Computer Architecture SAP-1

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

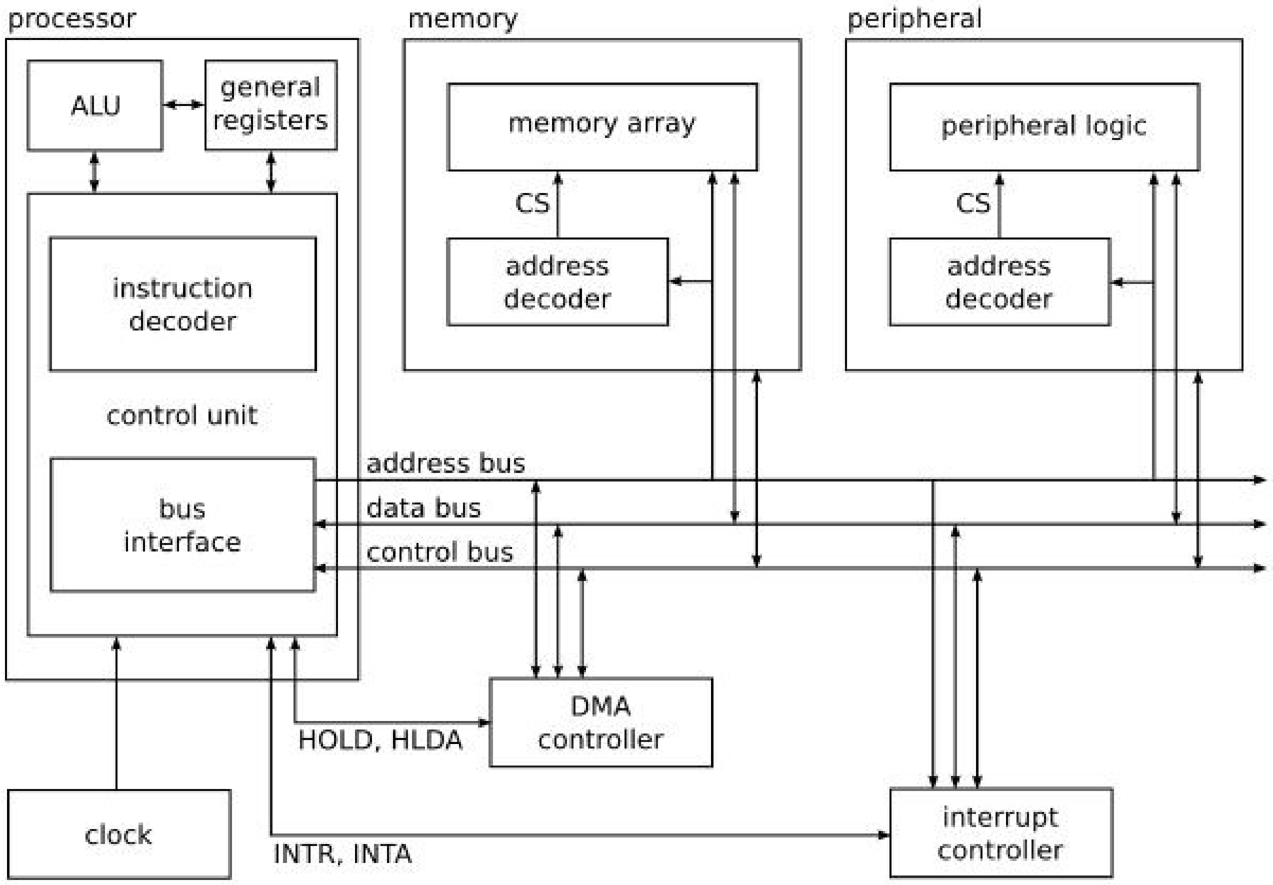
What is computer architecture?

- **Computer architecture** can be defined as a set of rules and methods that describe the functionality, management and implementation of computers. To be precise, it is nothing but rules by which a system performs and operates.
- The main role of Computer Architecture is to balance the performance, efficiency, cost and reliability of a computer system.
- **For Example** − Instruction set architecture (ISA) acts as a bridge between computer's software and hardware. It works as a programmer's view of a machine.

Structure

- Generally, computer architecture consists of the following –
  - Processor
  - Memory
  - Peripherals
- All the above parts are connected with the help of system bus, which consists of address bus, data bus and control bus.

