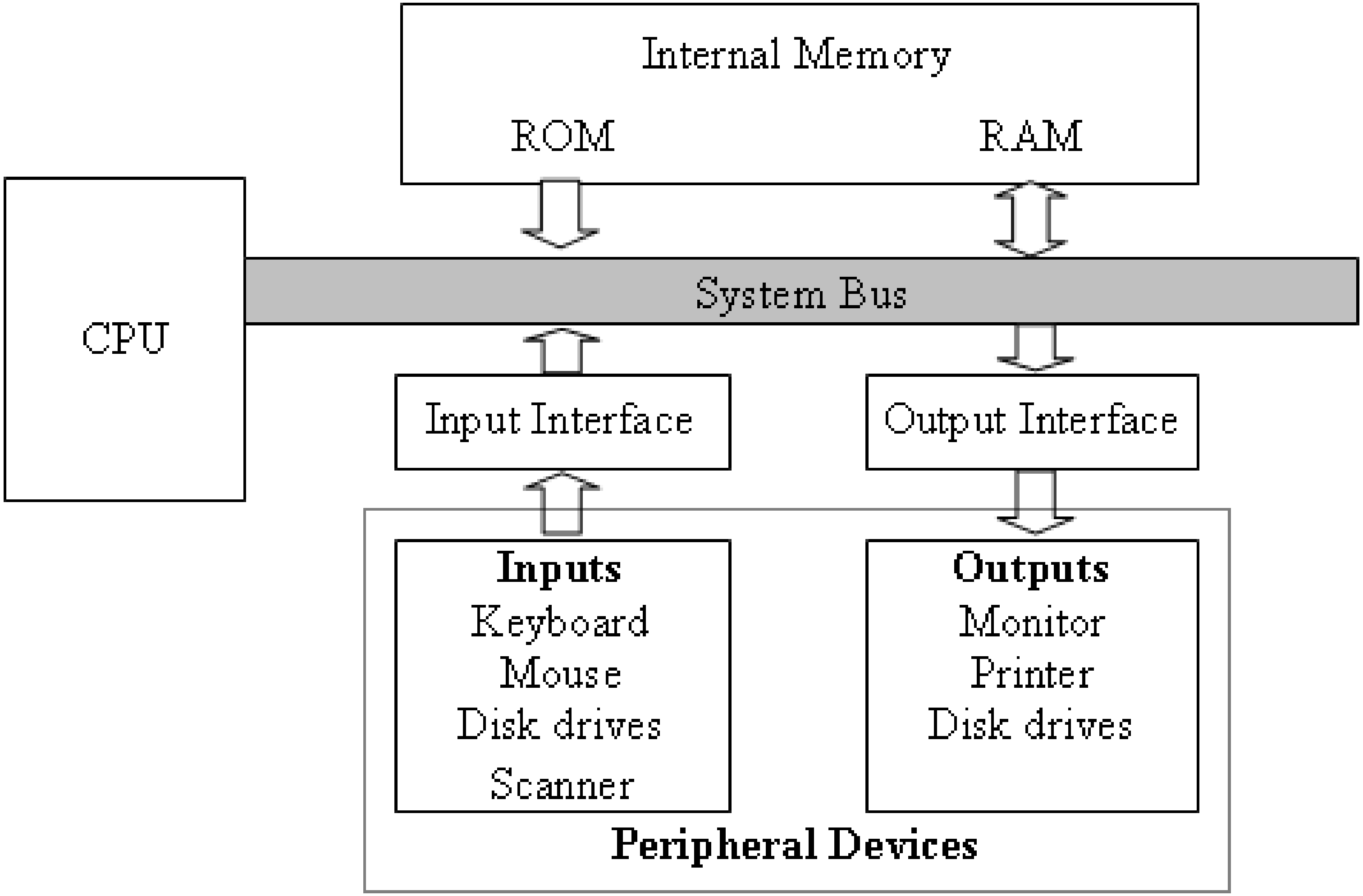
Architecture vs. organization (cont.)

- Microarchitecture, also known as computer organization is a lower level description of the system that involves:
  - How the hardware parts are interconnected; and
  - How hardware parts interoperate in order to implement the instruction set architecture.

Architecture vs. organization (cont.)

- System Design which includes all of the other hardware components within a computing system such as:
  - System interconnects such as computer buses and switches
  - Memory controllers and hierarchies
  - CPU off-load mechanisms such as direct memory access
  - Issues like multi-processing.

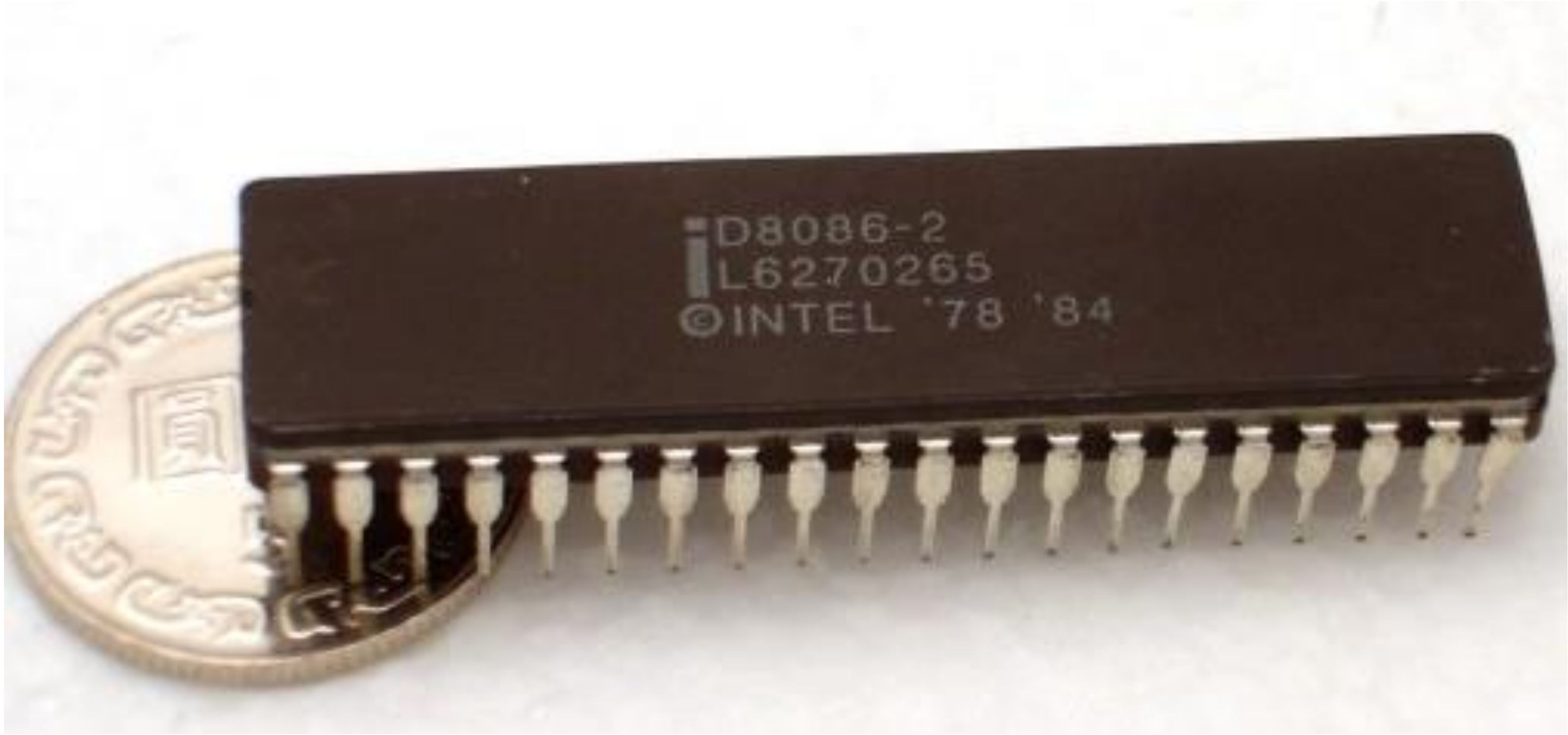
Computer Functional Block Diagram



Computer major components - CPU

- Central Processing Unit (CPU):
  - Fetch instructions from memory, decode and execute instructions;
  - CPU includes:
    - Control Unit (CU)
    - Arithmetic and Logic Unit (ALU)
    - Registers

