



## LECTURE HANDOUTS

L1

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

**Topic of Lecture:** Basics of Microprocessor

**Introduction :**

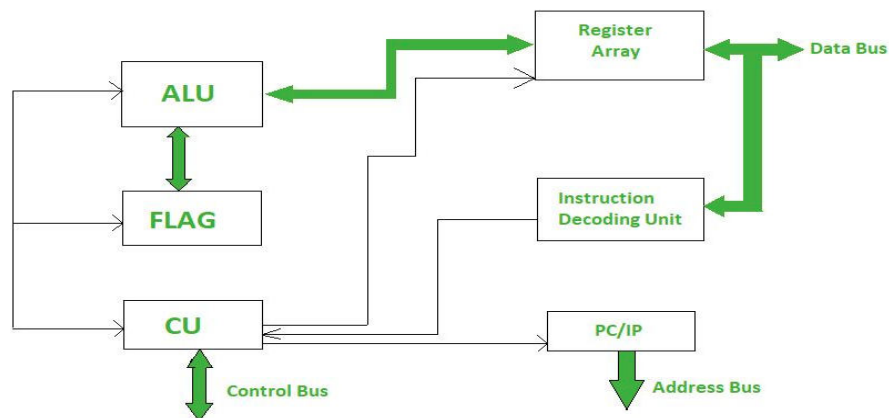
A microprocessor is an electronic component that is used by a computer to do its work. It is a central processing unit on a single integrated circuit chip containing millions of very small components including transistors, resistors, and diodes that work together.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Basics of Digital circuits

**Detailed content of the Lecture:**

A typical Microprocessor structure looks like this.



**Clock Speed of different Microprocessor:**

**16-bit Microprocessor**

- 8086: 4.7MHz, 8MHz, 10MHz
- 8088: more than 5MHz
- 80186/80188: 6MHz

## 32-bit Microprocessor

- INTEL 80386: 16MHz to 33MHz
- INTEL 80486: 16MHz to 100MHz

It does not have any 128-bit Microprocessor in work at present one among the reasons for this is that we are a long way from exhausting the 64 bit address space itself, we use it a constant rate of roughly 2 bits every 3 years. At present we have only used 48 bits of 64 bits so why require 128 bit address space. Also 128 bit Microprocessor would be much slower than the 64 bit Microprocessor.

### Types of Processor:

#### ➤ **Complex Instruction Set Computer (CISC) -**

CISC or Complex Instruction Set Computer is a computer architecture where instructions are such that a single instruction can execute multiple low level operations like loading from memory, storing into memory or an arithmetic operation etc. It has multiple addressing nodes within single instruction. CISC makes use of very few registers.

#### **Example:**

- Intel 386
- Intel 486
- Pentium
- Pentium Pro
- Pentium II
- Pentium III
- Motorola 68000
- Motorola 68020
- Motorola 68040 etc.

#### ➤ **Reduced Instruction Set Computer (RISC)**

RISC or Reduced Instruction Set Computer is a computer architecture where instructions are simple and designed to get executed quickly. Instructions get completed in one clock cycle this is because of the optimization of instructions and pipelining (a technique that allows for simultaneous execution of parts, or stages, of instructions to more efficiently process instructions). RISC makes use of multiple registers to avoid large interactions with memory. It has few addressing nodes.

#### **Example:**

- IBM RS6000
- MC88100
- DEC Alpha 21064
- DEC Alpha 21164

➤ **Explicitly Parallel Instruction Computing (EPIC)**

EPIC or Explicitly Parallel Instruction Computing permits computer to execute instructions parallel using compilers. It allows complex instructions execution without using higher clock frequencies. EPIC encodes its instruction into 128 bit bundles. Each bundle contains three instructions which are encoded in 41 bits each and a 5-bit template field (contains information about types of instructions in bundle and which instructions can be executed in parallel).

**Video Content / Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRPtvj7bFU&noredirect=1)

**Important Books/Journals for further learning including the page nos.:**

B H Khan, Non - Conventional Energy Resources ,Tata McGraw Hill Education Private Limited , New Delhi 2010-Page no (15-19)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L2

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

**Topic of Lecture:** Architecture of 8085 Microprocessor

**Introduction :**

The architecture of 8085 microprocessor mainly includes the timing & control unit, Arithmetic and logic unit, decoder, instruction register; interrupt control, a register array, serial input/output control. The most important part of the **microprocessor** is the central processing unit.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Processors and controllers
- Central processing unit
- Memory Devices

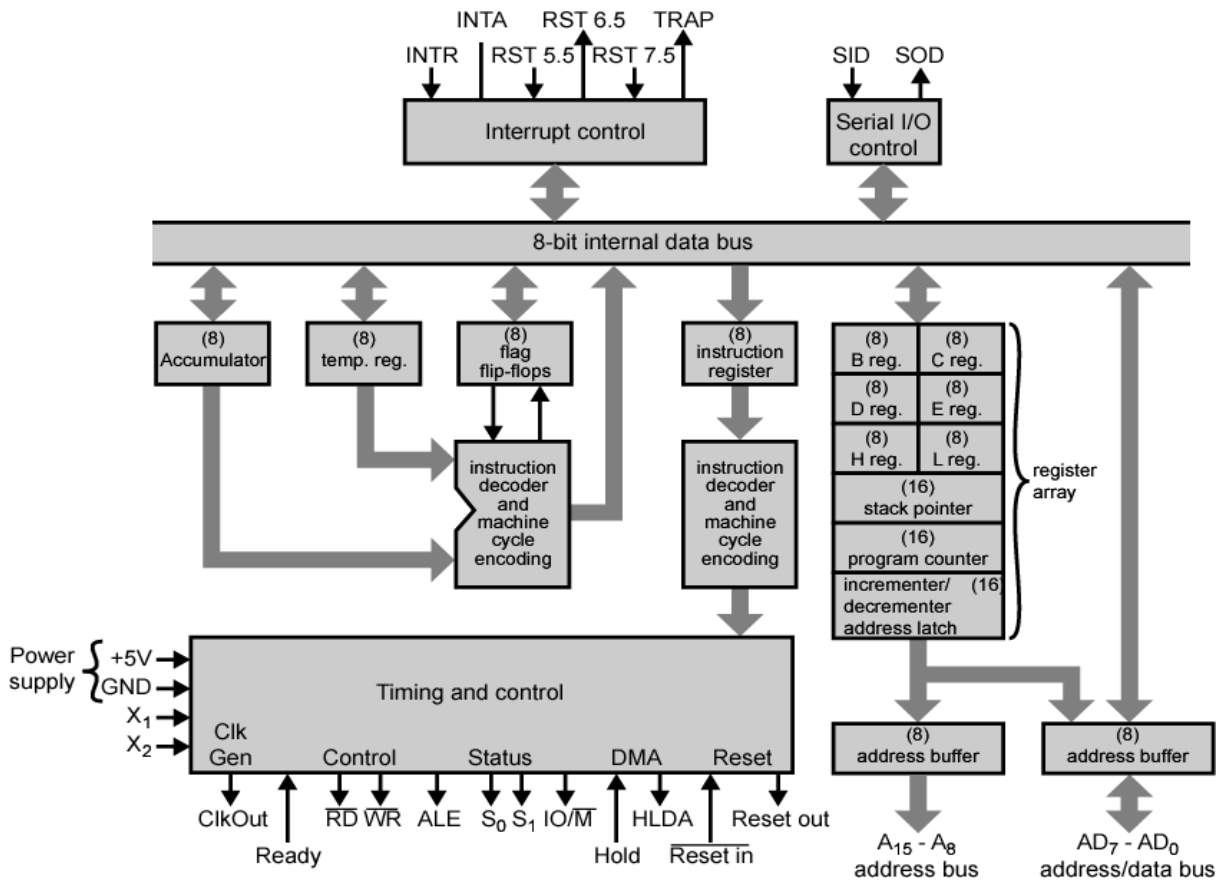
**Detailed content of the Lecture:**

**8085 MICROPROCESSOR ARCHITECTURE**

- 8-bit processor available as a 40-pin IC package
- It uses +5 V for power.
- It can run at a maximum frequency of 3 MHz. Its data bus width is 8-bit and address bus width is 16-bit thus it can address  $2^{16} = 64$  KB of memory.

**Arithmetic and Logic Unit**

- The ALU performs the actual numerical and logical operations such as Addition (ADD), Subtraction (SUB), AND, OR etc.
- It uses data from memory and from Accumulator to perform operations.
- The results of the arithmetic and logical operations are stored in the accumulator.



### Registers

- The 8085 includes six registers, one accumulator and one flag register, as shown in Figure. In addition, it has two 16-bit registers: stack pointer and program counter.
- The 8085 has six general-purpose registers to store 8-bit data; these are identified as B, C, D, E, H and L. they can be combined as register pairs - BC, DE and HL to perform.

### Accumulator

- The accumulator is an 8-bit register that is a part of ALU.
- This register is used to store 8-bit data and to perform arithmetic and logical operations.
- The result of an operation is stored in the accumulator.
- The accumulator is also identified as register A.

### Flag register

- The ALU includes five flip-flops, which are set or reset after an operation according to data condition of the result in the accumulator and other registers.
- They are called Zero (Z), Carry (CY), Sign (S), Parity (P) and Auxiliary Carry (AC) flags.
- Their bit positions in the flag register are shown in Figure. The microprocessor uses these flags to test data conditions.



D7	D6	D5	D4	D3	D2	D1	D0
S	Z	AC		P			CY

Flag register

### Program Counter (PC)

- This 16-bit register deals with sequencing the execution of instructions. This register is a memory pointer.
- The microprocessor uses this register to sequence the execution of the instructions.
- The function of the program counter is to point to the memory address from which the next byte is to be fetched.
- When a byte is being fetched, the program counter is automatically incremented by one to point to the next memory location.

### Stack Pointer (SP)

- The stack pointer is also a 16-bit register, used as a memory pointer.
- It points to a memory location in R/W memory, called stack.
- The beginning of the stack is defined by loading 16-bit address in the stack pointer.

### Instruction Register/Decoder

- It is an 8-bit register that temporarily stores the current instruction of a program.
- Latest instruction sent here from memory prior to execution.
- Decoder then takes instruction and decodes or interprets the instruction.

### Control Unit

- Generates signals on data bus, address bus and control bus within microprocessor to carry out the instruction, which has been decoded.

### **Video Content / Details of website for further learning (if any):**

[https://www.youtube.com/watch?v=\\_qtCAuLIaew](https://www.youtube.com/watch?v=_qtCAuLIaew)

### **Important Books/Journals for further learning including the page nos.:**

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (24-30)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L3

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

**Topic of Lecture:** Pin Diagram

### Introduction :

The 40 pins of the microprocessor can be divided into six groups such as address bus, data bus, control signals & status signals; power supply & frequency, externally started signals and serial input/output ports.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Basics of digital electronics

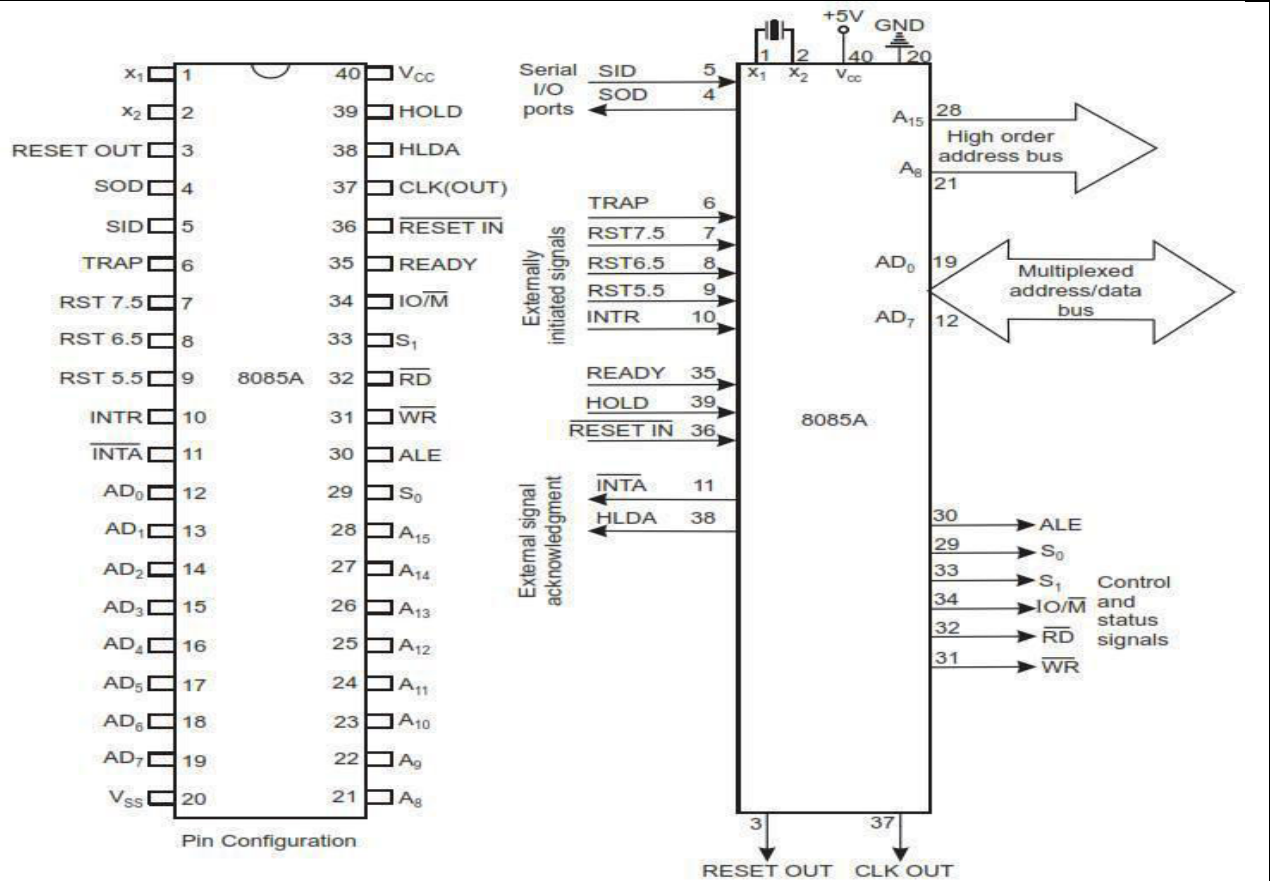
### Detailed content of the Lecture:

#### Timing and Control Unit

- ✓ The timing and control unit accepts information from the instruction decoder and generates different control signal. This unit synchronizes all the microprocessor operation and generates control and status signal necessary for communication between the microprocessor and peripherals.

#### A8 - A15 (Output 3 State):

- ✓ Address Bus: The most significant 8 bits of the memory address or the 8 bits of the I/O address, 3 states during Hold and Halt modes.



### AD0 - AD7 (Input / Output 3 state):

- Multiplexed Address/Data Bus; Lower 8 bits of the memory address (or I/O address) appear on the bus during the first clock cycle of a machine state.
- It then becomes the data bus during the second and third clock cycles. 3 stated during Hold and Halt modes.

### ALE (Output):

- Address Latch Enable: It occurs during the first clock cycle of a machine state and enables the address to get latched into the on chip latch of peripherals.
- The falling edge of ALE is set to guarantee setup and hold times for the address information. ALE can also be used to strobe the status information. ALE is never 3stated.

### SO, S1 (Output):

Data Bus Status. Encoded status of the bus cycle:

S1 S0

0 0 HALT

0 1 WRITE

1 0 READ

1 1 FETCH S1 can be used as an advanced R/W status.

### RD (Output 3state):

READ: indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer.



**WR (Output 3state):**

- WRITE: indicates the data on the Data Bus is to be written into the selected memory or I/O location.
- Data is set up at the trailing edge of WR. 3stated during Hold and Halt modes.

**READY (Input):**

- If Ready is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data.
- If Ready is low, the CPU will wait for Ready to go high before completing the read or write cycle.

**HOLD (Input):**

- HOLD: indicates that another Master is requesting the use of the Address and Data Buses.
- The CPU, upon receiving the Hold request will relinquish the use of buses as soon as the completion of the current machine cycle.
- Internal processing can continue. The processor can regain the buses only after the Hold is removed. When the Hold is acknowledged, the Address, Data, RD, WR, and IO/M lines are 3stated.

**HLDA (Output):**

- HOLD ACKNOWLEDGE: indicates that the CPU has received the Hold request and that it will relinquish the buses in the next clock cycle.
- HLDA goes low after the Hold request is removed. The CPU takes the buses one half clock cycle after HLDA goes low.

**INTR (Input):**

- INTERRUPT REQUEST is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of the instruction. If it is active, the Program Counter (PC) will be inhibited from incrementing and an INTA will be issued.
- During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.

**INTA (Output):**

- INTERRUPT ACKNOWLEDGE: is used instead of (and has the same timing as) RD during the Instruction cycle after an INTR is accepted.
- It can be used to activate the 8259 Interrupt chip or some other interrupt port.

**RESTART INTERRUPTS:**

These three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted.

RST 7.5 Highest Priority

RST 6.5

RST 5.5 Lowest Priority

**RESET IN (Input):**

- Reset sets the Program Counter to zero and resets the Interrupt Enable and HLDA flipflops.
- None of the other flags or registers (except the instruction register) are affected. The CPU is held in the reset condition as long as Reset is applied.

**RESET OUT (Output):**

- ✓ Indicates CPU is being reset. Can be used as a system RESET. The signal is synchronized to the processor clock.

**X1, X2 (Input):**

Crystal or R/C network connections to set the internal clock generator X1 can also be an external clock input instead of a crystal. The input frequency is divided by 2 to give the internal operating frequency.

**CLK (Output):**

Clock Output for use as a system clock when a crystal or R/C network is used as an input to the CPU. The period of CLK is twice the X1, X2 input period.

**IO/M (Output):**

IO/M indicates whether the Read/Write is to memory or I/O Tristated during Hold and Halt modes.

**SID (Input):**

Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.

**SOD (output):**

Serial output data line. The output SOD is set or reset as specified by the SIM instruction.

**Vcc:** +5 volt supply.

**Vss:** Ground Reference.

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=ii7PCV2zvms>

**Important Books/Journals for further learning including the page nos.:**

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (24-30)



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



## LECTURE HANDOUTS

L4

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

<b>Topic of Lecture:</b> Instruction Set			
<b>Introduction :</b>			
<ul style="list-style-type: none"> <li>➤ An instruction of a computer is a command given to the computer to perform a specified operation on given data.</li> <li>➤ In microprocessor, the instruction set is the collection of the instructions that the microprocessor is designed to execute.</li> </ul>			
<b>Prerequisite knowledge for Complete understanding and learning of Topic:</b>			
<ul style="list-style-type: none"> <li>➤ Digital Concepts</li> </ul>			
<b>Detailed content of the Lecture:</b>			
<b>DATA TRANSFER GROUP:</b>			
Opcode	Operand	Meaning	Example
MOV	Rd, Sc M, Sc Dt, M	Copy from the source (Sc) to the destination(Dt)	<b>Example</b> – MOV K, L
MVI	Rd, data M, data	Move immediate 8-bit	<b>Example</b> – MVI K, 55L
LDA	16-bit address	Load the accumulator	<b>Example</b> – LDA 2034K
LDAX	B/D Reg. pair	Load the accumulator indirect	<b>Example</b> – LDAX K
LXI	Reg. pair, 16-bit data	Load the register pair immediate	<b>Example</b> – LXI K, 3225L
LHLD	16-bit address	Load H and L registers direct	<b>Example</b> – LHLD 3225K
STA	16-bit address	16-bit address	<b>Example</b> – STA 325K
STAX	16-bit address	Store the accumulator indirect	<b>Example</b> – STAX K

SHLD	16-bit address	Store H and L registers direct	<b>Example</b> – SHLD 3225K
XCHG	None	Exchange H and L with D and E	<b>Example</b> – XCHG
SPHL	None	Copy H and L registers to the stack pointer	<b>Example</b> – SPHL
XTHL	None	Exchange H and L with top of stack	<b>Example</b> – XTHL
PUSH	Reg. pair	Push the register pair onto the stack	<b>Example</b> – PUSH K
POP	Reg. pair	Pop off stack to the register pair	the <b>Example</b> – POPK
OUT	8-bit port address	Output the data from the accumulator to a port with 8bit address	<b>Example</b> – OUT K9L
IN	8-bit port address	Input data to accumulator from a port with 8-bit address	<b>Example</b> – IN5KL

#### CONTROL INSTRUCTIONS:

Opcode	Operand	Meaning	Explanation
NOP	None	No operation	No operation is performed, i.e., the instruction is fetched and decoded.
HLT	None	Halt and enter wait state	The CPU finishes executing the current instruction and stops further execution. An interrupt or reset is necessary to exit from the halt state.
DI	None	Disable interrupts	The interrupt enable flip-flop is reset and all the interrupts are disabled except TRAP.
EI	None	Enable interrupts	The interrupt enable flip-flop is set and all the interrupts are enabled.
RIM	None	Read interrupt mask	This instruction is used to read the status of interrupts 7.5, 6.5, 5.5 and read serial data input bit.
SIM	None	Set interrupt mask	This instruction is used to implement the interrupts 7.5, 6.5, 5.5, and serial data output.

**LOGICAL INSTRUCTIONS:**

Opcode	Operand	Meaning	Explanation
CMP	R M	Compare the register or memory with the accumulator	The contents of the operand (register or memory) are M compared with the contents of the accumulator.
CPI	8-bit data	Compare immediate with the accumulator	The second byte data is compared with the contents of the accumulator.
ANA	R M	Logical AND register or memory with the accumulator	The contents of the accumulator are logically AND with M the contents of the register or memory, and the result is placed in the accumulator.
ANI	8-bit data	Logical AND immediate with the accumulator	The contents of the accumulator are logically AND with the 8-bit data and the result is placed in the accumulator.
XRA	R M	Exclusive OR register or memory with the accumulator	The contents of the accumulator are Exclusive OR with M the contents of the register or memory, and the result is placed in the accumulator.
XRI	8-bit data	Exclusive OR immediate with the accumulator	The contents of the accumulator are Exclusive OR with the 8-bit data and the result is placed in the accumulator.
ORA	R M	Logical OR register or memory with the accumulator	The contents of the accumulator are logically OR with M the contents of the register or memory, and result is placed in the accumulator.
ORI	8-bit data	Logical OR immediate with the accumulator	The contents of the accumulator are logically OR with the 8-bit data and the result is placed in the accumulator.
RLC	None	Rotate the accumulator left	Binary bit of the accumulator is rotated left by one position. Bit D7 is placed in the position of D0 as well as in the Carry flag.
RRC	None	Rotate the accumulator right	Each binary bit of the accumulator is rotated right by one position. Bit D0 is placed in the position of D7 as well as in the Carry flag. CY is modified according to bit D0.
RAL	None	Rotate the accumulator left through carry	Each binary bit of the accumulator is rotated left by one position through the Carry flag. Bit D7 is placed in the Carry flag, and the Carry flag is placed in the least significant position D0.

RAR	None	Rotate the accumulator right through carry	Each binary bit of the accumulator is rotated right by one position through the Carry flag. Bit D0 is placed in the Carry flag, and the Carry flag is placed in the most significant position D7. CY is modified according to bit D0.
CMA	None	Complement accumulator	The contents of the accumulator are complemented. No flags are affected.
CMC	None	Complement carry	The Carry flag is complemented. No other flags are affected.
STC	None	Set Carry	Set Carry

#### ARITHMETIC INSTRUCTUIONS:

Opcode	Operand	Meaning	Example
ADD	R M	Add register or memory, to the accumulator	<b>Example</b> – ADD K.
ADC	R M	Add register to the accumulator with carry	<b>Example</b> – ADC K
ADI	8-bit data	Add the immediate to the accumulator	<b>Example</b> – ADI 55K
ACI	8-bit data	Add the immediate to the accumulator with carry	<b>Example</b> – ACI 55K
LXI	Reg. pair, 16bit data	Load the register pair immediate	<b>Example</b> – LXI K, 3025M
DAD	Reg. pair	Add the register pair to H and L registers	<b>Example</b> – DAD K
SUB	R M	Subtract the register or the memory from the accumulator	<b>Example</b> – SUB K
SBB	R M	Subtract the source and borrow from the accumulator	<b>Example</b> – SBB K
SUI	8-bit data	Subtract the immediate from the accumulator	<b>Example</b> – SUI 55K
SBI	8-bit data	Subtract the immediate from the accumulator with borrow	<b>Example</b> – XCHG
INR	R M	Increment the register or the memory by 1	<b>Example</b> – INR K

INX	R	Increment register pair by 1	<b>Example</b> – INX K
DCR	R M	Decrement the register or the memory by 1	<b>Example</b> – DCR K
DCX	R	Decrement the register pair by 1	<b>Example</b> – DCX K
DAA	None	Decimal adjust accumulator	<b>Example</b> – DAA

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=G3iUO96XhC4>

**Important Books/Journals For Further Learning Including The Page Nos.:**

- Gaonkar.R.S, Microprocessor Architecture Programming And Applications With 8085,Wiley Eastern Ltd, New Delhi 2013-Page No(12-16)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L5

LECTURE HANDOUTS

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

**Topic of Lecture:** Addressing Modes

**Introduction :**

- Each instruction requires some data on which it has to operate.
- There are different techniques to specify data for instructions.
- These techniques are called addressing modes.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Instruction Sets

**Detailed content of the Lecture:**

Intel 8085 uses the following addressing modes:

- **Direct Addressing**

In this addressing mode, the address of the operand (data) is given in the instruction itself.

**Example**

**STA 2400H:** It stores the content of the accumulator in the memory location 2400H.

**32, 00, 24:** The above instruction in the code form.

In this instruction, 2400H is the memory address where data is to be stored.

It is given in the instruction itself. The 2nd and 3rd bytes of the instruction specify the address of the memory location.

Here, it is understood that the source of the data is accumulator.

- **Register Addressing**

In register addressing mode, the operand is in one of the general purpose registers. The Opcode specifies the address of the register(s) in addition to the operation to be performed.



### Example:

MOV A, B: Move the content of B register to register A.

**78:** The instruction in the code form.

In the above example, MOV A, B is 78H. Besides the operation to be performed the opcode also specifies source and destination registers.

The opcode 78H can be written in binary form as 01111000. The first two bits, i.e. 0 1 are for MOV operation, the next three bits 1 1 1 are the binary code for register A, and the last three bits 000 are the binary code for register B.

- **Register Indirect Addressing**

In Register Indirect mode of addressing, the address of the operand is specified by a register pair.

### Example

- **LXI H, 2500 H** - Load H-L pair with 2500H.
- **MOV A, M** - Move the content of the memory location, whose address is in H-L pair (i.e. 2500 H) to the accumulator.
- **HLT** - Halt.

In the above program the instruction MOV A, M is an example of register indirect addressing. For this instruction, the operand is in the memory. The address of the memory is not directly given in the instruction. The address of the memory resides in H-L pair and this has already been specified by an earlier instruction in the program, i.e. LXI H, 2500 H.

- **Immediate Addressing**

In this addressing mode, the operand is specified within the instruction itself.

### Example

LXI H, 2500 is an example of immediate addressing. 2500 is 16-bit data which is given in the instruction itself. It is to be loaded into H-L pair.

- **Implicit Addressing**

There are certain instructions which operate on the content of the accumulator. Such instructions do not require the address of the operand.

### Example

CMA, RAL, RAR, etc.

### Video Content/ Details of website for further learning (if any):

<https://www.youtube.com/watch?v=nJfTvmYjR90>

**Important Books/Journals for further learning including the page nos.:**

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (156-188)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L6

EEE

III/IV

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

**Topic of Lecture:** Interrupts of 8085

**Introduction :** □

- Interrupt is signals send by an external device to the processor, to request the processor to perform a particular task or work.
- The processor will check the interrupts always at the 2nd T-state of last machine cycle.
- If there is any interrupt it accept the interrupt and send the INTA (active low) signal to the peripheral.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Processors and controllers

□

**Detailed content of the Lecture:**

- The processor executes an interrupt service routine (ISR) addressed in program counter.
- It returned to main program by RET instruction.

**Types of Interrupts:**

It supports two types of interrupts.

- Hardware
- Software

**Software Interrupts:**

Interrupt number \* 8 = vector address

For RST 5,5 \* 8 = 40 = 28H

Vector addresses of all interrupts

**Hardware interrupts:**

If the interrupt is accepted then the processor executes an interrupt service routine.

The 8085 has five hardware interrupts

(1) TRAP      (2) RST 7.5      (3) RST 6.5      (4) RST 5.5      (5) INTR

□ The

### **(1)TRAP:**

TRAP has the highest priority and vectored interrupt.

In sudden power failure, it executes a ISR and send the data from main memory to backup memory.

There are two ways to clear TRAP interrupt.

- By resetting microprocessor (External signal)
- By giving a high TRAP ACKNOWLEDGE (Internal signal)

### **(2)RST 7.5:**

- The RST 7.5 interrupt is a maskable interrupt highest priority.
- It is edge sensitive. ie. Input goes to high and no need to maintain high state until it recognized.

### **Maskable interrupt:**

It is disabled by,

1. DI instruction
2. System or processor reset.
3. After reorganization of interrupt.

### **(3)RST 6.5 and 5.5:**

The RST 6.5 has the third priority whereas RST 5.5 has the fourth priority. INTR is a maskable interrupt. It is disabled by,

1. DI, SIM instruction
2. System or processor reset.
3. After reorganization of interrupt

1. It has lowest priority. It is a level sensitive interrupts. ie. Input goes to high and it is necessary to maintain high state until it recognized. The following sequence of events occurs when INTR signal goes high. The 8085 checks the status of INTR signal during execution of each instruction.

2. If INTR signal is high, then 8085 complete its current instruction and sends active low interrupt acknowledge signal, if the interrupt is enabled.

3. In response to the acknowledge signal, external logic places an instruction OPCODE on the data bus. In the case of multibyte instruction, additional interrupt acknowledge machine cycles are generated by the 8085 to transfer the additional bytes into the microprocessor.

4. On receiving the instruction, the 8085 save the address of next instruction on stack and execute received instruction.

### **SIM and RIM for interrupts:**

- The 8085 provide additional masking facility for RST 7.5, RST 6.5 and RST 5.5 using SIM instruction.
- The status of these interrupts can be read by executing RIM instruction.
- The masking or unmasking of RST 7.5, RST 6.5 and RST 5.5 interrupts can be performed by moving an 8-bit data to accumulator and then executing SIM instruction.
- The status of pending interrupts can be read from accumulator after executing RIM instruction.

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=nJfTvmYjR90>

**Important Books/Journals for further learning including the page nos.:**

Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (352-370)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L7

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

**Topic of Lecture:** Timing Diagram Opcode Fetch Machine Cycle

**Introduction :**

- Timing Diagram is a graphical representation.
- It represents the execution time taken by each instruction in a graphical format. The execution time is represented in T-states.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

**Detailed content of the Lecture:**

**Instruction Cycle**

The time required to execute an instruction is called instruction cycle.

**Machine Cycle**

The time required to access the memory or input/output devices is called machine cycle.

**T-State**

The machine cycle and instruction cycle takes multiple clock periods.

