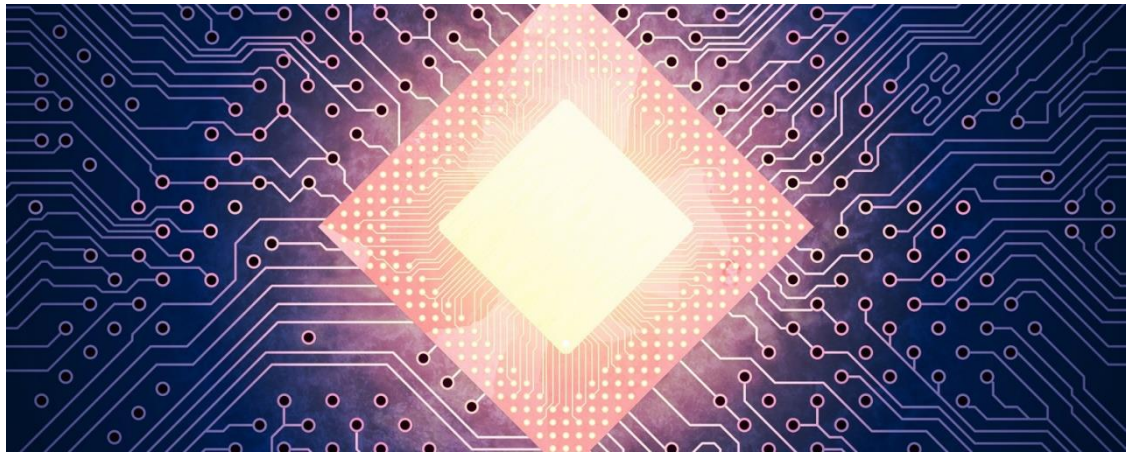
2



ET2223 Microprocessors Microcontrollers & Embedded Systems

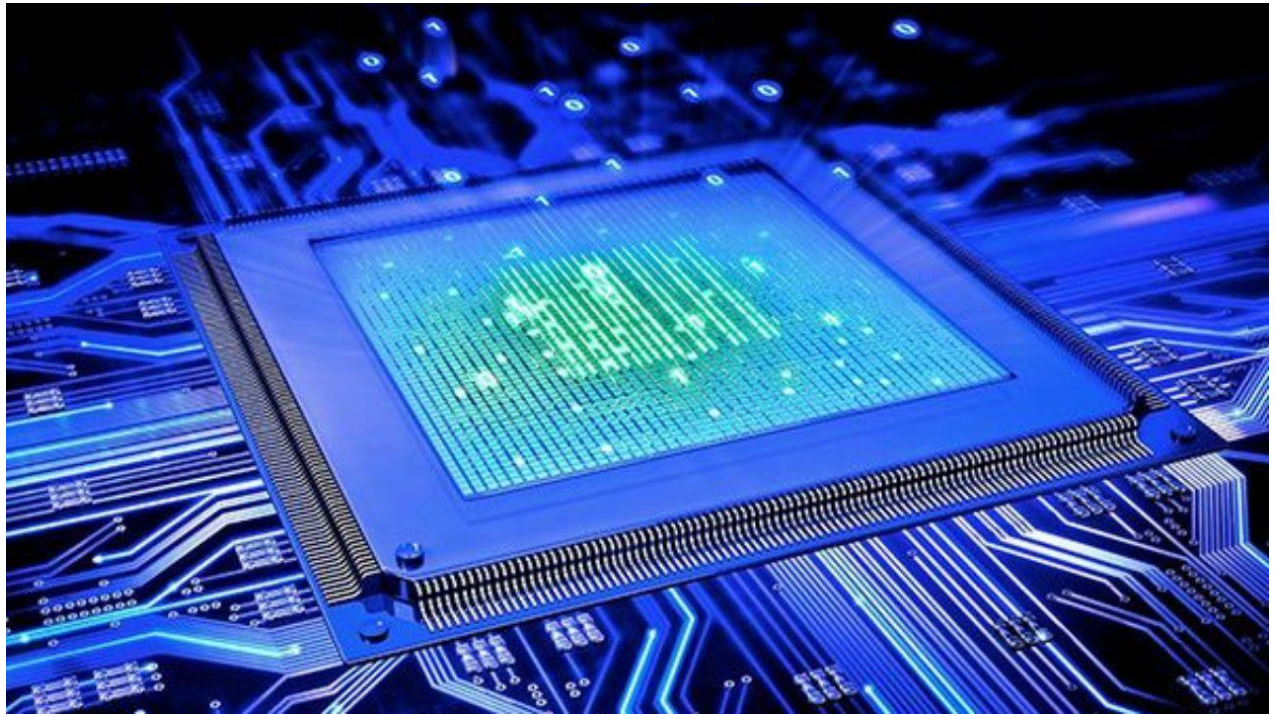
Microprocessor System

Sandali Goonatilleke

Department of Computer Engineering

General Sir John Kotelawala Defence University

Microprocessor



What is a Microprocessor

Microprocessor is a programmable logic device that can be used to control process, to turn devices on/off, or as data processing unit of a computer

- A miniature **electronic device** that handles arithmetic and logic operations
- A kind of an **integrated circuit** that can interpret and execute program instructions

History of Microprocessors

- **Fair child semiconductors** invented the **first Integrated Circuit in 1959**
- In 1971, the **first microprocessor Intel 4004** was invented

Generations of Microprocessors

1st Generation

1971 - Intel 4004 (4-bit microprocessor, 108kHz clock speed)

2nd Generation

In 1972 – Intel 8008 (8-bit microprocessor)

Intel 8080 – First commercially popular 8-bit microprocessor

History of Microprocessor (Cont'd)

3rd Generation

1978 – Intel 8086 (first 16-bit microprocessor)

4th Generation (32 bits microprocessors)

In the early 80s – Intel 80386

5th Generation

High-performance, high-speed 64 bits processors

Microprocessor and Microcomputer

- A computer that can be designed using a microprocessor is called microcomputer
- Microprocessor is the controlling unit of the micro-computer
- Fabricated on a small chip capable of performing ALU operations and communicating with the other devices connected to it
- Microprocessor communicates and operates in the binary numbers 0s and 1s

Microprocessor (Cont'd)

- It has a fixed set of instructions in the form of binary patterns called machine language
- Binary is difficult with humans and therefore binary instructions are given **abbreviated names** called **mnemonics** that forms assembly language

Microprocessor (Cont'd)

- A typical programmable machine or a microcomputer can be represented with three components
  - Microprocessor
  - Memory
  - I/O
- Three components work together and interact with each other to perform a given task

Microprocessor (Cont'd)

- The physical components of the system is known as hardware
- A set of instructions written for the microprocessor to perform a task is called program
- Collection of programs called software

