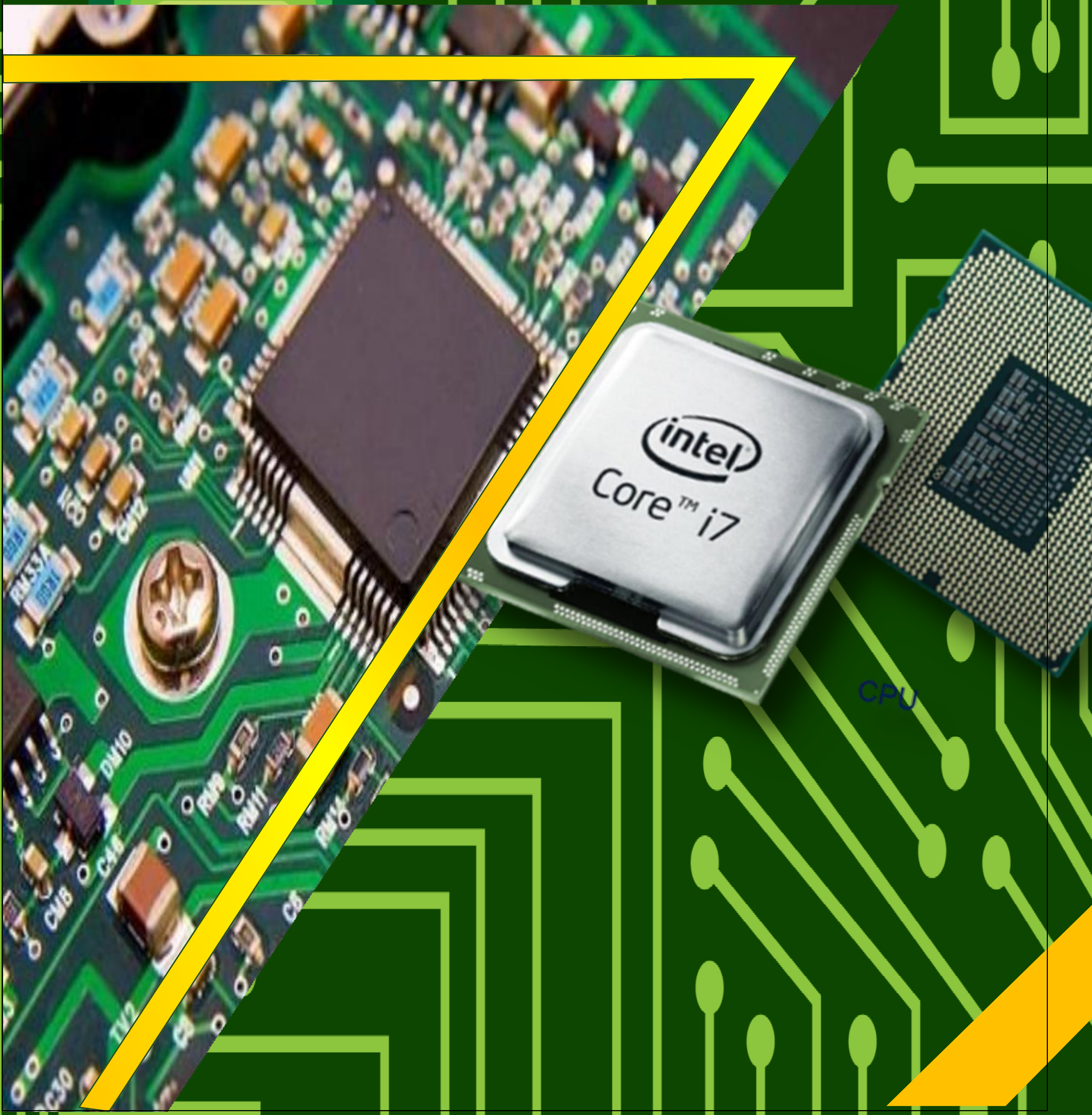
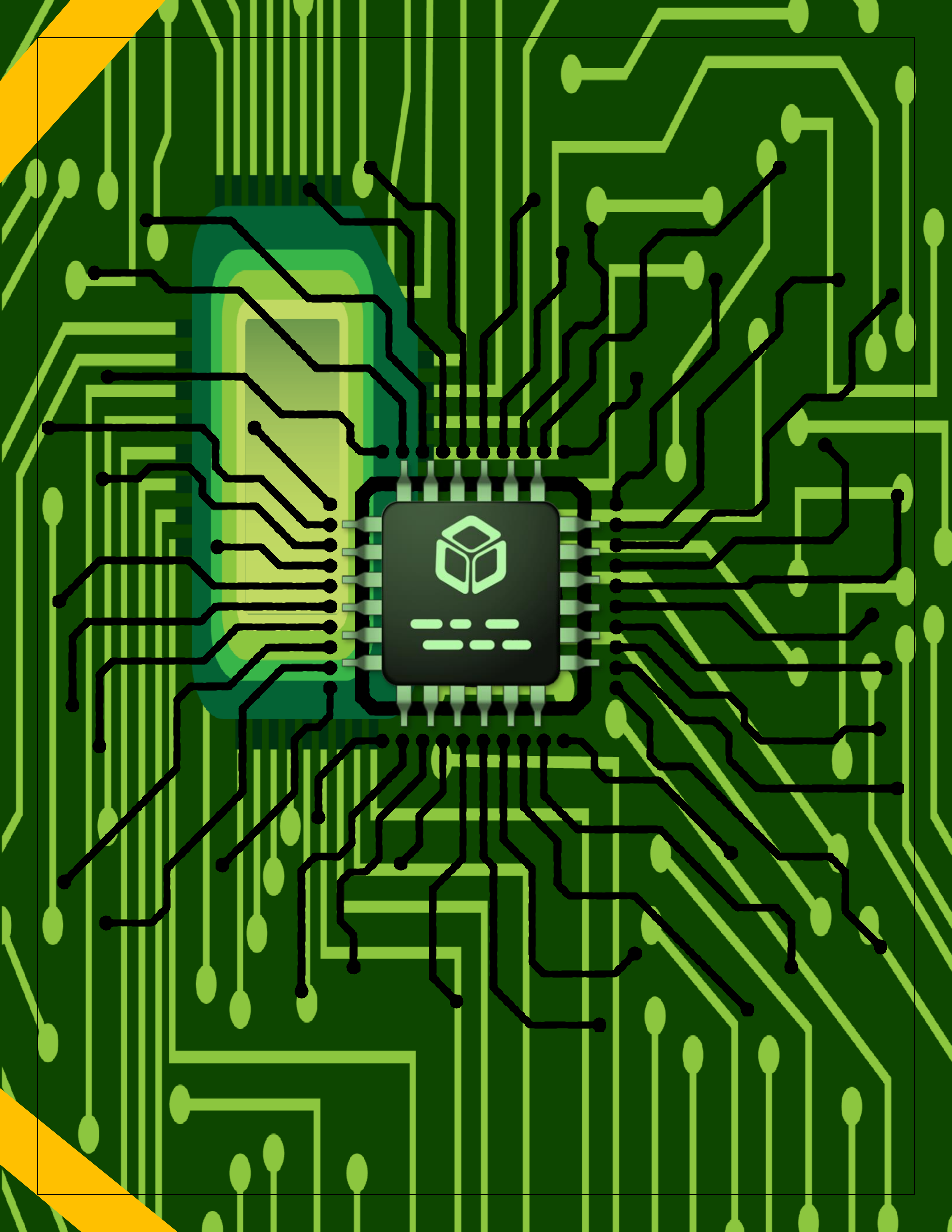
# Microprocessors &Microcontrollers





# TABLEOFCONTENT

## Contents:

**MODULE-1:8085MICROPROCESSORS**

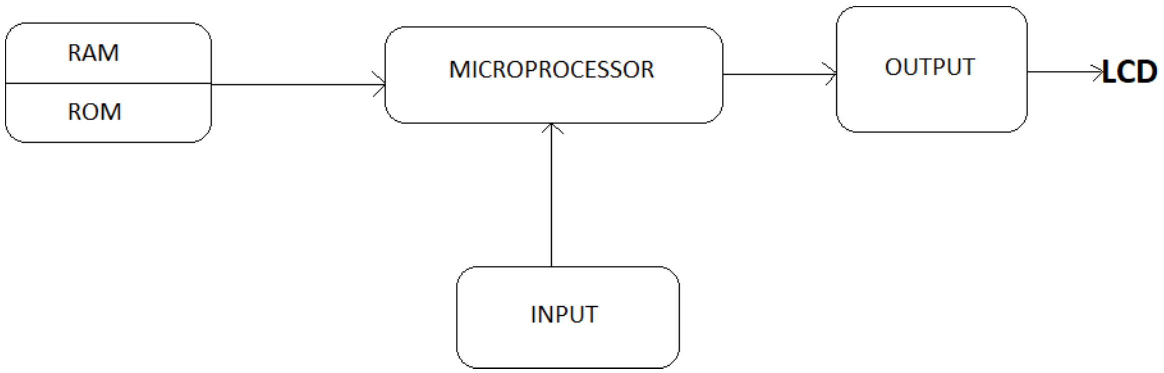
**MODULE-2:8086MICROPROCESSORS**

MODULE-3:INTERFACINGDEVICES

# TABLEOFCONTENT

# MODULE-1:8085MICROPROCESSORS

**Introduction:**Itismultipurpose,programmable,clockdriven,registerbased,electronicsdevice thattakesthedataintheformofbinary,processesandthe resultisdisplayedatoutputdevice or LCD display.



## Importantfeaturesof8085Microprocessor:

- The8085microprocessorisan8-bitprocessorthatmeansitsdatalineis8-bitand address line is 16-bit.
- Its internal clock frequency is up-to 5MHz so that the processor takes about 0.2μsec to complete an instruction.

## Pindiagramof8085 Microprocessor:

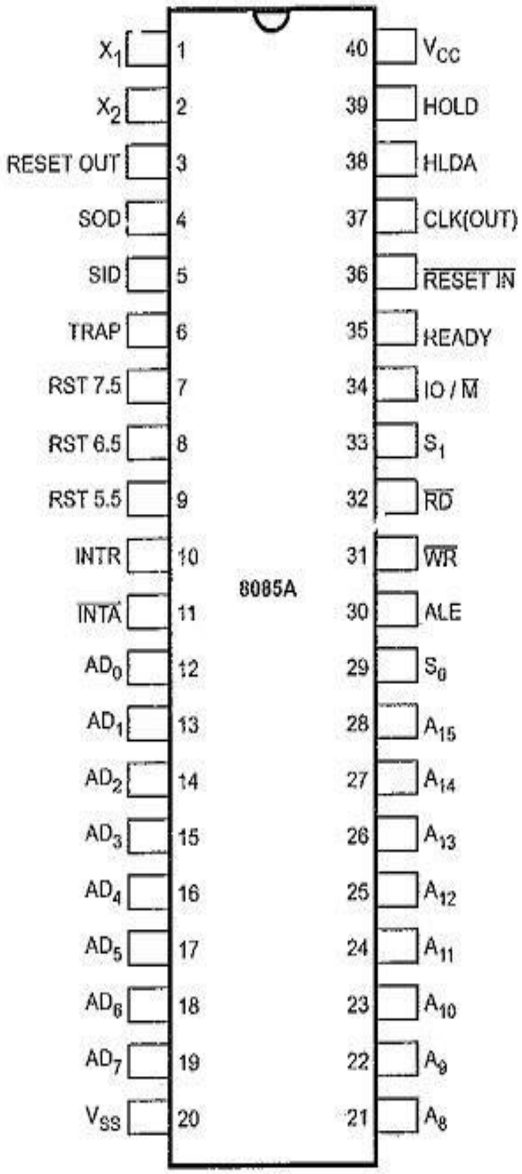


Fig. 1.3 (a) Pin configuration

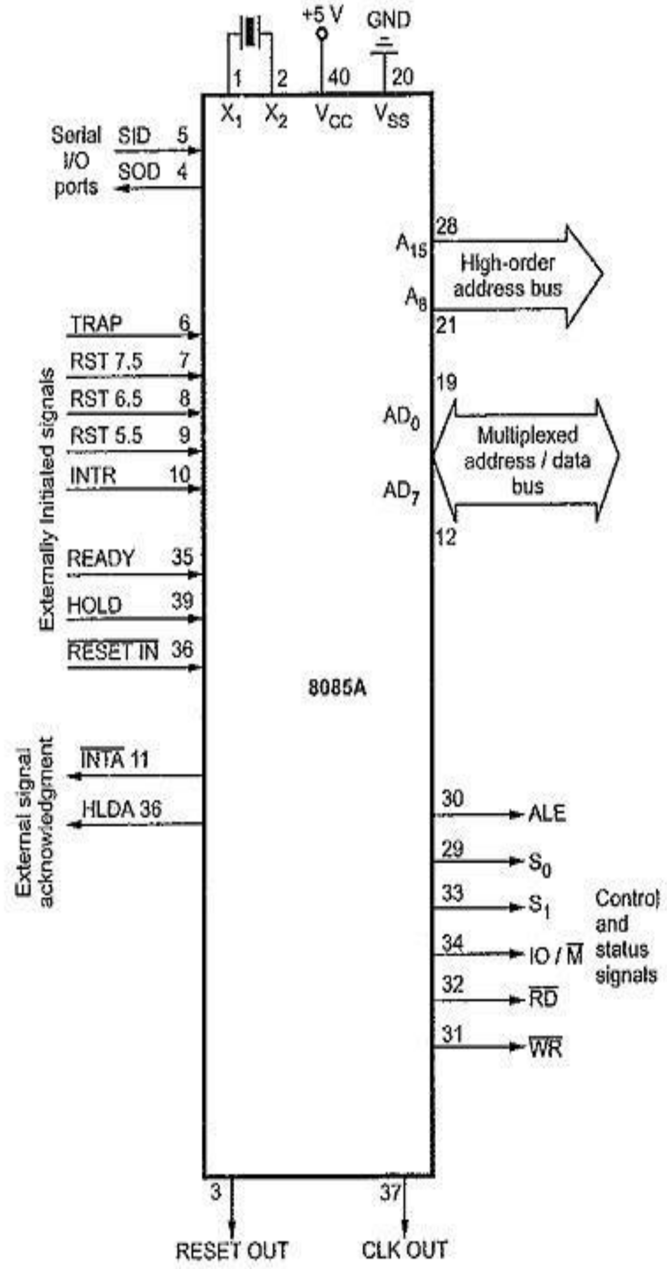


Fig. 1.3 (b) Functional pin diagram

## Pindescriptionfor8085Microprocessor:

Itconsistsof40 pinsandoperates at+5V DC.

