ALAGAPPA UNIVERSITY

**[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category–I University by MHRD-UGC]
State University Established by the Government of Tamil Nadu)**

KARAIKUDI – 630 003

DIRECTORATE OF DISTANCE EDUCATION

M.Sc. PHYSICS

III -SEMESTER

34533

MICROPROCESSOR AND ELECTRONIC INSTRUMENTATION

Author
Dr. K. Sankaranarayanan, M.Sc., M.Phil., Ph.D.,
Professor
Department of Physics
Alagappa University
Karaikudi– 630003.
(Units: 1-7)

Author
Dr. M. Ramesh Prabhu, M.Sc, Ph.D.,
Assistant Professor
Department of Physics
Alagappa University
Karaikudi– 630003.
(Units: 8-14)

SYLLABI-BOOK MAPPING TABLE

MICROPROCESSOR AND ELECTRONIC INSTRUMENTATION

Syllabi	Mapping in Book
UNIT I 8085 ARCHITECTURE Introduction -Intel 8085 - Architecture of Intel 8085 -Timing Diagram -Timing Diagram for opcode fetch cycle of 8085 - Timing Diagram for memory read cycle -Timing diagram for memory write cycle - I/O read and I/O write -Instruction set - Instruction and data formats - Addressing modes of 8085 - Status flags - Instruction set classification - Data transfer - Arithmetic operations - Logical operations - Branching operations -Interrupts	**Pages 1 - 16**
UNIT II 8086 ARCHITECTURE Introduction - Architecture of 8086 - 8086 pin diagram - Pin description and functions - Register organization - Maximum mode function of 8086 - Minimum mode system	**Pages 17 - 27**
UNIT III 8085 INSTRUCTIONS Introduction to instructions for 8085 - Instruction set - Software development tools - Sample programs - Assembly language program for addition of two 8-bit numbers and store the sum - Assembly language program for 8-bit decimal subtraction and store the result - Assembly language program to shift an 8-bit number left by one bit - Assembly language program to find larger of two numbers	**Pages 28 - 32**
UNIT IV INTERRUPTS AND STACK OPERATIONS OF 8085 Introduction - interrupts in 8085 - Interrupt service subroutine (ISS) - Subroutine – Flowcharting – Loops - Pseudo-instructions -Stack operations -Programming and applications: traffic control system	**Pages 33-41**
UNIT V MICROCONTROLLER 8051 Introduction to 8 bit microcontrollers - 8051 microcontroller architecture - Features of 8051 microcontroller - Pin diagram of 8051 microcontroller - 8051 Memory organization - General purpose registers- Special Function Register- Oscillator and clock circuit- 8051 Addressing modes- 8051 Interrupts of 8051	**Pages 42 - 56**
UNIT VI 8051 - INSTRUCTION SET AND PROGRMMING Introduction to 8051 instruction set - Types of instructions - Data transfer instructions - Arithmetic instructions - Logical instructions - Boolean or bit manipulation instructions - Program branching instructions - Programming of 8051	**Pages 57 - 63**
UNIT VII - INTRODUCTION TO INTERFACING DEVICES Introduction - Address space partitioning- Memory mapped I/O	**Pages 64-73**

CONTENTS

UNIT – I 8085 ARCHITECTURE

Structure

1.1 INTRODUCTION

Microprocessor is a controlling unit of a micro-computer, fabricated on a small chip capable of performing ALU (Arithmetic Logical Unit) operations and communicating with the other devices connected to it. Microprocessor consists of an ALU, register array, and a control unit. ALU performs arithmetical and logical operations on the data received from the memory or an input device. Register array consists of registers identified by letters like B, C, D, E, H, L and accumulator (A). The control unit controls the flow of data and instructions within the microprocessor.8085 is pronounced as "eighty-eighty-five" microprocessor. It is an 8-bit microprocessor designed by Intel in 1977 using NMOS technology. It has the following configuration:

•8-bit data bus. One byte consists of 8- bits.
• 16-bit address bus, which can address upto 64KB
• A 16-bit program counter
• A 16-bit stack pointer
• Six 8-bit registers arranged in pairs: BC, DE, HLand can be used either individually or in pair
• Requires +5V supply to operate at 3.2 MHZ single phase clock
It is used in many applications in different areas and some of them are washing machines, microwave ovens, mobile phones, etc.

1.2 INTEL 8085

It is an 8 bit NMOS microprocessor and fabricated with forty pins IC(integrated circuit) package on a single LSI (Large scale Integration) chip. It uses a single +5 volt d.c(Direct Current) supply for its operation. It clock speed is 3 MHZ. It consists of 3 main architectural sections.

1-Arithmetic Logic Unit(ALU)

2-Timing and Control unit

3- Registers

Arithmetic Logic Unit:

It performs various arithmetic and logical operations like addition, subtraction, logical AND, logical OR, logical NOT, Increment and Decrement etc.

Timing and Control unit:

It generates timing and control signals which are necessary for the execution of the instructions. It also controls the peripherals and flow of data.

Registers:

Registers:- It is basically a collection of flip flops used to store a binary word. They are used by themicroprocessor or by the users for the temporary storage of data, manipulation of data and instructions.8085 has the following registers:

1-8 bit accumulator i.e. register A.

2- 6nos of 8 bits general purpose registers i.e. B,C,D,E,H,L.

3- 16 bit registers i.e. Stack Pointer (SP).

4-16 bit Program counter (PC),Status register, Temporary register and Instruction register.

The register A holds the operand during program execution.

The general purpose registers B,C,D,E,H,L are to handle either 8 or 16 bit data. Two 8 bit registers can be combined to handle 16 bit data. This is called register pair. Valid pairs of 8085 are B-C, D-E and H-L.The H-L pair is also used to address memory location denoted as M.

1.3 ARCHITECTURE OF INTEL 8085

Address bus

A15-A8, it carries the most significant 8-bits of memory/IO address. One byte consists of 8- bits.

Address-Data bus

AD7-AD0, it carries the least significant 8-bit address and data in Multiplexed mode.

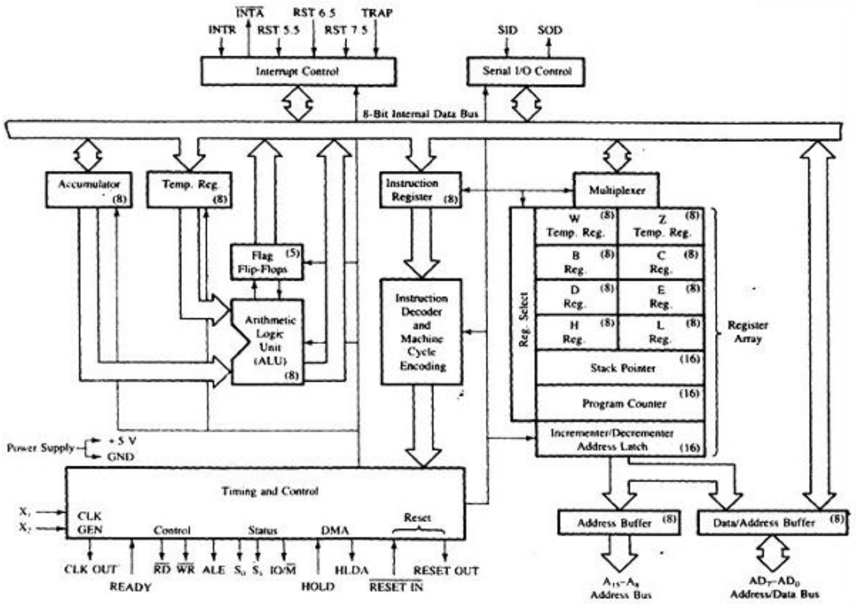


Figure 1.1 Architecture of Intel 8085

Control and Status Signals

These signals are used to identify the nature of operation going through the microprocessor. There are 3 control signal and 3 status signals.

Three control signals are RD, WR & ALE.

- **RD:** This signal indicates that the selected memory or IO device is to be read.

- **WR:** This signal indicates that the data on the bus is written into a selected memory or IO device.

- **ALE:** It is a positive going pulse generated when a new operation is started by the microprocessor. When the pulse goes high, it indicates address. When the pulse goes down it indicates data.

Three status signals are IO/M, S1& S0.

IO/M

This signal is used to differentiate between IO and Memory operations, i.e. when it is high; it indicates IO operation and when it is low then it indicates memory operation.

S1 & S0

These signals are used to identify the type of current operation. For example, if both S1 and S0 are 1, then it implies “ FETCH” operation and if both S1 and S0 are 0, then it meant for “HALT”.

Power supply

There are 2 power supply signals: VCC & VSS.

VCC indicates +5v power supply and VSS indicates ground signal.

Clock signals

There are 3 clock signals, i.e. X1, X2, CLK OUT.

- X1, X2: A crystal (RC, LC N/W) is connected at these two pins and is used to set frequency of the internal clock generator. This frequency is internally divided by 2.
- CLK OUT: This signal is used as the system clock for devices connected with the microprocessor.

Interrupts & externally initiated signals

Interrupts are the signals generated by the external devices to request the microprocessor to perform a task or to change the flow of execution of instructions. There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. A detailed discussionon interrupts follows in the interrupts section.

- INTA: It is an interrupt acknowledgement signal.

- RESET IN: This signal is used to reset the microprocessor by setting the program counter to zero.

- RESET OUT: This indicates that the CPU is being reset and this signal is also used to reset all the connected devices when the microprocessor is reset.

- READY: This signal indicates that the device is ready to send or receive data. If READY is low, then the CPU has to wait for READY to go high.

- HOLD: This signal indicates that another device is requesting the use of the address and data buses.

- HLDA (HOLD Acknowledge): It indicates that the CPU has received the HOLD request and it will relinquish the bus in the next clock cycle. HLDA is set to low after the HOLD signal is removed.

Serial I/O signals

There are 2 serial signals, i.e. SID and SOD.These signals are used for serial communication between the microprocessor and the peripherals.

- SOD (Serial output data line): The output SOD is set/reset as specified by the SIM instruction.

- SID (Serial input data line): The data on this line is loaded into the accumulator whenever a RIM instruction is executed.

Stack pointer-SP

Stack is a sequence of memory locations defined by the programmer in LIFO (Last In First Out) logic.That is last data to be placed in the stack is the first one to be removed.The stack pointer –SP register contains the address of the stack top. It indicates the address of the data filled-up location. The address of the next available location is SP+1.

Program counter-PC

It is the address of the next instruction to be executed.

Instruction register

It holds a copy of the current instruction until it is decoded. It is not accessed by the user.

Status register and Temporary register

It contains the status flags of 8085 microprocessor.

It is used to store intermediate results and for intermediate calculations.

1.4 TIMING DIAGRAM

It is a graphical representation of the steps which are carried out in a machine cycle for a particular operation of the microprocessor. It represents the pins (signals) involved in the operations and their status against each clock cycle.

1.4.1 TIMING DIAGRAM FOR OPCODE FETCH CYCLE OF 8085

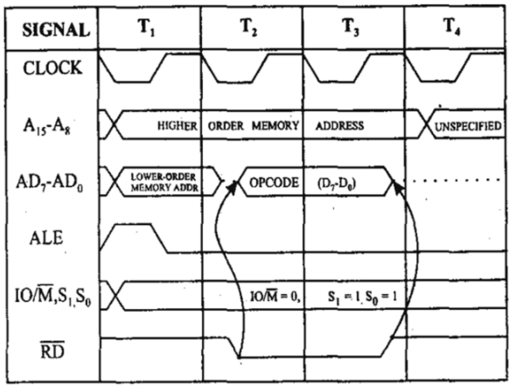


Figure 1.2 Timing Diagram for Opcode Fetch Machine Cycle

- Opcode is a command given to the microprocessor in a machine language. Each instruction of the processor has one byte opcode.
- The opcodes are stored in in-built memory. The processor executes the opcode fetch machine cycle to fetch the opcode from memory for every instruction.
- The time taken by the processor to execute the opcode fetch cycle is 4T.
- In this time, the first, 3 T-states are used for fetching the opcode from memory and the remaining T-states are used for internal operations by the processor.

Machine Cycle	Period/ Clock	Signal	Functions
M1	T1	IO/M	= 0 – Address in the A15-A8 (MSB) and AD7-AD0 (LSB) Bars meant for Memory Address.
		A15-A8	Carries 8 MSB bits
		AD7-AD0	Carries 8 LSB bits, it works in multiplexed mode.
		ALE	= 1 – Address Latch Enable helps to latch AD7-AD0 (8 – LSB) either in the memory or in an external latch in order to make AD7-AD0 available to transfer data in the next clock.
		S1 and S0	Both 1 meant for Fetch Operation
		RD	= 1 means no read operation.
	T2	IO/M	= 0 – Address in the A15-A8 (MSB) and AD7-AD0 (LSB) Bars meant for Memory Address
		A15-A8	Carries 8 MSB Address bits
		AD7-AD0	Carries 8 bits data, here opcode. Memory gets opcode from the register and place it on this bus.
		ALE	= 0
		S1 and S0	Both 1 meant for Fetch Operation
		RD	= 0 means Fetch operation. The data – opcode is placed in the data bus.
	T3	IO/M	= 0 – Address in the A15-A8 (MSB)
		A15-A8	Carries 8 MSB Address bits
		AD7-AD0	Carries 8 bits data, here opcode. It is placed in the instruction register (IR)
		ALE	= 0
		S1 and S0	Both 1 meant for Fetch Operation
		RD	= 1 means memory is disabled. The fetch operation is completed and the opcode is placed in the instruction register, IR.

	T4	IO/M	= remains 0
		A15-A8	No Data
		AD7-AD0	No Data
		ALE	= 0
		S1 and S0	Both 1 meant for Fetch Operation
		RD	= 1No Fetch Operation
			The opcode is decoded

1.4.2 TIMING DIAGRAM FOR MEMORY READ CYCLE

In a memory read cycle, the microprocessor reads the content of a memory location and the content is placed either in the accumulator or in any other CPU register. Ex-MVI A, 06H. For this instruction, the memory read cycle consumes 2 machine cycles, M1 and M2. In M1 the microprocessor completes the opcode fetch as described earlier and in M2 the microprocessor reads the content of a memory location. In the given example, the opcode is 3E and the data 06 will be placed in the register A. The timing diagram for Memory Read Cycle is given in the following figure.

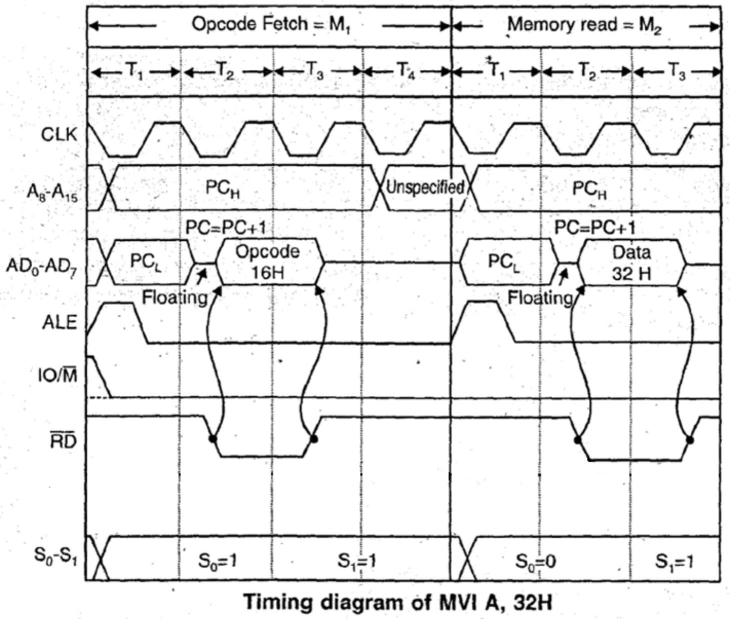


Figure 1.3 Timing Diagram for Memory Read

In the above figure, the fetching of opcode is completed at the end of the machine cycle M1. In M2, the following operations are taking place in the microprocessor.

If the length of the instruction is three bytes, it needs three machine cycles (M1, M2 and M3). For example, LXI D, 2500H instruction needs three machine cycles, to fetch the opcode, to read 8LSB of the data and to read 8MSB of the data. In the case of LDA 2500H, 4 machine cycles are needed to fetch the opcode, to read 8LSB of the address, to read 8MSB of the address and to transfer the content of the address to the Accumulator

Machine Cycle	Period/ Clock	Signal	Functions
M2	T1	IO/M	= 0 – Address in the A15-A8 (MSB) and AD7-AD0 (LSB) Bars meant for Memory Address.
		A15-A8	Carries 8 MSB bits
		AD7-AD0	Carries 8 LSB bits, it works in multiplexed mode.
		ALE	= 1 – Address Latch Enable helps to latch AD7-AD0 (8 – LSB) either in the memory or in an external latch in order to make AD7-AD0 available to transfer data in the next clock.
		S1 and S0	S1=1 and S0=0 meant for Read Operation
		RD	= 1 means no read operation.
	T2	IO/M	= 0 – Address/Data in the A15-A8 (MSB) and AD7-AD0 (LSB) Bars meant for Memory Address
		A15-A8	Carries 8 MSB Address bits
		AD7-AD0	Carries 8 bits data, here data. Memory gets opcode from the register and places it on this bus.
		ALE	= 0
		S1 and S0	S1=1 and S0=0 meant for Read Operation
		RD	= 0. It enables the memory for read operation and data is placed in the data bus.
	T3	IO/M	= 0 – Address in the A15-A8 (MSB)
		A15-A8	Carries 8 MSB Address bits
		AD7-AD0	Carries 8 bits data, it is placed in the CPU.
		ALE	= 0
		S1 and S0	S1=1 and S0=0 meant for Read Operation
		RD	= 1 means memory is disabled. The read operation is completed and the data is placed in the CPU.

1.4.3 TIMING DIAGRAM FOR MEMORY WRITE CYCLE

- In a memory write cycle the CPU sends data from the accumulator or any other register to memory.
- The status signal S0 and S1 are 1 and 0 respectively for write operation.
- The WR signal goes low in T2 indicating that the write operation is to be performed.
- During T2 the address/data bus is not disabledas in fetch or read operation but the data to be sent out to memory is placed on the address/data bus which works under multiplexed mode.
- In T3, the WR signal goes high indicating that the write operation is completed.

Example: - 1. MOV M, A and 2. STA 2400H

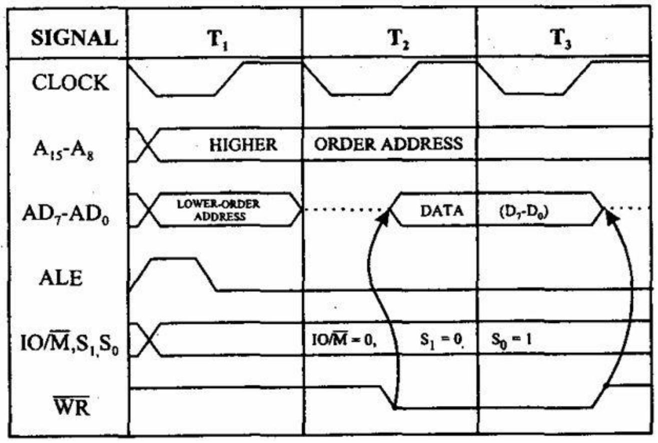


Figure 1.4 Timing Diagram for Memory Write cycle

1.5 I/O READ AND I/O WRITE

In I/O read cycle, the data from the I/O device or I/O port is placed in the accumulator. The memory read and the I/O read cycles are very similar except that the IO/M signal is high which means that the address in the A bus is for an Input device or port. The IN instruction can be used and it is of 2 bytes long and requires 3 machine cycles. They are to fetch the opcode, to read the address of the IO device or IO port and to read the data from IO device or IO port. In I/O write cycle, the microprocessor sends the data to IO device or port. As like I/O read cycle, the IO/M is high indicating the address in the bus is for the IO device or port. The OUT instruction is used to send the data from the microprocessor to the IO device or port by executing three machine cycle. First is to fetch the opcode, second is to read the address of the IO device or IO port and the third is to write the data in the IO device or IO port.

1.6 INSTRUCTION SET

8085 instruction set generally consists of the following types:

- Data transfer instructions.

- Arithmetic Instructions - add, subtract, increment and decrement.

- Logical Instructions - AND, OR, XOR and rotate.

- Branch Control Instructions –call, jump, return and restarts. Both conditional and unconditional jump, call and return instructions.

- In/Outinstructions and other instructions for setting/clearing flag bits, enabling/disabling interrupts, stack operations, etc.

1.6.1 INSTRUCTION AND DATA FORMATS

Intel 8085 is an 8-bit microprocessor and canhandles 8-bit data. All the memory locationsare designed to store 8-bit data. For, 16-bit data, consecutive memory locations are used. The address of memory location is of 16-bit i.e. 2 bytes.

The various techniques to specify data for instructions are:

(1) 8-bit or 16-bit data may be given directly or through the address of the memory location, I/O port or I/O device, where data resides, in the instruction itself.

(2) Instructions may specify with only one or two registers.Essentially, one of the operand is the accumulator and other operand is the content of the register.

The length of the instruction varies with the different ways of specifying data in the instruction.There are three types of instructions of Intel 8085 based on the length; (1)Single byte instruction, (2)2-byte instruction and (3)3-byte instruction.

Single-Byte instruction

The opcodeand the operands are in the instruction itself.These instructions are of one byte.

Ex-MOV A, C; Move the content of register C to A

79H is the opcode for MOV A, C. The binary form of opcode 78H is 01111001. The first twobits i.e. 01 for MOV operation; the next 3 bits i.e. 111 for register A and last 3 bits 001are for register C.

Two-Byte instruction

In case of two byte instruction the 1st byte of the instruction is opcode and 2nd byte is either data or address. Both bytes should be stored in two consecutive memory locations.

Ex-MVI C, 05; Move the data 05 to register C

MVI C,05 in the code form is 0E 05. The 1st byte i.e. 0E is the opcode for MVI C and 2nd byte i.e. 05 is the data which is to be moved to register C.

Three-Byte instruction

The three byte instruction has the 1st byte as opcode and 2nd and 3rd bytesare either 16-bit data or 16-bit address. They should be stored in three consecutive memory locations.

Ex- LDA 2500H, Load the accumulator with the content of the memory location, 2500H

In code form is 3A, 00, 25

The 1st byte, 3A is the opcode for LDA. The 2nd byte, 00 is 8 LSBs of addressand the 3rd byte, 25 is 8 MSBs ofaddress.

1.7 ADDRESSING MODES OF 8085

Operands are very essential for an instruction to perform operations. There are various ways to specify the data for instructions. The various formats for specifying the operands are called addressing modes and in 8085 the following addressing modes are available.

a) Direct addressing

b) Register addressing

c) Register indirect addressing

d) Immediate addressing

e) Implicit addressing

a) Direct addressing mode:

The address of the operand (data) is given in the instruction itself. For example, STA 2000H and IN 02H. In the first instruction, the data in the accumulator will be stored in the memory location, 2000H and the second instruction is used to read the data from the I/O port whose address is 02. The source of the data and the destination for the data are explicitly given in these instructions.

b) Register addressing mode:

In this addressing mode the operands are in the general purpose register. The opcode specifies the address of the operand (registers) in addition to the operation to be executed.

Ex: 1. MOV B, A and 2. ADD C. In the example 1, the content of the register A will be moved to the register B. The opcode for this operation is 47H. It specifies the operation as well as the operands. In the example 2, the content of the register C will be added to the content of the register A and the result will be stored in the register A. The opcode for this operation is 81H. Again, this opcode specifies the operands and the operation.

c) Register indirect addressing mode:

In register indirect addressing mode, the operand is in the memory location and the address of the memory location resides in H-L register pair, as denoted by the letter "M". M denotes the content of the address stored in the H-L register pair.

For example: LXI H, 2100H - Load the H-L register pair with the content of 2100H

MOV A,M – Move the content of 2100H to the accumulator

ADD C – Add the content of the register C with A, Now A=A+C

HLT – Halt

In the above example, the instruction MOV A, M lists under register indirect addressing mode.

d) Immediate addressing mode:

Under immediate addressing mode, the operand is specified within the instruction itself. Further, the mnemonics contains the letter "I" as the last letter and the data is explicitly present in the instruction.

Example: 1. MVI A,05H - Move immediate data 05H to Accumulator.

2. ORI 05H – Perform logical OR with data 05H and the content of the register A

e) Implicit addressing mode:

Instructions which operate on the content of the accumulator are listed under Implicit addressing mode. Here, the operand resides in the register A and do not require the address. RAL, RAR, CMA, CMC etc. are the examples.

1.8 STATUS FLAGS

In 8085, there are 5 status flags and each of the flag holds 1 bit. The value of the bit will be either 0 or 1. These flags are used to follow certain condition which arises during the arithmetic and logical operations which ultimately sets the value of individual flag. It is a set of 5 flip-flops. They are

i. Carry Flag(CS)

ii. Sign Flag(S)

iii. Zero Flag(Z)

iv. Parity Flag(P)

v. Auxiliary carry flag(AC)

Carry Flag

It holds carry out of the MSB resulting from the execution of an arithmetic operation.If there is a carry from addition ora borrow from subtraction or comparison, the carry flag is set to 1 otherwise 0.

Sign Flag

It is set to 1 if the MSB of the result of arithmetic or logical operation is 1 otherwise 0.

Zero Flag

It is set to 1 if the result of an arithmetic or logical operation is 0 otherwise 0 for non-zero result.

Parity Flag

The result of the operation contains even number of 1sset this flag to 1 otherwise 0 for odd number of 1s.

Auxiliary Carry Flag

It holds carry from bit 3 to 4 resulting from the execution of an arithmetic operation otherwise 0.

Program Status Word (PSW)

It is an 8-bit register out of which five bits indicate the 5 status flags and the remaining three bits are undefined. PSW and the accumulator treated as a 16 bit unit for stack operation.

1.9 INSTRUCTION SET CLASSIFICATION

An instruction is a binary pattern designed inside a microprocessor to perform a specific function. The entire group of instructions, called the instruction set, determines what functions the microprocessor can perform. These instructions can be classified into the following five functional categories: data transfer (copy) operations, arithmetic operations, logical operations, branching operations, and machine-control operations.

1.9.1 DATA TRANSFER

This group of instructions copy data from a location called a source to another location called a destination, without modifying the content of the source. In technical manual, the term data transfer is used for this copying function. However, the term transfer is misleading; it creates the impression that the content of the source are destroyed when, in fact, the content are retained without any modification. The various types of data transfer (copy) are listed below together with examples of each type:

Types	**Examples**
1. Between Registers.	1. Copy the content of the register B into register D. MOV D, B
2. Specific data byte to a register or a memory location.	2. Load register B with the data byte 32H. MVI B, 32H
3. Between a memory location and a register.	3. From a memory location 2000H to register B.MOV B, M
4. Between an I/O device and the accumulator.	4.From an input port to the accumulator. IN 02

1.9.2 ARITHMETIC OPERATIONS

These instructions perform arithmetic operations such as addition, subtraction, increment, and decrement.

Addition - Any 8-bit number, or the content of a register or the content of a memory location can be added to the content of the accumulator and the sum is stored in the accumulator. No two other 8-bit registers can be added directly (e.g., the content of register B cannot be added directly to the content of the register C). The instruction DAD is an exception; it adds 16-bit data directly in register pairs.

Subtraction - Any 8-bit number, or the content of a register, or the content of a memory location can be subtracted from the content of the accumulator and the result is stored in the accumulator. The subtraction is performed in 2's compliment, and the results if negative, are expressed in 2's complement. No two other registers can be subtracted directly.

Increment/Decrement - The 8-bit content of a register or a memory location can be incremented or decremented by 1. Similarly, the 16-bit content of a register pair (such as BC) can be incremented or decremented by 1. These increment and decrement operations differ from addition and subtraction in an important way; i.e., they can be performed in any one of the registers or in a memory locations.

1.9.3 LOGICAL OPERATIONS

These instructions perform various logical operations with the content of the accumulator.

AND, OR Exclusive-OR - Any 8-bit number, or the content of a register, or of a memory location can be logically ANDed, ORed, or Exclusive-ORed with the content of the accumulator. The result is stored in the accumulator.

Rotate- Each bit in the accumulator can be shifted either left or right to the next position. **Compare**- Any 8-bit number or the content of a register, or a memory location can be compared for equality, greater than, or less than, with the content of the accumulator.

Complement - The content of the accumulator can be complemented. All 0s are replaced by 1s and all 1s are replaced by 0s.

1.9.4 BRANCHING OPERATIONS

This group of instructions alters the sequence of program execution either conditionally or unconditionally.

Jump - Conditional jumps are an important aspect of the decision-making process in the programming. These instructions test for a certain conditions (e.g., Zero or Carry flag) and alter the program sequence when the condition is met. In addition, unconditional jump is also available.

Call, Return, and Restart - These instructions change the sequence of a program either by calling a subroutine or returning from a subroutine. The conditional Call and Return instructions also can test condition flags.

1.9.5 INTERRUPTS

- The processor has 5 interrupts. They are presented below in the order of their priority (from lowest to highest):

- INTR is an interrupt request signal, maskable and 8080A compatible interrupt. When the interrupt occurs the processor completes the instruction at hand and goes to the CALL instruction. INTA will be sent by the microprocessor after receiving the INTR.

- One of the 8 RST instructions (RST0 - RST7). The processor saves current program counter into stack and branches to memory location N * 8 (where N is a 3-bit number from 0 to 7 supplied with the RST instruction).

- CALL instruction (3 byte instruction). The processor calls the subroutine whose addressis specified in the second and third bytes of the instruction.