Microcomputer

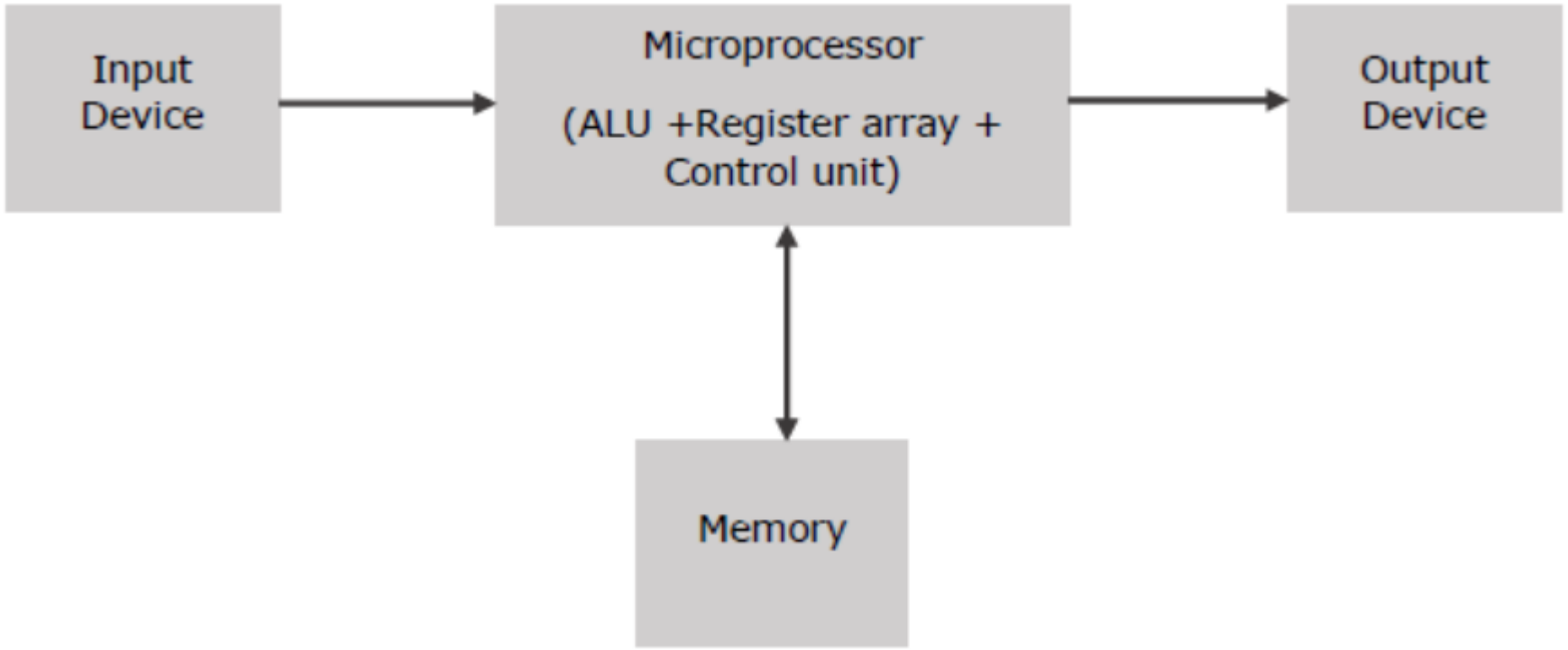


Figure: Basic Architecture of Microcomputer

Microcomputer

Memory

- Stores **binary information** as data and instructions
- To execute programs, the microprocessor reads instructions and data from the memory and performs the computing operations in its ALU section
- Results are either transferred to the output section for display or stored in memory for later use

Microcomputer

Memory

Contains two sections

- Read-only-memory (ROM)
- Read/write memory (R/WM) popularly known as Random-access-memory (RAM)

Microcomputer

System Bus

- Provides the communication path between the microprocessor and peripherals
- All of them share the same bus
- Group of wires to carry bits

Microcomputer (Cont')

Input/Output

- User can enter data and instructions into the memory through devices
- They are input devices
- Includes devices such as keyboard (hexadecimal keyboard), analog-to-digital converters
- Then, microprocessor reads the instructions and process data
- Results can be displayed by devices known as output devices

Microcomputer (Cont')

Input/Output

- It includes devices such as Light Emitting Diodes (LEDs), Cathode Ray Tube (CRT), printer, magnetic tape or another computer
- Typically, single board computers include LEDs and seven-segment LED as output devices

Microcomputer (Cont'd)

Microprocessor as CPU

- Consists of an ALU, register array, and a control unit

Arithmetic and Logic Unit

- Performs various arithmetical and logical operations on the data
- Mainly addition, subtraction, AND, OR, XOR…

Microcomputer (Cont'd)

Arithmetic and Logic Unit

- Add
- Subtract
- Logical AND
- Logical OR
- Logical Exclusive-OR
- Compare
- Left or Right shifts
- Rotates
- Increment
- Decrement
- Set bit
- Reset bit

Microprocessor

Register Array

- Consists of registers to store data temporarily
- Identified by letters like B, C, D, E, H, L and accumulator
- Some registers can be accessed by users

Microprocessor

Control Unit

- Controls the flow of data and instructions between the microprocessor , memory and peripherals
- Provides necessary timing and control signals to all the operations of the microprocessor