Computer major components - CPU



Intel 8086 microprocessor (1978)

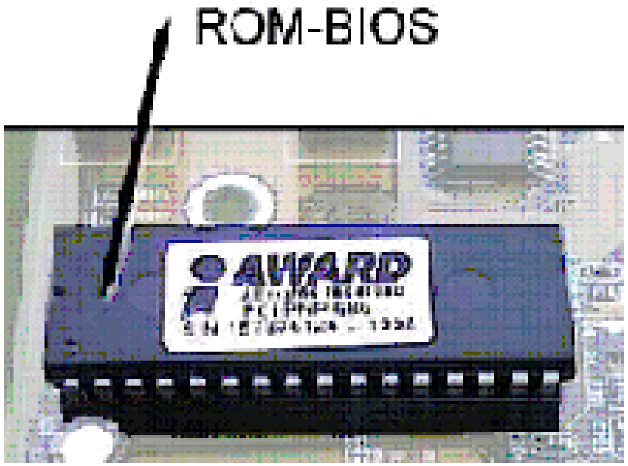
Computer major components - CPU

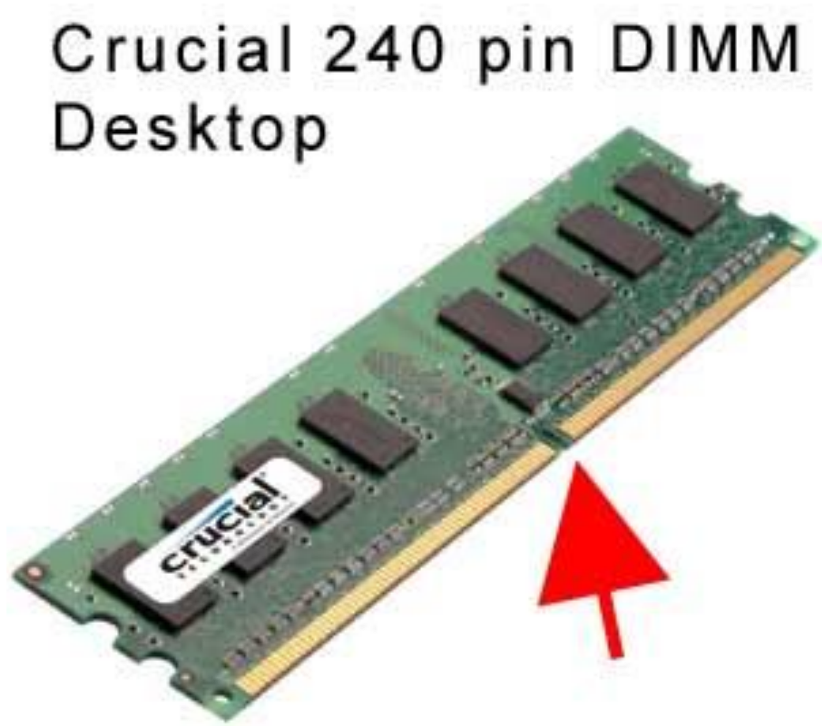
Intel core i7 microprocessor (2008)

Computer major components

- Internal memory
  - Store instructions and data for CPU to process
  - Internal memory includes:
    - ROM – Read Only Memory, stores instructions and data of the system;
    - RAM – Random Access Memory, stores instructions and data of the system and users.

Computer major components



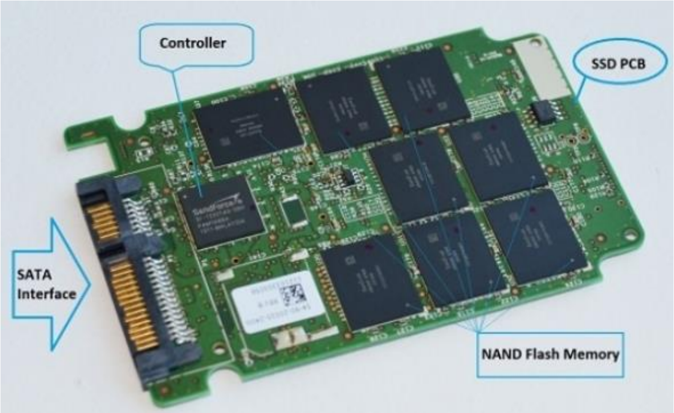


ROM and RAM memory

Computer major components

- Peripheral devices
  - Input devices: input data and control
    - Keyboard
    - Mice
    - Disk drives
    - Scanner
  - Output devices: output data
    - Monitor/screen
    - Printer
    - Plotter
    - Disk drives

Computer major components

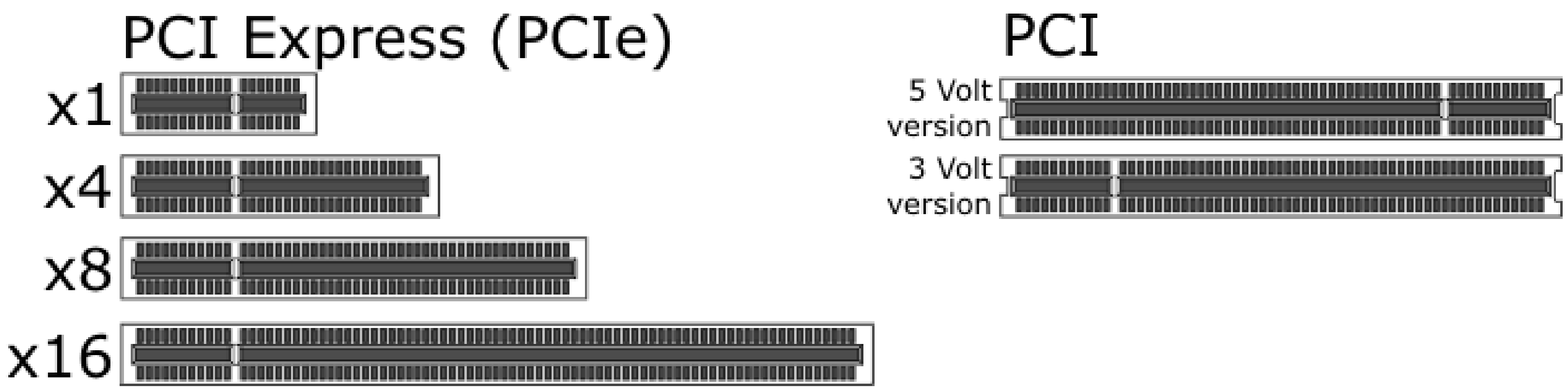


HDD and SSD

Computer major components

- System buses
  - Are sets of wires to connect CPU with other components of a computer.
  - Includes 3 buses:
    - Address bus (also called A Bus)
    - Data bus (also called D Bus)
    - Control bus (also called C Bus)