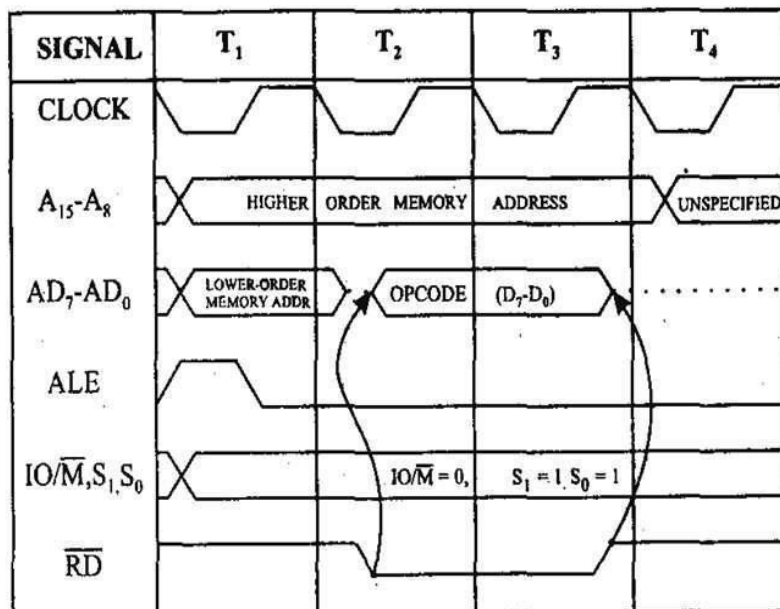
**Machine cycles of 8085**

The 8085 microprocessor has 5 (seven) basic machine cycles. They are

1. Opcode fetch cycle (4T)
2. Memory read cycle (3 T)
3. Memory write cycle (3 T)
4. I/O read cycle (3 T)
5. I/O write cycle (3 T)

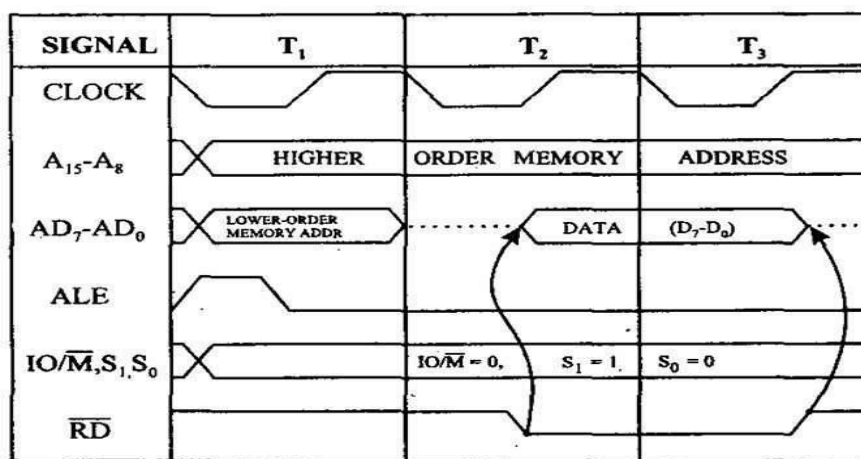
**1.Opcode fetch machine cycle of 8085 :**

- Each instruction of the processor has one byte opcode
- 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.



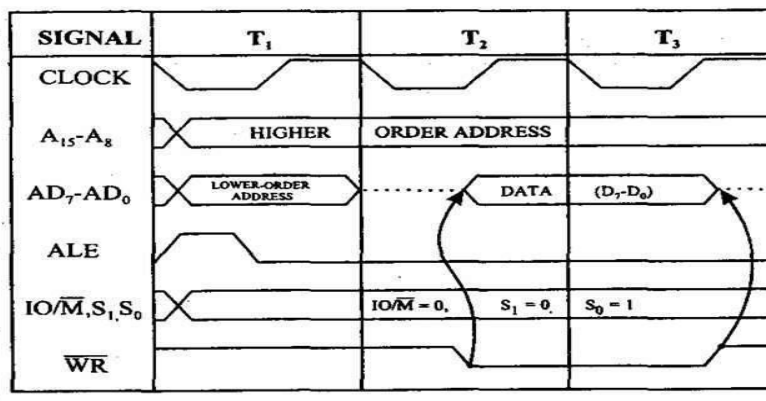
### 2. Memory Read Machine Cycle of 8085:

- The memory read machine cycle is executed by the processor to read a data byte from memory.
- The processor takes 3T states to execute this cycle.
- The instructions which have more than one byte word size will use the machine cycle after the opcode fetch machine cycle



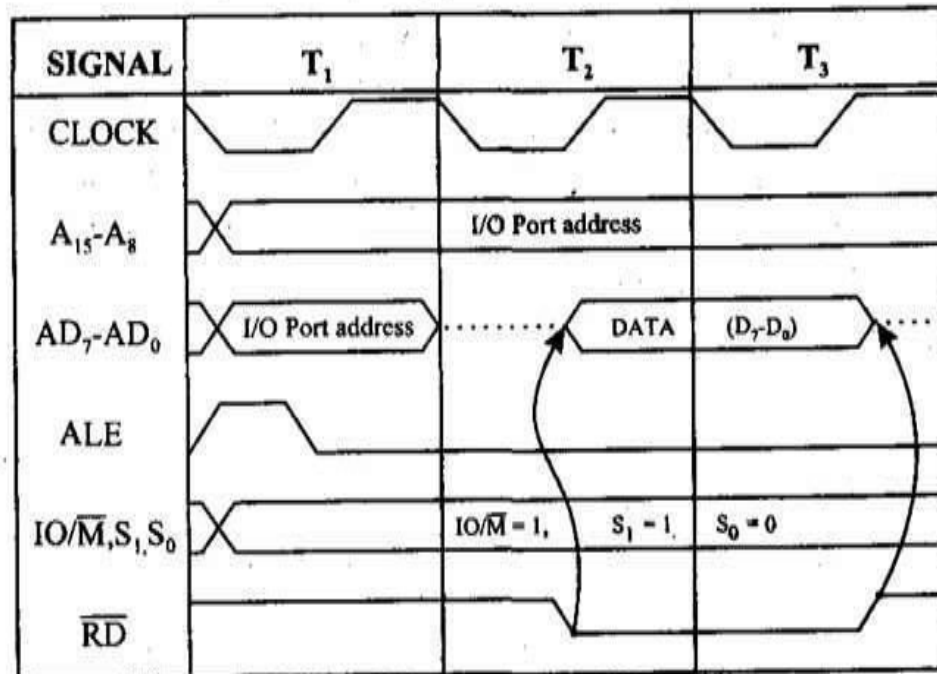
### 3. Memory Write Machine Cycle of 8085

The memory write machine cycle is executed by the processor to write a data byte in a memory location



#### 4. I/O Read Cycle of 8085

- The I/O Read cycle is executed by the processor to read a data byte from I/O port or from the peripheral, which is I/O, mapped in the system.
- The processor takes 3T states to execute this machine cycle.
- The IN instruction uses this machine cycle during the execution

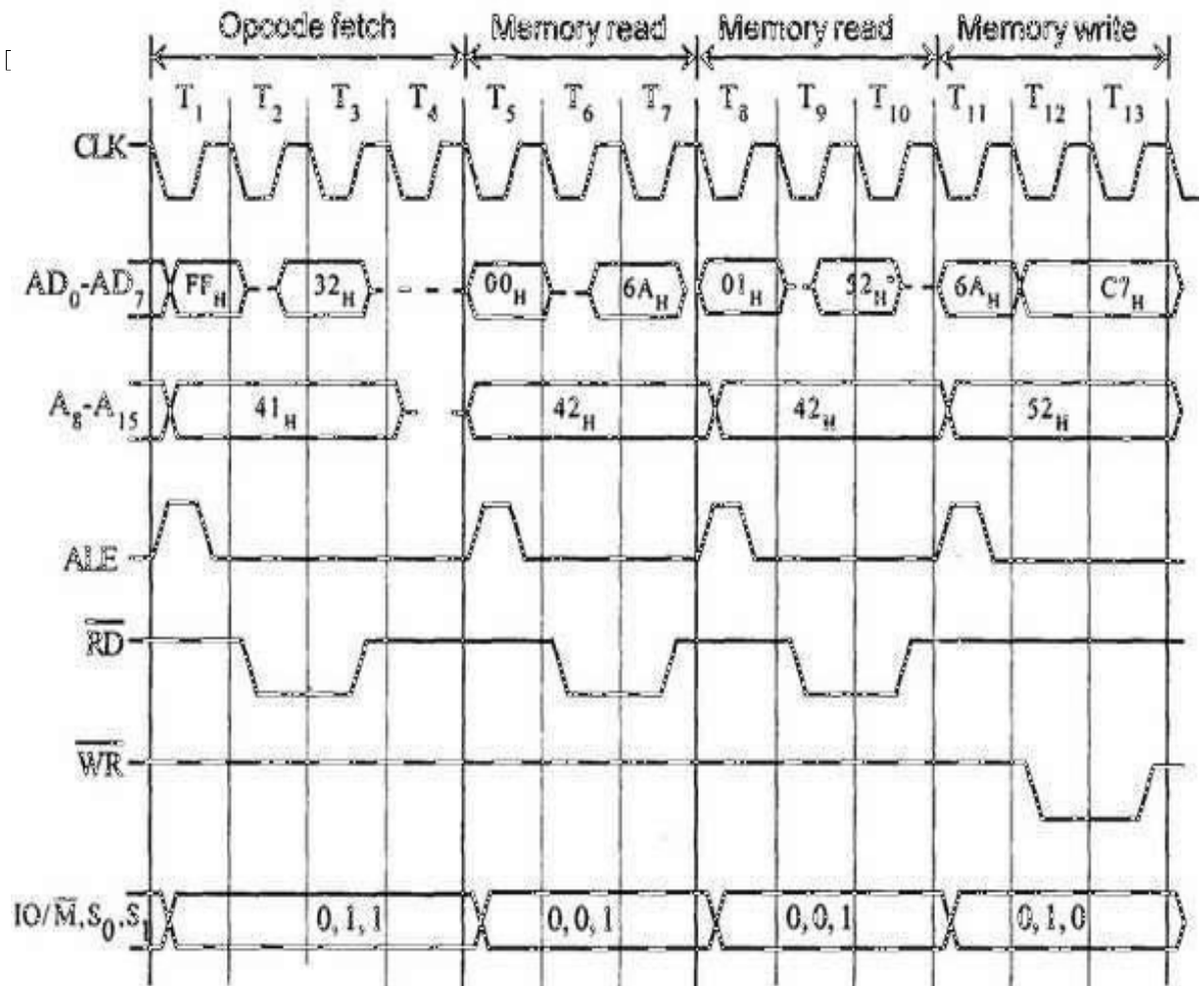


I/O Read Cycle

#### Timing diagram for STA 526AH

- STA means Store Accumulator -The contents of the accumulator is stored in the specified address(526A).
- The opcode of the STA instruction is said to be 32H. It is fetched from the memory
- 41FFH(see fig). - OF machine cycle
- Then the lower order memory address is read(6A). - Memory Read Machine Cycle
- Read the higher order memory address (52).- Memory Read Machine Cycle
- The combination of both the addresses are considered and the content from accumulator is written in 526A. - Memory Write Machine Cycle





Timing Diagram for STA 526A H

### Timing diagram for INR M

- Fetching the Opcode 34H from the memory 4105H. (OF cycle)
- Let the memory address (M) be 4250H. (MR cycle -To read Memory address and data) Let the content of that memory is 12H.
- Increment the memory content from 12H to 13H. (MW machine cycle)

### Video Content/ Details of website for further learning (if any):

<https://www.youtube.com/watch?v=EaRqC3dLoWY>

### Important Books/Journals for further learning including the page nos.:

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (78-83)



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L8

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

**Topic of Lecture:** Memory Read & Write Cycle

**Introduction :**

- Memory-mapping is a mechanism that maps a portion of a file, or an entire file, on disk to a range of addresses within an application's address space.
- The application can then access files on disk in the same way it accesses dynamic memory.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Timing Diagram

**Detailed content of the Lecture:**

Mapped Memory

- Simplicity is the advantage of mapping memory among the devices in a system:
  - There is no need to partition data. All devices see the complete memory image.
  - There is no need to allocate space in device memory or to manually copy data. All data transfers are implicitly performed by the kernel as needed.
- There is no need to use streams to overlap data transfers with kernel execution. All data transfers originate from the kernel and are asynchronous.

Application performance is the cost associated with this simplicity. Using mapped memory does mean that the programmer gives up control over the data movement between the host and devices. From the forums and experience, it is not unusual for kernel performance to drop when using mapped memory because there are no guarantees when or how often data will need to be transferred across the PCIe bus. Other considerations to using mapped memory include:

- If the contents of the mapped memory are modified, the application must synchronize memory accesses using streams or events to avoid any potential read-after-write, write-after-read, or write-after write hazards.
- The host memory needs to be page aligned. The simplest and most portable way to enforce this is to use `std::aligned_storage` when allocating mapped host memory.

The simplicity of using mapped memory is illustrated by `mapped_memory` which fills a mapped memory vector using one or more GPUs in the system. The highlighted command `mapped_memory` creates a mapped region of memory when passed the `mapped` flag. This region is freed at the end of the program with `mapped_memory::free`. Thrust was used to make this code concise and easy to read.

The `device_pointer_cast` method was used to correctly cast the mapped host memory for the `thrust::copy` sequence method.

The highlighted call to `mapped_memory::sync` ensures that the mapped data is synchronized between the host and devices prior to checking the results on the host. All data transfers occur transparently and asynchronously. Finally, the contents of the mapped region of memory are checked for correctness on the host and the mapped region is freed.

### Memory Maps

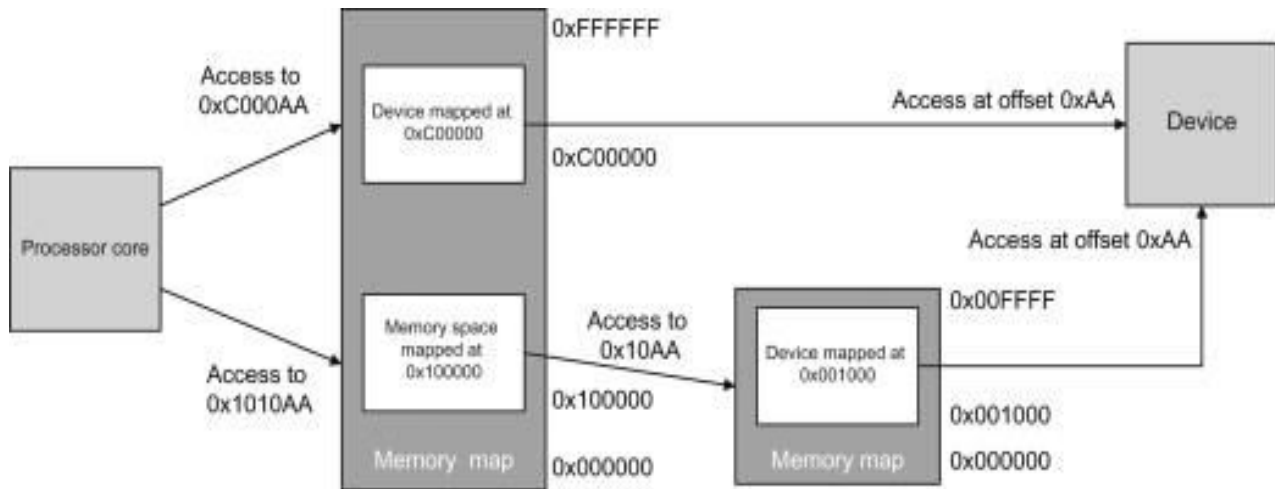
- The memory map is a fundamental service provided by the Simics framework and is a core component to enabling very fast simulation.
- Memory maps take care of routing memory accesses from their source to their destination, based on the address of the access.
- Memory maps replace the explicit modeling of buses in Simics. Memory maps provide the software view of the bus system, but not the hardware view.
- It is a typical transaction-level abstraction from the hardware behavior. Simics memory maps are dynamic and can be changed during the simulation, both by the user and more commonly from various devices and components that manipulate memory mappings to implement dynamic target behaviors.

An example of the memory map for a simple system. It is not a complete system, but instead it is just intended to show a few typical cases of memory mappings. The processor has a memory map that shows how RAM and a few devices are mapped, as well as a PCIe memory space in which PCIe devices are mapped.

The basic building block for creating memory maps are objects of the `memory-space` class. When a memory transaction is initiated from an origin, typically a processor model, the address  $a$  of the access is passed unmodified to the first memory space.

For a processor model, the first memory space typically represents the physical address space of the processor. If nothing is mapped at address  $a$ , an access error is signaled to the origin.

However, if the access falls within the range of a second memory space that is mapped in the first memory space at offset  $b$ , the transaction is passed on to the second memory space with the local address  $c=a-b$ . Memory spaces are traversed in this fashion until an unmapped address is reached or the transaction terminates in a device model. This process is shown in Figure.



**Video Content/ Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=jkT9Bgz8PAg>

**Important Books/Journals for further learning including the page nos.:**

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (41-45)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L9

## LECTURE HANDOUTS

EEE

III/IV

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : I - 8085 PROCESSOR

Date of Lecture:

**Topic of Lecture:** Memory Organization

### Introduction :

- The memory is made up of semiconductor material used to store the programs and data.
- Data in Microprocessor can move from one location to another.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Classification of Memories

### Detailed content of the Lecture:

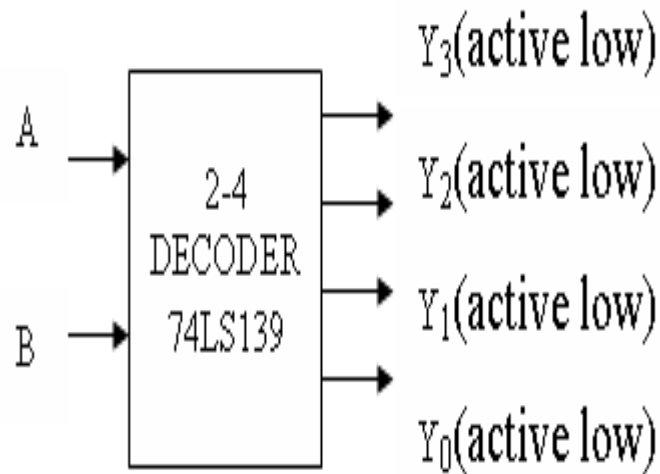
#### Memory Interfacing

Three types of memory is,

- Process memory
- Primary or main memory
- Secondary memory
- 

#### Typical EPROM and Static RAM

- Having two power supply pins (one for connecting required supply voltage and the other for connecting ground).
- The control signals needed for static RAM are chip select (chip enable), read control (output enable) and write control (write enable).
- The control signals needed for read operation in EPROM are chip select (chip enable) and read control (output enable).



**Block Diagram of 2-4 Decoder**

**Truth Table for 2-4 Decoder**

Input		Output			
B	A	$\bar{Y}_3$	$\bar{Y}_2$	$\bar{Y}_1$	$\bar{Y}_0$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

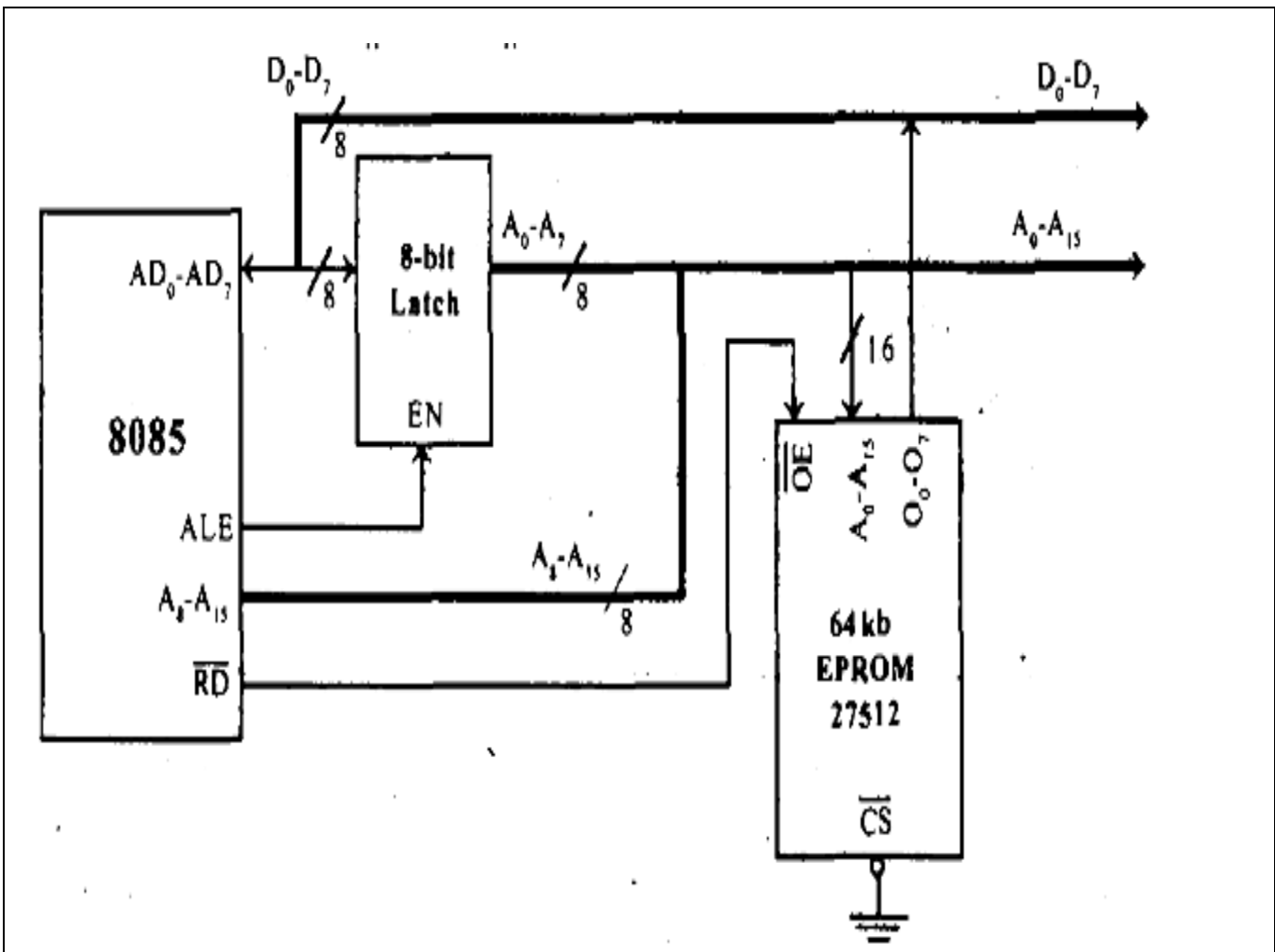
**Example for Memory Interfacing**

Consider a system in which the full memory space 64kb is utilized for EPROM memory. Interface the EPROM with 8085 processor.

The memory capacity is 64 Kbytes. i.e  
 $2^n = 64 \times 1000$  bytes where  $n =$

address lines. So,  $n = 16$ .

- In this system the entire 16 address lines of the processor are connected to address input pins of memory IC in order to address the internal locations of memory.
- The chip select (CS) pin of EPROM is permanently tied to logic low (i.e., tied to ground).
- Since the processor is connected to EPROM, the active low RD pin is connected to active low output enable pin of EPROM.



**Video Content/ Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=jkT9Bgz8PAg>

**Important Books/Journals for further learning including the page nos.:**

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (30-41)

Course Faculty



Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

Topic of Lecture: Basics of Microcontroller

### Introduction :

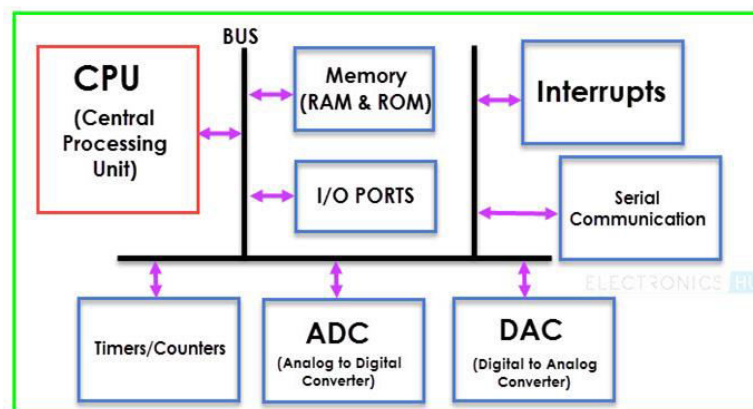
- Microcontroller performs some basic operations like addition, subtraction, multiplication, division and some logical operations using its Arithmetic and Logical Unit (ALU).
- New Microcontroller also perform operations on floating point numbers also.
- Data in Microcontroller can move from one location to another.
- It has a Program Counter (PC) register that stores the address of next instruction based on the value of PC, Microcontroller jumps from one location to another and takes decision.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Processors and controllers

### Detailed content of the Lecture:

A typical Microcontroller structure looks like this.



From the above image, you can understand that the three important (or major) components of a Microcontroller are:



- The CPU (Central Processing Unit)
- The Memory and
- The I/O Ports

This doesn't mean that other components are of less importance. But these can be considered as supporting devices. We will now see each of the Basic Components of a Microcontroller mentioned in the above structure.

## **CPU**

Central Processing Unit or CPU is the brain of the Microcontroller. It consists of an Arithmetic Logic Unit (ALU) and a Control Unit (CU). A CPU reads, decodes and executes instructions to perform Arithmetic, Logic and Data Transfer operations.

## **Memory**

Any Computational System requires two types of Memory: Program Memory and Data Memory. Program Memory, as the name suggests, contains the program i.e. the instructions to be executed by the CPU. Data Memory on the other hand, is required to store temporary data while executing the instructions.

Usually, Program Memory is a Read Only Memory or ROM and the Data Memory is a Random Access Memory or RAM. Data Memory is sometimes called as Read Write Memory (R/W M).

## **I/O Ports**

The interface for the Microcontroller to the external world is provided by the I/O Ports or Input/Output Ports. Inputs device like Switches, Keypads, etc. provide information from the user to the CPU in the form of Binary Data.

The CPU, upon receiving the data from the input devices, executes appropriate instructions and gives response through Output Devices like LEDs, Displays, Printers, etc.

## **Bus**

Another important component of a Microcontroller, but rarely discussed is the System Bus. A System bus is a group of connecting wire that connect the CPU with other peripherals like Memory, I/O Ports and other supporting components.

## **Timers/Counters**

One of the important components of a Microcontroller are the Timers and Counters. They provide the operations of Time Delays and counting external events. Additionally, Timers and Counters can provide Function Generation, Pulse Width Modulation, Clock Control, etc.

## **Serial Port**

One of the important requirement of a Microcontroller is to communicate with other device and peripherals (external). Serial Port proves such interface through serial communication. Most common serial communication implemented in Microcontrollers is UART.

## **Interrupts**

A very important feature of a Microcontroller is Interrupts and its Interrupt Handling Mechanism. Interrupts can be external, internal, hardware related or software related.

### **ADC (Analog to Digital Converter)**

Analog to Digital Converter or ADC is a circuit that converts Analog signals to Digital Signals. The ADC Circuit forms the interface between the external Analog Input devices and the CPU of the Microcontroller. Almost all sensors are analog devices and the analog data from these sensors must be converted in to digital data for the CPU to understand.

### **DAC (Digital to Analog Converter)**

Digital to Analog Converter or DAC is a circuit, that works in contrast to an ADC i.e. it converts Digital Signals to Analog Signals. DAC forms the bridge between the CPU of the Microcontroller and the external analog devices.

### **Advantages of Microcontrollers**

- A Microcontroller is a true device that fits the computer-on-a-chip idea.
- No need for any external interfacing of basic components like Memory, I/O Ports, etc.
- Microcontrollers doesn't require complex operating systems as all the instructions must be written and stored in the memory. (RTOS is an exception).
- All the Input/Output Ports are programmable.
- Integration of all the essential components reduces the cost, design time and area of the product (or application).

### **Disadvantages of Microcontrollers**

- Microcontrollers are not known for their computation power.
- The amount of memory limits the instructions that a microcontroller can execute.
- No Operating System and hence, all the instruction must be written.

### **Applications of Microcontrollers**

There are huge numbers of applications of Microcontrollers. In fact, the entire embedded systems industry is dependent on Microcontrollers. The following are few applications of Microcontrollers.

- Front Panel Controls in devices like Oven, washing Machine etc.
- Function Generators
- Smoke and Fire Alarms
- Home Automation Systems
- Automatic Headlamp ON in Cars
- Speed Sensed Door Locking System

### **Video Content/ Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRPtvj7bFU&noredirect=1)

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 Page No: 24-28

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L11

EEE

III/IV

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

## Topic of Lecture: Architecture of 8051 Microcontroller

### Introduction :

- A typical microcontroller consists of arithmetic and logic unit (ALU) in association with control unit to process the instruction execution.
- Almost all the microcontroller are based on the principle of store-program concept.
- Each microcontroller has a set of instructions, a list which is provided by the microcontroller manufacturer.
- A microcontroller is a programmable electronics chip that has computing and decision making capabilities similar to central processing unit of a computer.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Processors and controllers
- Central processing unit
- Memory Devices

### Detailed content of the Lecture:

#### 8051 MICROCONTROLLER ARCHITECTURE

Basic components present internally inside 8051 Microcontroller architecture are:

**CPU (Central Processing Unit):** CPU act as a mind of any processing machine. It synchronizes and manages all processes that are carried out in microcontroller. User has no power to control the functioning of CPU. It interprets the program stored in ROM and carries out from storage and then performs it projected duty. CPU manage the different types of registers available in 8051 microcontroller.

**Interrupts:** Interrupts is a sub-routine call that given by the microcontroller when some other program with high priority is request for acquiring the system buses the n interrupts occur in current running program.

Interrupts provide a method to postpone or delay the current process, performs a sub-routine task and then restart the standard program again.