- RST5.5 is a maskable interrupt. When this interrupt is received the processor saves the content of the PC register into stack and branches to 2CH address.

- RST6.5 is a maskable interrupt. When this interrupt is received the processor saves the content of the PC register into stack and branches to 34H address.

- RST7.5 is a maskable interrupt. When this interrupt is received the processor saves the content of the PC register into stack and branches to 3CH address.

- TRAP is a non-maskable interrupt. When this interrupt is received the processor saves the content of the PC register into stack and branches to 24H address.

- All maskable interrupts can be enabled or disabled using EI and DI instructions. RST 5.5, RST6.5 and RST7.5 interrupts can be enabled or disabled individually using SIM instruction.

UNIT – II 8086 ARCHITECTURE

Structure

2.1 INTRODUCTION

- A microprocessor is an Integrated Circuit with all the functions of a CPU however; it cannot be used stand-alone since unlike a microcontroller it has no memory or peripherals.8086 does not have a RAM or ROM inside it. It is a 16-bit Microprocessor (µp).
- It's ALU, internal registers works with 16bit binary word. 8086 has a 20 bit address bus and can access up to 2^{20}= 1 MB memory locations. 8086 has a 16bit data bus. It can read or write data to a memory/port either 16bit or 8 bit at a time. It can support up to 64K I/O ports. It provides 14 numbers of 16 -bit registers.
- Clock frequency range of 8086 is 6-10 MHz. It has multiplexed address and data bus AD0- AD15 and an address bus A16 – A19. It requires single phase clock with 33% duty cycle to provide internal timing.
- It can prefetch up to 6 instruction bytes from memory and queues them in order to speed up the instruction execution. It requires +5V power supply and built in a 40 pin dual in line package.
- 8086 is designed to operate in two modes, Minimum mode and Maximum mode. The minimum mode is selected by applying logic 1 to the MN / $\overline{MX}$ input pin.
- This is a single microprocessor configuration. The maximum mode is selected by applying logic 0 to the MN / $\overline{MX}$ input pin. This is capable of cascading with multi microprocessors configuration.

2.2 ARCHITECTURE OF 8086

The following figure represents the internal architecture of 8086 which represents the register and bus organization.

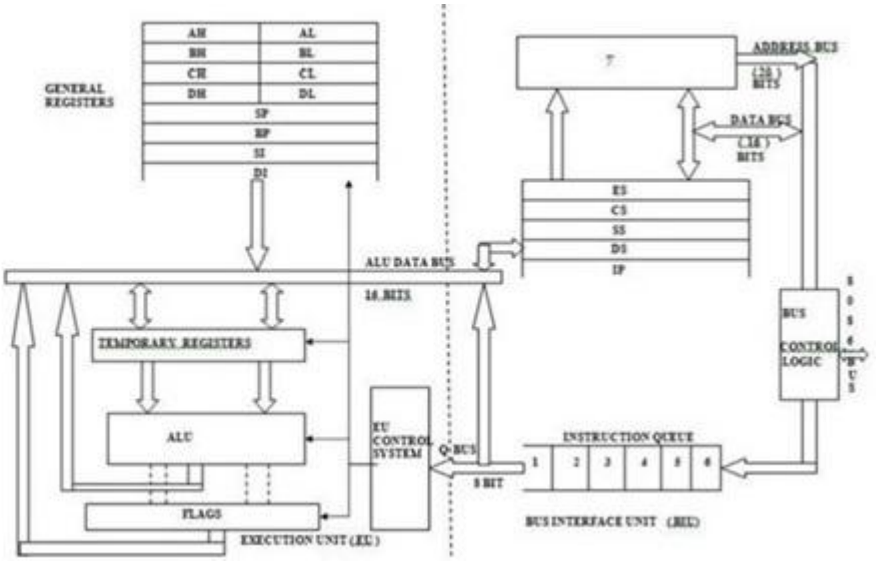


Figure 2.1 Architecture of 8086

2.3 8086 PIN DIAGRAM

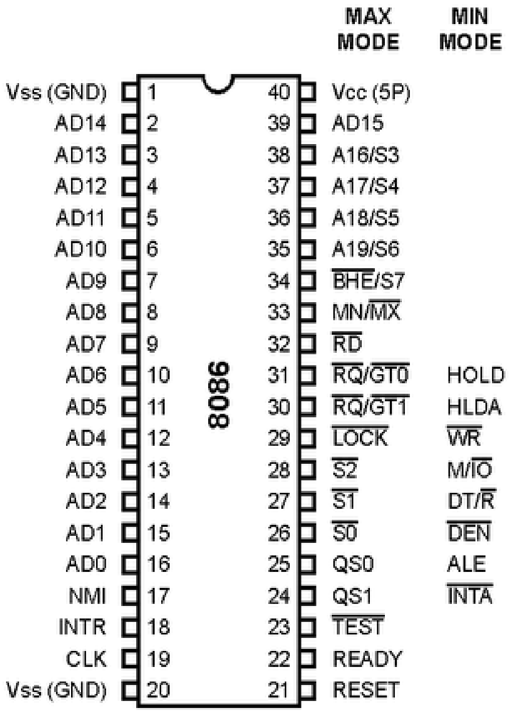


Figure 2.2 Pin diagram of 8086

2.3.1 PIN DESCRIPTIONAND FUNCTIONS

Power supply and frequency signals

It uses 5V DC supply at V_{CC} pin 40, and uses ground at V_{SS} pin 1 and 20 for its operation.

Clock signal

Clock signal is provided through pin19. It provides timing to the processor for operations. Its clock frequency is different for different versions, i.e. 5MHz, 8MHz and 10MHz.

Address/data bus and Address/status bus

Pins AD0-AD15are 16 bit address/data bus. AD0-AD7 carries low order byte data and AD8-AD15 carries higher order byte data. During the first clock cycle, it carries 16-bit address and after that it carries 16-bit data once low order byte is latched in the memory.

Pins A16-A19/S3-S6arethe 4 bit address/status bus. During the first clock cycle, it carries 4-bit address and later it carries status signals.

S7/$\overline{BHE}$ and $\overline{RD}$

BHE stands for Bus High Enable. It is available at pin 34 and used to indicate the transfer of data using data bus D8-D15. This signal is low during the first clock cycle, thereafter it is active. The status signal S7 is available during T2, T3 and T4.

It is available at pin 32 and its value is 0 during Read operation.

READY and RESET

It is available at pin 22. It is an acknowledgement signal from I/O devices to the microprocessor for data transfer. It is an active high signal. When it is high, it indicates that the device is ready to transfer data. When it is low, it indicates wait state.

It is available at pin 21 and is used to restart the execution. It causes the processor to immediately terminate its present activity. This signal is active high for the first 4 clock cycles to RESET the microprocessor.

INTR and INTA

It is available at pin 18. It is an interrupt request signal from the I/O devices, which is sampled during the last clock cycle of each instruction to determine if there is any interrupt request or not.

It is an interrupt acknowledgement signal and is available at pin 24. When the microprocessor receives this signal, it acknowledges the interrupt by raising this signal to 1.

NMI

It stands for non-maskable interrupt and is available at pin 17. It is an edge triggered input, which causes an interrupt request to the microprocessor.

$\overline{TEST}$

This signal is like wait state and is available at pin 23. When this signal is high, then the processor has to wait in IDLE state, else the execution continues.

MN/$\overline{MX}$

It stands for Minimum/Maximum mode operation and is available at pin 33. It dictatesthe mode of operation of the processor; when it is high, it works in the minimum mode else 0 for maxi mode.

ALE

It stands for address latch enable and is available at pin 25. A positive pulse is generated each time the processor begins any operation. This signal indicates the availability of a valid address on the address/data bus.

$\overline{DEN}$andDT/$\overline{R}$

It stands for Data Enable and is available at pin 26. It is used to enable Trans-receiver 8286. The trans-receiver is a device used to separate data from the address/data bus.

It stands for Data Transmit/Receive signal and is available at pin 27. It decides the direction of data flow through the trans-receiver. When it is high, data is transmitted out and vice-a-versa.

M/$\overline{IO}$

This signal is used to distinguish between memory and I/O operations. When it is high, it indicates memory access operation else 0 for I/O access operation. It is available at pin 28.

QS_1 and QS_0

These are queue status signals and are available at pin 24 and 25. These signals provide the status of instruction queue. Their conditions are shown in the following table −

QS_0	QS_1	Status
0	0	No operation
0	1	First byte of opcode from the queue
1	0	Empty the queue
1	1	Subsequent byte from the queue

$\overline{WR}$

It stands for write signal and is available at pin 29. It is used to write the data into the memory or to the output device depending on the status of M/$\overline{IO}$ signal.

LOCK

When this signal is active, it indicates to the other processors not to ask the CPU to leave the system bus. It is activated low using the LOCK prefix on any instruction and is available at pin 29.

RQ/GT$_1$, HOLD and RQ/GT$_0$, HLDA

These are the Request/Grant signals (Pins 30 and 31) used by the other processors requesting the CPU to release the system bus. When the signal is received by CPU, then it sends acknowledgment. RQ/GT$_0$ has a higher priority than RQ/GT$_1$.

HLDA

It stands for Hold Acknowledgement signal and is available at pin 30. This signal acknowledges the HOLD signal.

HOLD

This active HIGH signal indicates the processor that an external device is requesting to access the address/data buses. It is available at pin 31.

S$_0$, S$_1$, S$_2$

These are the status signals that provide the status of operation, which is used by the Bus Controller 8288 to generate memory & I/O control signals. These are available at pin 26, 27, and 28. Following is the table showing their status −

S$_2$	S$_1$	S$_0$	Status
0	0	0	Interrupt acknowledgement
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read

1	1	0	Memory write
1	1	1	Passive

2.4 REGISTER ORGANIZATION

The registers in 8086 are grouped as follows;

- ❖ Pointers and index registers
- ❖ Segment registers
- ❖ Instruction Pointer and Status Flags
- ❖ General data registers.

Intel 8086 uses 20 bit address lines. So its memory capacity is (2^{20}) = 1MB of memory. Memory segmentation is first introduced in this type of microprocessor. All the registers of 8086 are 16-bit registers. The general purpose registers can be used as either 8-bit register or 16-bit register. The register set of 8086 can be categorized into 4 different groups.

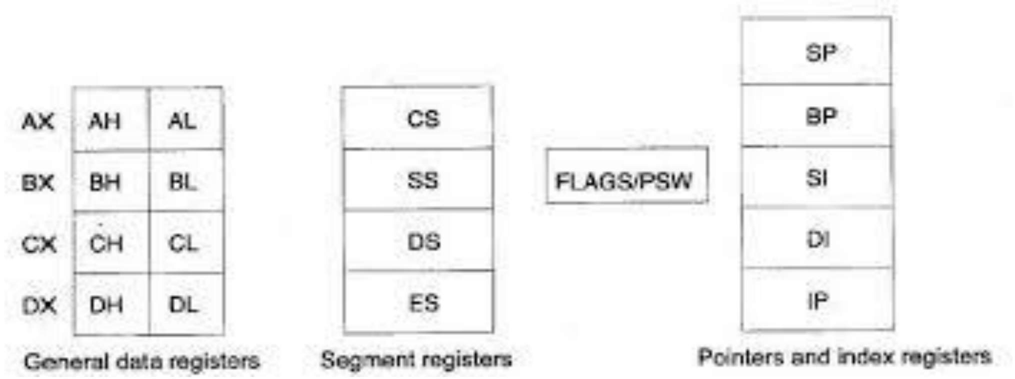


Figure 2.3 Organizations of Registrar

General data registers

- ▪ The registers AX, BX,CX and DX are the general purpose 16-bit registers. AX is used as 16-bit accumulator. Accumulator can be used for I/O operations, rotate and string manipulation.
- ▪ The lower 8-bit is designated as AL (accumulator) and higher 8-bit is designated as AH. BX is a 16 bit register mainly used as base register for memory address calculation, BL indicates the lower 8-bit and BH indicates the higher 8-bit of BX.
- ▪ Register CX is used as default counter (looping) and DX is used to hold I/O address during certain I/O instructions. Ex: If the result of multiplication is more than 16 bits the low order bits are stored in AX, the high-order 16 bits are stored in DX register

Segment Registers

- Intel 8086 is byte organized. Here 1 MB of memory is divided into 16 logical segments. Each segment contains 64 kilobytes of memory. There are four segment register in 8086; Code segment register (CS), Data segment register (DS), Extra segment register (ES) and Stack segment register (SS). Code segment register (CS) points out the starting address of the code segment memory, where the executable program is stored.
- Data segment register (DS): points the starting address of the data segment of the memory where the data is stored. Extra Segment Register (ES) also points the starting address of the extra segment memory locations. These registers act as base registers.

Stack Segment Register (SS)

By default, the processor assumes that all data referenced by the stack pointer (SP) and base pointer (BP) registers is located in the stack segment. SS register can be changed directly using POP instruction. It is used for addressing stack segment of memory and is used to store stack data. The starting address of the each segment can be obtained from CS, DS, SS and ES registers.

Pointers and Index Registers

The pointers IP, BP, SP usually contain offsets within the code, data and stack segments respectively.

• Stack Pointer (SP) is a 16-bit register pointing to program stack in stack segment.

•Base Pointer (BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based indexed or register indirect addressing. The index registers are particularly useful for string manipulation. SI is used to store the offset of source data in data segment. DI is used to store the offset of destination in data or extra segment.

Status Register

It determines the current state of the processor. They are modified automatically by CPU after mathematical operations. This allows to determine the type of the result and to determine the conditions to transfer the flow of execution to other parts of the program.

Carry flag (CY)

It is set whenever there is a carry or borrow out of the MSB of a result. CY=1, Carry is generated CY=0, Carry is not generated.

Parity flag (PF):It is set if the result has even parity of 1s. If parity is odd, PF is reset.

Auxiliary carry flag (AC): Holds a carry after addition or a borrow after subtraction between bit 3 and 4 of the result.

Zero flag (ZF): Indicates the result of an arithmetic or logic operation. It is set if the result is zero

Sign flag (SF): Indicates the arithmetic sign of the result after an addition or subtraction. If S = 1, the result is negative. If S = 0, the result is positive.

Overflow flag (OF): An overflow condition indicates that a result has exceeded the capacity of the machine when a condition that can occur when signed numbers are added or subtracted.

ControlFlags

Trap flag (TF): Debugging feature of the microprocessor by setting single step mode of execution in the microprocessor. In single step mode the 8086 executes a software interrupt. The interrupt vector table contains certain locations reserved for single step mode.

Interrupt flag (IF) : interrupt controls operation. If I = 1, the INTR pin is enabled. If I = 0, the INTR pin is disabled.

Direction flag (DF) Controls the selection of increment and decrement for the DI and SI registers during string instructions. If DF=1, the string bytes are accessed from low memory address to high memory address. The contents of SI and DI are decremented. If DF=0, the string bytes are accessed from high memory address to low memory address. The contents of SI and DI are incremented.

2.5 MAXIMUM MODE FUNCTION OF 8086

In the maximum mode, the 8086 is operated by strapping the MN/$\overline{MX}$ pin to

ground.

- ❖ In this mode, the processor derives the status signal S2, S1, S0. Another chip called bus controller derives the control signal using this status information. In the maximum mode, there may be more than one microprocessor in the system configuration. The components in the system are same as in the minimum mode system.
- ❖ The basic function of the bus controller chip IC8288 is to derive control signals like $\overline{RD}$ and $\overline{WR}$ (for memory and I/O devices), $\overline{DEN}$,

  DT/$\overline{R}$, ALE etc. using the information by the processor on the status

  lines.
- ❖ The bus controller chip has input lines S2, S1, S0 and CLK. These inputs to 8288 are driven by CPU. It derives the outputs ALE, $\overline{DEN}$,

  DT/$\overline{R}$, $\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$,$\overline{IORC}$, $\overline{IOWC}$ and $\overline{AIOWC}$.

- Maximum mode of operation AEN and IOB are generally grounded. DEN pin is usually tied to +5V.
- INTA pin used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device. $\overline{IORC}$, $\overline{IOWC}$are I/O

  read command and I/O write command signals respectively.
- These signals enable an IO interface to read or write the data from or to the address port. The $\overline{MRDC}$, $\overline{MWTC}$ are memory read command

  and memory write command signals respectively and may be used as memory read or write signals.
- Here the only difference in timing diagram between minimum mode and maximum mode is the status signals used and the available control and advanced command signals. R0, S1, S2 are set at the beginning of bus cycle.
- When the 8086 is set for the maximum-mode configuration, it provides signals for implementing a multiprocessor / coprocessor system environment.
- These are known as local or private resources. Coprocessor also means that there is a second processor in the system. In this, two processorsdo not access the bus at the same time.
- One passes the control of the system bus to the other and then may suspend its operation. In the maximum-mode 8086 system, facilities are provided for implementing allocation of global resources and passing bus control to other microprocessor or coprocessor.

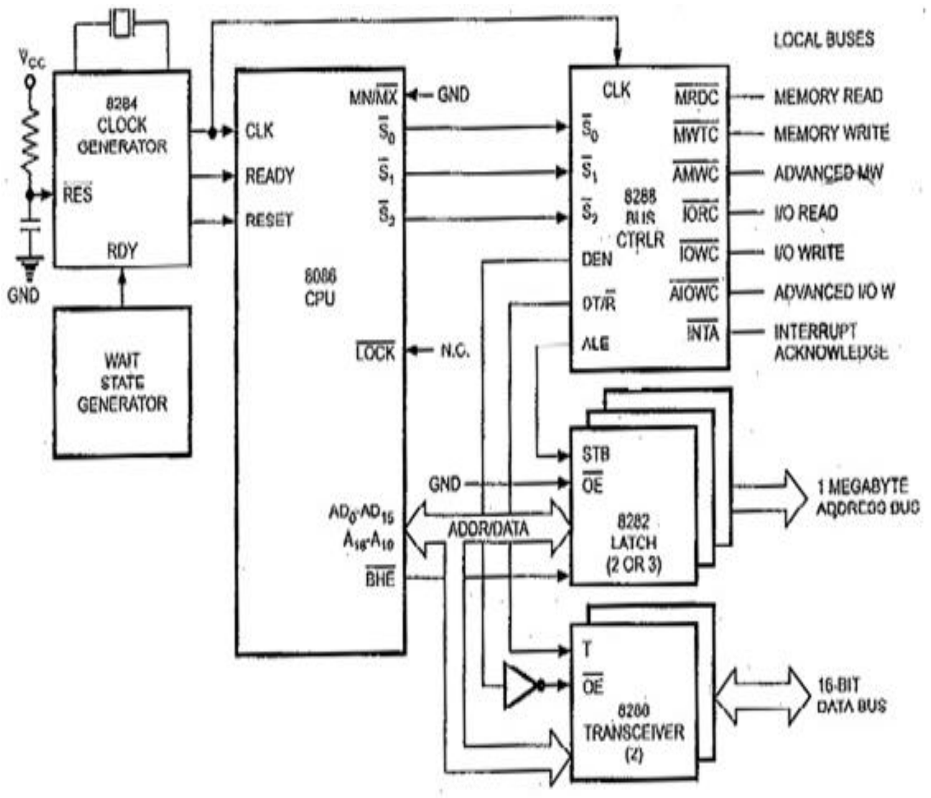


Figure 2.4 Maximum mode functions of 8086

2.6 MINIMUM MODE SYSTEM

- Microprocessor 8086 is operated in minimum mode by strapping its MN/$\overline{MX}$ pin to logic1.
- In this mode, all the control signals are given out by the microprocessor itself. There is a single microprocessor in the minimum mode system.The remaining components in the system are latches, Transceivers, clock generator, memory and I/O devices.
- Latches are generally buffered output of D-type flip-flops like 74LS373 or 8282.They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal
- In minimum mode 8086 configuration, transceiver is the bidirectional buffers and they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address/data signals
- They are controlled by two signals namely, $\overline{DEN}$and DT/$\overline{R}$.The $\overline{DEN}$signal indicates the direction of data, i.e. from or to the processor.
- The opcode fetch and read cycles are similar. Hence the timing diagram can be categorized in two parts, the first is the timing diagram for read cycle and the second is the timing diagram for write cycle.

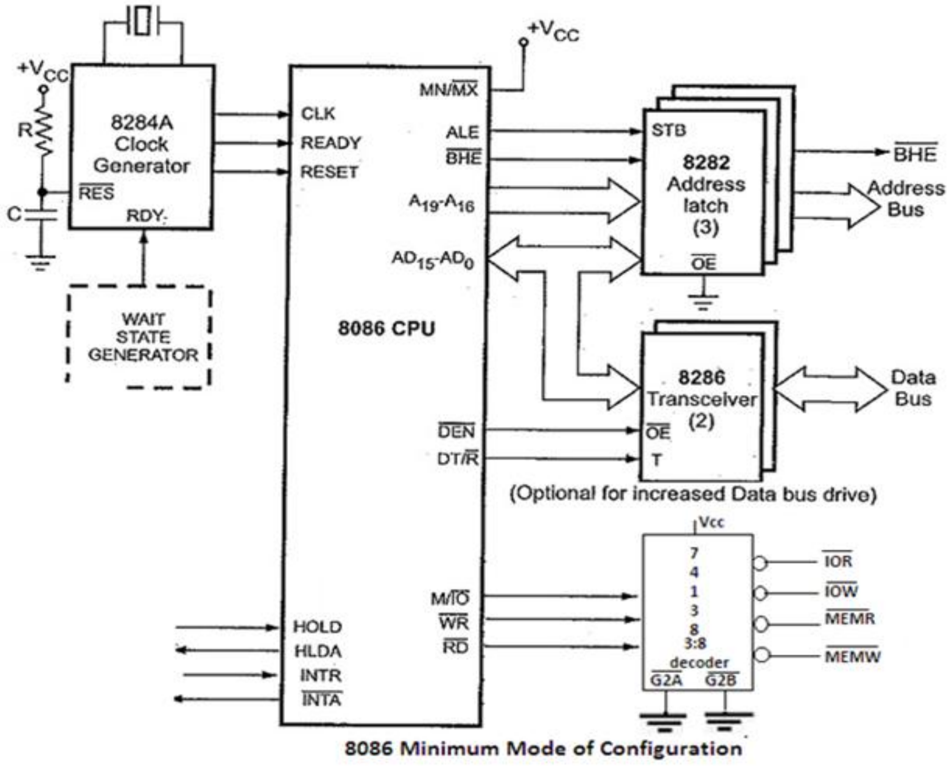


Figure 2.5Minimum mode functions of 8086

The $\overline{BHE}$and AD0 signals are used to select the proper byte or bytes of memory orI/O word to be read or write.The M/$\overline{IO}$, $\overline{RD}$ and $\overline{WR}$signals indicate the type of data transfer.

UNIT –III INSTRUCTIONS (8085)

Structure

3.1 INTRODUCTION TO INSTRUCTIONS FOR 8085

- An instruction is a binary pattern designed inside a microprocessor to perform a specific function.
- The entire group of instructions that a microprocessor supports is called Instruction Set.8085 has 246 instructions.
- Each instruction is represented by an 8-bit binary value.These 8-bits of binary value is called Op-Code or Instruction Byte.

An instruction determines what specified operation the microprocessor can perform. These instructions can be classified into the following five functional categories: data transfer (copy) operations, arithmetic operations, logical operations, branching operations, and machine-control operations.

3.2 INSTRUCTION SET

• 8085 instruction set consists of the following instructions:

• **Data moving instructions.** Some of the examples are;

MOV r1, r2 – The content of the register r2 will be copied in the register r1

MOV r, M – The content of the memory location whose address is in H-L pair will be copied to the register r and

MVI r, data – The data will be copied in the register r

• **Arithmetic operations**- add, subtract, increment and decrement.

ADD M – Add the content of the memory location whose address is in H-L pair will be added with accumulator

SUB r – Subtract the content of the register from accumulator

• **Logical operations** - AND, OR, XOR and rotate.

• **Control transfer** - conditional, unconditional, call subroutine, return from subroutine and restarts.

• **Input/Output instructions**. Examples, IN and OUT

• **Other** - setting/clearing flag bits, enabling/disabling interrupts, stack operations, etc.

3.3 SOFTWARE DEVELOPMENT TOOLS

Software can be developed based on high-level language and assembly language. Software developed out of high-level language can run on any computer or microcomputer having standard complier for that language. It is not computer oriented and is recommended for large programs where large volume of data is to be handled. The assembly language is machine oriented and is recommended for small to moderate programs where small volume of data is to be handled.

A programming tool or software development tool is a computer program that software developers use to create, debug, maintain, or otherwise support other programs and applications. The term usually refers to relatively simple programs, that can be combined together to accomplish a task. The most basic tools are a source code editor and a compiler or interpreter, which are used ubiquitously and continuously. Other tools are used more or less depending on the language, development methodology, and individual engineer, and are often used for a discrete task, like a debugger or profiler.

Assembly language program

Simple Steps to write an assembly language program are

- Analyze the problem
- Develop program logic
- Write an algorithm
- Make flow chart
- Select appropriate instructions and write using Assembly language of 8085

3.4 SAMPLE PROGRAMS

3.4.1 ASSEMBLY LANGUAGE PROGRAM FOR ADDITION OF TWO 8-BIT NUMBERS AND STORE THE SUM

Mnemonics	Operand	Comments
LXI	H, 2500 H	Get 1st No. in H-L pair whose address is 2500
MOV	A, M	Copy the 1st no. in accumulator
INX	H	Increment the H-L pair, now 2501
ADD	M	Add 2nd no. with Accumulator
STA	2502 H	Store the sum in 2502 H.
HLT		Stop the program.

DATA

2500- 03 H
2501- 05 H

RESULT : The sum 08H is stored in memory location 2502 H.

3.4.2 ASSEMBLY LANGUAGE PROGRAM FOR 8-BIT DECIMAL SUBTRACTION AND STORE THE RESULT

Mnemonics	Operand	Comments
LXI	H, 2502 H	Get 2nd no. in H-L pair.
MVI	A, 99	Load 99 in accumulator.
SUB	M	9’s compliment of 2nd no.
INR	A	10’s compliment of 2nd no.
DCX	H	Get address of 1st no.
ADD	M	Add 1st no. & 10’s compliment of 2nd no.
DAA		Decimal Adjustment.
STA	2503 H	Store result in 2503 H.
HLT		Stop the program.

DATA

2501- 96 H.

2502- 38 H.

The result is stored in memory location 2503 H.

RESULT

2503- 58 H.

3.4.3 ASSEMBLY LANGUAGE PROGRAM TO SHIFT AN 8-BIT NUMBER LEFT BY ONE BIT

Shift the data 65H left by one bit which is stored in memory 2501H.
The binary representation of 65H is given below:
65=0110 0101
Result of shifting65H left by one bit =1100 1010=CA

To shift a number left by one bit the number is added to itself. If 65 is added to 65, the result isCA as shown below.
65H =0110 0101 + 65H =0110 0101 = 1100 1010=CA

The result is to be stored in memory 2502H.

PROGRAM

Memory Address	Machine Codes	Mnemonics	Operands	Comments
2000	3A,01, 25	LDA	2501H	Get data in accumulator
2003	87	ADD	A	Shift it left by one bit
2004	32,02, 25	STA	2502H	Store result in 2502H
2007	76	HLT		Halt

DATA
2501-65 H

Result
2502-CA H

3.4.4 ASSEMBLY LANGUAGE PROGRAM TO FIND LARGER OF TWO NUMBERS

Example: Find the larger of 98H and 87H.

The first number 98H is placed in the memory location 2501H.
The 2nd number 87H is placed in the memory location 2502H

The result is to be stored in the memory location 2503H.

PROGRAM

```
Memory Machine Mnemonics Operands Comments
addr   code

 2000  21,01,25LXI   H,2501HGet 1st no in H-L pair
 2003  7E MOV          A,M  Move 1st no to accumulator
 2004  23       INX    H        Get  2nd no in H-Lpair
 2005  BE       CMP      M    Compare 2nd no with 1st no
                                               Is the 2nd no >1st no?
 2006  D2,0A,20        JNC AHEAD No, Goto AHEAD
 2009  7E MOV          A,M    Yes   get   2nd   no.   in   the   acc200A
       32,03,25 AHEAD STA 2503H Store larger number in
                                                2503H
 200D  76                  HLT                           STOP
```

DATA:
2501-98H
2502-87H
Result is 98H and it is stored in memory location 2503H

RESULT:
2503-98H

UNIT –IV INTERRUPTS AND STACK OPERATIONS OF 8085

Structure

4.1 INTRODUCTION - INTERRUPTS IN 8085

Interrupts are the signals generated by the external devices to request the microprocessor to perform a task. There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. The TRAP is the highest priority followed by RST 7.5, RST 6.5, RST 5.5 and INTR (lowest priority).

When microprocessor receives any interrupt signal from the peripheral or an external device which is requesting its services, it completes the execution of the current instruction and store the content of the program counter on the stack. It sends an acknowledgement (INTA) to the peripheral which is requesting for its service.

It also resets the interrupt enable flip-flop before executing the Interrupt Service Subroutine (ISS) in order to prevent the occurrence of new interrupt during the execution of ISS. Now, the program control is transferred to the specific CALL locations by generating CALL signal and after executing the ISS, the program control is again transferred to the main program (stack-top) from where it had received the interrupt.

Interrupt are classified into following groups based on their parameter −

Vectored interrupt − Vectored Interrupts are those which have fixed vector addresses (starting address of the ISS) and on executing these, program control is transferred to that specific address.Vector addresses are calculated by the formula 8 * TYPE.In this type of interrupt, the interrupt address is known to the processor. **For example:** RST7.5, RST6.5, RST5.5, TRAP.