The diagram given below depicts the computer architecture –



- Thank you

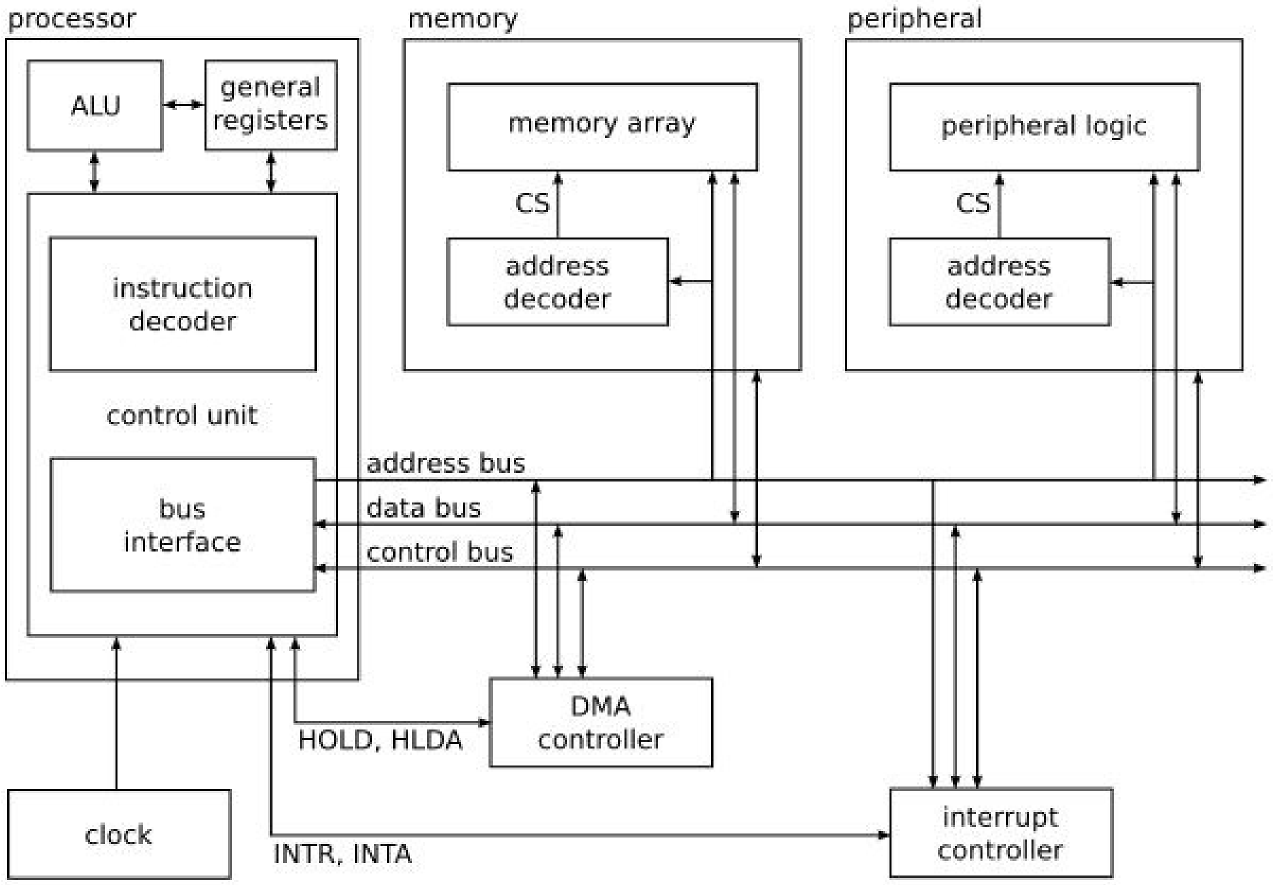
Computer Architecture SAP-1

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

The diagram given below depicts the computer architecture –



Computer Instructions

- Computer instructions are a set of machine language instructions that a particular processor understands and executes. A computer performs tasks on the basis of the instruction provided.
- The basic computer has 16-bit instruction register (IR) which can denote either memory reference or register reference or input-output instruction.

PC direction or Instruction execution process

- Essential PC directions are the principal tasks that a PC can perform. These directions are executed by the focal handling unit (central processor) of a PC, and they structure the reason for additional perplexing tasks. A few instances of essential PC directions include:

1. **Load:** This guidance moves information from the memory to a computer processor register.
2. **Store:** This guidance moves information from a computer chip register to the memory.
3. **Add:** This guidance adds two qualities and stores the outcome in a register.

PC direction or Instruction execution process

4.Subtract: This guidance deducts two qualities and stores the outcome in a register.

5.Multiply: This guidance duplicates two qualities and stores the outcome in a register.

6.Divide: This guidance isolates two qualities and stores the outcome in a register.

7.Branch: This guidance changes the program counter to a predefined address, which is utilized to execute restrictive and genuine leaps.

8.Jump: This guidance changes the program counter to a predefined address.

9.Compare: This guidance looks at two qualities and sets a banner demonstrating the consequence of the examination.

10.Increment: This guidance adds 1 to a worth in a register or memory area.

- Thank you

Basics of Computer Architecture Instructions

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

Basic computer instructions

- Basic computer instructions are the elementary operations that a computer system can perform.
- These instructions are typically divided into three categories: data movement instructions, arithmetic and logic instructions, and control instructions.
  - Data movement instructions are used to move data between different parts of the computer system. These instructions include load and store instructions
  - Arithmetic and logic instructions are used to perform mathematical operations and logical operations on data stored in the system.
  - Control instructions are used to control the flow of instructions within the computer system.

Instruction Structure

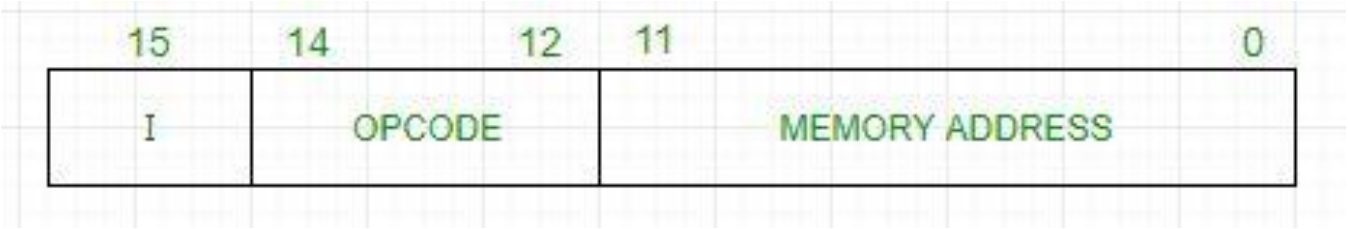
- An instruction comprises of groups called fields. These fields include:
  - The Operation code (Opcode) field which specifies the operation to be performed.
  - The Address field which contains the location of the operand, i.e., register or memory location.
  - The Mode field which specifies how the operand will be located.

A basic computer has three instruction code formats which are:

1. Memory - reference instruction
2. Register - reference instruction
3. Input-Output instruction

Memory Reference

- These instructions refer to memory address as an operand. The other operand is always accumulator. Specifies 12-bit address, 3-bit opcode (other than 111) and 1-bit addressing mode for direct and indirect addressing.



- **Example –** IR register contains = 0001XXXXXXXXXXXX, i.e. ADD after fetching and decoding of instruction we find out that it is a memory reference instruction for ADD operation.

- The End

Basics of Computer Architecture Instructions

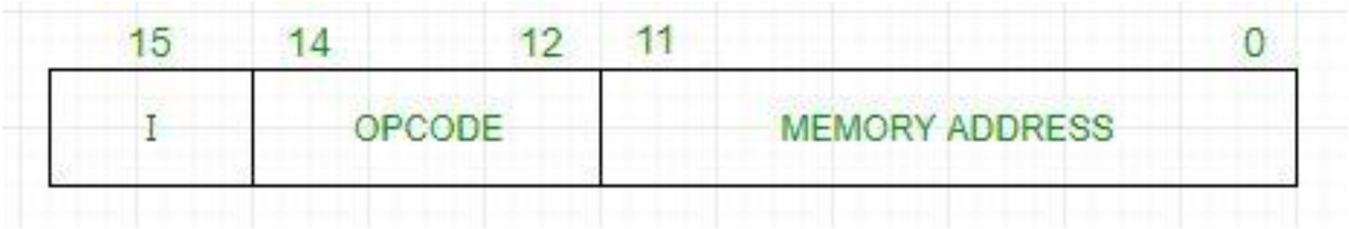
MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

Memory Reference

- These instructions refer to memory address as an operand. The other operand is always accumulator. Specifies 12-bit address, 3-bit opcode (other than 111) and 1-bit addressing mode for direct and indirect addressing.



- **Example –** IR register contains = 0001XXXXXXXXXXXX, i.e. ADD after fetching and decoding of instruction we find out that it is a memory reference instruction for ADD operation.

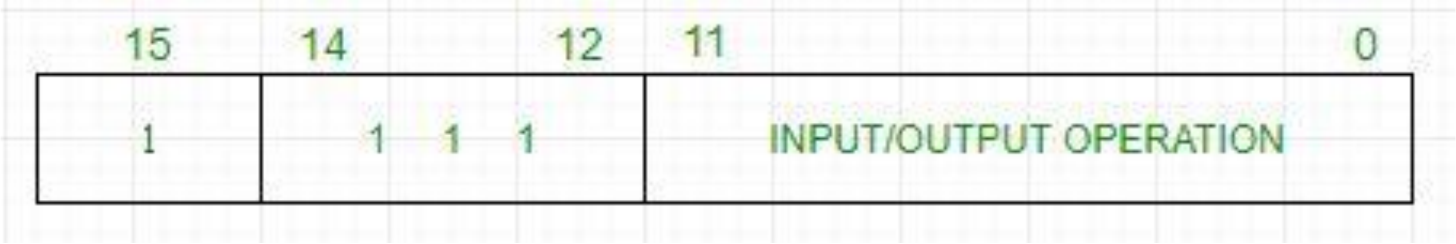
Register Reference

- These instructions perform operations on registers rather than memory addresses. The IR(14 – 12) is 111 (differentiates it from memory reference) and IR(15) is 0 (differentiates it from input/output instructions). The rest 12 bits specify register operation.