Terms Related to Microprocessor

1. **Instruction Set** – Set of instructions that the microprocessor can understand
2. **Bandwidth** – Number of bits processed in a single instruction
3. **Clock Speed (Clock Rate)** – Number of operations per second the microprocessor can perform (MHz, GHz)
4. **Word Length** – Width of internal data bus, registers, ALU, etc ranges from 4-64 bits.
5. **Data Types** – Has multiple data type formats like binary, BCD, ASCII, signed and unsigned numbers

Exercise

List Features of the Microprocessor

Microprocessor Instruction Set

- Microprocessors recognize and operate in binary numbers
- Each Microprocessor has its own binary words, instructions, meaning and language
- The words are formed combining a number of bits for a given machine
- The **word** is the number of bits the microprocessor recognizes and processes at a time

Microprocessor Instruction Set

- The length (word) ranges from 4 bits for small microprocessor based computers, to 32 bits for such large computers

Example: a 16-bit microprocessor has a word length equal to two bytes

- The instruction is defined as a complete task the microprocessor can perform
- It can be made up of one or more words

Microprocessor Instruction Set

- Each machine has its own set of instructions based on the design of its microprocessor
- To be intelligible to the microprocessor, instructions must be written in binary language also known as machine language
- Therefore, microprocessor manufactures have devised English like words to represent the binary instructions of a machine and programmers can write programs using these words

Microprocessor Instruction Set

- They are called **assembly language programs**
- Assembly language is **specific to a given machine**
- Assembly language programs are not transferable from one machine to another

Example:

1101010 → Machine Language

Add A → Assembly Language

Machine Language

- The number of bits in a word for a given machine is fixed
- The microprocessor design engineer selects combinations of bit patterns and gives a specific meaning to each combination by using electronic logic circuits called **instructions**
- **The set of instructions designed into the machine known as machine language**

Machine Language

- A Binary language composed of 1s and 0s
- Words, instructions and their meaning are specific to each computer

Z80 Machine Language

- Microprocessor with **8-bit** word length
- Has 159 instructions
- **00111100** is an instruction that increments the number in the register called the accumulator (A) by one
- **10000000** is an instruction which adds the number in the register called B to the number in the accumulator and keeps the sum in the accumulator

Classification of Microprocessors

There are three types

- RISC Processors
- CISC Processors
- Special Processors

Classification of Microprocessors

- **Explore RISC and CISC processors**
- **Compare and contrast RISC and CISC processors**

Classification of Microprocessors (Cont'd)

Explore special processors

Classification of Microprocessors

RISC Processor (Reduced Instruction Set Computer Processor)

- Designed to reduce the execution time by simplifying the instruction set of the computer
- Each instruction requires only one clock cycle to execute results in uniform execution time
- This reduces the efficiency and more RAM to store the instructions
- The compiler is needed to convert high-level language instructions into machine code

Classification of Microprocessors

Characteristics of RISC Processors

- Instructions are simple. So that, decoding of instructions is also simple
- The size of instructions is under the one-word size
- Instruction takes one clock cycle to execute
- The number of registers is more
- The address modes are also simple
- Can be used for pipelining

Examples: IBM RS6000, DEC Alpha 21064, DEC Alpha 21164

Classification of Microprocessors

CISC Processors (Complex Instruction Set Computer Processors)

- Designed to minimize the number of instructions per program, ignoring the number of cycles per instruction
- The compiler has to do very little work to translate a high-level language into assembly level language/machine code because the length of the code is relatively short
- Small RAM is required to store the instructions

Classification of Microprocessors

Characteristics of CISC Processors

- As the instructions are complex hence, the decoding of instructions is also complex
- The size of instructions is greater than the one-word size
- Instruction can take more than one clock cycle to execute
- The number of registers is less since most of the operations are performed in memory itself
- The address modes are also complex

Examples: Intel 386, Intel 486, Pentium, Pentium Pro, Pentium II

Classification of Microprocessors

Exercise

1. **Explore EPIC microprocessors**
2. **Find advantages of RICS and CISC processors**

Microcomputer System

- Contains primarily three components
  - **The microprocessor unit (MPU)**
  - **Memory**
  - **I/O**
- MPU is the **central player**, it communicates with memory and I/O devices, processes data and controls timing of all its operations
- Memory and I/O are integral parts of a micro-computer system
- Memory – latches and registers

