Ref: 'Online course on EE-390', KFUPM

Extra Handout No 2

By Dr Sheikh Sharif Iqbal

<u>Microprocessors</u>

Objective:

- To review the evolution of microprocessors

- To introduce basic architectures of microprocessors

- To review number system and data types

Slide 1: **<u>Microprocessors</u>**

- Microprocessor is an integrated circuit that stores and manipulates information as dictated by a set of instructions.

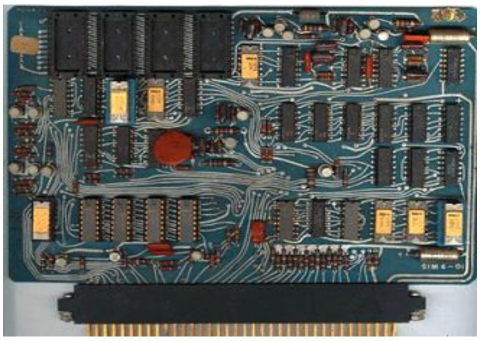
- In micro-computers, one or more microprocessors serve as the central processing unit (CPU), whereas microcontrollers coordinate all the functions of digital control devices. *(Remember microprocessors used for dedicated functions and not in reprogrammable general purpose computers are called microcontrollers of embedded-microprocessors)*

- The performance of microprocessors are expressed in terms of its Bandwidth (*number of bits processed in a single instruction*), Instruction set (*set of instructions that can be executed*) and Clock-speed (*number of executed-instructions per second*) *(Note that, bench marks are called MIPS (million instructions per second) and iCOMP index etc.)*

-The evolution of and basic characteristics of Intel microprocessors are tabulated in the next slide. *(Note that, since Intel started and pioneered the microprocessor industry, only Intel microprocessors are listed.)*

Slide 2: **Evolution and Basic Characteristics of Microprocessors**

- The 1st microprocessor, Intel-4004, had the following characteristics:



- o Introduced: Nov 1971
- o Clock Speed: 108 KHz
- o Int. register width: 8 bit
- o Bus width: 4 bits
- o No. of Transistors: 2300
- o Min. feature size: 10 micron
- o Main memory size: 640 Bytes

(In that same year, other companies like Texas Instruments' TMS 1000, and Garrett AiResearch's Central Air Data Computer are also released. These devices are mainly used to manupulate arighmetic data in calculators.)

- The characteristics of the following microprocessors are tabulated:

	8008	**8080**	**8086**
Introduced	1972	1974	1978
Clock Speed	200 KHz	2 MHz	5 – 10 MHz
Data Bus Width	8 bits	8 bits	16 bits
Register Width	8 bits	8 bits	16 bits
Number of Transistor	3,500 (10 microns)	6,000 (6 microns)	29,000 (3 microns)
Main Memory	16	64	1 MB
Brief Description	Data/character manipulation	10X performance of the 8008	10Xperformance of the 8080

- In 1972, Intel released the world's first 8-bit microprocessor called 8008. This processor was capable of manipulating arithmetic as well as charter data

- Two years later, Intel manufactured the first truly usable microprocessor CPU called 8080. Its performance was 10 times better than its predeccor.

- Intel 8086 was the 1st 16 bit microprocessor. This invention gave birth of the Intels 80x86 microprocessor family that rulled the computer world for considerable amount of time. .

- *The characteristics of the microprocessors are tabulate above.*

Slide 3: **<u>Evolution and Basic Characteristics (cont'd)</u>**

- Intel 8080 was widely accepted as multichip 8-bit microcomputers and used in Electronic instruments (printer, cash register etc.)