**X1&X2**

- Pinno-1&2areusedasX1&X2respectively.

- These are two external pins which are connected to an external crystal oscillator which providesaninternalclockfrequencyup-to5MMHztotheprocessorsothattheprocessor takes about 0.2μsec to compute an instruction.

**RESETIN’&RESETOUT**

- Pinno-36&3areusedasRESETIN’&RESETOUT’ respectively.
- Thesepinsarebasicallyusedtorestarttheprocessor iftheprogramhangsinbetween.

**SOD**

- Pinno-4is usedasSerialOutputDatapin.
- ThroughthispinthedatafromtheprocessorissendtotheoutputdeviceorLCDscreen.

**SID**

- Pinno-5isusedas SerialInputDatapin.
- ThroughthispinwecantransferthedatafromIOdevicetotheprocessor.

**TRAP**

- Pinno-6isusedas TRAPpin.
- Itisanon-maskableinterruptthatmeansitsbit lineisalwayshighandtherearenologic gatesorflipfloptocontrolthisinterrupt.Sooncethisinterruptisactivatedtheprocessor willstopthecurrentprogramandjumptosomeotherprogramwhichisrequiredatthat specific moment of time and till the completion of that specific program the processor cannot return back to original program.

**RST7.5,RST6.5&RST5.5**

- Pinno-7, 8& 9areusedasRST7.5, RST6.5&RST5.5 respectively.
- Thesearethemaskableinterruptpins.
- Oncethisinterrupt isactivated theprocessorwill stop thecurrent program and jump to some other program which is required at that specific moment of time and till the completion of that specific program the processor return back to original program.

**INTR**

- Pinno-10isusedasInterruptRequest pin.
- Thispinisusedtoreceiveaninterrupt requestsignal.Itisatypeofmaskableinterrupt.

**INTA’**

- Pinno-11isusedasINTA’(Interruptacknowledgement)pin.

- Iftheinterruptisactivatedthentheprocessorwillsendanacknowledgemessagethrough INTA′ pin i.e. INTA′=0, INTA=1.

**AD0-AD7&A8-A15**

- Pinno-12topinno-19areusedasbothdatalineandaddresslinepin(AD0-AD7).
- Pinno-21topinno-28areusedasaddresslinepin(A8-A15).
- Thatmeansin8085 microprocessorshave8-bitdatalineand16-bitaddressline.

**VSS**

- Pinno-1isusedasVSSorGNDpin.

**S0&S1**

- Pinno-29andpinno-33isusedforS0&S1respectively.
- These are 2 status signal pins which are basically used to check read, write and opcode operation,whereS1isusedformemoryreadoperationandS0isusedformemorywrite operation.

| S1 | S0 | OPERATION |
|---|---|---|
| 0 | 0 | HALT |
| 0 | 1 | WRITE |
| 1 | 0 | READ |
| 1 | 1 | FETCH |

**ALE**

- Pinno-30isusedasALE(AddressLatchEnable)pin.

**WR′**

- Pinno-31isusedasWR′pin.
- Itisusedforwriteoperation(WR′=0,WR=1).
- Itisbasicallyusedtoseparatetheaddressfromtheaddress anddatabus.

**RD'**

- Pinno-32isusedas RD’pin.
- Itisareadsignalusedfor readoperation.Itisalsoanactivelow signal

**IO/M’**

- Pinno-34is usedforInputOutput/Memory’ pin.
- If the data is transferred to one processor to another processor then it is IO operation. So, in that case IO/M’=1
- If the data is transferred within the processor itself that means either the data is transferred toaccumulator or to some other memory address or register sointhat case it is either a memory read or memory write operation so IO/M’=0.

**READY**

- Pinno.-35isusedasREADYpin.
- ThisisanacknowledgmentsignalfromtheslowerI/Odevicesormemory.
- Whenhigh,it indicates that thedeviceis readytotransfer data,elsethemicroprocessor is in the wait state.

**CLK**

- Pinno-37is usedas CLK pin.
- Thispintellsabouttheclock pulse.
- ThroughthispinwecanconnecttootherdigitalICpinsandbasicallyusetoprovide square wave pulse or clock pulse or clock frequencies.

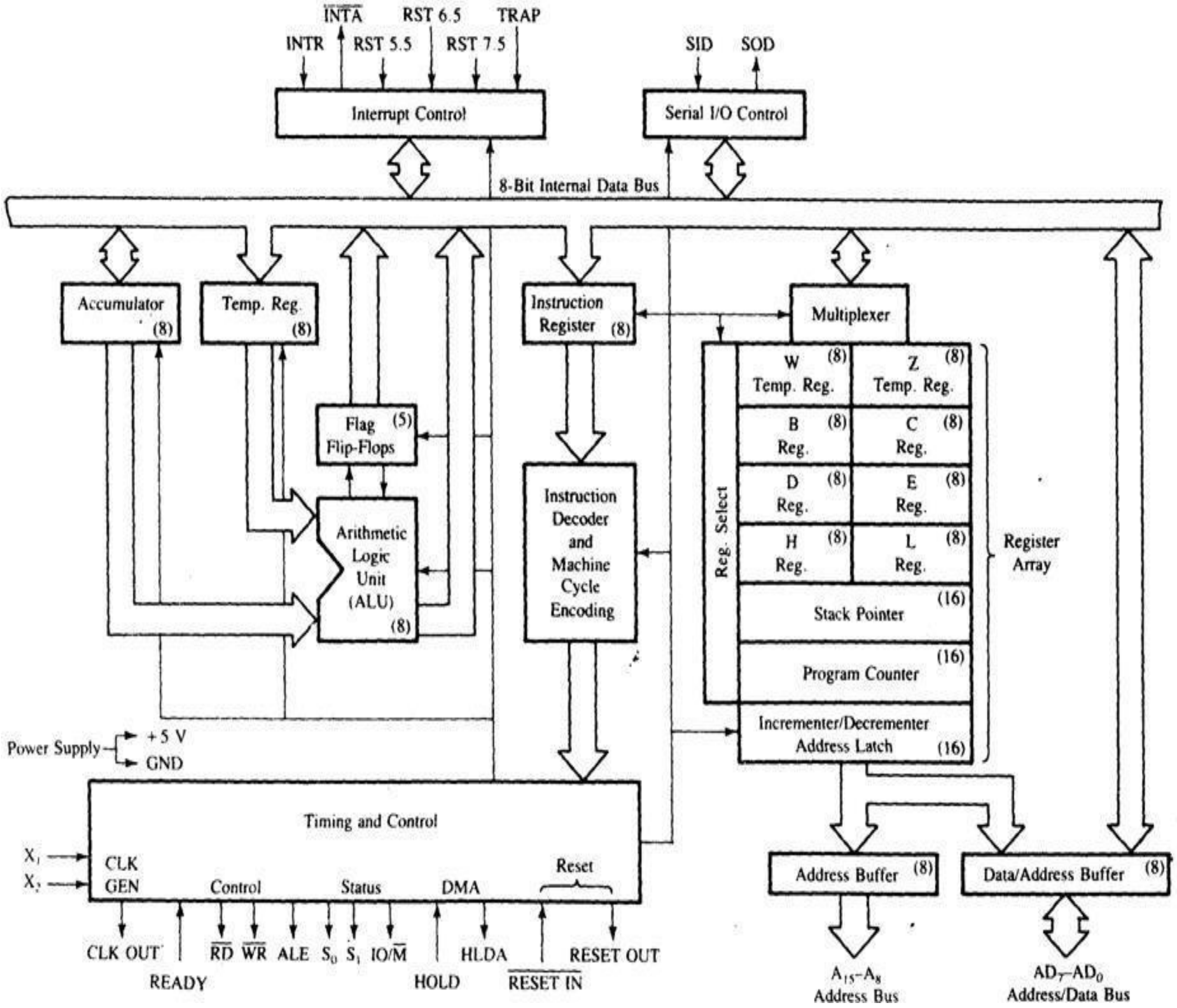
**HOLD& HLDA**

- Pinno-38isusedas HLDAandpinno-39 isusedasHOLDpin.
- Once the HOLD pin is activated then the processor will not allow any external data to interferethecurrentprogramandiftheholdissuccessfullyactivatedthentheprocessor will send an acknowledge message through the HLDA pin.

**VCC**

- Pinno-40isusedasVCCpin.
- ThroughVCCpin+5VDCsupply isprovidedtotheIC.

# INTERNALARCHITECTUREOF8085MICROPROCESSOR



The total internal architecture of 8085 microprocessor can be divided into 3 major units they are;

## i. ArithmeticandLogicUnit(ALU):

The main function of ALU is to perform arithmetic operation such as addition, subtraction, multiplication and division etc. and logical operation such as AND, OR, EX-OR etc.

To ALU 3 other sub units are attached which helps in performing different arithmetic and logical operation they are:

### A. Temporaryregister:

The main function of temporary register is to store the data temporary before the data is transforming to ALU to performing different arithmetic and logically units.

**B. Statusflag:**

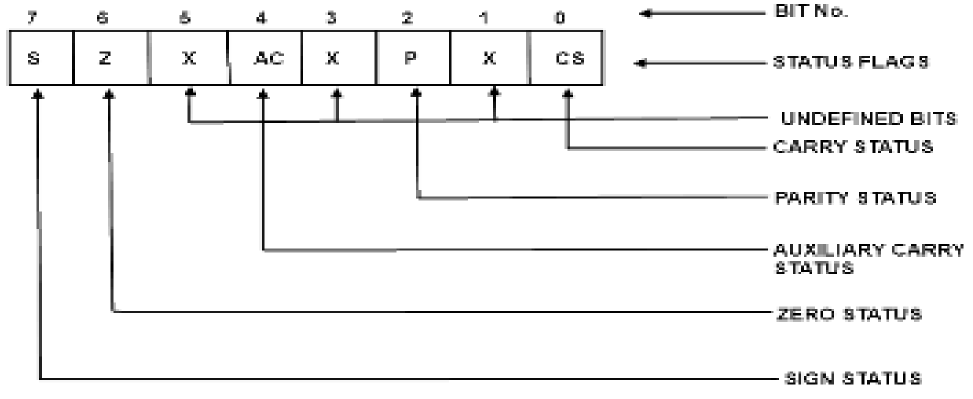


Figure 2: Status Flags of Intel 8085

The main function of status flag is to check the status of the output program or to checkwhethertheresultisrightorwrong.Itconsistsof8-bitoutofwhich5areactive flags and three re undefined bits.

- **Sineflag:**Afterthearithmeticoperationiftheresultisnegativethensign flag is tends to logic 1 otherwise it will tend to logic 0.
- **Zeroflag:**Afterthearithmeticoperationiftheresultiszerothenzeroflag will tend to logic 1 otherwise it will tend to logic 0.
- **Auxiliaryflag:**Afterthearithmeticoperationifthereisacarryfrom3$^{rd}$to 4$^{th}$bitthenauxiliarycarrywilltendstologic1otherwiseitwilltendtologic 0.
- **Parityflag:**Afterthearithmeticoperationiftheresultofthesumcontains evenno. of1'sthenparityflagwilltendtologic1otherwiseitwilltendto logic 0.
- **Carry flag:** After the arithmetic operation if the result is more than 8 bits then there will be a carry from 7-8 bit so carry flag will tends to logic 1.

**C. Accumulator:**

Thefinalresultofarithmeticandlogicaloperationisstoredinaccumulator.

ii. **Setsof register:**

Registersaretemporarystoringdeviceandactsasflipflop.Wecanstorethedatainregister temporarilyandwecandeleteit.In8085microprocessorthereare6generalpurpose register they are B, C, D, E, H, L and each register can store 8-bits of data individually that meansBcanstore8-bitofdata,Ccanstore8-bitsofdataandsoon.Butinpairformi.e.BC, DE, HL ARE STORES 16-bits of data. This are known as general purpose register because in most of the programming we use these registers commonly for storing 16-bits or 8-bitsof data.

- **Special purpose register**

  These are 16-bits registers which are used for some specific purpose. The special purpose registers are:

  **A. Stackpointer**

  - **Stack:** Stack is a set of memory location whose address is different from main memory. We can transfer the data from main memory to stack memory by push instruction and we can retrieve the data from stack memorytomainmemorybypopinstruction.Thedatatransformfrommain memorytostackmemorybyFIFO(FirstinFirstOut)sequenceandisretrieve from stack memory to main memory by LIFO (Last in Last Out) sequence.
  - To locate a particular memory address, we take the help of stack pointer and is given by LXI SP, 9904- load the content of stack pointer into the memory address 9904.

  **B. Programcounter**

  - Itholdstheaddressofthenextinstructionoritcheckswhethertheaddress for next instruction is available or not.

  **C. Incrementdecrementlatch**

  - Itisbasicallyusedforincrementanddecrementoperation.

### iii. Timingandcontrolunit:

Themainfunctionof timingandcontrol unitsistocheck howmuchtimethe processor will taketoexecutetheoperation.Tothetiming andcontrol unit anexternal crystal oscillatoris attachedwhichprovideandinternalclockfrequencyup-to5MHz.Sothattheprocessorwill take a specific amount of time i.e. T=1/F, here F=5MHz so T=0.2μsec time is taken by the processor to execute the operation.

Tothetimingandcontrolunitstwootherunitsareattachedtheyareinstructionregisterand instruction decoder.