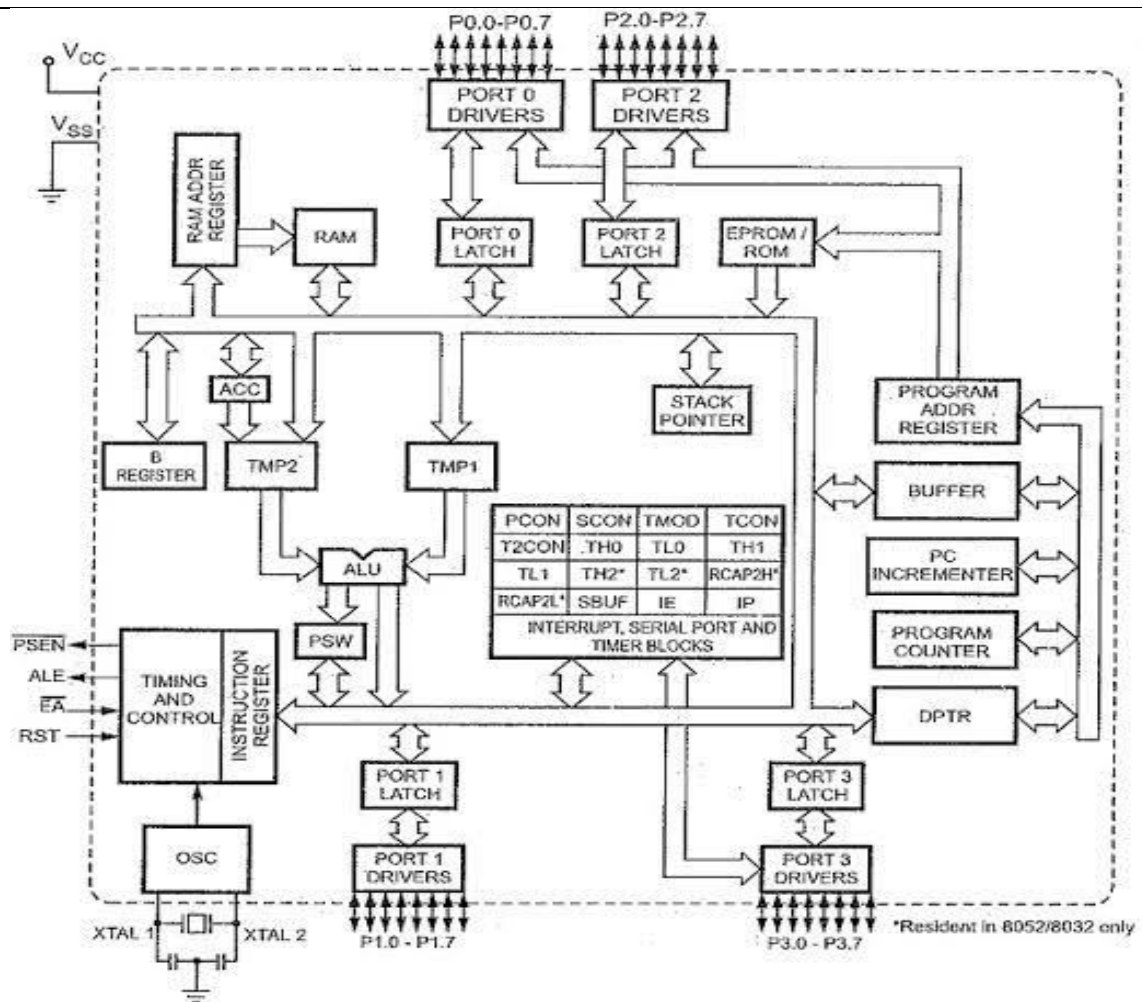


Fig. 12.2 Intel 8051/8031 architecture

### Types of interrupt in 8051 Microcontroller:

Let's see the five sources of interrupts in 8051 Microcontroller:

- Timer 0 overflow interrupt - TF0
- Timer 1 overflow interrupt - TF1
- External hardware interrupt - INT0
- External hardware interrupt - INT1
- Serial communication interrupt - RI/TI

**Memory:** For operation Micro-controller required a program. This program guides the microcontroller to perform the specific tasks. This program installed in microcontroller required some on chip memory for the storage of the program.

Microcontroller also required memory for storage of data and operands for the short duration. In microcontroller 8051 there is code or program memory of 4 KB that is it has 4 KB ROM and it also comprise of data memory (RAM) of 128 bytes.

**Bus:** Bus is a group of wires which uses as a communication canal or acts as means of data transfer. The different bus configuration includes 8, 16 or more cables. Therefore, a bus can bear 8 bits, 16 bits all together.

**Types of buses in 8051 Microcontroller:**

Let's see the two types of bus used in 8051 microcontroller:

- **Address Bus:** 8051 microcontrollers is consisting of 16 bit address bus. It is generally be used for transferring the data from Central Processing Unit to Memory.
- **Data bus:** 8051 microcontroller is consisting of 8 bits data bus. It is generally be used for transferring the data from one peripherals position to other peripherals.
- **Oscillator:** As the microcontroller is digital circuit therefore it needs timer for their operation. To perform timer operation inside microcontroller it required externally connected or on-chip oscillator. Microcontroller is used inside an embedded system for managing the function of devices. Therefore, 8051 uses the two 16 bit counters and timers. For the operation of this timers and counters the oscillator is used inside microcontroller.

**Video Content / Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRPtvj7bFU&noredirect=1)

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 28-32)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L12

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

**Topic of Lecture:** I/O Ports of 8051

**Introduction :**

8051 microcontrollers have 4 I/O ports each of 8-bit, which can be configured as input or output. Hence, total 32 input/output pins allow the microcontroller to be connected with the peripheral devices. Pin configuration, i.e. the pin can be configured as 1 for input and 0 for output as per the logic state.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

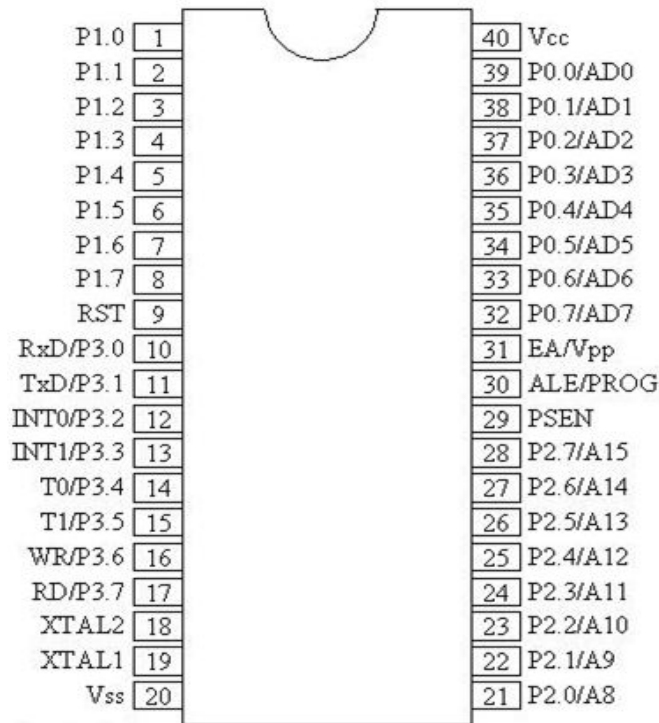
- Architecture of 8051

**Detailed content of the Lecture:**

- In 8051 Microcontroller, I/O operations are performed by using four ports and 40 pins. I/O operation port uses 32 pins with each port has 8 pins. The remaining 8-pins are used for providing
- Let's see the 40-Pin Plastic Dual Inline Package (PDIP) integrated circuit of microcontroller:
- In DIP structure of microcontroller we can recognize the first and last pin by using the cut present at middle of IC this cut is called as Notch of microcontroller. The first pin is present at left side of notch and last pin is present at right side of notch.

**I/O Ports and their Functions:**

In microcontroller there are four Input/output ports P0, P1, P2, and P3, each port is 8-bit port having 8 pins each. During RESET, all the ports are used as input ports. When the port gets first 0, then it becomes an output port. For reconfigure it as an input, the high signal (1) must be sent to a port.



### Port 0 (Pin No 32 - Pin No 39):

Port 0 contains 8 pins. It can be used as input or output. In general we connect P0 with 10K-ohm pull-up resistors for using it as an input or output port being an open drain.

It is also referred as AD0-AD7, which allowing it to be used as both address and data transfer port. When we want to access the external ROM, then P0 is used as both Data and Address Bus.

Address Latch Enable (ALE) Pin no 31 indicates if PO is used as address or data.

When ALE=0 then it provides data D0-D7.

When ALE=1 then it provides address A0-A7.

Let's see the structure of port 0 with externally connected pull up resistor:

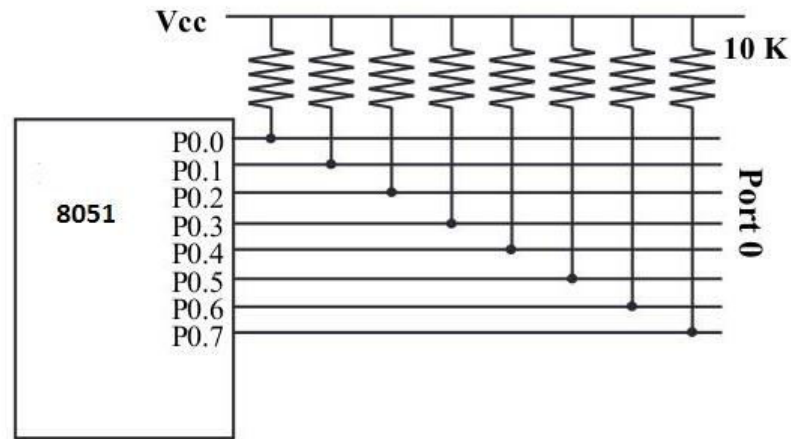
Let's see an assembly language code for making the Port 0 to be worked as an input:-

```
MOV A, #FFH; (comments: A=FFH i.e. A=1111 1111)
MOV P0, A; (Port 0 all pin have 1's so that it work as Input)
```

### Port 1 (Pin No 1-Pin No 8):

It is also an 8-bit Port and can be worked as either input or output. It doesn't require external connected pull-up resistors because they are already present internally. Upon reset, Port 1 worked as an input port.





If port 1 is configured as an output port, then to use port 1 as input port again, we write 1 to all bits of port 1 as shown in the below code:-

```
MOV A, #FFH    ; Value of Accumulator register A=FF hex
MOV P1, A     ; Port 1 worked as an input port
MOV A, P1    ; get data from P1
MOV R5, A    ; save it in Register R5
ACALL DELAY  ; wait
MOV A, P1    ; get another data from P1
MOV R6, A    ; save it in R6
ACALL DELAY  ; wait
MOV A, P1    ; get data from P1
MOV R7, A    ; save it in R7
```

### Port 2 (Pin No 21-Pin No 28):

Port 2 uses a total of 8 pins and it can also be used as input and output operation. Same as Port 1, P2 also not require external pull- up resistors. Port 2 can be used along with P0 to provide 16-bit address for an external memory. Therefore it is designated as (A0-A7), as shown in pin diagram.

If Port 2 is configured as an output port, then for using it as an input port again we write 1 to all bits of port 2 as shown in the below code:-

```
MOV A, #FFH    ; Value of Accumulator register A=FF hex
MOV P2, A     ; Port 2 worked as an input port
BACK:
MOV A, P1    ; get data from P2
MOV P1, A    ; save it in Port 1
SJMP BACK   ; Keep doing that
```

### Port 3 (Pin No 10-Pin No 17):

Port 3 is also of 8 bits and it can be used as Input/output. This port provides some important signals.

P3.1 and P3.0 are RxD (Receiver) and TxD (Transmitter) respectively and it is collectively used for serial communication.

P3.3 and P3.2 pins are used as external interrupts.

P3.5 and P3.4 are used as timers T1 and T0 respectively.

P3.6 and P3.7 are Write (WR) and Read (RD) pins.

Let's see the Port 3 table showing an individual pin function:

Pin	Function	P3 Bit
10	Receiver (RxD)	P3.0
11	Transmitter (TxD)	P3.1
12	Complement of INT 0	P3.2
13	INT 1	P3.3
14	Timer 0(TO)	P3.4
15	Timer 1(T1)	P3.5
16	Write (WR)	P3.6
17	Complement of read (RD)	P3.7

#### I/O Ports and Bit Addressability:

It is a mostly used feature of 8051 while writing code for 8051. Sometimes there is a need to use only 1 or 2 bits of the port instead of using entire 8-bits. 8051 microcontroller provides the feature to use each bit of the ports.

While using a port in a single-bit manner, we provide the syntax "SETB X.Y", where X is the port number varies from 0 to 3, and Y is a bit number varies from 0 to 7.

For example: - "SETB P1.3" sets high bit 3 of port 1.

#### Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=WU8uvapSlic>

#### Important Books/Journals for further learning including the page nos.:

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 Page No: 93-106



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L13

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

**Topic of Lecture:** Pin Diagram of 8051

**Introduction :**

8051 microcontroller families (89C51, 8751, DS89C4xO, 89C52) come in different packages like quad-flat package, leadless chip carrier and dual-in-line package. These all packages consist of 40 pins which are dedicated to several functions such as I/O, address, RD, WR, data and interrupts. But, some companies offer a 20-pin version of the microcontrollers for less demanding applications by reducing the number of I/O ports. Nevertheless, a vast majority of developers use the 40-pin chip.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Architecture of 8051

**Detailed content of the Lecture:**

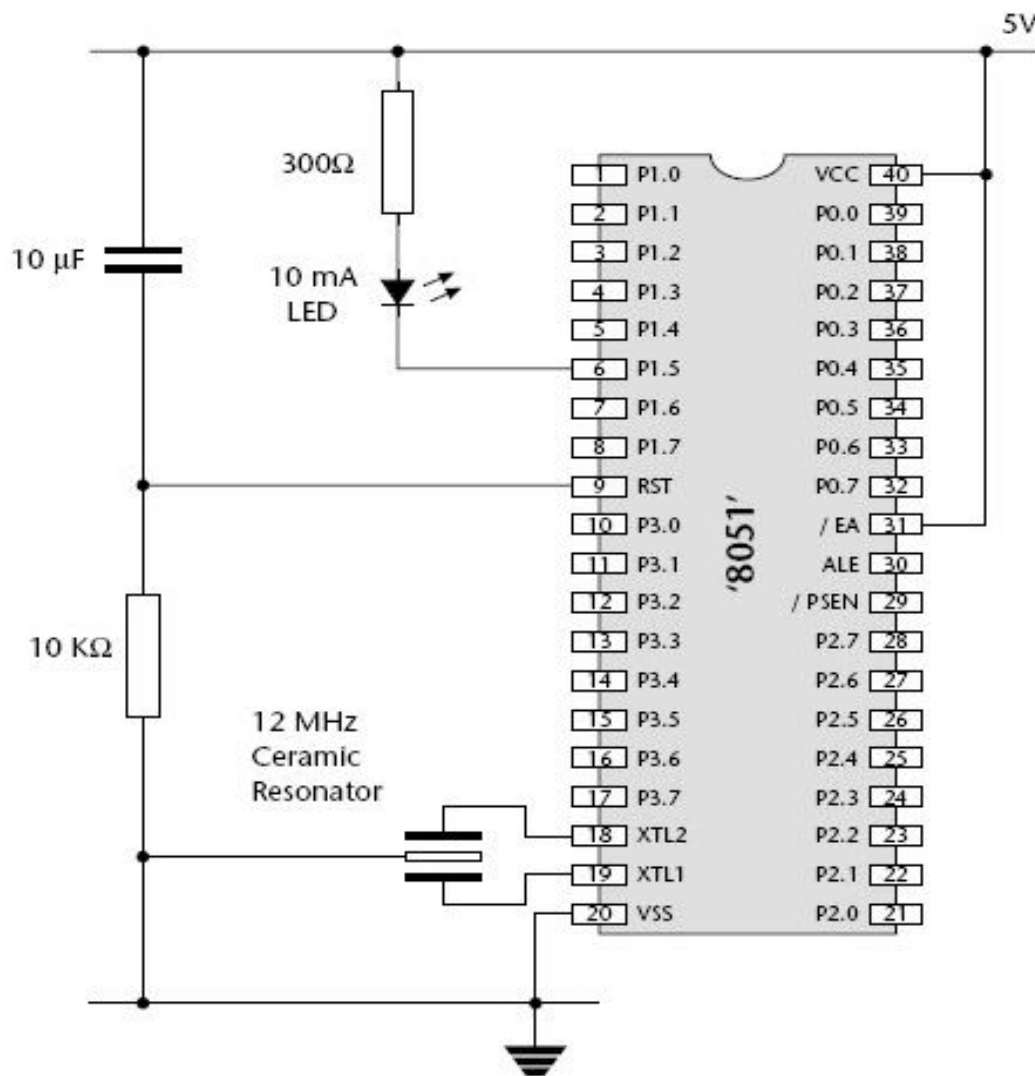
**8051 Microcontroller Pin Diagram**

Let's see the 40 pin Dual Inline Package (DIP) of microcontroller integrated circuit with externally connected power supply and oscillator.

Consider the brief description of pin configuration inside microcontroller is:

**Pins 1-8:** These pins belongs to Port 1 of microcontroller. Port 1 is used as domestically pulled up, quasi bi directional input/output port.

**Pin 9:** It is a RESET pin which is utilized to set the microcontroller 8051 to its primary value. During the beginning of an application the RESET pin is to be set elevated for two machine rotations.



**Pins 10-17:** These pins belong to Port 3 of microcontroller. Port 3 can be used for number of functions such as timer input, interrupts, serial communication indicator for transmitting (TxD) and receiving (RxD). It is also known as domestic pull up port with quasi bi direction port embedded within.

**Pins 18 and 19:** These pins are generally be used for interfacing outer crystal oscillator with given system clock.

**Pin 20:** This pin titled as  $V_{ss}$ . It symbolizes ground voltage or 0 V is connected to this pin of microcontroller.

**Pin 21-28:** These pins belong to port 2 of microcontroller. Port 2 can be used as Input/output port, senior order address bus are multiplexed with this quasi bi directional port.

**Pin 29:** This pin belongs to Program Store Enable or PSEN. It is used for interpreting the sign from outer program memory.

**Pin 30:** This pin belongs to External Access or EA input is used for permit or prohibits outer memory interfacing. If there is no outer memory need, this pin is set to high by linking it with supply voltage .

**Pin 31:** This pin belongs to Address Latch Enable or ALE is used for de-multiplexing the address data indication of port 0 for outer memory interfacing.

**Pin 32-39:** These pins belong to Port 0 of the microcontroller. Port 0 can be used as input/output port, lower order address and data bus signals are multiplexed with this port. This pin act as bi directional Input/output port and outer connected pull up resistors are necessary for utilizing these ports as Input/output.

**Pin 40:** This pin is used to provide power supply to the circuit.

**Video Content/ Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=ii7PCV2zvms>

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 218-223)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L14

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

**Topic of Lecture:** Instruction set

### Introduction :

An 8051 Instruction consists of an Opcode (short of Operation - Code) followed by Operand(s) of size Zero Byte, One Byte or Two Bytes. The Op-Code part of the instruction contains the Mnemonic, which specifies the type of operation to be performed.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Architecture
- Pin diagram

### Detailed content of the Lecture:

- An instruction is a command to the microprocessor to perform a given task on a specified data. Each instruction has two parts: one is task to be performed, called the operation code (opcode), and the second is the data to be operated on, called the operand. The operand (or data) can be specified in various ways. It may include 8-bit (or 16-bit ) data, an internal register, a memory location, or 8-bit (or 16-bit) address.

In some instructions, the operand is implicit.

### Instruction word size

The 8051 instruction set is classified into the following three groups according to word size:

1. One-word or 1-byte instructions
2. Two-word or 2-byte instructions
3. Three-word or 3-byte instructions

### One-Byte Instructions

A 1-byte instruction includes the opcode and operand in the same byte.

Operand(s) are internal register and are coded into the instruction

Task	Op code	Operand	Binary Code	Hex Code
Copy the contents of the accumulator in the register C.	MOV	C,A	0100 1111	4FH
Add the contents of register B to the contents of the accumulator.	ADD	B	1000 0000	80H
Invert (compliment) each bit in the accumulator.	CMA		0010 1111	2FH

- These instructions are 1-byte instructions performing three different tasks. In the first instruction, both operand registers are specified. In the second instruction, the operand B is specified and the accumulator is assumed. Similarly, in the third instruction, the accumulator is assumed to be the implicit operand. These instructions are stored in 8-bit binary format in memory; each requires one memory location

### MOV rd, rs

rd ← rs copies contents of rs into rd.

- The destination of the data, rdd is the code of the source register. Example: MOV A,B
- Coded as 01111000 = 78H = 170 octal (octal was used extensively in instruction design of such processors).
- **ADD r**
- $A \leftarrow A + r$

### Two-Byte Instructions

- In a two-byte instruction, the first byte specifies the operation code and the second byte specifies the operand. Source operand is a data byte immediately following the opcode. For example

Task	Opcode	Operand	Binary Code	Hex Code	
Load an 8-bit data byte in the accumulator.	MVI	A, Data	0011 1110	3E	First Byte
			DATA	Data	Second Byte

The instruction would require two memory locations to store in memory.

### MVI r,data

$r \leftarrow \text{data}$

Example: MVI A,30H coded as 3EH 30H as two contiguous bytes. This is an example of immediate addressing.

### ADI data

$A \leftarrow A + \text{data}$

OUT port

0011 1110

DATA

where port is an 8-bit device address. (Port) ← A. Since the byte is not the data but points directly to where it is located this is called direct addressing.

### Three-Byte Instructions

- In a three-byte instruction, the first byte specifies the opcode, and the following two bytes specify the 16-bit address. Note that the second byte is the low-order address
- and the third byte is the high-order address.
- opcode + data byte + data byte

Task	Opcode	Operand	Binary code	Hex Code	
Transfer the program sequence to the memory location 2085H.	JMP	2085H	1100 0011	C3	First byte
			1000 0101	85	Second Byte
			0010 0000	20	Third Byte

This instruction would require three memory locations to store in memory. Three byte instructions - opcode + data byte + data byte

#### LXI rp, data16

rp is one of the pairs of registers BC, DE, HL used as 16-bit registers. The two data bytes are 16-bit data in L H order of significance.

rp <-- data16

LXI H,0520H coded as 21H 20H 50H in three bytes. This is also immediate addressing.

#### LDA addr

A <-- (addr) Addr is a 16-bit address in L H order. Example: LDA 2134H coded as 3AH 34H 21H. This is also an example of direct addressing.

#### Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=0uKdINT0EMU>

#### Important Books/Journals for further learning including the page nos.:

- Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 Page No: (70-75)

Course Faculty





# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L 15

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

**Topic of Lecture:** Instruction Set

### Introduction :

An 8051 Instruction consists of an Opcode (short of Operation - Code) followed by Operand(s) of size Zero Byte, One Byte or Two Bytes. The Op-Code part of the instruction contains the Mnemonic, which specifies the type of operation to be performed.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Architecture
- Pin diagram

### Detailed content of the Lecture:

- An instruction is a command to the microprocessor to perform a given task on a specified data. Each instruction has two parts: one is task to be performed, called the
- Operation code (opcode), and the second is the data to be operated on, called the operand. The operand (or data) can be specified in various ways. It may include 8-bit (or 16-bit ) data, an internal register, a memory location, or 8-bit (or 16-bit) address.
- In some instructions, the operand is implicit. 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.

### DATA TRANSFER CROUP

The data transfer instructions move data between registers or between memory and registers.

- MOV Move
- MVI Move Immediate
- LDA Load Accumulator Directly from Memory
- STA Store Accumulator Directly in Memory
- LHLD Load H & L Registers Directly from Memory

- SHLD Store H & L Registers Directly in Memory
- An 'X' in the name of a data transfer instruction implies that it deals with a register pair (16-bits);
- LXI Load Register Pair with Immediate data
- LDAX Load Accumulator from Address in Register Pair STAX Store Accumulator in Address in Register Pair XCHG Exchange H & L with D & E
- XTHL Exchange Top of Stack with H & L

## ARITHMETIC GROUP

The arithmetic instructions add, subtract, increment, or decrement data in registers or memory.

- ADD Add to Accumulator
- ADI Add Immediate Data to Accumulator
- ADC Add to Accumulator Using Carry Flag
- ACI Add Immediate data to Accumulator Using Carry
- SUB Subtract from Accumulator
- SUI Subtract Immediate Data from Accumulator
- SBB Subtract from Accumulator Using Borrow (Carry) Flag
- SBI Subtract Immediate from Accumulator Using Borrow (Carry) Flag
- INR Increment Specified Byte by One

DCR Decrement Specified Byte by One

INX Increment Register Pair by One

DCX Decrement Register Pair by One

- DAD Double Register Add; Add Content of Register Pair to H & L Register Pair

## LOGICAL GROUP

- This group performs logical (Boolean) operations on data in registers and memory condition flags.
- The logical AND, OR, and Exclusive OR instructions enable you to set specific bits in the
  - ANA Logical AND with Accumulator
  - ANI Logical AND with Accumulator Using Immediate Data
  - ORA Logical OR with Accumulator
  - OR Logical OR with Accumulator Using Immediate Data
  - XRA Exclusive Logical OR with Accumulator
  - XRI Exclusive OR Using Immediate Data
- The Compare instructions compare the content of an 8-bit value with the contents of the accumulator;
  - CMP Compare
  - CPI Compare Using Immediate Data
- The rotate instructions shift the contents of the accumulator one bit position to the left or right:
  - RLC Rotate Accumulator Left RRC Rotate Accumulator Right RAL Rotate Left Through Carry RAR Rotate Right Through Carry

- Complement and carry flag instructions: CMA Complement Accumulator
- CMC Complement Carry Flag
- STC Set Carry Flag

## **BRANCH GROUP**

- The branching instructions alter normal sequential program flow, either unconditionally or
- conditionally. The unconditional branching instructions are as follows
- JMP Jump CALL Call RET Return
- Conditional branching instructions examine the status of one of four condition flags to determine
- Whether the specified branch is to be executed. The conditions that may be specified are as

follows:

- NZ Not Zero (Z = 0) Z Zero (Z = 1)
- NC No Carry (C = 0) C Carry (C = 1)
- PO Parity Odd (P = 0) PE Parity Even (P = 1) P Plus (S = 0)
- M Minus (S = 1)
- Thus, the conditional branching instructions are specified as follows: Jumps Calls Returns
- C CC RC (Carry)
- INC CNC RNC (No Carry) JZ CZ RZ (Zero)
- JNZ CNZ RNZ (Not Zero) JP CP RP (Plus)
- JM CM RM (Minus)
- JPE CPE RPE (Parity Even) JP0 CPO RPO (Parity Odd)
- Two other instructions can affect a branch by replacing the contents or the program counter:
- PCHL Move H & L to Program Counter RST Special Restart Instruction Used with Interrupts
- POP Pop Two Bytes of Data off the Stack XTHL Exchange Top of Stack with H & L SPHL Move content of H & L to Stack Pointer

## **I/O INSTRUCTIONS**

- IN Initiate Input Operation
- OUT Initiate Output Operation

## **MACHINE CONTROL INSTRUCTIONS**

- EI Enable Interrupt System
- DI Disable Interrupt System
- HLT Halt
- NOP No Operation

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=0uKdINT0EMU>

**Important Books/Journals for further learning including the page nos.:**

- Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 Page No: 70-75

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L16

## LECTURE HANDOUTS

EEE

III/IV

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

**Topic of Lecture:** 8051 ADDRESSING MODES

### Introduction :

There are 5 different ways to execute this instruction and hence we say, we have got 5 addressing modes for 8051. They are 1) Immediate addressing mode 2) Direct addressing mode 3) Register direct addressing mode 4) Register indirect addressing mode 5) Indexed addressing mode.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Instruction Set

### Detailed content of the Lecture:

- The instructions MOV B, A or MVI A, 82H are to copy data from a source into a destination. In these instructions the source can be a register, an input port, or an 8-bit number (00H to FFH).
- Similarly, a destination can be a register or an output port. The sources and destination are operands.
- The various formats for specifying operands are called the ADDRESSING MODES. For 8085, they are:
  1. Immediate addressing.
  2. Register addressing.
  3. Direct addressing.
  4. Indirect addressing.
  5. Indexed addressing mode.

### (1) Immediate addressing

Data is present in the instruction. Load the immediate data to the destination provided.

**Example:** MVI R,data

## **(2) Register addressing**

Data is provided through the registers.

**Example:** MOV Rd, Rs

## **(3) Direct addressing**

- Used to accept data from outside devices to store in the accumulator or send the data stored in the accumulator to the outside device. Accept the data from the port 00H and store them into the accumulator or Send the data from the accumulator to the port 01H.

**Example:** IN 00H or OUT 01H

## **(4) Indirect Addressing**

- This means that the Effective Address is calculated by the processor.
- And the contents of the address (and the one following) is used to form a second address.
- The second address is where the data is stored. Note that this requires several memory accesses; two accesses to retrieve the 16-bit address and a further access (or accesses) to retrieve the data which is to be loaded into the register.

## **(5) Indexed Addressing Mode**

- Indexed addressing mode is widely used in accessing data elements of look-up table entries located in the program ROM space of the 8051.
- The instruction used for this purpose is :  
MOVC A, @ A+DPTR
- The 16-bit register DPTR and register A are used to form the address of the data element stored in on-chip ROM.
- Because the data elements are stored in the program (code) space ROM of the 8051, the instruction MOVC is used instead of MOV. The "C" means code.
- In this instruction the contents of A are added to the 16-bit register DPTR to form the 16-bit address of the needed data

### **Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=sLbw1stNkXM>

### **Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 Page No: 109-131



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L17

## LECTURE HANDOUTS

EEE

III/IV

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

**Topic of Lecture:** Timing Diagram

### Introduction :

The timing diagram depicts the machine cycles generated for each instruction.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Instruction set
- Addressing Modes



### Detailed content of the Lecture:

- Timing Diagram is a graphical representation. It represents the execution time taken by each instruction in a graphical format. The execution time is represented in T-states.

### Instruction Cycle

- The time required to execute an instruction is called instruction cycle.

### Machine Cycle

- The time required to access the memory or input/output devices is called machine cycle.