- But, Intels 1^{st} 16-bit microprocessor, the 8086, provided the required performance to construct a general purpose micro-computer. A year later, Intel released an 8-bit bus version of this processor, called 8088

- 80x86 processors have the ability to handle 8-bit, 16-bit and special purpose datas and had a powerful instruction set (like minicomputers)

- At this point of time, microprocessor design was split into two paths;

  - General purpose or reprogrammable microprocessors for computers (such as 80286, 80386, 80486, Pentium (80586))

  - Dedicated or special-purpose embedded microprocessors used in microcontrollers and digital control devices (80186…80386EX)

(Next slide briefly tabulates some of the basic characteristics of the advanced members of 80x86 microprocessor family.)

Slide 4: **<u>Basic Characteristics of recent members of 80x86 family</u>**

- Although this course (EE 390: Digital System Engineering) focuses on Intel 8086 and 8088 microprocessors, the basic characteristics of a few recent 80x86 family processors are presented for comparison.

- The tabulated characteristics shows that recent microprocessors have a clock speed in the range of GHz and have a wider internal registers and bus system, making them very fast and effecient

	80386-DX	**Pentium-II**	**Itanium (Merc.)**
Introduced	1985	1997	2001
Clock Speed	16-33 MHz	200-300 MHz	733-800 MHz
Data Bus Width	32 bits	64 bits	128 bits
Register Width	32 bits	32 bits	64 bits
Number of Transistor	275000 (1.5-1 microns)	7.5 million (0.28 microns)	≈40 million (180 nm)
Main mem.	4 GByte	64 GByte	--
Brief Description	1st to process 32 bit data	Dual bus and Intel MMX	Intel and HP Server

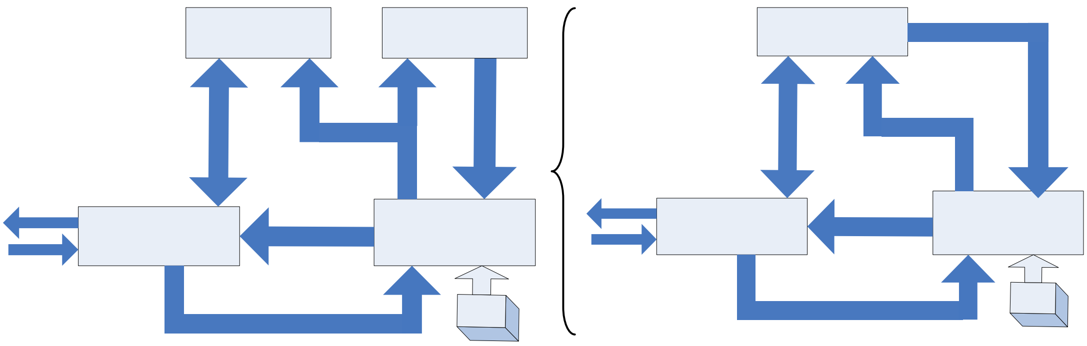
(The Table lists some of the basic characteristics of 1st 32-bit microprocessor (such as 80386DX) and one of the 64-bit (Pentium-pro) and 128 bit (Itanium) processors. Note the increase in computing power proviced by these later members of the 80x86 microprocessor family)

Slide 5: **<u>Basic Architectures of Microprocessors</u>**

- Microprocessor architectures represent that conceptual design and fundamental operation of the structure. Two basic architectures are:

  - Harvard architecture: This has physically different memory locations for Data and Instruction storage. Although early computers used this architecture, recently it is more popular in general purpose small microcontrollers (such as PIC controller) *(Colossus Mark-1 started this architecture.)*

  - Princeton architecture: Von-Neumann proposed single memory storage for both data and storage in this “storage program” architecture. This allowed computers to become more flexible.