**Instruction registers:** The main function of instruction registers is to store to opcode of an instruction.

**Instruction:**Instructionsaresetofcommandgiventotheprocessortoperformaspecific operation.

e.g.MVIA,38

Eachinstructionhas2partsi.e. opcodeandoperand

- **Opcode**:Thefirstpartofinstructionwhichspecifiessometaskwedonebytheprocessor. Here MVI is the opcode which means move immediately the data.
- **Operand**:The second part of instruction which is basically used to store the data or to perform a memory read or memory write operation.
  - ✓ **Memory read:** When the data is transfer to accumulator then that is the memoryread operation.
  - ✓ **Memory write:** When the data is transferred to some other memory address apart from accumulator then that is memory write.

**Instructiondecoder:**Itsmainfunctionistoconvertthemnemonicstoitsmachinecode.

**Mnemonics:** Mnemonicsaretheliterallanguageunderstoodbyuser-oruser-friendly language.

## INSTRUCTIONSETSOF8085MICROPROCESSORS

Instructionsaresets ofcommandsgiventotheprocessortoperform aspecificoperation. Eachinstructioncanbedefined intotwopartsoneisthe opcode and anotherisoperand.

- **According to word length:** Word length means the no. of bits or bytes a specific instruction occupies. Accordingly, there are 3-types i.e.1-byte, 2-byte, 3byte.
  - **a) 1-byte:** In 1-bytes instruction the opcode is define but the operand data is not directly specified but is specified by some register.
    e.g. ADD B- In this case the opcode is defined i.e. ADD which has got a machine code of 8-bit, but the operand is specified by register so it is 1-byte instruction.
  - **b) 2-byte:**In2-byteinstructionopcodeispresentfollowedby8-bitoperanddata.
    e.g.MVIA,30-Inthiscasetheopcodeisdefinedi.e.MVIwhichhasgotamachine codeof8-bit and theoperand datais 30 whichhas thebinaryvalue00110000 i.e. 8-bit. So, the total size is 16-bits and called 2-byte instruction.
  - **c) 3-byte:**In3-byteinstructiontheopcodeispresentandfollowedby16-bitoperand data.
    e.g.STA9100-Inthis casethe opcodeis definedi.e.STAwhichhas got amachine codeof8-bitandtheoperanddatais9100whichhasthebinaryvalue

1001000100000000 i.e. 16-bit. So, the total size is 24-bits and called 3-byte instruction.

- **According to operation:** according to the type of operation it performs the total instruction sets can be classified into 5 different types, they are

  a) **Datatransforminstruction:**Itisbasicallyusedfortransferringthedatafromone register to another register or register to memory without changing the content.
  E.g.MOVA,B-MovethecontentofBregisterstoaccumulator. MOV
  A,$34_H$-Move the 8-bit data $34_H$ to accumulator.

  b) **Arithmeticinstruction:** These are basically used for arithmetic operationsuch as addition,subtraction,multiplication,division,increment,decrement etc.Inthiscase the final result may change.
  e.g.ADDB-Addthecontent ofregisterBwiththeaccumulator.

  c) **Logical instruction:** These instructions are basically used for logical operation such as ANA (AND), ORA (OR), XRA (XOR), CMP (COMPARE) etc.

  d) **Branch control instruction:** The instruction under this group are basically used for conditional or unconditional jump operation.
  E.g.
  JNZ- Jump if the counter data in the register C is not zero to label loop.
  JNC- Jump if there is no carry or borrow to label loop.

  e) **IO and machine control:** The instruction under this group are basically used for controlling the device or for transferring the data from one device to another deviceorforset orresetofthestatusflagorforstack operationcomesunderthis group.
  e.g.HALT(tostoptheprogram)
  INO2 (to receive the data from port 02)
  OUT01(totransferdatathroughport01) STC
  (set the carry flag to logic-1)

## ADDRESSINGMODEOF8085MICROPROCESSOR

Itisthetechniquethroughwhichwearespecifyingdataforoperationorhowtheoperanddata is specified accordingly in 8086 microprocessors. There are 8 different addressing mode according to the type of operation it performs.

1. Registeraddressingmode
2. Immediateaddressing mode
3. Directaddressingmode
4. Registerindirectaddressingmode
5. Implicitaddressing mode

1. **Register addressingmode:** Inthistypeof addressing modetheoperand datais not directly specified in the instruction itself but it is specified by some register.
   E.g.*MOVA,B*-Movethe contentofBregistertoaccumulator.
2. **Immediateaddressingmode:**Inthistypeofaddressingmodetheoperanddata is directly specified in the instruction itself.
   E.g.*MOIA,08$_H$*-Moveimmediatelythedata08$_H$toaccumulator.
3. **Directaddressingmode:**Indirectaddressingmodetheoperandaddressis directly specified in the instruction itself.
   E.g.STA9100-Storedthecontentofaccumulatortomemoryaddress9100.
4. **Registerindirectaddressingmode:**Inthistypeofaddressingmodetheoperand data is not directly transferred to the accumulator; at first it stored in some memory address and the transferred to the accumulator.
   E.g.LXIH,9100-Loadthecontent ofHLpair intothememoryaddress9100. MOV A, M- Move the content of the memory address 9100 to accumulator.
5. **Implicit addressing mode:** There are certain instruction through which we can automaticallycomparewiththepreviousvalueofaccumulatorandtheinstruction under this group comes under implicit addressing mode.
   E.g.CMPM-Comparethecontent ofmemorywithaccumulator.

   RAL- Rotate the content of accumulator to left by 1-bit.

   RAR-Rotatethecontentof accumulatortorightby1-bit.

**ASSEMBLYLANGUAGEPROGRAMMINGFOR8BITADDITIONLA**

| BEL | MNEMONICS | OPERANDS | COMMENT |
|---|---|---|---|
| | MVI | A,49 | Move immediately the data$49_H$ to accumulatorA. |
| | MVI | C, 56 | Moveimmediatelythedata$56_H$toregister C. |
| | ADD | C | Add the content of register C with accumulator A and the result is stored in accumulator. |
| | STA | 9100 | Storethecontentofaccumulatorintothe memoryaddress9100. |
| | END | | |

DATA:

$1^{ST}$data=$49_H$

$2^{nd}$data=$56_H$

RESULTinmemorylocation9100=9F

**OR**

**ASSEMBLYLANGUAGEPROGRAMMINGFOR8BITADDITIONUSINGMEMORYLOCATION**

| LABEL | MNEMONICS | OPERANDS | COMMENT |
|---|---|---|---|
| | LXI | H,9100 | LoadthecontentofHLpairintothe memoryaddress9101. |
| | MOV | A, M | Move the content of memory to accumulator. |
| | INX | H | IncrementthecontentofHLpairbynext bit. |
| | ADD | M | Add the content of memory with accumulatorandtheresultisstoredin accumulator. |
| | STA | 9103 | Storethecontentofaccumulatorintothe memoryaddress9103. |
| | END | | |

DATA:

Memorylocation9101=$49_H$

Memorylocation9102=$56_H$ResultinMemorylocation9103=9F

## ASSEMBLYLANGUAGEPROGRAMMINGFOR8BITMULTIPLICATIONRESULTING 16 BIT NUMBER.

| LABEL | MNEMONICS | OPERANDS | COMMENT |
|---|---|---|---|
| | MVI | E, 05 | Moveimmediatelythedata05to register E. |
| | MVI | C, 05 | Moveimmediatelythedata03to register C which is used for counter operation. |
| | MVI | D,00 | Move the initial data 00 to register D. |
| | LXI | H,0000 | Load the 16-bit data 0000 to HL pair. |
| LOOP | DAD | D | Add the content of HL pair with DEpairandtheresultisstoredin HL pair. |
| | DCR | C | Decrement the content of register C by 1 bit. |
| | JNZ | “LOOP” | Jump till the counter data in register C is not zero to label LOOP. |
| | SHILD | 9105 | Store the content of HL pair into the memory address 9105 &9106. |
| | END | | |

RESULT:
Inmemorylocation9105=0F
Inmemorylocation9106 =00

## ASSEMBLYLANGUAGEPROGRAMMINGFORDIVISIONOF8BITBYA8BITNUMBER.

| LABEL | MNEMONICS | OPERAND | COMMENT |
|---|---|---|---|
| | MVI | A,13 | Move the hexadecimal equivalent of 13 to accumulator A. |
| | MVI | B,05 | Move the hexadecimal equivalent of 05 to register B. |
| | MVI | C, 00 | Movetheinitialdata00to register C, which is basicallyused tostorethe quotient. |
| LOOP | SUB | B | Subtract the content of register B from accumulator A and the resultisstoredin accumulator. |
| | INR | C | Incrementthecontentof register C by 1 bit. |
| | CMP | B | Compare the content of register B with accumulator A. |
| | JNC | “LOOP” | Jumpifthereisnocarryor borrow to label loop. |
| | END | | |

RESULT:<br>
Quotient is in register= $03_H$<br>
Remainderisinaccumulator=$04_H$

**PROGRAMFORFINDINGTHESMALLESTNUMBERRFROMAGIVENDATA**

| **LABEL** | **MNEMONICS** | **OPERANDS** | **COMMENT** |
|---|---|---|---|
| | LXI | H, 9101 | LoadthecontentofHLpairintothememory address 9101. |
| | MOV | C,M | MovethecontentofmemorytoregisterC whichisusedforcounteroperation. |
| | INX | H | Increment the content of memory to accumulator. |
| | MOV | A, M | Move the content of memory to the accumulator. |
| | DCR | C | DecrementthecontentofregisterCby1 bit. |
| | INX | H | IncrementthecontentofHLpair address by next bit. |
| | CMP | M | Compare the content of memory with accumulator. |
| | JC | “AHED” | Jump with carry or borrow to label “AHED”. |
| | MOV | A, M | Move the content of memory to accumulator. |
| | DCR | C | DecrementthecontentofregisterCby1 bit. |
| | JNZ | “LOOP” | JumpinthecounterdataintheregisterCis notzerotolabelloop. |
| | STA | 9300 | Storethecontentofaccumulatorintothe memoryaddress9300. |
| | END | | |

DATA:

In memory address 9101= $03_H$

memory address 9102= $86_H$

Inmemoryaddress9103=$58_H$

Inmemoryaddress9104=$75_H$

RESULT:

Inmemoryaddress9300=$58_H$ In

## PROGRAMTOFINDTHELARGESTNUMBERFROMAGIVENDATAARRAY

| LABEL | MNEMONICS | OPEANDS | COMMENT |
|---|---|---|---|
| | LXI | H,9101 | LoadthecontentofHLpairintothe memory 9101. |
| | MOV | C,M | Move the content of memory to the register C which is used for count operation. |
| | SUB | A | Subtractthecontentofaccumulator fromaccumulator(A=0)andresultis storedinaccumulator. |
| LOOP | INX | H | Increment the content of HL pair address by next bit. |
| | CMP | M | Compare the content of memory with accumulator. |
| | JNC | “AHED” | Jump if there is no carry or borrow to label “AHED”. |
| | MOV | A,M | Movethecontentofmemorytothe accumulator. |
| AHED | DCR | C | DecrementthecontentofregisterCby 1 bit. |
| | JNZ | “LOOP” | Jump till the counter data in register C id not zero to label “LOOP”. |
| | STA | 9105 | Storethecontentofaccumulatortothe memory address 9105. |
| | END | | |

DATA:

In memory address 9101= 03$_H$

memory address 9102= 98$_H$

Inmemoryaddress9103=A5$_H$

Inmemoryaddress9104=29$_H$

RESULT:

Inmemoryaddress9105=A5$_H$ In

## PROGRAMFORBINARYTOGRAYCOADCONVERSION

| LABEL | MNEMONICS | OPERAND | COMMENT |
|---|---|---|---|
| | MVI | A,48 | Moveimmediatelythedata48to accumulatorA. |
| | MOV | C, A | Movethecontentofaccumulatorto register C. |
| | STC | | Setthecarryflagtologic-1. |
| | CMC | | Complementofcarrybit. |
| | RAR | C | Rotatethecontentofaccumulatorto rightby1bit. |
| | XRA | | X-ORofregisterC. |
| | END | | |

RESULT:Graycodeisinaccumulator=$6C_H$.

48 -01001000

C -01001000

STC -101001000

CMC-001001001

RAR- 0001001001

XRA- 001101101=>6C

## TIMING DIAGRAM

Itisagraphicalrepresentationofinstructioncycleoritshowshowthecontrolsignalsare affected when we are performing an opcode operation r an execution operation.

**Instructioncycle:**

Itisthetotaltimerequiredtoreadtheopcodeofaninstructionfromthememoryandto perform the execution operation.

Instructioncycle=Opcodefetchcycle+Executioncycle

- **Opcodefetchcycle:**
  It is the total time required to read the opcode of an instruction from the memory. Normallytheopcode fetchcycle hasgotfourT-statesor Timeperiod($T_1$,$T_2$,$T_3$,$T_4$)during which a specific operation is being perform.
  - **T1-state:**Duringthisstateofmachinecycletheaddressisreadfromthememory.

- **T2 & T3-state:** During these states the hexadecimal code or the machine code of an instruction is read from the memory.
- **T4state-**Thisstateisknownaswaitstateorduringthisperiodthedataistransferred from the opcode fetch cycle to execution cycle.

- **Executioncycle:** Itisthetotaltimerequiredtoperformthememoryreadormemorywriteoperation.

  In execution cycle the final result is stored so we do not require any wait state (time gap for data overlap).

  If the operand data is 8-bit then we required 3T states and if the operand data is 16-bitthen we required 6T states.

- ✓ DuringT1stateofexecutioncycletheaddressisreadfromthememory.
- ✓ DuringT2&T3statesofexecutioncycletheoperanddata isreadfromthememory.