Computer major components - PCI

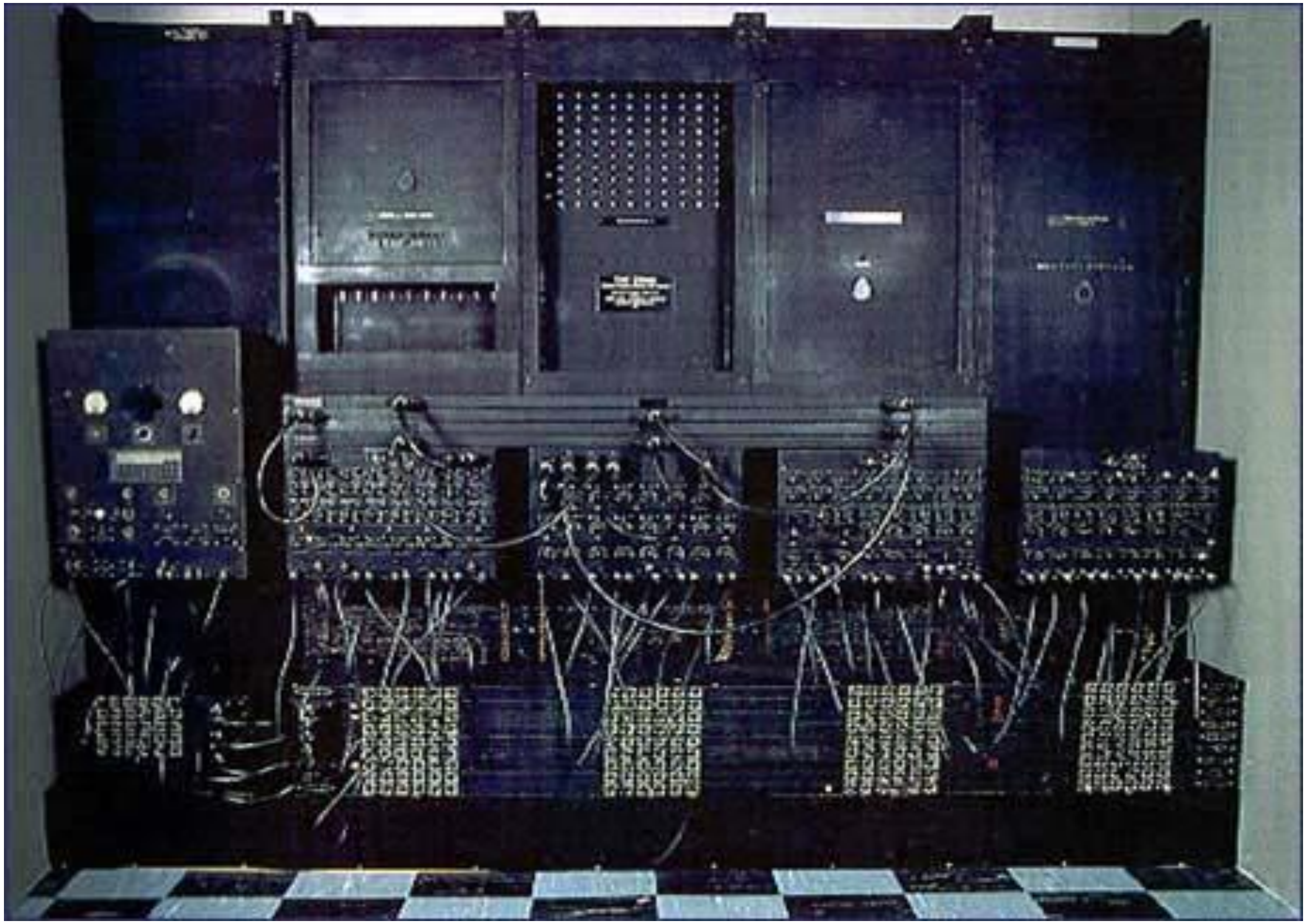


PCIe bus/ PCI bus

History of computers

- History of computers can be divided into 5 generations based on the development of electronic circuits.
- 1st generation (1944-1959):
  - Use vacuum tubes as main electronic elements
  - Use magnetic tape as input/output devices
  - Density: ~ 1000 elements per cubic foot (= 30.48 cm)
  - Samples: ENIAC - Electronic Numerical Integrator and Computer, costs 500,000 USD.

History of computers - ENIAC



History of computers – 2nd generation

- 2nd generation (1960-1964):
  - Use transistors
  - Density: ~ 100,000 elements per cubic foot
  - Samples: UNIVAC 1107, UNIVAC III, IBM 7070, 7080, 7090, 1400 series, 1600 series.
  - The first UNIVAC was delivered to the United States Census Bureau on March 31, 1951 and was dedicated on June 14th that year. Originally priced at US$159,000, the UNIVAC I rose in price until they were between $1,250,000 and $1,500,000.

History of computers – UNIVAC

History of computers – 3^{rd} generation

- 3^{rd} generation (1964-1975):
  - Use Integrated Circuit (IC)
  - Density: ~ 10 million elements per cubic foot
  - Samples: UNIVAC 9000 series, IBM System/360, System 3, System 7

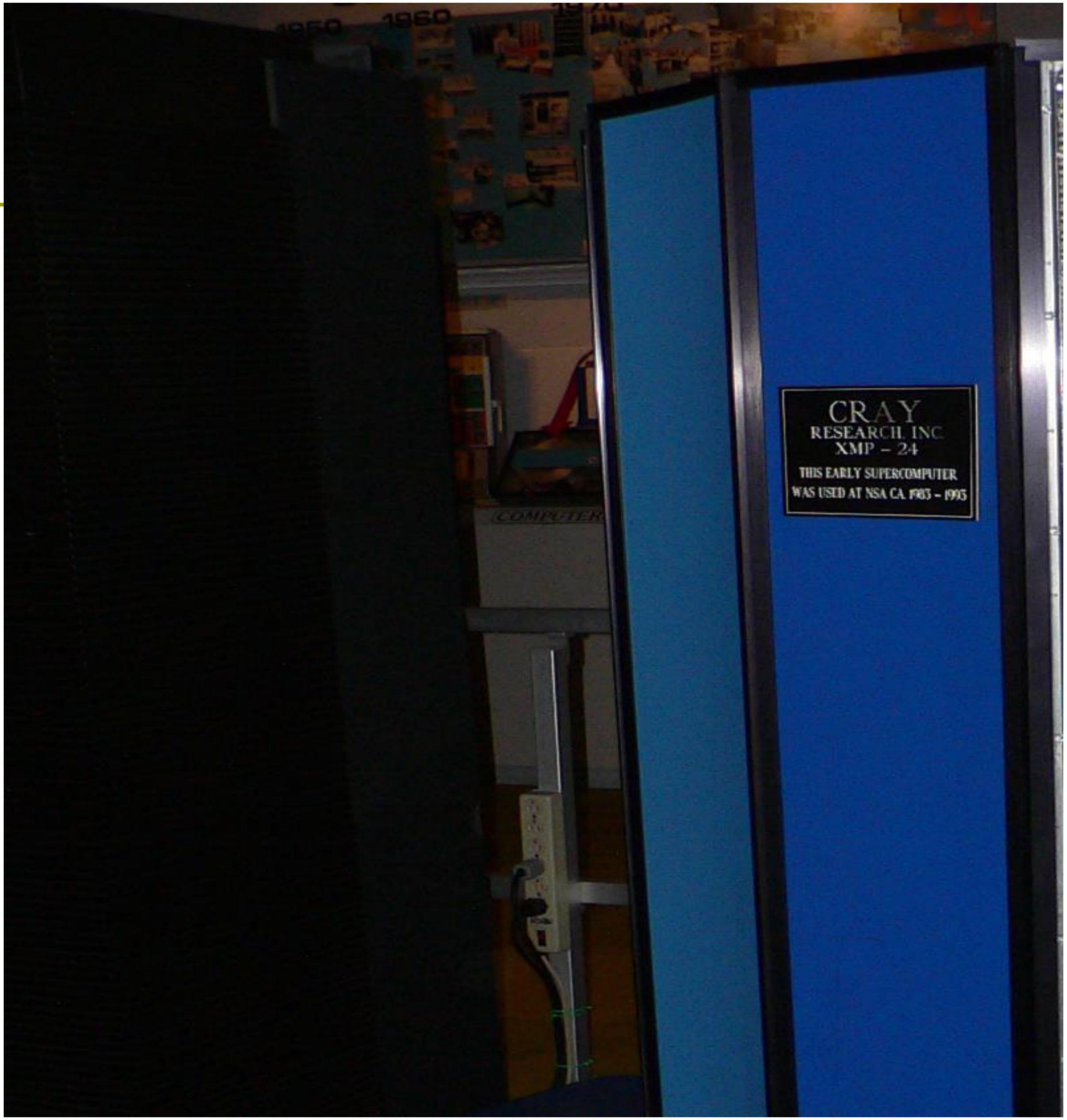
History of computers – UNIVAC 9400

History of computers – 4^{th} generation

- 4^{th} generation (1975-1989):
  - Use LSI – Large Scale Integrated Circuit
  - Density: ~ 1 billion elements per cubic foot
  - Samples: IBM System 3090, IBM RISC 6000, IBM RT, Cray 2 XMP