Data Mermory

Instruction Memory

Harvard Architecture

Data

Address

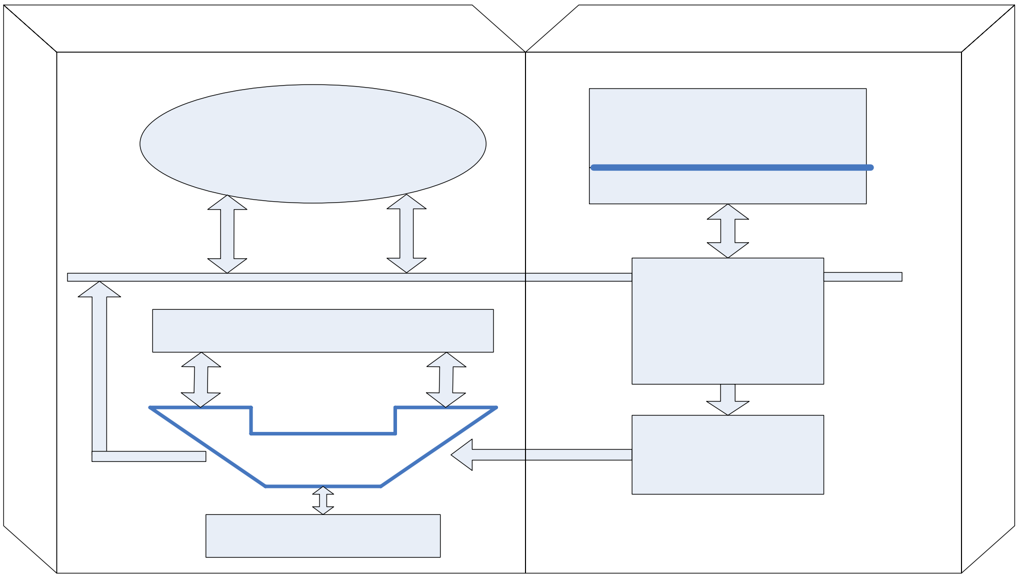
and

Control

Instruction

Slide 6: **<u>Micro-architecture of 8086 and 8088 Microprocessor</u>**

- Micro-architecture is the circuit building blocks that implements the software and hardware architecture and specifies information about the execution-unit, pipelining and the instruction-set of the processor.



(The internal architecture for 8086 and 8088 microprocessors are shown in above figure. Note the different components of the two processing units BIU and EU, which allow the microprocessor to employ parallel processing)

Slide 7: **Micro-architecture of 8086 and 8088 (cont'd)**

- The micro-architecture of 8086 and 8088 employ parallel processing using its two units: Execution-unit (EU) and Bus-interface-unit (BIU).

- The BIU uses multiplexed system bus to fetch instruction, read/write data operands for memory or for input/output peripherals. It is also responsible for instruction queuing and physical-address generation.

(Note the components of BIU are shown in the figure of previous slide)

- The pipelined architecture implemented by the instruction queue, allows 8086/8088 to pre-fetch up to 6/4 bytes of instructions.

- Execution unit is responsible to decoding and executing instructions. It accesses instructions form the instruction-queue and data from the general-purpose-registers or memory (with the help of BIU)

(The ALU of the execution unit (shown in previous figure) performs the arithmetic, logical and shift operation required by the instruction. During execution the EU tests and updates the status of the Flag registers based on the result of the execution.)

Slide 9: **Number System**

- Number systems are characterized by their base number. Thus, a number system with "base-n" should contain "n" different digits including zero, such as; 0, 1, 2, 3, 4, …. (n-1).

-Popular number systems are tabulated below:

Number system	Base	Digits or symbols
Binary	2	0, 1 $_{Bin}$
Octal	8	0, 1, 2, 3, 4, 5, 6, 7 $_{Oct}$
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9 $_{Dec}$
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F $_{Hex}$

- The correspondence between digits of different number systems (Decimal, Binary and Hexadecimal) are tabulated below:

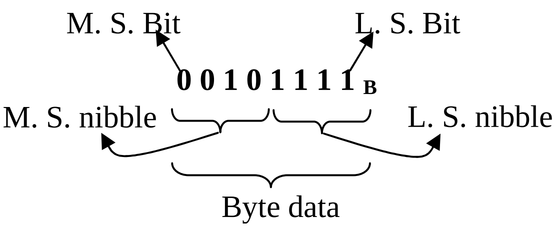
(Note that subscripts are used to identify the number system used for that particular number.)