### T-State

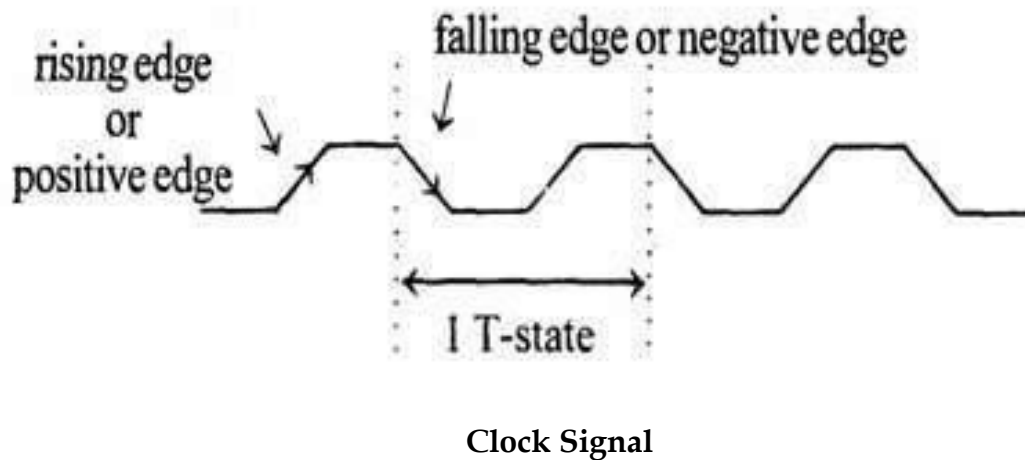
- The machine cycle and instruction cycle takes multiple clock periods.
- A portion of an operation carried out in one system clock period is called as T-state

The 8085 microprocessor has 5 (seven) basic machine cycles. They are

- Opcode fetch cycle (4T)
- Memory read cycle (3 T)
- Memory write cycle (3 T)

- I/O read cycle (3 T)
- I/O write cycle (3 T)

*Time period,  $T = 1/f$ ; where  $f =$  Internal clock frequency*



**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=2thCkcQj58Y>

**Important Books/Journals for further learning including the page nos.:**

- Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 Page No: (239-240)

**Course Faculty**





# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L18

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : II - 8051 CONTROLLER

Date of Lecture:

**Topic of Lecture:** Timing Diagram

**Introduction :**

The timing diagram depicts the machine cycles generated for each instruction.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

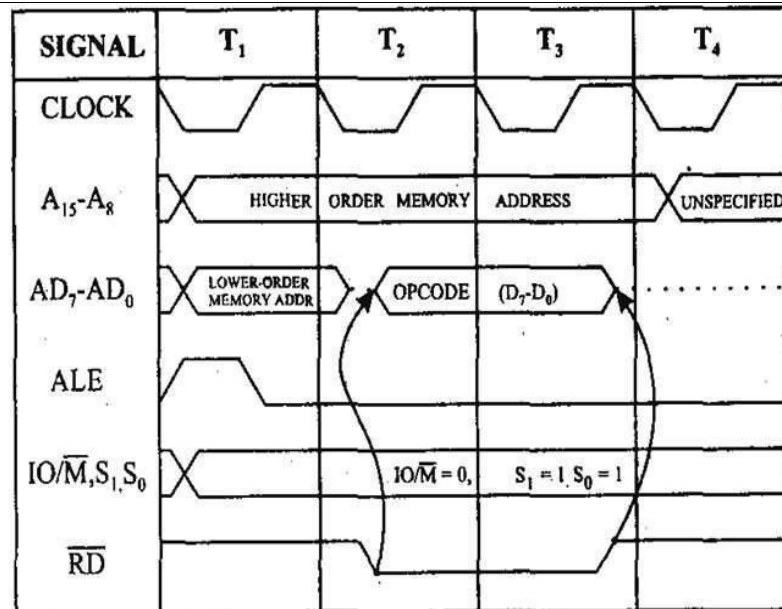
- Architecture of 8051
- Instruction set
- Addressing modes

**Detailed content of the Lecture:**

**1.Opcode fetch machine cycle of 8051 :**

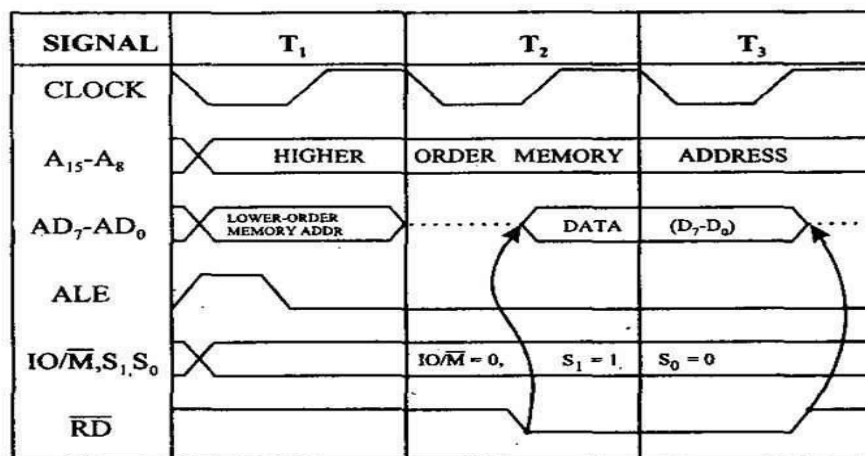
Each instruction of the processor has one byte opcode.

- The opcodes are stored in memory. So, the processor executes the opcode fetch machine cycle to fetch the opcode from memory.
- Hence, every instruction starts with opcode fetch machine cycle.
- 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.



### 2. Memory Read Machine Cycle of 8051:

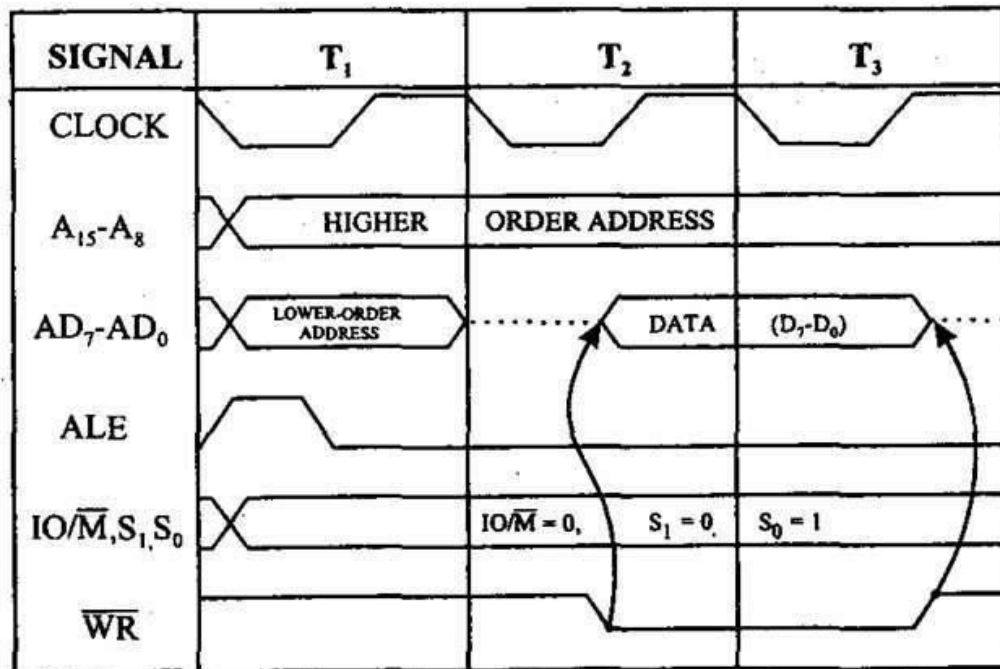
- The memory read machine cycle is executed by the processor to read a data byte from memory.
- The processor takes 3T states to execute this cycle.
- The instructions which have more than one byte word size will use the machine cycle after the opcode fetch machine cycle.



### 3. Memory Write Machine Cycle of 8051

The memory write machine cycle is executed by the processor to write a data byte in a memory location.

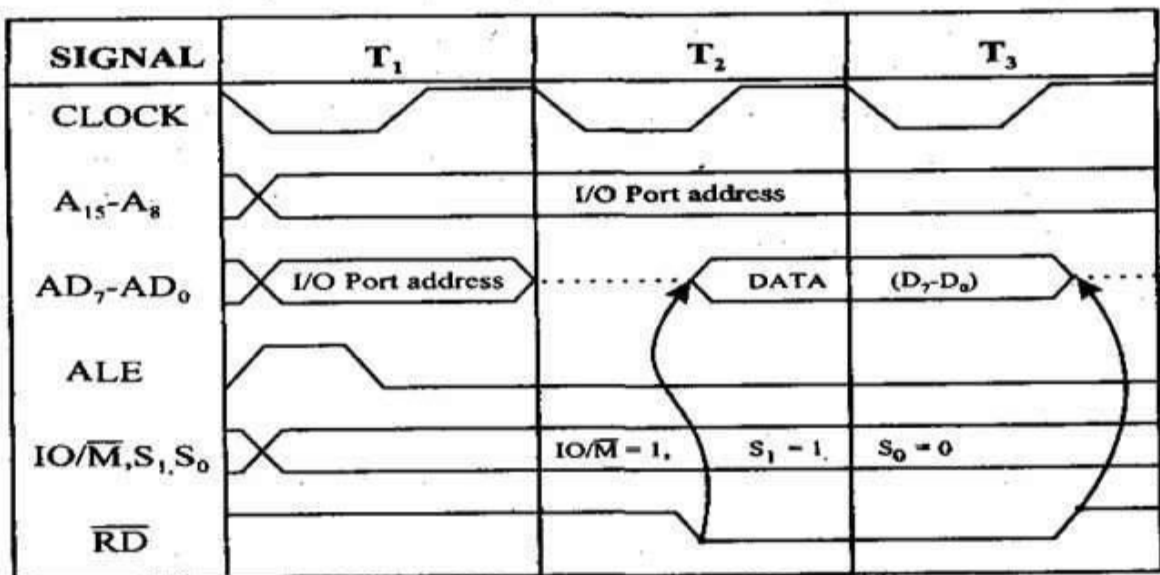
The processor takes 3T states to execute this machine cycle.



Memory Write Machine Cycle

#### 4. I/O Read Cycle of 8051

- The I/O Read cycle is executed by the processor to read a data byte from I/O port or from the peripheral, which is I/O, mapped in the system.
- The processor takes 3T states to execute this machine cycle.
- The IN instruction uses this machine cycle during the execution.



□

### **Timing diagram for INR M**

Fetching the Opcode 34H from the memory 4105H. (OF cycle)

Let the memory address (M) be 4250H. (MR cycle -To read Memory address and data) Let the content of that memory is 12H.

Increment the memory content from 12H to 13H. (MW machine cycle)

### **Video Content/ Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=2thCkcQj58Y>

### **Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 Page No: (255-260)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L19

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code: 16EEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS

Date of Lecture:

**Topic of Lecture:** Basic programming of 8085

### Introduction :

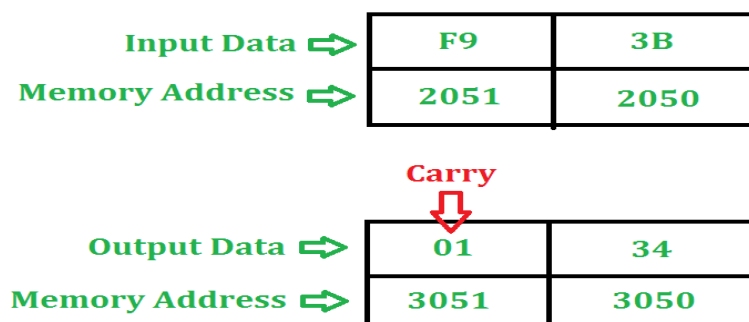
- Microprocessor is a multipurpose, programmable, clock-driven, register-based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions and provide results as output.
- The memory addresses given in the program are for a particular microprocessor kit

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic concepts of 8085

### Detailed content of the Lecture:

- Load the first number from memory location 2050 to accumulator.
- Move the content of accumulator to register H.
- Load the second number from memory location 2051 to accumulator.
- Then add the content of register H and accumulator using "ADD" instruction and storing result at 3050
- The carry generated is recovered using "ADC" command and is stored at memory location 3051



## Program

MEMORY ADDRESS	MNEMONICS	COMMENT
2000	LDA 2050	A←-[2050]
2003	MOV H, A	H←A
2004	LDA 2051	A←-[2051]
2007	ADD H	A←-A+H
2006	MOV L, A	L←A
2007	MVI A 00	A←00
2009	ADC A	A←A+A+carry
200A	MOV H, A	H←A
200B	SHLD 3050	H→3051, L→3050
200E	HLT	

- **LDA 2050** moves the contents of 2050 memory location to the accumulator.
- **MOV H, A** copies contents of Accumulator to register H to A
- **LDA 2051** moves the contents of 2051 memory location to the accumulator.
- **ADD H** adds contents of A (Accumulator) and H register (F9). The result is stored in A itself. **For all arithmetic instructions A is by default an operand and A stores the result as well**
- **MOV L, A** copies contents of A (34) to L
- **MVI A 00** moves immediate data (i.e., 00) to A
- **ADC A** adds contents of A(00), contents of register specified (i.e A) and carry (1). As ADC is also an arithmetic operation, A is by default an operand and A stores the result as well
- **MOV H, A** copies contents of A (01) to H
- **SHLD 3050** moves the contents of L register (34) in 3050 memory location and contents of H register (01) in 3051 memory location
- **HLT** stops executing the program and halts any further execution

**Video Content / Details of website for further learning (if any):**

[https://www.geeksforgeeks.org > microprocessor-tutorials](https://www.geeksforgeeks.org/microprocessor-tutorials)

[https://www.geeksforgeeks.org > assembly-language-program-8085-micro.](https://www.geeksforgeeks.org/assembly-language-program-8085-micro)

**Important Books/Journals for further learning including the page nos.:**

R.S. Gaonkar, 'Microprocessor Architecture Programming and Application', with 8085, Wiley Eastern Ltd., New Delhi, 2013 (Page No: 135-154)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L20

## LECTURE HANDOUTS

EEE

III/V

**Course Name with Code: 16EEC09/MICROPROCESSORS AND MICROCONTROLLERS**

**Course Faculty : Ms.R.PUNITHA**

**Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS**

**Date of Lecture:**

**Topic of Lecture:** Basic programming (ALP) of 8051

### Introduction :

1. Microprocessors are typically programmed using semi-English-language statements (assembly language).
2. In addition to assembly languages, microcomputers use a more understandable human oriented language called high-level language.

### Prerequisite knowledge for Complete understanding and learning of Topic:

3. Instruction set of 8051
4. Addressing modes of 8051
5. Memory organization of 8051

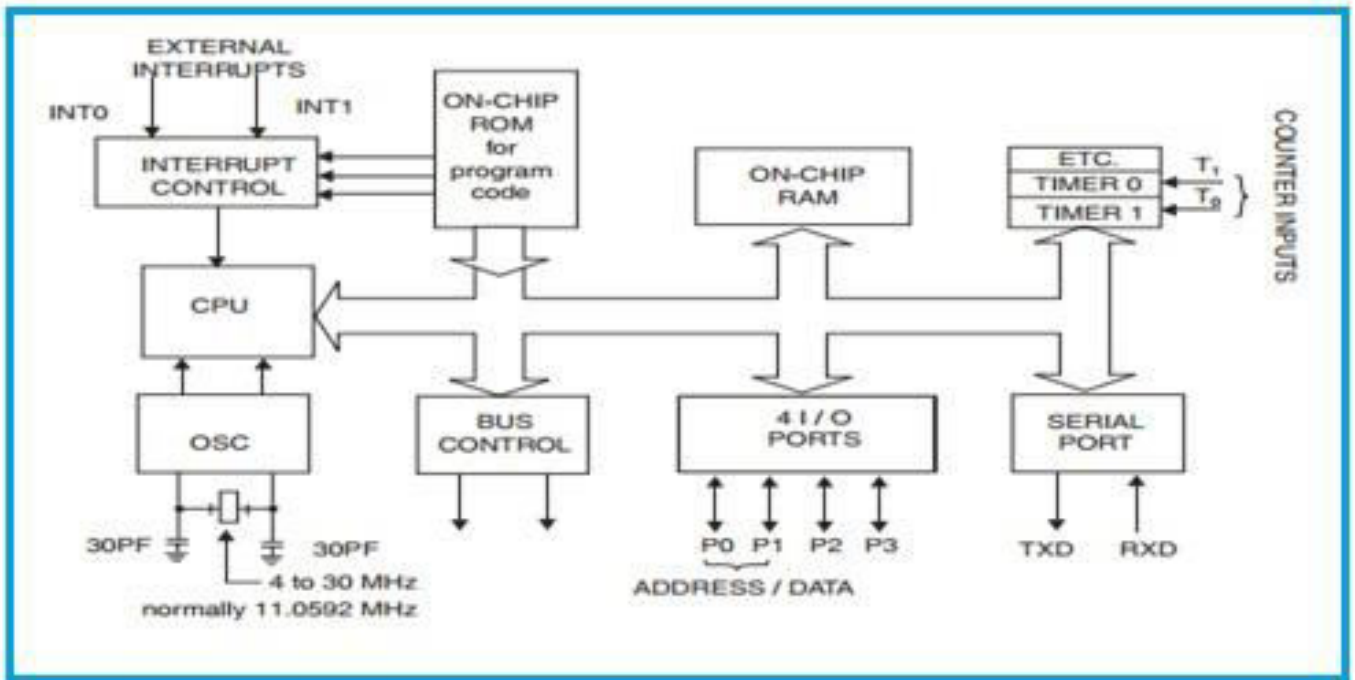
### Detailed content of the Lecture:

8051 is the microcontroller developed by Intel. Following are the features of 8051 Microcontroller:

- It houses 8 bit CPU.
- 4 KByte ROM used for code or program storage
- 128 Byte RAM to store data or variables used in program
- 32 Input/Output lines with 4 ports ( 8 lines per port)
- 2 Timers used for introducing delay and for setting the baud rate for data communication.
- 1 serial Port with one TxD and one RxD line used for serial communication with external devices.
- 6 Interrupt Sources
- Clock oscillator circuit runs at 12MHz frequency.

The figure-1 depicts 8051 microcontroller architecture. Refer 8051 Architecture➤





Following table summarizes similarities and difference between 8051 and 8085.

Specifications	8051	8085
Device Type	It is a microcontroller.	It is a microprocessor.
Timers/Counters	YES	No, Need to be connected externally.
Data bus width	8 lines	8 lines
Address Bus lines	16	16
DMA Access signals	NO	YES, Has HOLD and HLDA signals
Internal RAM, ROM	YES	No, need to be connected externally with 8085 chip, if data and code size requirement is more.
Programming	8051 programming	8085 programming

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=tC4WvbM3hZA>

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 Page no( 37-55)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



## LECTURE HANDOUTS

L21

EEE

III/V

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS

Date of Lecture:

**Topic of Lecture:** Loop Structures

### Introduction :

- The programming technique used to instruct the microprocessor to repeat tasks is called looping.
- A loop is set up by instructing the microprocessor to change the sequence of execution and perform the task again.
- A program with a continuous loop does not stop repeating the task until the system is reset.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic Programming of 8085
- Basic Programming of 8051

### Detailed content of the Lecture:

#### LOOPING STRUCTURES

- Loops are control structures used to repeat a given section of code a certain number of times or until a particular condition is met.
- Visual Basic has three main types of loops: for next loops, do loops and while loops.
- The availability of low cost, low power and small weight, computing capability makes it useful in different applications.
- Now a days, a microprocessor based systems are used in instructions, automatic testing product, speed control of motors, traffic light control, light control of furnaces etc.
- In the sequence of instructions to be executed, it is often necessary to transfer program control to a different location. There are many instructions in the 8051 to achieve this.
- This chapter covers the control transfer instructions available in 8051 Assembly language.

- In the first section we discuss instructions used for looping, as well as instructions for conditional and unconditional jumps.
- In the second section we examine CALL instructions and their uses.
- In the third section, time delay subroutines are described for both the traditional 8051 and its newer generation.

## LOOP INSTRUCTIONS

- Repeating a sequence of instructions a certain number of times is called a *loop*.
- The loop is one of most widely used actions that any microprocessor performs.
- In the 8051, the loop action is performed by the instruction “DJNZ reg, label”.
- In this instruction, the register is decremented; if it is not zero, it jumps to the target address referred to by the label.
- Prior to the start of the loop the register is loaded with the counter for the number of repetitions.
- Notice that in this instruction both the register decrement and the decision to jump are combined into a single instruction.

### Example -1

Write a program to  
 (a) clear ACC, then  
 (b) add 3 to the accumulator ten times.

**Solution:**

```

;This program adds value 3 to the ACC ten times

      MOV  A,#0      ;A=0, clear ACC
      MOV  R2,#10    ;load counter R2=10
AGAIN:  ADD  A,#03    ;add 03 to ACC
      DJNZ R2,AGAIN  ;repeat until R2=0(10 times)
      MOV  R5,A      ;save A in R5
  
```

In the program in Example 1, the R2 register is used as a counter. The counter is first set to 10. In each iteration the instruction DJNZ decrements R2 and checks its value. If R2 is not zero, it jumps to the target address associated with the label “AGAIN”. This looping action continues until R2 becomes zero. After R2 becomes zero, it falls through the loop and executes the instruction immediately below it, in this case the “MOV R5 , A” instruction.

Notice in the DJNZ instruction that the registers can be any of R0 – R7. The counter can also be a RAM location.

### Example- 2

What is the maximum number of times that the loop in Example-1 can be repeated?

#### Solution:

Since R2 holds the count and R2 is an 8-bit register, it can hold a maximum of FFH (255 decimal); therefore, the loop can be repeated a maximum of 256 times.

### Loop inside a loop

As shown in Example-2, the maximum count is 256. What happens if we want to repeat an action more times than 256? To do that, we use a loop inside a loop, which is called a *nested loop*. In a nested loop, we use two registers to hold the count. See Example 3.

### Example- 3

Write a program to (a) load the accumulator with the value 55H, and (b) complement the ACC 700 times.

#### Solution:

Since 700 is larger than 255 (the maximum capacity of any register), we use two registers to hold the count. The following code shows how to use R2 and R3 for the count.

```
        MOV  A, #55H
        MOV  R3, #10
NEXT:   MOV  R2, #70
AGAIN:  CPL  A
        DJNZ R2, AGAIN
        DJNZ R3, NEXT

;A=55H
;R3=10, the outer loop count
;R2=70, the inner loop count
;complement A register
;repeat it 70 times (inner loop)
```

### Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=mz4FJE8P56g>

### Important Books/Journals for further learning including the page nos.:

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (206-239)



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L22

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS

Date of Lecture:

**Topic of Lecture:** Counting and Indexing

### Introduction :

- **Counting** : This technique allows programmer to **count** how many times the instruction/set of instructions are executed.
- **Indexing** : This technique allows programmer to point or refer the data stored in sequential memory locations one by one.

### Prerequisite knowledge for Complete understanding and learning of Topic:

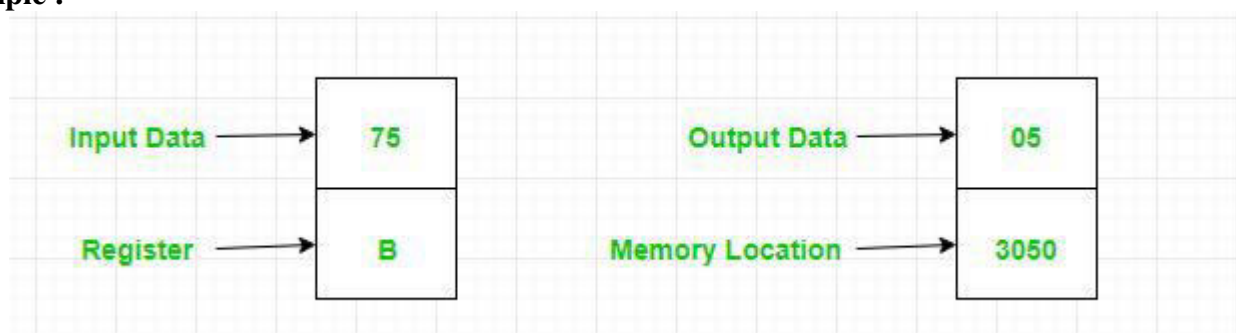
- Basic Programming of 8085
- Basic Programming of 8051

### Detailed content of the Lecture:

[8085 program to count the number of ones in contents of register B](#)

**Problem** – Write an assembly language program to count the number of ones in contents of register B and store the result at memory location 3050.

### Example :



**Algorithm :**

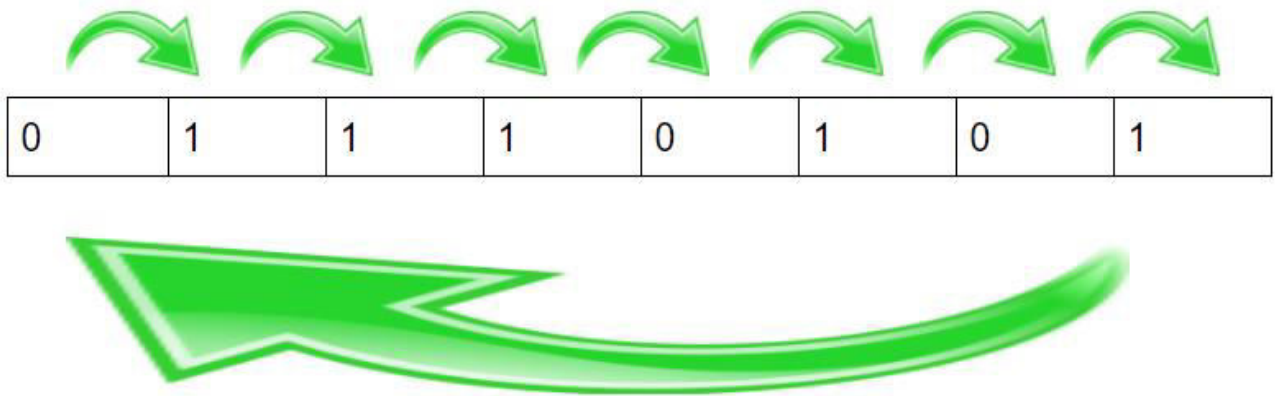
- Convert the decimal number in Accumulator to its binary equivalent
- Rotate the digits of the binary number right without carry
- Apply a loop till count is not zero to change the values of D register and count
- Copy the value of D register to accumulator and store the result

**Program :**

MEMORY ADDRESS	MNEMONICS	COMMENTS
2000	MVI B 75	$B \leftarrow 75$
2002	MVI C 08	$C \leftarrow 75$
2004	MVI D 00	$D \leftarrow 00$
2006	MOV A, B	$A \leftarrow B$
2007	RRC	Rotate right without carry
2008	JNC 200C	Jump if Not Carry
200B	INR D	$D \leftarrow D+1$
200C	DCR C	$C \leftarrow C-1$
200D	JNZ 2007	Jump if Not Zero
2010	MOV A, D	$A \leftarrow D$
2011	STA 3050	$A \rightarrow 3050$
2014	HLT	Stops execution

**Explanation –**

1. **MVI B 75** moves 75 decimal number into B register
2. **MVI C 08** moves 08 decimal number into C register, which is taken as counter as the number is of 8 bites
3. **MVI D 00** moves 00 number into d register
4. **MOV A, B** moves the contents of B register into A (accumulator) register
5. **RRC** rotates the contents of A (which is 75 with binary equivalent 01110101) right



6. **JNC 200C** jumps to 200C address and perform the instructions written there if the carry flag is not zero
7. **INR D** increases the value of D register by adding one to its contents
8. **DCR C** decreases the value of C register by subtracting one from its contents
9. **JNZ 2007** jumps to 2007 address and perform the instructions written there if the zero flag is not zero
10. **MOV A, D** moves the contents of B register into A register
11. **STA 3050** store the contents of A at 3050 memory location
12. **HLT** stops execution

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=vi4yZOWgDc8>

**Important Books/Journals for further learning including the page nos.:**

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (252-260)

**Course Faculty**





# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L23

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS

Date of Lecture:

**Topic of Lecture:** Counting & Indexing

### Introduction :

- **Counting** : This technique allows programmer to **count** how many times the instruction/set of instructions are executed.
- **Indexing** : This technique allows programmer to point or refer the data stored in sequential memory locations one by one.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic Programming of 8085
- Basic Programming of 8051

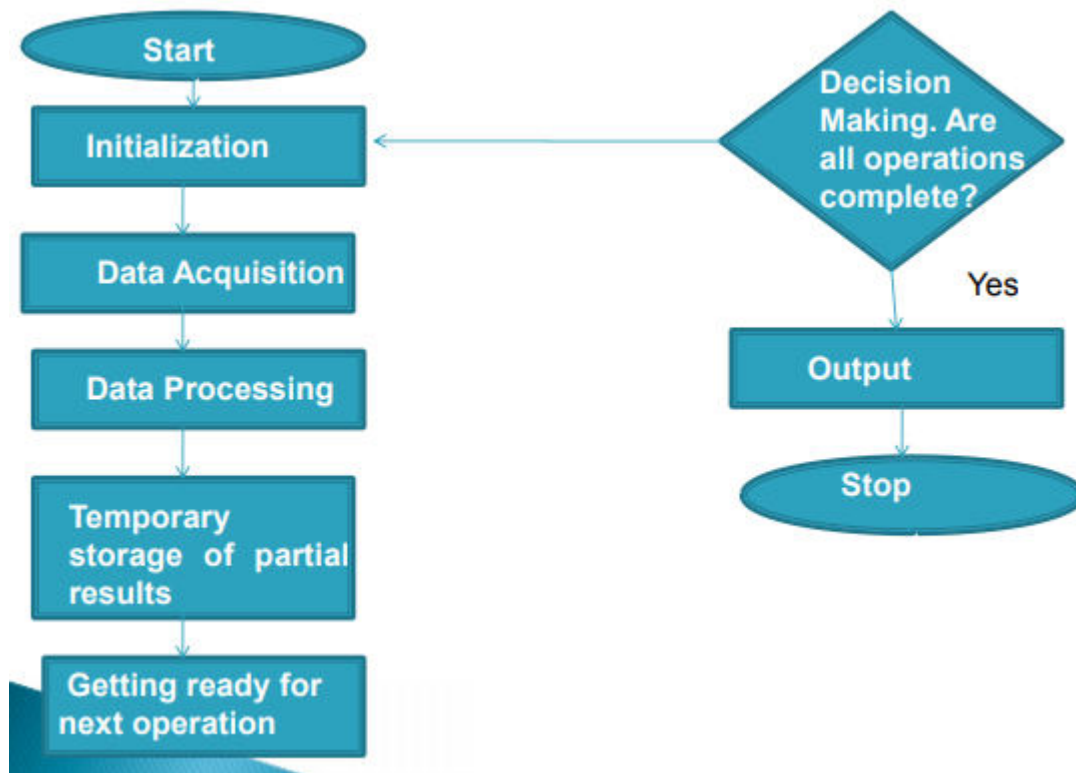
### Detailed content of the Lecture:

- Continuous loop-repeat task continuously
- Conditional loop-repeats a task until certain data conditions are met.
- Flowchart of continuous loop

Looping-In this technique, the program is instructed to execute certain set of instructions repeatedly to execute a particular task number of times.

Counting-This technique allows programmer to count how many times the instruction/set of instructions are executed.

Indexing-This technique allows programmer to point or refer the data stored in sequential memory location one by one.



Division of two 8 bit Numbers:

```

LDA 2000H ;divisor
MOV B,A
LDA 2001H ;dividend
MVI C, 00H
LOOP: CMP B
JC BRANCH
SUB B
INR C
JMP LOOP
BRANCH: STA 3000H ;remainder
MOV A,C
STA 3001H ;quotient
HLT
  
```

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=1d9sSoYYjcA>

**Important Books/Journals for further learning including the page nos.:**

- Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (260-262)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L24

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS

Date of Lecture:

**Topic of Lecture:** Subroutine and its programming

### Introduction :

- In computer programming, a subroutine is a sequence of program instructions that performs a specific task, packaged as a unit.
- In different programming languages, a subroutine may be called a procedure, a function, a routine, a method, or a subprogram. The generic term callable unit is sometimes used.

### Prerequisite knowledge for Complete understanding and learning of Topic:

Basic Programming of 8085

### Detailed content of the Lecture:

#### Subroutine in 8085

This unit can then be used in programs wherever that particular task have to be performed. A subroutine is often coded so that it can be started (called) several times and from several places during one execution of the program, including from other subroutines, and then branch back (return) to the next instruction after the call, once the subroutine's task is done.

It is implemented by using Call and Return instructions. The different types of subroutine instructions are

- Unconditional Call instruction
- Conditional Call instruction
- Unconditional Return instruction
- Conditional Return instruction

### Unconditional Call instruction

CALL address is the format for unconditional call instruction. After execution of this instruction program control is transferred to a sub-routine whose starting address is specified in the instruction. 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 subroutine and value of PC is pushed into stack only if condition is satisfied.