Interrupt	Vector Address
TRAP (RST 4.5)	24 H
RST 5.5	2C H
RST 6.5	34 H

RST 7.5	3C H

- **Non-Vector interrupt** − In this type of interrupt, the vector interrupt address is not known to the processor. The interrupting device gives the address of the sub-routine for these interrupts. INTR is the only non-vectored interrupt in 8085 microprocessor.
- **Maskable interrupt** –Interrupts which are enabled and disabledby software using instructionssuch as EI or DI and SIM (Set Interrupt Mask). The bit pattern of the accumulator enables/disables the individual interrupt. These interrupts are either edge-triggered or level-triggered. For example:RST7.5, RST6.5, RST5.5.
- **Non-Maskable interrupt** − Non-maskable Interrupts are those which cannot be disabled or ignored by microprocessor through software. TRAP is a non-maskable interrupt. It consists of both level as well as edge triggering and is used in critical power failure conditions.
- **Software interrupt** − Software Interrupts are those which are interrupted by means of mnemonics of the microprocessor. In this type of interrupt, the programmer has to add the instructions into the program to execute the interrupt. There are 8 software interrupts in 8085, i.e. RST0, RST1, RST2, RST3, RST4, RST5, RST6, and RST7.
- For Software interrupts vector addresses are listed by:

Interrupt	**Vector Address**
RST 0	00 H
RST 1	08 H
RST 2	10 H
RST 3	18 H
RST 4	20 H
RST 5	28 H
RST 6	30 H
RST 7	38 H

- **Hardware interrupt** − When the microprocessor receives interrupt signals through pins (hardware), they are known as Hardware Interrupts. There are 5 Hardware Interrupts in 8085 microprocessor. They are – INTR, RST 7.5, RST 6.5, RST 5.5 and TRAP.

Note − INTA is not an interrupt; it is used by the microprocessor for sending acknowledgement. TRAP has the highest priority and TRAP is the lowest priority interrupt. Priority defines the importance of the particular interrupt when there are simultaneous occurrences of interrupts.

4.2 INTERRUPT SERVICE SUBROUTINE (ISS)

A small program or a subroutine that is to be executed by the microprocessor or by an external device at the time of receiving the interrupt signal is called an ISS.

TRAP

It is a non-maskable interrupt, having the highest priority among all interrupts. Bydefault, it is enabled until it gets acknowledged. In case of power failure, it executes an ISS and sends the data to the backup memory. This interruptsignal transfers the program control to the location 0024H.

RST7.5

It is a maskable interrupt, having the second highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches the control to 003CH address.

RST 6.5

It is amaskable interrupt, having the third highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches the control to 0034H address.

RST 5.5

It is amaskable interrupthaving the fourth highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches the control to 002CH address.

INTR

It is a maskable interrupt, having the lowest priority among all the above mentioned interrupts. There are 8 numbers of CALL-locations for INTR interrupts. RST n instruction is used to transfer the program control to a specific location. It can be disabled by resetting the microprocessor.

When **INTR signal goes high**, the following events can occur −

- The microprocessor checks the status of INTR signal during the execution of each instruction.
- When the INTR signal is high, then the microprocessor completes its current instruction and sends active low interrupt acknowledge signal.
- When instructions are received, then the microprocessor saves the address of the next instruction on stack and executes the received instruction.

4.3 SUBROUTINE

In a microprocessor, a subroutine is a sequence of program instructions that may perform a specific task, written as a unit. This unit can then be used in the main program wherever that particular task is to be performed. A subroutine is often called several times from several places during the execution of the main programand then return back after completing the task to the next instruction after the call instruction in the main program. It is implemented by using Call and Return instructions. The program control can also be transferred from one subroutine to another by using the same was as described in the case of main program to subroutine. The different types of subroutine instructions are

Unconditional CALL instruction

CALL address is the format for unconditional call instruction. After execution of this instruction,the program control is transferred to a subroutine whose starting address is specified in the instruction.Before transferring the program control, value of PC (Program Counter) is transferred to the memory stack and value of SP (Stack Pointer) is decremented by 2.

Conditional CALL instruction

In these instructions program control is transferred to a subroutine if the condition stated in the instruction is satisfied else the program control will follow as usual i.e to the next instruction followed to the conditional CALL instruction.Before transferring the program control, value of PC (Program Counter) is transferred to the memory stack and value of SP (Stack Pointer) is decremented by 2.

Unconditional Return instruction –

RET is the instruction used to mark the end of subroutine. It has no suffix. After execution of this instruction program control is transferred back to the main program from where it had branched. To achieve this, the value of PC (Program Counter) is retrieved from the memory stack and value of SP (Stack Pointer) is incremented by 2.

Conditional Return instruction –

The execution of these instructions,transfer the program control back to the main program if the condition stated in the instruction is satisfied. To achieve this, the value of PC (Program Counter) is retrieved from the memory stack and value of SP (Stack Pointer) is incremented by 2. The instruction has no suffix.

Advantages of Subroutine –

- Decomposing a complex programming task into simpler steps.
- Reducing duplicate code of a task within a program.
- Enabling reuse of the code of a task across multiple programs.
- Improving tractability of the control logic or makes debugging of a program easy.

4.4 FLOWCHARTING

Flowcharting is a pictorial tool to represent a program. It allows to break down a program into blocks and to show the logical relationships between them pictorially. Constructing flowcharts promotes better understanding of the program for debugging and for improvement.A flowchart is a visual representation of the sequence of steps and decisions needed to execute a program. Different diagrammatic shapes are used to represent a step and these shapes are connected by lines with directional arrows to represent the flow of execution or program control. This allows anyone to logically follow the program from beginning to end.

A flowchart is a powerful tool. With proper design and construction, it communicates the steps in a program very effectively and efficiently.

Flow Chart Symbols

The flowchart has different shapes as described in the diagram.

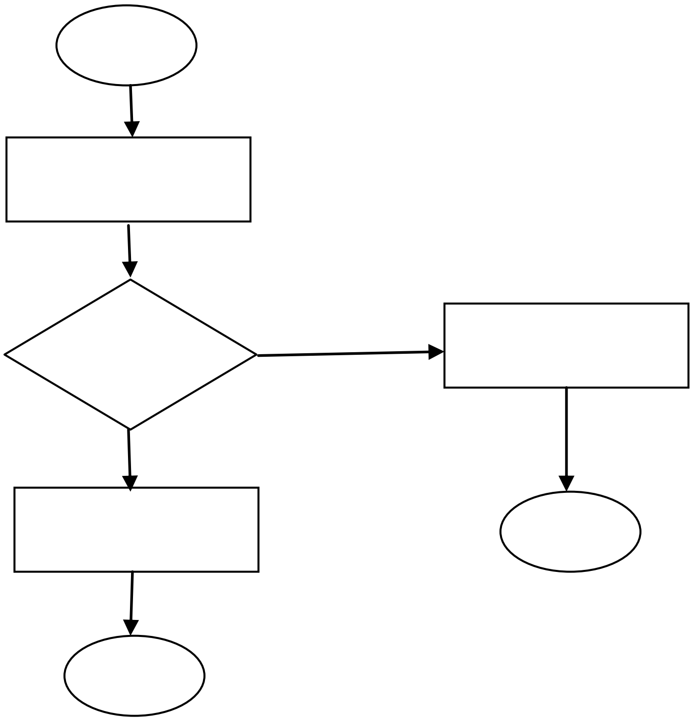


Figure 4.1 Model of flow chart system

These shapes are known as flowchart symbols. To construct a flowchart for a program, there are simple guidelines which can be used as a guide for constructing flowcharts.

- Divide the program in to simple tasks
- Indentify the inputs
- Construct the logical flow successively and clearly
- Decide the break points to ascertain the flow of execution
- Generate outputs

- End the program

4.5 LOOPS

The programming technique used to instruct the microprocessor to repeat a taskis called looping.This task is generally accomplished by using branching instructions. For example: JMP

Classification of loops

1. Continuous loop

2.Unconditional loop

Continuous loop:

- Repeats a task continuously.
- A continuous loop is set up by using the unconditional branch control instruction
- A program with a continuous loop does not stop repeating the tasks until the system is reset.

Conditional loop:

- A conditional loop is set up by a conditional branch control instruction.
- These instructions check values of status flags (Z,CY,P,S) and repeat the tasks if the conditions are satisfied.
- These loops include counting and indexing.

Application of Conditional loop as counter:

- A counter is a typical application of the conditional loop.
- A microprocessor needs a counter,flag to accomplish the looping task.
- Counter is set up by loading an appropriate count in a register.
- Counting is performed by either increment or decrement the counter.
- Loop is set up by a conditional branch control instruction.
- End of counting is indicated by a flag.

4.6 PSEUDO-INSTRUCTIONS

Pseudo-instructions are special commands or directives to the assembler to assemble the program. These instructions have mnemonics but do not have a direct machine language equivalent. They are not part of instructions, no operation code and not executable statements. Duringassembly, the assembler translates each pseudoinstruction into one or more machine. The following are the examples for pseudo instructions;ORG indicates the starting memory location for loading the instructions and END indicates the end of the program, no instructions to be translated into machine code.

4.7 STACK OPERATIONS

A stack is a reserved area for temporary storage of data in the RAM. The stack is defined usually at the highest address of the available RAM location using LXI SP, XX99 H instruction. It means the location is available for storing. An 8-bit stack pointer is used to hold the address of the top of the stack.The stack works on the principle of LIFO(Last in first out). The SP is decremented by one when data is stored and incremented by one when data is retrieved. The data which is stored at last will be retrieved at first. The contents are stored and retrieved on stack by using PUSH and POP instructions respectively.

Status of Stack before PUSH B instruction		
Memory Location	Stack Content	Stack Pointer (SP) Content
2504		
2505		
2506		
2507	5B	2507
2508		

After PUSH B instruction, the SP content will be 2505 since the content of Register B will occupy 2506 and the content of Register C will occupy 2505.

Status of Stack before POP B instruction		
Memory Location	Stack Content	Stack Pointer (SP) Content
2504		
2505	1A	2505
2506	2B	
2507	5B	
2508		

After POP B instruction, the SP content will be 2507 since the content of Register B will be 2B and the content of Register C will be 1A.

4.8 PROGRAMMING AND APPLICATIONS: TRAFFIC CONTROL SYSTEM

The 8085 is a popular microprocessor used for various applications such as traffic light control, temperature control, stepper motor control, etc. The following text describes a microprocessor based traffic control system and a model assembly language program to activates the traffic lights at a desired direction and at a desired time. The normal function of traffic light system ensures that traffic moves as smoothly and safely as possible and that pedestrians are protected when they cross the roads.It minimizes the accidents and helps to streamline the flow of vehicles.

In this scheme, the traffic lights are interfaced to a microprocessor through a programmable peripheral Interface 8255 to automatically switch ON/OFF the traffic lights through its ports in desired sequence. Once the address of the particular traffic light is indentified, then it can be switched ON/OFF by sending a valid data from the microprocessor to that location through the ports by using the OUT instruction.A delay program sequence will be used to design the desired time for keeping the lights in ON or OFF condition. Looping can also be used to continue the task of switching the lights in sequence continuously. The one such assembly language program for the application of traffic light control scheme is written below;

Program

```
          MVI A, 80H: Initialize 8255, Port A and Port B

          OUT 83H (CR): in output mode

START     MVI A, 09H

          OUT 80H (PA): Send data on Port A to glow R1 and R2

          MVI A, 24H

          OUT 81H (PB): Send data on Port B to glow G3 and G4

          MVI C, 28H: Load multiplier count for delay

          CALL DELAY: Call delay subroutine

          MVI A, 12H

          OUT (81H) PA: Send data on Port A to glow Y1 and Y2

          OUT (81H) PB: Send data on Port B to glow Y3 and Y4

          MVI C, 0AH: Load multiplier count for delay
```

```
            CALL: DELAY: Call delay subroutine
            MVI A, 24H
            OUT (80H) PA: Send data on Port A to glow G1 and G2
            MVI A, 09H
            OUT (81H) PB: Send data on Port B to glow R3 and R4
            MVI C, 28H: Load multiplier count (40ıo) for delay
            CALL DELAY: Call delay subroutine
            MVI A, 12H
            OUT PA: Send data on Port A to glow Y1 and Y2
            OUT PB: Send data on Port B to glow Y3 and Y4
            MVI C, 0AH: Load multiplier count for delay
            CALL DELAY: Call delay subroutine
            JMP START
```

Delay Subroutine:

```
DELAY:      LXI D, Count: Load count to give 0.5 sec delay
BACK:       DCX D: Decrement the counter by one
            MOV A, D
            ORA E: Check whether count is 0
            JNZ  BACK address: If not zero, go to BACK
            DCR C: Decrement the counter by one
            JNZ DELAY address: Check if multiplier zero, go to DELAY
            RET: Return to main program
```

UNIT – V MICROCONTROLLER 8051

Structure

5.1 INTRODUCTION TO 8 BIT MICROCONTROLLERS

The 8051 is a single chip microcomputer contains all essential hardware elements like CPU, ROM/EPROM, RAM and I/O Ports and software. It has been designed for dedicated applications and is in wide use today. The application areas includes domestic, all kinds of industries, defense and space. Today's microcontrollers are quite powerful as measured by the activities they can carry out. They are relatively low cost and function with very low power. Basically they are fabricated out of IC technology that executes a user program, normally for the purpose of controlling some device. Microcontrollers have several features arising from their embedded system environment. Some of them are; ready to interface with sensors and actuators, in-built peripheral devices, work with simple dedicated program, smaller in size, consume less power and less in price.

5.2 8051 MICROCONTROLLER ARCHITECTURE

8051Microcontroller's block diagram is shown below. Let's have a closer look on features of 8051 microcontroller design:

CPU (Central Processor Unit):

CPU is the brain of any processing machine andthe main function is to fetch and decode the instructions stored in the memory in order to complete the assigned task successfully. With the help of CPU all the components of microcontroller is connected into a single system for effective management. User has no power over the functioning of CPU.

Memory:

Memory stores all programs and data. Program is a set of commands to the microcontroller to perform certain tasks. Microcontrollers are built with certain amount of ROM for the storage of program source codes.Microcontroller also needs a memory to accumulate data or operands for a short term or momentarily data storage for functioning. This will employ Random Access Memory or RAM. Microcontroller 8051 contains code memory or program memory of 4KB ROM and it also comprise of data memory (RAM) of 128 bytes.

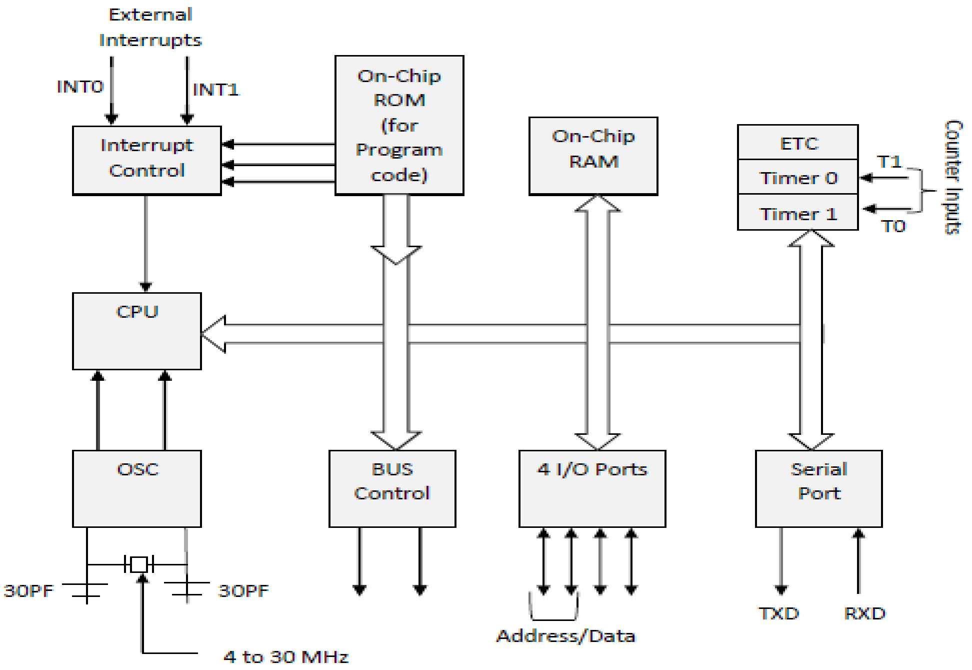


Figure 5.1 Block Diagram of 8051 Microcontroller

Bus:

Bus is a group of wires which transfer Data or Address in the form of 0s and 1s. A logic '1' on each line of the bus is commonly represented by 5V or 3.3V. Logic '0' is commonly represented by a 0V signal. This bus comprises of 8, 16 or more wires and is only capable of conveying one bit of data at a time in a wire. As a result, a bus can bear 8 bits or 16 bits all together. Any device that the CPU needs to communicate with is connected (in either in parallel with others or in serial) to the bus.There are two types of buses:

1. **Address Bus:** Microcontroller 8051 consists of 16 bit address bus. Its purpose is to select which of the external devices (or memory locations) is allowed to use the data bus by transmitting the corresponding address from Central Processing Unit to Memory or device.

2. **Data Bus:** Microcontroller 8051 comprise of 8 bits data bus. It is employed to cart data between memory or devices with CPU.

Interrupts:

Interrupts are special signals that effect the CPU to suspend its current activity and perform some other task for the hardware device that require urgent attention. A microcontroller may need to support or monitor several hardware subsystems each of which will occasionally require urgent attention. CPU's are commonly designed to handle interrupts from various sources. Each interrupt signal is associated with a particular memory location which contains the address of the subroutine (a set of instructions) that should be executed on demand. When subroutine task is finished then the implementation of core program initiates automatically as usual. There are 5 interrupt signals in 8051 Microcontroller, two out of five are peripheral interrupts, two are timer interrupts and one is serial port interrupt. Interrupts facilitate the users to postpone or delay the current process, carry out a sub-routine task and then all over again restart the standard program implementation.An Interrupt vector is a memory location which contains the address of an interrupt service routine. Interrupt service subroutine is aset of codes that is to be executed on receipt of an interrupt signal. Interrupt vector table is a collection of interrupt vectors.

Oscillator:

Microcontroller 8051 consists of an on-chip oscillator which toils as a time source for CPU (Central Processing Unit). It harmonize the various functions of the devices and memory in to a single system. A microcontroller may be in-built with one or more timer or counters. The timers and counters control all counting and timing operations within a microcontroller. The main operations performed by timers are pulse generations, clock functions, frequency measuring, modulations, making oscillations, etc.

Input/output Port: Microcontroller is employed in embedded systems to manage the functions of various devices.For this, Micro-controller 8051 consists of 4 input/output ports to unite it to other peripherals.

Some of the Applications of 8051 microcontroller

1. **Energy Management:** Competent measuring device systems aid in calculating energy consumption in domestic and industrialized applications. These meter systems are prepared competent by integrating microcontrollers.
2. **Touch screens:** A high degree of microcontroller suppliers integrate touch sensing abilities in their designs. Transportable devices such as media players, gaming devices and cell phones are some illustrations of micro-controller integrated with touch sensing screens.
3. **Automobiles:** The microcontrollers are extensively utilized in motor vehicles to control engine parameters. In addition, works such as cruise power and anti-brake mechanism has created it more capable with the amalgamation of micro-controllers.

4. **Medical Devices:** Handy medicinal gadgets such as glucose and blood pressure monitors bring into play micro-controllers, to put on view the measurements, as a result, offering higher dependability in giving correct medical results.

5.3 FEATURES OF 8051 MICROCONTROLLER

8051 Microcontroller Features

- **8 – Bit ALU**: ALU or Arithmetic Logic Unit is the heart of a microcontroller. It performs arithmeticand bitwise (logic) operation on binary numbers. The ALU in 8051 is an 8 – Bit ALU and can perform operations on 8 – bit data.
- **8 – Bit Accumulator**:The Accumulator is an important register of 8 bit associated with the ALU.
- **RAM**: 8051 Microcontroller has 128 Bytes of RAM which includes SFRs and Input / Output Port Registers.
- **ROM**: 8051 has 4 KB of on-chip ROM (Program Memory).
- **I/O Ports**: 8051 has four 8 – bit Input / Output Ports which are bit addressable and bidirectional.
- **Timers / Counters**: 8051 has two 16 – bit Timers / Counters.
- **Serial Port**: 8051 supports full duplex UART Communication.
- **External Memory**: 8051Microcontroller can access two 16 – bit address line at once: one each for RAM and ROM. The total external memory that an 8051 Microcontroller can access for RAM and ROM is 64KB (2^{16} for each type).
- **Additional Features**: Interrupts, on-chip oscillator, Boolean Processor, Power Down Mode, etc.
- **Clock:**1 Microsecond instruction cycle with 12 MHz Crystal.

5.3.1 PIN DIAGRAM OF 8051 MICROCONTROLLER

8051 microcontroller consists of 40 pins, where in 32 pins are meant for the 4 ports P0, P1, P2 and P3. Each port was assigned with 8 pins. The remaining 8 pins are dedicated toVcc, GND, XTAL1, XTAL2, RST, EA, ALE and PSEN. These 8 pins are common in 8051 families.

- **Vcc:** Voltage supply is +5V.
- **GND:** Ground connection.
- **XTAL1 & XTAL2: O**n-chip oscillator but requires an external clock to run it and most often is connected XTAL1 and XTAL2.
- **RST:** Reset pin which is active high. This pin is often referred to as power or reset button. Activating this pin will cause all values in the register to be lost and terminate all activities.
- **EA:** On-chip ROM to store programmes.
- **PSEN:** This stands for program code enabled. In 8051, an external ROM holds the program code, so this pin is connected to the ROM.

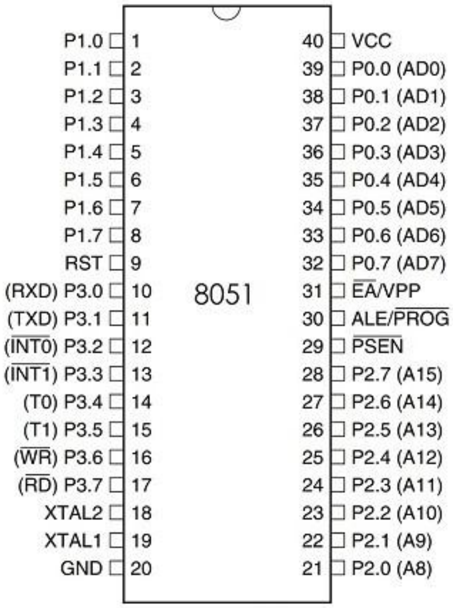


Figure 5.2Pin diagram of 8051 microcontroller

- **ALE:** Address Latch Enable – an output pin which is active High and is used for demultiplexing the address and data.
- **P0:** Pins 32 to 39. It can be used as input or output port. To use the pins of P0 as both input or output, each pin must be connected externally to a pull-up register. Data 0 for output and 1 for input.
- **P1:** Pins 1 to 8. It can be used as input or output. In contrast to P0, this port doesn't need any pull-up register. Since it already has pull-up register externally. P1 can be configured as an output upon reset. To make the P1 as input, it must be programmed as such by writing 1 to all of its bits.
- **P2:** Pins 21 to 28. It can also be used as input or output like P1. P2 doesn't need any pull-up register since it already has registers internally. P2 can be configured as an output upon reset. To make it input, it must be programmed as such by writing 1 to all its bit.
- **P3:** Pins 10 to 17.It can also be used as input or output. P3 doesn't need any pull-up register as same as P1 and P2. P3 has the additional function of providing some extremely important functions and signals such as interrupts. P3.0 and P3.1 are used for **RXD** and **TXD** serial communication signal. P3.2 and P3.3 are set aside for external interrupts. P3.4 and P3.5 are used for timer 0 and 1. Finally, P3.6 and P3.7 are used to provide the write and read signal for external memory connected to the 8051 microcontroller system.

5.4 8051 MEMORY ORGANIZATION

The 8051 microcontroller's memory is divided into Program Memory and Data Memory. Program Memory (ROM) is used to store the program being executed, while Data Memory (RAM) is used for temporarily storing and keeping intermediate results and variables. This architecture has the advantage that a program cannot accidently overwrite itself and is being used in embedded systems. In addition, it uses the same address, in different memories, for program and data. The microprocessor accesses the correct memory based on the nature of the operation in progress.

Program Memory (ROM)

Program Memory (ROM) is used for saving program (CODE) being executed. The memory is generally read only. The 8051 executes programs stored in program memory only.

Internal Data Memory

The internal RAM is 128 byte with address space from 00H to 7FH, i.e. first 128 registers and this part of RAM is divided in several blocks. The 128 bytes of internal data memory are accessed directly by their address. The upper 128 bytes of data memory (from 80H to 0FFH) can be addressed only indirectly. The address 00H to 1FH are reserved for register banks R0-R3 and the stack pointer. The address 20H – 2FH are designated for bit-addressable memory. From 30H-7FH are used for read and write storage called scratch pad.

Stack Pointer

The stack is a portion of the RAM used by CPU for storing the information (data or an address) temporarily. The CPU uses this storage area when there is shortage of registers. The Stack Pointer (SP) register is used to locate the stack memory for read and write. The 8 bit stack can take values of 00H to FFH. On power on, the SP register in the microprocessor contains the value 07H, implying that 08H is available for storing by PUSH instruction. POP instruction is used to get the data from the stack.

External Data Memory

Up to 64K Bytes of external data memory can be accessed. Access to external memory is slower than access to internal data memory. Several 8051 devices provide on-chip XRAM space that is accessed with the same instructions as the traditional external data space. This XRAM space is typically enabled via proper setting of Special Function Register and overlapping the external memory space. The register must be manually set by program, before any access to external memory or XRAM space is made.

5.5GENERAL PURPOSE REGISTERS

In total, 34 general purpose registers are in 8051 microcontroller including register A (Accumulator) and register B. The RAM memory of the 8051 microcontroller is divided into 3 areas such as register banks, bit-addressable area, and scratch-pad area. The banks contain different general purpose registers such as R0-R7, and all such registers are byte-addressable registers that store or remove only 1-byte of data at a time. There are four register banks with each bank having 8 addressable 8-bit registers, and only one register bank can be accessed at a time. But, by changing the register bank's number in the flag register, onecan access other register banks.

Banks and Registers

The B0, B1, B2, and B3 stand for banks and each bank contains eight general purpose registers ranging from 'R0' to 'R7'. All these registers are byte-addressable registers. Data transfer between general purpose registers to general purpose registers is not possible. These banks are selected by the Program Status Word (PSW) register.

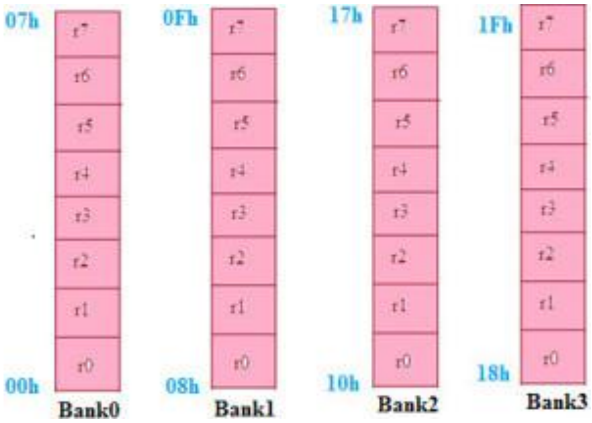


Figure 5.3 Banks and Registers systems

PSW (Program Status Word) Register

The PSW register is a bit and byte-addressable register. It indicates the status of the current operation being done in the controller. RS1 and RS0 pins in the PSWdenotethe bank selection.The physical address of the PSW starts from D0Hpin and the individual bits are also accessed with D0H (PSW0) to D7H (PSW7).

Carry Flag (C):

The flag is affected after an 8bit addition or subtraction carried out. This flag is set when there is a carry generated out from D7 bit. It can also be set to 1 using "SETB C" and set to 0 by "CLR C" instructions are used.

Auxillary Carry Flag(AC):

This auxiliary carry is affected when there is a carry generated from the D3 to D4 during addition or subtraction operations.

Overflow Flag (OV):

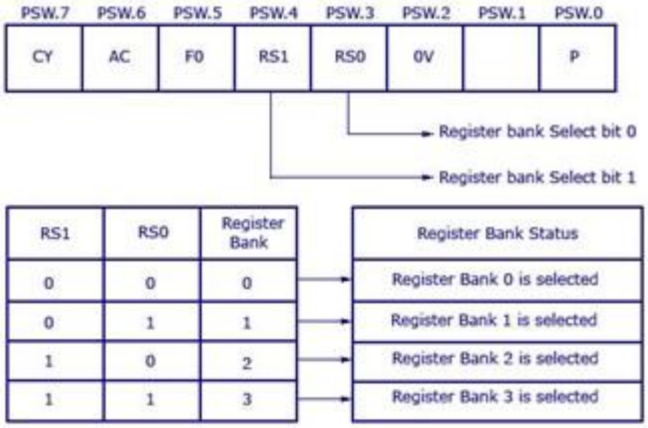
This flag is set whenever the result of a signed number is too large, causing the higher order bit is overflow into sign bit. This can be used to detect the errors in signed arithmetic operations.

Parity Flag (P)

This flag reflects the number of 1s in the register A. If register A contains even number of 1s, then P=0 else P=1 for odd number of 1s.

5.5.1SPECIAL FUNCTION REGISTER

A Special Function Register (SFR) is a register within a microprocessor that controls or monitors the various functions of a microprocessor. These SFRs can be addressed directly as like internal RAM using addresses from 80H to FFH and can also be read or written. In 8051, register A, B, DPTR, and PSW are a part of the group of registers commonly referred to as SFR (special function registers). An SFR can be accessed by its name or by its address.Not all the address spaces of 80 to FF are used by the SFR. Unused locations, 80H to FFH, are reserved and must not be used by the 8051 programmer.



PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
CY	AC	F0	RS1	RS0	0V		P

RS1	RS0	Register Bank	Register Bank Status
0	0	0	Register Bank 0 is selected
0	1	1	Register Bank 1 is selected
1	0	2	Register Bank 2 is selected
1	1	3	Register Bank 3 is selected

Figure 5.4 Process Status Word (PSW)

5.6 OSCILLATOR AND CLOCK CIRCUIT

System Clock

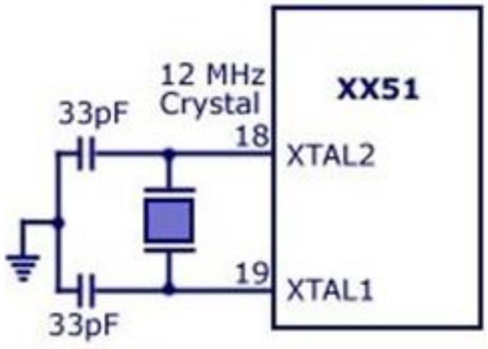


Figure 5.5 8051 clock circuit

The 8051 requires an external oscillator circuit. The oscillator circuit generates 12M pulses in one second. The pulse is used to synchronize the system operation in a controlled pace. A machine cycle is the minimum amount of time required for an instruction to be executed by 8051.The microcontrollers are designed to run at specified maximum and minimum frequencies. Minimum frequency refers to the dynamic characteristic of the internal memories and must always operate above this frequency in order to retain the data.

An example 8051 clock circuit is shown above. In general cases, a quartz crystal is used to make the clock circuit and connected to XTAL 1 and XTAL 2.

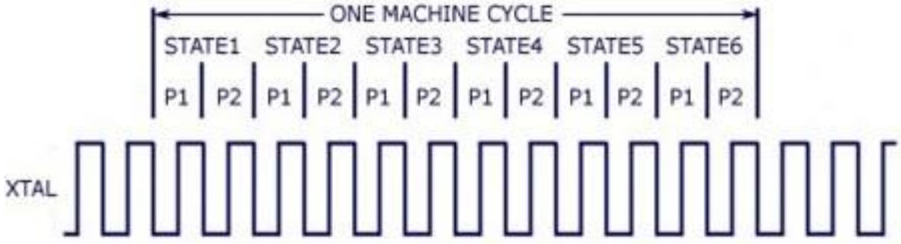


Figure 5.6 System clock

The above picture represents the machine cycle. The number of machine cycle consumes for an instruction varies according to the task.

5.7 8051 ADDRESSING MODES

The addressing mode refers to the form in which a given address or data is being accessed by the microcontroller to perform different types of operations. A total of 5 types of addressing modes in µC 8051. They are

1. Immediate Addressing
2. Register Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Indexed Addressing

1. Immediate Addressing Mode

Data is immediately available in the instruction.

For example -

ADD A, #77; Adds 77 (data - decimal) to A and stores in A

MOV R2, #4DH; load 4D (data-hexadecimal) to registerR2

MOV DPTR, #1000H; Moves 1000 (16 bit hexadecimal) to data pointer

2. Register Addressing Mode

The instruction involves the use of registers (R0-R7) to hold the data to be manipulated. The four register banks consisting of eight registers (R0-R7). One of these four banks is selected by a 2 bit field in PSW register

For example-

ADD A, R5; Adds content of R5 to A and stores in A. The op-code is 00101101B, The upper five bits 00101 refers the instruction and the lower three bits 101 refers the register A.

MOV R6, A : save the content of register A in register A6. Data can be moved between A and Rn(n= 0 to7) or vice-versa, but data movement between Rn registers is forbidden. For instance, MOV R3,R2 is not allowed and invalid.

3. DirectAddressing

The direct addressing mode, the content of the memory location whose address is specified in the instruction will be processed. It is used to access the RAM locations 30H to 7FH since other locations are designated for register banks and stack (00H to 1FH) and bit addressable space (20H to 2FH). In direct addressing mode the absence of the sign # indicates that the data followed the instruction is address.

For example -

MOV A, 088H; Moves the content of SFR TCON (address 088H)to A

MOV R3, 55H; Move the content of memory location 55H to R3

MOV R3, #55H; Load the data 55H in to R3.

4. Register Indirect Addressing

The Register Indirect Addressing mode utilizes the Registers R0 and R1 to process the operation as pointers with a sign @ preceded before them.

For example -

MOV A, @R0; Moves the content of address in the RAM pointed by the register R0 to A. Here the register R0 holds the address of the RAM location.

MOV @R0, A; Moves the content of A into the memory location whose address is held by the register R0.

5. Indexed Addressing

The indexed addressing mode is effectively used to access the data of look-up table entries located in the ROM space of the microcontroller. The registers A and DPTR are used to index the address of the data stored in ROM.

For example –

MOVC A, @A+DPTR; Here the effective address will be indexed (using contents of A and DPTR i.e, A+DPTR) before the execution of the instruction. If A = 03 and DPTR is 0402(16 bit address), then A+DPTR is 0405 (16 bit address). The content in the memory location (ROM) whose address is 0405 will be moved to the register A. The letter C in the instruction denotes code space.

5.8 8051 INTERRUPTS OF 8051

An interrupt is a special feature in the microcontroller that can seek the attention of the microcontroller to perform a specific task instead of the current task wherein the microcontroller is engaged thereby the flow of execution can be diverted. Depending upon interrupt priority, the microcontroller may execute the specific task coded in the interrupt vector address.

8051 architecture handles 5 interrupt sources, out of which two are internal (Timer Interrupts), two are external and one is a serial port interrupt. In addition, the reset pin is also included as the sixth interrupt. Each of these interrupts has their unique interrupt vector address. Highest priority interrupt is the Reset, with vector address 0000H.

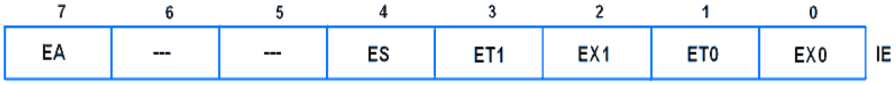
Vector Address: This is the address where the controller transfers its flow of execution after receiving the interrupt in order to execute the ISR (interrupt service routine).

Interrupt	Flag	Interrupt vector address
Reset	-	0000H
INT0 (External interrupt 0)	IE0	0003H
Timer 0	TF0	000BH
INT1 (External interrupt 1)	IE1	0013H
Timer 1	TF1	001BH
Serial	TI/RI	0023H

Each of the interrupt is either enabled (unmasked) or disabled (masked) through the program by using the SFR, IE (Interrupt Enable Register) located at the memory address 0A8H. The details are given below.

IE register: Interrupt Enable Register

IE register is used to enable/disable interrupt sources.



7	6	5	4	3	2	1	0	
EA	---	---	ES	ET1	EX1	ET0	EX0	IE

Bit 7 – EA: Enable All Bit

1 = Enable all interrupts

0 = Disable all interrupts

Bit 6,5 – Reserved bits

Bit 4 – ES: Enable Serial Interrupt Bit

1 = Enable serial interrupt

0 = Disable serial interrupt

Bit 3 – ET1: Enable Timer1 Interrupt Bit

1 = Enable Timer1 interrupt

0 = Disable Timer1 interrupt

Bit 2 – EX1: Enable External1 Interrupt Bit

1 = Enable External1 interrupt

0 = Disable External1 interrupt

Bit 1 – ET0: Enable Timer0 Interrupt Bit

1 = Enable Timer0 interrupt

0 = Disable Timer0 interrupt

Bit 0 – EX0: Enable External0 Interrupt Bit

1 = Enable External0 interrupt

0 = Disable External0 interrupt

Interrupt priority

Priority to the interrupt can be assigned by using **interrupt priority register (IP)**

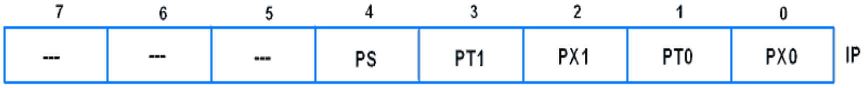
Interrupt priority after Reset:

Priority	Interrupt source	Intr. bit / flag
1	External Interrupt 0	INT0
2	Timer Interrupt 0	TF0
3	External Interrupt 1	INT1
4	Timer Interrupt 1	TF1
5	Serial interrupt	(TI/RI)

In the table, interrupts priorities upon reset are shown. As per 8051 interrupt priorities, lowest priority interrupts are not served until microcontroller is finished with higher priority ones. In a case when two or more interrupts arrives microcontroller queues them according to priority.

IP Register: Interrupt priority register

8051 has interrupt priority register to assign priority to interrupts.



7	6	5	4	3	2	1	0	
---	---	---	PS	PT1	PX1	PT0	PX0	IP

Bit 7,6,5 – Reserved bits.

Bit 4 – PS: Serial Interrupt Priority Bit

1 = Assign high priority to serial interrupt.

0 = Assign low priority to serial interrupt.

Bit 3 – PT1: Timer1 Interrupt Priority Bit

1 = Assign high priority to Timer1 interrupt.

0 = Assign low priority to Timer1 interrupt.

Bit 2 – PX1: External Interrupt 1 Priority Bit

1 = Assign high priority to External1 interrupt.

0 = Assign low priority to External1 interrupt.

Bit 1 – PT0: Timer0 Interrupt Priority Bit

1 = Assign high priority to Timer0 interrupt.

0 = Assign low priority to Timer0 interrupt.

Bit 0 – PX0: External0 Interrupt Priority Bit

1 = Assign high priority to External0 interrupt.

0 = Assign low priority to External0 interrupt.

Reset

Reset is the highest priority interrupt. Upon reset pin is enabled, 8051 microcontroller starts executing code from 0000H address.

Internal interrupt (Timer Interrupt)

8051 has two internal interrupts namely timer0 and timer1. Whenever timer overflows, timer overflow flags (TF0/TF1) are set. Then the microcontroller jumps to their vector address to serve the interrupt. For this, EA and timer interrupt should be enabled.

Serial interrupt

8051 has serial communication port and have related serial interrupt flags (TI/RI). When the last bit (stop bit) of a byte is transmitted, TI serial interrupt flag is set and when last bit (stop bit) of receiving data byte is received, RI flag get set.

Timers/Counters

In 8051, two 16 bit timers/counters are provided to count the event or to estimate the elapsed time by means of counting the internal/external periodic pulses. Each can be programmed independently either to count internal closk or to count external input pulses. A timer with “n” stages divides the input clock frequency by 2^n. A 16 bit timer would counts from 0000H to FFFFH. The counting will be monitored by overflow flag, which is set on the FFFFH to 0000H overflow of the count. TMOD (Timer Mode Register) and TCON (Timer Control Register) are the special SFRs to set the mode of operation and configure of the timerand timer status respectively.

Timer Mode Control (TMOD)

TMOD is an 8-bit register used for selecting timer or counter and mode of timers. Lower 4-bits are used for control operation of timer 0 or counter 0, and remaining 4-bits are used for control operation of timer1 or counter1.This register is present in SFR register, the address for SFR register is 89th. The two timers of 8051 can be set as timers or event counters by the status of the

C/T bit in the TMOD register. Mode 1 is the commonly used mode and the timers are 16 bits in size. It is common to configure the timer to cause an interrupt when it overflows. The interrupt service routine then initializes the timer.

Timer Control Register (TCON)

TCON is an 8 bit register and TR0 and TR1 flags in the TCON register enable a timer to run, when set. The address of the TCON is 88H.

UNIT – VI 8051 INSTRUCTION SET AND PROGRMMING

Structure

6.1 INTRODUCTION TO 8051 INSTRUCTION SET

An instruction is a command given to the microcontroller in a particular order in which they must be executed in order to perform a specific operation on the given data. These commands are called Instruction Set.An Instruction Set is unique to a microcontroller.As a typical 8-bit processor, the 8051 Microcontroller instructions have 8-bit Opcodes. As a result, the instruction set can have up to 2^8 = 256 Instructions.

6.2 TYPES OF INSTRUCTIONS

An 8051 Instruction consists of an Opcode(Operation Code) followed by Operand(s) of size Zero Byte, One Byte or Two Bytes. The Opcode part of the instruction contains the Mnemonic, which specifies the type of operation to be performed. All Mnemonics or the Opcodes part of the instruction are of One Byte size. The operand part of the instructiondefines the data being processed by the instructions. A simple instruction consists of just the opcode. Other instructions may include one or more operands. Instruction can be one-byte instruction, which contains only opcode, or two-byte instructions, where the second byte is the operand or three byte instructions, where the operand makes up the second and third byte.

Based on the operation they perform, the Instruction Set of 8051 Microcontroller is divided into five basic groups. They are:

- Data Transfer Instructions
- Arithmetic Instructions
- Logical Instructions
- Boolean or Bit Manipulation Instructions
- Program Branching Instructions

6.2.1 DATA TRANSFER INSTRUCTIONS

The Data Transfer Instructions are associated with transfer of data between registers or external program memory or external data memory. The instructions associated with Data Transfer are given below.

MOV A,Rn	Moves the register to the accumulator
MOV A,direct	Moves the direct byte to the accumulator
MOV A,@Ri	Moves the indirect RAM to the accumulator
MOV A,#data	Moves the immediate data to the accumulator
MOV Rn,A	Moves the accumulator to the register
MOV Rn,direct	Moves the direct byte to the register
MOV Rn,#data	Moves the immediate data to the register
MOV direct,A	Moves the accumulator to the direct byte
MOV direct,Rn	Moves the register to the direct byte
MOV direct,direct	Moves the direct byte to the direct byte
MOV direct,@Ri	Moves the indirect RAM to the direct byte
MOV direct,#data	Moves the immediate data to the direct byte
MOV @Ri,A	Moves the accumulator to the indirect RAM
MOV @Ri,direct	Moves the direct byte to the indirect RAM
MOV @Ri, #data	Moves the immediate data to the indirect RAM
MOV DPTR, #data16	Loads the data pointer with a 16-bit constant
MOVC A,@A + DPTR	Moves the code byte relative to the DPTR to the accumulator
MOVC A,@A + PC	Moves the code byte relative to the PC to the accumulator
MOVX A,@Ri	Moves the external RAM (eight-bit address) to A
MOVX A,@DPTR	Moves the external RAM (16-bit address) to A
MOVX @Ri,A	Moves A to the external RAM (eight-bit address)
MOVX @DPTR,A	Moves A to the external RAM (16-bit address)
PUSH direct	Pushes the direct byte onto the stack
POP direct	Pops the direct byte from the stack
XCH A,Rn	Exchanges the register with the accumulator
XCH A,direct	Exchanges the direct byte with the accumulator
XCH A,@Ri	Exchanges the indirect RAM with the accumulator
XCHD A,@Ri	Exchanges the low-order nibble indirect RAM with A

6.2.2. ARITHMETIC INSTRUCTIONS

Arithmetic Instructions are used to perform addition, subtraction, multiplication and division. The arithmetic instructions also include increment by one, decrement by one and a special instruction called Decimal Adjust in

Accumulator.The instructions associated with the Arithmetic Instructions of the 8051 are as follows:

ADD A,Rn	Adds the register to the accumulator
ADD A,direct	Adds the direct byte to the accumulator
ADD A,@Ri	Adds the indirect RAM to the accumulator
ADD A,#data	Adds the immediate data to the accumulator
ADDC A,Rn	Adds the register to the accumulator with acarry flag
ADDC A,direct	Adds the direct byte to A with a carry flag
ADDC A,@Ri	Adds the indirect RAM to A with a carry flag
ADDC A,#data	Adds the immediate data to A with carry a flag
SUBB A,Rn	Subtracts the register from A with a borrow
SUBB A,direct	Subtracts the direct byte from A with a borrow
SUBB A,@Ri	Subtracts the indirect RAM from A with a borrow
SUBB A,#data	Subtracts the immediate data from A with a borrow
INC A	Increments the accumulator
INC Rn	Increments the register
INC direct	Increments the direct byte
INC @Ri	Increments the indirect RAM
DEC A	Decrements the accumulator
DEC Rn	Decrements the register
DEC direct	Decrements the direct byte
DEC @Ri	Decrements the indirect RAM
INC DPTR	Increments the data pointer
MUL A,B	Multiplies A and B
DIV A,B	Divides A by B
DA A	Decimal adjust in accumulator

The arithmetic instruction has no knowledge about the data format i.e. signed, unsigned, ASCII, BCD, etc. Also, the operations performed by the arithmetic instructions affect flags like carry, auxiliary carry, overflow, zero, etc. in the PSW Register.

6.2.3 LOGICAL INSTRUCTIONS

The next group of instructions are the Logical Instructions, which perform logical operations like AND, OR, XOR, NOT, Rotate, Clear and Swap. Logical Instructions are performed on Bytes of data on a bit-by-bit basis. Instructions associated with the Logical operations are as follows:

ANL A,Rn	AND register to accumulator
ANL A,direct	AND direct byte to accumulator
ANL A,@Ri	AND indirect RAM to accumulator
ANL A,#data	AND immediate data to accumulator

ANL direct,A	AND accumulator to direct byte
ANL direct,#data	AND immediate data to direct byte
ORL A,Rn	OR register to accumulator
ORL A,direct	OR direct byte to accumulator
ORL A,@Ri	OR indirect RAM to accumulator
ORL A,#data	OR immediate data to accumulator
ORL direct,A	OR accumulator to direct byte
ORL direct,#data	OR immediate data to direct byte
XRL A,Rn	Exclusive OR register to accumulator
XRL A,direct	Exclusive OR direct byte to accumulator
XRL A,@Ri	Exclusive OR indirect RAM to accumulator
XRL A,#data	Exclusive OR immediate data to accumulator
XRL direct,A	Exclusive OR accumulator to direct byte
XRL direct,#data	Exclusive OR immediate data to direct byte
CLR A	Clears the accumulator
CPL A	Complements the accumulator
RL A	Rotates the accumulator left
RLC A	Rotates the accumulator left through carry
RR A	Rotates the accumulator right
RRC A	Rotates the accumulator right through carry
SWAP A	Swaps nibbles within the accumulator

6.2.4 BOOLEAN OR BIT MANIPULATION INSTRUCTIONS

As the name suggests, Boolean or Bit Manipulation Instructions will deal with bit variables. As discussed earlier, there is a special bit-addressable area in the RAM and some of the Special Function Registers (SFRs) are also bit addressable.The instructions corresponding to the Boolean or Bit Manipulation are listed below;

CLR C	Clears the carry flag
CLR bit	Clears the direct bit
SETB C	Sets the carry flag
SETB bit	Sets the direct bit
CPL C	Complements the carry flag
CPL bit	Complements the direct bit
ANL C,bit	AND direct bit to the carry flag
ANL C,bit	AND complements of direct bit to the carry
ORL C,bit	OR direct bit to the carry flag
ORL C,bit	OR complements of direct bit to the carry
MOV C,bit	Moves the direct bit to the carry flag
MOV bit,C	Moves the carry flag to the direct bit

6.2.5 PROGRAM BRANCHING INSTRUCTIONS

The Program Branching Instructions includes conditional and unconditional jumps, subroutine call, return and no operation. These instructions control the flow of program execution. The Branching Instructions are as follows;

ACALL addr11	Absolute subroutine call
LCALL addr16	Long subroutine call
RET Return	Return from subroutine
RETI Return	Return from interrupt
AJMP addr11	Absolute jump
LJMP addr16	Long jump
SJMP rel	Short jump (relative address)
JMP @A + DPTR	Jump indirect relative to the DPTR
JZ rel	Jump if accumulator is zero
JNZ rel	Jump if accumulator is not zero
JC rel	Jump if carry flag is set
JNC rel	Jump if carry flag is not set
JB bit,rel	Jump if direct bit is set
JNB bit,rel	Jump if direct bit is not set
JBC bit,rel	Jump if direct bit is set and clears bit
CJNE A,direct,rel	Compares direct byte to A and jumps if not equal
CJNE A,#data,rel	Compares immediate to A and jumps if not equal
CJNE Rn,#data rel	Compares immediate to the register and jumps if not equal
CJNE @Ri,#data,rel	Compares immediate to indirect and jumps if not equal
DJNZ Rn,rel	Decrements register and jumps if not zero
DJNZ direct,rel	Decrements direct byte and jumps if not zero
NOP	No operation

All these instructions, except the NOP (No Operation) affect the Program Counter (PC) in one way or other. Some of these instructions have decision making capability before transferring control to other part of the program.

6.3 PROGRAMMING OF 8051

1. Write a program to add the contents of locations 50H and 51H and store the result in locations in 52H and 53H.

```
ORG 0000H             ; Set program counter 0000H
```

```
MOV A,50H          ; Load the contents of Memory location
                   50H into A
ADD A,51H          ; Add the contents of memory 51H with
                   content of A
MOV 52H,A          ; Save the LS byte of the result in 52H
MOV A, #00         ; Load 00H into A
ADDC A, #00        ; Add the immediate data and carry to A
MOV 53H,A          ; Save the MS byte of the result in location
                   53H
END
```

2. Write a program to subtract a 16 bit number stored at locations 51H-52H from 55H-56H and store the result in locations 40H and 41H. Assume that the least significant byte of data or theresult is stored in low address. If the result is positive, then store 00H, else store 01H in 42H.

```
ORG 0000H          ; Set program counter 0000H
MOV A, 55H         ; Load the contents of memory location 55
                   into A
CLR C              ; Clear the borrow flag
SUBB A,51H         ; Sub the contents of memory 51H from
                   contents of A
MOV 40H, A         ; Save the LSByte of the result in location
                   40H
MOV A, 56H         ; Load the contents of memory location
                   56H into A
SUBB A, 52H        ; Subtract the content of memory 52H from
                   the content A
MOV 41H,           ; Save the MSbyte of the result in location
                   41H.
MOV A, #00         ; Load 00 into A
ADDC A, #00        ; Add the immediate data and the carry flag
                   to A
MOV 42H, A         ; If result is positive, store00H, else store
                   0lH in 42H
END
```

3. Write a program to multiply two 8 bit numbers stored at locations 70H and 71H and store the result at memory locations 52H and 53H. Assume that the least significant byte of the result is stored in low address.

```
ORG 0000H     ; Set program counter 00 OH
MOV A, 70H  ; Load the contents of memory location 70h into A
MOV B, 71H  ; Load the contents of memory location 71H into B
MUL AB        ; Perform multiplication
```

```
        MOV 52H,A    ; Save the LS byte of the result in location 52H
        MOV 53H,B   ; Save the MS byte of the result in location 53 H
        END
```

4. Find the content of register A after the following instructions are executed

```
        MOV A, #36H         :       A = 36H – 0011 0110
        ANL A, # 28H        ;       A = A AND 28H – 0010 00110 AND
0010 1000
                                    A = 20H – 0010 0000
```

5. Find the value of the register A after the following instructions are executed

```
        MOV A, #62H         :       A = 62H – 0110 0010
        ORL A, # 39H        :       A = A OR 39H – 0110 0010 or 0011
1001
                                    A = 7BH – 0111 1011
```

UNIT –VII INTRODUCTION TO INTERFACING DEVICES

Structure

7.1 INTRODUCTION

Any application which is based on microprocessor involves several peripheral devices connected together with a microprocessor as a single system. To facilitate this, the microprocessor should essentially be designed with the features of accommodating these peripherals and to communicate with them by sharing data. The basic function of interfacingis that the microprocessor should be able to read fromand/or write into memory chip or input/output devices. Therefore the microprocessor has to identify a peripheral device through an addressand enable input or output buffer as to read orwrite to that device.

Address space may refer to a range of either physical or virtual addresses accessible to a processor or reserved for a computational entity, such as a device, a file, a server, or a networked computer process. As unique identifier each address specifies an entity's location or unit of memory. A processor's address space is always limited by the width of its address bus and registers. Address space may be differentiated as either single, in which addresses are expressed as incrementally increasing integers starting at zero to maximum, or segmented, in which addresses are expressed as separate segments augmented by offsets (values added to the base index to produce secondary addresses).

7.2 ADDRESS SPACE PARTITIONING

Intel 8085 uses a 16-bit wide address bus or addressing memory and I/O devices. It can access 2^{16}=64k bytes of memory and I/O devices. There are two schemes for the allocation of address to memories or I/O devices.

1. Memory mapped I/O scheme
2. I/O mapped I/O scheme

7.2.1 MEMORY MAPPED I/O SCHEME

This scheme has only one address space which covers all possible addresses that a microprocessor can generate. A distinct portion of the space is assigned to memories and I/O devices. One unique address is assigned to either memory location or I/O device. The main advantage of this scheme is that all the data transfer instructions of the microprocessor can be used for both memory as well as I/O devices since each entity has distinct address. For example, MOV A, M will be valid for data transfer from the memory location or I/O device whose address is in H-L pair. This scheme is suitable for small system which involves limited number of memories and I/O devices.

7.2.2 I/O MAPPED I/O SCHEME

This scheme has common address space for both memory and I/O devices. Identification of memory and I/O device is done using a special signal IO/M in addition to the address bus. When the signal is high the address carried by the address bus is for I/O device and the low state of the signal refers for a memory location. This scheme is suitable for large system since the entire address space can be utilized to handle large number of memory and I/O devices. Additionally, instructions IN and OUT are used to address the I/O device for readingand writing the data.

The following table list outs the differences between Memory mapped I/O and I/O mapped I/O

Memory mapped I/O	I/O mapped I/O
Each address of the I/O port and memory location is unique	I/O and memorylocations share same address space
Considerable memory location are allotted for I/O ports where instruction and data cannot be stored	No such problem.
The I/O ports have 16 bit address. Therefore 64k is shared between I/O system and memory system	The port address will be only 8 bit. Therefore 256 separate I/O devices can be connected. The I/O map is independent of memory map.
Data transfer between any CPU register and I/O ports are possible.	Data transfer is possible only through A register to I/O ports.
Instructions like LDA,STA, MOV M, R, MOV R,M instructions can be used to transfer data to any I/O device. I/O devices are not explicit in the program.	Only IN, OUT instructions are used to transfer data. I/Ooperations are very explicit.
Main memory space is limited. Program debugging is not easy.	Separate memory is used for I/O ports Instructions. Program debugging is easy.

7.3 INTERFACING DEVICES

Interface is the path for making communication between two components. Interfacing is of two types, memory interfacing and I/O interfacing.

7.3.1 MEMORY INTERFACING

Memory Interfacing refers the provision of a decoder which decodes the address sent by the microprocessor to identify a memory location in order to read or write the content in that memory location by the microprocessor. If the decoder decodes the address sent by the microprocessor to identify an I/O device, then it is called I/O interfacing. The application of an interfacing device 74LS138, a 1 to 8 lines decoder is described in this section. The pin diagram of 74LS138 is given below;

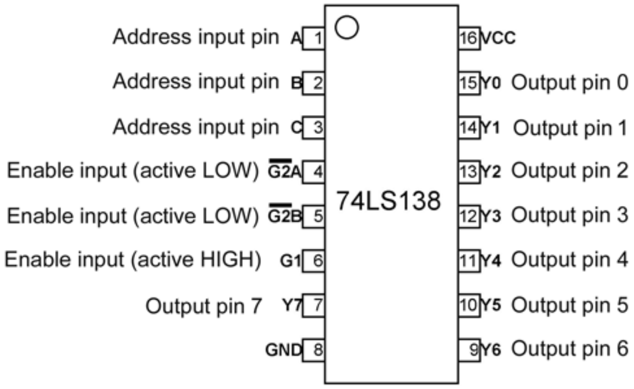
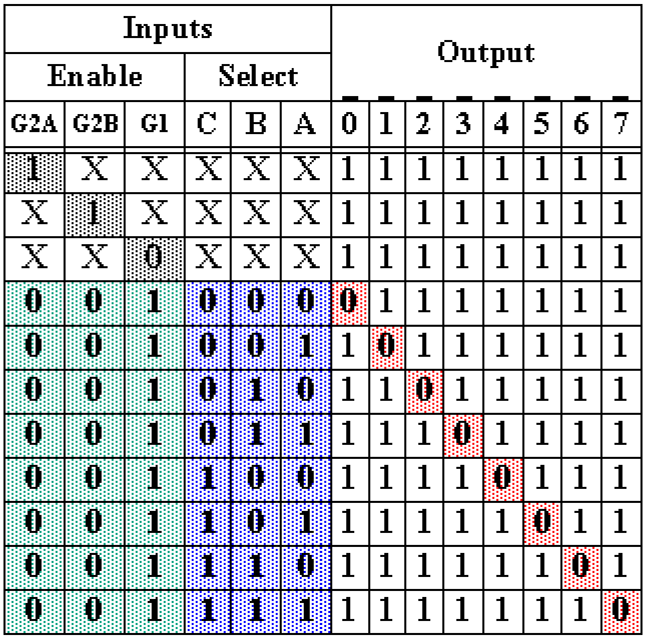


Figure 7.1 Pin diagram of 74LS138

Pins 1-3 are address pins. Pins 4-6 are enable pins to activate the decoder. Pin 8 is for circuit ground and pin 16 is meant for Vcc. Pins 7 and 9-15 are output pins to locate the memory. For interfacing this decoder with a microprocessor for accessing the memory, the pins G1 should be high and both G2B and G2A should be low provided 16 and 8 should be connected to the power source. Pins A,B and C are used for feeding the address. Based on the address or the input to A, B and C any one of the 8 outputs will be selected. The truth table for the decoder is given below.



Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Figure 7.2Truth table for the decoder

In the above truth table, the first three rows are invalid and the remaining addresses can be used to decode any one of the output by properly connecting this outputs to memories.

7.3.2 I/O INTERFACING

Thevarious communication devices like the keyboard, mouse, printer, etc. are to be interfaced with a microprocessor in order to realize as a single system. By using a decoder like 74LS138, one can interface 8 such devices to a microprocessor by assigning each device with one address. In order to realize this, another 74LS138 chip is to be interfaced with proper chip select option to decode either the memory or the I/O device. A scheme to achieve this feature is to utilize the IO/$\overline{M}$ signal of the microprocessor in addition to the address inputs to activate the decoder either connected to memory or to the I/O device. A simple schematic scheme given below explains this interface feature.

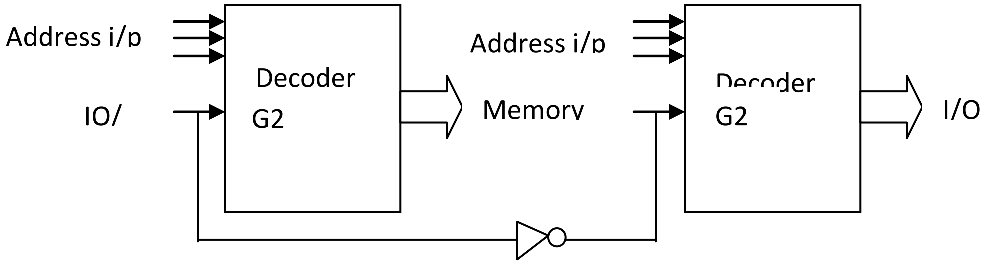


Figure 7.3 Scheme for I/O interface

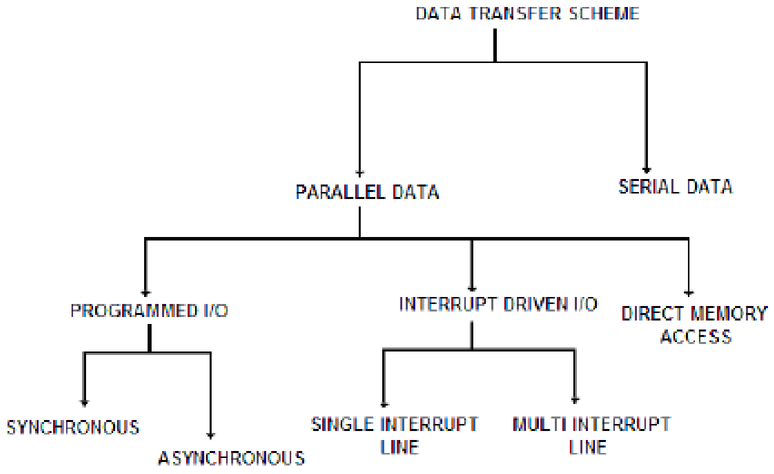
7.4 DATA TRANSFER SCHEMES

In an 8085 microprocessor based system several input and output devices are integrated. The data transfer between these different components is the key factor for successful application as a system. The data transfer may take place between microprocessor and memory, microprocessor and I/O devices and memory and I/O devices. Each of these components may have different technical characteristics including speed of operation which affects the data transfer between them. In this context, a special data transfer schemes and devices are very much essential to solve the mismatch between a microprocessor and several of its peripheral devices particularly I/O devices since the speed of the memory is almost compatible with the speed of 8085 microprocessor.

Several data transfer schemes of 8085 microprocessor are introduced for smooth data transfer. The data transfer schemes of 8085 microprocessor were categorized depending upon the capabilities of the I/O devices for accepting serial or parallel data format. The following figure represents the various data transfer schemes.

The 8085 microprocessor is a parallel device. That means it transfers eight bits of data simultaneously over eight data lines (parallel I/O mode).

However in many situations, the parallel I/O mode needs special devices to



implement.

Figure 7.4 Data transfer scheme

7.4.1 SERIAL I/O MODE TRANSFER

In serial I/O mode transfer, a single bit of data on a single line at a time is transferred. For serial I/O data transmission mode, 8-bit parallel word is converted to a stream of eight serial bit using parallel-to-serial converter. Similarly, in serial reception of data, the microprocessor receives a stream of 8-bit one by one which are then converted to 8- bit parallel word using serial-to-parallel converter. Here the microprocessor is able to process the data in parallel mode.

7.4.2 PARALLEL DATA TRANSFER SCHEME

Parallel data transfer scheme is faster than serial I/O transfer. In parallel data transfer, 8-bit data are transferred all together on 8 parallel wires. For 8085 microprocessor, mainly three types of parallel data transfer schemesare observed. Those are

a. Programmed data transfer scheme

b. Direct Memory Access (DMA) data transfer scheme

7.4.2.1 PROGRAMMED DATA TRANSFERS

Program data transfer scheme involves CPS under the control of programs which reside in the memory. These programs are executed by the CPU when an I/O device is ready to transfer data. This scheme is beneficial when a small amount of data is transferred at the rate of one byte of data at a time with slow I/O devices. Programmed data transfer can be further classified as,

1. Synchronous data transfer.
2. Asynchronous data transfer.
3. Interrupt driven transfer.

1. Synchronous data transfer:

This data transfer happens “at the same time” (synchronous) between the microprocessor and the communicating device. The communicating device may be of memory or I/O device. The I/O devices compatible with microprocessor in speed are usually very limited but compatible memories are available. Whenever data is to be obtained from the device or transferred to the device, the user program may issue suitable instruction addressing the device along with data. At the end of the execution of this instruction, the data transfer would have been completed. Based on the speed of transfer, the technique of data transfer is rarely used for I/O devices but invariably used with compatible memory devices. This is the simplest of all data transfer schemes. If an output device connected to the 8085in memory mapped mode, the following simple instruction may be used for transferring the content of the accumulator to the memory location already referred in the H_L pair registers. If the device is connected in I/O mapped mode, then the OUT instruction may be issued. For example OUT 2 will copy the content of A to the device attached to the port whose address is 2. To get the data from the port, IN 2 or MOV A, M instruction may very well be used. Here M represents the address 2. There are two data transfer synchronizing techniques are available. They are polling and Interrupts.

2. Asynchronous data transfers:

Asynchronous means “at irregular intervals”. The speeds of the microprocessor and the I/O device do not match; asynchronous data transfer may be used. Under this scheme, first the microprocessor issues get ready instruction to the device; subsequently the microprocessor waits and checks for reply until the device is ready. The device will send a ready signal to the microprocessor and upon receiving the ready signal by the microprocessor;the data transfer will be initiated. This form of data transfer is also known as Hand shaking; since some signals (hand shake signals) are exchanged between the I/O device and the microprocessor before the actual data transfer takes place. The following figure shows a schematic diagram for asynchronous data transfer.Asynchronous data transfer is used for slow I/O device but the drawback is that the precious time of the µp is wasted in waiting.

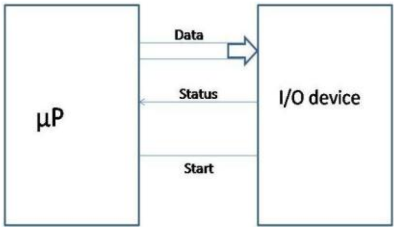


Figure 7.5 Asynchronous data transfer

3. Interrupt driven data transfer:

In this scheme, the data transfer occurs after the I/O device issued a special signal called an interrupt to the microprocessor. The main advantage of this scheme is that the microprocessor does not need to wait by wasting its precious time while checking the readiness of the I/O device. After receiving the interrupt from the I/O device, the microprocessor completesthe current instruction in hand and saves the content of the program counter on the stack first, and then takes up a subroutine called ISS(Interrupt Service Subroutine) of the I/O device which interrupted the microprocessor.The execution of the ISS is to transfer data from or to the ISS device. Different ISSs are to be provided for different I/O devices. After completing the data transfer, the control of the microprocessor returns back to the next instruction in the main program where it diverted the flow of execution due to the interruption by the I/O device. Interrupt driven data transfer is used for slow I/O devices. For example, microprocessor based A/D or D/A convertion.

7.4.2.2 DMA DATA TRANSFER SCHEME

In DMA data transfer scheme CPU does not participate. Data are directly transferred from an I/O device to the memory device or vice versa. The data transfer is controlled by the I/O device or a DMA controller. When a large block of data is to be transferred, DMA is used. If bulk of data are transferred through CPU, it takes appreciable time and the process becomes slow. An I/O device which wants to send data using DMA technique, sends a HOLD signal to the microprocessor, then the CPU frees the control of buses as soon as the current machine cycle is completed. The CPU sends HOLD acknowledgement signal to that I/O device to indicate that it has received the HOLD request and it will give up the buses in the next machine cycle. The I/O device takes over the control of buses and initiates data transfer directly to the memory or reads data from the memory. DMA transfer scheme is a faster scheme as compared to programmed data transfer scheme. It is used to transfer data from mass storage devices such as hand disks, floppy disks etc, It also used for high-speed printers.

DMA data transfer schemes are of the following three types.

i. Burst modeDMA data transfer

ii. Cycle stealing technique of DMA transfer

iii. Demand transfer mode DMA.

Burst mode DMA data transfer

If the I/O device withdraws the DMA request only after all the data bytes have been transferred is called burst mode data transfer. By this technique a block of data is transferred. Incase of magnetic disks,block transfer of data is a must since transfer cannot be stopped or slowed down in the middle without loss of data.

Cycle stealing techniques DMA transfer

In this technique, DMA data transfer happens at different intervals based on the availability of the bus without much interference in CPU's process. After transferring one byte or several bytes, the I/O device withdraws the DMA request and steals the bus cycle through an interfacing circuitry when the CPU is not using the system bus. Data is transferred by a sequence of DMA cycles.

Demand transfer mode DMA

The bulk of data transfer takes place whenever there is a demand created by the I/O device or memory device. The following figure shows some various types of data transfer schemes.

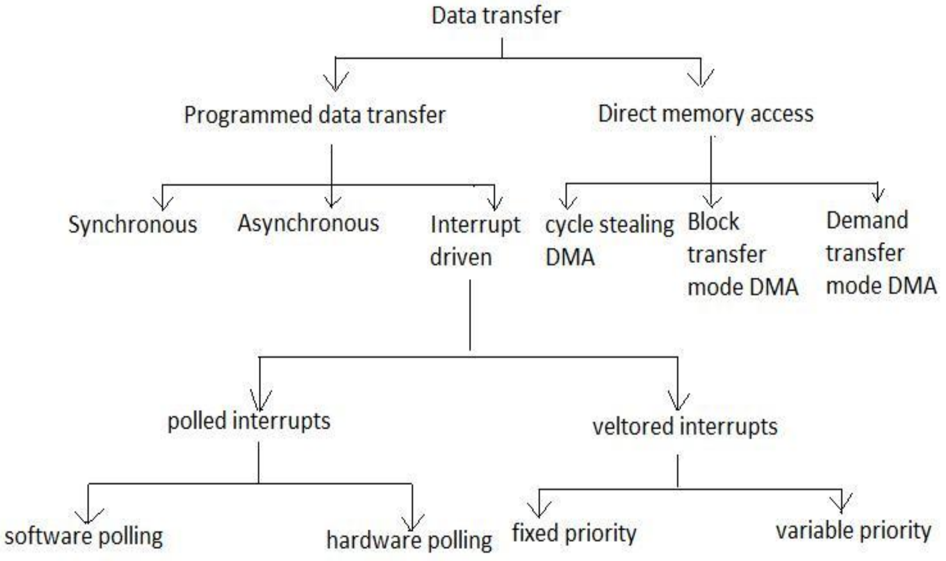


Figure 7.6 Types of data transfer schemes

7.5 INTERRUPTS

An interrupt is a condition that causes the microprocessor to temporarily work on a different task, and then later return to its previous task. Interrupts can be internal or external. Internal interrupts, or "software interrupts," are triggered by a software instruction and operate similarly to a jump or branch instruction. An external interrupt, or a "hardware interrupt," is caused by an external hardware module. As an example, many computer systems use interrupt driven I/O, a process where pressing a key on the keyboard or clicking a button on the mouse triggers an interrupt. The processor stops what it is doing, it reads the input from the keyboard or mouse, and then it returns to the current program.

7.6 I/O PORTS

I/O stands for Input and Output. The information given into the microprocessor is known as input. The most common device for input is the keyboard or a signal from sensor. The information generated by the computer is called output. The most common device for output is either a display or a printer. The buffer where the information are shared from or to the microprocessor is called I/O port. An input port is a place for unloading data by the peripheral and read by the microprocessor. An output is a place for unloading the data by the microprocessor and read by the peripherals. If different data formats are being exchanged, the interface must be able to convert serial data to parallel form and vice-versa. The following figure shows a schematic connection of the CPU, I/O ports and I/O devices.

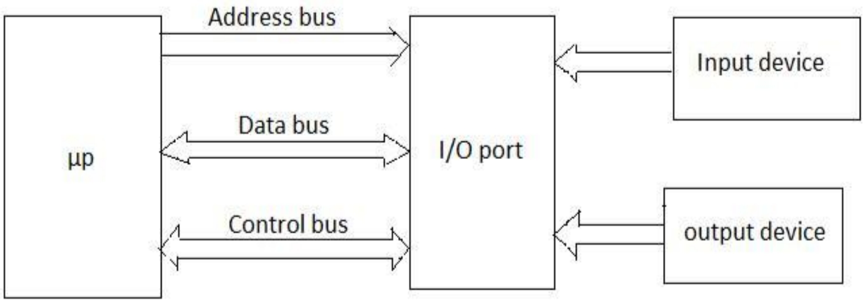


Figure 7.7 Interfacing of I/O device through I/O port

An I/O port may be programmable or non-programmable. A non-programmable port behaves as an input port if it is been designed and connected in the input mode. Similarly a port connected in the output mode acts as an output port. But a programmable I/O port can be programmed through software to act either as an input port or as an output ports besides the electrical connections remain same.

UNIT – VIII INTERFACING DEVICES (8255 AND 8259)

Structure

8.1 PROGRAMMABLE PERIPHERAL INTERFACE (PPI)

A programmable peripheral interface is a multiport device. The ports may be programmed in a variety of ways as required by the programmer. The device is very useful for interfacing various peripheral devices to the microprocessor. Intel 8255 is a programmable peripheral interface (PPI).

8.2 INTEL 8255

The Intel 8255 is a programmable peripheral interface (PPI). It has two versions namely, the Intel 8255A and the Intel 8255A-5. General descriptions for both are same. There are some differences in their electrical characteristics. It’s main functions are to interface peripheral devices to the microprocessor. It has 3 numbers of 8-bit ports, namely port A, port B and port C. The port C is further divided into two 4-bits ports, port C upper and port C lower. These are given as port C lower i.e., PC3 – PC0 and port C upper i.e., PC7 – PC4. And are arranged in group of 12 pins each thus designated as Group A and Group B.Thus, a total of 4 ports in 8255. Each port can be programmed either as an input port or as an output port.

The two operating modes in which 8255 can be programmed are as follows:

i.Bit set/reset (BSR) mode

ii.I/O mode

The bits of port C gets set or reset in the BSR mode. The other mode of 8255 i.e., I/O mode is further classified into:

Mode 0: Simple input/output

Mode 1: Input output with handshaking

Mode 2: Bidirectional I/O handshaking

Mode 0 and Mode 1 both are same but the only difference is mode 1 does not support bidirectional handshaking.This means if 8255 is programmed to mode 1 input, then it will particularly be connected to an input device and performs the input handshaking with the processor.

But if it is programmed to mode 2 then due to bidirectional nature, the PPI will perform both input and output operation with the processor according to the command received.

8.3 CONTROL WORD

According to the requirement, a port can be programmed to act either as an input port or an output port. For programming the ports of 8255 a control word is formed. The bits of control word are shown in the following figure.

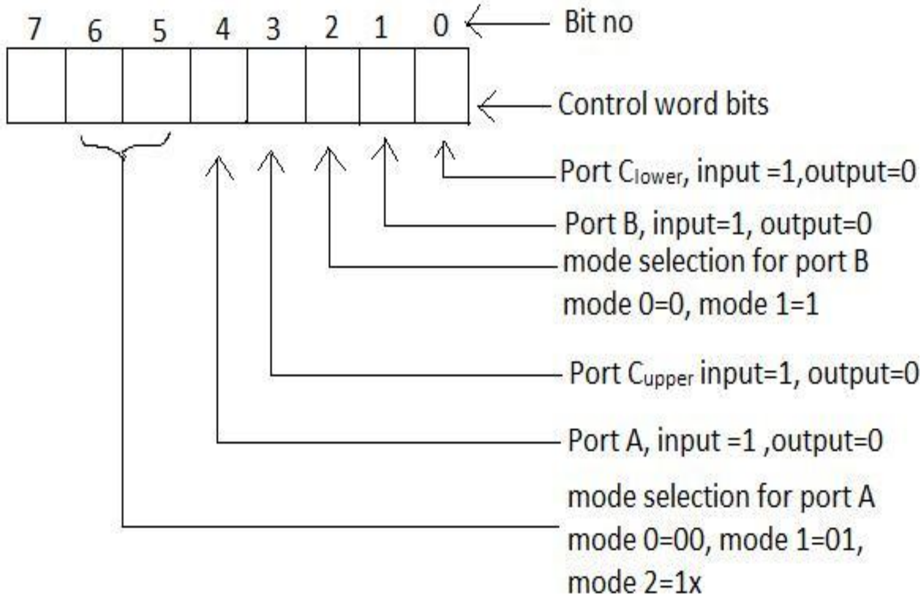

7 6 5 4 3 2 1 0
Bit no
Control word bits
Port C_{lower}, input =1,output=0
Port B, input=1, output=0
mode selection for port B
mode 0=0, mode 1=1
Port C_{upper} input=1, output=0
Port A, input =1 ,output=0
mode selection for port A
mode 0=00, mode 1=01,
mode 2=1x

Figure 8.1 Control word bits for Intel 8255

Control word is written into the control word register which is within 8255 by using the instruction OUT followed by the address of the port.

Bit No 0:

It is for port $c_{lower.}$ To make port C_{lower} an input port, the bit is set to 1.To make port C_{lower} an output port, the bit is set to 0.

Bit No 1:

It is for port B. To make port B and input port, the bit is set to 1.

To make port b an output port, the bit is set to 0.

Bit No 2:

It is for the selection of the mode for the port B. If the port B has to operate in mode 0, the bit is set to 0. For mode 1 operation of the port B, it is set to 1.

Bit No 3:

It is for the port $C_{upper.}$ To make port C_{upper} an input port, the bit set to 1.

To make port C_{upper} an output port, the bit is set to 0.

Bit No 4:

It is for the port A. To make portA, an input port, the bit set to 1.

To make port A, an output port, the bit is set to 0.

Bit No 5 and 6:

These bits are to define the operating mode of the port A. For the various modesof port A these bits can defined as follows:

Mode of Port A	Bit no 6	Bit no 5
Mode 0	0	0
Mode 1	0	1
Mode 2	1	0 or 1

Bit No 7: It is set to 1 if ports A,B and C are defined as input/output ports. It is set to 0 if the

individual pins of the port C are to be set or reset.

Examples:

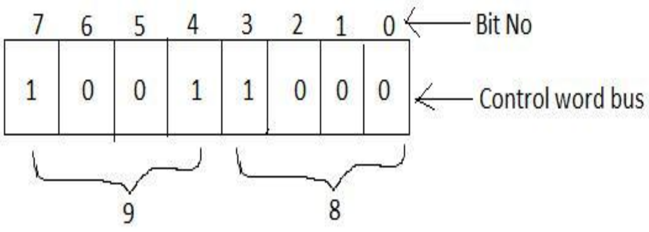
The following example illustrates how to make a control word

Make control word when the ports of Intel/8255 are defined as follows;

Port A as an input port. Mode of the port A – mode 0, Port B as an output port, Mode of the port B – mode 0, Mode of the port B – mode 0, Port C_{upper} as an input port and Port C_{lower} as an output port.

Solution:

The control word bits for the above definition of the ports are as follows:



The control word =98H

Bit No.0 is set to 0, as the port C_{lower} is an output port.

Bit No.1 is set to 0, as the port B is an output port.

Bit No.2 is set to 0, as the port B has to operate in mode 0.

Bit No.3 is set to 1, as the port C_{upper} is an input port.

Bit No.4 is set to 1 , as the port a is an input port.

Bit No5 and 6 are set to 00 as the port A has to operate in mode 0.

Bit No 7 is set to 1, as the ports of A ,B and C are used as simple input/output port.

Thus the control word= 98H.

8.4 INTERNAL ARCHITECTURE OF INTEL 8255

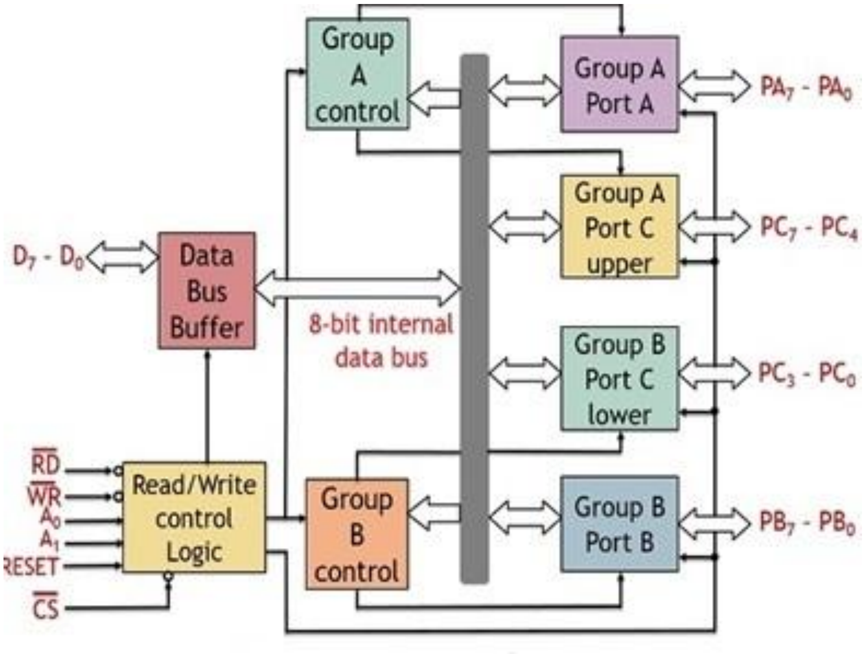


Figure 8.2 Architecture of 8255

The operation performed by each unit/signal is discussed below;

Data bus buffer: It is used to connect the internal bus of 8255 with the system bus so as to establish proper interfacing between the two. The data bus buffer allows the read/write operation to be performed from/to the CPU.The buffer allows the passing of data from ports or control register to CPU in case of write operation and from CPU to ports or status register in case of read operation.

Read/ Write control logic: This unit manages the internal operations of the system. This unit holds the ability to control the transfer of data and control or status words both internally and externally.Whenever there exists a need for data fetch then it accepts the address provided by the processor through the bus and immediately generates command to the control groups (A and B) for the particular operation.

Group A and Group B control: These two groups are handled by the CPU and functions according to the command generated by the CPU. The CPU sends control words to the group A and group B control and they in turn sends the appropriate command to their respective port. Group A has the access of the port A and higher order bits of port C while group B controls port B and the lower order bits of port C.

$\overline{CS}$: It is for chip select. A low signal at this pin enabling the communication between the 8255 and the processor so that the data transfer operation gets enabled.

$\overline{RD}$ – It is for read operation. A low signal at this pin shows that CPU is performing read operation at the ports or status word or 8255 is providing data or information to the CPU through data buffer.

$\overline{WR}$ – It is for write operation. A low signal at this pin allows the CPU to perform write operation over the ports or control register of 8255 using the data bus buffer.

A0 and A1: Used to select the desired port among all the ports of the 8255 by forming conjunction with RD and WR signals.

The table below shows the operation of the control signals for the selection of ports.

Port / control word register	Port / control word register address
Port A	00
Port B	01
Port c	02
Control word register	03

The instruction IN00 means that it is for the port A of 8255. When this instruction is executed data are transferred from the port A to the accumulator. The instruction OUT 03 will transfer the content of the accumulator to the control word register of 8255.

Reset: It is an active high signal and a high signal at this pin clears the control registers and the ports which are set in the input mode.

During the reset, initializing the ports to input mode alone is done to prevent the accidental circuit breakdown at the output mode since there exist chances of electrical destruction of 8255 along with the processor due to the physical interface between the output ports with the microprocessor.

8.4.1 PIN CONFIGURATION OF 8255

The following picture represents the pin configuration of PPI 8255. The pins for various ports are PA0-PA7 of Port A, PB0-PB7 of Port B, PC0-

PC3 of Port C_{lower} and PC4-PC7 of Port C_{upper}. Vcc is input power and GND is circuit ground.

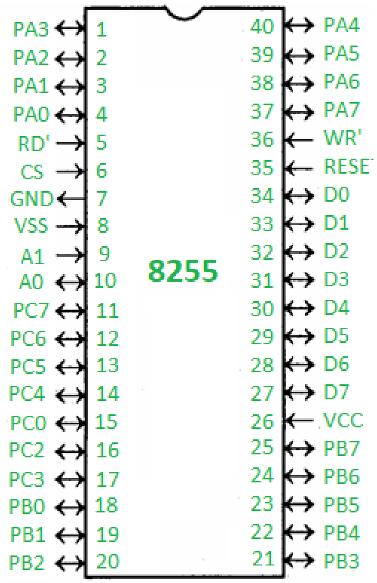


Figure 8.3Pin configuration of 8255

8.5 PROGRAMMABLE INTERRUPT CONTROLLER (PIC)

The programmable interrupt controllers is used when several I/O devices are to transfer data using interrupt and they are to be connected to the same interrupt level of the microprocessor. Particularly when the number of the I/O devices is more than the number of interrupt levels of the microprocessor, such controllers are required. The Intel 8259 is a single chip programmable interrupt controller. It is compatible with 8085 and 8086 microprocessors. It is a 28 pin DIP IC package and uses N-MOS technology. It receives 8 interrupt inputs (IR0-IR7) and send a single interrupt (INT) as output to the microprocessor. It requires a single +5V supply for its operation. The figure shows the schematic diagram of Intel 8259.

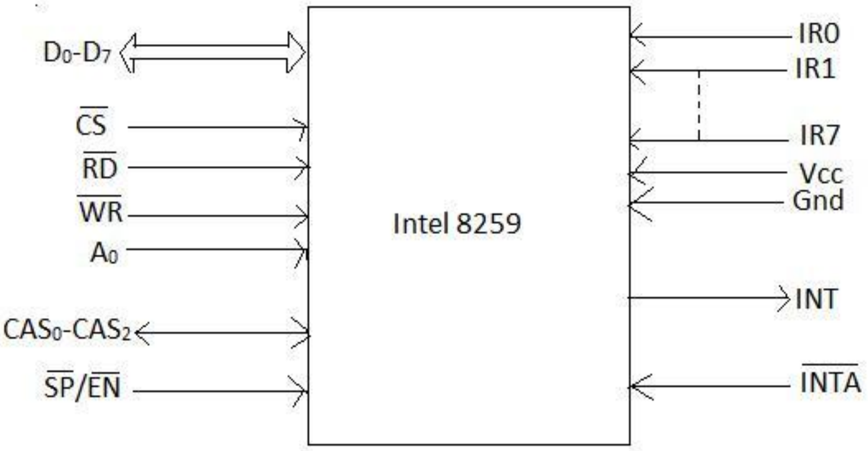


Figure 8.4 Schematic diagram of Intel 8259

8.6 INTERNAL ARCHITECTURE OF 8259

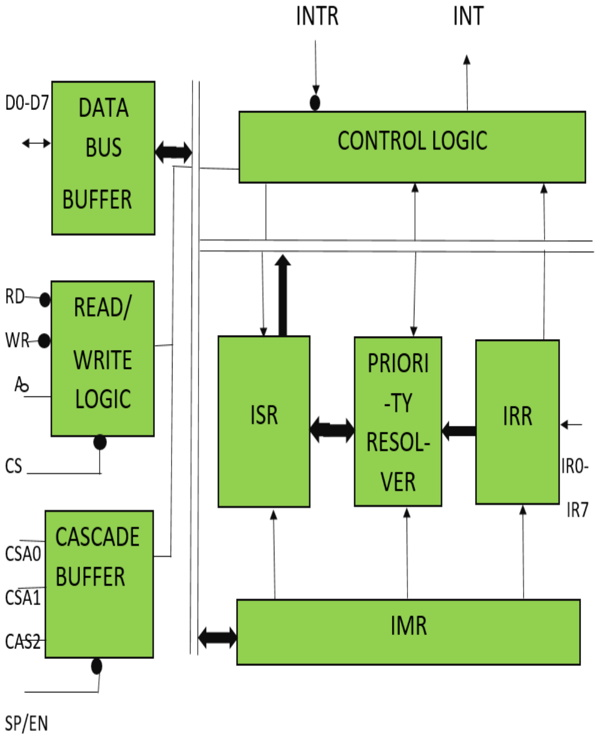


Figure 8.5 Internal architecture of 8259

The block diagram consists of Data Bus Buffer, Read/Write Logic, Cascade Buffer, Control Logic and Priority Resolver blocks and 3 registers namely ISR, IRR, IMR. The nature of the each block and register are explained.

Data bus buffer

It is a mediator between 8259 and 8085 or 8086 microprocessor as a buffer. It takes the control word from the microprocessor and transfers it to the control logic of 8259. Also, after selection of Interrupt by 8259, it transfer the opcode of the selected Interrupt and address of the Interrupt service sub routine (ISR) to the connected microprocessor. The data bus buffer consists of 8 bits data bus as represented as D0-D7 in the block diagram. A maximum of 8 bits data can be transferred at a time.