- **Example –** IR register contains = 0111001000000000, i.e. CMA after fetch and decode cycle we find out that it is a register reference instruction for complement accumulator.

Input/Output

- These instructions are for communication between computer and outside environment. The IR(14 – 12) is 111 (differentiates it from memory reference) and IR(15) is 1 (differentiates it from register reference instructions). The rest 12 bits specify I/O operation.



- **Example –** IR register contains = 1111100000000000, i.e. INP after fetch and decode cycle we find out that it is an input/output instruction for inputing character. Hence, INPUT character from peripheral device.

- The End

Basics of Registers

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

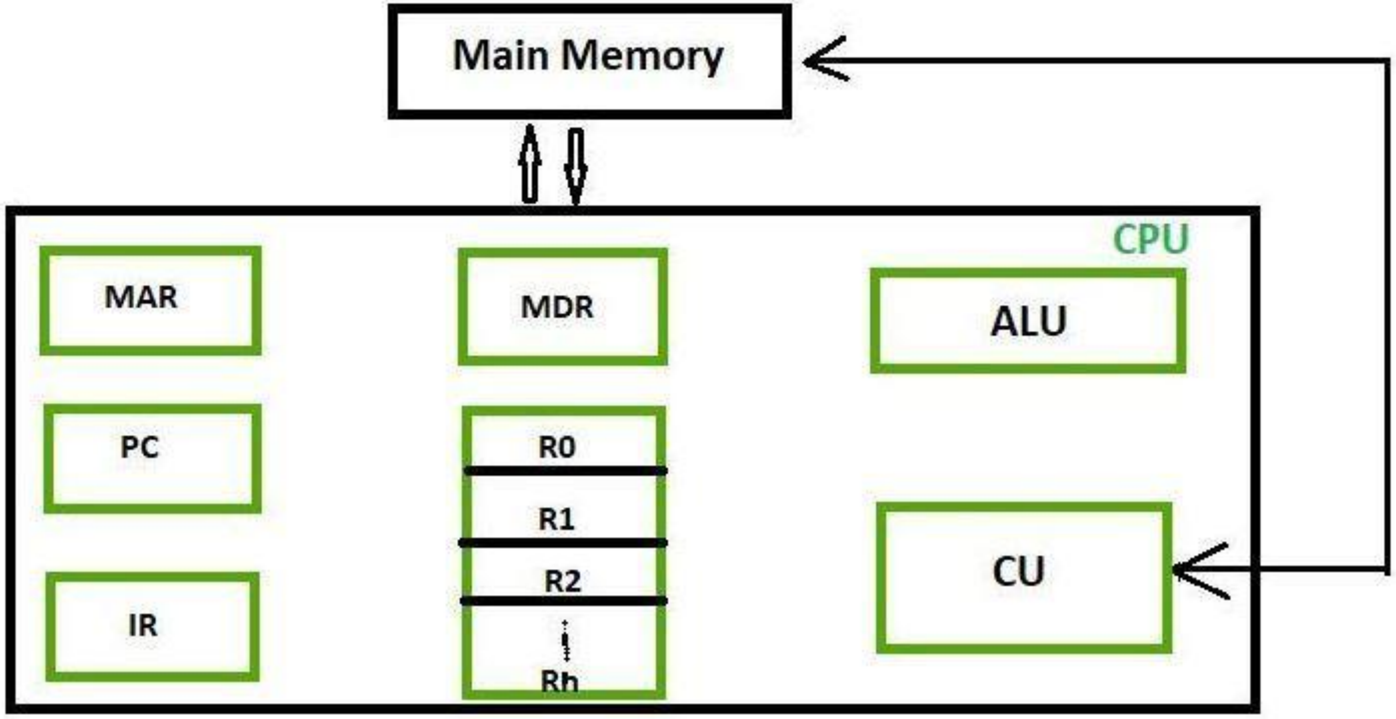
PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

General Purpose Register

- A register serves as a quick memory for accepting, storing, and sending data and instructions that the CPU will need right away.
- A register is a collection of flip-flops, Single bit digital data is stored using flip-flops.
- By combining many flip-flops, the storage capacity can be extended to accommodate a huge number of bits.
- We must utilize an n-bit register with n flip flops if we wish to store an n-bit word.

- The gates govern the flow of information, i.e., when and how the information is sent into a register, whereas the flip-flops store the binary information.



Working of Registers

- When we provide the system with input, that input is stored in registers, and when the system returns results after processing, those results are also drawn from the registers. so that the CPU can use them to process the data that the user provides.
- Registers are performed based on three operations:
  - **Fetch:** The Fetch Operation is used to retrieve user-provided instructions that have been stored in the main memory. Registers are used to fetch these instructions.
  - **Decode:** The Decode Operation is used to interpret the Instructions, which means that the CPU will determine which Operation has to be carried out on the Instructions after the Instructions have been decoded.
  - **Execute:** The CPU manages the Execute Operation. The results that the CPU generates are then stored in the memory before being presented on the user screen.

- The End

Basics of Registers

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

Types of Registers

- Status and control registers.
- General-purpose data registers.
- Special purpose register.

For better understanding about register we will dig into General Purpose Register.

General-Purpose Registers

- General purpose registers are extra registers that are present in the CPU and are utilized anytime data or a memory location is required. These registers are used for storing operands and pointers. These are mainly used for holding the following:
  - Operands for logical and arithmetic operations
  - Operands for address calculation
  - Memory pointers

There are 3 types of General-purpose data registers they are:

- **Data registers**
- **Pointer registers**
- **Index registers**

Data registers

Data registers consists of four 32-bit data registers, which are used for arithmetic, logical and other operations. Data registers are again classified into 4 types they are:

- **AX:** This is known as the accumulator register. Its 16 bits are split into two 8-bit registers, AH and AL, allowing it to execute 8-bit instructions as well.
- **BX:** This is called a Base register. It has 16 bits and is split into two registers with 8 bits each, BH and BL. An address register is the BX register.
- **CX:** This is known as the Count register. Its 16 bits are split into two 8-bit registers, CH and CL, allowing it to execute 8-bit instructions as well. This acts as a counter for loops.
- **DX:** This is known as the Data register. Its 16 bits are split into two 8-bit registers, DH and DL so that it can execute 8-bit instructions as well. In I/O operations, the data register can be used as a port number. It is also applied to division and multiplication.

Pointer registers

- The pointer registers consist of 16-bit left sections (SP, and BP) and 32-bit ESP and EBP registers.
  - **SP:** This is known as a Stack pointer used to point the program stack. For accessing the stack segment, it works with SS. It has a 16-bit size. It designates the item at the top of the stack. The stack pointer will be (FFFE)H if the stack is empty. The stack segment is relative to its offset address.
  - **BP:** This is known as the Base pointer used to point data in the stack segments. We can utilize BP to access data in the other segments, unlike SP. It has a 16-bit size. It mostly serves as a way to access parameters given via the stack. The stack segment is relative to its offset address.