**Q.Iftheexternalclockfrequencyis5MHzthenhowmuchtimetheprocessorwilltaketo execute the operation**

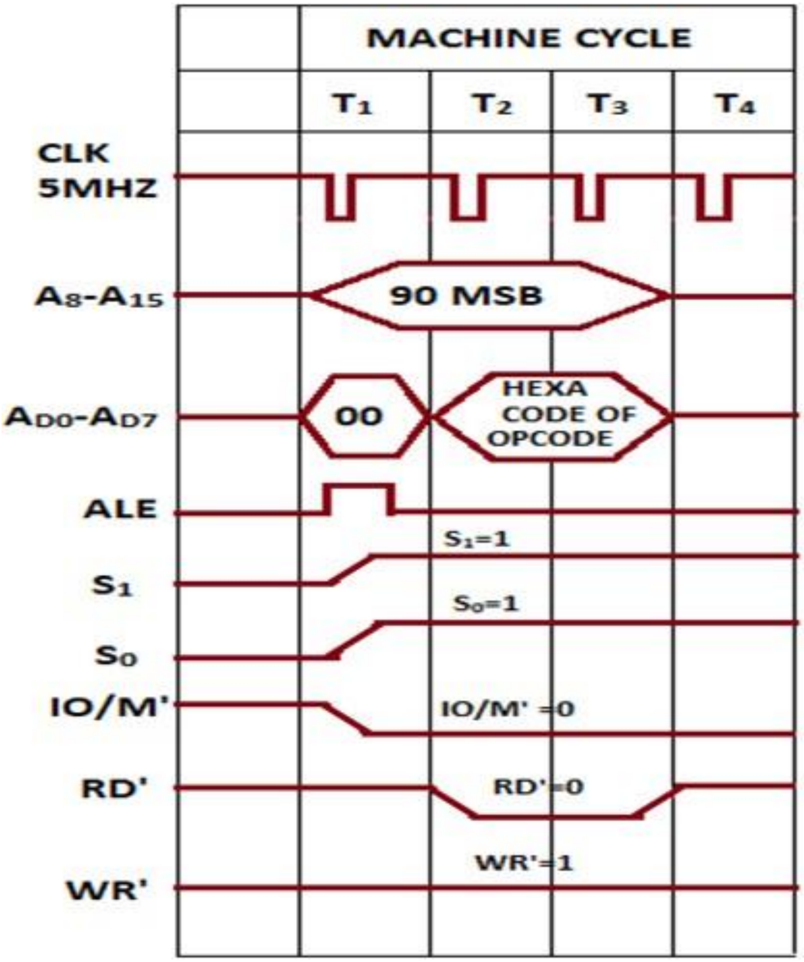
I. **MVIA,30**
II. **MOVA,B**
III. **LXIH,9105**

Solution:Givendataf=5MHz

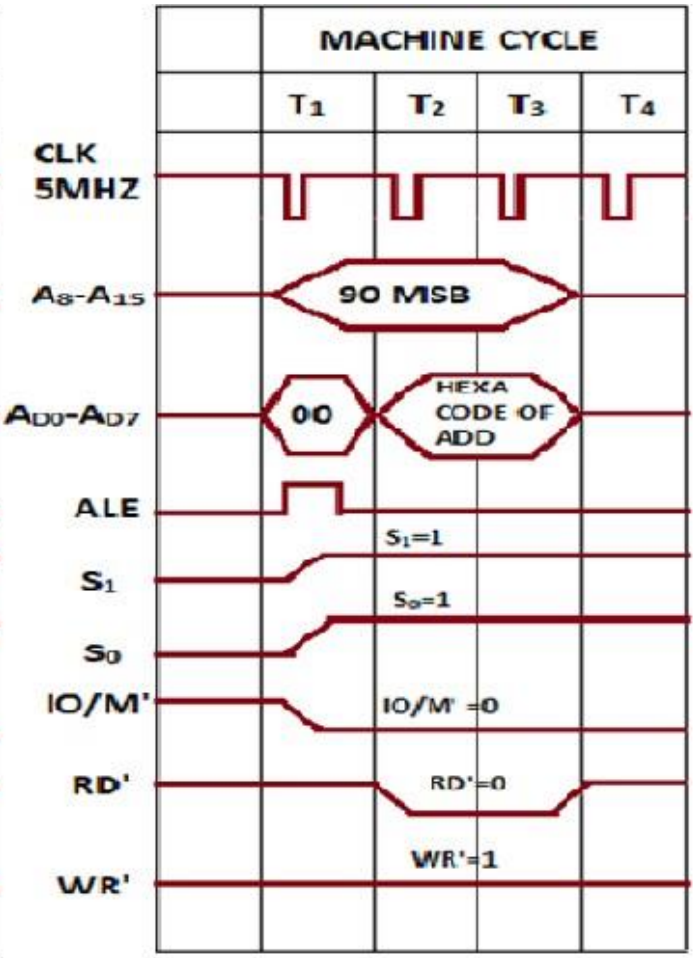
.'.T= 1/f= 1/5MHz= 0.2μsec

I. MVIA,30= 4T+3T= 7T=7*0.2μsec= 1.4μsec
II. MOVA,B= 4T+0T= 4T=4*0.2μsec= 0.8μsec
III. LXIH,9105= 4T+6T=10T = 10*0.2μsec=2μsec

## Timingdiagramofopcodefetchcycle&ADDB



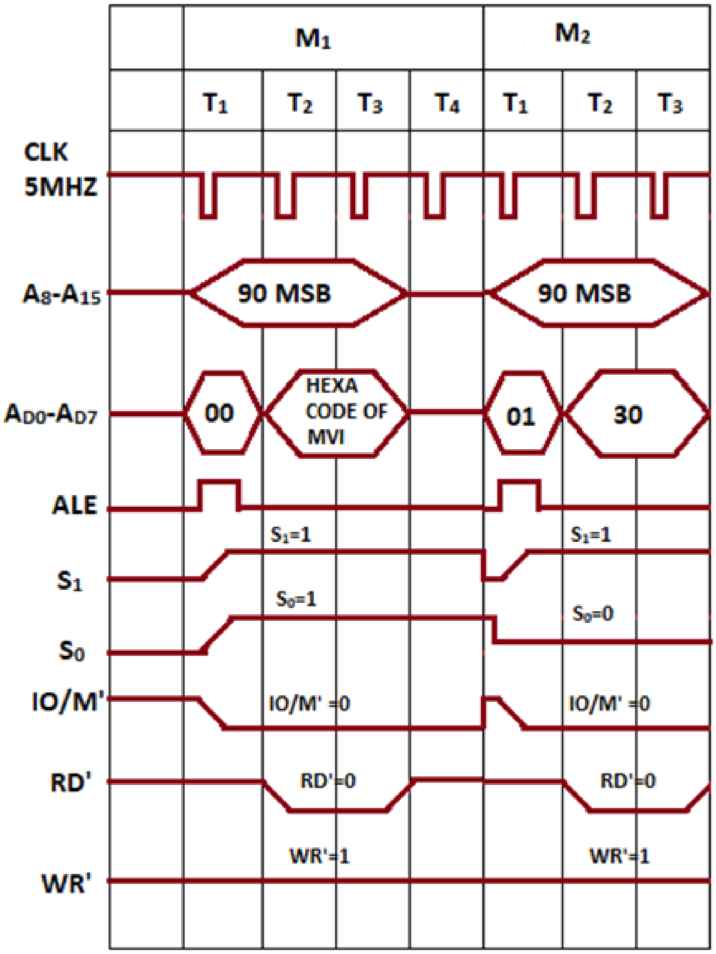
[ Timing diagram of opcode fetch cycle ]



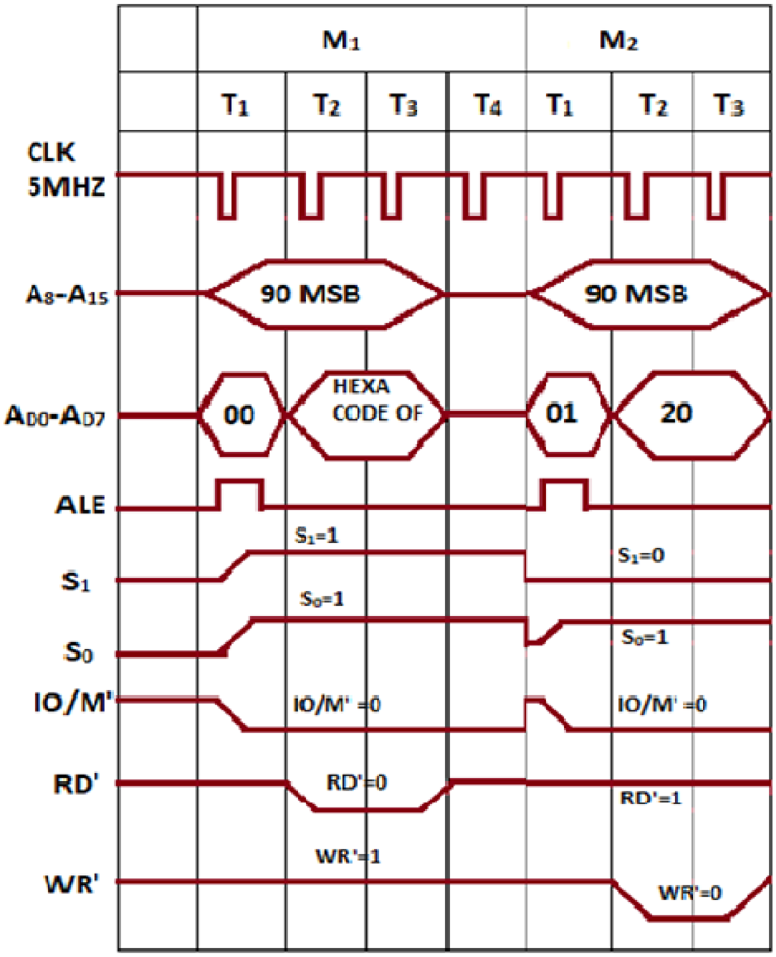
[ Timing diagram of ADD B ]

## TimingdiagramofMVIA,30&MVIB,20

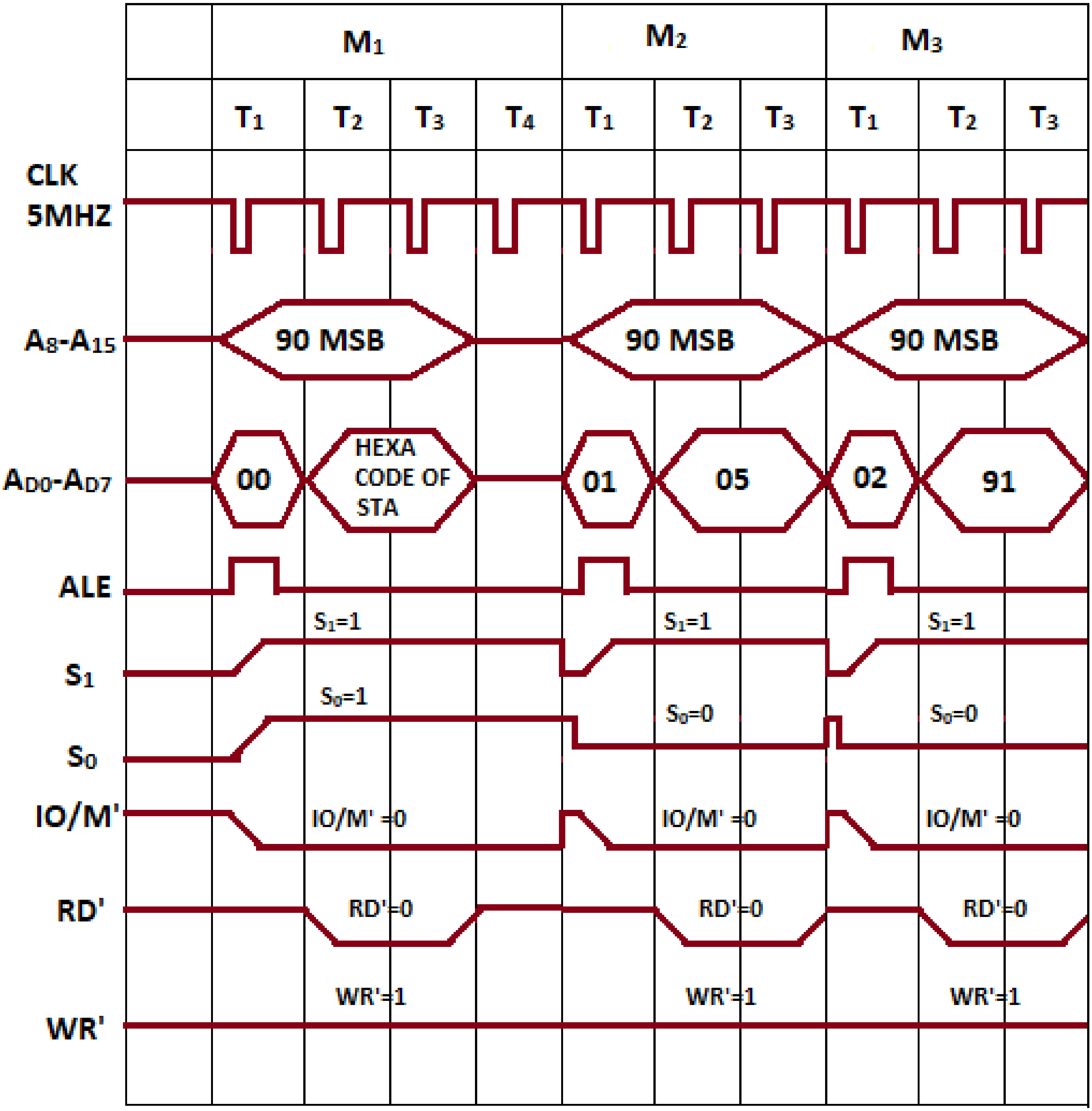
[ Timing diagram of MVI A, 30 ]