History of computers

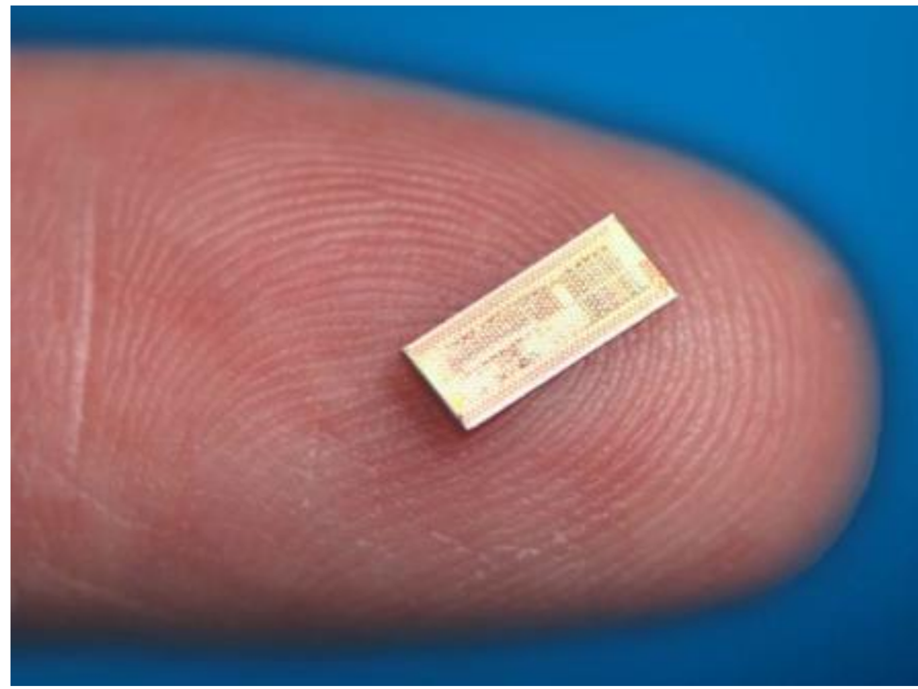
Cray 2
XMP



History of computers – 5^{th} generation

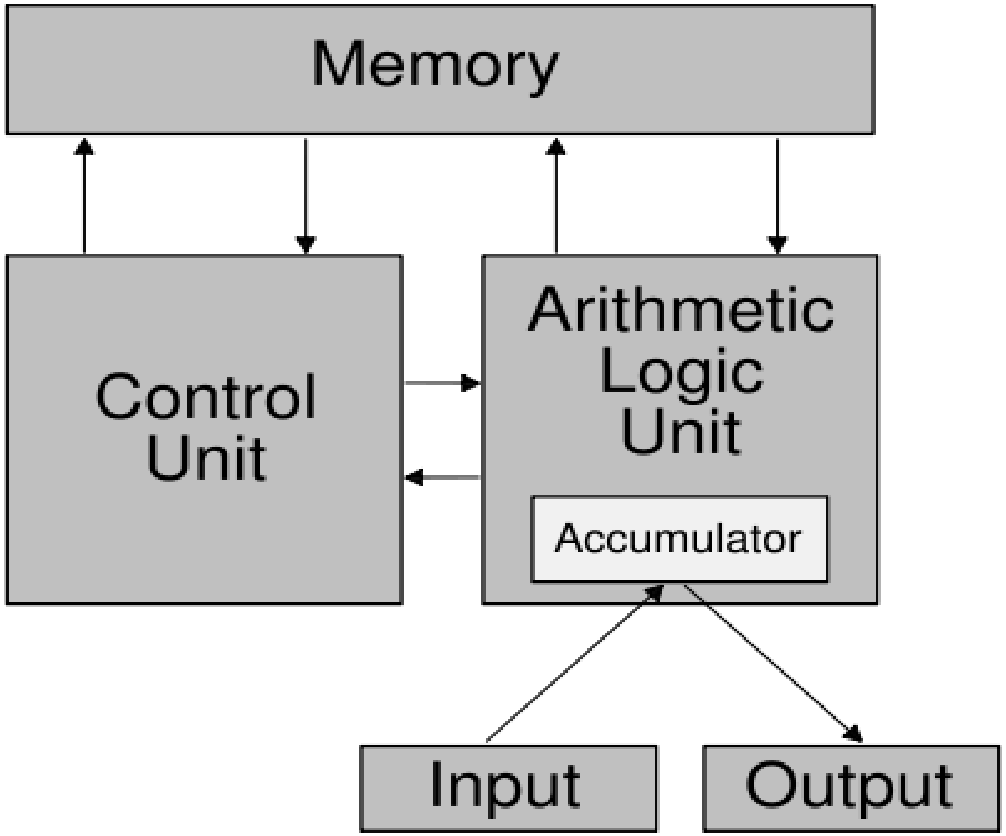
- 5^{th} generation (1990-now):
  - Use VLSI – Very Large Scale Integrated Circuit
  - Density: use 0.18μm – 0.032μm
  - Samples: Pentium II, III, IV, M, D, Core Duo, Core 2 Duo, Core Quad, Core i3, Core i5, Core i7...
  - Support parallel processing
  - Very high performance
  - Integrate voice and image processing