Index registers

- The 16-bit rightmost bits of the 32-bit ESI and EDI index registers. SI and DI are sometimes employed in addition and sometimes in subtraction as well as for indexed addressing.
  - **SI:** This source index register is used to identify memory addresses in the data segment that DS is addressing. Therefore, it is simple to access successive memory locations when we increment the contents of SI. It has a 16-bit size. Relative to the data segment, it has an offset.
  - **DI:** The function of this destination index register is identical to that of SI. String operations are a subclass of instructions that employ DI to access the memory addresses specified by ES. It is generally used as a Destination index for string operations.

- The End

Shift Registers in Digital Logic

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

Flip-flop

- Flip flops can be used to store a single bit of binary data (1 or 0).
- However, in order to store multiple bits of data, we need multiple flip-flops. N flip flops are to be connected in order to store n bits of data.
- A **Register** is a device that is used to store such information.
- It is a group of flip-flops connected in series used to store multiple bits of data.
- The information stored within these registers can be transferred with the help of **shift registers**.

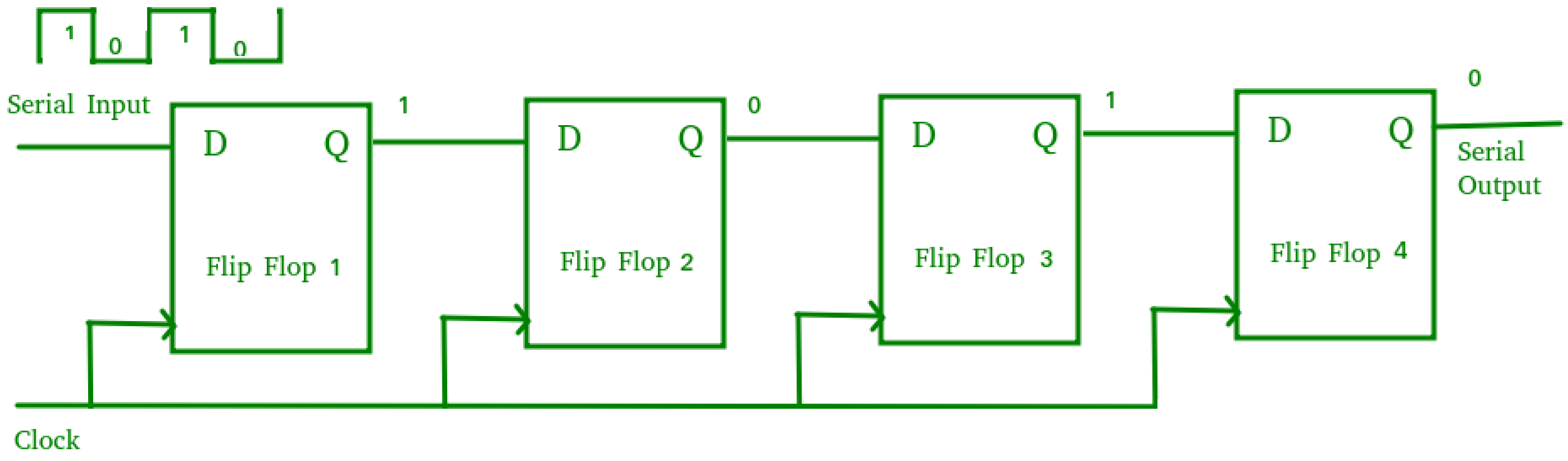
Shift Register

- Shift Register is a group of flip flops used to store multiple bits of data.
- The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses.
- An n-bit shift register can be formed by connecting n flip-flops where each flip-flop stores a single bit of data.
- The registers which will shift the bits to the left are called “Shift left registers”.
- The registers which will shift the bits to the right are called “Shift right registers”. Shift registers are basically of following types.

Serial-In Serial-Out Shift Register (SISO)

- The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as a Serial-In Serial-Out shift register.
- Since there is only one output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In Serial-Out Shift Register.
- The logic circuit given below shows a serial-in serial-out shift register.
- The circuit consists of four D flip-flops which are connected in a serial manner.
- All these flip-flops are synchronous with each other since the same clock signal is applied to each flip-flop.

Serial-In Serial-Out Shift Register (SISO)



The above circuit is an example of a shift right register, taking the serial data input from the left side of the flip flop. The main use of a SISO is to act as a delay element.

- The End

Shift Registers in Digital Logic

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

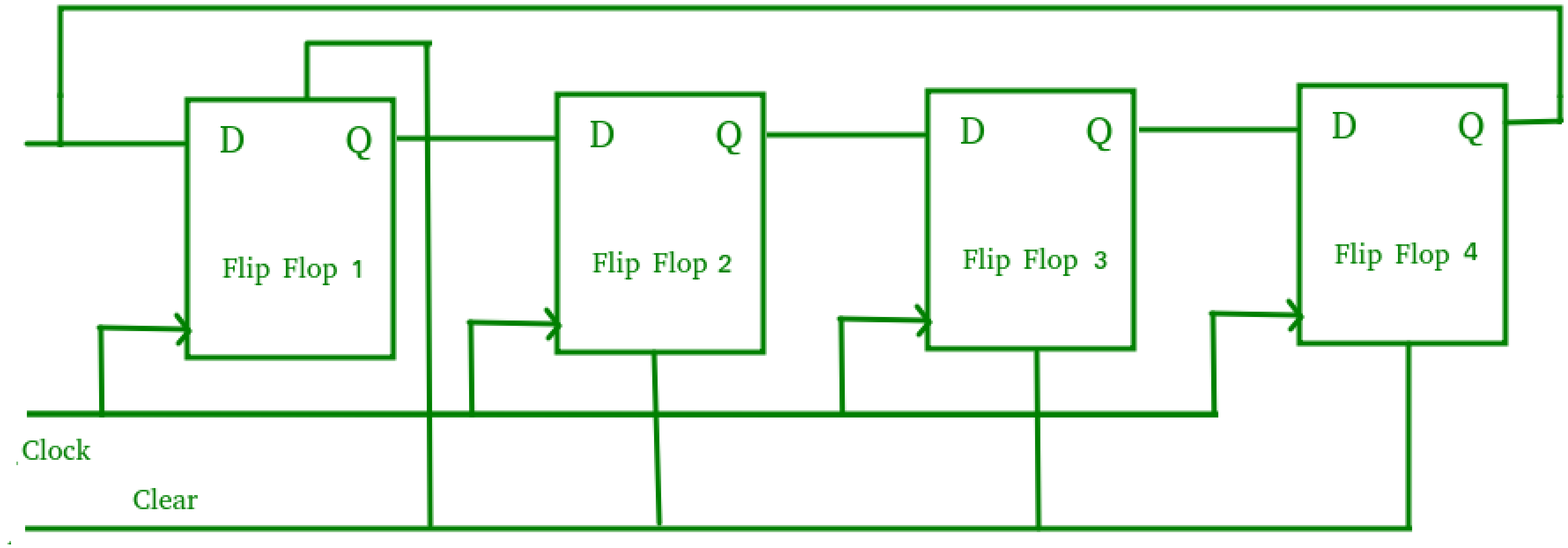
Shift Register Counter

- Shift Register Counters are the shift registers in which the outputs are connected back to the inputs in order to produce particular sequences. There are basically two types:
  - Ring Counter
  - Johnson Counter
- **Ring Counter**

A ring counter is basically a shift register counter in which the output of the first flip-flop is connected to the next flip-flop and so on and the output of the last flip-flop is again fed back to the input of the first flip-flop, thus the name ring counter. The data pattern within the shift register will circulate as long as clock pulses are applied. The logic circuit given below shows a Ring Counter.