NSTRUCTION	PARAMETER	COMMENT
CC	16-bit address	Call at address if cy (carry flag) = 1
CNC	16-bit address	Call at address if cy (carry flag) = 0
CZ	16-bit address	Call at address if ZF (zero flag) = 1
CNZ	16-bit address	Call at address if ZF (zero flag) = 0
CPE	16-bit address	Call at address if PF (parity flag) = 1
CPO	16-bit address	Call at address if PF (parity flag) = 0
CN	16-bit address	Call at address if SF (signed flag) = 1
CP	16-bit address	Call at address if SF (signed flag) = 0

### Unconditional Return instruction

RET is the instruction used to mark the end of sub-routine. It has no parameter. After execution of this instruction program control is transferred back to main program from where it had stopped. Value of PC (Program Counter) is retrieved from the memory stack and value of SP (Stack Pointer) is incremented by 2.

### Conditional Return instruction

By these instructions program control is transferred back to main program and value of PC is popped from stack only if condition is satisfied. There is no parameter for return instruction.

INSTRUCTION	COMMENT
RC	Return from subroutine if cy (carry flag) = 1
RNC	Return from subroutine if cy (carry flag) = 0
RZ	Return from subroutine if ZF (zero flag) = 1
RNZ	Return from subroutine if ZF (zero flag) = 0
RPE	Return from subroutine if PF (parity flag) = 1
RPO	Return from subroutine if PF (parity flag) = 0
RN	Return from subroutine if SF (signed flag) = 1
RP	Return from subroutine if SF (signed flag) = 0

#### Advantages of Subroutine

1. Decomposing a complex programming task into simpler steps.
2. Reducing duplicate code within a program.
3. Enabling reuse of code across multiple programs.
4. Improving tractability or makes debugging of a program easy.

#### Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=2aCR5l26jJY>

#### Important Books/Journals for further learning including the page nos.:

Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (291-293)



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L25

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code: 16EEEC09 / MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS

Date of Lecture:

**Topic of Lecture:** PIC Microcontroller Concepts

### Introduction :

- PIC micro chips are designed with a Harvard architecture, and are offered in various device families.
- The baseline and mid-range families use 8-bit wide data memory, and the high-end families use 16-bit data memory. The latest series, PIC32MZ is a 32-bit MIPS-based microcontroller.
- Instruction words are in sizes of 12-bit (PIC10 and PIC12), 14-bit (PIC16) and 24-bit (PIC24 and dsPIC). The binary representations of the machine instructions vary by family and are shown in PIC instruction listings.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

1.Processors and controllers

**Detailed content of the Lecture:**

### Architecture of PIC Microcontroller

- CPU (Central Processing Unit)
- Memory Organization
- Random Access Memory (RAM)
- Read Only Memory (ROM)
- Electrically Erasable Programmable Read Only Memory (EEPROM)
- Flash Memory
- Stack
- BUS

The PIC architecture is characterized by its multiple attributes:

- Separate code and data spaces (Harvard architecture)
  - Except PIC32: The MIPS M4K architecture's separate data and instruction paths are effectively merged into a single common address space by the System Bus Matrix module.
- A small number of fixed-length instructions
- Most instructions are single-cycle (2 clock cycles, or 4 clock cycles in 8-bit models), with one delay cycle on branches and skips
- One accumulator (W0), the use of which (as source operand) is implied (i.e. is not encoded in the opcode)
- All RAM locations function as registers as both source and/or destination of math and other functions.
- A hardware stack for storing return addresses
- A small amount of addressable data space (32, 128, or 256 bytes, depending on the family), extended through banking
- Data-space mapped CPU, port, and peripheral registers
- ALU status flags are mapped into the data space
- The program counter is also mapped into the data space and writable (this is used to implement indirect jumps).

There is no distinction between memory space and register space because the RAM serves the job of both memory and registers, and the RAM is usually just referred to as the register file or simply as the registers.

### **Data space (RAM)**

- PICs have a set of registers that function as general-purpose RAM. Special-purpose control registers for on-chip hardware resources are also mapped into the data space. The addressability of memory varies depending on device series, and all PIC devices have some banking mechanism to extend addressing to additional memory. Later series of devices feature move instructions, which can cover the whole addressable space, independent of the selected bank. In earlier devices, any register move had to be achieved through the accumulator.
- To implement indirect addressing, a "file select register" (FSR) and "indirect register" (INDF) are used. A register number is written to the FSR, after which reads from or writes to INDF will actually be from or to the register pointed to by FSR. Later devices extended this concept with post- and pre- increment/decrement for greater efficiency in accessing sequentially stored data. This also allows FSR to be treated almost like a stack pointer (SP).

### **Code space**

- The code space is generally implemented as on-chip ROM, EPROM or flash ROM. In general, there is no provision for storing code in external memory due to the lack of an external memory interface. The exceptions are PIC17 and select high pin count PIC18 devices.

### **Word size**

- All PICs handle (and address) data in 8-bit chunks. However, the unit of addressability of the code space is not generally the same as the data space. For example, PICs in the baseline (PIC12) and mid-range (PIC16) families have program memory addressable in the same wordsize as the instruction width, i.e. 12 or 14 bits respectively. In contrast, in the PIC18 series, the program memory is addressed in 8-bit increments (bytes), which differs from the instruction width of 16 bits.
- In order to be clear, the program memory capacity is usually stated in number of (single-word) instructions, rather than in bytes.

## Stacks

- PICs have a hardware call stack, which is used to save return addresses. The hardware stack is not software-accessible on earlier devices, but this changed with the 18 series devices.
- Hardware support for a general-purpose parameter stack was lacking in early series, but this greatly improved in the 18 series, making the 18 series architecture more friendly to high-level language compilers.

## Instruction set

- PIC's instructions vary from about 35 instructions for the low-end PICs to over 80 instructions for the high-end PICs. The instruction set includes instructions to perform a variety of operations on registers directly, the accumulator and a literal constant or the accumulator and a register, as well as for conditional execution, and program branching.
- Some operations, such as bit setting and testing, can be performed on any numbered register, but bi-operand arithmetic operations always involve W (the accumulator), writing the result back to either W or the other operand register. To load a constant, it is necessary to load it into W before it can be moved into another register. On the older cores, all register moves needed to pass through W, but this changed on the "high-end" cores.
- PIC cores have skip instructions, which are used for conditional execution and branching. The skip instructions are "skip if bit set" and "skip if bit not set". Because cores before PIC18 had only unconditional branch instructions, conditional jumps are implemented by a conditional skip (with the opposite condition) followed by an unconditional branch. Skips are also of utility for conditional execution of any immediate single following instruction. It is possible to skip skip instructions. For example, the instruction sequence "skip if A; skip if B; C" will execute C if A is true or if B is false.

The 18 series implemented shadow registers, registers which save several important registers during an interrupt, providing hardware support for automatically saving processor state when servicing interrupts.

In general, PIC instructions fall into five classes:

1. Operation on working register (WREG) with 8-bit immediate ("literal") operand. E.g. `movlw` (move literal to WREG), `andlw` (AND literal with WREG). One instruction peculiar to the PIC is `retlw`, load immediate into WREG and return, which is used with computed branches to produce lookup tables.
2. Operation with WREG and indexed register. The result can be written to either the Working register (e.g. `addwf reg,w`) or the selected register (e.g. `addwf reg,f`).
3. Bit operations. These take a register number and a bit number, and perform one of 4 actions: set or clear a bit, and test and skip on set/clear. The latter are used to perform conditional branches. The usual ALU status flags are available in a numbered register so operations such as "branch on carry clear" are possible.
4. Control transfers. Other than the skip instructions previously mentioned, there are only two: `goto` and `call`.
5. A few miscellaneous zero-operand instructions, such as return from subroutine, and `sleep` to enter low-power mode.

## Advantages

- Small instruction set to learn
- RISC architecture
- Built-in oscillator with selectable speeds
- Easy entry level, in-circuit programming plus in-circuit debugging PICkit units available for less than \$50
- Inexpensive microcontrollers



**Limitations**

- One accumulator
- Register-bank switching is required to access the entire RAM of many devices
- Operations and registers are not orthogonal; some instructions can address RAM and/or immediate constants, while others can use the accumulator only.

The following stack limitations have been addressed in the **PIC18** series, but still apply to earlier cores:

- The hardware call stack is not addressable, so preemptive task switching cannot be implemented
- Software-implemented stacks are not efficient, so it is difficult to generate reentrant code and support local variables

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=EY15NUjnM-A>

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 112-117)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L26

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code : 16EEC09 /MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS

Date of Lecture:

**Topic of Lecture:** PIC 16C6X Architecture

### Introduction :

- The PIC16CXX is a family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.
- All PIC16/17 microcontrollers employ an advanced RISC architecture. The PIC16CXX microcontroller family has enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with separate 8-bit wide data.

### Prerequisite knowledge for Complete understanding and learning of Topic:

- 1.Processors and controllers
- 2.Central processing unit
- 3.Memory Devices

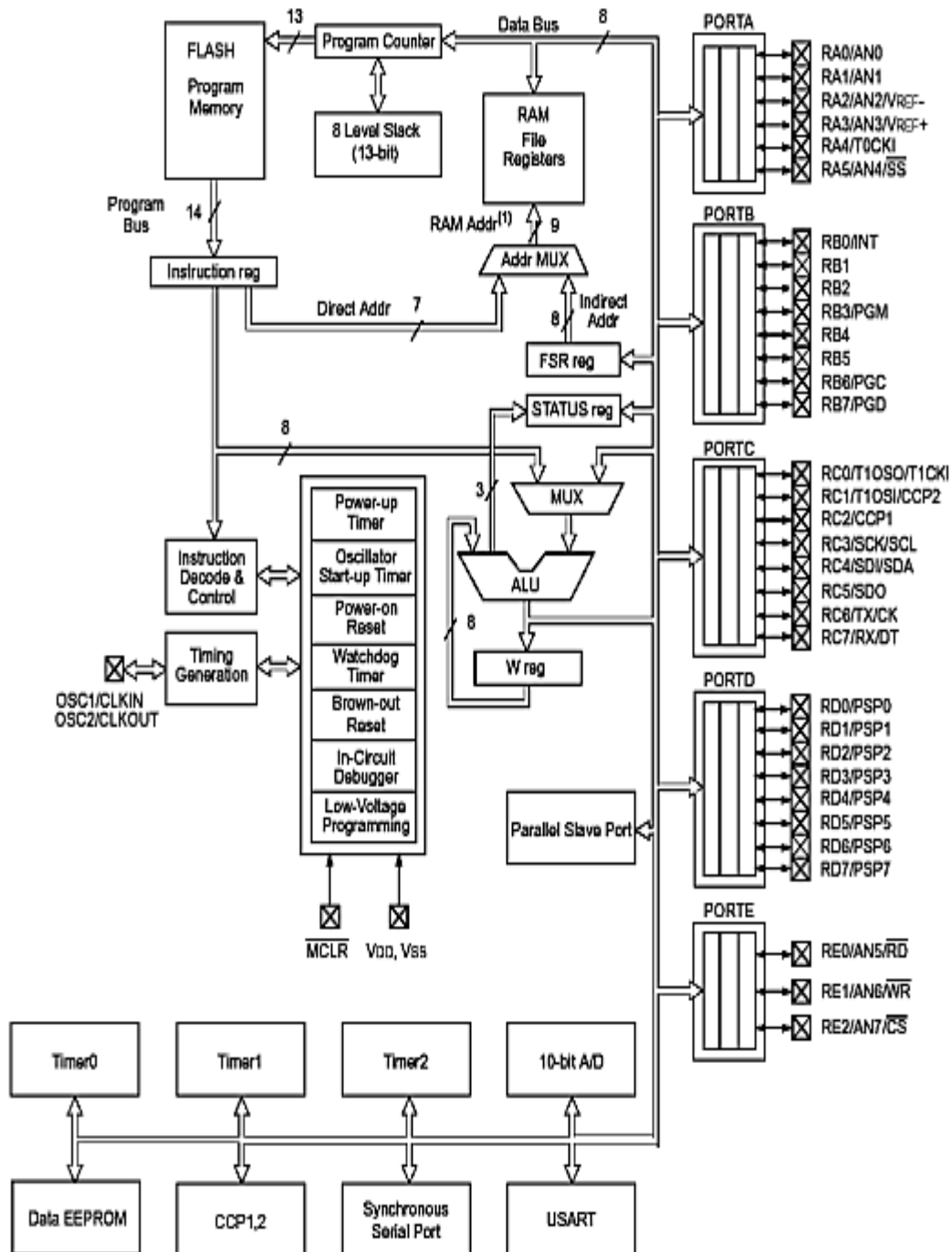
### Detailed content of the Lecture:

#### PIC16C6X

- PIC16CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class. The PIC16C61 device has 36 bytes of RAM and 13 I/O pins.
- In addition a timer/counter is available. The PIC16C62/62A/R62 devices have 128 bytes of RAM and 22 I/O pins. In addition, several peripheral features are available, including: three timer/counters, one Capture/Compare/PWM module and one serial port. The Synchronous Serial Port can be configured as either a 3-wire Serial Peripheral Interface the two-wire Inter-Integrated Circuit (I2C) bus.
- The PIC16C63/R63 devices have 192 bytes of RAM, while the PIC16C66 has 368 bytes. All three devices have 22 I/O pins. In addition, several peripheral features are available, including: three timer/counters, two Capture/Compare/PWM modules and two serial ports.

# Architectural Overview

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

- The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture, in which, program and data are accessed from separate memories using separate buses.
- This improves bandwidth over traditional von Neumann architecture where program and data may be fetched from the same memory using the same bus. Separating program and data busses further allows instructions to be sized differently than 8-bit wide data words.
- Instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two stage pipeline overlaps fetch and execution of instructions
- Consequently, all instructions execute in a single cycle (200 ns @ 20 MHz) except for program branches. The PIC16C61 addresses 1K x 14 of program memory. The PIC16C62/62A/R62/64/64A/R64 address 2K x 14 of program memory, and the PIC16C63/R63/65/65A/R65 devices address 4K x 14 of program memory. The PIC16C66/67 address 8K x 14

**Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=5BjrjeNYMZg>

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 118-124)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L27

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code : 16EEC09 / MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : III - PROGRAMMING AND ADVANCED CONTROLLERS

Date of Lecture:

**Topic of Lecture:** PIC 16C7X Architecture

### Introduction :

- The PIC16C7X is a family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers with integrated analog-to-digital (A/D) converters, in the PIC16CXX mid-range family.
- The PIC16C7X family has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption.
- There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals

### Prerequisite knowledge for Complete understanding and learning of Topic:

1.Processors and controllers

### Detailed content of the Lecture:

#### PIC16C6X PIC16C7X Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module(s)
- Capture is 16-bit, max. resolution is 12.5 ns, Compare is 16-bit, max. resolution is 200 ns, PWM max. resolution is 10-bit
- 8-bit multichannel analog-to-digital converter

- Synchronous Serial Port (SSP) with SPI<sup>□</sup> and I<sup>2</sup>C<sup>□</sup>
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls
- Brown-out detection circuitry for Brown-out Reset (BOR)
- 

PIC16C7X Features	72	73	73A	74	74A	76	77
Program Memory (EPROM) x 14	2K	4K	4K	4K	4K	8K	8K
Data Memory (Bytes) x 8	128	192	192	192	192	368	368
I/O Pins	22	22	22	33	33	22	33
Parallel Slave Port	–	–	–	Yes	Yes	–	Yes
Capture/Compare/PWM Modules	1	2	2	2	2	2	2
Timer Modules	3	3	3	3	3	3	3
A/D Channels	5	5	5	8	8	5	8
Serial Communication	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART	SPI/I <sup>2</sup> C, USART
In-Circuit Serial Programming	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Brown-out Reset	Yes	–	Yes	–	Yes	Yes	Yes
Interrupt Sources	8	11	11	12	12	11	12

## PIC16C7X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC16C7X Product Identification System section at the end of this data sheet. When placing orders, please use that page of the data sheet to specify the correct part number.

For the PIC16C7X family, there are two device “types” as indicated in the device number:

1. C, as in PIC16C74. These devices have EPROM type memory and operate over the standard voltage range.

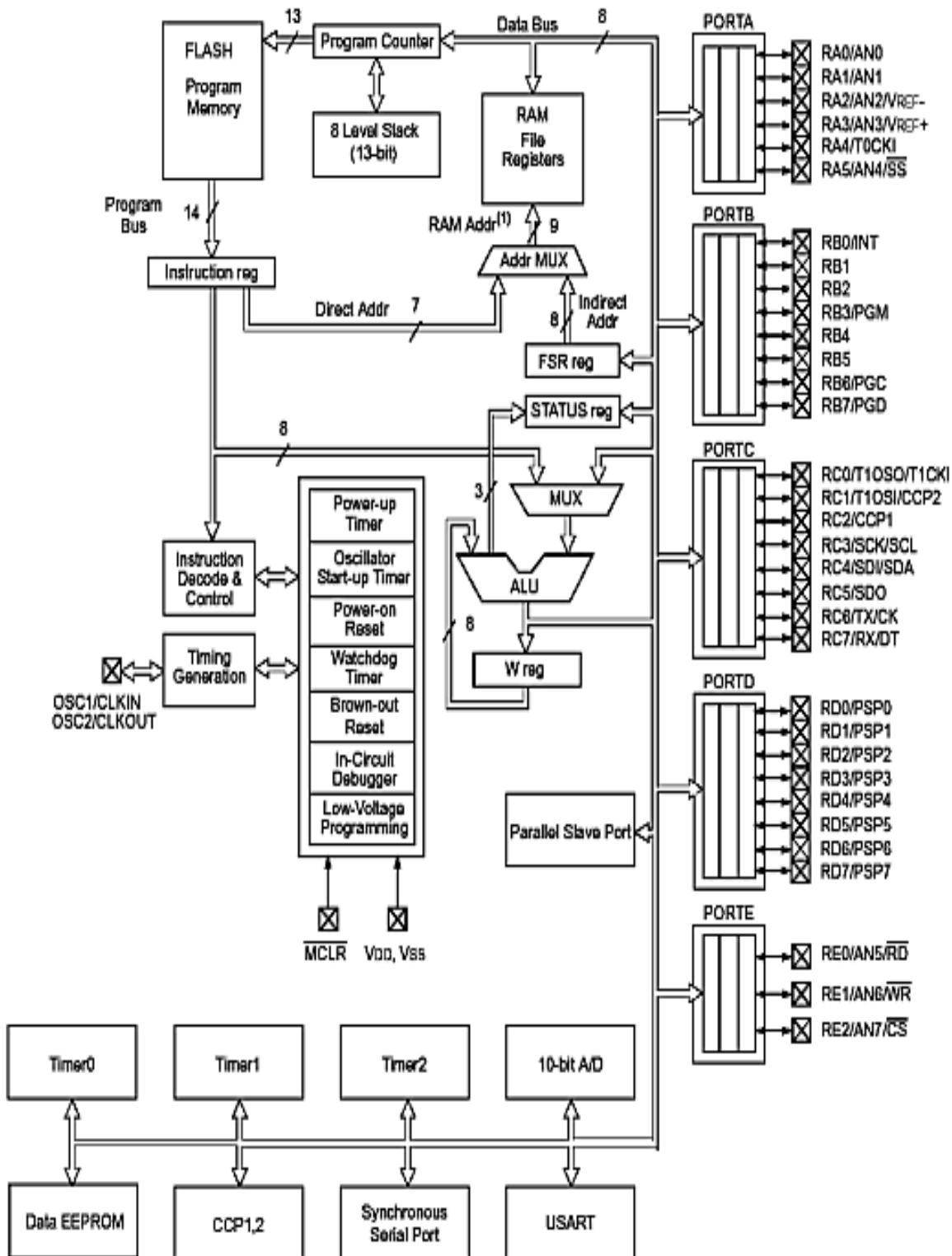
2. **LC**, as in PIC16LC74. These devices have EPROM type memory and operate over an extended voltage range.

### ARCHITECTURAL OVERVIEW

- The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture, in which, program and data are accessed from separate memories using separate buses.
- This improves bandwidth over traditional von Neumann architecture in which program and data are fetched from the same memory using the same bus.
- Separating program and data buses further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single word instructions.
- A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions

Device	Program Memory	Data Memory
PIC16C72	2K x 14	128 x 8
PIC16C73	4K x 14	192 x 8
PIC16C73A	4K x 14	192 x 8
PIC16C74	4K x 14	192 x 8
PIC16C74A	4K x 14	192 x 8
PIC16C76	8K x 14	368 x 8
PIC16C77	8K x 14	386 x 8

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.



**Video Content / Details of website for further learning (if any):**

<https://www.promelec.ru › datasheet › pic16c7x>

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 132-135)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L28

## LECTURE HANDOUTS

EEE

III/IV

Course Name with Code: 16EEEC09 / MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : IV - PROGRAMMING AND INTERFACING OF 8085&8051

Date of Lecture:

### Topic of Lecture: Study on Need of Interfacing

#### Introduction :

Interfacing is one of the important concepts in microcontroller 8051 because the microcontroller is a CPU that can perform some operation on a data and gives the output. However to perform the operation we need an input device to enter the data and in turn output device displays the results of the operation.

#### Prerequisite knowledge for Complete understanding and learning of Topic:

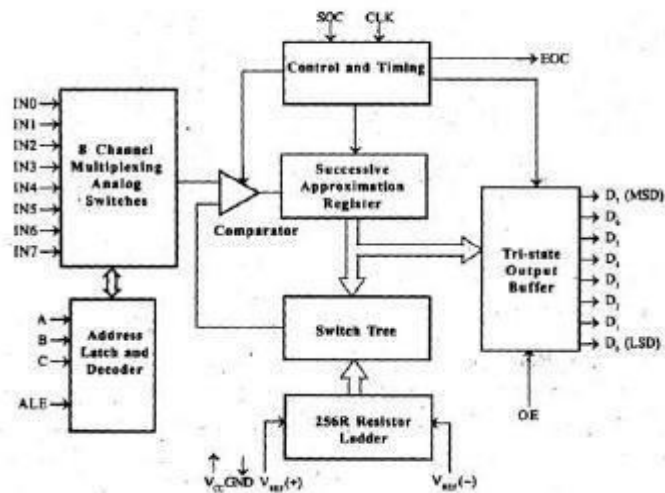
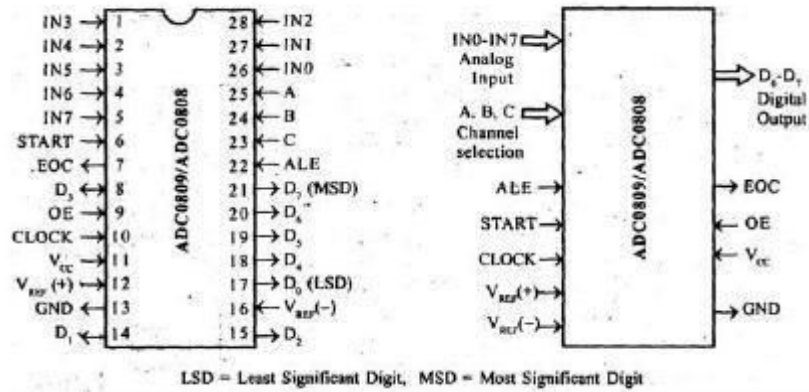
- Basics of Processor and Controller

#### Detailed content of the Lecture:

### ADC Interfacing with 8085 Microprocessor

#### Features

- The ADC0809 is an 8-bit successive approximation type ADC with inbuilt 8-channel multiplexer.
- The ADC0809 is suitable for interface with 8086 microprocessor.
- The ADC0809 is available as a 28 pin IC in DIP (Dual Inline Package).
- The ADC0809 has a total unadjusted error of  $\pm 1$  LSD (Least Significant Digit).
- The ADC0808 is also same as ADC0809 except the error. The total unadjusted error in ADC0808 is  $\pm 1/2$  LSD.



- The successive approximation register (SAR) performs eight iterations to determine the digital code for input value. The SAR is reset on the positive edge of START pulse and start the conversion process on the falling edge of START pulse.
- A conversion process will be interrupted on receipt of new START pulse. The End-Of-Conversion (EOC) will go low between 0 and 8 clock pulses after the positive edge of START pulse.
- The ADC can be used in continuous conversion mode by tying the EOC output to START input. In this mode an external START pulse should be applied whenever power is switched ON.
- The 256R ladder network has been provided instead of conventional R/2R ladder because of its inherent monotonic, which guarantees no missing digital codes. Also the 256R resistor network does not cause load variations on the reference voltage.
- The comparator in ADC0809/ADC0808 is a chopper-stabilized comparator. It converts the DC input signal into an AC signal, and amplifies the AC sign using high gain AC amplifier Then it converts AC signal to DC signal.
- This technique limits the drift component of the amplifier, because the drift is a DC component and it is not amplified/passed by the AC amplifier. This makes the ADC extremely insensitive to temperature, long term drift and input offset errors.

- In ADC conversion process the input analog value is quantized and each quantized analog value will have a unique binary equivalent. The quantization step in ADC0809/ADC0808 is given by,

$$Q_{\text{step}} = \frac{V_{\text{REF}}}{2^8} = \frac{V_{\text{REF}(+)} - V_{\text{REF}(-)}}{256_{10}}$$

The digital data corresponding to an analog input ( $V_{in}$ ) is given by,

$$\text{Digital data} = \left( \frac{V_{in}}{Q_{\text{step}}} - 1 \right)_{10}$$

**Video Content / Details of website for further learning :**

[www.youtube.com/watch?v=liRPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRPtvj7bFU&noredirect=1)

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 449-452)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L29

## LECTURE HANDOUTS

EEE

III/IV

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : IV - PROGRAMMING AND INTERFACING OF 8085&8051

Date of Lecture:

**Topic of Lecture:** Programmable peripheral Interface 8255(PPI)

### Introduction :

- The 8255 provides 24 parallel input/output lines with a variety of programmable operating modes. The 8255 is a member of the MCS-85 Family of chips, designed by Intel for use with their 8085 and 8086 microprocessors and their descendants. It was first available in a 40-pin DIP and later a 44-pin PLCC packages.

### Prerequisite knowledge for Complete understanding and learning of Topic:

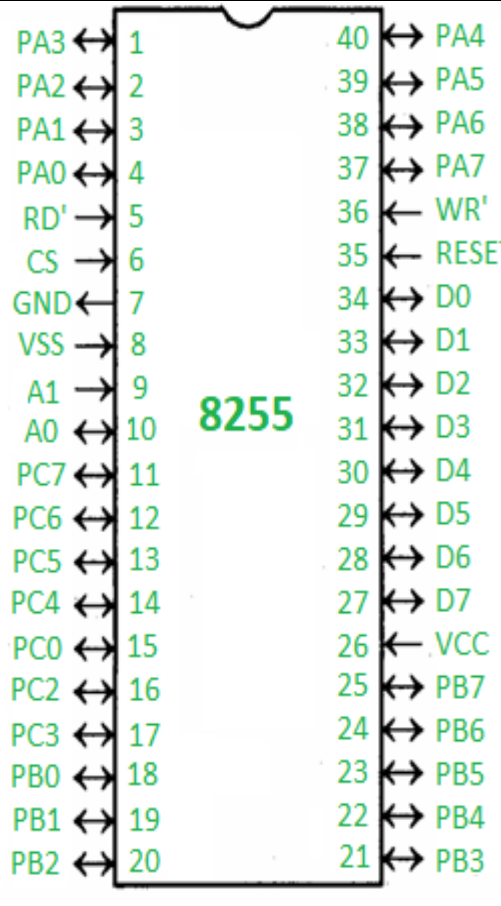
- Interfacing Concepts

### Detailed content of the Lecture:

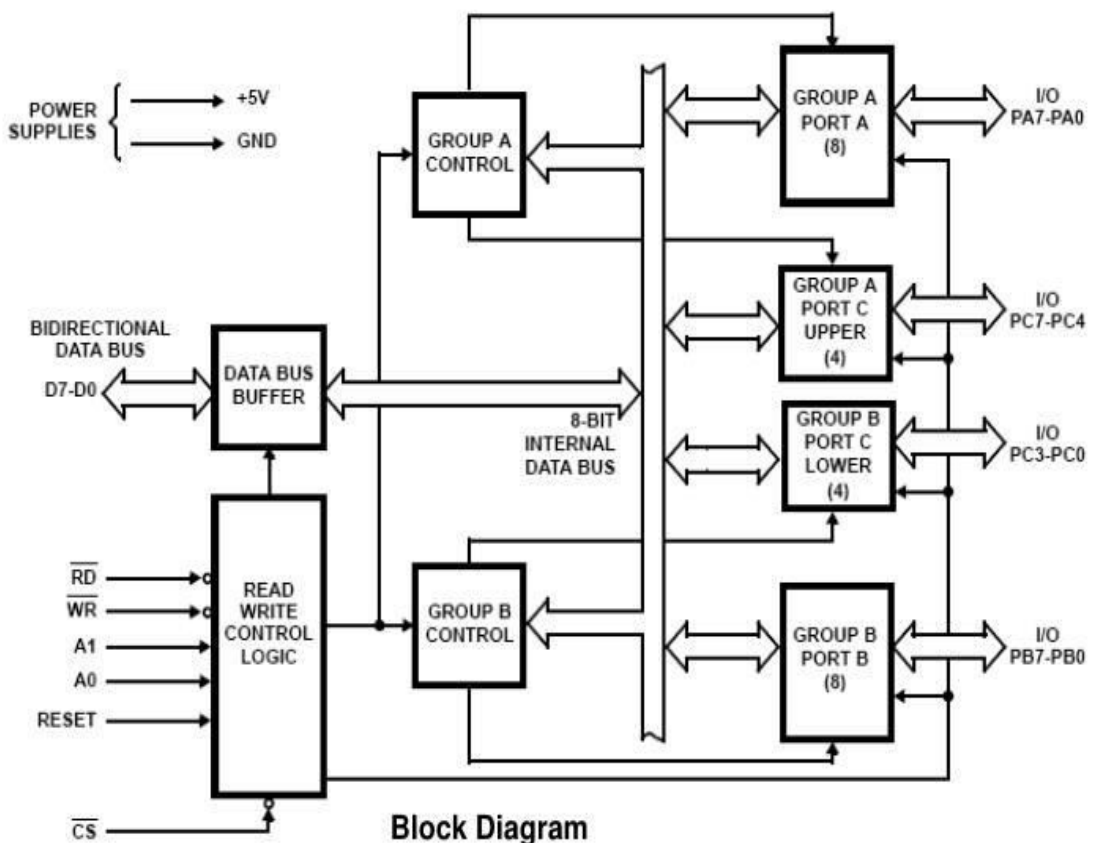
#### PIN DIAGRAM OF 8255

The signal description of 8255 are briefly presented as follows :

- **PA7-PA0:** These are eight port A lines that acts as either latched output or buffered input lines depending upon the control word loaded into the control word register.
- **PC7-PC4 :** Upper nibble of port C lines. They may act as either output latches or input buffers lines.This port also can be used for generation of handshake lines in mode 1 or mode 2.
- **PC3-PC0 :** These are the lower port C lines, other details are the same as PC7-PC4 lines.
- **PB0-PB7 :** These are the eight port B lines which are used as latched output lines or buffered input lines in the same way as port A.
- **RD :** This is the input line driven by the microprocessor and should be low to indicate read operation to 8255.
- **WR :** This is an input line driven by the microprocessor. A low on this line indicates write operation.
- **CS :** This is a chip select line. If this line goes low, it enables the 8255 to respond to RD and WR signals, otherwise RD and WR signal are neglected



## 8255: Programmable Peripheral Interface



**A1-A0** : These are the address input lines and are driven by the microprocessor. These lines A1-A0 with RD, WR and CS from the following operations for 8255. These address lines are used for addressing any one of the four registers,i.e. three ports and a control word register as given in table below

- In case of 8086 systems, if the 8255 is to be interfaced with lower order data bus, the A0 and A1 pins of 8255 are connected with A1 and A2 respectively.