Binary (base 2)	Decimal (base 10_D)	Hexadecimal (base 16_D or 10_H)
0000_B	0_D	0_H
0001_B	1_D	1_H
0010_B	2_D	2_H
0011_B	3_D	3_H
0100_B	4_D	4_H
0101_B	5_D	5_H
0110_B	6_D	6_H
0111_B	7_D	7_H
1000_B	8_D	8_H
1001_B	9_D	9_H
1010_B	10_D	A_H
1011_B	11_D	B_H
1100_B	12_D	C_H
1101_B	13_D	D_H
1110_B	14_D	E_H
1111_B	15_D	F_H
o 10000_B	16_D	10_H

- Thus, ‘14’ in decimal = ‘1110’ in binary = ‘E’ in hexadecimal system

- Microprocessors use Binary number system, where basic storage unit is a bit (‘0’ or ‘1’). Example of an 8-bit data is: 00101111_B.

- Binary data are also grouped to form other storage units, such as, Nibble (4-bits), Byte (8-bits). Word (16-bits) & Double-word (32-bits)



Addition	Substation
0 – 0 = 0	0 + 0 = 0
0 – 1 = -1 (with **borrow**)	0 + 1 = 1
1 – 0 = 1	1 + 0 = 1
1 – 1 = 0	1 + 1 = **1**0 (the **1** is carried)

- *Note in the left figure, the rightmost and leftmost binary digits are called least-significant-bit (LSB) and most-significant-bit (MSB), respectively.*
- *Also a nibble consist of four binary digits, with rightmost nibble called Least significant nibble and leftmost nibble called Most significant nibble. Note that a*
- *Byte of binary data consists of eight binary bits; where word consists of sixteen binary digits and double-word consist of thirty-two binary digits.*
- *In the right side of the slide, the basic rules of binary addition and subtraction are reproduced.*
- *Remember the* ***2^n rule*** *states that there will be 2^n different combination of a n-bit binary number.*

Slide 10: **Conversion between number systems**

Animate this slide

- Let use review the conversion techniques between Binary and Decimal number systems using the following examples:

Binary to Decimal conversion: Exp 1: $1010_B \rightarrow (1x2^3 + 0x2^2 + 1x2^1 + 0x2^0) = 10_D$

Example 2: $0110\ 0111_B \rightarrow (0x2^7 + 1x2^6 + 1x2^5 + 0x2^{4+}0x2^3 + 1x2^2 + 1x2^1 + 1x2^0) = 103_D$

- *So the example 1, the 4-bit binary number '1010' is converted to its equivalent decimal number of 10.*
- *In example 2, it is shown that 8-bit binary number of '01100111' is equal to '103' in decimal number system.*

Decimal to Binary conversion:

Convert Dec. → Bin.	Operation	Quotient	Remainder
Example 3: **Convert 6_D** into Binary number	$6_D \div 2_D$	3_D	0 → LSB
	$3_D \div 2_D$	1_D	1
	$1_D \div 2_D$	0_D	1 → MSB
	Thus, $6_D = 110_B$		
Example 4: **Convert 13_D** into Binary number	$13_D \div 2_D$	6_D	1 → LSB
	$6_D \div 2_D$	3_D	0
	$3_D \div 2_D$	1_D	1
	$1_D \div 2_D$	0_D	1 → MSB
	Thus, $13_D = 1101_B$		

- In example 3, a decimal number of 6 is converted into its equivalent binary number of 110. The meaning of MSB and LSB is explained in the previous slide.
- Example 4 shows that a decimal number '13' is equal to a binary number of '1101'.