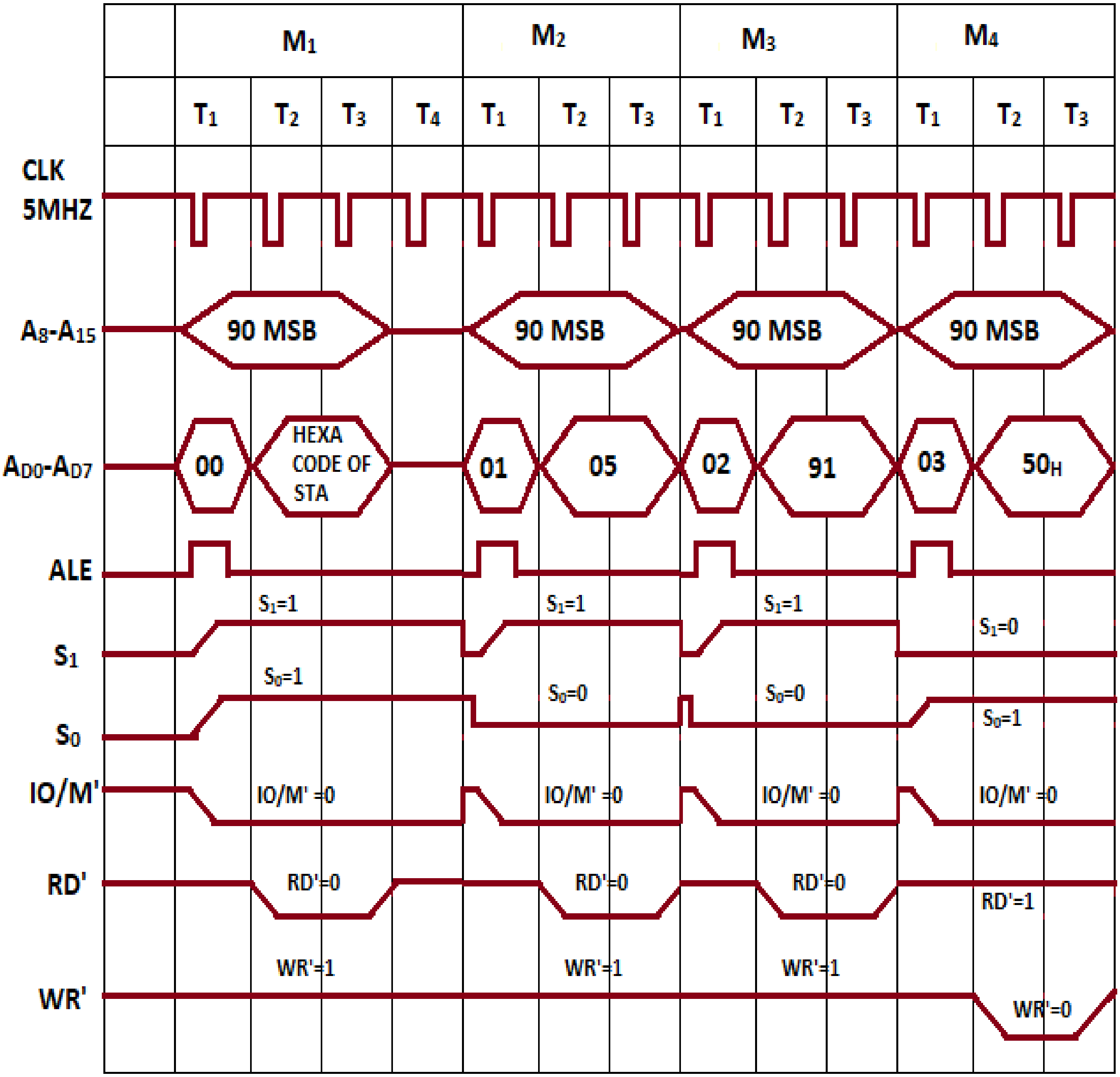
[ Timing diagram of MVI B, 20 ]



**TimingdiagramofSTA9105**



## TimingdiagramofSTA9105iftheaccumulatorcontains$50_H$



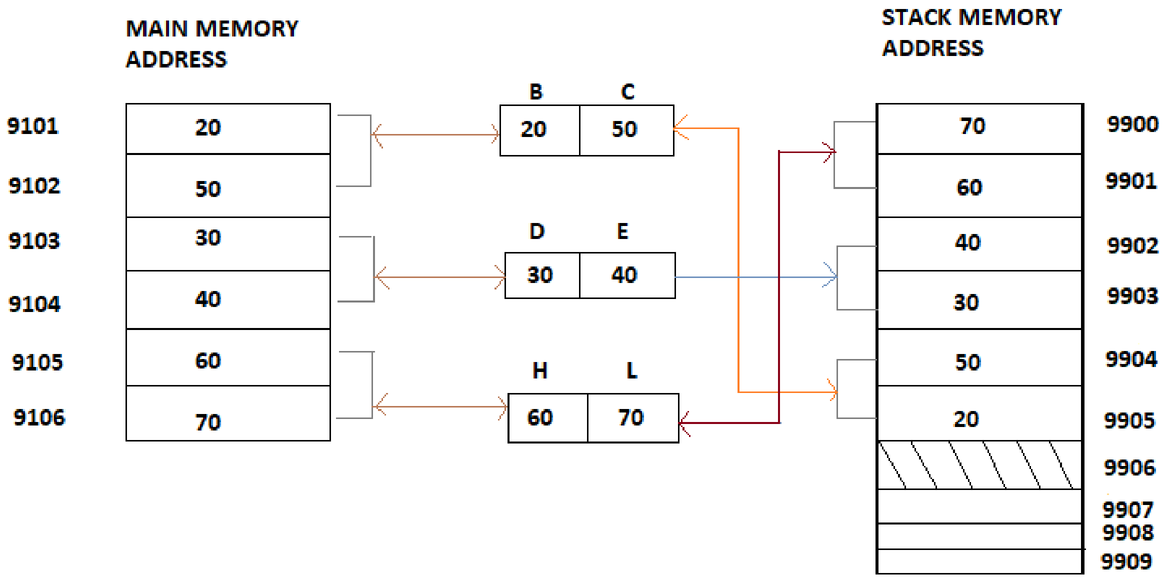
## STACKANDSUBROUTIN:

**Stack:**Stackisthesetswhoseaddressisdifferentfrommainmemoryaddress.Wecantransfer the data from main memory to stack memory by PUSH instruction and the data is transferred in FIFO sequence and we can retrieve the data from stack memory to main memory by POP instruction and the data is retrieve by LIFO sequence.

**Subroutine:** Subroutineis asubtaskoritisasmallprograminbetweenthemainprogram.

**Q. Initialize the stack memory address 9906. Write a program to transfer the data from mainmemorytostack&thenretrieveit.Supposethedatacontaininmainmemoriesare: 9101-2050$_H$, 9103- 3040$_H$, 9105- 6070$_H$.**

Solution:



| MNEMONICS | OPERAND | COMMENT |
|---|---|---|
| LXI | SP,9906 | |
| LXI | B,9901 | Load the data in BC register. |
| LXI | D,9903 | Load the data in DE register. |

| | | |
|---|---|---|
| LXI | H, 9905 | Load the data in HL register. |
| PUSH | B | Store the data in BC register. |
| PUSH | D | Store the data in DE register. |
| PUSH | H | Storethedatainregister. |
| POP | H | Retrieve the data from register. |
| POP | D | Retrieve the data from register. |
| POP | B | Retrieve the data from register. |

**Multiplechoicequestionanswer:**

1. The8085Upis

   A) 16 bit
   B) 8bit
   C) 20 bit
   D) 32 bit
   Answer:B

2. Theaddresslineof8085upis

   A) 16 bit
   B) 8bit
   C) 24 bit
   D) noneofthese.
   Answer: A

3. ThefunctionofALU is

   A) Arithmetic operation
   B) Arithmeticandlogicaloperation
   C) Logicaloperation
   D) Noneofthese.
   Answer: B

4. Thefunctionofcrystal oscillator

   A) providinginternalclockfrequency
   B) impedancematching
   C) reducingspeed
   D) Disconnectingports
   Answer: A

5. The no.Ofgeneral-purposeregisterin8085upis

   A)6
   B) 5
   C)4
   D)3
   Answer:A

6. Theno.Ofaddressingmodein8085 upis

   A)5
   B) 4
   C)3
   D)2

Answer:A

7. The no.Ofinstructionsetin8085upis

A)4

B) 3

C)5

D)2

Answer:C

8. Theno.Ofinterruptin8085 upis

A) 5

B) 6

C) 7

D) 256

Answer:A

9. Timingdiagramis

A) interfacing

B) connecting

C) Analyzing.

D) Graphicalrepresentationofinstructioncycle

Answer: D

10. MVIA,30is

A) Immediateaddressing mode

B) Implicit

C) Direct

D) Indirect

Answer: A

11. Theno.Ofpinin8085up is

A) 40.

B) 30

C) 28

D) 32

Answer:A

12. Theinterrupthaving highestpriorityis

A) TRAP

B) INTR

C) RST0

D) RST1

Answer:A

13. Theclockatwhich80i85upoperateis

    A) 4MHz

    B) 6

    C) 5.

    D) Noneofthese

    Answer: C

14. MVIA,30is

    A) 2Byteinstruction

    B) 1

    C) 3

    D) 4.

    Answer:A

15. MOVA,Bis

    A) 1Byteinstruction

    B) 2

    C) 4

    D) 3

    Answer:A

16. TheinstructionSTAA.9000His

    A) 3Byteinstruction.

    B) 1.

    C) 2.

    D) Noneofthese

    Answer: A

17. TheInstructionLXIH,9100His

    A) 2Byteinstruction.

    B) 3.

    C) 4

    D) Noneofthese

    Answer: B

18. TheinstructionMOVA,Bis

    A) Directaddressingmode

    B) indirectaddressing mode

    C) Immediateaddressing mode

    D) Registeraddressingnode

    Answer: D

19. The no.OfTstatesininstructionisMVIA,30His

A) 4

B) 7

C) 10

D) Noneofthese.

Answer: B

20. Theinstructioncycleconsistof

A) opcodefetchcycle.

B) opcodefetchcycleandexecutioncycle.

C) executioncycle

D) Noneofthese.

Answer: B

## ASSIGNMENTFULL MARKS-100

### SECTION-A

### (ANSWERALLQUESTIONS)

**Shortanswer type question: 2*8=16**

a) Whatarethedifferenttypes ofaddressingmodesin8085microprocessor?

b) Which opcode isusedtotransferandwhich opcode isusedto retrieve thedata from main memory to stack memory?

c) Howmanyno. ofmachinecycles arerequiretoexecutetheinstructionMVIA, 47?

d) GivesomeexampleofIOandmachinecontrolgroup instruction.

e) Explainthetermsopcodeandoperandinaninstructionwithexample.

f) Draw the table of status code of 8085 microprocessor and indicate the read, write, halt and fetch operation.

g) Whatisthefunctionofprogramcounterin8085microprocessor.

h) Explainthedifferentregistersin8085microprocessor.

### SECTION-B

**Focusedanswertypequestion: 6*6=36**

a) Whatisinterruptoperation?Whatarethedifferentinterruptpinsarein8085microprocessor and how they work?

b) Explainthestatusflagofthe8085microprocessor.

c) Whatistimingdiagram?DrawthetimingdiagramofMVIA, B.

d) Writeaprogramtoconvert abinarycodeto graycode.

e) ExplaintheT-statesinopcodefetchcycleandexecutionfetch cycle.

f) Explainsomeimportantfeaturesof8085microprocessor.

## SECTION-C

**Longanswertypequestion: 16*3=48**

a) Withpropersketchdescribetheinternalarchitectureof8085microprocessor.

b)

(i) Explainthetermstackandsub-routine.Supposethedatacontentofmainmemory, BC=2050$_H$,DE=1234$_H$,HL=2640$_H$.Writeaprogramtoinitializethestackmemory address at 9080$_H$ and transfer the data from main memory to stack memory and retrieve it.

(ii) Iftheexternalclockfrequencyis5MHzthenhowmuchtimetheprocessorwilltake to execute an operation,

- MVIA,30
- MOVA,B
- LXIH,9105
- ADDB

c)

(i) Writeanassemblylanguageprogramfor8-bitmultiplicationresultinga16-bit number.

(ii) DrawthetimingdiagramofLXIH,9100ifthedatacontentinthe registeris0756$_H$.

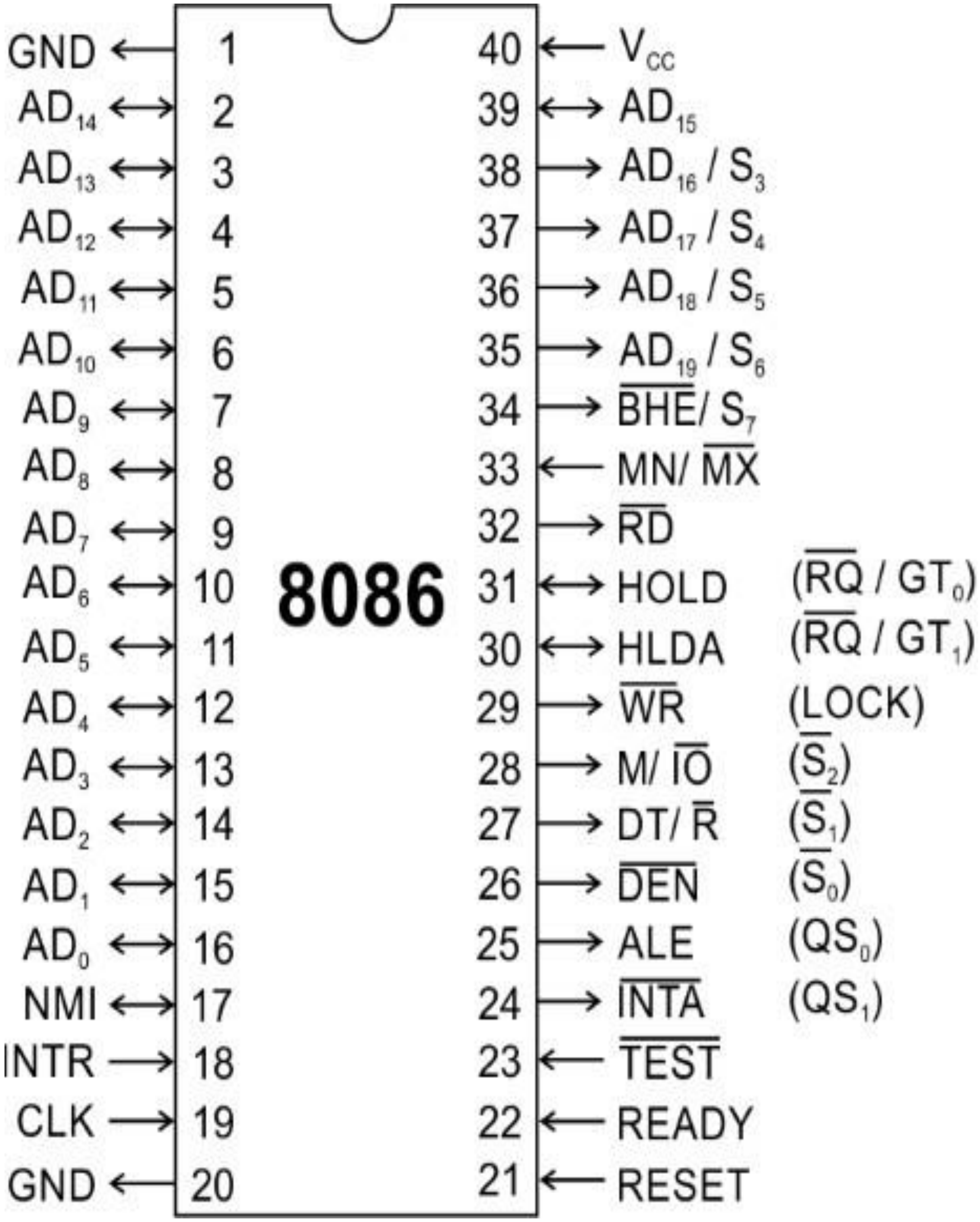
# MODULE-2:8086MICROPROCESSORS

## Importantfeaturesof8086Microprocessor:

- Itisa16bitprocessorthatmeansitsdatalineis16bitandaddresslineis20bit.
- Itconsists of40pinICchipandoperatesat +5VDC.
- Itcanoperatein3differentclockfrequenciesi.e.5MHz,8MHzand 10MHz.
- Itconsistsof9activestatusflagsoutofwhich6areconditionalflagand3are control flag.