Read/Write logic

This block works only when the value of pin CS is low (as this pin is active low). This block is responsible for the flow of data depending upon the inputs of RD and WR. These two pins are active low pins used for read and write operations.

Control logic

It controls the functioning of every block. It input pin INTR receives interrupt request from microprocessor and enable pin INT for output to the microprocessor. If 8259 is enabled, and the connected microprocessor's interrupt is high then this causes the value of the 8259's output INT pin high and in this way 8259 responds to the request made by microprocessorinterfaced with it.

Interrupt request register (IRR)

It stores all the interrupt levels which are requesting for Interrupt services.

Interrupt service register (ISR)

It stores the interrupt level which is currently being executed.

Interrupt mask register (IMR)

It stores the interrupt levels which have to be masked by storing the masking bits of that interrupt level.

Priority resolver

It examines all the three registers and set the priority of interrupts.According to the priority of the interrupts, interrupt with highest priority is set in ISR register. Also, it reset the interrupt level which is already been serviced in IRR.

Cascade buffer

It is used to increase the Interrupt handling capability by employing CSA lines to control multiple interrupt structure.

SP/EN (Slave program/Enable buffer) pin is set to high to work in master mode else in slave mode. In case of non-buffered mode SP/EN pin is used to specify whether 8259 work as master or slave and in buffered mode, SP/EN pin is used as an output to enable data bus.

The priority can be assigned to the I/O devices which are connected to PIC. 8 I/O devices can be connected to 8259 through IR0 − IR7 lines. In this interrupt driven system, the interrupt controller accepts request from an I/O device and determines which of the incoming request is of the highest priority. After checking the priority of the interrupt request, the 8259 sends an interrupt signal to the microprocessor through the INT line. Thenthe microprocessor sends the acknowledgesignal through INTA line and all the interrupt of lower priority are inhibited.The 8259 sends a unique CALL instruction to the microprocessor so that it can take up the ISS attached to the I/O device which has requested for data transfer. 8259 chips can be cascaded to receive upto 64 vectored priority interrupts without additional circuitry. The following schematic diagram represents a scheme for interfacing 8259 with 8085 or 8086.

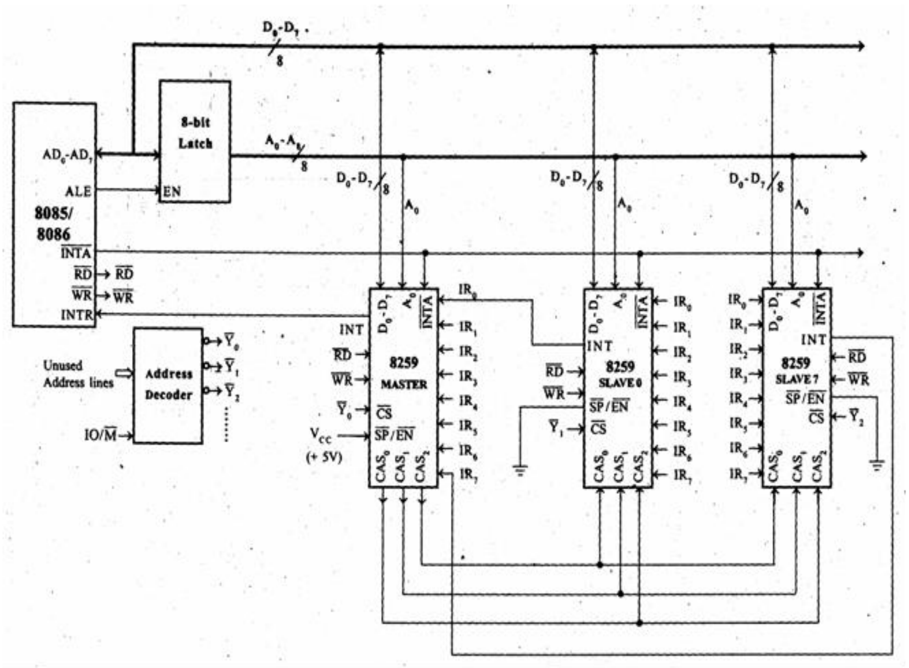


Figure 8.6 Scheme for interfacing 8259 with 8085

UNIT – IX INTERFACING DEVICES (8257 AND 8251)

Structure

9.1 INTRODUCTION TO DMA CONTROLLER

In direct memory access (DMA) data transfer schemes, data are directly transferred from I/O device to RAM or from RAM to I/O devices without any interference of the CPU. With the use of a DMA controller, the device sends a request to the CPU through HOLD signal to transfer data directly to/from the memory. The DMA data transfer is initiated only after receiving HLDA signal from the CPU.8257 is a 4-channel Direct Memory Access controller. It is specially designed by Intel for data transfer at the highest speed.

Following is the sequence of operations performed by a DMA −

- Initially, when any device has to send data between the device and the memory, the device has to send DMA request (DRQ) to DMA controller.
- The DMA controller sends Hold request (HRQ) to the CPU and waits for the CPU to declare the HLDA.
- Then the microprocessor tri-states all the data bus, address bus, and control bus. The CPU leaves the control over the bus and acknowledges the HOLD request through HLDA signal.
- Now the CPU is in HOLD state and the DMA controller has to manage the operations over the buses between the CPU, memory and I/O devices.

9.1.1 FEATURES OF 8257

Here is a list of some of the prominent features of 8257 −

- It has four DMA channels which can be used over four I/O devices.
- Each channel has 16-bit address and 14-bit counter.

- Each channel can transfer data up to 64Kb.
- Each channel can be programmed independently.
- Each channel can perform read, write and verify the transfer operations.
- It generates MARK signal to the peripheral device that 128 bytes have been transferred.
- It requires a single phase clock.
- It's frequency ranges from 250Hz to 3MHz.
- It operates in 2 modes, i.e., **Master mode** and **Slave mode**.

The more description about the features of 8259 are given as follows. Four I/o devices can be interfaced to the microprocessor through this device. It is capable of performing three operation namely read, write and verify. Transfer of data from the memory to the I/O device is read and transfer of data from the I/O device to the memory is write operation. On receiving a request from the I/O device, the 8257 generats a sequential memory address which allows the I/O device, to read or write directly to or from the memory. Each channel incorporates two 16- bit resisters namely,DMA address register and Byte count register.

These register are initialized with the address of the first memory location is to be accessed before a channel is enable. During DMA operation it stores the next memory locations is to be accessed in the next DMA cycle. 14-LSBs of the byte count registers store the number of bytes to be transferred. So, 2^{14}(16384) bytes of data can directly be transferred to the memory from the I/O device or from the memory to the I/O device. The 2 MSBs of the byte count register indicate the operation which is to be performed by the controller on that channel. Besides these registers the 8257 also includes mode set register and starter register.

9.1.2 ARCHITECTURE OF 8257

The following image shows the internal architecture of 8257 with details about the various functional blocks.

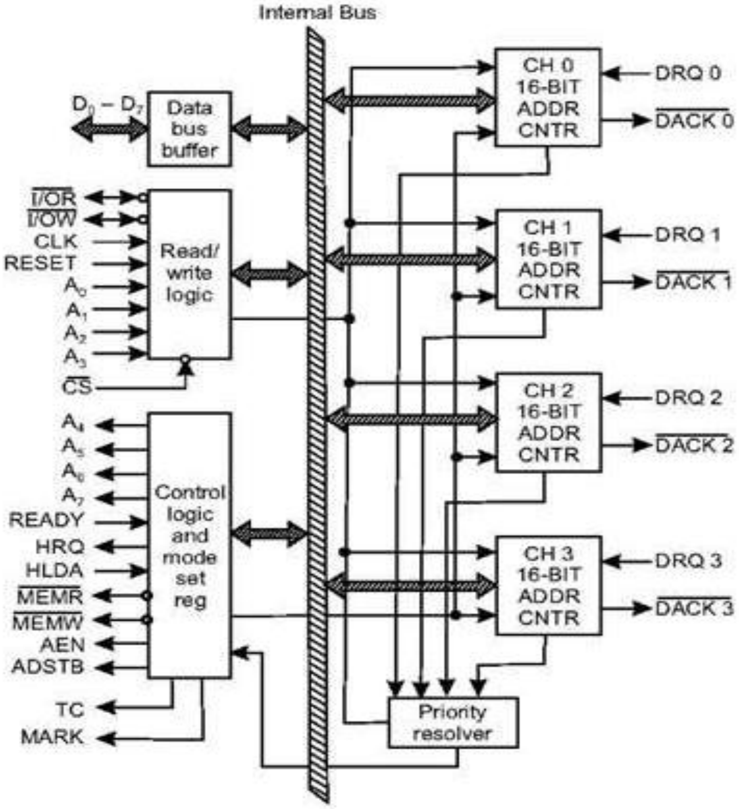


Figure 9.1 Architecture of 8257

The following image shows the pin diagram of the 8257 DMA controller.

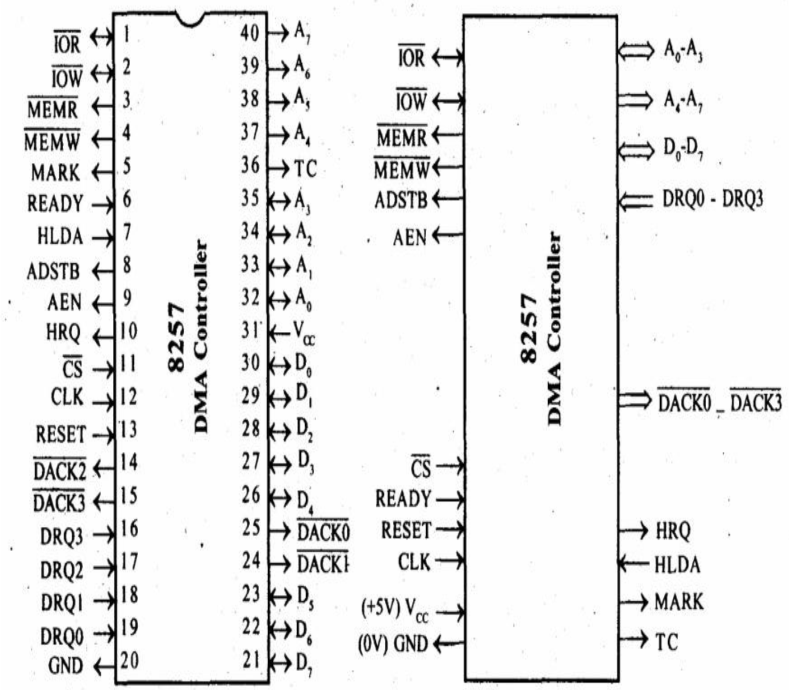


Figure 9.2Functional block of 8257

DRQ_0–DRQ_3

These are the four DMA channel request inputsused by the peripheral devices for DMA services. When the fixed priority mode is selected, then DRQ_0 has the highest priority and DRQ_3 has the lowest priority among them.

$\overline{DACK0} - \overline{DACK3}$

These are the active-low DMA acknowledge lines by 8257 against the request made by the peripherals. These lines can also act as strobe lines for the requesting devices.

D_o – D_7

These are bidirectional, data lines which are used to interface the system bus with the internal data bus of DMA controller. In the Slave mode, it carries command words to 8257 and status word from 8257. In the master mode, these lines are used to send higher byte of the generated address to the latch. This address is further latched using ADSTB signal.

$\overline{\text{IOR}}$

It is an active-low bidirectional tri-state input line, which is used by the CPU to read internal registers of 8257 in the Slave mode. In the master mode, it is used to read data from the peripheral devices during a memory write cycle.

$\overline{\text{IOW}}$

It is an active low bi-direction tri-state line, which is used to load the contents of the data bus to the 8-bit mode register or upper/lower byte of a 16-bit DMA address register or terminal count register. In the master mode, it is used to load the data to the peripheral devices during DMA memory read cycle.

CLK

It is a clock frequency signal which is required for the internal operation of 8257.

RESET

This signal is used to RESET the DMA controller by disabling all the DMA channels.

A_o - A_3

These are the four least significant address lines. In the slave mode, they act as an input, which selects one of the registers to be read or written. In the master mode, they are the four least significant memory address output lines generated by 8257.

$\overline{\textbf{CS}}$

It is an active-low chip select line. In the Slave mode, it enables the read/write operations to/from 8257. In the master mode, it disables the read/write operations to/from 8257.

A_4 - A_7

These are the higher nibble of the lower byte address generated by DMA controller in the master mode.

READY

It is an active-high asynchronous input signal, which makes DMA controller ready by inserting wait states.

HRQ

This signal is used to receive the hold request signal from the output device. In the slave mode, it is connected with a DRQ input line 8257. In Master mode, it is connected with HOLD input of the CPU.

HLDA

It is the hold acknowledgement signal which indicates the DMA controller that the bus has been granted to the requesting peripheral by the CPU when its status is 1.

$\overline{\textbf{MEMR}}$

It is the low memory read signal, which is used to read the data from the addressed memory locations during DMA read cycles.

$\overline{\textbf{MEMW}}$

It is the active-low tristate signal which is used to write the data to the addressed memory location during DMA write operation.

ADST

This signal is used to convert the higher byte of the memory address generated by the DMA controller into the latches.

AEN

This signal is used to disable the address bus/data bus.

TC

It stands for byte count (‘Terminal Count’) of the data transfer. If TC=1, then the current DMA is complete.

MARK

The mark will be activated after each 128 cycles or integral multiples of it from the beginning. It indicates the current DMA cycle is the 128th cycle since the previous MARK output to the selected peripheral device.

V_{cc}

It is the input power required for the operation of the circuit.

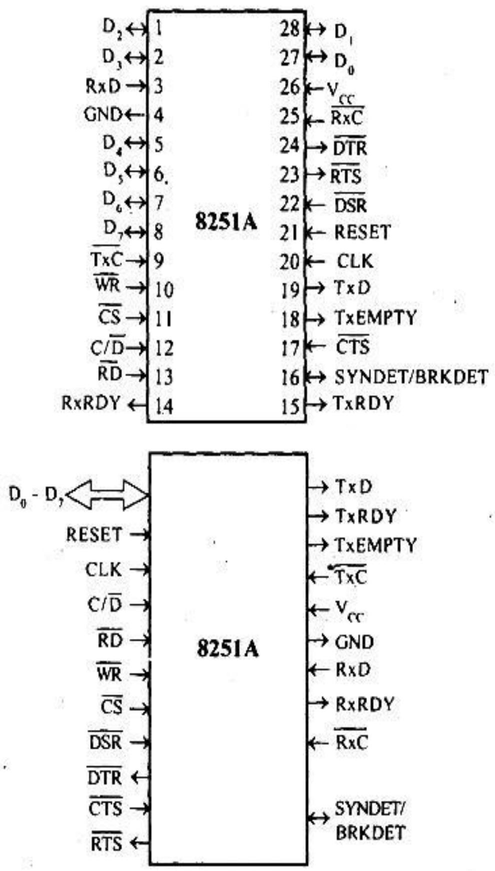
9.2 PROGRAMMABLE COMMUNICATION INTERFACE: INTEL 8251

8251 is a Universal Synchronous Asynchronous Receiver Transmitter (USART) and used as a mediator between microprocessor and peripheral to transmit serial data into parallel form and vice versa.

1. It takes data serially from peripheral devicesand converts into parallel data to the microprocessor for transmission.
2. It also receives parallel data from the microprocessor and converts it into serial form for making communication with the peripheral device.
3. It is packed in a 28 pin DIP.

9.2.1 PIN DESCRIPTION OF 8251 USART

The following picture represents the pin description of Intel 8251 USART



Pin	Description
D_0-D_7	Parallel data
$C/\overline{D}$	Control register or Data buffer select
$\overline{RD}$	Read control
$\overline{WR}$	Write control
$\overline{CS}$	Chip Select
CLK	Clock pulse (TTL)
RESET	Reset
$\overline{TxC}$	Transmitter Clock
TxD	Transmitter Data
$\overline{RxC}$	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready
TxRDY	Transmitter Ready
$\overline{DSR}$	Data Set Ready
$\overline{DTR}$	Data Terminal Ready
SYNDET/ BRKDET	Synchronous Detect / Break Detect
$\overline{RTS}$	Request To Send Data
$\overline{CTS}$	Clear To Send Data
TxEMPTY	Transmitter Empty
V_{cc}	Supply (+5V)
GND	Ground (0 V)

Figure 9.3 Pin description of Intel 8251 USART

9.2.2 ARCHITECTURE OF INTEL 8051A

The schematic diagram of the internal architecture of 8051A is given below. The function of the each block is described in the following section.

Read/Write control logic

- The Read/Write Control logic interfaces the 8251A with CPU and determines the functions of the 8251A according to the control word written into its control register.
- It monitors the data flow.This section has three registers and they are control register, status register and data buffer.
- The active low signals RD, WR, CS and C/D(Low) are used for read/write operations with these three registers.
- When C/D is high, the control register is selected for writing control word or reading status word.When C/D is low, the data buffer is selected for read/write operation.
- When the reset is high, it forces 8251A into the idle mode.The clock input is necessary for 8251A for communication with CPU and this clock does not control either the serial transmission or the reception rate.

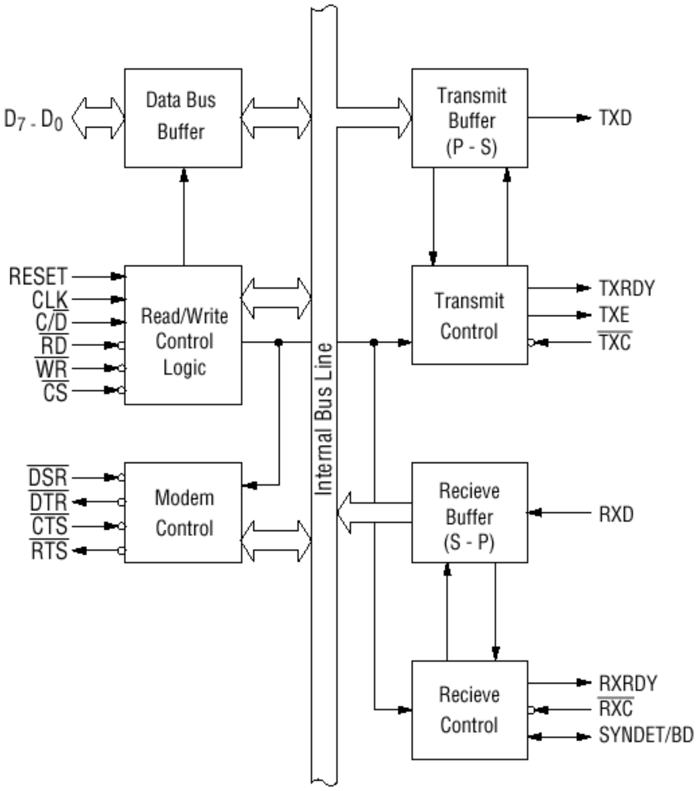


Figure 9.4 architecture of 8051A

Transmitter section

- The transmitter section accepts parallel data from CPU and converts them into serial data.

- The transmitter section is double buffered, i.e., it has a buffer register to hold an 8-bit parallel data and another register called output register to convert the parallel data into serial bits.

- When output register is empty, the data is transferred from buffer to output register. Now the processor can again load another data in buffer register. If buffer register is empty, then TxRDY goes to high.If output register is empty then TxEMPTY goes to high.

- The clock signal, TxC (low) controls the rate at which the bits are transmitted by the USART.The clock frequency can be 1,16 or 64 times the baud rate.

Receiver Section

- The receiver section accepts serial data and convert them into parallel data. The receiver section is double buffered, i.e., it has an input register to receive serial data and convert to parallel, and a buffer register to hold the parallel data.

- When the RxD line goes low, the control logic assumes it as a START bit, waits for half a bit time and samples the line again.If the line is still low, then the input register accepts the following bits, forms a character and loads it into the buffer register.

- The CPU reads the parallel data from the buffer register.When the input register loads a parallel data to buffer register, the RxRDY line goes high.

- The clock signal RxC (low) controls the rate at which bits are received by the USART.

- During asynchronous mode, the signal SYNDET/BRKDET will indicate the break in the data transmission.During synchronous mode, the signal SYNDET/BRKDET will indicate the reception of synchronous character.

MODEM Control

- The MODEM control unit allows to interface a MODEM to 8251A and to establish data communication through MODEM over telephone lines.This unit takes care of handshake signals for MODEM interface.

- The 8251A can be either memory mapped or I/O mapped in the system.Using a 3-to-8 decoder generates the chip select signals for I/O mapped devices.

- The address lines A4, A5 and A6 are decoded to generate eight chip select signals (IOCS-0 to IOCS-7) and in this, the chip select signal IOCS-2 is used to select 8251A.

- The address line A7 and the control signal IO / M(low) are used as enable for decoder.

- The address line A0 of 8085 is connected to C/D(low) of 8251A to provide the internal addresses.

- The data lines D0 - D7 are connected to D0 - D7 of the processor to achieve parallel data transfer.

- The RESET and clock signals are supplied by the processor. The processor clock is directly connected to 8251A. This clock controls the parallel data transfer between the processor and 8251A.

- The output clock signal of 8085 is divided by suitable clock dividers like programmable timer 8254 and then used as clock for serial transmission and reception.

- The TTL logic levels of the serial data lines and the control signals necessary for serial transmission and reception are converted to RS232 logic levels using MAX232 and then terminated on a standard 9-pin D-.type connector.

- In 8251A the transmission and reception baud rates can be different or same. The device which requires serial communication with processor can be connected to this 9-pin D-type connector using 9-core cable

- The signals TxEMPTY, TxRDY and RxRDY can be used as interrupt signals to initiate interrupt driven data transfer scheme between processor and 8251 A.

UNIT – X MICROPROCESSOR APPLICATIONS

Structure

10.1 ANALOG TO DIGITAL DATA CONVERTER

General Characteristics of Digital and Analog Data:

Digital Data:- Evenly spaced discontinuous values and Temporally and quantitatively discrete.

Analog Data (Natural Phenomena): Continuous range of values and Temporally and quantitatively continuous

For the measurement and control of physical quantity such as speed, displacement etc transducers are used which give electrical voltage proportional to the measuring physical quantity. This electrical voltage obtained as an output of a transducer is an analog quantity and it must be converted into digital quantity by the A/D converter (ADC) before it is applied to a microprocessor. In this aspect, an analog to digital converter is very much essential. There are many methods to convert analog signals to digital signals. These converters find more applications as an intermediate device to convert the signals from analog to digital form, displays output on LCD through a microcontroller. The objective of an A/D converter is to determine the output signal word corresponding to an analog input signal. The digital output varies from 0-2^n where n is the number of bits. ADC needs a clock to operate. The time taken to convert the analog to digital value depends on the clock frequency.

ADC 0808

ADC0808 is a A/D converter which has 8 analog inputs and 8 digital outputs. ADC0808 allows us to monitor up to 8 different transducers using only a single chip. This eliminates the need for external zero and full scale adjustments.

ADC0808 is a monolithic CMOS device, offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability and consumes minimal power. These features make this device ideally suited to applications. The pin diagram of ADC0808 is shown in figure below:

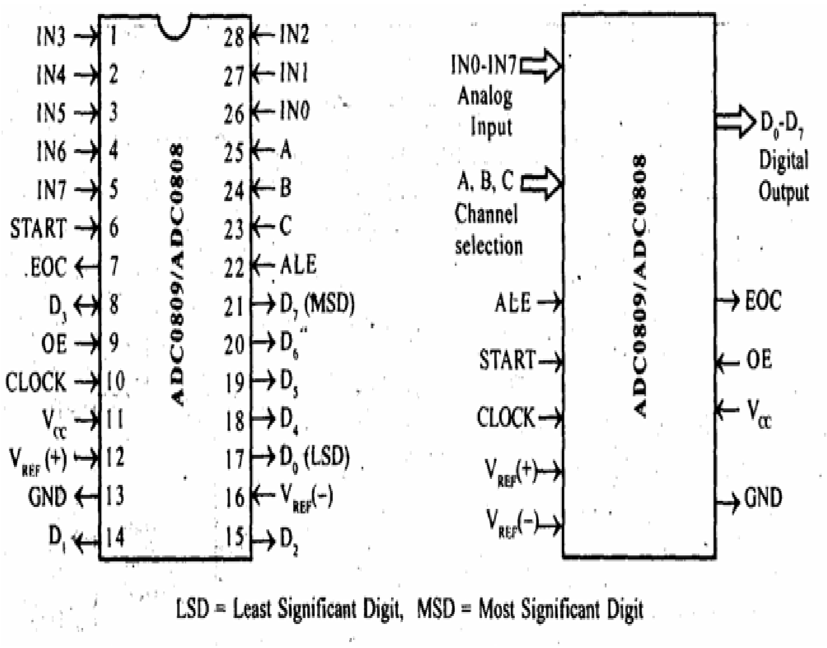
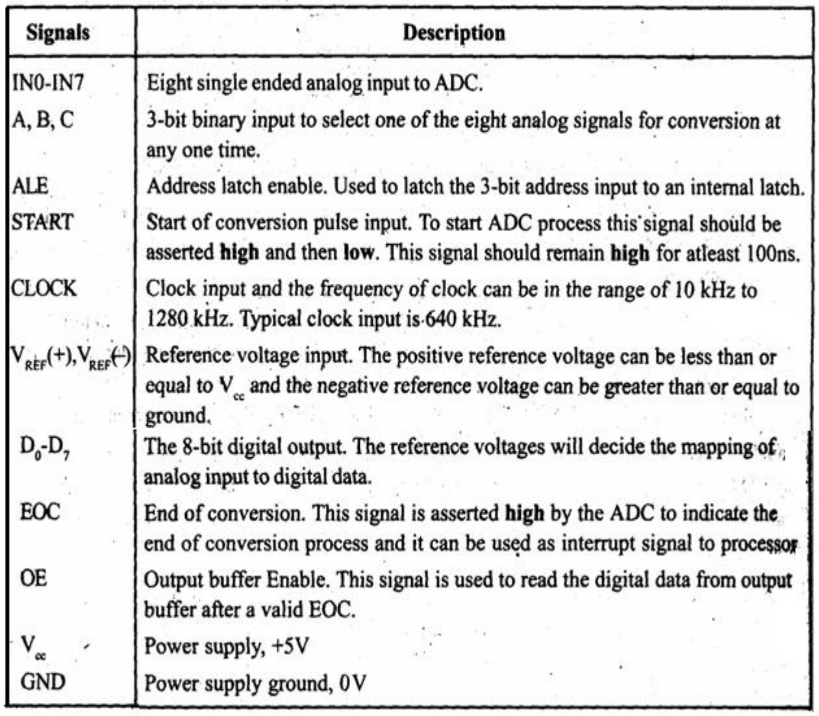


Figure 10.1 Pin diagram of ADC0808



Signals	Description
IN0-IN7	Eight single ended analog input to ADC.
A, B, C	3-bit binary input to select one of the eight analog signals for conversion at any one time.
ALE	Address latch enable. Used to latch the 3-bit address input to an internal latch.
START	Start of conversion pulse input. To start ADC process this signal should be asserted **high** and then **low**. This signal should remain **high** for atleast 100ns.
CLOCK	Clock input and the frequency of clock can be in the range of 10 kHz to 1280 kHz. Typical clock input is 640 kHz.
$V_{REF}(+)$,$V_{REF}(-)$	Reference voltage input. The positive reference voltage can be less than or equal to V_{cc} and the negative reference voltage can be greater than or equal to ground.
D_0-D_7	The 8-bit digital output. The reference voltages will decide the mapping of analog input to digital data.
EOC	End of conversion. This signal is asserted **high** by the ADC to indicate the end of conversion process and it can be used as interrupt signal to processor
OE	Output buffer Enable. This signal is used to read the digital data from output buffer after a valid EOC.
V_{cc}	Power supply, +5V
GND	Power supply ground, 0V

Main Features:

Easy interface to all microprocessors

No zero or full-scale adjust required

8-channel multiplexer with address logic

0V to 5V input range with single 5V power supply

Outputs meet TTL voltage level specifications

Carrier chip package with 28-pin

Working Principle of Successive Approximation type A/D Converter

The most popular method of analog to digital conversion is successive approximation method.Here an unknown voltage Vin, the voltage to be converted into digital form is compared with a fraction of the reference voltage Vref and correspondingly the value of the bit is set to either 1 or 0. The setting of bit starts from MSB to LSB and continues up to the ‘n’ bit.

In the first step the unknown voltage Vin is compared with ½Vr. If Vin>=1/2 Vr, the MSB of the digital output is set to 1. If Vin<1/2Vr, MSB is set to 0.

In the next step Vin is compared with (1/2b1+1/4)Vr, where b1 is the MSB value which was already determined. If Vin>=1/2(b1+1/4)Vr the 2nd bit is set to 1. If Vin<1/2(b1+1/4)Vr the 2nd bit is set to 0. To obtain the 3rd bit of the digital output again Vin is compared with (1/2b1+1/4b2+1/8)Vr. This process will extend up to the nth bit. 8 times for 8 bit converter.