Slide 11: **Conversion between number systems (cont'd)**

Binary to Hexadecimal conversion: For a binary number, less then 4-bit wide, this process is similar to binary to decimal conversion (shown in example 1). Otherwise, the binary number is grouped into nibbles (4-bits) starting from LSB and each nibble is individually converted into one hexadecimal digit. Padding can be used to insert a '0_B' in the left

Example 5: **In** "***101 1110 0111***$_B$" ; the least significant nibble is **$0111_B = 7_H$** and **$1110_B = E_H$** and **101_B** $= 5_H$; So equal hexadecimal number is: "***$5E7_H$***"

*(Since the left-most nibble (101_B in red) has three binary digits, we can always insert an extra "0_B" in the left (called padding) to make it a group of 4-bit. Thus is also due to the "**Rule of Fours**")*

Hexadecimal to Binary (via decimal) conversion: One way to achieve this is to substitute the equivalent 4-bit binary digits that represent each hexadecimal number (as listed in 2nd table of slide 8).

- In the 2nd method, the number to be converted is repeatedly divided by the base of hexadecimal number; 16_D or 10_H. Such as,

---

Convert Dec. ➔ Bin.	Operation	Quotient	Remainder	Resulted binary
Example 6: **Convert $E7_H$** into Binary number	**$E7_H$** $\div 10_H$	**E_H *or* 14_D**	**7_H** ➔ L.S.digit	**7_H** $= 0111_B$
	E_H $\div 10_H$	0_H	**E_H** ➔ M.S.digit	**E_H** $=$ **1110_B**
	Thus, **$E7_H$** = **1110** 0111_B			

Slide 12: **Data Types for 80x86 Microprocessors**

- The major data types processed by the microprocessors are: Numbers (Integers, Real) and Characters (ASCII)

Hyperlink: American Standard Code for Information Interchange

- Integers Numbers can be in the form of Unsigned or Signed,

  - Unsigned bytes range from 0 to $+255_D$ (*or* 0 to FF_H) and unsigned Words ranges from 0 to 65535_D (*or* 0 to $FFFF_H$).

  - Signed bytes ranges from -128_D to $+127_D$ and signed Words ranges from -32767_D to $+32768_D$. Signed integers use MSB as "Sign bit". Thus, MSB=0_B represents positive numbers and MSB=1_B represents negative numbers. Such as, $-3_H = 1101_B$

  - Typically, *2's complement* is used to express a negative number. Such as; -17_H => 2's comp. of $+17_H$ => $E9_H$

(2's complement is performed by inverting the logic levels (0's → 1's and 1's → 0's) of a binary number and then adding "1_B" to the inverted data.)

(Detail definition of Byte and Word is given in the following slide.)

- Real numbers may be rational or irrational numbers and often expressed by decimal or fractional representation.

Slide 13: **Data Types for 80x86 Microprocessors (cont'd)**

- 80x86 microprocessor family can process numbers that are coded as Binary Coded Decimals (BCD), as shown in given Table.

  - In BCD, group of 4-bits are used to store decimals (0_D to 9_D).
  - BCD data can be stored in either unpacked and packed from.

- For 80x86 microprocessors, characters are represented by two types of binary codes: ASCII and UNICODE:

  - ASCII is an 8-bit binary code, where 7-bits represent the character and the MSB (called parity-bit) is used for checking transmission errors. From ASCII table, codes are read as; '5_{ASCII}' => '35_H' ; 'A_{ASCII}' => '41_H' and *space* 'SP_{ASCII}' => '20_H'
  - UNICODE is a 16-bit code that provides a unique number for a character irrespective of the platform or program or language. ASCII printable codes can be represented by Unicode just by appending them to the left by "00_H", e.g., '$A_{Unicodel}$' => '0041_H'

Hyperlink 1: BCD representation of Decimal data

Decimal	BCD digit
0_D	0000
1_D	0001
2_D	0010
3_D	0011
4_D	0100
5_D	0101
6_D	0110
7_D	011
8_D	1000
9_D	1001