Microcomputer System

Generalized Microprocessor Unit

- MPU is a programmable logic device with a designed set of instructions
- When MPU is executing a program, it communicates frequently with memory and I/O devices
- The process contains fetch, decode and execute operations

Microcomputer System

Generalized Microprocessor Unit

- Communication process and related operations between MPU and external devices classified into two categories
  - Program initiated operations
  - Peripheral initiated operations
- Memory
  - ROM
  - R/W memory

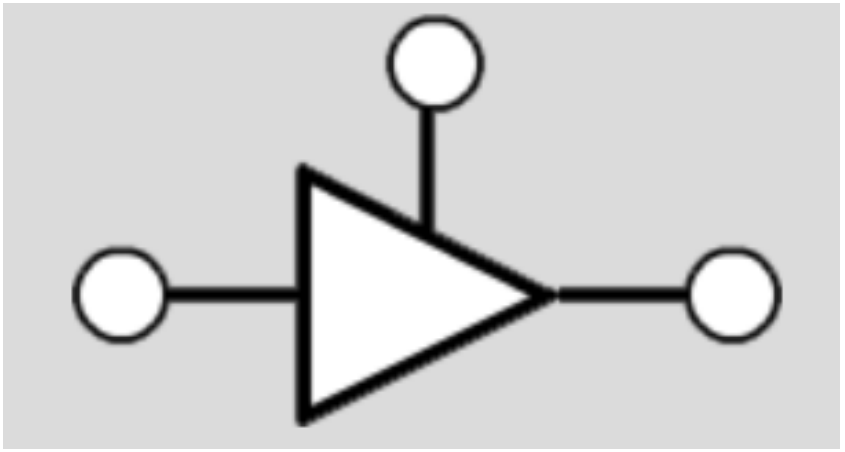
Microcomputer System

Flip-Flop or Latch as Storage Elements

- Memory – Circuit can store bits (0/1)
- **FF** or **latch** basic elements of memory
- To write or store bit in a latch – input data bit + enable signal
- In the stored bit always available at output

Microcomputer System

Tri State Buffer



Enable Input	Input A	Output
false	false	hi-Z
false	true	hi-Z
true	false	false
true	true	true