Clock Pulse	Q1	Q2	Q3	Q4
0	1	0	0	1
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1

- The circuit consists of four D flip-flops which are connected.
- Since the circuit consists of four flip-flops the data pattern will repeat after every four clock pulses as shown in the truth table.
- A Ring counter is generally used because it is self-decoding.
- No extra decoding circuit is needed to determine what state the counter is in.


D
Q
Flip Flop 1
D
Q
Flip Flop 2
D
Q
Flip Flop 3
D
Q
Flip Flop 4
Clock
Clear

- The End

Registers in Computer Organization

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

Types of Register in Computer Organization

- In Computer Organization, the register is utilized to acknowledge, store, move information and directions that are being utilized quickly by the CPU. There are different kinds of registers utilized for different reasons. Some of the commonly used registers are:
  - AC (accumulator)
  - DR (Data registers)
  - AR (Address registers)
  - PC (Program counter)
  - MDR (Memory data registers)
  - IR (index registers)
  - MBR (Memory buffer registers)

- These registers are utilized for playing out the different operations.
- When we perform some operations, the CPU utilizes these registers to perform the operations.
- When we provide input to the system for a certain operation, the provided information or the input gets stored in the registers.
- Once the ALU arithmetic and logical unit process the output, the processed data is again provided to us by the registers.
- The sole reason for having a register is the quick recovery of information that the CPU will later process.
- The CPU can use RAM over the hard disk to retrieve the memory, which is comparatively a much faster option, but the speed retrieved from RAM is still not enough.
- Therefore, we have catch memory, which is faster than registers. These registers work with CPU memory like catch and RAM to complete the task quickly.

Types of Register in Computer Organization

S.NO	NAME	SYMBOL	FUNCTIONING
1	Accumulator	AC	An accumulator is the most often utilized register, and it is used to store information taken from memory.
2	Memory address registers	MAR	Address location of memory is stored in this register to be accessed later. It is called by both MAR and MDR together
3	Memory data registers	MDR	All the information that is supposed to be written or the information that is supposed to be read from a certain memory address is stored here
4	General-purpose register	GPR	Consist of a series of registers generally starting from R0 and running till Rn - 1. These registers tend to store any form of temporary data that is sent to a register during any undertaking process. More GPR enables the register to register addressing, which increases processing speed.

Types of Register in Computer Organization

5	Program counter	PC	These registers are utilized in keeping the record of a program that is being executed or under execution. These registers consist of the memory address of the next instruction to be fetched. PC points to the address of the next instruction to be fetched from the main memory when the previous instruction has been completed successfully. Program Counter (PC) also functions to count the number of instructions. The incrementation of PC depends on the type of architecture being used. If we use a 32-bit architecture, the PC gets incremented by 4 every time to fetch the next instruction.
6	Instructions registers	IR	Instruction registers hold the information about to be executed. The immediate instructions received from the system are fetched and stored in these registers. Once the instructions are stored in registers, the processor starts executing the set instructions, and the PC will point to the next instructions to be executed

- The End

Registers in Computer Organization

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

Types of Register in Computer Organization

7	Condition code registers		These have different flags that depict the status of operations. These registers set the flags accordingly if the result of operation caused zero or negative
8	Temporary registers	TR	Holds temporary data
9	Input registers	INPR	Carries input character
10	Output registers	OUTR	Carries output character
11	Index registers	BX	We use this register to store values and numbers included in the address information and transform them into effective addresses. These are also called base registers. These are used to change operand address at the time of execution, also stated as BX
12	Memory buffer register	MBR	MBR - Memory buffer registers are used to store data content or memory commands used to write on the disk. The basic functionality of these is to save called data from memory. MBR is very similar to MDR

Types of Register in Computer Organization

13	Stack control registers	SCR	Stack is a set of location memory where data is stored and retrieved in a certain order. Also called last in first out (LIFO), we can only retrieve a stack at the second position only after retrieving out the first one, and stack control registers are mainly used to manage the stacks in the computer. SP - BP is stack control registers. Also, we can use DI, SI, SP, and BP as 2 byte or 4-byte registers. EDI, ESI, ESP, and EBP are 4 - byte registers
14	Flag register	FR	Flag registers are used to indicate a particular condition. The size of the registered flag is 1 - 2 bytes, and each registered flag is furthermore compounded into 8 bits. Each registered flag defines a condition or a flag. The data that is stored is split into 8 separate bits. Basic flag registers - Zero flags Carry flag Parity flag Sign flag Overflow flag.

Types of Register in Computer Organization

15	Segment register	SR	Hold address for memory
16	Data register	DX	Hold memory operand

- The End

General purpose registers in 8086 microprocessor

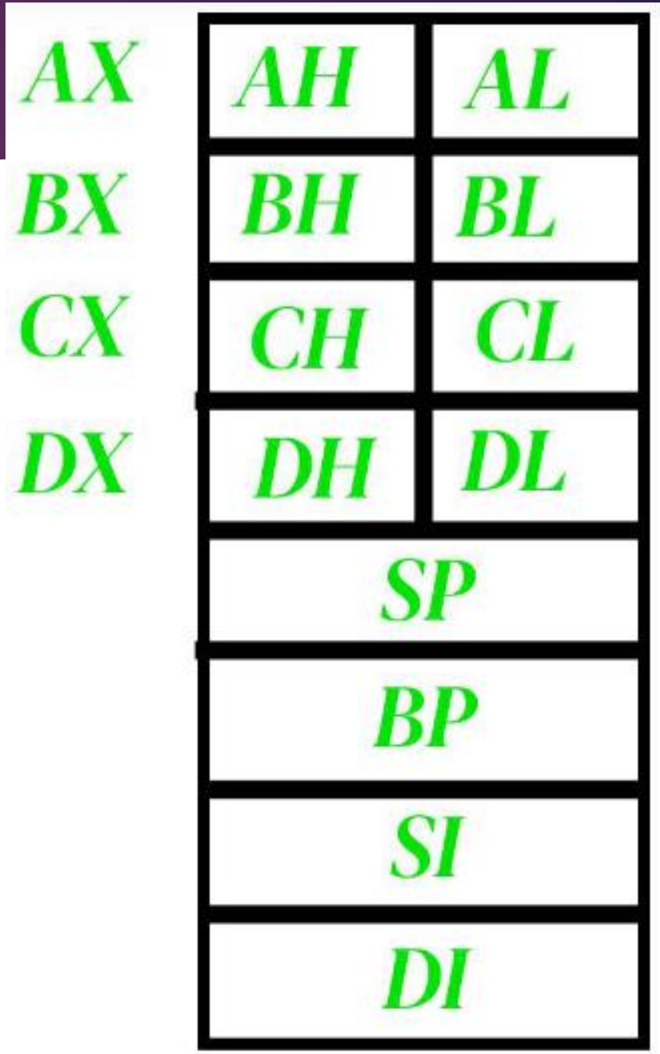
MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

GPR in 8086

- General-purpose registers are used to store temporary data within the microprocessor.
- There are 8 general-purpose registers in the 8086 microprocessor.