History of computers – 5th generation

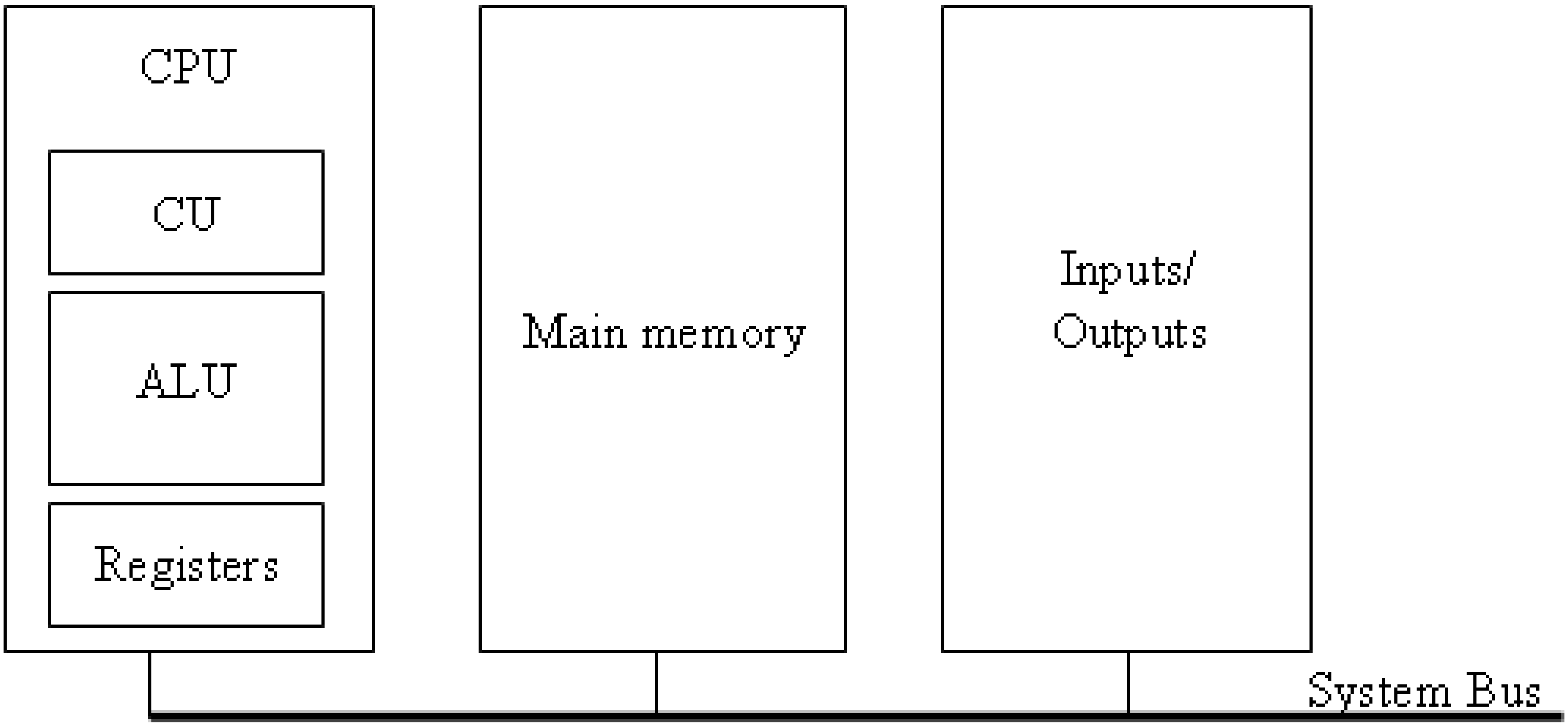


Intel Core 2 Extreme and Atom microprocessors

Von-Neumann old architecture



Von-Neumann modern architecture



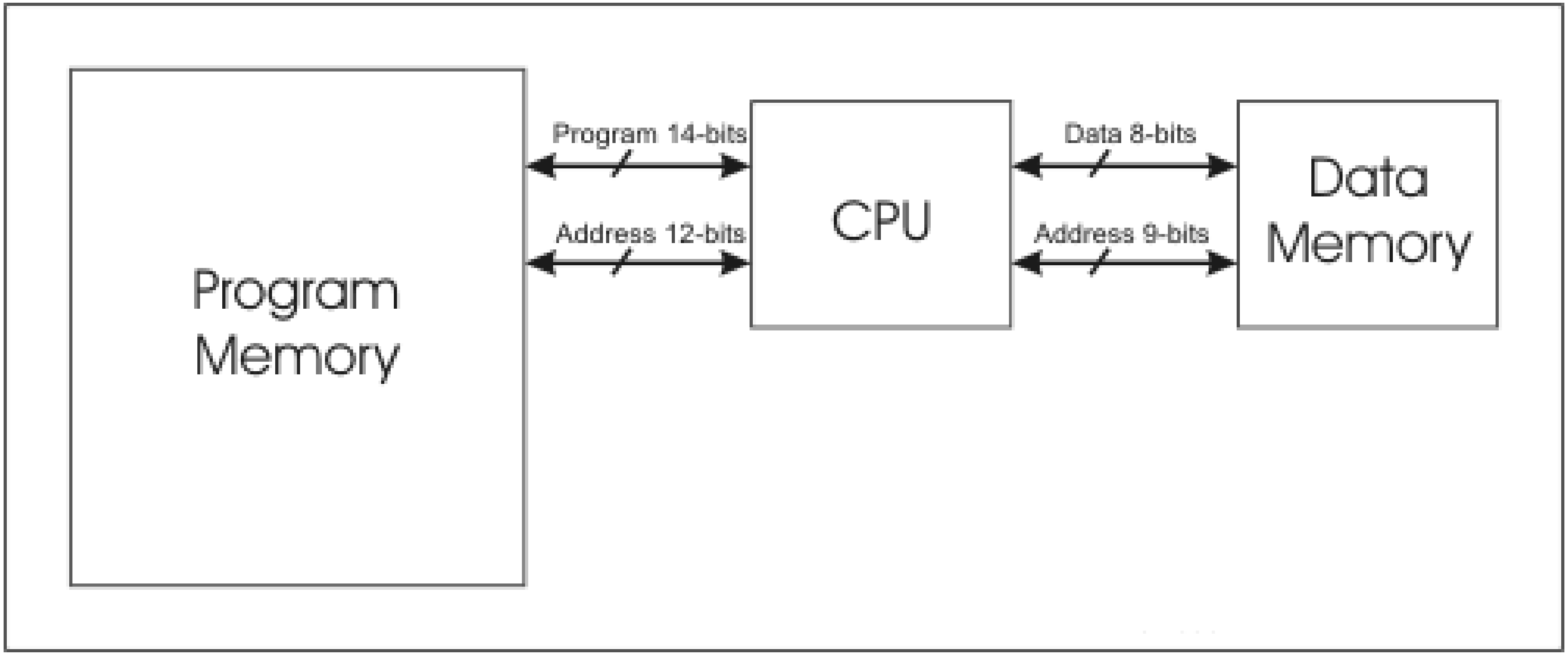
Characteristics of von-Neumann architecture

- Von-Neumann architecture was introduced by John von-Neumann in 1945.
- Von-Neumann architecture computers are based on three main concepts:
  - Data and instructions are stored in a shared read/write memory;
  - Memory is addressed based on segments and not depending on what it stores;
  - Instructions of a program are executed one after another.

Characteristics of von-Neumann architecture (cont.)

- Instruction execution is divided into 3 main stages:
  - CPU fetches instruction from memory
  - CPU decodes and executes instruction; if instruction requires data, read data from memory
  - CPU writes results into memory if any

Harvard architecture



Harvard architecture (cont.)

- Memory is divided into 2 parts:
  - Program memory
  - Data memory
- CPU uses 2 system buses to communicate to memory:
  - One A bus for program memory and another A bus for data memory;
  - One D bus for program memory and another D bus for data memory.

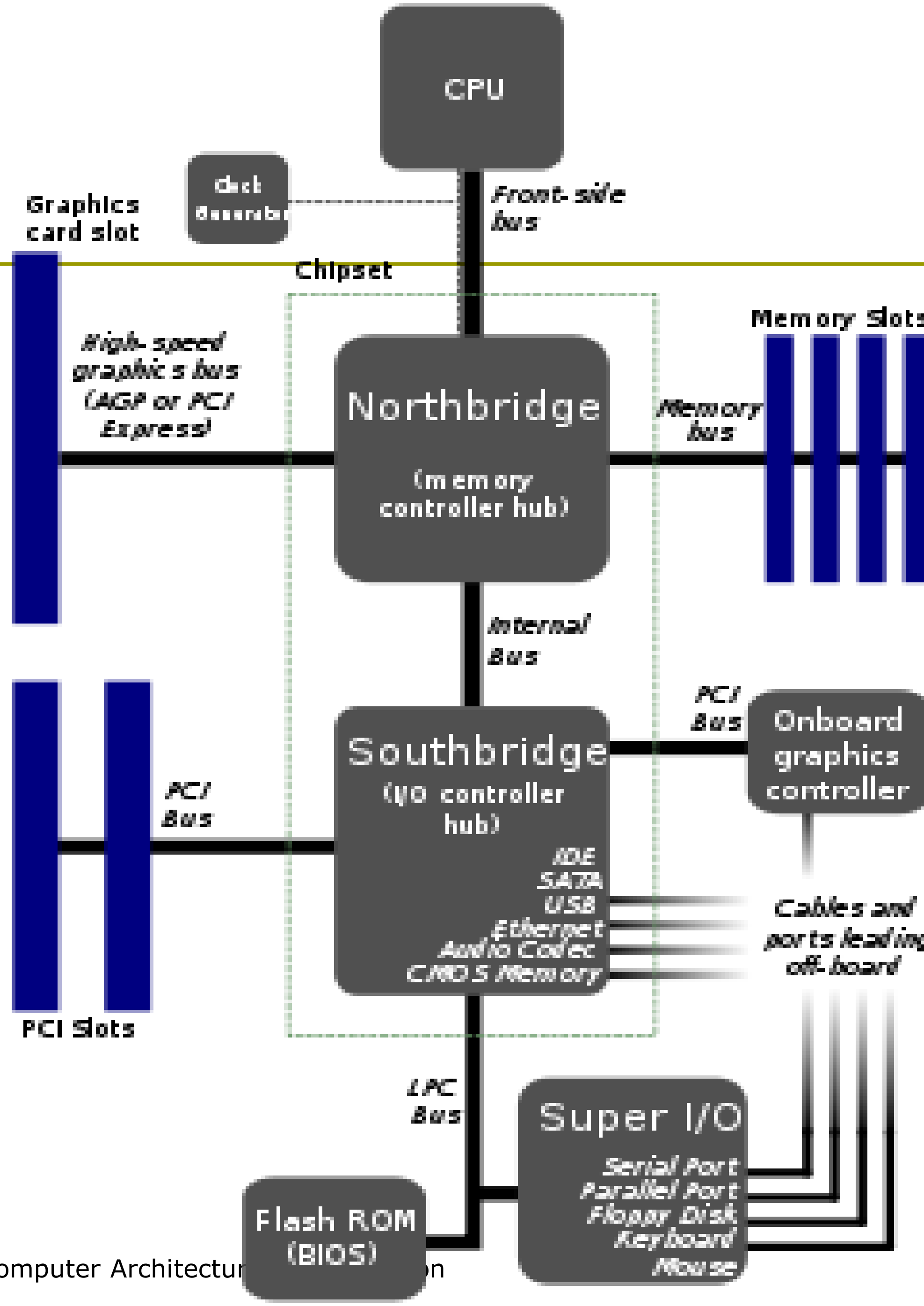
Harvard architecture (cont.)

- Faster because bus bandwidth is larger.
- Supports more memory read/write accesses at the same time → reduce memory access conflict.