| 8085 | 8086 |
|---|---|
| **1.** The 8085 microprocessor is an 8-bit processor that means its data line is 8 bit and address line is 16 bits.<br>**2.** It operates in 3-5MHz clock frequency.<br>**3.** It consists of 5 active status flags.<br>**4.** It consists of single unit and there is no division in architecture takes place.<br>**5.** It consists of single mode and basically used for simple input output operation. | **1.** It is a 16-bit processor or that means its data line is 16 bit and address line is 20 bits.<br>**2.** It can operate in 3 difference CLK frequencies i.e. 5MHz, 8MHz, 10MHz.<br>**3.** It consists of 9 active status flags outofwhich6 areconditionalflag and 3 are control flag.<br>**4.** The total internal architecture can be divided into 2 different units theyarethebusinterfaceunitand execution unit.<br>5. Itcanoperatein2differentmodes i.e.minimummodeandmaximum mode. |

## Pin diagramof8086Microprocessor:



## Pindescriptionfor8086Microprocessor:

Itconsists of40pinsandoperatesat+5VDC.

**GND**

- There aretwogroundpinsinthe8086,pin1andpin20.

### AD0toAD19

- Pinno-2topinno-16 andpinno-39 arebothused asdataline andaddressline(AD0to AD15).
- Pinno-38topinno-35are usedonlyforthe addressline(AD16toAD19).

### NMI

- Pinno-17isusedforNon-MaskableInterruptRequest.
- Thefunction of NMIpinissameasthatoftrappinof8085microprocessor.
- Thispinisbasicallyusedforinterruptoperationi.e.tostopthecurrentprogramandjump to some other program and till the completion of that specific program, the processor cannot return by to the original program.

### INTR

- Pinno-18isusedasInterruptRequest pin.
- Thispinisusedtoreceiveaninterrupt requestsignal.Itisatypeofmaskableinterrupt.

### CLK

- Pinno-19is usedas CLK pin.
- Thispintellsabouttheclock pulse.
- ThroughthispinwecanconnecttootherdigitalICpinsandbasicallyusetoprovide square wave pulse or clock pulse or clock frequencies.

### RESET

- Pinno.-21isusedasRESET pin.
- Byusingthispin,theprogramcontrolreturnstoFFFF$0_H$.
- Basically,itisusedtorestarttheprocessor iftheprogramhangsin between.

### READY

- Pinno.-22isusedasREADYpin.
- Thisisanacknowledgmentsignalfromtheslowerl/Odevicesormemory.
- Whenhigh,it indicates that thedeviceis readytotransfer data,elsethemicroprocessor is in the wait state.

### TEST'

- Pinno-23isusedas**TEST'**pin.
- Thisisalsoanactivelowsignal.Thispinisusedforwaitinstructionwhenthe8086is connected with the 8087 microprocessors.

- Normallytheprocessorspeedis fast andI/O devicespeedisslow.So,sometimeweare transferring the data to the processor but the data does not reach to the processor. So, in that case the processor goes to wait state and READY pin becomes ***LOGIC 0*** or deactivated soin that cases the **TEST'**pinbecomes activated i.e. **TEST'=0** and **TEST=1.** So, the processor goes to wait state.
- Again,whenthedatareachtheprocessorthenREADYpinbecomesactivatedortendsto ***LOGIC1***and**TEST'**pinbecomesdeactivatedi.e.**TEST'=1**and**TEST=0.**
- READYpinisoppositeof **TEST'pin.**

## MinimumandmaximumModePins-

Total8pins,fromPin24topin31workdifferentlyfordifferentmodes(maximumorminimum).

### ❖ Minimummode

- Forsimpleoutputoperationthe8086microprocessorscanoperateinminimum mode and in that case the MN/MX'=1 i.e. MN'=1, MX=0, MX=0, where MN is minimummodeandMXismaximummodeand,inthatcase,theminimummode pins get activated.
- The minimum mode pins are ***HOLD, HLDA, WR', DT/R', DEN', ALE, INTR', M/IO'.***

### HOLD& HLDA

- Pinno-30isusedas HLDAandpinno-31 isusedasHOLDpin.
- Once the HOLD pin is activated then the processor will not allow any external data to interfere the current program and if the hold is successfully activated then the processor will send an acknowledge message through the HLDA pin.

### WR'

- Pinno-29isusedasWR'pin.
- Itisusedforwriteoperation(WR'=0,WR=1).

### M/IO'

- Pinno-28isusedasmemory/input output pin.
- Ifthedataistransferredwithintheprocessorthen itisamemoryoperation.

**DT/R'**

- Pinno27isusedasDT/R'pin(Datatransfer andreceiverpin).
- If the DT/R'=1, DT=1, R'=1, R=0, then in that case it is a data transfer or write operationandthedatatransferredfromtheprocessortotheexternalIOdevice.
- If the DT/R'=0, DT=0, R'=0, R=1, then in that case it is a data receiver or read operation and the data received from the external IO device to the processor.

**DEN'**

- Pinno-26is usedasDEN'(DataEnable)pin**.**
- Itisbasicallyusedtocheckthevalidityofthedata.

**ALE**

- Pinno-25isusedasALE(AddressLatchEnable)pin.
- Itisbasicallyusedtoseparatetheaddressfromtheaddress anddatabus.

**INTA'**

- Pinno-24isusedasINTA'(Interruptacknowledgement)pin.
- If the interrupt is activated then the processor will send an acknowledge message through INTA' pin i.e. INTA'=0, INTA=1.

- **Maximum mode**
  - For multipurpose operation when more than 1 IC chips are used then the 8086 microprocessors can operates in maximum mode and in that case the MN/MX'=0 i.e. MN=0, MX'=0, MX=1.
  - Maximummodepinsare**$QS_1$,$QS_0$,$S_0$',$S_1$',LOCK',RQ'/$GT_1$,RQ'/$GT_0$**

**$QS_1$and$QS_0$**

- Pinno24and25areusedas$QS_1$and$QS_0$pinrespectively.

- These are queue status signals and are available at pin 24 and 25. These signals providethestatus ofinstruction queue. Their conditions are shown in the following table −

| $QS_0$ | $QS_1$ | Status |
|---|---|---|
| 0 | 0 | Nooperation |
| 0 | 1 | Firstbyteofopcodefromthe queue |
| 1 | 0 | Emptythe queue |
| 1 | 1 | Subsequentbytefromthequeue |

**$S_0'$,$S_1'$,$S_2'$**

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

| $S_2$ | $S_1$ | $S_0$ | Status |
|---|---|---|---|
| 0 | 0 | 0 | Interruptacknowledgement |
| 0 | 0 | 1 | I/ORead |
| 0 | 1 | 0 | I/OWrite |
| 0 | 1 | 1 | Halt |

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | Opcodefetch |
| 1 | 0 | 1 | Memoryread |
| 1 | 1 | 0 | Memorywrite |
| 1 | 1 | 1 | Passive |

**LOCK**

- When this signal is active, it indicates to the other processors not to ask theCPUtoleavethe system bus.It isactivated using theLOCKprefix on any instruction and is available at pin 29.

**RQ/$GT_1$andRQ/$GT_0$**

- These are the Request/Grant signals used by the other processors requestingtheCPUtoreleasethesystembus.Whenthesignalisreceived byCPU,thenitsendsacknowledgment.RQ/$GT_0$hasahigherprioritythan RQ/$GT_1$.

**RD'**

- Pinno-32isusedas RD'pin.
- Itisareadsignalusedfor readoperation.Itisalsoanactivelow signal.

**MN/MX'**

- Pinno-33isusedasMN/ MX'
- Thispinisusedforminimumormaximummodeofthemicroprocessor.Whenthispinis 1, the microprocessor works in minimum mode, and when the pin is at 0, the maximum mode is followed.

**BHE'/S7**

- Pinno-34isusedasBHE'pin.
- BHE stands for Bus HighEnable. It is an active low signal, i.e. it is active whenit is low. It is used to indicate the transfer of data over the higher order data bus (D8 to D15).

- BHE' decides whether the data bus will carry 16-bit data or 8-bit data. When BHE’ is enabled(i.e.0),thenthebuswillcarry16-bitdata,elseonly8-bitdatathroughthelower order data bus lines. It is multiplexed with status pin S7.

**VCC**

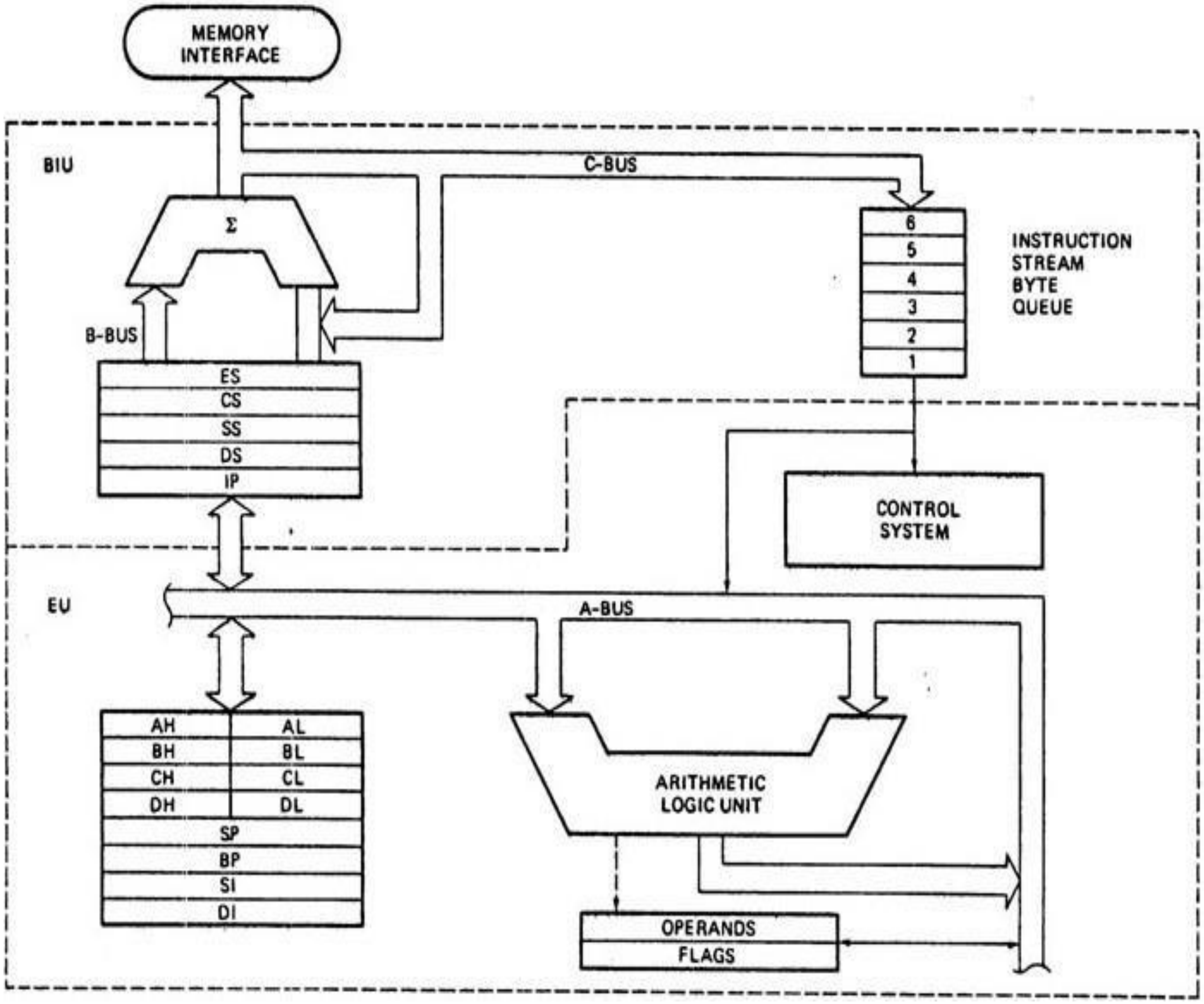
- Pinno-40isusedasVCCpin.
- ThroughVCCpin+5VDCsupplyisprovidedtotheIC.

## INTERNALARCHITECTUREOF8086MICROPROCESSOR

Thetotalinternalarchitectureof8086microprocessorcanbebasicallydividedintotwodifferent units.