1. AX: This is the accumulator. It is of 16 bits and is divided into two 8-bit registers AH and AL to also perform 8-bit instructions. It is generally used for arithmetical and logical instructions but in 8086 microprocessor it is not mandatory to have an accumulator as the destination operand. Example:

ADD AX, AX (AX = AX + AX)

2. BX: This is the base register. It is of 16 bits and is divided into two 8-bit registers BH and BL to also perform 8-bit instructions. It is used to store the value of the offset. Example:

MOV BL, [500] (BL = 500H)

The End

General purpose registers in 8086 microprocessor

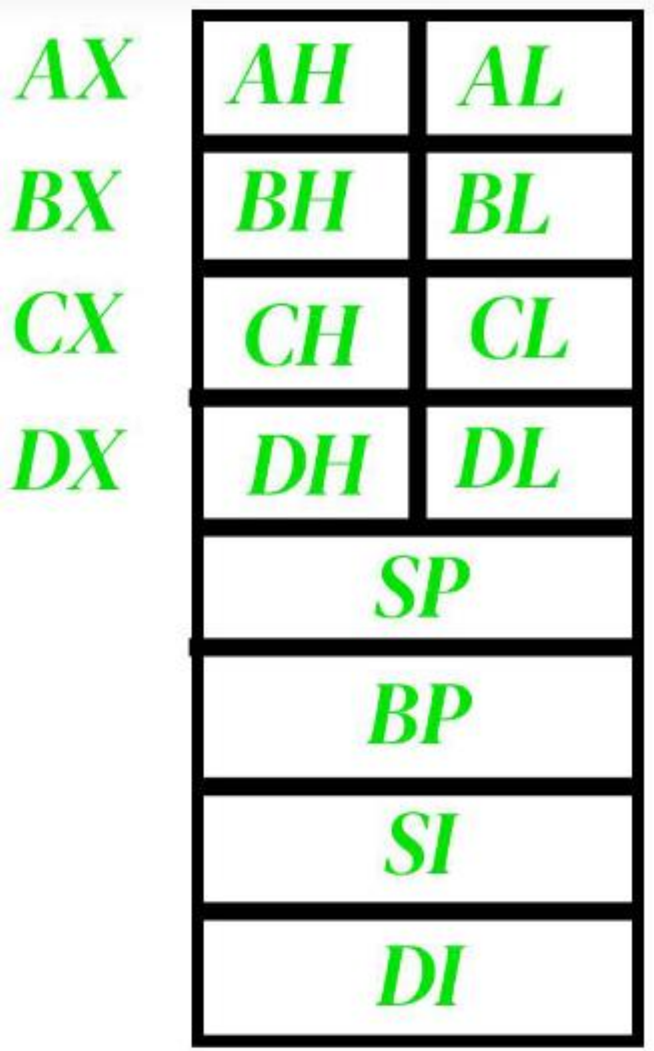
MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

GPR in 8086

- General-purpose registers are used to store temporary data within the microprocessor.
- There are 8 general-purpose registers in the 8086 microprocessor.



1. AX: This is the accumulator. It is of 16 bits and is divided into two 8-bit registers AH and AL to also perform 8-bit instructions. It is generally used for arithmetical and logical instructions but in 8086 microprocessor it is not mandatory to have an accumulator as the destination operand. Example:

ADD AX, AX (AX = AX + AX)

2. BX: This is the base register. It is of 16 bits and is divided into two 8-bit registers BH and BL to also perform 8-bit instructions. It is used to store the value of the offset. Example:

MOV BL, [500] (BL = 500H)

3. CX: This is the counter register. It is of 16 bits and is divided into two 8-bit registers CH and CL to also perform 8-bit instructions. It is used in looping and rotation. Example:

MOV CX, 0005

LOOP

4. DX: This is the data register. It is of 16 bits and is divided into two 8-bit registers DH and DL to also perform 8-bit instructions. It is used in the multiplication and input/output port addressing. Example:

MUL BX (DX, AX = AX * BX)

5. SP: This is the stack pointer. It is of 16 bits. It points to the topmost item of the stack. If the stack is empty the stack pointer will be (FFFE)H. Its offset address is relative to the stack segment.

6. BP – This is the base pointer. It is of 16 bits. It is primarily used in accessing parameters passed by the stack. Its offset address is relative to the stack segment.

7. SI – This is the source index register. It is of 16 bits. It is used in the pointer addressing of data and as a source in some string-related operations. Its offset is relative to the data segment.

8. DI – This is the destination index register. It is of 16 bits. It is used in the pointer addressing of data and as a destination in some string-related operations. Its offset is relative to the extra segment.

The End

Digital Arithmetic Circuits

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

Arithmetic Circuits

- **Arithmetic circuits** can perform seven different arithmetic operations using a single composite circuit.
- It uses a full adder (FA) to perform these operations.
- A multiplexer (MUX) is used to provide different inputs to the circuit in order to obtain different arithmetic operations as outputs.
- These circuits can be operated with binary values 0 and 1.

Binary Adder

- The most basic arithmetic operation is addition.
- The circuit, which performs the addition of two binary numbers is known as **Binary adder**.
- First, let us implement an adder, which performs the addition of two bits.

Half Adder

Half adder is a combinational circuit, which performs the addition of two binary numbers A and B are of **single bit**. It produces two outputs sum, S & carry, C.

- ✓ When we do the addition of two bits, the resultant sum can have the values ranging from 0 to 2 in decimal.
- ✓ We can represent the decimal digits 0 and 1 with single bit in binary.
- ✓ But, we can't represent decimal digit 2 with single bit in binary.
- ✓ So, we require two bits for representing it in binary.

The **Truth table** of Half adder is shown below.

Inputs		**Outputs**	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- The End

Digital Arithmetic Circuits

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

Full Adder

Full adder is a combinational circuit, which performs the **addition of three bits** A, B and C_{in}. Where, A & B are the two parallel significant bits and C_{in} is the carry bit, which is generated from previous stage. This Full adder also produces two outputs sum, S & carry, C_{out}, which are similar to Half adder.