Microcomputer System

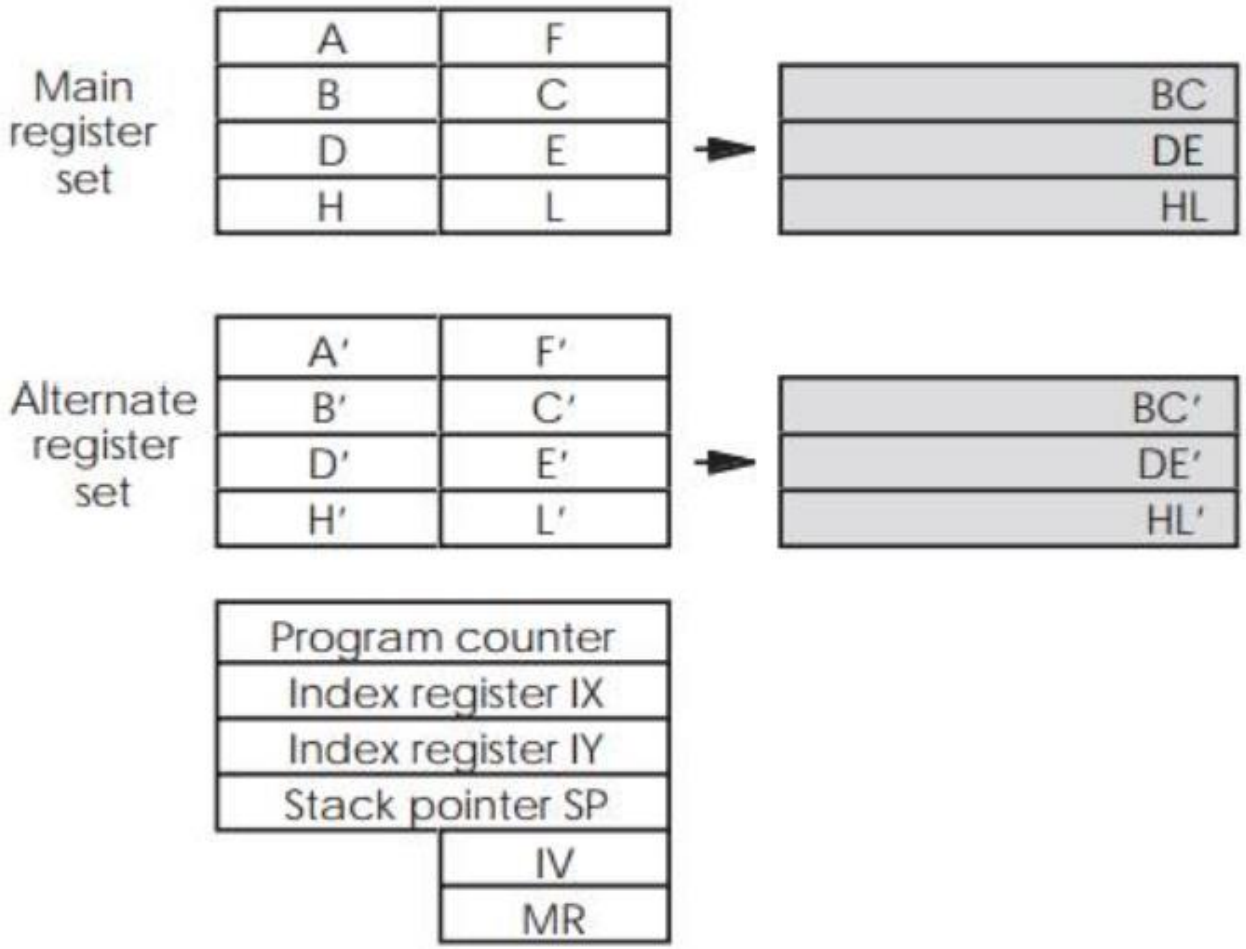
Tri State Buffer

- If tri states buffer connected , the stored bit can be only read when the buffer enabled
- Similarly input buffer
- The latch can store only one bit called memory cell
- **4 latches grouped together to form a register which has 4 I/O lines**
- **The number of bits stored in a register is called a memory word**

Z80 Microprocessor

- **Zilog Z80**
- An **8-bit microprocessor** introduced by Zilog in 1976
- An extension and enhancement of the Intel 8080
- Mainly aimed at embedded systems
- Z80 also became one of the most widely used CPUs in desktop_computers and home computers from the 1970s to the mid-1980s

Z80 Microprocessor Programming Model



Z80 Microprocessor Programming Model (Cont'd)

Accumulator

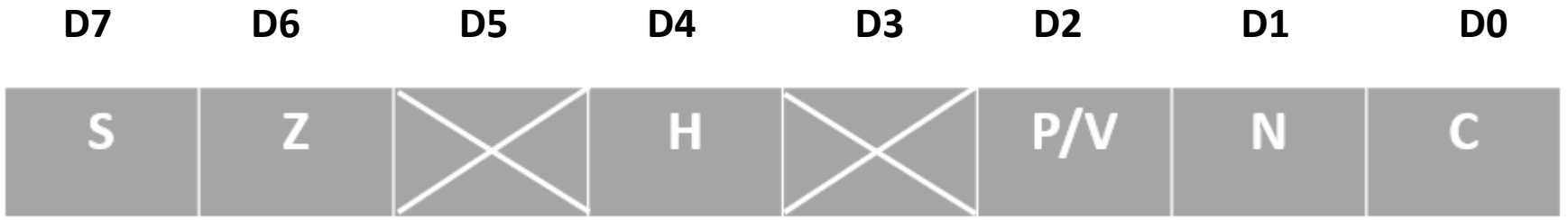
- **8 bit register**, part of ALU
- Identified as **register A**
- This register is used to store 8-bit data
- Perform arithmetic and logical operations
- Results are stored in accumulator
- A' cannot directly addressed, but can exchange with A

Z80 Microprocessor Programming Model (Cont'd)

Flag Register

- ALU includes 6 flip-flops that are set/reset according to the data conditions often an **ALU operation**
- **The status of each flip-flop is known as flag**
- The status of six flags stored in flag register so that can be examined if necessary