***1.BusInterfacesUnit(BIU)***

***2.Execution Unit (EU)***



## 1.BusInterfacesUnit(BIU):

- Itisresponsiblefortransferofdataandaddressbetweentheprocessor,memoryand input output device.
- It receives the data from the IO device and stored the data in a 6 bytes instruction queues in FIFO sequence and this data is transferred to the execution unit for arithmetic and logical operation.
- Thefunctionofdifferentunitsofbusinterfaceunitare

### 6bytesinstructionqueue

- Itsfunctionistoreceive6no.of8bitdataatatimeandstoresthedatainitand then this data can be transferred to execution unit for performing arithmetic and logical operation i.e. execution operation.
- The data is received from IO device to the 6 bytes instruction queue in FIFO sequence.

### Segmentregister

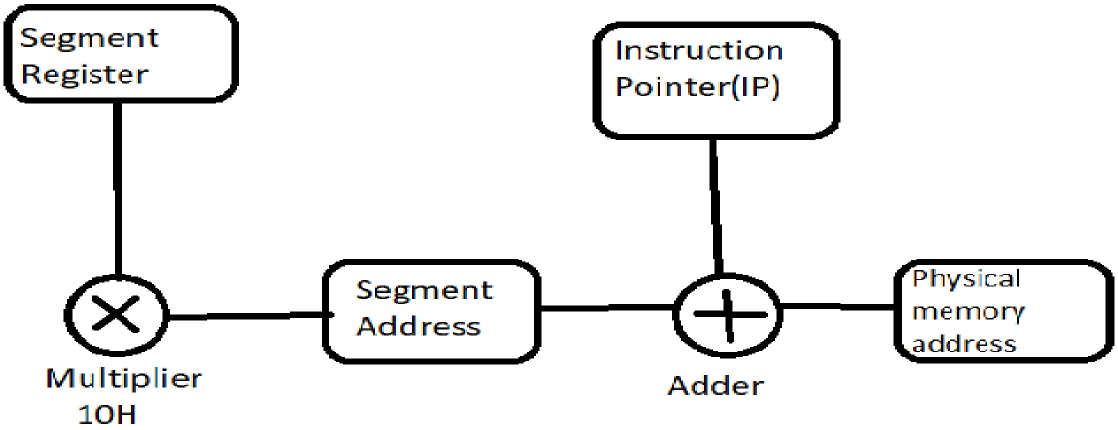
- Thereare4segmentregister
  - Code segment register (CS): It basically used to store the opcode of aninstruction.
  - Data segment register (DS): It basically used to store the operand of an instruction.
  - Extra segment register (ES): It is basically used to store the character or string instruction such as consonant, vowel, character etc.
  - Stacksegmentregister(SS):Stackisasetofmemorylocationwhose address is different from main memory address.
    - ✓ We can transfer the data from main memory to stack memory by pushinstructionandwecanreceivethedatafromstackmemoryto main memory by pop instruction. So, segment register is basically used to store the stack memory value.
    - ✓ To locate a particular memory address, we take the help of stack pointer and given by the commandLXI SP 9605$_H$.

### Instructionpointer(IP)

- Itsfunctionissameasthat ofprogram counterof8085microprocessorand is basicallyused tocheckwhethertheaddressfornext instructionisavailableor not. So, it stores the OFF-SET address.

### Buscontroland addressgeneration

- It is basically used to generate 20 bits effective memory address or physical memory address.
- One address is generated from the segment register which is of 16 bit and when the address goes to the bus control and address generation it gets multiplied by the multiplier circuit of value $10_H$. So, at the output we get a segment address of 20 bit.
- Another address is generated from the instruction pointer which is of 16 bit and is known as OFF-SET address or assembly line address.
- So, whenthis address goes tothe bus control and address generation, it gets added ofwiththehelp ofadder circuit whichis present insidethebus control and address generation.
- Hencethe 20-bitsegmentaddressisaddedupwiththe16-bitIPaddresswith the help of adder circuit and at the output we get a 20-bit effective memory address or physical memory address.
- ***Effectivememoryaddress(EMA)orPhysical memoryaddress(PMA)=Segment address*10$_H$ +Instruction pointer***
- Theblockdiagramofphysicaladdressgenerationisshownasfollows.



***[Blockdiagramofphysical address generations]***

## 2.ExecutionUnit(EU):

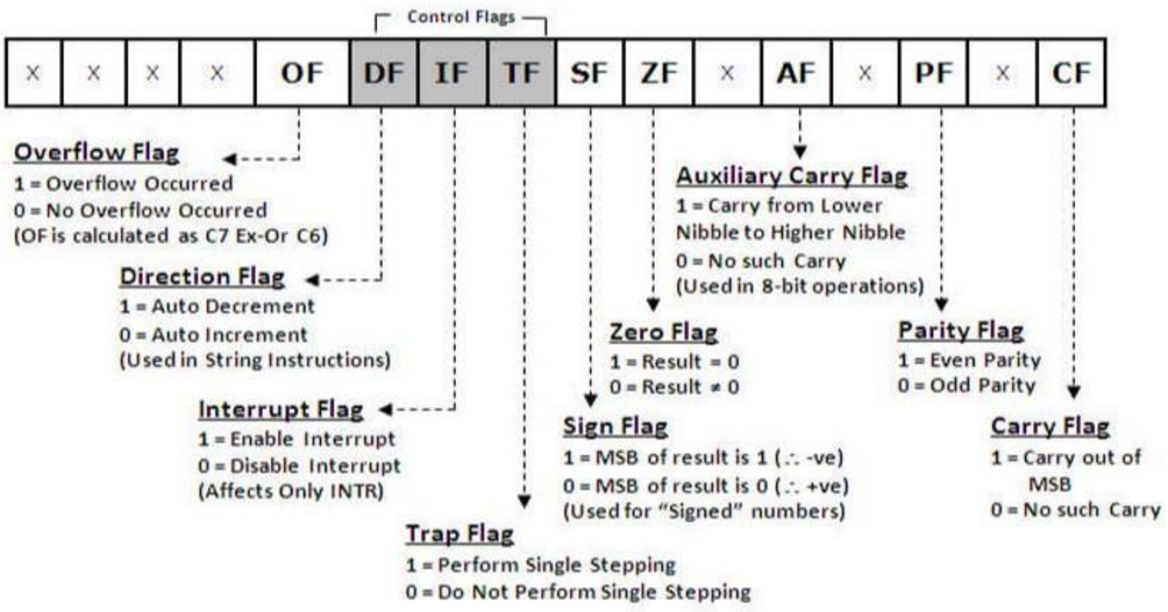
- Theexecutionunitreceivestheopcodeofaninstructionfromthe6byteinstruction queuedecodesitandperformthearithmeticandlogical operationandstores the result.
- Thefunctionofdifferentunitsofexecutionunitsare

## ALU

- It is basically used to perform the arithmetic and logical operation such as addition, subtraction, multiplication, division, increment, decrement, comparison.
- Afterthearithmeticandlogicaloperation,theresultischeckbythestatusflag.

## StatusFlag

- Thestatusflagsarebasicallyusedtocheckwhethertheresultiswritingorwrong.
- Accordingly,in8086microprocessorthereare9activestatusflagsoutofwhich6are conditional flag and 3 are control flag.



- **Conditionalflag:**Conditionalflagsarethosewheretheoutputdependsuponthe input.Theconditionalflagsare*Signflag,zeroflag,Auxiliaryflag,Parityflag,Carry flag and overflow flag.*
  - **Sineflag:**Afterthearithmeticoperationiftheresultisnegativethensign flag is tends to logic 1 otherwise it will tend to logic 0.
  - **Zeroflag:**Afterthearithmeticoperationiftheresultiszerothenzeroflag will tend to logic 1 otherwise it will tend to logic 0.
  - **Auxiliaryflag:**Afterthearithmeticoperationifthereisacarryfrom3rdto 4thbitthenauxiliarycarrywilltendstologic1otherwiseitwilltendtologic 0.
  - **Parityflag:**Afterthearithmeticoperationiftheresultofthesumcontains evenno. of1'sthenparityflagwilltendtologic1otherwiseitwilltendto logic 0.

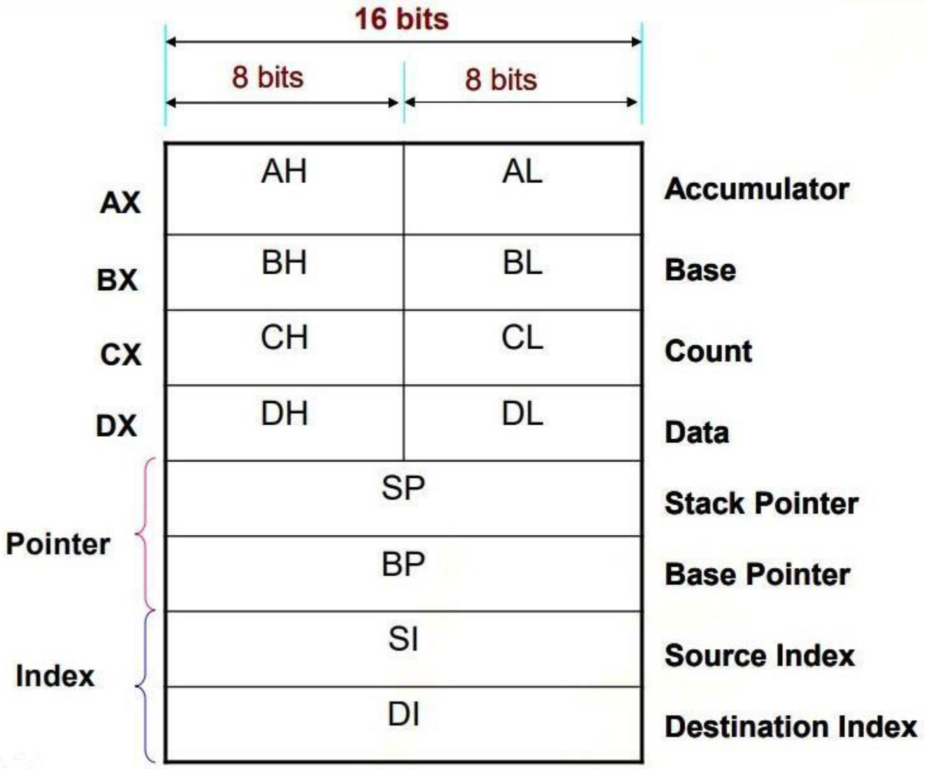
- **Carry flag :** After the arithmetic operation if the result is more than 8 bit thentherewillbeacarryfrom7-8bitsocarryflagwilltendstologic1and incaseof16bitoperationifthereisacarryfrom15to16bitthecarryflag will tends to logic 1 otherwise it will tends to logic 0.
- **Overflowflag:**Afterthearithmeticoperationiftheresultismorethan16 bits,in that case the data cannot be stored in accumulator or destination register.So,inthatcasetheoverflowflagwilltendstologic1otherwiseit will tend to logic 0.

So,inconditionalflagthe outputisdependsontheinput

- **Controlflag:**Thecontrolflagsarebasicallyusedforcontrol-orientedactivitysuch as to stop the program, set or reset the operation, status flag manipulation, interruptoperation.So,inthiscaseitisusedforcontrol-orientedactivityandhere theoutputisdoesnotdependsuponinput. ThecontrolflagsareDirectionalflag, Interrupt flag, Trap flag.
  - **Directionalflag:**Incaseofcharacterorstringoperationinthatcasethe directional flag will tends logic 1 otherwise it will tend to logic 0.
  - **Interrupt flag and Trap flag:** These two flags are basically used as interrupt operation.

**Resistors**

- There are two types of register in 8086 microprocessor such as General-purposeregister and Special purpose register.



- **General-purpose register:** The general-purpose register are $A_H$ and $A_L$ i.e. A higher order and A lower order. Similarly, $B_H$ and $B_L$, $C_H$ and $C_L$, $D_H$ and $D_L$ respectively. Eachregister canstoreindividually 8bit of dataand combine form it can store 16bit of data so,

  $A_H+A_L=A_X$***(16bit)*** $B_H$
  $+ B_L = B_{X(}$***16 bit)*** $C_H$
  $+ C_L = C_{X(}$***16 bit)***
  $D_H+D_L=D_{X(}$***16bit)***

  e.g.***MOV AH,08***$_H$-Moveimmediatelythedata08to$A_H$register.

  ***MOVAL,08***$_H$-Moveimmediatelythedata08to$A_L$ register.

***MOVAX, 1264***$_H$-Movethe16-bitdata1264HtoA$_x$register.

- Since these registers are commonly used for storing the data temporarily for any arithmetic and logical operation so these are known as general-purpose register.

- **Special purpose register:** The special purpose registers are Stack pointer (SP), Base pointer (BI), Source index register (SI),Destination index register (DI). These are known as special purpose register because they are used for some specific operation and these are 16-bit registers.

  - **Stack pointer (SP):** Through stack pointer we can locate to a particularstack pointer address e.g**. LXI SP, 9505$_H$.**
  - **Base pointer (BP):** It is basically use to store the OFF-SET address (value of instruction pointer address).
  - **Sourceindexregister(SI):**Itisbasicallyusedtostorethestringaddress e.g.***MOVSI,[2000***$_H$***]***-MovetheOFF-SET address2000$_H$toSI register.
  - **Destination index register (DI):** It is basically used to store the end address e.g. **MOV DI, [2005$_H$]**-Move immediately the OFF-SET address 2005$_H$ to DI register.

# INSTRUCTIONSETSOF8086MICROPROCESSORS

Instructions are sets of commands given to the processor to perform a specific operation accordinglyin8086microprocessorthetotalinstructionsetcanbedividedinto8differenttypes according to type of operation it performs. The instructions are;

1. Datatransforminstruction
2. Arithmeticinstruction
3. Logicalinstruction
4. Branchcontrolinstruction
5. Iterationcontrolinstruction
6. Interruptinstruction
7. Processorcontrol
8. Stringinstruction

1. **Datatransforminstruction:**Itisbasicallyusedfortransferringthedatafromone register to another register or register to memory without changing the content. E.g.MOVAX,BX-MovethecontentofBXregistertoAXregister. MOV AX,$1234_H$-Move the 16-bit data $1234_H$ to $A_X$ register.
2. **Arithmeticinstruction:**Thesearebasicallyusedforarithmeticoperationsuchas addition, subtraction, multiplication, division, increment, decrement etc. In this case the final result may change.
3. **Logical instruction:** These instructions are basically used for logical operation such as ANA (AND), ORA (OR), XRA (XOR), CMP (COMPARE) etc.
4. **Branch control instruction:** The instruction under this group are basically used for conditional or unconditional jump operation.
   E.g.
   JNZ- Jump if the counter data in the register C is not zero to label loop.
   JNC- Jump if there is no carry or borrow to label loop.
5. **Iterationcontrolinstruction:**Theseinstructionsarebasicallyusedinlabelportion for loop operation.
   E.g.LOOP,CALL,AHEAD,STOP.
6. **Interrupt instruction :** The instruction under this group are basically used for interrupt operation that means to stop the current program and jump to some other program which is required at the specific moment of time and after the completion of that specific program the processor can again return back to the originalprogramandthesearegivenbythroughcertainsoftwareinstructionsuch as INT, $INT_0$, $INT_1$, $INT_2$, INTR.