Specifications:

Resolution: 8 Bits

Total Unadjusted Error: ±½ LSB and ±1 LSB

Single Supply: 5 VDC

Low Power: 15 mW

Conversion Time: 100 µs

Generally, the ADC0808 input which is to be changed over to digital form can be selected by using three address lines A, B, C which are pins 23, 24 and 25. The step size is chosen dependent upon set reference value. Step size is the change in analog input to cause a unit change in the output of ADC. ADC0808 needs an external clock to operate. The continuous 8-bit digital output corresponding to instantaneous value of analogue input. The most extreme level of input voltage must be reduced proportionally to +5V. The address lines A, B, C are connected to microcontroller for the commands. In this the interrupt follows the low to high operation. When the start pin is held high no conversion begins, but when the start pin is low the conversion will start within 8 clock periods. The point when the conversion is completed the EOC pin goes low to indicate the finish of conversion and data ready to be picked up. The output enable (OE) is then raised high. This enables the TRI-STATE outputs, allowing the data to be read.

The ADC 0808 IC requires clock signal of typically 550 kHz, ADC0808 is used to convert the data into digital from required for the microcontroller.

Application of ADC0808:

The ADC0808 has got many applications including in the fields of music, communication, and scientific instruments.

Interfacing of ADC with microprocessor

The analog to digital converter chips 0808 and 0809 are 8- bit CMOS, successive approximation converters. This technique is one of the fast techniques for analog to digital conversion. The conversion delay is 100µs at a clock frequency of 640 KHz, which is quite low as compared to other converters.

The process of analog to digital conversion is a slow process, and the microprocessor has to wait for the digital data till the conversion is over. After the conversion is over, the ADC sends end of conversionEOC signal to inform the microprocessor that the conversion is over and the result is ready at the output buffer of the ADC. These tasks of issuing an SOC pulse to ADC, reading EOC signal from the ADC and reading the digital output of the ADC are carried out by the CPU using 8255 I/O ports

General algorithm for ADC interfacing contains the following steps: 1. Ensure the stability of analog input, applied to the ADC. 2. To issue the start of conversion pulse to ADC 3. Read the end of conversion signal to mark the end of conversion process. 4. Read the digital data output of the ADC.

Analog input voltage must be constant at the input of the ADC right from the start of conversion till the end of the conversion to get correct results.This may be ensured by a sample and hold circuit which samples the analog signal and holds it constant for specific time duration. The microprocessor may issue a hold signal to the sample and hold circuit. If the applied input changes before the complete conversion process is over, the digital equivalent of the analog input calculated by the ADC may not be correct.

10.2 MICROPROCESSOR BASED TEMPERATURE MONITORING CONTROLLER

Microprocessor based temperature monitoring and controlling is one of the very important applications in industry. Temperature control is a process in which change of temperature of an object is measured or detected, and the supply of heat energy into or out of the object is adjusted in order to achieve a desired temperature. An Automatic Temperature Control Unit mainly divided into three parts-Temperature input unit, Processing unit and Control output unit.

The 8085 based scheme forms the basic processing unit, the Analog-to-Digital converter unit, and temperature input unit with temperature sensor. The switching ON/OFF of the heater controls the heat supplied to the object. Basic operation of the temperature control unit,requires two set points, Upper set point and Lower set point. Whenever the temperature of the object exceeds the upper limit the power is turned-off. If the temperature drops below the lower limit, then the power is turned on. In this way, the temperature is maintained around the set limit.

The main hardware of the system consists of 8085 Microprocessor unit, ADC interface board, Temperature sensor and Amplifier (741 op-Amp) is shown in the below figure.

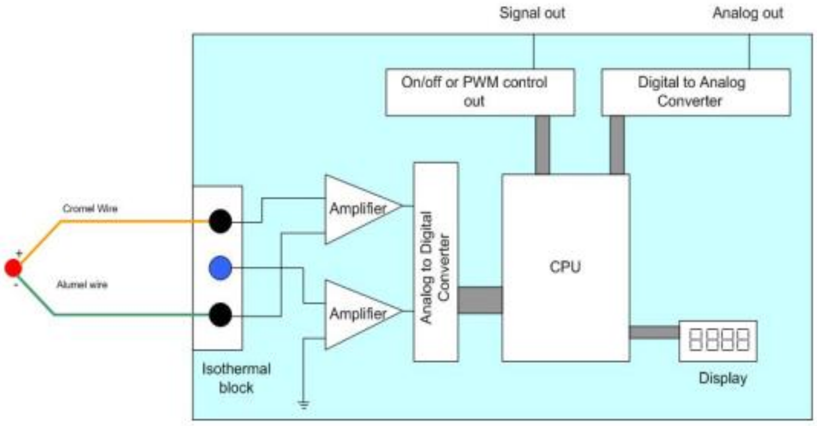


Figure 10.2Scheme for temperature monitoring and controlling

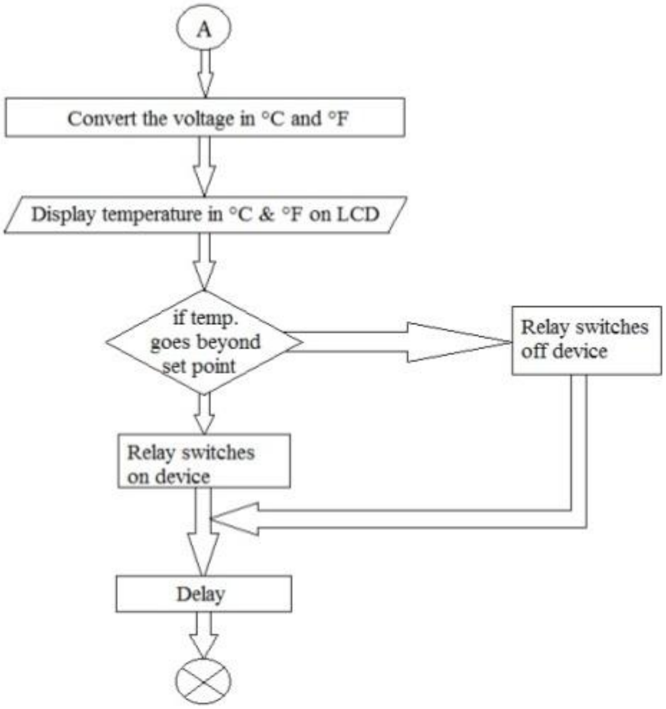

A
Convert the voltage in °C and °F
Display temperature in °C & °F on LCD
if temp. goes beyond set point
Relay switches off device
Relay switches on device
Delay

Figure 10.3Flow chart for the microprocessor based temperature monitoring and controlling

10.3 MICROPROCESSOR BASED STEPPER MOTOR CONTROLLER

A motor consists of stator and a rotor. A stepper motor is a motor controlled by a series of electromagnetic coils. The center shaft has a series of magnets mounted on it, and the coils surrounding the shaft are alternately given current creating magnetic fields which repulse or attract the magnets on the shaft, causing the motor to rotate in terms of steps, rather than continuous as in case of normal motors. A sequence of pulses is applied to the windings of the stepper motor, in a proper sequence in order to make the shaft to rotate in steps. The number of pulses required for one complete rotation of the shaft is equal to its number of internal teeth (angle of rotation) on its rotor. The stator and the rotor teethes lock with each other to fix a position of the shaft.

With a pulse applied to the winding input, the rotor rotates by one teeth position or an angle X. The angle X may be calculated as:

$$X=\frac{360^\circ}{\text{number of rotor teeths}}$$

After the rotation of the shaft through angle X, the rotor locks itself with the next tooth in the sequence on the internal surface of stator.This design allows for very precise control of the motor by proper pulsing.These are used in printers, disk drives, and other devices where precise positioning of the motor is necessary and be controlled with digital circuits.

The internal schematic of a typical stepper motor with four windings with an angle of rotation 90° and its rotor are shown in the following figures.

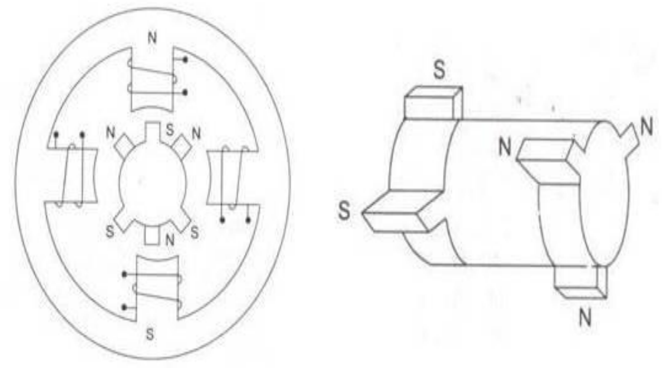


Figure 10.4

Schematic diagram of stepper motor with four windings

A typical pulse scheme to realize the rotation in both clockwise and anticlockwise is given in the following table.This scheme is called a wave scheme. The four windings, say Wa, Wb, Wc and Wd are applied with the required voltages pulses (as per the motor specification), in a cyclic fashion. By reversing the sequence of excitation, the direction of rotation of the stepper motor shaft may be reversed.

By selecting a proper input circuit along with the required output in digital form will help to interface the stepper motor with 8085 or 8086. A simple scheme represented in the below figure will explain a method of interfacing stepper motor with 8085.

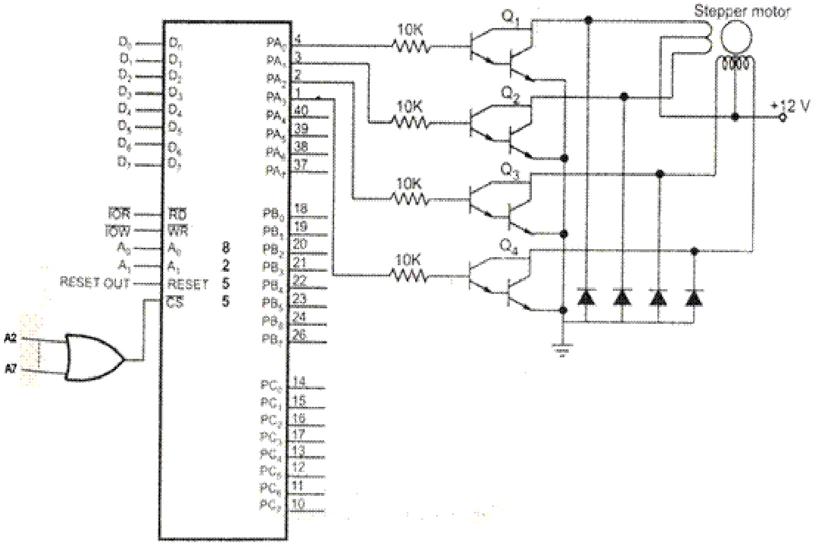


Figure 10.4Interfacing stepper motor with 8085

Motion	step	A	B	C	D
Clock Wise Direction	1	1	0	0	0
	2	0	1	0	0
	3	0	0	1	0
	4	0	0	0	1
	1(2nd round)	1	0	0	0
Anti-clock wise Direction	1	1	0	0	0
	2	0	0	0	1
	3	0	0	1	0
	4	0	1	0	0
	1(2nd round)	1	0	0	0

UNIT – XI AMPLIFIERS AND COMPARATORS

Structure

11.1 INTRODUCTION TO INSTRUMENTATION AMPLIFIER

Instrumentation amplifiers (in-amps) are circuit elements designed to allow users to extract and amplify the difference between two signals or sourceswhile rejecting any signals that are common to both inputs.The in-amp, therefore, provides the very important function of extracting small signals from transducers and other signal sources.

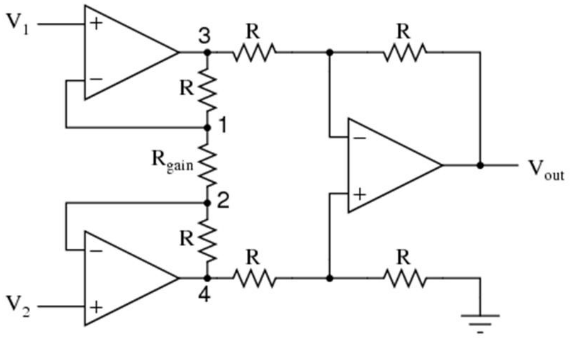


Figure 11.1Block diagram ofInstrumentation amplifiers

An instrumentation amplifier is a closed-loop gain block that has a differential input and an output that is single-ended with respect to a reference terminal. Most commonly, the impedances of the two input terminals are balanced and have high values, typically 10^9or greater. The input bias currents should also be low, typically 1 nA to 50 nA. As with operational amplifiers, output impedance is very low, nominally only a few milliohms, at low frequencies.

A major issue that arises from using a single op-amp as a differential amplifier has a lower common-mode rejection (CMR), usually arises from a negative feedback loop to increase stability. The feedback loop allows for common-mode voltages to appear on the output of the amplifier, reducing the usable range of the amplifier. Common -mode rejection (CMR), the property of canceling out any signals that are common (the same potential on both inputs), while amplifying any signals that are differential (a potential difference between the inputs), is the most important function of an instrumentation amplifier. Both dc and ac common-mode rejection are important. Any errors due to dc common-mode voltage (i.e., dc voltage present at both inputs) will be reduced up to 120 dB by instrumentation amplifiers available in the market.

Instrumentation amplifiers consist of several operational amplifiers (op-amps) and are primarily designed to have many advantages over op-amps such as high CMR, Low Offset Voltage and Offset Voltage Drift,High Input Impedance, Low Input Bias and Offset Current Errors, Low Noise, Low Nonlinearity and high gain. Many in-amps consist of three op-amps, with two serving as a buffer for the two input circuits and the third as a differential op-amp.

11.2 OPERATION OF INSTRUMENTATION AMPLIFIER

This section will cover the usual operation of instrumentation operators. This in-amp is in two stages, namely a buffering stage and a differential stage. The in-amp starts with two signals that the user would like to get the differential output. The signals are applied one each to a non-inverting terminal of the op-amps in the first stage. This allows for the input device to see a large resistance and reduce current draw. In an ideal op-amp, the voltage at the non-inverting terminal is equal to that at the inverting terminal. Therefore, at points 1 and 2, the voltage is V_1 and V_2, respectively.

The second stage of the example in-amp is a differential op-amp, with the inputs being the voltages at points 3 and 4, namely V_3 and V_4. The output of this stage is the final output, V_{out}. The output of this amplifier is

$$V_{out} = (V_4 - V_3)$$

Having the feedback resistors on both pins being equal makes this a unity-gain differential op-amp. In order to get the output in terms of the inputs, the relationship between $V_4 - V_3$ and $V_1 - V_2$ must be found.

Between point 3 and 4, there are three resistors with no connection to ground. This causes the current through all three resistors, I, to be equal. Therefore

$$I = \frac{V_3 - V_4}{2R + R_{gain}}$$

Also, the voltages at point 1 and 2 are known, yielding the equation

$$I = \frac{V_1 - V_2}{R_{gain}}$$

When the two equations are combined and rearranged, the equations yield

$$V_3 - V_4 = (V_1 - V_2)\,\frac{2R+1}{R_{gain}}$$

This relationship can be plugged into the equation for V_{out} to yield a final equation of

$$V_3 - V_4 = (V_2 - V_1)\,\frac{2R+1}{R_{gain}}$$

11.3 SAMPLE AND HOLD CIRCUIT

The sample-and-hold circuit is a critical component of most data acquisition systems. It captures an analog signal and holds it in its output during some operation (most commonly analog-digital conversion).Typically, the samples are taken at uniform time intervals thus, the sampling rate (or clock rate) of the circuit can be determined.

When the sample-and-hold is in the sample (or track) mode, the output follows the input with only a small voltage offset.

The operation of an S/H circuit can be divided into sample mode (or as acquisition mode) and hold mode, whose durations need not be equal. In hold mode, the output of the circuit is equal to the previously sampled input value. In sample mode, the output can either track the input, in which case the circuit is often called a track-and-hold (T/H) circuit, or it can be reset to some fixed value. In some circuits the output is held over the whole period of the sampling clock. This is achieved by having separate circuitry to perform the sampling and the holding operations. The basic circuit of sample and hold circuit is given below;

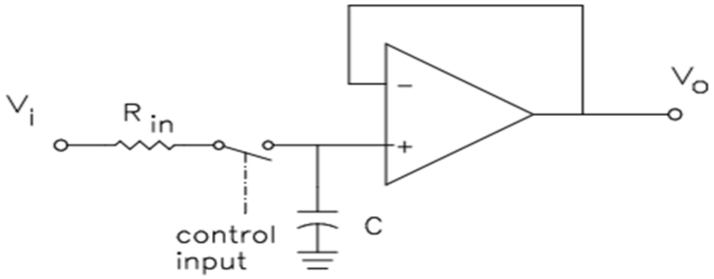


Figure 11.2Basic circuit of sample and hold circuit

Control input open and closes (solid-state switch) at sampling rate fs. The modes of operation of the control switch - tracking (switchclosed) and hold (switch open). The sample and hold parameters are defined as follows;

acquisition time - time for instant switch closes until Vi within defined % (~ ≥99.3%) of input.
aperture time - time taken by the switch to open.
decay rate - rate of discharge of C when circuit is in hold mode.

For a given analog signal, if the highest analog frequency is fmax. According to the Nyquist Theorem, the sampling rate must be at least 2fmax, or twice the highest analog freqsuency component. The sampling in an analog-to-digital converter is actuated by a pulse generator (clock). If the sampling rate is less than 2fmax, some of the highest frequency components in the analog input signal will not be correctly represented in the digitized output.

11.4 COMPARATORS

Analog-to-Digital (A/D or ADC) and Digital-to-Analog (D/A or DAC) circuits are in wideuse today because of the need to translate analog signal to digital format for processing by microprocessors and translate the output of microprocessors (digital signal) to analog signal to drive the loads. The heart of most ADC circuit is the “comparator” whose schematic diagram is given in the below figure. A comparator compares the value of input signal to a reference voltage. If the input signal voltage is larger than the reference voltage, comparator outputwill be in a set voltage (for example, V +, for a “HIGH” or “ON” state). If the input signal voltage is smaller than the reference voltage, comparator output will be in a different, yet another set voltage (for example, V −, for a “LOW” or “OFF” state).

That means it takes two input voltages, then compares them and gives a differential output voltage either high or low-level signal. The comparator is used to sense when an arbitrary varying input signal reaches reference level or a defined threshold level. The comparator can be designed by using various components like diodes, transistors, op-amps. The comparators find in many electronic applications that may be used to drive logic circuit.

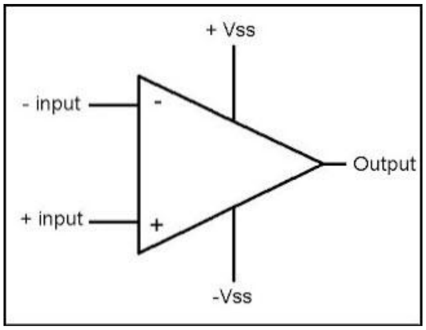


Figure 11.3 Block diagram of Comparator

11.5 DIGITAL TO ANALOG CONVERTER

Digital to Analog Converter (DAC) is a device that transforms digital data into an analog signal. A DAC can reconstruct sampled data into an analog signal with precision. The digital data may be produced from a microprocessor, Application Specific Integrated Circuit (ASIC), or Field Programmable Gate Array (FPGA), but ultimately the data requires the conversion to an analog signal in order to interact with the real world.

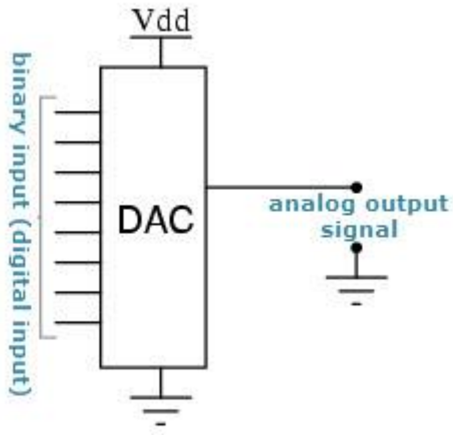


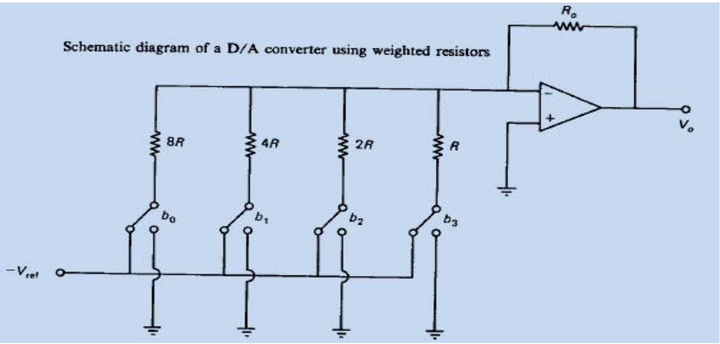
Figure 11.4 Block diagram of Digital to Analog Converter

11.6 D/A CONVERTER ARCHITECTURES

There are two methods commonly used for digital to analog conversion: Weighted Resistors method and the other one is using the R-2R ladder network method.

11.6.1 WEIGHTED RESISTORS METHOD

The below shown schematic diagram is DAC using weighted resistors. The basic operation of DAC is the ability to add inputs that will ultimately correspond to the contributions of the various bits of the digital input. In the voltage domain, that is if the input signals are voltages, the addition of the binary bits can be achieved using the inverting summing amplifier shown in the below figure.


Schematic diagram of a D/A converter using weighted resistors

In the voltage domain, that is if the input signals are voltages, the addition of the binary bits can be achieved using the inverting summing amplifier shown in the above figure.

The input resistors of the op-amp have their resistance values weighted in a binary format. When the receiving binary 1 the switch connects the resistor to the reference voltage. When the logic circuit receives binary 0, the switch connects the resistor to ground. All the digital input bits are simultaneously applied to the DAC.

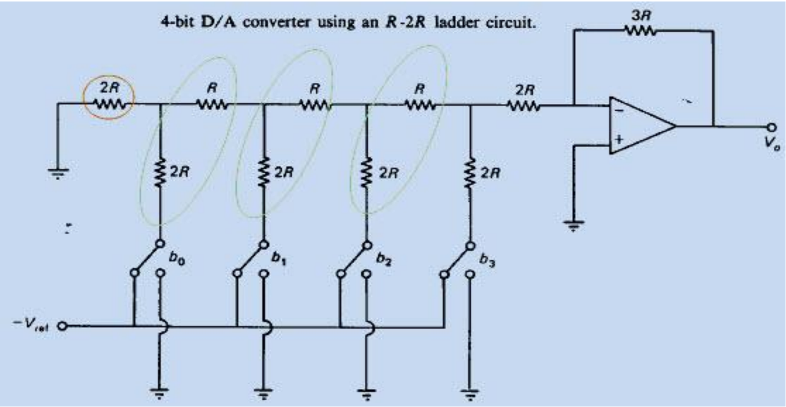
The DAC generates analog output voltage corresponding to the given digital data signal. For the DAC the given digital voltage is b3 b2 b1 b0 where each bit is a binary value (0 or 1). The output voltage produced at output side is

$$V0=R0/R\ (b3+b2/2+b1/4+b0/8)\ Vref$$

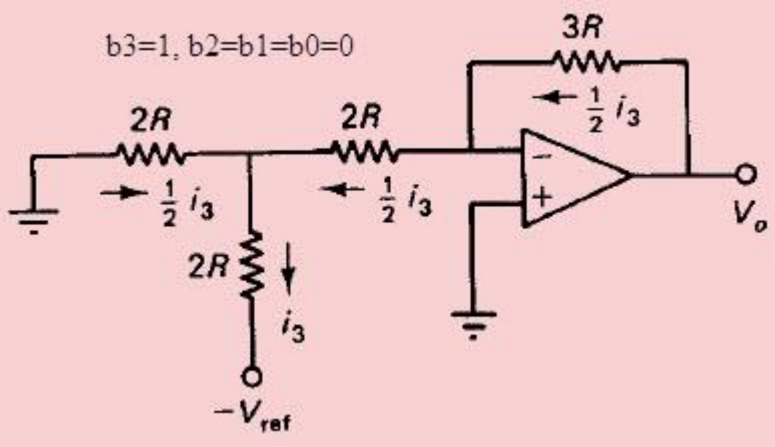
As the number of bits is increasing in the digital input voltage, the range of the resistor values becomes large and accordingly, the accuracy becomes poor.

11.6.2 RESISTOR LADDER NETWORK METHOD

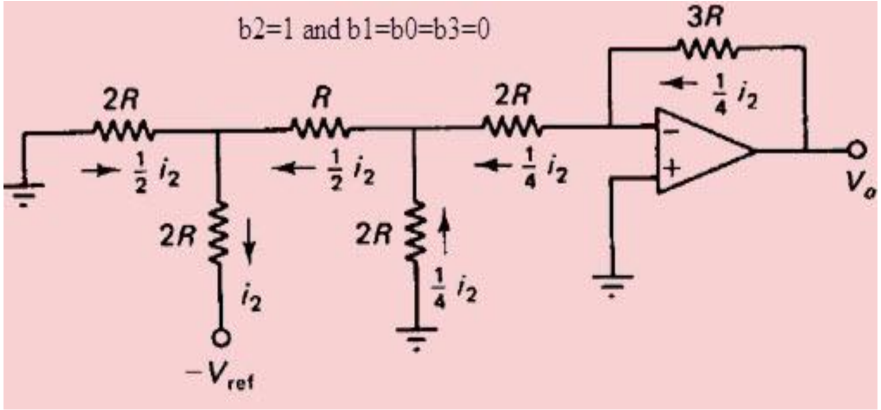
The R-2R ladder DAC constructed as a binary-weighted DAC that uses a repeating cascaded structure of resistor values R and 2R. This improves the precision due to the relative ease of producing equal valued-matched resistors (or current sources).


4-bit D/A converter using an R-2R ladder circuit.

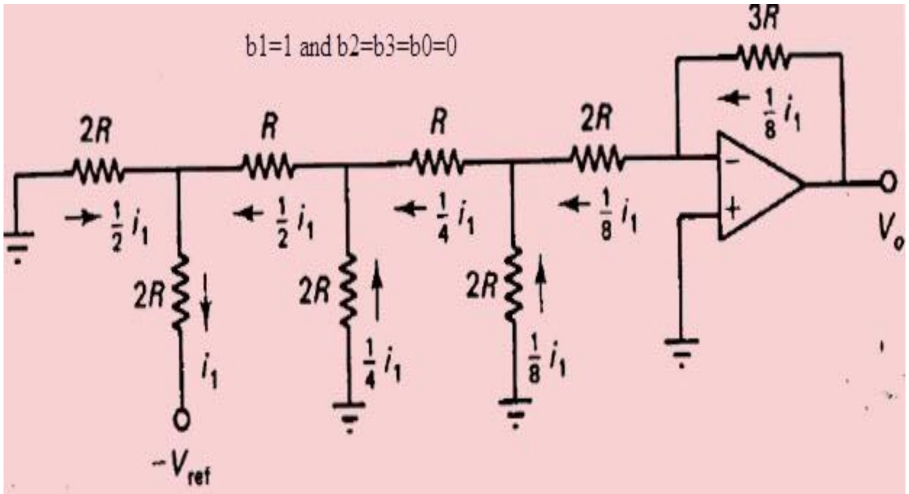
The above figure shows the 4-bit R-2R ladder DAC. In order to achieve high-level accuracy, we have chosen the resistor values as R and 2R. Let the binary value B3 B2 B1 B0, if b3=1, b2=b1=b0=0, then the circuit is shown in the figure below it is a simplified form of the above DAC circuit. The output voltage is V0=3R(i3/2)= Vref/2



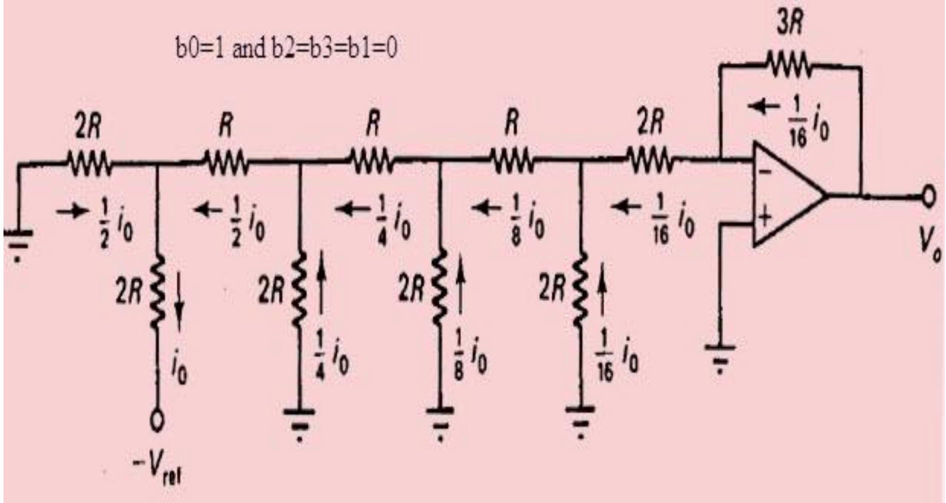
Similarly, If b2=1, and b3=b1=b0=0, then the output voltage is V0=3R(i2/4)=Vref/4 and the circuit is simplified as below



If b1=1 and b2=b3=b0=0, then the circuit shown in the figure below it is a simplified form of the above DAC circuit. The output voltage is V0=3R(i1/8)= Vref/8



Finally, the circuit is shown in below corresponding to the case where b0=1 and b2=b3=b1=0. The output voltage is V0=3R(i0/16) = Vref/16



In this way, we can find that when the input data is b3b2b1b0 (where individual bits are either 0 or 1), then the output voltage is

$$V_o = (\tfrac{1}{2}b_3 + \tfrac{1}{4}b_2 + \tfrac{1}{8}b_1 + \tfrac{1}{16}b_0)V_{ref}$$
$$= \tfrac{1}{2}(b_3 + \tfrac{1}{2}b_2 + \tfrac{1}{4}b_1 + \tfrac{1}{8}b_0)V_{ref}$$

UNIT – XII TEMPERATURE TRANSDUCERS

Structure

12.1 INTRODUCTION

A device which converts a physical quantity into the proportional electrical signal is called a transducer. The electrical signal produced may be a voltage, current or frequency. A transducer uses many effects to produce such conversion. The process of transforming signal from one form to other is called transduction. A transducer is also called pick up. The transduction element transforms the output of the sensor to an electrical output, as shown in theFig.