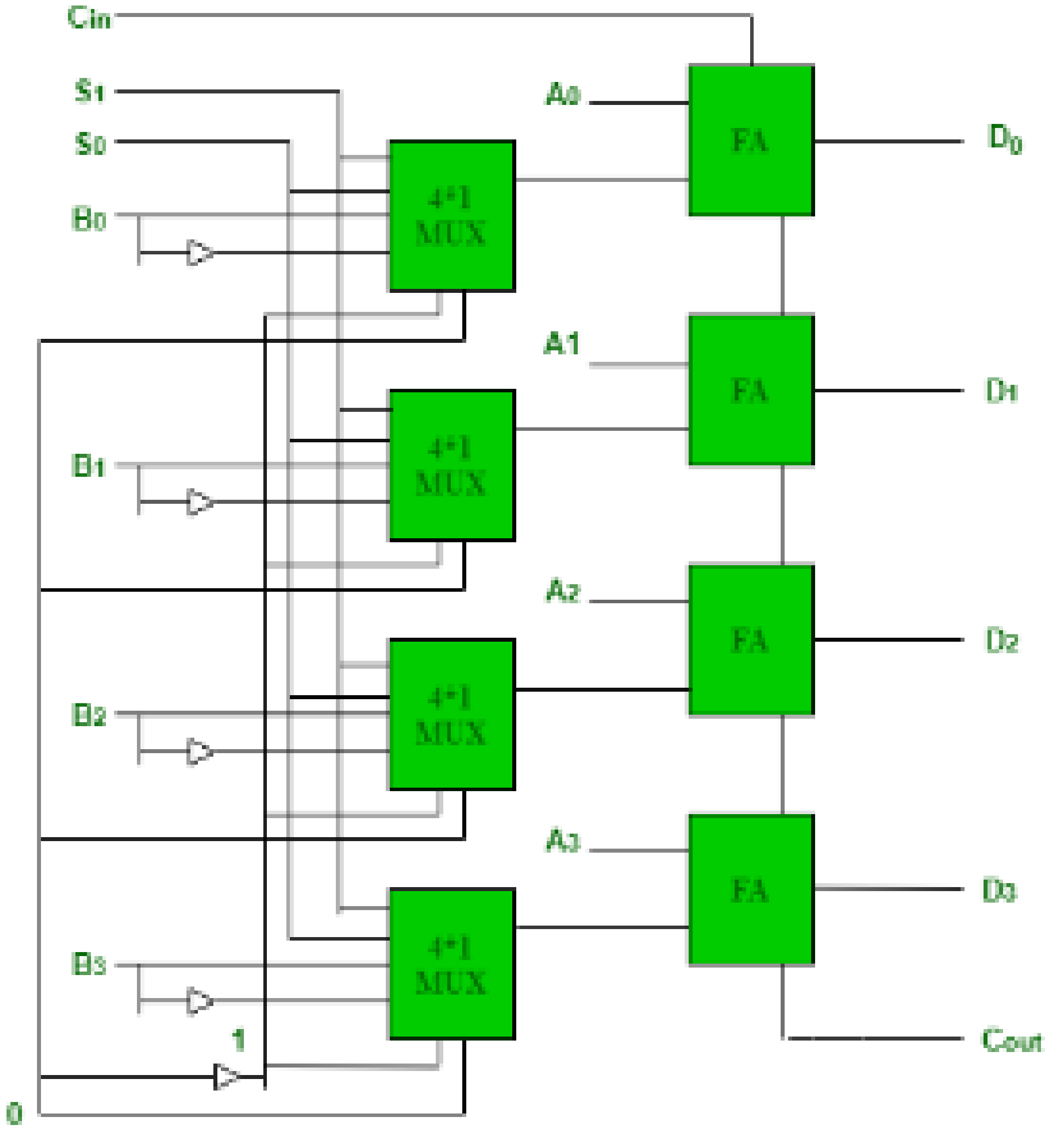
- ✓ When we do the addition of three bits, the resultant sum can have the values ranging from 0 to 3 in decimal.
- ✓ We can represent the decimal digits 0 and 1 with single bit in binary.
- ✓ But, we can't represent the decimal digits 2 and 3 with single bit in binary.
- ✓ So, we require two bits for representing those two decimal digits in binary.

The **Truth table** of Full adder is shown below.

Inputs			Outputs	
A	**B**	**Cin**	**Cout**	**S**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

4-bit Arithmetic Circuit

- Consider the following 4-bit Arithmetic circuit with inputs A and B.
- It can perform seven different arithmetic operations by varying the inputs of the multiplexer and the carry (C_0).



Truth Table for the above Arithmetic Circuit :

Hence, the different operations for the inputs A and B are –

1. A + B (adder)
2. A + B + 1
3. A + B’
4. A – B (subtracter)
5. A
6. A + 1 (incrementer)
7. A – 1 (decrementer)

S_0	S_1	C_0	MUX Output	Full Adder Output
0	0	0	B	A + B
0	0	1	B	A + B + 1
0	1	0	B'	A + B'
0	1	1	B'	A + B' + 1 = A – B
1	0	0	0	A
1	0	1	0	A + 1
1	1	0	1	A – 1
1	1	1	1	A – 1 + 1 = A

- The End

Control Unit in Computer Architecture

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

What is Design of Control Unit in Computer Architecture?

- A control unit drives the corresponding processing hardware by generating a set of signals that are in sync with the master clock.
- The two major operations performed by the control unit are instruction interpretation and instruction sequencing.
- The main function of a control unit is to fetch the data from the main memory, determine the devices and the operations involved with it, and produce control signals to execute the operations.

Types of Control Unit

- **Hardwired Control**
  - ✓ In the hardwired organization, the control logic is executed with gates, flip-flops, decoders, and other digital circuits.
  - ✓ It can be optimized to make a quick mode of operation.
  - ✓ In the micro-programmed organization, the control data is saved in the control memory.
  - ✓ The control memory is programmed to start the needed sequence of micro-operations.
  - ✓ A hardwired control requires changes in the wiring among the various elements if the design has to be modified or changed.

- The End

Control Unit in Computer Architecture

MD ABIR HOSSAIN
BSC IN COMPUTER SCIENCE & ENGINEERING, SONARGAON UNIVERSITY.