Z80 Microprocessor Programming Model (Cont'd)



S – Sign
Z – Zero
H – Half Carry
P/V – Parity/Overflow
N – Add/Subtract
C - Carry

Z80 Microprocessor Programming Model (Cont'd)

Flags

- **Carry Flag** – If arithmetic operation generates a carry or borrow a bit, the flag is set otherwise reset
- **Zero Flag** – 8 bit operation result zero, z flag is set otherwise it is reset
- **Sign Flag** – After ALU operation, the MSB D7 is 1 the sign flag is set, otherwise reset. Intepretation depend on the number system

Z80 Microprocessor Programming Model (Cont'd)

Parity/Overflow Flag

- Use to check the parity, number of 1s
- If the number of 1s in the result is **even**, this flag is **set**
- If the number of 1s in the result **odd**, this flag is **reset**

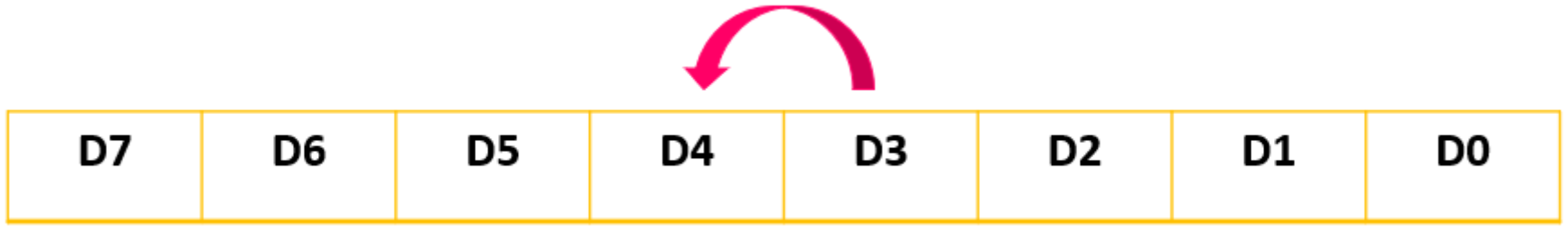
Add/Subtract Flag

- Used in internally for BCD operations to distinguish +/-

Z80 Microprocessor Programming Model (Cont'd)

Half Carry Flag

- An arithmetic operation, this flag is affected by carry or borrow between D3, D4
- Otherwise is reset
- Used in Binary Coded Decimal (BCD)



Z80 Microprocessor Programming Model (Cont'd)

General Purpose and Alternate Registers

- There are **6 general purpose registers in Z80 microprocessor**
- They are **B,C,D,E,H,L**
- These registers used for storing data during program execution
- They can be combined as pairs such as BC, DE, HL to perform 16 bit operations or hold a memory address

Z80 Microprocessor Programming Model (Cont'd)

General Purpose and Alternate Registers

- The programmer can use these registers to load or copy data

Example: LD B,C copies data from register C into register B

- **Conceptually registers can view as memory locations**
- Some microprocessors use memory as registers as they don't have these kind of registers

Z80 Microprocessor Programming Model (Cont'd)

General Purpose and Alternate Registers

- The alternate registers such as (B', C', D', E', H' AND L') they are not directly available for the programmers and can be access via **exchange instructions**

Z80 Microprocessor Programming Model (Cont'd)

Z80 Instruction Set

- The instruction set of a microprocessor determine the capability of its operations

Example: Z80 instruction set include an in instruction that can copy content from one block of memory location for another

- Z80 microprocessor has 158 instructions

Z80 Microprocessor Programming Model (Cont'd)

Z80 Instruction Set

- Each instruction has two parts

  Opcode (Operation Code) – The task to be performed (Load, Add….)