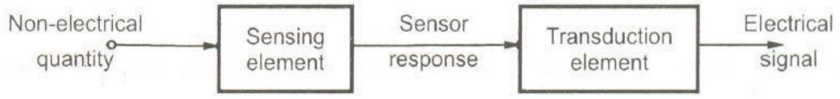


Figure 12.1 Basics of Transducers

A transducer will have basically two main components. They are

1. SensingElement

The physical quantity or its rate of change is sensed and responded to by this part of the transistor.

2. TransductionElement

The output of the sensing element is passed on to the transduction element. This element is responsible for converting the non-electrical signal into its proportional electricalsignal.

There may be cases when the transduction element performs the action of both transduction and sensing. The best example of such a transducer is a thermocouple. A thermocouple is used to generate a voltage corresponding to the heat that is generated at the junction of two dissimilar metals.

12.2 CLASSIFICATION OF TRANSDUCERS

The Classification of Transducers is done in many ways. Some of the criteria for the classification are based on their area of application, Method of energy conversion, Nature of output signal, According to Electrical principles involved, Electrical parameter used, principle of operation, &typical applications.The transducers can be classified broadly

i. On the basis of transduction formused
ii. As primary and secondarytransducers
iii. As active and passivetransducers
iv. As transducers and inversetransducers.

Broadly one such generalization is concerned with energy considerations wherein they are classified as active &passive transducers. A component whose output energy is supplied entirely by its input signal (physical quantity under measurement) is commonly called a “passive transducer”. In other words the passive transducers derive the power required for transduction from an auxiliary source. Active transducers are those which do not require an auxiliary power source to produce their output. They arealso known as self-generating type since they produce their own voltage or current output.

There are four commonly used temperature sensor types:

1. Negative Temperature Coefficient (NTC) thermistor

A thermistor is a thermally sensitive resistor that exhibits a large, predictable, and precise change in resistance correlated to variations in temperature. An NTC thermistor provides a very high resistance at low temperatures. As temperature increases, the resistance drops quickly. A large change in resistance per °C or small changes in temperature are reflected very fast and with high accuracy (0.05 to 1.5 °C). The effective operating range is -50 to 250 °C for glass encapsulated thermistors.

2. Resistance Temperature Detector (RTD)

An RTD, also known as a resistance thermometer, measures temperature by correlating the resistance of the RTD element with temperature. An RTD consists of a film or, for greater accuracy, a wire wrapped around a ceramic or glass core. The most accurate RTDs are made using platinum but lower-cost RTDs can be made from nickel or copper. However, nickle and copper are not as stable or repeatable. Platinum RTDs offer a fairly linear output that is highly accurate (0.1 to 1 °C) across -200 to 600 °C. While providing the greatest accuracy, RTDs also tend to be the most expensive of temperature sensors.

3. Thermocouple

This temperature sensor type consists of two wires of different metals connected at two points. The varying voltage between these two points reflects proportional changes in temperature. Thermocouples are nonlinear, requiring conversion when used for temperature control and compensation, typically accomplished using a lookup table. Accuracy is low, from 0.5 °C to 5 °C.

However, they operate across the widest temperature range, from -200 °C to 1750 °C.

4. Semiconductor-based sensors

A semiconductor-based temperature sensor is placed on integrated circuits (ICs). These sensors are effectively two identical diodes with temperature-sensitive voltage vs current characteristics that can be used to monitor changes in temperature. They offer a linear response but have the lowest accuracy of the basic sensor types at 1 °C to 5 °C. They also have the slowest responsiveness (5s to 60s) across the narrowest temperature range (-70 °C to 150 °C).

UNIT – XIII DISPLACEMENT TRANSDUCERS

Structure

13.1 INTRODUCTION-POTENTIOMETER

A potentiometer is a resistive sensor used to measure linear displacements as well as rotary motion. In a potentiometer an electrically conductive wiper slides across a fixed resistive element. A voltage is applied across the resistive element. Thus a voltage divider circuit is formed. The output voltage(Vout) is measured as shown in the figure below. The output voltage is proportional to the distance travelled.

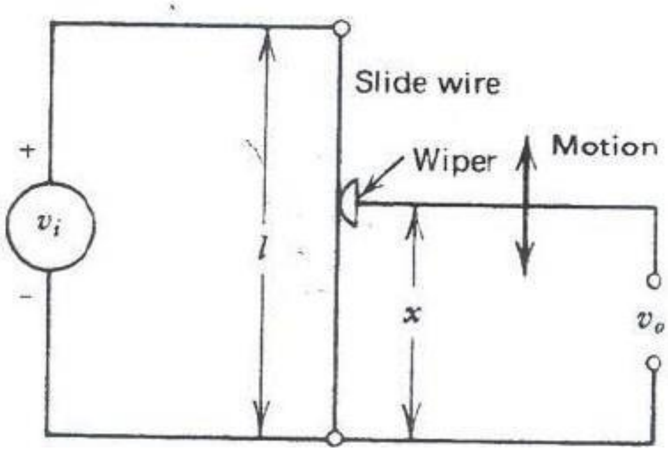


Figure 13.1 Block diagram of a potentiometer

There are two types of potentiometer, linear and rotary potentiometer. The linear potentiometer has a slide or wiper. The rotary potentiometer can be a single turn or multi turn.

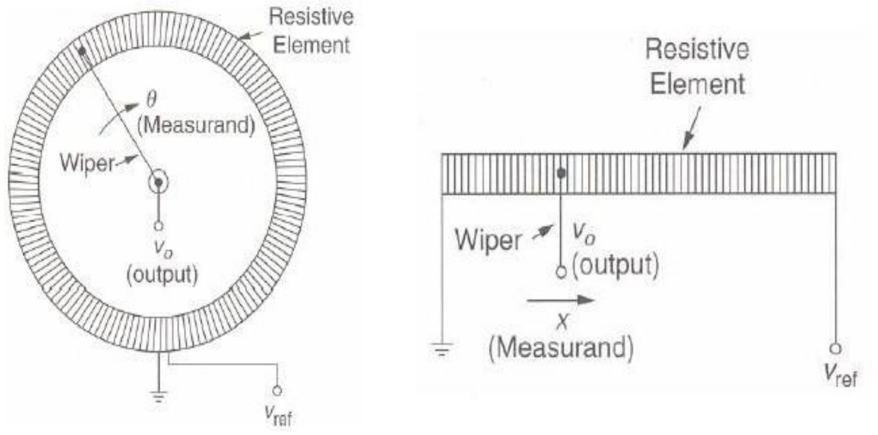


Figure 13.2 Types of potentiometer

The important parameters while selecting a potentiometer are

- Operating temperature
- Shock and vibration
- Humidity
- Contamination and seals
- Life cycle

13.2 TYPES OF POTENTIOMETER

13.2.1 WIRE-WOUND TYPE POTENTIOMETER

- The resistance range between 10Ω and 10MΩ
- The resistance increase in a stepwisemanner.
- It is possible to construct potentiometers with 100 –200 turns per cm length (The resolution range between 0.1 to 0.05mm).
- Linear potentiometers are available in many lengths up to1m.
- Helical potentiometers are commercially available with 50 to 60 turns (The angular displacement is between 18000 – 21600degree)
- Potentiometer life exceeds 1 millioncycles.

13.2.2 THIN FILM TYPE POTENTIOMETER

- Higher resolution.
- Lower noise.
- Longer life (exceed 10 millioncycles)
- Resistance of 50 to 100 Ω/mm can be obtained with conductive plasticfilm.

- Commercially available resolution is 0.001mm.

13.2.3 SOME OF THE ADVANTAGES OF THE POTENTIOMETER

- Easy to use
- low cost
- High amplitude output
- Proven technology
- Easily available

13.2.4 SOME OF THE DISADVANTAGES OF THE POTENTIOMETER

- Since the wiper is sliding across the resistive element there is a possibility of friction andwear. Hence the number of operating cycles is limited.
- Limited bandwidth
- Inertialloading

13.2.5 SOME OF THE APPLICATIONS OF THE POTENTIOMETER

- Linear displacementmeasurement
- Rotary displacementmeasurement
- Volume control
- Brightness control
- Liquid level measurements using float

13.3 STRAIN GAUGE

Strain gage is one of the most popular types of transducer. It has got a wide range of applications. It can be used for measurement of force, torque, pressure, acceleration and many other parameters. The basic principle of operation of a strain gage is simple: when strain is applied to a thin metallic wire, its dimension changes, thus changing the resistance of the wire.

13.3.1 RESISTIVE STRAIN GAUGE

Resistance gauges can be classified as transducers, i.e., devices for converting a mechanical displacement into an electrical signal.The resistance strain gauge is a valuable tool in the field of experimental stress analysis. It operates on the principle, discovered by the British Physicist William Thompson (later Lord Kelvin) in 1856,that the electrical resistance of a copper or iron wire changes when the wire is either stretched or compressed.

Resistance gauges are made in a variety of shapes, sizes, and types, mostly about the size of a postage stamp and gauge lengths as short as 0.038

cm are available to measure strains as small as 0.000001 inch per inch. These gauges can be used on the surface of almost any solid material or imbedded in the interior of concrete; being light, they are particularly suitable for measuring rapidly varying strains and the strains in rotating shafts.

13.4 CAPACITIVE TRANSDUCERS

A capacitor consists of two conductors (plates) that are electrically isolated from one another by a nonconductor (dielectric). When the two conductors are at different potentials (voltages), the system is capable of storing an electric charge. The storage capability of a capacitor is measured in farads.The principle of operation of capacitive transducers is based upon the equation for capacitance of a parallel plate capacitor as shown inFig.

$$Capacitance \qquad C = \frac{\varepsilon A}{D}$$

Where,

A = Overlapping area of plates; m^2,

d = Distance between two plates;m,

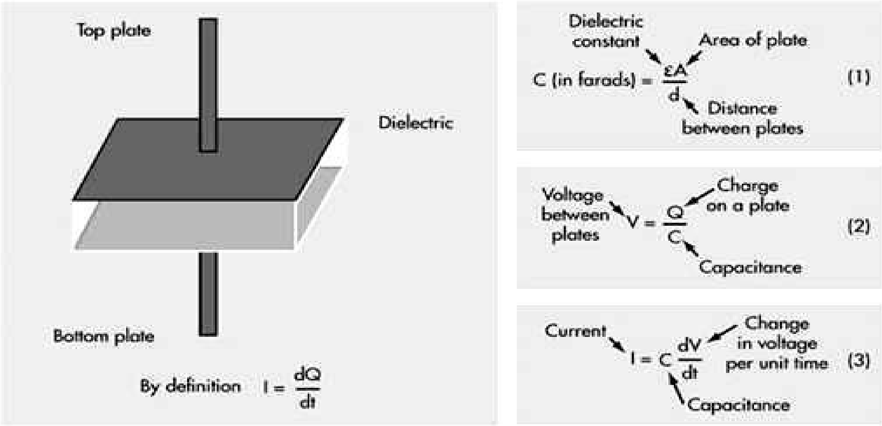


Figure 13.3 Parallel plate capacitor

The capacitance is measured with a bridge circuits. The output impedance Z of a capacitive transducer is:

Z = 1/2πfC

Where: Z =Impedance

f = frequency, 50 Hz.

C =capacitance

The capacitive transducers work on the principle of change in capacitance of the capacitor. This change in capacitance could be caused by change in overlapping area A of the plates, change in the distance d between the plates and change in dielectric constant of the medium.

In most of the cases the above changes are caused by the physical variables, such as, displacement, force or pressure. The capacitive transducers are commonly used for measurement of linear displacement.

The major advantages of capacitive transducers are that they require extremely small forces to operate them and hence are very useful for use in small systems. They are extremely sensitive and require small power to operate them. Owing to their good frequency response they are very useful for dynamicstudies.

The disadvantages of capacitive transducers include their non-linear behavior on account of edge effects and the effects of stray capacitances especially when the transducers have a low value of capacitance.Capacitive transducers can be used for measurement of both linear and angular displacements.

13.5LINEAR VARIABLE DIFFERENTIAL TRANSFORMER (LVDT)

The main function of LVDT is to convert the rectangular movement of an object to the equivalent electrical signal. LVDT is used to calculate displacement and works on the transformer principle.

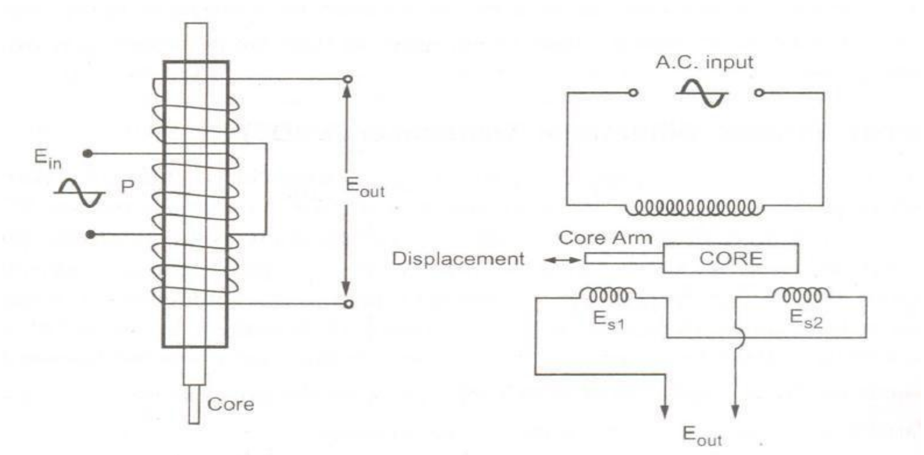


Figure 13.4 Skeletal picture of the LVDT

In the above picture, when an externally applied force moves the core to the left-hand position, more magnetic flux links the left-hand coil than the righthand coil. The emf induced in the left-hand is therefore larger than the induced emf of the right-hand. The magnitude of the output voltage is then equal to the difference between the two secondary voltages and it is in phase with the voltage of the left-handcoil. The output voltage of the LVDT against the position of the core is explained with the help of the graph below.

The main benefit of this transducer when compared with other LVDT types is its toughness as there is no material contact across the sensing components.

Because of dependency of the output voltage on the combination of magnetic flux, this transducer can have an unlimited resolution. Hence a small displacement can be noticed by an appropriate signal conditioning circuit, and the transducer’s resolution is exclusively determined by the merit of the DAS (data acquisition system).

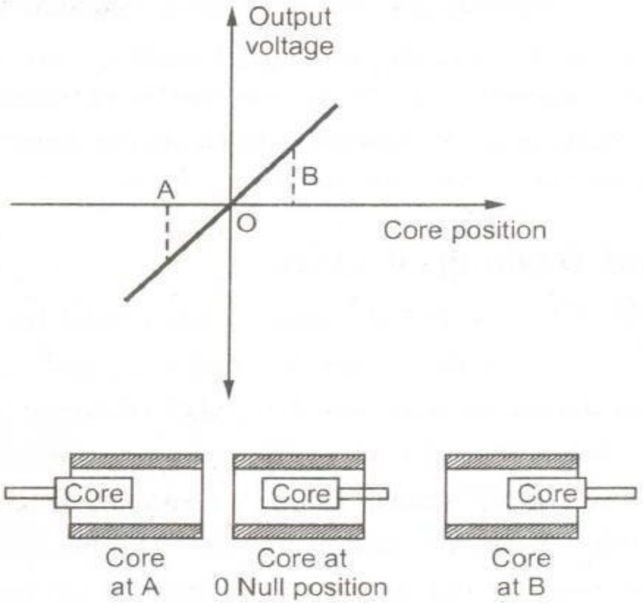


Figure 13.5Output voltage of the LVDT

13.5.1 CONSTRUCTION OF LVDT

The main features of the LVDT construction are as follows,

- The transformerconsists of a primary winding P and two secondary winding ES1 and ES2 wound on a cylindrical former(which is hollow in nature and will containcore).
- Both the secondary windings have equal number of turns and are identically placed on the eitherside of primarywinding
- The primary winding is connected to an AC source which produces a flux in the air gap and voltages are induced in secondarywindings.
- A movable soft iron core is placed inside the former and displacement to be measured isconnected to the ironcore.
- The iron core is generally of high permeability which helps in reducing harmonics and high sensitivity ofLVDT.
- The LVDT is placed inside a stainless steel housing because it will provide electrostaticand electromagneticshielding.

- The both the secondary windings are connected in such a way that resulted output is the difference of the voltages of twowindings.

13.5.2 ADVANTAGES OF LVDT

- High Range - The LVDTs have a very high range for measurement of displacement.they can used for measurement of displacements ranging from 1.25mm to250mm
- No Frictional Losses - As the core moves inside a hollow former so there is no loss ofdisplacement input as frictional loss so it makes LVDT as very accuratedevice.
- HighInputandHighSensitivity-TheoutputofLVDTissohighthatitdoes notneedany amplification.the transducer possesses a high sensitivity which is typically about40V/mm.
- Low Hysteresis - LVDTs show a low hysteresis and hence repeatability is excellent underall conditions
- Low Power Consumption - The power is about 1W which is very as compared to othertransducers.
- Direct Conversion to Electrical Signals - They convert the linear displacement to electricalvoltage which is easy toprocess.

13.5.3 DISADVANTAGES OF LVDT

- LVDT is sensitive to stray magnetic fields so they always require a setup to protect them fromstray magneticfields.
- They are affected by vibrations andtemperature.It is concluded that they are advantageous as compared than any other inductive transducers.

13.5.4 APPLICATIONS OF LVDT

They are used in applications where displacements ranging from fraction of mm to few cm are to be measured. The LVDT acting as a primary Transducer converts the displacement to electrical signal directly.

They can also acts as the secondary transducers. E.g. the Bourbon tube which acts as a primary transducer and covert pressure into linear displacement. Then LVDT coverts this displacementinto electrical signal which after calibration gives the ideas of the pressure offluid

UNIT – XIV PHOTOELECTRIC TRANSDUCERS

Structure

14.1 PHOTOELECTRIC TRANSDUCER

The photoelectric transducer converts the light energy into electrical energy. It is made up of semiconductor material. The photoelectric transducer uses a photosensitive element, which ejects the electrons when the beam of light absorbs by it.The discharges of electrons vary the property of the photosensitive element. The mobility of electrons produces one of the three effects.

- The resistance of the material changes.
- The output current of the semiconductor changes.
- The output voltage of the semiconductor changes.

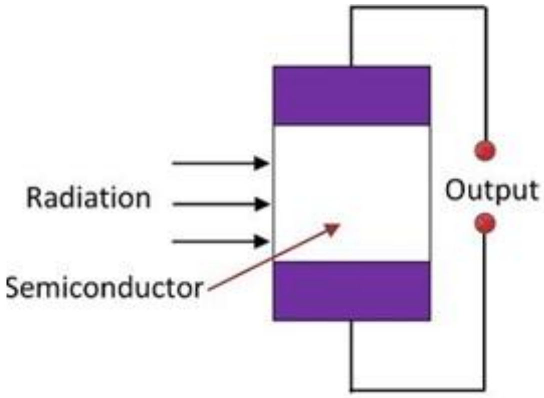


Figure 14.1 Block diagram of photoelectric transducer

The magnitude of the change in either resistance or current or voltage is proportional to the total light absorbed by the photosensitive element.The schematic figure shown below represents the photoelectric transducer.

14.2 PHOTO-VOLTAIC CELL

A p-type silicon and p-type silicon are diffused to form a photovoltaic cell (p-n junction). When this cell is exposed to light, photons are absorbed by the semiconductor crystal which causes significance number of free electrons in the crystal, this phenomenon is called photovoltaic effect.

The photovoltaic effect is the basis of generation of current across the semiconductor junction.Large sets of PV cells can be connected together to form solar modules, arrays, or panels. The use of PV cells and batteries for the generation of usable electrical energy from solar light is known as photovoltaic. One of the major advantages of photovoltaic is non-polluting electrical energy generation from unlimited solar energy. It can provide energy at cost effective over years and with minimal maintenance.

The current or voltage across the junction is proportional to the amount of light incident on that cell. Hence, it can be used as a transducer.

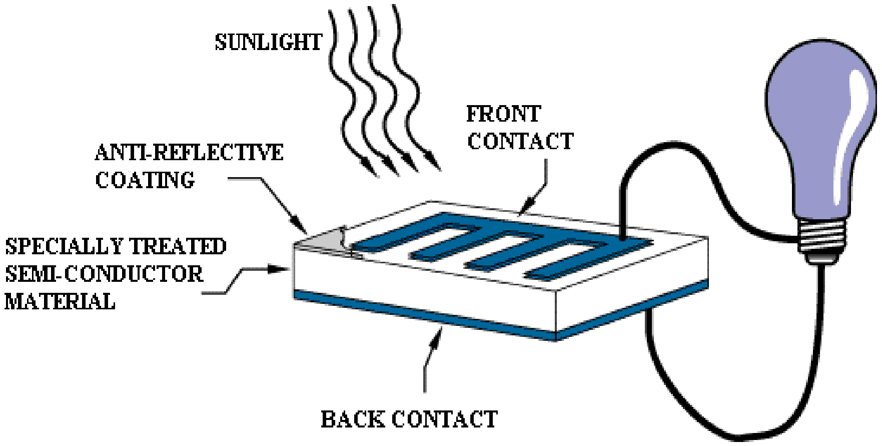


Figure 14.2 Block diagram of photo-voltaic cell

14.3 TYPES OF PHOTOVOLTAIC CELL

Photovoltaic cell or Solar Cell can be manufactured in a variety of ways and from many different semiconducting materials. Solar cells can be constructed from brittle crystalline structures or thin-films or combination of both. Crystalline solar cells can be further classified into two categories—monocrystallineand polycrystalline based solar cell. Solar cells can also be classified by their number of layers or "p-n junctions". Most commercial PV cells are only single-junction, but multi-junction photovoltaic cells have also been developed.The photo conversion efficiencyof the cell varies according to the material used or design and structure of the cell.

14.4 PHOTOCONDUCTIVE CELL

The photoconductive cell converts the light energy into an electric current. It uses the semiconductor material like cadmium selenide, Ge, Si, as a photo sensing element.When the beam of light falls on the semiconductor material, their conductivity increases and the material works like a closed switch.In this effect, the electrical resistance of the material varies with the amount of incident light.

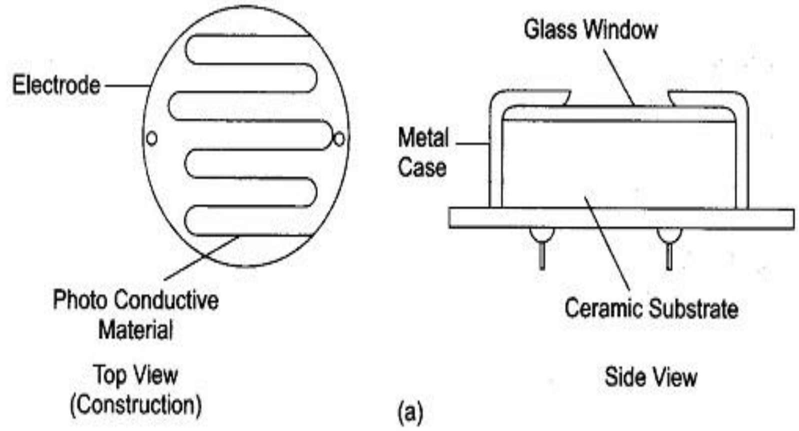


Figure 14.3Schematic structure of the photoconductive cell

The resistance of a typical cell under dark condition (no light condition) may be more than 100 kilo ohms.This resistance is called the dark resistance. When the cell is illuminated, the resistance mayfall to a few hundred ohms and allows the current to rises up. Cell sensitivity may be expressed in terms of the magnitude of the cell current for a given voltage and given level of illumination. The major drawback of the photoconductive cells is that temperature variations cause substantial variations in resistance. Therefore care has to be taken to avoid any fluctuations in the temperature during use.

14.5 PIEZO-ELECTRIC TRANSDUCER

The piezoelectric transducer uses the piezoelectric material which has a special property, i.e. the material induces voltage under pressure or stress applied to it. The material which shows such property is known as the electro-resistive element.

14.5.1 PIEZOELECTRIC EFFECT

The voltage develops because of the displacement of the charges due to the application of pressue. The effect is reversible, i.e. if the varying potential applies to a piezoelectric transducer, it undergoes material deformation or change in the dimension of the material. This effect is known as the piezoelectric effect.

14.5.2 WORKING PRINCIPLE OF PIEZOELECTRIC TRANSDUCER

Piezoelectric Transducer works with the principle of piezoelectricity. The surfaces of the piezoelectric crystal, usually quartzare coated with thin layers of conducting material such as silver. When stress is applied, the ions in the material move towards the conducting surfacesand results in the accumulation of charges. The polarity of the produced charge depends upon the direction of the applied stress. This charge is used for calibration of stress. Stress can be applied in two forms as compressive stress and tensile stress as shown below.The orientation of the crystal also affects the amount of voltage generated. Crystal in a transducer can be arranged in longitudinal position or transverse position.

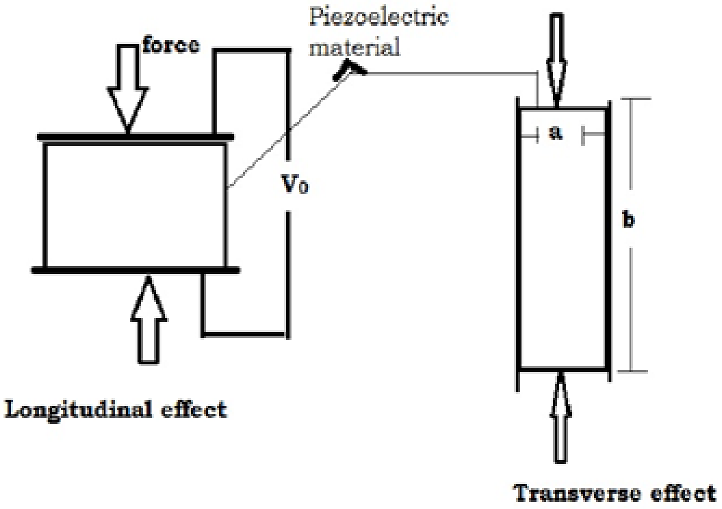


Figure 14.4 Block diagram of crystal in a transducer

A charge amplifier is used to measure the produced charge without dissipation. The magnitude of the voltage across the material is proportional to the pressure applied which is amplified using charge amplifier and used for calibration of applied stress.

The following are the uses of the Piezoelectric transducers.

1. The piezoelectric material has high stability and hence it is used for stabilizing the electronic oscillator.
2. The ultrasonic generators use the piezoelectric material. This generator is used in SONAR for underwater detection and in industrials apparatus for cleaning.

3. It is used in microphones and speakers for converting the electric signal into sound.
4. The piezoelectric material is used in electric lighter.

REFERENCES

1. Microcontroller 8051, D. KarunaSagar, Narosa publication.
2. Introduction to Microcontrollers and their Applications, T.R. Padmanabhan, Narosa publication.
3. Fundamentals of Microprocessors and Microcomputers - B. Ram, 8th Edition, DhanpatRai Publications (P) Ltd., New Delhi, 2010.
4. Microprocessors and Microcontrollers – A. NagoorKani, 2nd Edition, McGraw Hill Education, 2012.
5. Transducers and Instrumentation, D.V.S. Murthy, 2nd Edition, PHI Learning Private Limited -New Delhi, 2008.
6. Microprocessor Architecture, Programming and Applications- R. Gaonkar, Pearson, 1998.
7. Advanced Microprocessor and Principles- A.K. Ray, K.M. Bhurchandi, 3rd Edition, Tata McGraw Hill Publication Co. Ltd. New Delhi. 2003.

MODEL QUESTION PAPER
MICROPROCESSOR AND ELECTRONIC INSTRUMENTATION

Time: 3hrs **PART - A** Max marks:75

I. Answer all the questions not exceeding 50 words each. (10 x 2 = 20)

1. Write the addressing modes of 8085 microprocessor.

2. Give any two examples for logical instructions.

3. Define synchronous data transfer?

4. State the principle of DMA controller.

5. What is a microcontroller?

6. Write the advantages of microcontrollers?

7. How many instructions group are available in 8051?Name them.

8. Give the function of the following instruction.

MOVX, A, @DPTR.

9. Give the principle of a stepper motor.

10. What is meant by interfacing of 8051?

PART – B

II. Write any 5 questions of the following not exceeding 650 words each. (5 x 5 = 25)

11. Explain in detail the data transfer instructions.

12. Write aprogram using 8085 to multiply two 8 bitnumbers.

13. Discuss about asynchronous data transfer.

14. Describe a programmable peripheral interface(8255 A).

15.Short note on microcontroller hardware.

16. Explain the function of timers in a microcontroller.

17. Elaborate about bit manipulations.

18. Briefly explain about the pulse measurement.

PART – C

III. Answer any ONE from each section of the following not exceeding1500 words each (2 x 15 = 30)

19. (a) With suitable diagrams, describe the pins of 8085 andexplain its architecture.

(OR)

19. (b) Explain about :8253 A (PIT) and 8259 (PIC)

20. (a) Elaborately explain the memory structure of 8051.

(OR)

20. (b) Write an 8051 program to find the biggest and smallestnumbers from a given list of numbers.