	IF	ID	EX	MEM	WB				
↓ *i*		IF	ID	EX	MEM	WB			
t →			IF	ID	EX	MEM	WB		
				IF	ID	EX	MEM	WB	
					IF	ID	EX	MEM	WB

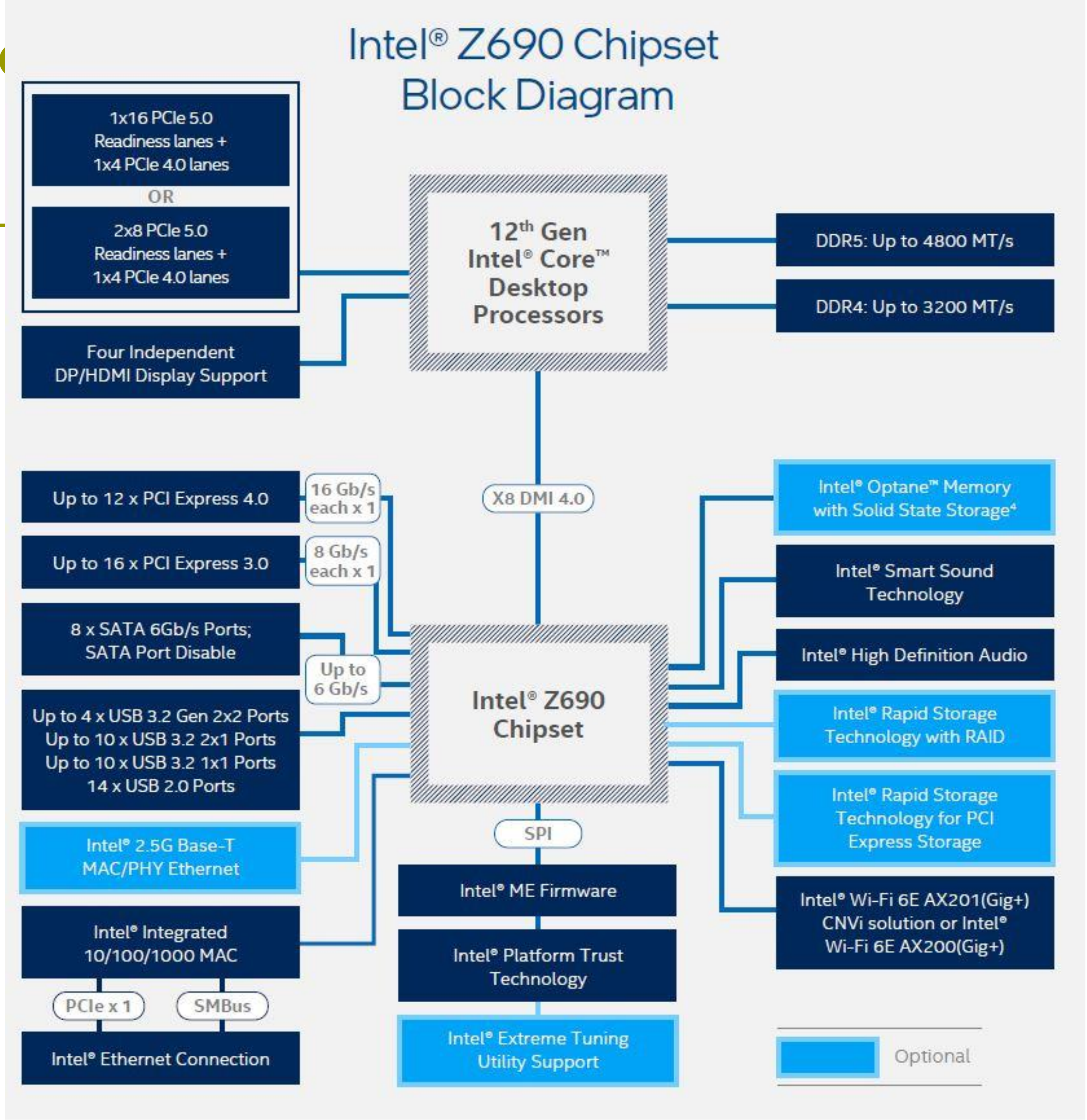
Modern computer organization

- Computer systems with Northbridge and Southbridge chipsets



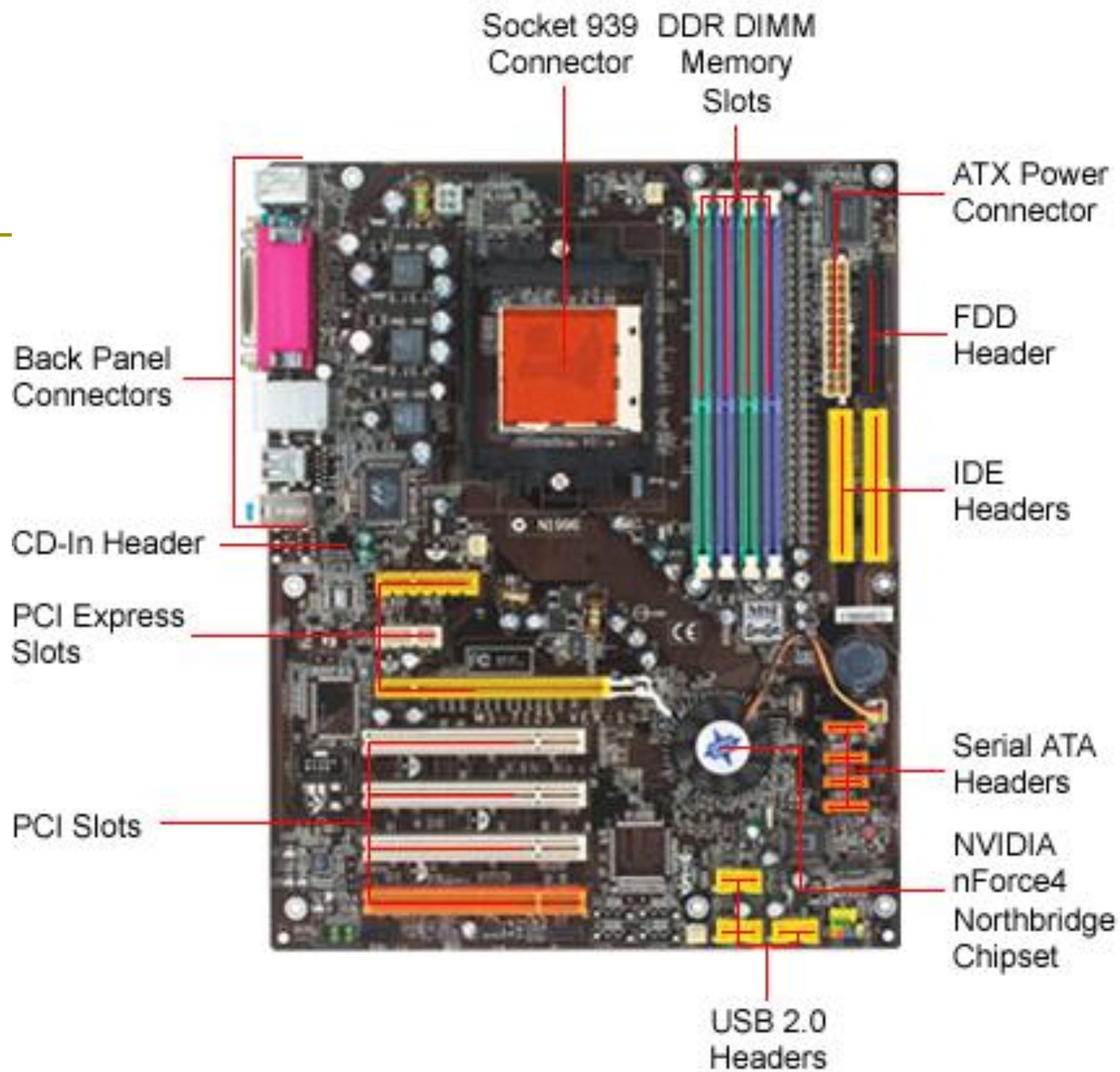
Modern c

Computer systems with only Southbridge chipsets (CPU has integrated MMU and graphic chip)