**D0-D7** :

These are the data bus lines those carry data or control word to/from the Microprocessor.

**RESET** :

A logic high on this line clears the control word register of 8255. All ports are set as input ports by default after reset.

### **Operational Modes of 8255**

There are two main operational modes of 8255:

1. Input/output mode
2. Bit set/reset mode

#### **Input/Output Mode**

There are three types of the input/output mode. They are as follows:

#### **Mode 0**

In this mode, the ports can be used for simple input/output operations without handshaking. If both port A and B are initialized in mode 0, the two halves of port C can be either used together as an additional 8-bit port, or they can be used as individual 4-bit ports. Since the two halves of port C are independent, they may be used such that one-half is initialized as an input port while the other half is initialized as an output port. The input output features in mode 0 are as follows:

1. O/p are latched.
2. I/p are buffered not latched.
3. Port do not have handshake or interrupt capability.

#### **Mode 1**

When we wish to use port A or port B for handshake (strobed) input or output operation, we initialise that port in mode 1 (port A and port B can be initialised to operate in different modes,ie, for eg, port A can operate in mode 0 and port B in mode 1). Some of the pins of port C function as handshake lines.

For port B in this mode (irrespective of whether is acting as an input port or output port), PC0, PC1 and PC2 pins function as handshake lines.

If port A is initialised as mode 1 input port, then, PC3, PC4 and PC5 function as handshake signals. Pins PC6 and PC7 are available for use as input/output lines.

The mode 1 which supports handshaking has following features: 1. Two ports i.e. port A and B can be use as 8-bit i/o port. 2. Each port uses three lines of port c as handshake signal and remaining two signals can be function as i/o port. 3. interrupt logic is supported. 4. Input and Output data are latched.

#### **Mode 2**

Only group A can be initialised in this mode. Port A can be used for bidirectional handshake data transfer. This means that data can be input or output on the same eight lines (PA0 - PA7). Pins PC3 - PC7 are used as handshake lines for port A. The remaining pins of port C (PC0 - PC2) can be used as input/output lines if group B is initialised in mode 0. In this mode, the 8255 may be used to extend the system bus to a slave microprocessor or to transfer data bytes to and from a floppy disk controller.

**Bit Set/Reset (BSR) mode**

In this mode only port b can be used (as an output port). Each line of port C (PC0 - PC7) can be set/reset by suitably loading the command word register.no effect occurs in input-output mode.

The individual bits of port c can be set or reset by sending the signal OUT instruction to the control register.

**Video Content/ Details of website for further learning :**

[https://www.youtube.com/watch?v=\\_M8hDkRAL6M](https://www.youtube.com/watch?v=_M8hDkRAL6M)

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 452-458)

**Course Faculty**





## LECTURE HANDOUTS

L30

EEE

III/V

Course Name with Code :16EEEC09 / MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : IV - PROGRAMMING AND INTERFACING OF 8085&8051

Date of Lecture:

**Topic of Lecture:** Programmable Interrupt controller

### Introduction :

The Intel 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single 5V supply. Circuitry is static, requiring no clock input.

### Prerequisite knowledge for Complete understanding and learning of Topic:

Interfacing Concepts

### Detailed content of the Lecture:

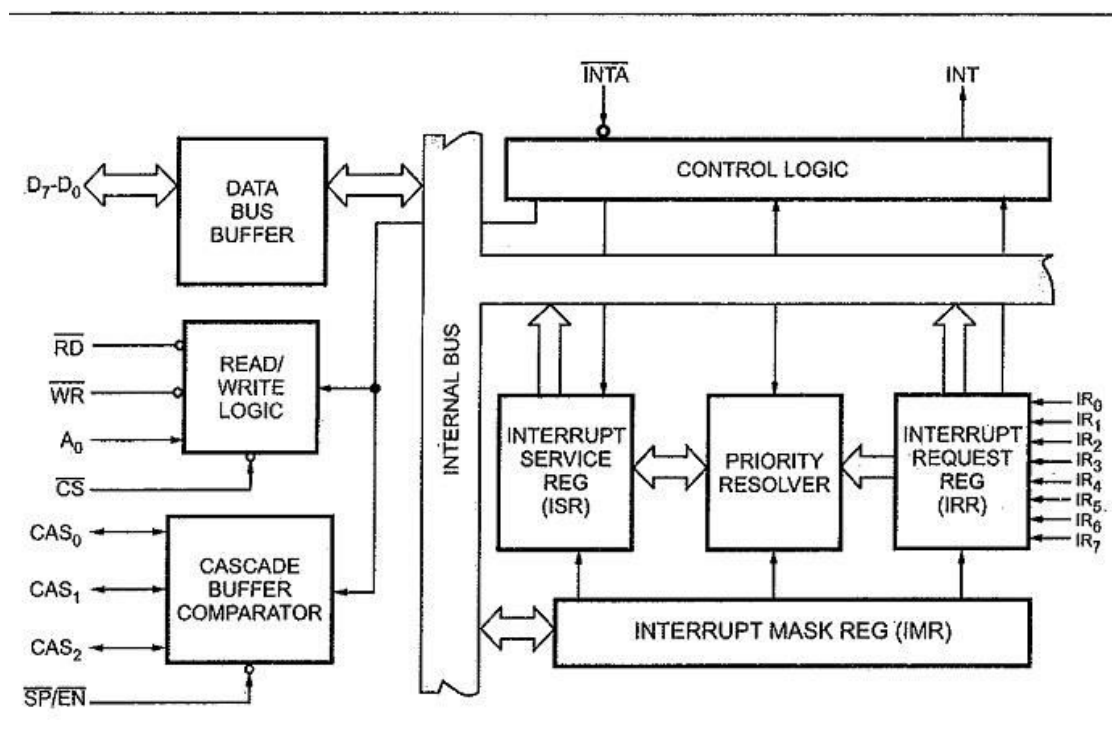
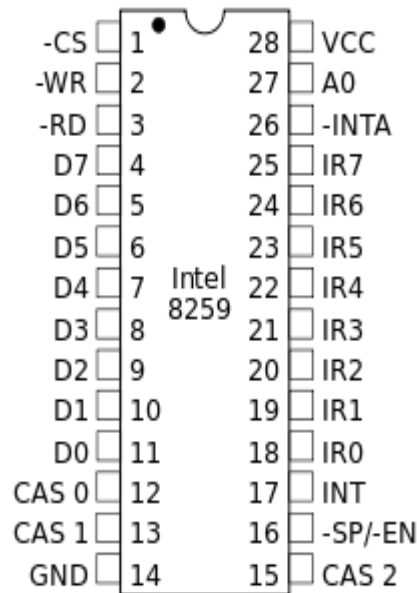


Fig. 14.71 Block diagram of 8259A

## PIN Diagram of 8259:



The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements. The 8259A is fully upward compatible with the Intel 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).

### INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

### PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during INTA pulse.

### INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower quality.

INT (INTERRUPT) This output goes directly to the CPU interrupt input. The VOH level on this line is designed to be fully compatible with the 8080A, 8085A and 8086 input levels.

### **INTA (INTERRUPT ACKNOWLEDGE)**

INTA pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (mPM) of the 8259A.

### **DATA BUS BUFFER**

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

### **READ/WRITE CONTROL LOGIC**

The function of this block is to accept OUTPUT commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

### **CS (CHIP SELECT)**

A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected. WR (WRITE) A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A. RD (READ) A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.

### **A0**

This input signal is used in conjunction with WR and RD signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

### **Video Content / Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=d7oFD-zQpuQ>

### **Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 458-462)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L31

EEE

III/IV

Course Name with Code : 16EEEC09/ MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : IV - PROGRAMMING AND INTERFACING OF 8085&8051

Date of Lecture:

**Topic of Lecture:** Timer 8254

**Introduction :**

The Intel 8253 and 8254 are Programmable Interval Timers (PTIs) designed for microprocessors to perform timing and counting functions using three 16-bit registers. Each counter has 2 input pins, i.e. Clock & Gate, and 1 pin for "OUT" output. To operate a counter, a 16-bit count is loaded in its register. On command, it begins to decrement the count until it reaches 0, then it generates a pulse that can be used to interrupt the CPU.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Basics of Timer Concepts

**Detailed content of the Lecture:**

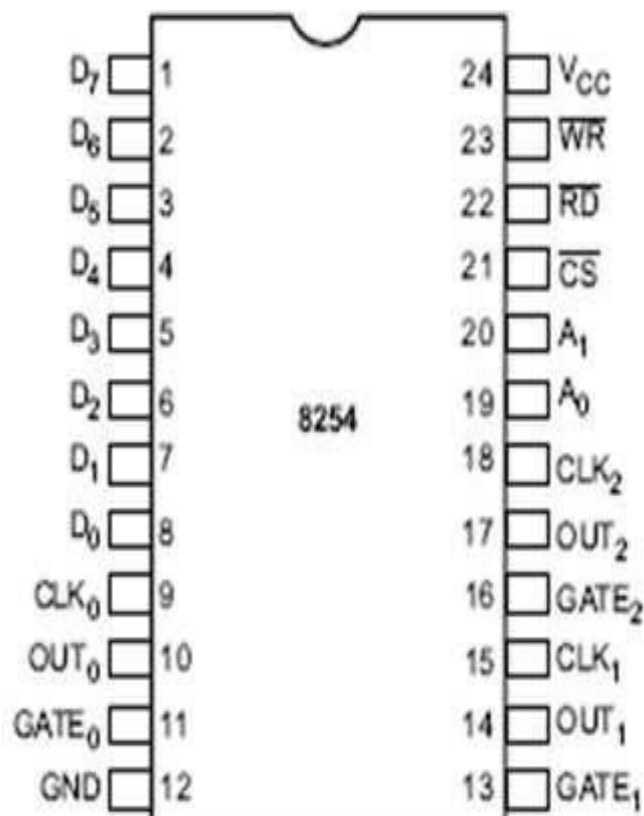
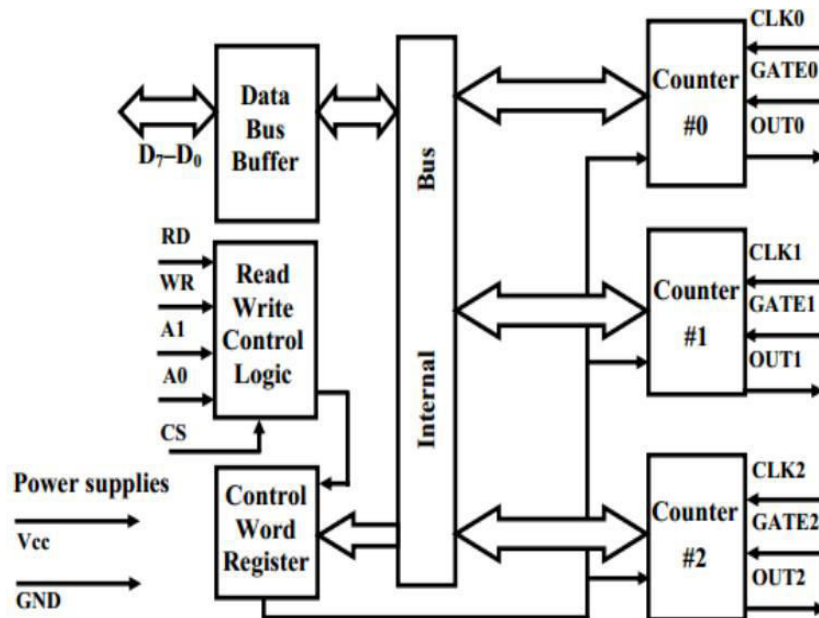
The most prominent features of 8253/54 are as follows –

- It has three independent 16-bit down counters.
- It can handle inputs from DC to 10 MHz.
- These three counters can be programmed for either binary or BCD count.
- It is compatible with almost all microprocessors.
- 8254 has a powerful command called READ BACK command, which allows the user to check the count value, the programmed mode, the current mode, and the current status of the counter.
- 8254 is a device designed to solve the timing control problems in a microprocessor.
- It has 3 independent counters, each capable of handling clock inputs up to 10 MHz and size of each counter is 16 bit.
- It operates in +5V regulated power supply and has 24 pin signals. All modes are

software programmable.

- The 8254 is an advanced version of 8253 which did not offered the feature of read back command.
- The basic block diagram of 8254 is:

## 8254 Block Diagram



### Data Bus Buffer:

It is a tri-state, bi-directional, 8-bit buffer, which is used to interface the 8253/54 to the system data bus. It has three basic functions –

- Programming the modes of 8253/54.
- Loading the count registers.
- Reading the count values.

### Read/Write Logic:

It includes 5 signals, i.e. RD, WR, CS, and the address lines  $A_0$  &  $A_1$ . In the peripheral I/O mode, the RD and WR signals are connected to IOR and IOW, respectively. In the memorymapped I/O mode, these are connected to MEMR and MEMW.

$A_1$	$A_0$	Result
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

Address lines  $A_0$  &  $A_1$  of the CPU are connected to lines  $A_0$  and  $A_1$  of the 8253/54, and CS is tied to a decoded address. The control word register and counters are selected according to the signals on lines  $A_0$  &  $A_1$ .

### Control Word Register:

This register is accessed when lines  $A_0$  &  $A_1$  are at logic 1. It is used to write a command word, which specifies the counter to be used, its mode, and either a read or write operation. Following table shows the result for various control inputs.

$A_1$	$A_0$	RD	WR	CS	Result
0	0	1	0	0	Write Counter 0
0	1	1	0	0	Write Counter 1
1	0	1	0	0	Write Counter 2
1	1	1	0	0	Write Control Word
0	0	0	1	0	Read Counter 0
0	1	0	1	0	Read Counter 1
1	0	0	1	0	Read Counter 2
1	1	0	1	0	No operation

X	X	1	1	0	No operation
X	X	X	X	1	No operation
<b>Video Content / Details of website for further learning (if any):</b>					
<a href="https://www.youtube.com/watch?v=ZtnhEqfw17c">https://www.youtube.com/watch?v=ZtnhEqfw17c</a>					
<b>Important Books/Journals for further learning including the page nos.:</b>					
Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (449-462)					

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L32

EEE

III/IV

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : IV - PROGRAMMING AND INTERFACING OF 8085&8051

Date of Lecture:

**Topic of Lecture:** DMA controller

**Introduction :**

- DMA mode of transfer is the fastest among all the mode of data transfer
- In this mode the device may transfer data directly / memory without any interface from CPU
- DMA is initiated only after receiving HLDA signal from CPU
- DMA mode of data transfer between several device DMA controller is used.

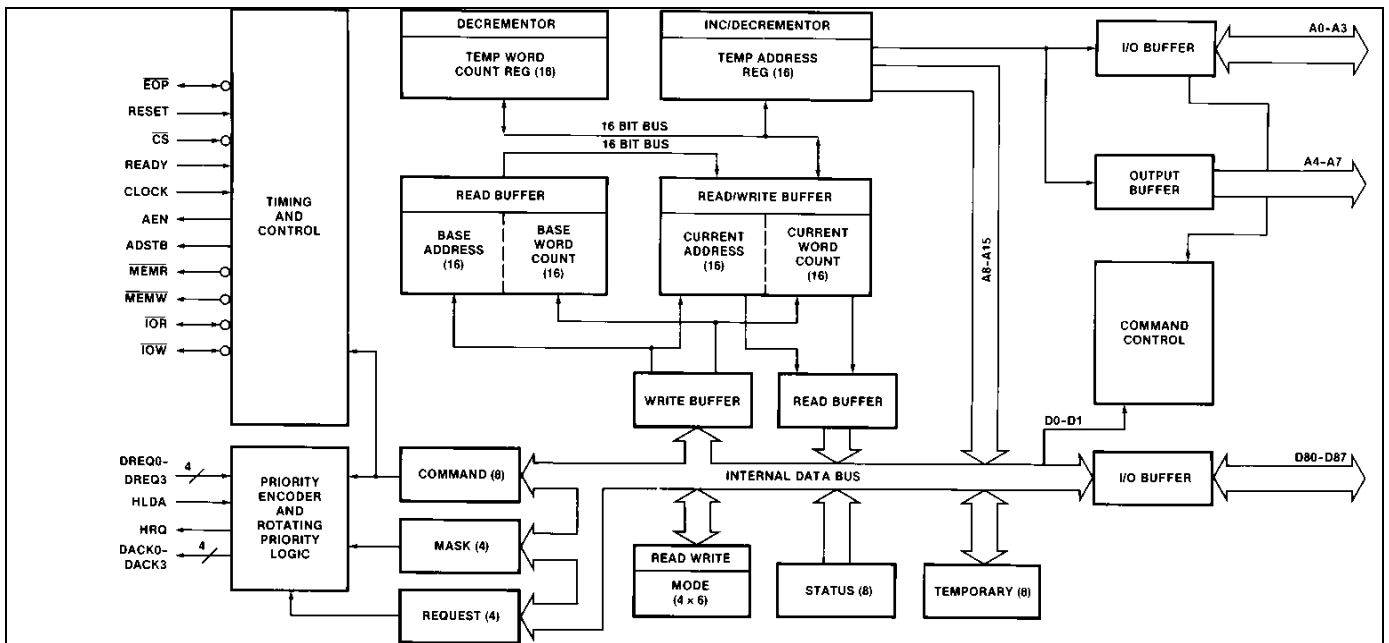
**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Working of DMA Controller

**Detailed content of the Lecture:**

- DMA controller provides an interface between the bus and the input-output devices. Although it transfers data without intervention of processor, it is controlled by the processor.
- The processor initiates the DMA controller by sending the starting address, Number of words in the data block and direction of transfer of data .i.e. from I/O devices to the memory or from main memory to I/O devices. More than one external device can be connected to the DMA controller.





### I/O buffer

The 8-bit bidirectional data buffer is interfaced with internal bus of 8237 and also with external data bus

### Timing and control unit

The control logic unit control the sequence of DMA operations of the following control signals are AEN, MEMR, MEMW

### RESET

Reset is an active high input which clears the Command, Status, Request and Temporary registers.

### READY

Ready is an input used to extend the memory read and write pulses from the 8237A to accommodate slow memories or I/O peripheral devices.

### CLOCK INPUT

Clock Input controls the internal operations of the 8237A and its rate of data transfers.

The input may be driven at up to 5 MHz

### MEMR

#### MEMORY READ:

The Memory Read signal is an active low three- state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.

### MEMW

#### MEMORY WRITE:

The Memory Write is an active low three-state output used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.

**IOR****I/O READ:**

I/O Read is a bidirectional active low three-state line.

In the Idle cycle, it is an input control signal used by the CPU to read the control registers.

In the Active cycle, it is an output control signal used by the 8237A to access data from a peripheral during a DMA Write transfer.

**IOW****I/O WRITE:**

I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the 8237A

**A0 - A3****ADDRESS:**

The four least significant address lines are bidirectional three-state signals. In the Idle cycle they are inputs and are used by the CPU to address the register to be loaded or read.

In the Active cycle they are outputs and provide the lower 4 bits of the output address.

**A4-A7****ADDRESS:**

The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during the DMA service.

**Video Content / Details of website for further learning:**

<https://www.youtube.com/watch?v=WtNicaqCS10>

**Important Books/Journals for further learning including the page nos.:**

Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (490-497)



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L33

EEE

III/IV

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : IV - PROGRAMMING AND INTERFACING OF 8085&8051

Date of Lecture:

**Topic of Lecture:** DMA controller

**Introduction :**

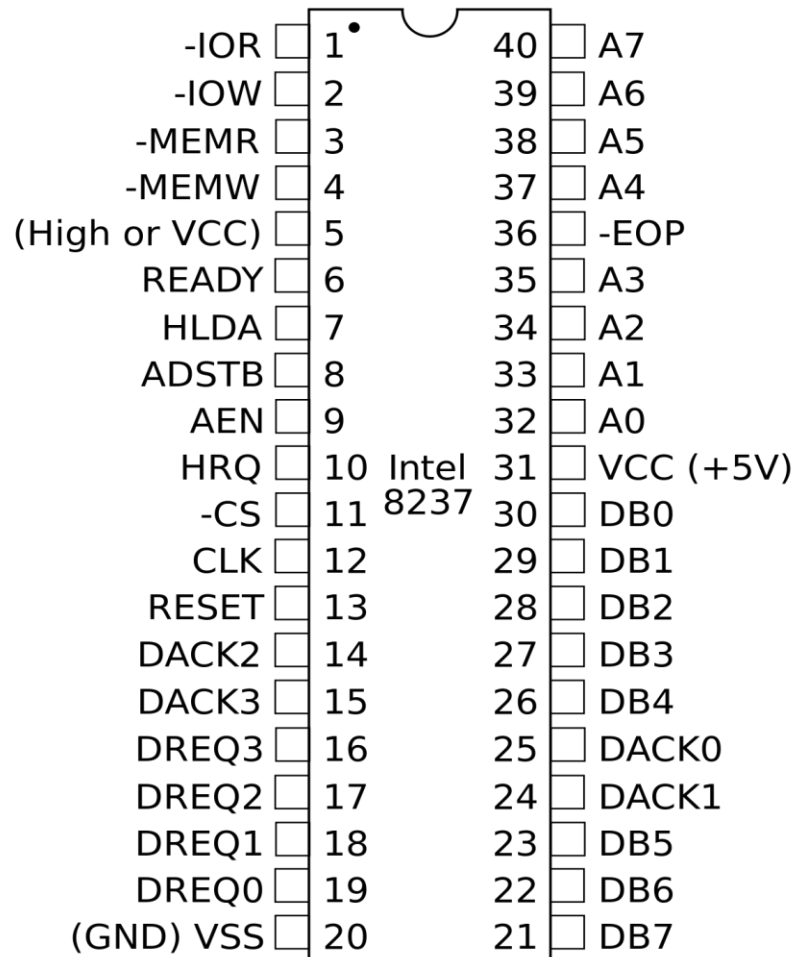
- The term DMA stands for direct memory access. The hardware device used for direct memory access is called the DMA controller.
- DMA controller is a control unit, part of I/O device's interface circuit, which can transfer blocks of data between I/O devices and main memory with minimal intervention from the processor.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Architecture
- Working of DMA Controller

**Detailed content of the Lecture:**

- DMA controller provides an interface between the bus and the input-output devices. Although it transfers data without intervention of processor, it is controlled by the processor.
- The processor initiates the DMA controller by sending the starting address, Number of words in the data block and direction of transfer of data .i.e. from I/O devices to the memory or from main memory to I/O devices. More than one external device can be connected to the DMA controller.

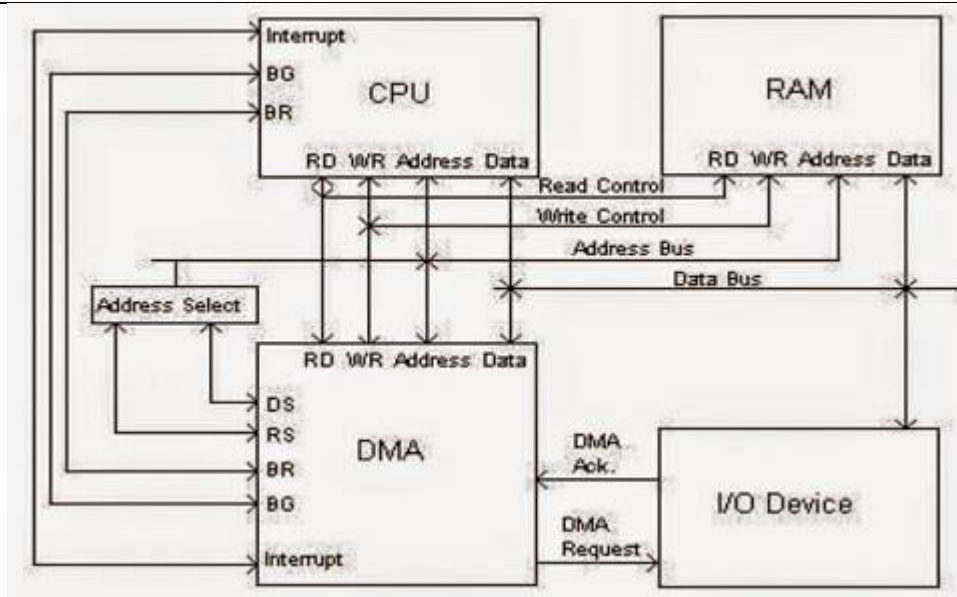


### Working of DMA Controller

- If the DMA controller is free, it requests the control of bus from the processor by raising the bus request signal. Processor grants the bus to the controller by raising the bus grant signal, now DMA controller is the bus master.
- The processor initiates the DMA controller by sending the memory addresses, number of blocks of data to be transferred and direction of data transfer.

The DMA transfers the data in three modes which include the following.

- Burst Mode:** In this mode DMA handover the buses to CPU only after completion of whole data transfer. Meanwhile, if the CPU requires the bus it has to stay idle and wait for data transfer.
- Cycle Stealing Mode:** In this mode, DMA gives control of buses to CPU after transfer of every byte. It continuously issues a request for bus control, makes the transfer of one byte and returns the bus. By this CPU doesn't have to wait for a long time if it needs a bus for higher priority task.
- Transparent Mode:** Here, DMA transfers data only when CPU is executing the instruction which does not require the use of buses.



### Advantages

- DMA speeds up the memory operations by bypassing the involvement of the CPU.
- The work overload on the CPU decreases.
- For each transfer, only a few numbers of clock cycles are required

### Disadvantages

- Cache coherence problem can be seen when DMA is used for data transfer.
- Increases the price of the system

### Video Content/ Details of website for further learning:

<https://www.youtube.com/watch?v=WnNicaqCS10>

### Important Books/Journals for further learning including the page nos.:

Gaonkar.R.S, Microprocessor architecture programming and applications with 8085,wiley eastern ltd, New Delhi 2013-Page no (490-497)

Course Faculty



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L34

EEE

III/IV

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : IV - PROGRAMMING AND INTERFACING OF 8085&8051

Date of Lecture:

**Topic of Lecture:** Universal Synchronous Asynchronous Receiver Transmitter (USART)8251

**Introduction :**

8251 Receives parallel data from the CPU is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication. & Also receives serial data from the outside transmits serial data after conversion. & transmits parallel data to the CPU after conversion

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Communication Modes

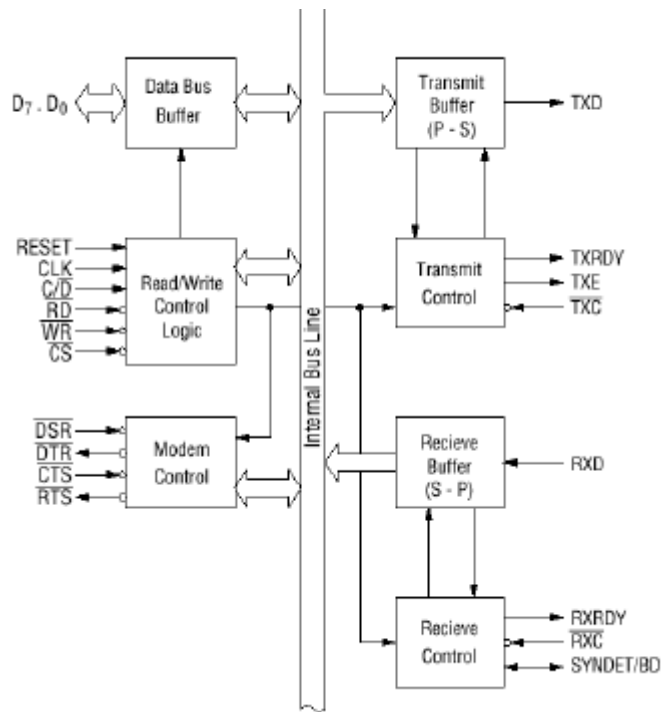
**Detailed content of the Lecture:**

- The 8251 is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication.
- As a peripheral device of a microcomputer system, the 8251receives parallel data from the CPU and transmits serial data after conversion.
- This device also receives serial data from the outside and transmits parallel data to the CPU after conversion

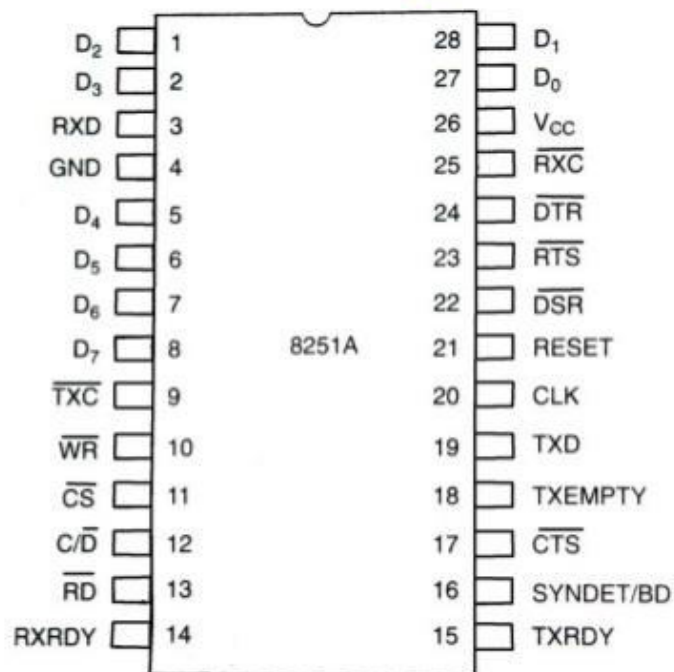
**Control Words**

There are two types of control word.

1. Mode instruction (setting of function)
2. Command (setting of operation)



## Pin diagram

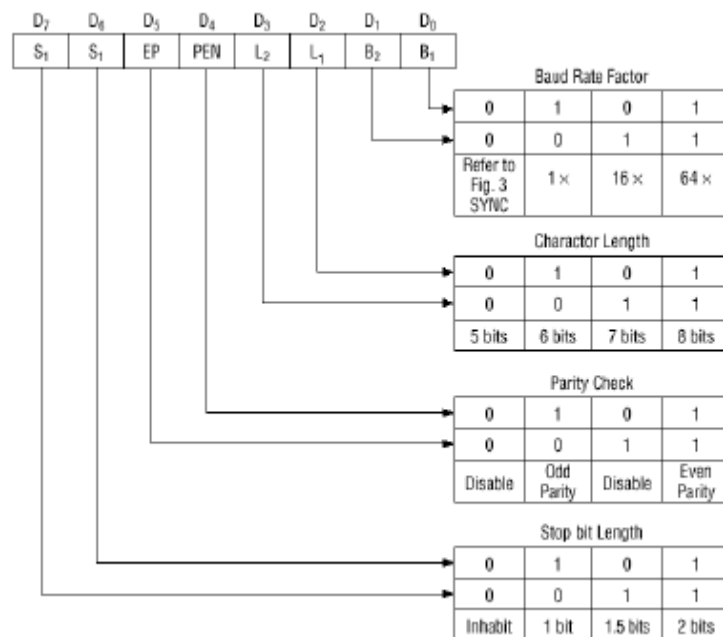


### 1) Mode Instruction

Mode instruction is used for setting the function of the 8251. Mode instruction will be in "wait for write" at either internal reset or external reset. That is, the writing of a control word after resetting will be recognized as a "mode instruction." Items set by mode instruction are as follows:

- Synchronous/asynchronous mode
- Stop bit length (asynchronous mode)
- Character length
- Parity bit
- Baud rate factor (asynchronous mode)
- Internal/external synchronization (synchronous mode)
- Number of synchronous characters (Synchronous mode)

The bit configuration of mode instruction is shown in Figure. In the case of synchronous mode, it is necessary to write one-or two byte sync characters. If sync characters were written, a function will be set because the writing of sync characters constitutes part of mode instruction.