-This is the hyperlinked table for BCD representation of Decimal data. BCD is popular format when seven-segment display is used with digital systems

Hyperlink 2:
*In unpacked BCD, a single BCD digit is stored in one byte, where the four least significant bits contain the digit and the upper four bits are set to '0'. Such as; BCD digit of "00001001" stores one digit decimal number of "**9_D**"*

Hyperlink 3:
*In packed BCD, two BCD digits are stored in one byte, where each group of four bits store one digit. Such as; BCD digit of "0010 0101" stores two digit decimal number of "**25_D**"*

Hyperlink 4: American Standard Code for Information Interchange

Hyperlink 5: ASCII TABLE of codes for control and printable characters are tabulated below:

Character set of ASCII Code (read it as P_1P_0 ; such as "A"=41_H)

P_0 →

P_1 ↓

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>0</td><td>NUL</td><td>SOH</td><td>STX</td><td>ETX</td><td>EOT</td><td>ENQ</td><td>ACK</td><td>BEL</td><td>BS</td><td>HT</td><td>LF</td><td>VT</td><td>FF</td><td>CR</td><td>SO</td><td>SI</td></tr>
<tr><td>1</td><td>DLE</td><td>DC1</td><td>DC2</td><td>DC3</td><td>DC4</td><td>NAK</td><td>SYN</td><td>ETB</td><td>CAN</td><td>EM</td><td>SUB</td><td>ESC</td><td>FS</td><td>GS</td><td>RS</td><td>US</td></tr>
<tr><td>2</td><td>SP</td><td>!</td><td>"</td><td>#</td><td>$</td><td>%</td><td>&</td><td>'</td><td>(</td><td>)</td><td>*</td><td>+</td><td>,</td><td>-</td><td>.</td><td>/</td></tr>
<tr><td>3</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>:</td><td>;</td><td><</td><td>=</td><td>></td><td>?</td></tr>
<tr><td>4</td><td>@</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td></tr>
<tr><td>5</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td>[</td><td>\</td><td>]</td><td>^</td><td>_</td></tr>
<tr><td>6</td><td>`</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td><td>g</td><td>h</td><td>i</td><td>j</td><td>k</td><td>l</td><td>m</td><td>n</td><td>o</td></tr>
<tr><td>7</td><td>p</td><td>q</td><td>r</td><td>s</td><td>t</td><td>u</td><td>v</td><td>w</td><td>x</td><td>y</td><td>z</td><td>{</td><td>|</td><td>}</td><td>~</td><td>DEL</td></tr>
</table>

Speech: This is the table for ASCII control and printable characters

---

Slide 14: **Data Organization for 80x86 Microprocessors**

- Typical storage-units of numbers (*examples use only binary form*) are:

- Bit: One Binary digit (0 or 1). Three Bit data ➔ '010_B' or '111_B'

- Nibble: Consist of our Bits. Example, '**0**10**1**$_B$' or '**1**10**0**$_B$', where the leftmost bit is called Most-Significant-Bit (MSB) and the right most bit is called the Least-Significant-Bit (LSB).

  - Remember LSB of a nibble is considered to be bit-0 and MSB of nibble is considered to be bit-3 of the binary data.

  - Also remember one nibble of binary data corresponds to one Hexadecimal digit. Example, '$1111_B = F_H$' or '$0101_B = 5_H$'

- Byte: Consist of eight Bits. Example, '0101**1101**$_B$' or '**1111**0000$_B$', where the leftmost nibble is called the Most-Significant-Nibble and the right most nibble is called the Least-Significant-Nibble. A kilobyte (KByte) is 2^10 Byte = 1024 Bytes

- Word: Consist of sixteen Bits OR two consecutive bytes OR four nibbles. Example, '**1111** 0000 0101 **1101**$_B$', where the leftmost byte is called the Most-Significant-Byte and the right most byte is called the Least-Significant-Byte.

- Double-Word: Consist of 32 Bits of binary data OR two consecutive Words