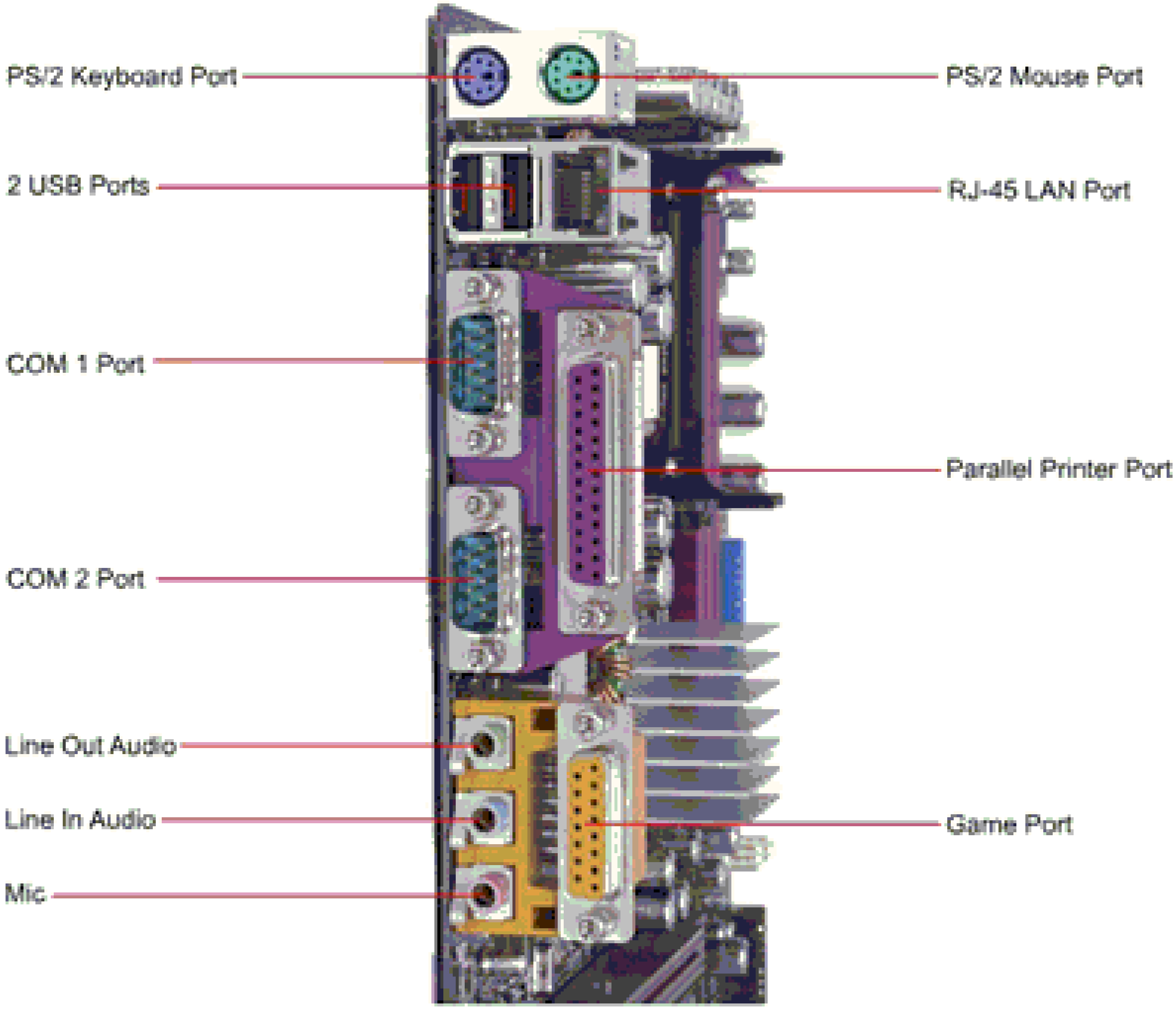


Sample desktop mother board

Sample desktop mother board



Sample desktop mother board



Sample laptop mother board

Data representation in computers

- In computer systems, binary numbering system is used to represent data.
- Binary numbering system uses just 2 digits 0 and 1; 0 represents the logic value of False and 1 represents the logic value of True.
- The hexadecimal numbering system is also used mainly for data presentation. It uses 16 digits: 0-9, A, B, C, D, E, F.

Decimal numbering system

- Decimal numbering system is base 10 and uses 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- A number in decimal numbering system can be represented using a polynomial as:

  $a_n a_{n-1 \dots} a_1 = a_n * 10^{n-1} a_{n-1} * 10^{n-2*} {}_{\dots *} a_1 * 10^0$

  For examples:

  $123 = 1*10^2 + 2 * 10^1 + 3*10^0 = 100+20+3$

  $123.456 \quad = 1*10^2 + 2*10^1 + 3*10^0 + 4*10^{-1} + 5*10^{-2} + 6*10^{-3}$

  $= 100 + 20 + 3 + 0.4 + 0.05 + 0.006$

Binary numbering system

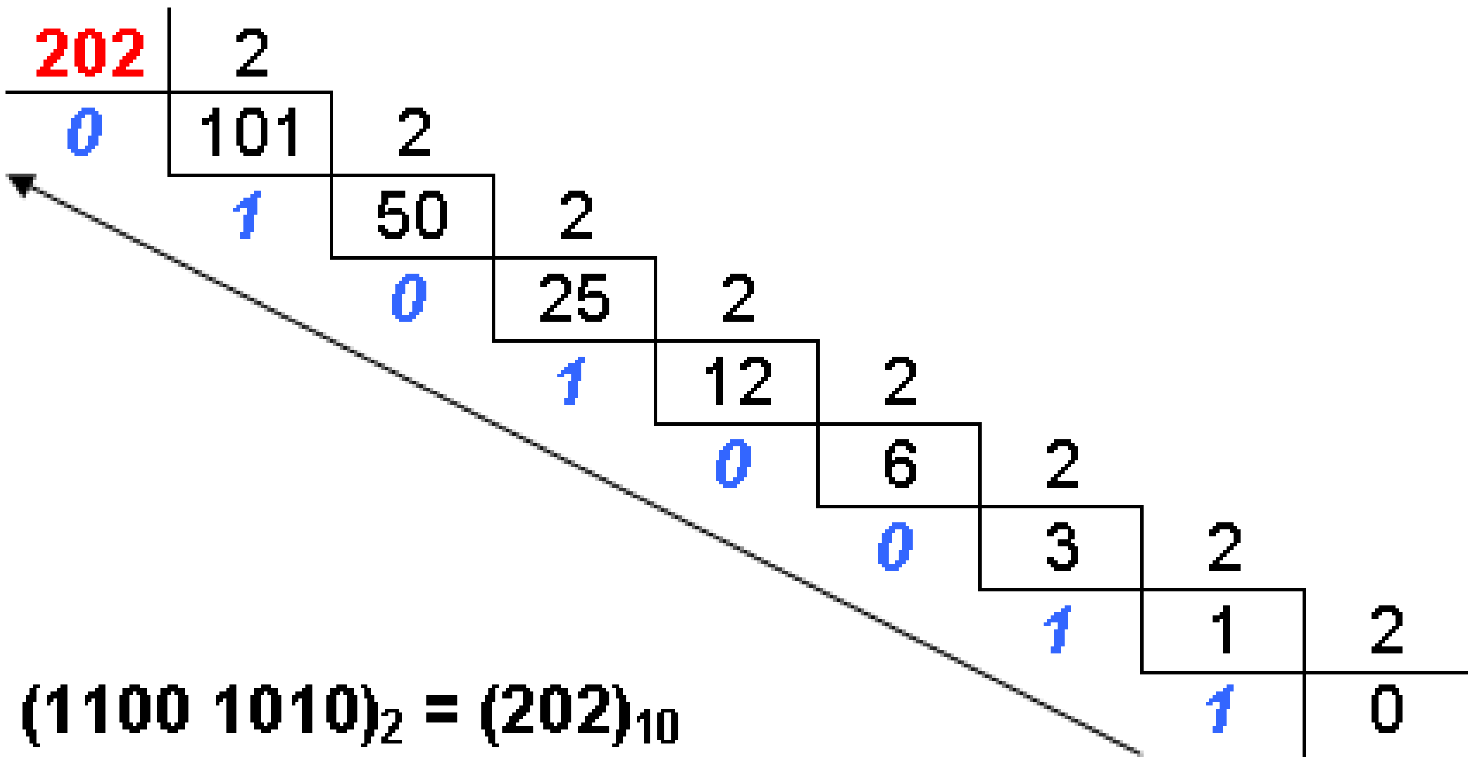
- Binary numbering system uses base 2 with only 2 digits: 0 and 1.
- We can also use a polynomial to represent a binary number:

$(11001010)_2$

$= 1*2^7 + 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0$

$= 128 + 64 + 8 + 2 = (202)_{10}$

Convert decimal to binary numbers



$(1100\ 1010)_2 = (202)_{10}$

Hexadecimal numbering system

- Hexadecimal numbering system uses base 16 with 16 digits: 0-9, A, B, C, D, E, F.
- Each digit in hexadecimal numbering system is represented by 4 digits in binary numbering system.