**Video Content/Details of website for further learning (if any):**

<https://www.youtube.com/watch?v=hfj1ZB2bURM>

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 202-208)





# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L 35

## LECTURE HANDOUTS

EEE

III/IV

**Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS**

**Course Faculty : Ms.R.PUNITHA**

**Unit : IV- PROGRAMMING AND INTERFACING OF 8085&8051**

**Date of Lecture:**

**Topic of Lecture:** Keyboard/Display Controller

### **Introduction :**

- 8279 programmable keyboard/display controller is designed by Intel that interfaces a keyboard with the CPU.
- The keyboard first scans the keyboard and identifies if any key has been pressed. It then sends their relative response of the pressed key to the CPU and vice-a-versa.

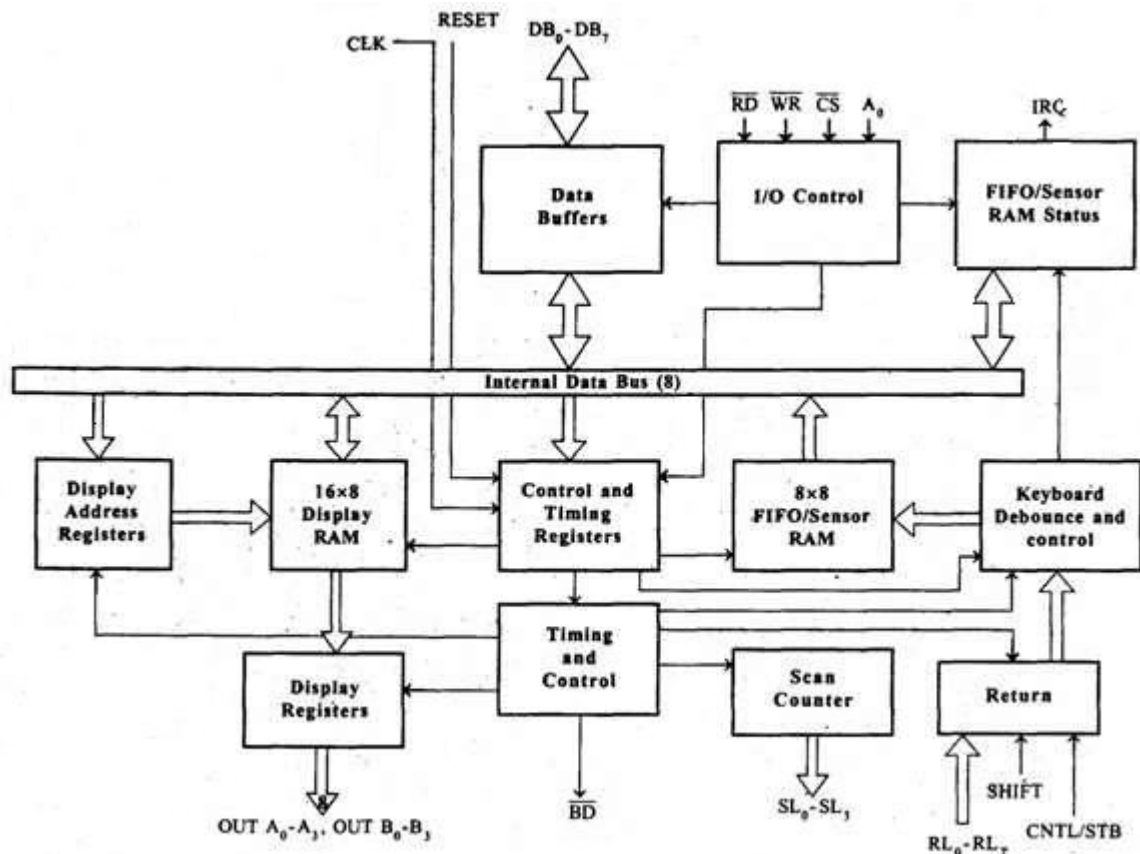
### **Prerequisite knowledge for Complete understanding and learning of Topic:**

- Basic operation on keyboard

### **Detailed content of the Lecture:**

- The Keyboard can be interfaced either in the interrupt or the polled mode. In the Interrupt mode, the processor is requested service only if any key is pressed, otherwise the CPU will continue with its main task.
- In the Polled mode, the CPU periodically reads an internal flag of 8279 to check whether any key is pressed or not with key pressure.
- The keyboard consists of maximum 64 keys, which are interfaced with the CPU by using the key-codes. These key-codes are de-bounced and stored in an 8-byte FIFORAM, which can be accessed by the CPU. If more than 8 characters are entered in the FIFO, then it means more than eight keys are pressed at a time. This is when the overrun status is set.
- If a FIFO contains a valid key entry, then the CPU is interrupted in an interrupt mode else the CPU checks the status in polling to read the entry. Once the CPU reads a key entry, then FIFO is updated, and the key entry is pushed out of the FIFO to generate space for new entries.

## Architecture



### I/O Control and Data Buffer

This unit controls the flow of data through the microprocessor. It is enabled only when D is low. Its data buffer interfaces the external bus of the system with the internal bus of the microprocessor. The pins A0, RD, and WR are used for command, status or data read/write operations.

### Control and Timing Register and Timing Control

This unit contains registers to store the keyboard, display modes, and other operations as programmed by the CPU. The timing and control unit handles the timings for the operation of the circuit.

### Scan Counter

It has two modes i.e. Encoded mode and Decoded mode. In the encoded mode, the counter provides the binary count that is to be externally decoded to provide the scan lines for the keyboard and display.

In the decoded scan mode, the counter internally decodes the least significant 2 bits and provides a decoded 1 out of 4 scan on SL<sub>0</sub>-SL<sub>3</sub>.

### Operational Modes

There are two modes of operation on 8279 – Input Mode and Output Mode.

### Input Mode

**Scanned Keyboard Mode** – In this mode, the key matrix can be interfaced using either encoded or decoded scans. In the encoded scan, an 8×8 keyboard or in the decoded scan, a 4×8 keyboard can be interfaced. The code of key pressed with SHIFT and CONTROL status is stored into the FIFO RAM.

### **Output Mode**

- **Display Scan** – This mode allows 8/16 character multiplexed displays to be organized as dual 4-bit/single 8-bit display units.
- **Display Entry** – This mode allows the data to be entered for display either from the right side/left side.

### **Video Content / Details of website for further learning:**

[https://www.youtube.com/watch?v=nAsnr\\_uG2mI](https://www.youtube.com/watch?v=nAsnr_uG2mI)

### **Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 431-438)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L 36

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : IV- PROGRAMMING AND INTERFACING OF 8085&8051

Date of Lecture:

**Topic of Lecture:** Keyboard/Display Controller

### Introduction :

- 8279 programmable keyboard/display controller is designed by Intel that interfaces a keyboard with the CPU.
- The keyboard first scans the keyboard and identifies if any key has been pressed. It then sends their relative response of the pressed key to the CPU and vice-a-versa.

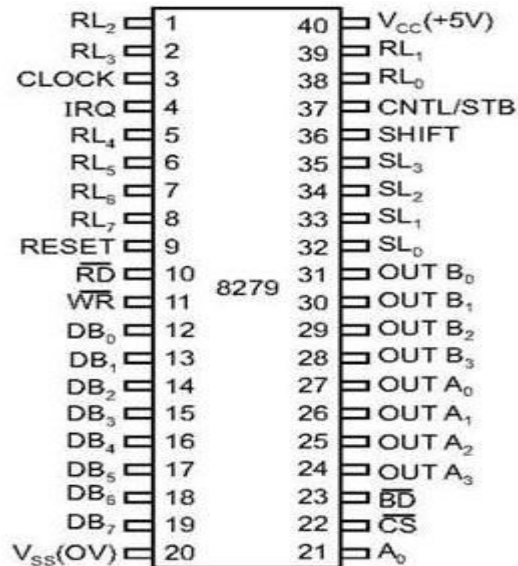
### Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic operation on keyboard

### Detailed content of the Lecture:

- Intel 8279 is a general purpose keyboard display controller that simultaneously drives the display of a system and interface a keyboard with the CPU
- The keyboard display interface scan the keyboard to identify the if any key has been pressed and send the code of the pressed key to the CPU
- It also transmits the data received from the CPU to the display device
- The keyboard is interfaced either in the interrupt or polled method
- In the interrupt mode the processor is requested service only if any is key Pressed otherwise the CPU can proceed
- In the polled mode the CPU periodically read an internal flag of 8279 to check for a key pressure

### Pin Description



### I/O control and Data bus

The I/O control section controls the flow of data to the 8279. The data bus buffer interface the external bus of the system with internal bus of 8279. The I/O is enabled only if CS is low. The pin A0, RD and WR select the command status or read / write operation carried out by the CPU with 8279.

### DB0-DB7

These are bi directional data bus lines. The data and command word to the CPU are transferred.

### CS

Chip select. A low on this line enables 8279 for normal read or write operation.

### A0

A high on this line indicates the transfer of a command or status information.

A low on this line indicates the transfer of data.

This is used to select one of the internal registers of 8279.

### RD,WR

Input / output, Read / Write. The input pins enable the data buffer to receive or send data over the data bus.

## **Control and Timing Register**

The register store the keyboard and display mode and other operating conditions performed by CPU

The registers are A0=1 and WR=0

### **CLK**

This is a clock input used to generate internal timing required by 8279

### **RESET**

This pin is used to reset 8279. A high on this line reset 8279

### **BD**

The output pin is used to blank the display during digital switching

### **Scan counter**

There are two ways in which scan line can be interfaced to the keyboard

The scan counter has two modes to scan key matrix and refresh the display

Encoded mode

Decoded mode

### **Video Content / Details of website for further learning:**

[https://www.youtube.com/watch?v=nAsnr\\_uG2mI](https://www.youtube.com/watch?v=nAsnr_uG2mI)

### **Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 431-438)





# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L37

EEE

III / V

**Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS**

**Course Faculty : Ms.R.PUNITHA**

**Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLERS**

**Date of Lecture:**

**Topic of Lecture:** Keyboard Interfacing

**Introduction :**

Keypads are a part of HMI or Human Machine Interface and play really important role in a small embedded system where human interaction or human input is needed. Matrix keypads are well known for their simple architecture and ease of interfacing with any microcontroller.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

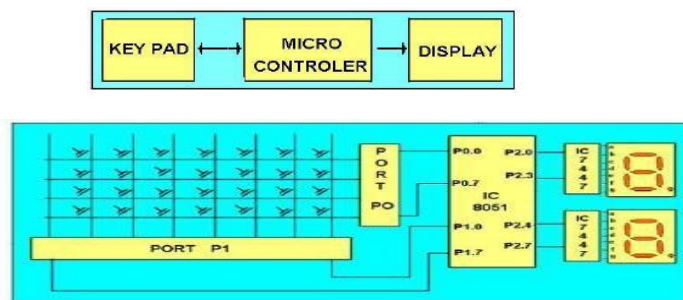
- Interfacing functions

**Detailed content of the Lecture: Keyboard Interfacing**

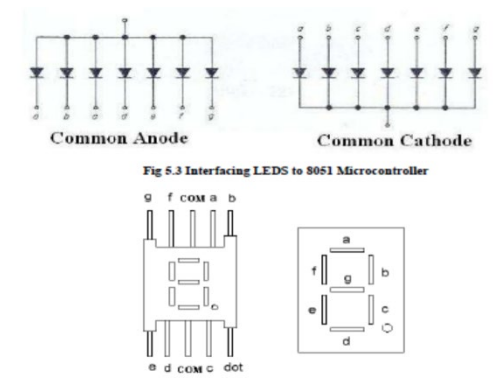
- The key board here we are interfacing is a matrix keyboard. This key board is designed with a particular rows and columns.
- These rows and columns are connected to the microcontroller through its ports of the micro controller 8051. We normally use 8\*8 matrix key board.
- So only two ports of 8051 can be easily connected to the rows and columns of the key board. Whenever a key is pressed, a row and a column gets shorted through that pressed key and all the other keys are left open.
- When a key is pressed only a bit in the port goes high. Which indicates microcontroller that the key is pressed. By this high on the bit key in the corresponding column is identified.
- Once we are sure that one of key in the key board is pressed next our aim is to identify that key. To do this we firstly check for particular row and then we check the corresponding column the key board.



- To check the row of the pressed key in the keyboard, one of the row is made high by making one of bit in the output port of 8051 high .
- This is done until the row is found out. Once we get the row next job is to find out the column of the pressed key.
- The column is detected by contents in the input ports with the help of a counter. The content of the input port is rotated with carry until the carry bit is set.
- The contents of the counter is then compared and displayed in the display. This display is designed using a seven segment display and a BCD to seven segment decoder IC 7447.
- The BCD equivalent number of counter is sent through output part of 8051 displays the number of pressed key.



### SEVEN SEGMENT DISPLAY



**Video Content / Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1)

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (R1-363-370)

Course Faculty



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L38

## LECTURE HANDOUTS

EEE

III / V

Course Name with Code: 16EEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLERS

Date of Lecture:

**Topic of Lecture:** LCD Interfacing

### Introduction :

- LCD is one of the most used display unit.
- Interfacing LCD with 8051 microcontroller might look quite complex to newbie, but after understanding the concept it would look very simple and easy.
- Although it may be time taking because you need to understand and connect 16 pins of LCD to the microcontroller. So first let's understand the 16 pins of LCD module.
- Display units are the most important output devices in embedded projects and electronics products..

### Prerequisite knowledge for Complete understanding and learning of Topic:

- 8085 & 8051 Features

### Detailed content of the Lecture:

- LCD display is an inevitable part in almost all embedded projects and this article is about interfacing a 16×2 LCD with 8051 microcontroller.
- Many guys find it hard to interface LCD module with the 8051 but the fact is that if you learn it properly, its a very easy job and by knowing it you can easily design embedded projects like digital voltmeter / ammeter, digital clock, home automation displays, status indicator display, digital code locks, digital speedometer/ odometer, display for music players etc etc.
- Thoroughly going through this article will make you able to display any text (including the extended characters) on any part of the 16×2 display screen. In order to understand the interfacing first you have to know about the 16×2 LCD module.

We can divide it in five categories, Power Pins, contrast pin, Control Pins, Data pins and Backlight pins.

Category	Pin NO.	Pin Name	Function
Power Pins	1	VSS	Ground Pin, connected to Ground
	2	VDD or Vcc	Voltage Pin +5V
Contrast Pin	3	V0 or VEE	Contrast Setting, connected to Vcc thorough a variable resistor.
Control Pins	4	RS	Register Select Pin, RS=0 Command mode, RS=1 Data mode
	5	RW	Read/ Write pin, RW=0 Write mode, RW=1 Read mode
	6	E	Enable, a high to low pulse need to enable the LCD
Data Pins	7-14	D0-D7	Data Pins, Stores the Data to be displayed on LCD or the command instructions
Backlight Pins	15	LED+ or A	To power the Backlight +5V
	16	LED- or K	Backlight Ground

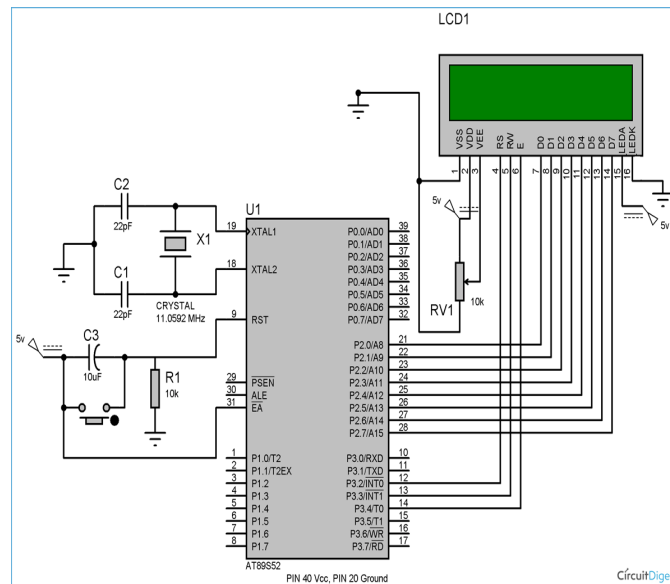
All the pins are clearly understandable by their name and functions, except the control pins, so they are explained below:

**RS:** RS is the register select pin. We need to set it to 1, if we are sending some data to be displayed on LCD. And we will set it to 0 if we are sending some command instruction like clear the screen (hex code 01).

**RW:** This is Read/write pin, we will set it to 0, if we are going to write some data on LCD. And set it to 1, if we are reading from LCD module. Generally this is set to 0, because we do not have need to read data from LCD. Only one instruction "Get LCD status", need to be read some times.

**E:** This pin is used to enable the module when a high to low pulse is given to it. A pulse of 450 ns should be given. That transition from HIGH to LOW makes the module ENABLE.

## Circuit Diagram and Explanation



- Circuit diagram for **LCD interfacing with 8051 microcontroller** is shown in the above figure. If you have basic understanding of 8051 then you must know about EA(PIN 31), XTAL1 & XTAL2, RST pin(PIN 9), Vcc and Ground Pin of 8051 microcontroller. I have used these Pins in above circuit. If you don't have any idea about that then I recommend you to read this Article LED Interfacing with 8051 Microcontroller before going through LCD interfacing.
- So besides these above pins we have connected the data pins (D0-D7) of LCD to the Port 2 (P2\_0 - P2\_7) microcontroller. And control pins RS, RW and E to the pin 12,13,14 (pin 2,3,4 of port 3) of microcontroller respectively.
- PIN 2(VDD) and PIN 15(Backlight supply) of LCD are connected to voltage (5v), and PIN 1 (VSS) and PIN 16(Backlight ground) are connected to ground.
- Pin 3(V0) is connected to voltage (Vcc) through a variable resistor of 10k to adjust the contrast of LCD. Middle leg of the variable resistor is connected to PIN 3 and other two legs are connected to voltage supply and Ground.

### Video Content / Details of website for further learning:

[www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1)

### Important Books/Journals for further learning including the page nos.:

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (R1-352-362)



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



## LECTURE HANDOUTS

L39

EEE

III / V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLERS

Date of Lecture:

**Topic of Lecture:** Stepper Motor Control

**Introduction :**

Stepper Motor Control using 8051 Microcontroller. A stepper motor is a brushless and synchronous motor which divides the complete rotation into number of steps. Each stepper motor will have some fixed step angle and motor rotates at this angle.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Stepper motor operation

**Detailed content of the Lecture: Stepper motor Interfacing/Control using 8085 and 8051**

**Stepper Motor**

A stepper motor is a device that translates electrical pulses into mechanical movement in steps of fixed step angle.

- The stepper motor rotates in steps in response to the applied signals.
- It is mainly used for position control.
- It is used in disk drives, dot matrix printers, plotters and robotics and process control circuits.

The complete board consists of transformer, control circuit, keypad and stepper motor as shown in snap.

The circuit has inbuilt 5 V power supply so when it is connected with transformer it will give the supply to circuit and motor both. The 8 Key keypad is connected with circuit through which user can give the command to control stepper motor. The control circuit includes micro controller 89C51, indicating LEDs, and current driver chip ULN2003A. One can program the controller to control the operation of stepper motor. He can give different commands through keypad like, run clockwise, run anticlockwise, increase/decrease RPM, increase/decrease revolutions, stop motor, change the mode, etc.

Unipolar stepper motor:- unipolar stepper motor has four coils. One end of each coil is tied together and it gives common terminal which is always connected with positive terminal of supply. The other ends of each coil are given for interface. Specific color code may also be given. Like in my motor orange is first coil (L1), brown is second (L2), yellow is third (L3), black is fourth (L4) and red for common terminal.

By means of controlling a stepper motor operation we can

1. Increase or decrease the RPM (speed) of it
2. Increase or decrease number of revolutions of it
3. Change its direction means rotate it clockwise or anticlockwise

To vary the RPM of motor we have to vary the PRF (Pulse Repetition Frequency). Number of applied pulses will vary number of rotations and last to change direction we have to change pulse sequence.

So all these three things just depends on applied pulses. Now there are three different modes to rotate this motor

1. Single coil excitation
2. Double coil excitation
3. Half step excitation

The table given below will give you the complete idea that how to give pulses in each mode

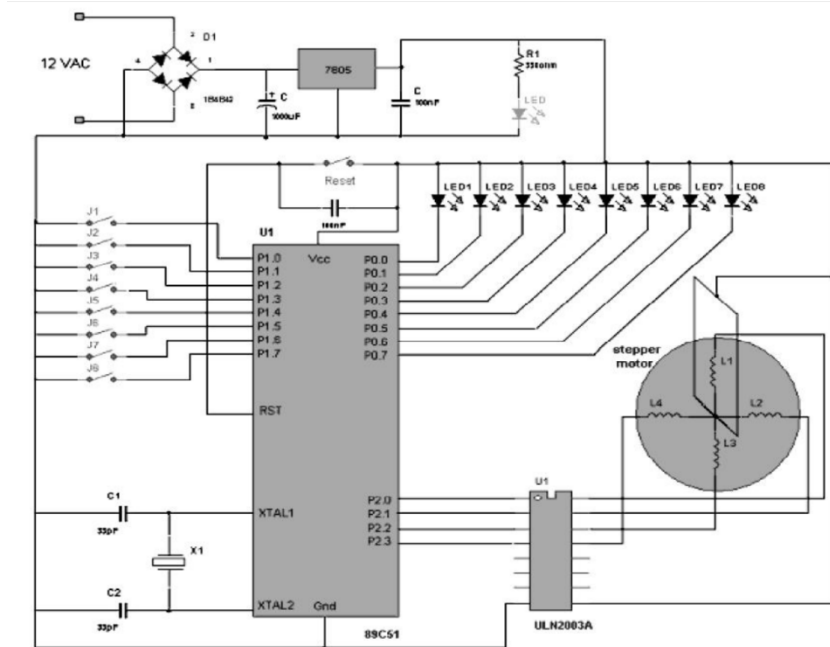
Single coil excitation		Double coil excitation		Half step excitation	
Clockwise	Anticlockwise	Clockwise	Anticlockwise	Clockwise	Anticlockwise
L4 L3 L2 L1	L4 L3 L2 L1	L4 L3 L2 L1	L4 L3 L2 L1	L4 L3 L2 L1	L4 L3 L2 L1
0 0 0 1	0 0 0 1	0 0 1 1	0 0 1 1	0001	0001
0 0 1 0	1 0 0 0	0 1 1 0	1 0 0 1	0011	0011
0 1 0 0	0 1 0 0	1 1 0 0	1 1 0 0	0010	1000
1 0 0 0	0 0 1 0	1 0 0 1	0 1 1 0	0110	1001
				0100	0100
				1100	1100
				1000	0010
				1001	0110

The circuit consists of very few components. The major components are 7805, 89C51 and ULN2003A.

**Connections:-**

1. The transformer terminals are given to bridge rectifier to generate rectified DC.
2. It is filtered and given to regulator IC 7805 to generate 5 V pure DC. LED indicates supply is ON.
3. All the push button micro switches J1 to J8 are connected with port P1 as shown to form serial keyboard.
4. 12 MHz crystal is connected to oscillator terminals of 89C51 with two biasing capacitors.
5. All the LEDs are connected to port P0 as shown
6. Port P2 drives stepper motor through current driver chip ULN2003A.
7. The common terminal of motor is connected to Vcc and rest all four terminals are

connected to port P2 pins in sequence through ULN chip.



**Video Content / Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1)

**Important Books/Journals for further learning including the page nos.:**

R.S. Gaonkar, 'Microprocessor Architecture Programming and Application', with 8085, Wiley Eastern Ltd., New Delhi, 2013 (R3 498-507)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



## LECTURE HANDOUTS

L40

EEE

III / V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLERS

Date of Lecture:

<b>Topic of Lecture: Stepper motor Control</b>	
<b>Introduction :</b>	
<ul style="list-style-type: none"><li>• stepper motor is a device that translates electrical pulses into mechanical movement in steps of fixed step angle.</li><li>• stepper motor rotates in steps in response to the applied signals.</li><li>• mainly used for position control.</li><li>• used in disk drives, dot matrix printers, plotters and robotics and process control circuits.</li></ul>	<p>A</p> <p>The</p> <p>It is</p> <p>It is</p>
<b>Prerequisite knowledge for Complete understanding and learning of Topic:</b>	
<ul style="list-style-type: none"><li>➤ Interfacing</li><li>➤ Motor Driver Circuit</li><li>➤ Stepper motor and its operation</li></ul>	
<b>Detailed content of the Lecture: Stepper motor Control</b>	
<ul style="list-style-type: none"><li>• Stepper motors have a permanent magnet called rotor (also called the shaft) surrounded by a stator. The most common stepper motors have four stator windings that are paired with a center-tap. This type of stepper motor is commonly referred to as a four-phase or unipolar stepper motor.</li><li>• The center tap allows a change of current direction in each of two coils when a winding is grounded, thereby resulting in a polarity change of the stator.</li></ul>	
<b>Interfacing</b>	
<ul style="list-style-type: none"><li>• Even a small stepper motor require a current of 400 mA for its operation.</li><li>• But the ports of the microcontroller cannot source this much amount of current. If such a motor is directly connected to the microprocessor/microcontroller ports, the motor may</li></ul>	



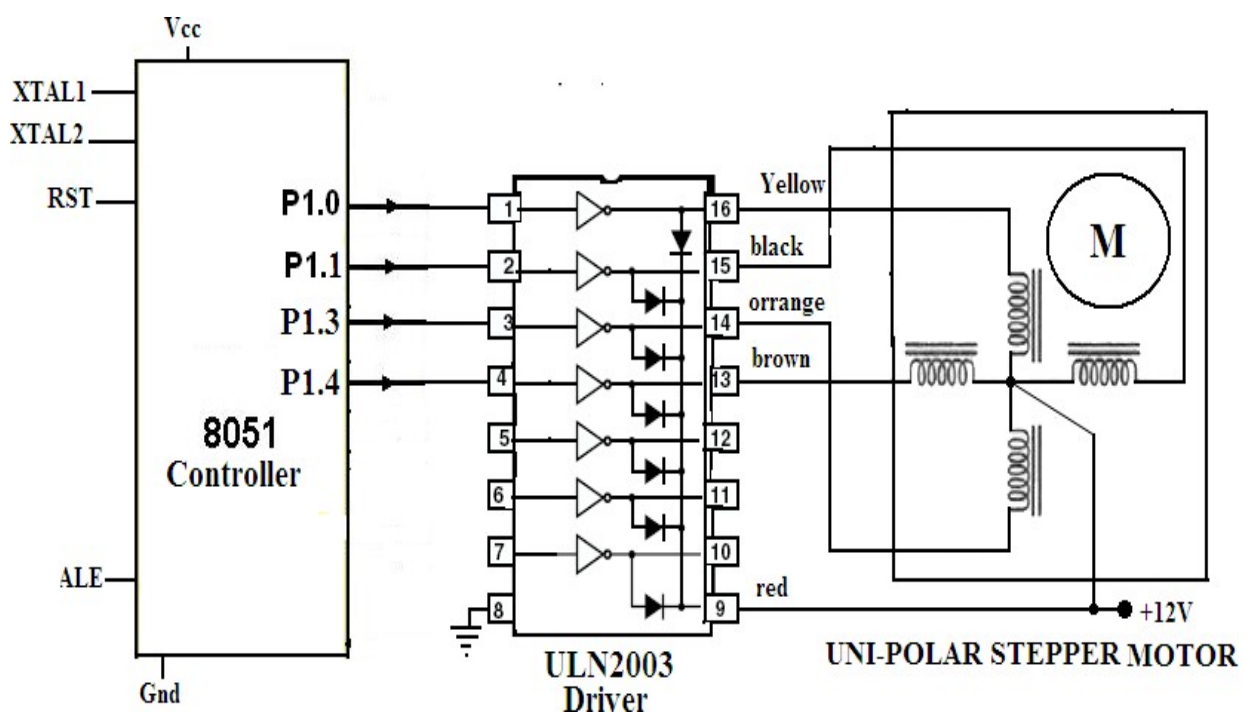
draw large current from the ports and damage it. So a suitable driver circuit is used with the microprocessor/microcontroller to operate the motor.

### Motor Driver Circuit (ULN2003)

- Stepper motor driver circuits are available readily in the form of ICs. ULN2003 is one such driver IC which is a High-Voltage High-Current Darlington transistor array and can give a current of 500mA. This current is sufficient to drive a small stepper motor. Internally, it has protection diodes used to protect the motor from damage due to back emf and large eddy currents. So, this ULN2003 is used as a driver to interface the stepper motor to the microcontroller.

### Operation

- The important parameter of a stepper motor is the step angle.
- It is the minimum angle through which the motor rotates in response to each excitation pulse. In a four phase motor if there are 200 steps in one complete rotation then then the step angle is  $360/200 = 1.8^\circ$ . So to rotate the stepper motor we have to apply the excitation pulse. For this the controller should send a hexa decimal code through one of its ports.
- The hex code mainly depends on the construction of the stepper motor. So, all the stepper motors do not have the same Hex code for their rotation. (refer the operation manual supplied by the manufacturer.) For example, let us consider the hex code for a stepper motor to rotate in clockwise direction is 77H , BBH , DDH and EEH. This hex code will be applied to the input terminals of the driver through the assembly language program



## ASSEMBLY LANGUAGE PROGRAM (8051)

```
Main : MOV A, # 0FF H      ; Initialization of Port 1 MOV
      P1, A                ;
      MOV A, #77 H        ; Code for the Phase 1
      MOV P1, A           ;
      ACALL DELAY        ; Delay subroutine
      MOV A, # BB H      ; Code for the Phase II
      MOV P1, A          ;
      ACALL DELAY        ; Delay subroutine.
      MOV A, # DD H      ; Code for the Phase III
      MOV P1, A          ;
      ACALL DELAY        ; Delay subroutine
      MOV A, # EE H      ; Code for the Phase 1
      MOV P1, A          ;
      ACALL DELAY        ; Delay subroutine SJMP
      MAIN; Keep the motor rotating continuously.
```

### DELAY Subroutine

```
      MOV R4, #0FF H      ; Load R4 with FF
      MOV R5, # 0FF      ; Load R5 with FF
LOOP1: DJNZ R4, LOOP1     ; Decrement R4 until zero,wait
LOOP2: DJNZ R5, LOOP2     ; Decrement R5 until zero,wait
      RET                 ; Return to main program .
```

### Video Content / Details of website for further learning (if any):

<https://www.zseries.in> > 8085 microprocessor > stepper motor

### Important Books/Journals for further learning including the page nos.:

R.S. Gaonkar, 'Microprocessor Architecture Programming and Application', with 8085, Wiley Eastern Ltd., New Delhi, 2013 (R3 498-507)

Course Faculty



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L41

LECTURE HANDOUTS

EEE

III / V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLERS

Date of Lecture:

**Topic of Lecture: Washing Machine Control**

**Introduction :**

AT89S51 microcontroller is used to control the process of washing cycle and to drive the external output devices such as water inlet valve, wash motor, water drain valve. The control strategy program for AT89S51 microcontroller is implemented by using assembly language.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Basic operations of 8085 & 8051

**Detailed content of the Lecture: Washing Machine Control**

- Many washing m/c shell in the market has mechanical controlled sequence for activated the timer and the sequence back and forth for their motor; washing motor or spinning motor. Spinning motor control only has one direction only, and its simple could be changed to the discrete mechanical timer which sells on the market. But washing motor control has 2 direction
- for this purpose, it means to squeeze the clothes, it must go to forward and then reversed. The sequence is like this :
  - First, go to forward direction for about a few seconds
  - Than stop, while the chamber is still rotate
  - Second, go back to reverse direction for about a few seconds
  - Than stop, while the chamber is still rotate
  - And so on, back and forth, until the the timer elapsed

**SCHEMATIC**

- Timing sequence like the above description, can be implemented with many way, by using discrete electronic components, timer, using a program or a microcontroller or microprocessor, etc.
- Because I am learning the PIC microcontroller for right now, I will implement this function using this microcontroller, but for you who familiar with another kind of

microcontroller my adapted it to your purpose.