PART-TIME TEACHER, CST,

SIRAJGANJ POLYTECHNIC INSTITUTE, SIRAJGANJ

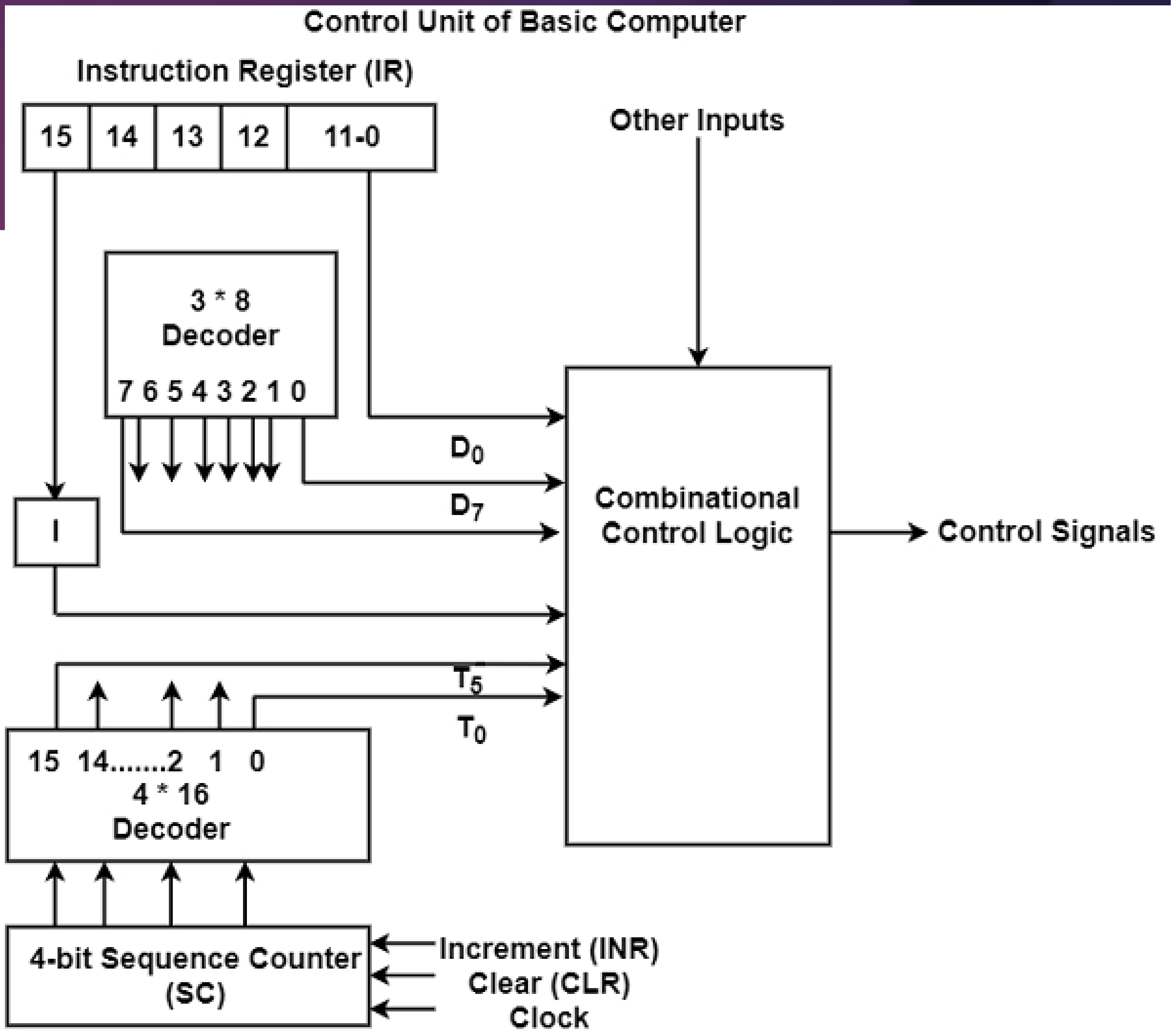
What is Design of Control Unit in Computer Architecture?

- A control unit drives the corresponding processing hardware by generating a set of signals that are in sync with the master clock.
- The two major operations performed by the control unit are instruction interpretation and instruction sequencing.
- The main function of a control unit is to fetch the data from the main memory, determine the devices and the operations involved with it, and produce control signals to execute the operations.

Types of Control Unit

- **Hardwired Control**
  - ✓ In the hardwired organization, the control logic is executed with gates, flip-flops, decoders, and other digital circuits.
  - ✓ It can be optimized to make a quick mode of operation.
  - ✓ In the micro-programmed organization, the control data is saved in the control memory.
  - ✓ The control memory is programmed to start the needed sequence of micro-operations.
  - ✓ A hardwired control requires changes in the wiring among the various elements if the design has to be modified or changed.

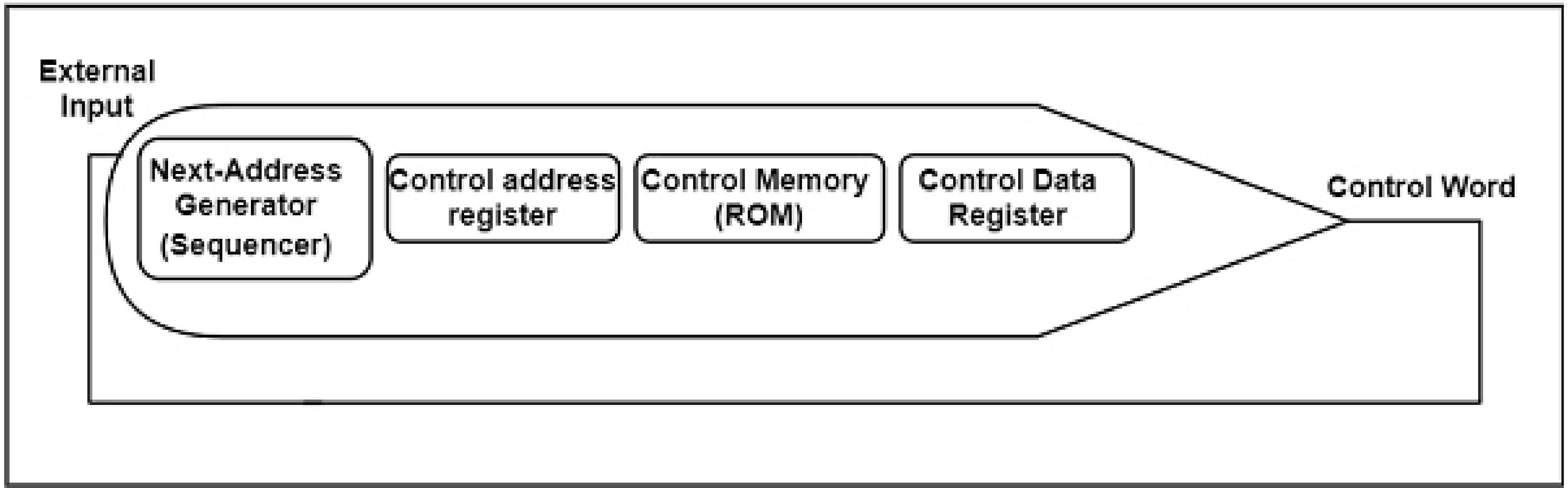
- The block diagram of the control unit is displayed in the figure.
- It includes two decoders, a sequence counter, and several control logic gates.
- Some instruction that is read from the memory is placed in the Instruction Register (IR).
- Therefore, the IR is divided into three elements such as I bit, opcode, and bits from 0 through 11.
- The opcodes are decoded with a 3 * 8 decoder whose outputs are indicated by symbols D_0 through D_7.



Microprogrammed Control

- The microprogrammed control stores its control data in the control memory. It can start the important set of micro-operations, the control memory is programmed. The changes and modifications in a micro-programmed control can be completed by upgrading the microprogram in the control memory.
- The figure displays the general configuration of a microprogrammed control organization.

Micro Programmed Control Organization



Microprogrammed Control

- The control is pretended to be a Read-Only Memory (ROM), where all the control data is saved permanently.
- ROM supports the address of the microinstruction.
- The other register is the control data register that stores the microinstruction that is read from the memory.
- It includes a control word that holds one or more microoperations for the data processor.
- The next address should be evaluated during this operation is done.
- It is evaluated in the next address generator.
- Therefore, it is transferred to the control address register to be read.
- The next address generator is referred to as the microprogram sequencer.
- It depends on the inputs to a sequencer, it decides the address of the next microinstruction.
- The microinstructions can be determined in different approaches.

- The End