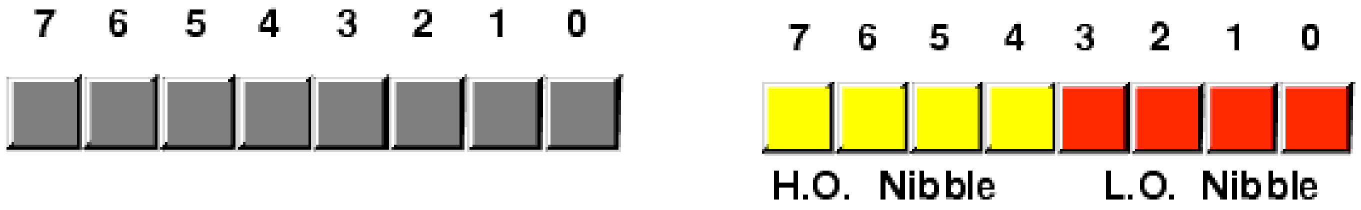
Hexa	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Data Organization

- Bits:
  - Bit is the smallest data unit.
  - One bit can only store 2 values: 0 or 1, true or false.
- Nibbles:
  - A nibble is a group of 4 bits
  - One nibble can store up to 16 values, from $(0000)_2$ to $(1111)_2$, or one hexadecimal number.

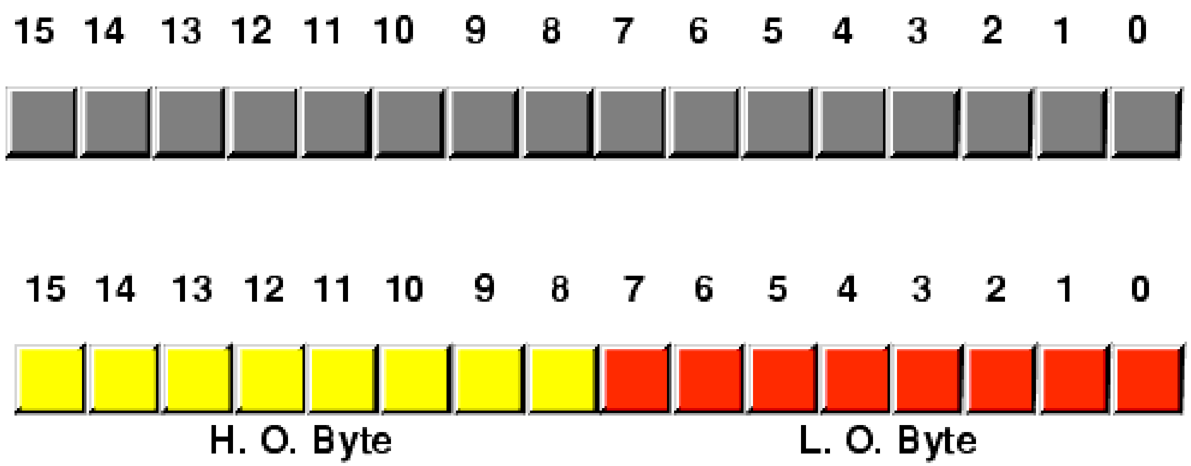
Data Organization

- Bytes:
  - A byte is a group of 8 bits or two nibbles.
  - One byte can store up to 256 values, from $(0000\ 0000)_2$ to $(1111\ 1111)_2$, or from $(00)_{16}$ to $(FF)_{16}$.



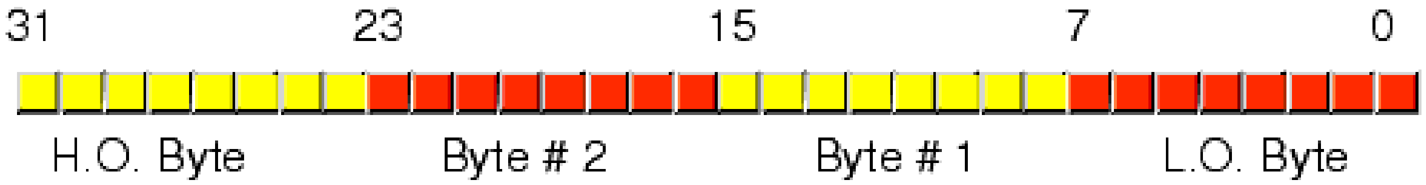
Data Organization

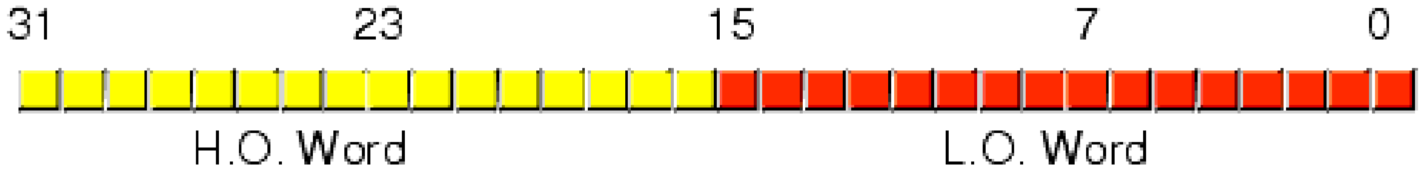
- Words:
  - A word is a group of 16 bits, or 2 bytes
  - One word can store up to 2^{16} (65536) values, from $(0000)_{16}$ to $(FFFF)_{16}$.



Data Organization

- Double words:
  - A double word is a group of 32 bits, or 4 bytes, or 2 words
  - One double word can store up to 2^{32} values, from $(0000\ 0000)_{16}$ to $(FFFF\ FFFF)_{16}$.





Signed and Unsigned Numbers

- In binary numbering system, the left most bit is used to represent the sign of the signed numbers.
  - The left most bit is 0 → positive number
  - The left most bit is 1 → negative number
- For examples, we use 4 bits to represent numbers:
  - 0011, 0111, 0101 are positive numbers
  - 1011, 1111, 1101 are negative numbers
- For unsigned numbers, all bits are used to store the value.

Signed and Unsigned Numbers

- Representation ranges: n bits can represent:
  - Signed numbers: from -2^{n-1} to $+2^{n-1}$
    - 8 bits: from -128 to +128
    - 16 bits: from -32768 to +32768
    - 32 bits: from -2,147,483,648 to +2,147,483,648
  - Unsigned numbers: from 0 to 2^n
    - 8 bits: from 0 to 256
    - 16 bits: from 0 to 65536
    - 32 bits: from 0 to 4,294,967,296

ASCII Table

- ASCII or American Standard Code for Information Interchange is a character-encoding scheme based on the ordering of the English alphabet.
- Use 8 bits to represent a character.
- ASCII includes definitions for 128 characters:
  - 33 are non-printing control characters
  - 94 are printable characters, and the space is considered an invisible graphic.
- Other positions (129-255) in ASCII are reserved for future use.

ASCII Table – Control chars

Binary	Oct	Dec	Hex	Abbr	PR[t 1]	CS[t 2]	CEC[t 3]	Description
000 0000	000	0	00	NUL	NUL	^@	\0	Null character
000 0001	001	1	01	SOH	SOH	^A		Start of Header
000 0010	002	2	02	STX	STX	^B		Start of Text
000 0011	003	3	03	ETX	ETX	^C		End of Text
000 0100	004	4	04	EOT	EOT	^D		End of Transmission
000 0101	005	5	05	ENQ	ENQ	^E		Enquiry
000 0110	006	6	06	ACK	ACK	^F		Acknowledgment
000 0111	007	7	07	BEL	BEL	^G	\a	Bell
000 1000	010	8	08	BS	BS	^H	\b	Backspace[t 4][t 5]
000 1001	011	9	09	HT	HT	^I	\t	Horizontal Tab
000 1010	012	10	0A	LF	LF	^J	\n	Line feed

ASCII Table – Printable chars

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	SP
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l