- By using PIC micro, it can be made more compact. First I plan to make 2 buttons,
- 1 for set the timer and another for reset the timer or for the emergency stop push button. Then to know the timer works or not, I need a visual display. For this purpose I will use
- 7-segmen display showing the rest of the timer. To run the motor sequence of course I need a pair of relays (power relays, about 3 Amperes output), one for forward and another for reverse option.
- I will use the very common family of PIC micro, ie : 16F84A, because this is the most popular type and very simples used and very much used. Also can be obtained easily in the market. But this is the medium type of PIC micro family.
- It has 1kByte of memory (EEPROM type) and 13 I/O pins. It can be reprogrammable thousands times. Because the I/O just only 13 pins, I used a BCD to 7-segmen chip. So it will left a few I/O pins for expanded in the future. You can omitted this chip for timing sequence purpose and save one IC price, because the I/O just exactly enough.

I/O port A-0 = SET push button

I/O port A-1 = RST push button

I/O port A-2 = Reserved

I/O port A-3 = Reserved

I/O port A-4 = Reserved

I/O port B-0 = Forward Relay (Run motor forward)

I/O port B-1 = Reverse Relay (Run motor reverse)

I/O port B-2 = Activated unit 7-segmen (multiplexed)

I/O port B-3 = Activated ten 7-segmen (multiplexed)

I/O port B-4 = BCD data A (for 7-segmen)

I/O port B-5 = BCD data B (for 7-segmen)

I/O port B-6 = BCD data C (for 7-segmen)

I/O port B-7 = BCD data D (for 7-segmen)

Also integrated power supply to run it modularly

### **Working of Washing Machine**

- The direction of rotation can be controlled When switchS1 is in position A, coil L1 of the motor receives the current directly, whereas coil L2 receives the current with a phase shift due to capacitor C. So the rotor rotates in clockwise direction (see Fig. 2(a)).
- When switch S1 is in position B, the reverse happens and the rotor rotates in anti-clockwise direction Thus switch S1 can change the rotation direction. The motor cannot be reversed instantly.
- It needs a brief pause between switching directions, or else it may get damaged. For this purpose, another spin direction control timer (IC2) is employed. It is realised with an IC 555. This timer gives an alternate „on“ and „off“ time duration of 10 seconds and 3 seconds, respectively.

- So after every 10 seconds of running (either in clockwise or anticlockwise direction), the motor stops for a brief duration of 3 seconds. The values of R3 and R4 are calculated accordingly. The master timer is realised with monostable IC555 (IC1) and its „on“ time is decided by the resistance of 1-megaohm potmeter VR. A 47-kilo-ohm resistor is added in series so that even when the VR knob in zero resistance position ,the net series resistance is not zero.

**Video Content / Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1)

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 465-472)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L42

LECTURE HANDOUTS

EEE

III / V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLERS

Date of Lecture:

**Topic of Lecture: Washing Machine Control**

**Introduction :**

AT89S51 microcontroller is used to control the process of washing cycle and to drive the external output devices such as water inlet valve, wash motor, water drain valve. The control strategy program for AT89S51 microcontroller is implemented by using assembly language.

**Prerequisite knowledge for Complete understanding and learning of Topic:**

- Basic operations of 8085 & 8051

**Detailed content of the Lecture: Washing Machine Control**

- Many washing m/c shell in the market has mechanical controlled sequence for activated the timer and the sequence back and forth for their motor; washing motor or spinning motor. Spinning motor control only has one direction only, and its simple could be changed to the discrete mechanical timer which sells on the market. But washing motor control has 2 direction
- for this purpose, it means to squeeze the clothes, it must go to forward and then reversed. The sequence is like this :
  - First, go to forward direction for about a few seconds
  - Than stop, while the chamber is still rotate
  - Second, go back to reverse direction for about a few seconds
  - Than stop, while the chamber is still rotate
  - And so on, back and forth, until the the timer elapsed

**SCHEMATIC**

- Timing sequence like the above description, can be implemented with many way, by using discrete electronic components, timer, using a program or a microcontroller or microprocessor, etc.
- Because I am learning the PIC microcontroller for right now, I will implement this function using this microcontroller, but for you who familiar with another kind of

microcontroller my adapted it to your purpose.

- By using PIC micro, it can be made more compact. First I plan to make 2 buttons,
- 1 for set the timer and another for reset the timer or for the emergency stop push button. Then to know the timer works or not, I need a visual display. For this purpose I will use
- 7-segmen display showing the rest of the timer. To run the motor sequence of course I need a pair of relays (power relays, about 3 Amperes output), one for forward and another for reverse option.
- I will use the very common family of PIC micro, ie : 16F84A, because this is the most popular type and very simples used and very much used. Also can be obtained easily in the market. But this is the medium type of PIC micro family.
- It has 1kByte of memory (EEPROM type) and 13 I/O pins. It can be reprogrammable thousands times. Because the I/O just only 13 pins, I used a BCD to 7-segmen chip. So it will left a few I/O pins for expanded in the future. You can omitted this chip for timing sequence purpose and save one IC price, because the I/O just exactly enough.

I/O port A-0 = SET push button

I/O port A-1 = RST push button

I/O port A-2 = Reserved

I/O port A-3 = Reserved

I/O port A-4 = Reserved

I/O port B-0 = Forward Relay (Run motor forward)

I/O port B-1 = Reverse Relay (Run motor reverse)

I/O port B-2 = Activated unit 7-segmen (multiplexed)

I/O port B-3 = Activated ten 7-segmen (multiplexed)

I/O port B-4 = BCD data A (for 7-segmen)

I/O port B-5 = BCD data B (for 7-segmen)

I/O port B-6 = BCD data C (for 7-segmen)

I/O port B-7 = BCD data D (for 7-segmen)

Also integrated power supply to run it modularly

### **Working of Washing Machine**

- The direction of rotation can be controlled When switchS1 is in position A, coil L1 of the motor receives the current directly, whereas coil L2 receives the current with a phase shift due to capacitor C. So the rotor rotates in clockwise direction (see Fig. 2(a)).
- When switch S1 is in position B, the reverse happens and the rotor rotates in anti-clockwise direction Thus switch S1 can change the rotation direction. The motor cannot be reversed instantly.
- It needs a brief pause between switching directions, or else it may get damaged. For this purpose, another spin direction control timer (IC2) is employed. It is realised with an IC 555. This timer gives an alternate „on“ and „off“ time duration of 10 seconds and 3 seconds, respectively.

- So after every 10 seconds of running (either in clockwise or anticlockwise direction), the motor stops for a brief duration of 3 seconds. The values of R3 and R4 are calculated accordingly. The master timer is realised with monostable IC555 (IC1) and its „on“ time is decided by the resistance of 1-megaohm potmeter VR. A 47-kilo-ohm resistor is added in series so that even when the VR knob in zero resistance position ,the net series resistance is not zero.

**Video Content / Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1)

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 465-472)

**Course Faculty**





# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



## LECTURE HANDOUTS

L 43

EEE

III/V

Course Name with Code : 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLER

Date of Lecture:

### Topic of Lecture: LED control

#### Introduction :

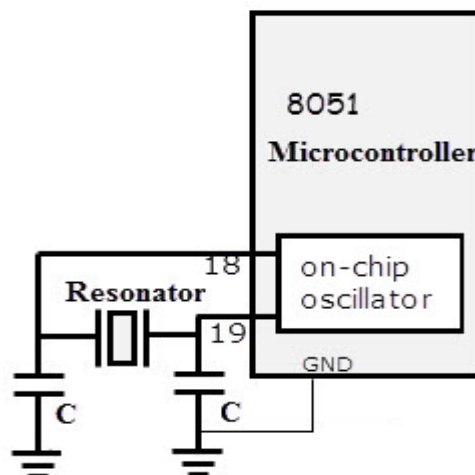
- The main principle of this circuit is to interface **LEDs** to the **8051** family micro **controller**. Commonly, used **LEDs** will have voltage drop of 1.7v and current of 10mA to glow at full intensity. This is applied through the output pin of the micro **controller**.

#### Prerequisite knowledge for Complete understanding and learning of Topic:

- LED
- Interfacing of 8051

#### Detailed content of the Lecture: LED control

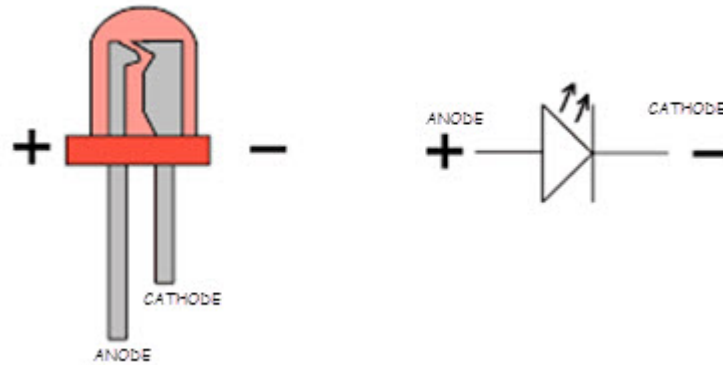
8051 has an on-chip oscillator, but it requires an external clock to run it. A quartz crystal is connected in between the XTAL pins of the MC. This crystal needs two same value capacitors (33pF) for generating a clock signal of the desired frequency. Features of 8051 Microcontroller have explained in our previous article.



## LED (Light Emitting Diode)

LED is a semiconductor device used in many electronic devices, mostly used for signal transmission /power indication purposes. It is very cheaply and easily available in a variety of shape, color, and size. The LEDs are also used for design message display boards and traffic control signal lights etc.

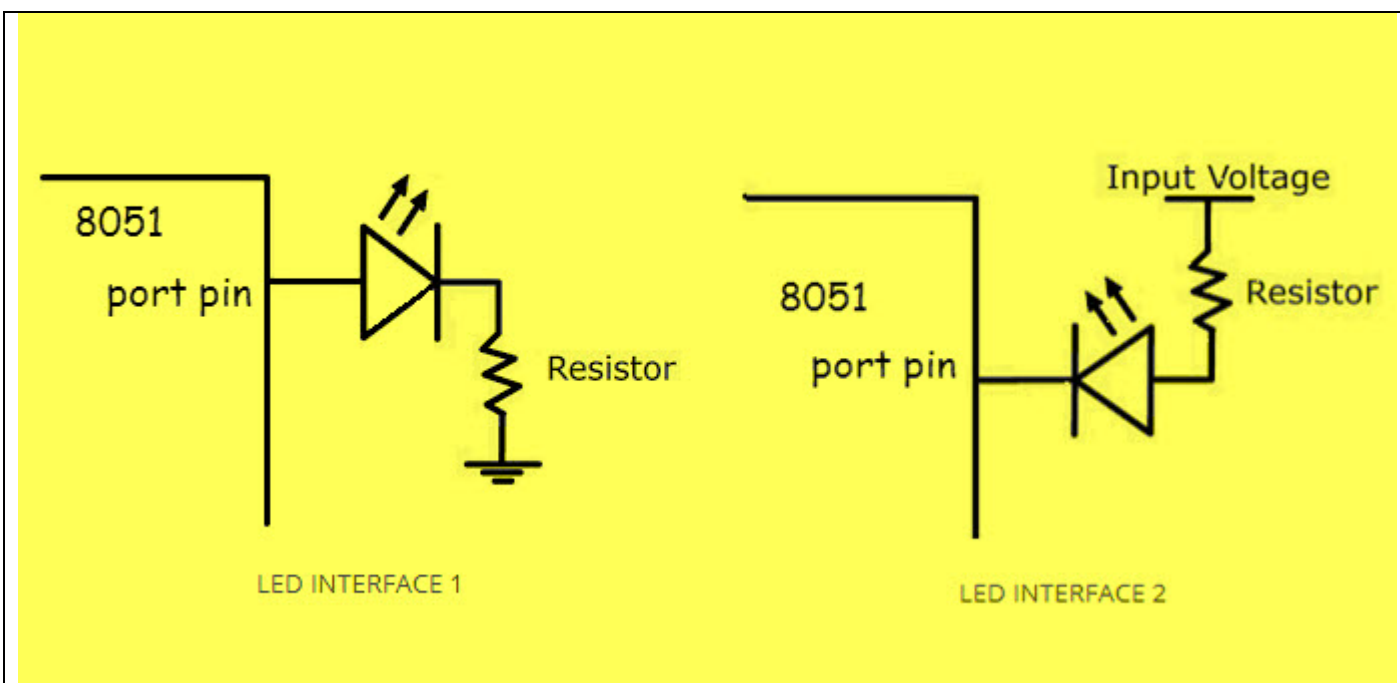
It has two terminals positive and negative as shown in the figure.



The only way to know polarity is either to test it with a multimeter or by carefully observing inside the LED. The larger end inside the led is -ve (cathode) and the shorter one is +ve (anode), that is how we find out the polarity of the LED. Another way to recognize the polarity is, connecting leads, POSITIVE terminal has more length than NEGATIVE terminal.

## LED Interfacing to 8051

There are two ways which we can interface LED to the Microcontroller 8051. But the connections and programming techniques will be different. This article provides the information on LED interfacing with 8051 and LED blinking code for AT89C52/ AT89C51 Microcontroller.



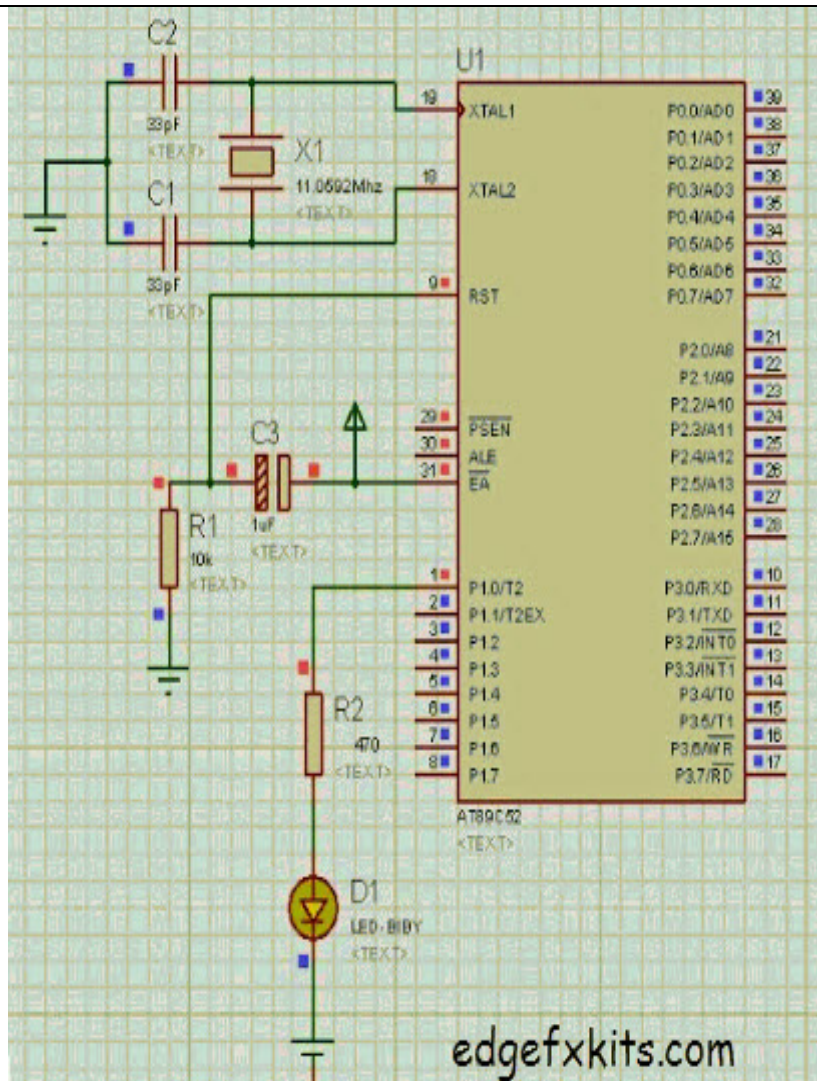
### Interfacing LED to 8051 Methods

Observe carefully the interface LED 2 is in forward biased because the input voltage of 5v connected to the positive terminal of the LED, So here the Microcontroller pin should be at LOW level. And vice versa with the interface 1 connections.

The resistor is important in LED interfacing to limit the flowing current and avoid damaging the LED and/or MCU.

- Interface 1 will glow LED, only if the PIN value of the MC is HIGH as current flows towards the ground.
- Interface 2 will glow LED, only if the PIN value of the MC is LOW as current flows towards PIN due to its lower potential.

The circuit diagram is shown in below. An LED is connected to the pin-0 of port-1.



**Video Content / Details of website for further learning (if any):**

<https://www.electronicshub.org/led-interfacing-8051>

**Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 476-481)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L 44

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLERS

Date of Lecture:

Topic of Lecture: Servo Motor Control

### Introduction :

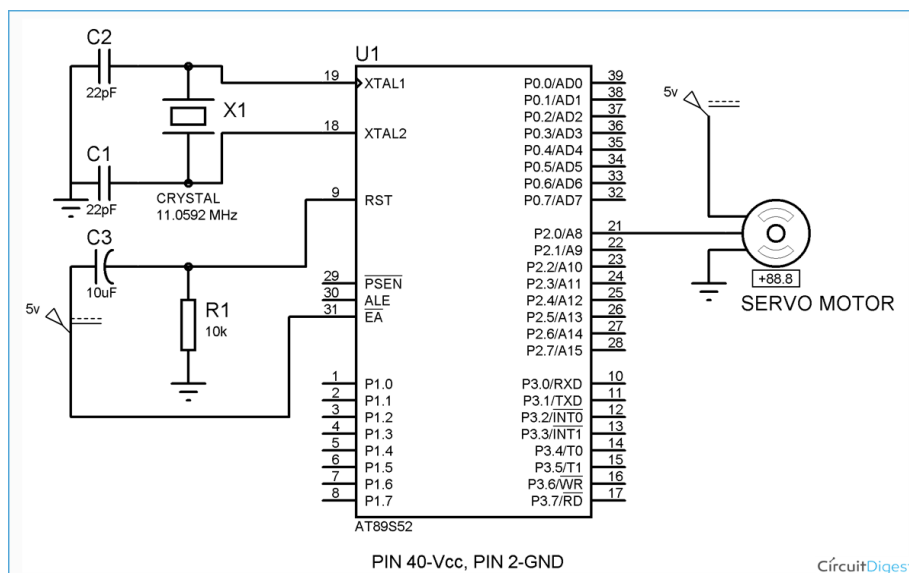
Servo motors are very useful in electronics and embedded systems. You can find the use of Servo motor everywhere around you, they are used in toys, robots, CD tray of computer, cars, aeroplane etc. The reason of this wide scope is that, servo motor is very reliable and precise.

We can rotate it to any particular angle. They are available in wide range, from high torque motor to low torque motors. In this tutorial we are going to interface a servo motor to 8051 microcontroller (AT89S52).

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Servo motor working Principle

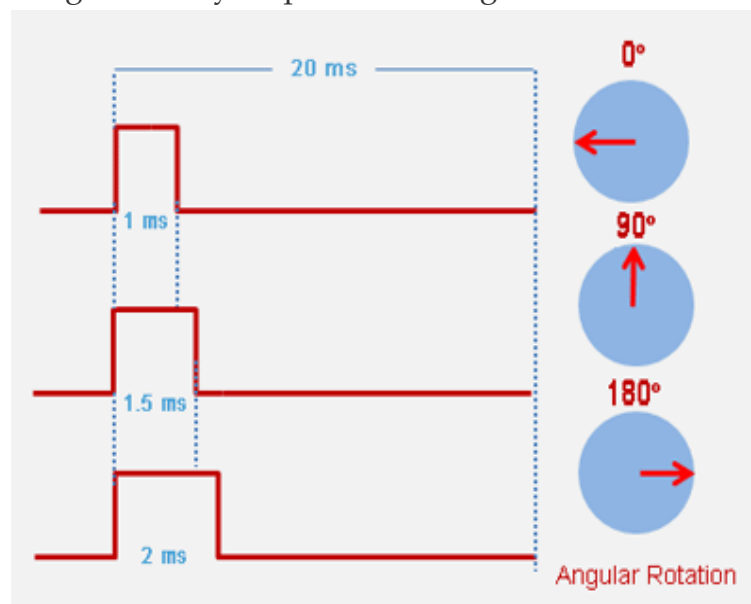
### Detailed content of the Lecture: Servo Motor Control



- Servos are DC motors with built in gearing and feedback control loop circuitry. And no

motor drivers required.

- They are extremely popular with robot, RC plane, and RC boat builders.
- Most servo motors can rotate about 90 to 180 degrees. Some rotate through a full 360degreesormore.
- However, servos are unable to continually rotate, meaning they can't be used for driving wheels, unless they are modified (how to modify), but their precision positioning makes them ideal for robot legs and arms, rack and pinion steering, and sensor scanners to name a few.
- Since servos are fully self contained, the velocity and angle control loops are very easy to implement, while prices remain very affordable. To use a servo, simply connect the black wire to ground, the red to a 4.8-6V source, and the yellow/white wire to a signal generator (such as from your microcontroller).
- Vary the square wave pulse width from 1-2 ms and your servo is now position/velocity controlled.
- Pulse width modulation (PWM) is a powerful technique for controlling analog circuits with a processor's digital outputs.
- PWM is employed in a wide variety of applications, ranging from measurement and communications to power control and conversion.
- The general concept is to simply send an ordinary logic square wave to your servo at a specific wave length, and your servo goes to a particular angle (or velocity if your servo is modified).The wavelength directly maps to servo angle.



- Servo motor interfacing with 8051 microcontroller, the black wire connected to the ground pin and the motor get the power from the red wire. The control of servo motor connected port0 of 8051 microcontroller.

- The 11.0592MHz crystal oscillator is used to provide the clock pulsed to the microcontroller and 22pf ceramic capacitors used to stabilize the operation of crystal. 10KΩ and 10uf capacitor is used to provide the power on reset to the microcontroller.

### **Servo Motor Working Principle**

- The servo motor working principle mainly depends upon the 'Fleming left hand rule'. Basically servo motors are adapted with DC motors, a position sensor, a Gear reduction, and an electronic circuit.
- The DC motors achieve powered from a battery and run at high speed and low torque. We assembled shaft and gear connected to DC motors then we can increase and decrease the motor speed gradually.

### **Servo Motor Applications**

- It is used in Press machines for cutting the pieces to size
- It is used in the Sugar filling station
- It is used in Labeling applications
- It is used Packing system with random timing function
- It is used Used in aero planes

### **Advantages of Servo Motor**

- If a motor gets heavy load the driver will increase the current to the motor coil as its efforts to rotate the motor. Mainly, there is no out-of-step condition.
- High-speed operation is possible by the servo motors.

### **Video Content / Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1)

### **Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 482-490)

**Course Faculty**



# MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



L 45

## LECTURE HANDOUTS

EEE

III/V

Course Name with Code: 16EEEC09/MICROPROCESSORS AND MICROCONTROLLERS

Course Faculty : Ms.R.PUNITHA

Unit : V - APPLICATIONS OF PROCESSORS AND CONTROLLERS

Date of Lecture:

Topic of Lecture: Servo Motor Control

### Introduction :

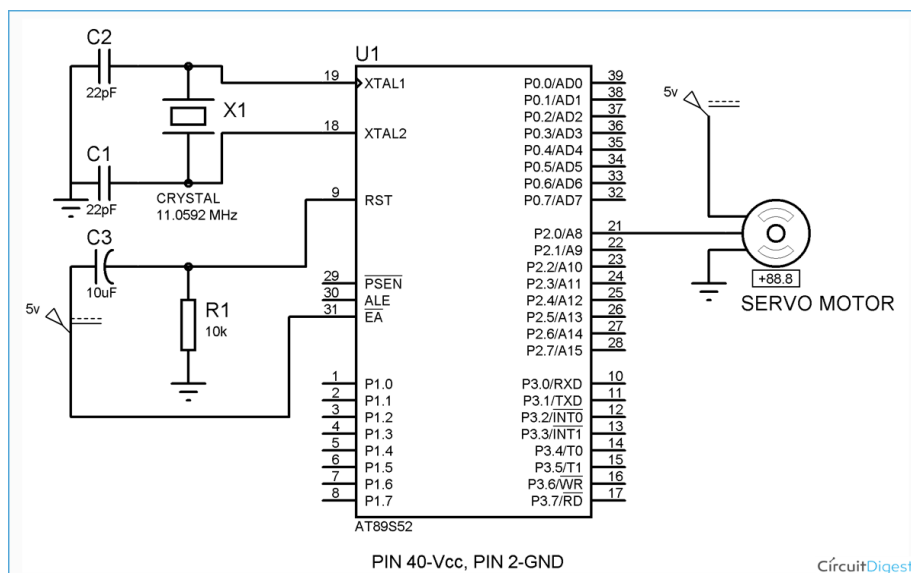
Servo motors are very useful in electronics and embedded systems. You can find the use of Servo motor everywhere around you, they are used in toys, robots, CD tray of computer, cars, aeroplane etc. The reason of this wide scope is that, servo motor is very reliable and precise.

We can rotate it to any particular angle. They are available in wide range, from high torque motor to low torque motors. In this tutorial we are going to interface a servo motor to 8051 microcontroller (AT89S52).

### Prerequisite knowledge for Complete understanding and learning of Topic:

- Servo motor working Principle

### Detailed content of the Lecture: Servo Motor Control

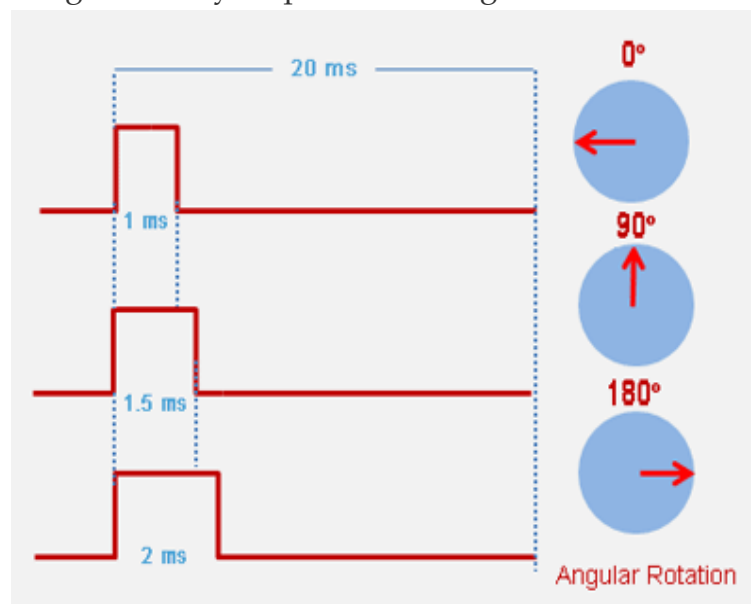


- Servos are DC motors with built in gearing and feedback control loop circuitry. And no



motor drivers required.

- They are extremely popular with robot, RC plane, and RC boat builders.
- Most servo motors can rotate about 90 to 180 degrees. Some rotate through a full 360degreesormore.
- However, servos are unable to continually rotate, meaning they can't be used for driving wheels, unless they are modified (how to modify), but their precision positioning makes them ideal for robot legs and arms, rack and pinion steering, and sensor scanners to name a few.
- Since servos are fully self contained, the velocity and angle control loops are very easy to implement, while prices remain very affordable. To use a servo, simply connect the black wire to ground, the red to a 4.8-6V source, and the yellow/white wire to a signal generator (such as from your microcontroller).
- Vary the square wave pulse width from 1-2 ms and your servo is now position/velocity controlled.
- Pulse width modulation (PWM) is a powerful technique for controlling analog circuits with a processor's digital outputs.
- PWM is employed in a wide variety of applications, ranging from measurement and communications to power control and conversion.
- The general concept is to simply send an ordinary logic square wave to your servo at a specific wave length, and your servo goes to a particular angle (or velocity if your servo is modified).The wavelength directly maps to servo angle.



- Servo motor interfacing with 8051 microcontroller, the black wire connected to the ground pin and the motor get the power from the red wire. The control of servo motor connected port0 of 8051 microcontroller.

- The 11.0592MHz crystal oscillator is used to provide the clock pulsed to the microcontroller and 22pf ceramic capacitors used to stabilize the operation of crystal. 10KΩ and 10uf capacitor is used to provide the power on reset to the microcontroller.

### **Servo Motor Working Principle**

- The servo motor working principle mainly depends upon the 'Fleming left hand rule'. Basically servo motors are adapted with DC motors, a position sensor, a Gear reduction, and an electronic circuit.
- The DC motors achieve powered from a battery and run at high speed and low torque. We assembled shaft and gear connected to DC motors then we can increase and decrease the motor speed gradually.

### **Servo Motor Applications**

- It is used in Press machines for cutting the pieces to size
- It is used in the Sugar filling station
- It is used in Labeling applications
- It is used Packing system with random timing function
- It is used Used in aero planes

### **Advantages of Servo Motor**

- If a motor gets heavy load the driver will increase the current to the motor coil as its efforts to rotate the motor. Mainly, there is no out-of-step condition.
- High-speed operation is possible by the servo motors.

### **Video Content / Details of website for further learning (if any):**

[www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1](http://www.youtube.com/watch?v=liRtPtvj7bFU&noredirect=1)

### **Important Books/Journals for further learning including the page nos.:**

Muhammad Ali Mazidi & Janice Gilli Mazidi, R.D.Kinely, The 8051 Micro Controller and Embedded Systems, PHI Pearson Education, 5th Indian reprint, 2003 (Page No: 482-490)

**Course Faculty**