  Operand – Identifies the data to be operated on

Z80 Microprocessor Programming Model (Cont'd)

Instruction Format

- **A command to the microprocessor to perform a given task**
- The size of Z80 instructions ranges from one to four bytes
- Thus, the number of memory registers required to write them varies
- The Z80 instruction set can be classified into four groups according to the length of an instructions
- 1 byte to 4 bytes

Z80 Microprocessor Programming Model (Cont'd)

1 – Byte Instruction

- In 1 byte instruction, the opcode and the operand are included in the same byte

Task	Opcode	Operand	Binary Code
Copy the contents of register B into Accumulator A	LD	A,B	01111000 (78H)
Add the contents of register B to the contents of A	ADD	A,B	10000000 (80H)

Z80 Microprocessor Programming Model (Cont'd)

1 – Byte Instruction

The one byte instruction is performing two different tasks

2 – Byte Instructions

In 2 byte instructions, the first byte specifies the opcode and the second byte specifies the operand

Z80 Microprocessor Programming Model (Cont'd)

2 – Byte Instructions

Task	Opcode	Operand	Binary Code
Copy the contents of register B into Accumulator A	LD	A,B	01111000 (78H)
Add the contents of register B to the contents of A	ADD	A,B	10000000 (80H)

Z80 Microprocessor Programming Model (Cont'd)

3 – Byte Instructions

1st byte specifies the opcode and the following two bytes specify the 16 bit address or data

Task	Opcode	Operand	Binary Code
Copy data from memory address 2080H into the Accumulator	LD	A,2080H	00111010 10000000 (80H) 00100000 (20H)

Z80 Microprocessor Programming Model (Cont'd)

The Z80 instruction set can be divided into six categories

Data copy (transfer) or load operations

Arithmetic operations

Logic operations

Bit manipulation

Branch operations

Machine control operations

Z80 Microprocessor Programming Model (Cont'd)

Data Copy or Load Operations

The Z80 has numerous instructions that copy data from one location called source to another location called destination (source content won't erased or destroyed)

Z80 Microprocessor Programming Model (Cont'd)

Data Copy or Load Operations

	Data Copy Operations	Examples
1.	From one register into another register	Copy the contents of register B into the Accumulator LD A,B (LD means load)

Z80 Microprocessor Programming Model (Cont'd)

Data Copy or Load Operations

	Data Copy Operation	Examples

2.	(a)Specific data byte into a register or a memory location
	
	(b)Specific 16 bit data into a register pair
		Load register B with the hexadecimal number 32
		LD B,32H
		
		Load register pair HL, with hexadecimal number 2050
		LD HL,2050H

Z80 Microprocessor Programming Model (Cont'd)

Data Copy or Load Operations

	Data Copy Operation	Examples
3.	From a memory location into a register or vice-versa	Copy data from memory location 2080H into the Accumulator LD A, (2080H)

Z80 Microprocessor Programming Model (Cont'd)

Data Copy or Load Operations

	Data Copy Operation	Examples
4.	(a) From an input port into the Accumulator	Read data from input port 01H and copy into the Accumulator IN A,(01H)
	(b) From the Accumulator into an output port	Write (read) the contents of the Accumulator into port 07H 001 (07H)A

Z80 Microprocessor Programming Model (Cont'd)

Write and Execute Assembly Language Program

- Assembly language program is a sequence of instruction written in mnemonics to perform a specific task
- To write a program, need to divide a given problem into small steps and translate these step into operation of Z80 can perform

Z80 Microprocessor Programming Model (Cont'd)

Adding two Hexadecimal Numbers

- Write instructions to load two hexadecimal numbers 32H & A2H into registers B and C
- Add the numbers
- Display the sum at LED output PORT 1

Problem Analysis

- Load the numbers into the registers
- Add the numbers
- Display the sum at the output PORT 1

Z80 Microprocessor Programming Model (Cont'd)

Assembly Language Program

LD B, 32H	Load Register B with 32H
LD C, A2H	Load register C with A2H
LD A,C	Copy contents of C into Accumulator to perform addition B and C cannot add directly
ADD A,B	Add two bytes and save the sum in A
OUT(01H), A	Display Accumulator contents at port 01H
HALT	End

Z80 Microprocessor Programming Model (Cont'd)

HEXCODE

06 32	2 byte instruction
OE A2	2 byte instruction
79	1 byte instruction
80	1 byte instruction
D3 01	2 byte instruction
76	1 byte instruction

Thank You!