7. **Processor control:** The instruction under this group are basically used for status flag manipulation and for machine control operation.

   E.g.*CLC*-Clearthecarry bit.

   *CLD*-Clearthedirectional flag.

   *STC*-Setthecarry bittologic1.

   *STD*-Setthedirectionaltologic1.

   *CMC*-Complementofcarrybit.

   *HALT*

   *PUSH*

   *POP*

8. **Stringinstruction:**Stringisaseriesofbytesorseriesofwordstoredinasequential memory location.

   In 8085 microprocessor there are certain instructions which are basically used to move or store a string of data bytes.

   E.g.*MOVS*-Movethestring(singlebit). *MOV SP*- Move the string of data bytes. *CMP S*- Comparison of string.

   *CMPSP*-Comparisonofstringofdatabytes.

## ADDRESSINGMODEOF8086MICROPROCESSOR

Itisthetechniquethroughwhichwearespecifyingdataforoperationorhowtheoperand data is specified accordingly in 8086 microprocessors. There are 8 different addressing mode according to the type of operation it performs.

1. Registeraddressingmode
2. Immediateaddressing mode
3. Directaddressingmode
4. Registerindirectaddressingmode
5. Baseaddressingmode
6. Indexaddressingmode
7. Baseindexaddressing mode
8. Baseindexand displacement

1. **Register addressing mode:**In this type ofaddressingmode the operanddata is not directly specified in the instruction itself but it is specified by some register.

   E.g.*MOV AX,BX*-MovethecontentofBXregistertoAX register.

   *MOVAH,AL*-Movethecontent ofALregistertoAHregister.

2. **Immediateaddressingmode:**Inthistypeofaddressingmodetheoperanddatais directly specified in the instruction itself.

   E.g.*MOV AH,$08_H$*- Moveimmediatelythedata$08_H$toAH register.

   *MOVAX,$1264_H$*-Movethe16-bit data$1264_H$toAX register.

3. **Directaddressingmode:**Indirectaddressingmodetheoperandaddressisdirectly specified in the instruction itself.

   E.g.

   *MOVAX,[$9000_H$]*-Movethecontent ofmemoryaddress$9000_H$toAX register.

   *MOVBX,[$9001_H$]*-Movethecontentof memoryaddress$9001_H$toBXregister.

4. **Register indirect addressing mode:** In this typeof addressing modethe operand data is not directly transferred to the accumulator; at first it stored in some memory address and the transferred to the accumulator.

   E.g.*MOVBX,[9001] MOV*
   *AX, BX*

5. **Base addressing mode:** In this type of addressing mode the operand address is one of thecontentsofbasepointerorstackpointer.Basically,itstorestheOFF-SETvalueofthe instruction pointer.

   E.g. *MOV BX, [$9005_H$]*- Move the content of the memory address $9005_H$ to BX register base pointer.

6. **Index addressing mode:** It is basically used to store the string address or the end address. The source index register is used to store the starting address and destination index register is used to store the end address.

   E.g.

   *MOVSI,[$9000_H$]*-Movethecontentofmemoryaddress$9000_H$totheSIregister.

   *MOVDI,[$9005_H$]*-Movethecontentofmemoryaddress$9000_H$to DI register.

7. **Baseindexaddressingmode:**InthismodetheoperandOFF-SETisthesumofthe content of base register or BX and index register SI or DI.

   E.g.

   *MOVAX,[BX+SI]*-Movethecontentofbasepointerorstackpointerandthe source index value to AX register or accumulator.

   *MOV AX,[BX+DI]*- Move the sum of the content of base register and destination index register value to accumulator.

8. **Baseindexanddisplacement:**InthisaddressingmodetheoperandOFF-SETisthesum of the content off base pointer + SI or DI + 8-bit or 16-bit displacement value.

   E.g.*MOVAX,[AX+SI+$08_H$]*

# PROGRAMIN8086 MICROPROCESSORS

## Programtofindlargestno.ofdata array:-

| LABEL | MNEMONICS | OPERAND | COMMENT |
|---|---|---|---|
|  | MOV | AX, 0000 | Movethe16-bitdata0000Hto accumulator or AX register. |
|  | MOV | SI, [7400] | Movethecontentofthememory address 7400H to SI register. |
|  | MOV | CX,SI | MovethecontentofSIregistertoCX registerwhichisbasicallyusedto store the counter data. |
| BACK | INC | SI | Incrementthecontent ofsource indexregisterbynextbittoprovide thenextcounterdata. |
|  | INC | SI | Increment the content of the source index address by next bit to provide the next value. |
|  | CMP | AX, (SI) | Compare the content of AX register withSIregisterwhereAXactsas accumulator in 8086 microprocessors. |
|  | JA | NEXT | Jumpiftheno.inAXregisteris greaterthan SIregistertolevelnext. |
|  | MOV | AX, (SI) | MovethecontentofSIregistertoAX register. |
| NEXT | LOOP | BACK | Go to the label back till the counter datainSIregisterasreachthe highestvalue. |
|  | MOV | (7565),AX | Move the content of AX resistor to memory address resister tomemory address. |
|  | INT | 2F |  |
|  | END |  |  |

DATA:

|  |  |  |  |  |
|---|---|---|---|---|
| 7400-05 | 7403-83 | 7406-39 | 7409-84 | RESULTAX=9630 |
| 7401-00 | 7404-58 | 7407-46 | 740A-30 |  |
| 7402-41 | 7405-72 | 7408-53 | 740B-96 |  |

## Programtofindsmallestno.fromadataarray:-

| LABEL | MNEMONICS | OPERAND | COMMENTS |
|---|---|---|---|
| | MOV | AX,FFFF | MovethehighestvalueFFFF$_H$toAX register. |
| | MOV | SI, [7400] | Move the content of memory address 7400$_H$ to SI register. |
| | MOV | CX, (SI) | Move the content of SI register to CXregisterwhichisbasicallyused forcounteroperation. |
| BACK | INC | SI | IncrementthecontentofSIbynext bit to provide the next counter data. |
| | INC | SI | IncrementthecontentofSIaddress bynextbittoprovidethenextdata. |
| | CMP | AX, (SI) | ComparethecontentofAXregister with the SI register. |
| | JB | NEXT | Jumpwithborrowandcarry. |
| | MOV | AX, (SI) | Move the content of SI register to AX register. |
| NEXT | LOOP | BACK | Gotothelabelbacktillthecounter dataofSIregisterorCXregisterhas reach the highest value. |
| | MOV | (7565),AX | MovethecontentofAXregisterto the memory address 7565$_H$ and 7566$_H$. |
| | END | | |

DATA:

| | | | |
|---|---|---|---|
| 7400-05 | 7406-39 | 7565-39 | RESULTAX=4639 |
| 7401-00 | 7407-46 | 7566-46 | |
| 7402-41 | 7408-53 | | |
| 7403-83 | 7409-84 | | |
| 7404-58 | 740A-30 | | |
| 7405-72 | 740B-96 | | |

## MINIMUMMODECONFIGURATIONOF8086MICROPROCESSOR

Whenthe single processor is used, thenthe 8086 microprocessor operates in minimum mode MN/MX’ tends to logic 1 i.e. MN=1, MX’=1, MX=0

So, for simple input output operation or when a single processor is used then the 8086 microprocessor operates in minimum mode and in that case the control signal pins which are attachtominimummodegetactivated.ThecontrolsignalpinsareRD’,WR’,ALE,IO/M’,HOLD, HLDA, DEN’.

The different units which are attached to the minimum mode control pin to perform different memory read and IO read, IO write operation is known as minimum mode configuration.

Thedifferentunitsare:

i. **Latches:**
   - Thereare2to3latchesarepresentandtheselatchesareoctalinnaturethatmeans each latchcan stores8 bitsofdatasothe twolatches canstore16 bitofdata and 3 latches can stores 20 bits of address.
   - Latchesaretemporarystoragedeviceor flipflop
   - The ALE signal is connected to the latches and its function is to separate the address from address and data bus.
   - The bus high enable pin is also attach to the latches and its main function is to check the validity of address and the address and data line are also attached to latches.

ii. **Transreceiver:**
   - Transreceivermeanstransferandreceptionofdata.
   - Through the trans receiver we can transfer the data from the processor to the IO device and in that case the DT/R’ tends to logic 1 i.e. DT=1, R’=1, R=0.
   - IfDT/R’tendstologic0thenDT=0,R’=0,R=1soreadoperationisperformi.e.data is received from the IO device to processor.
   - Data enable pin is also attached to the trans receiver and its main function is tocheck the validity of the data.

iii. **RAM:**
   - Itisknownasrandomaccess memory.
   - Itstemporarystoresthedataanditvolatileinnaturethatmeansifthepower supply OFF the data gets deleted.
   - Herawecanperformbothread and writeoperation.

iv. **EPROM:**

- EPROM-ErasableProgrammableReadOnlyMemory.
- Itisnon-volatileinnature andisusedtostoresthelibraryfunction.
- Itisonlyusedforreadoperation.

v. **IOdeviceorperipheraldevice:**

- Itisbasicallyusedforprovidinginputdatatotheprocessor.
- In minimum modesingle processorisusedsoasingle IOdevice isconnectedto processor.

vi. **Clockgenerator**:

- It is basically used for providing input clock frequency to the processor so that we can check how much time the processor will take to execute an operation.
- Itcanvaryfrom5MHz to10MHz.

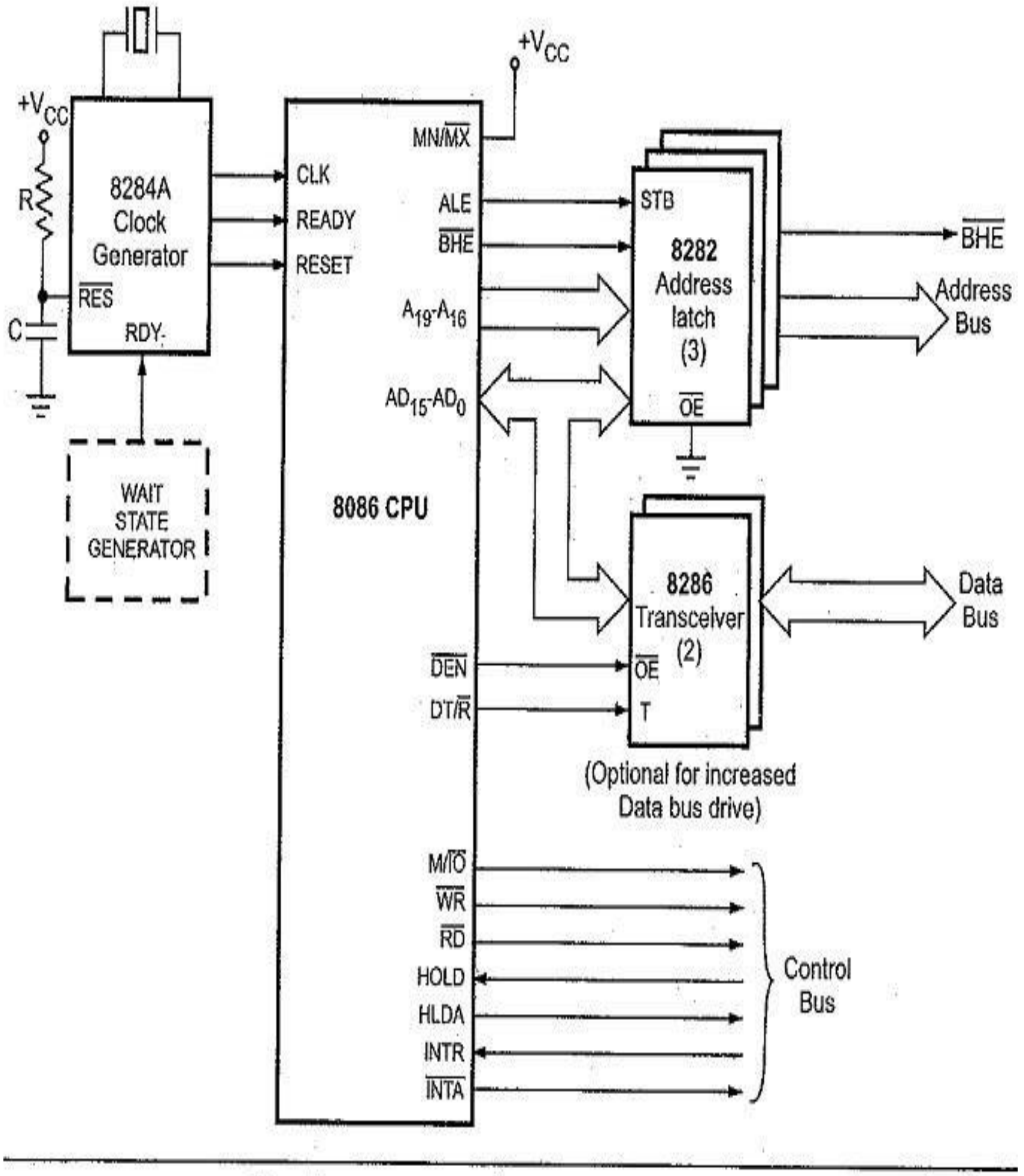


Fig. 10.2 Typical minimum mode configuration

## Timingdiagramofminimummodeoperationof8086microprocessor:

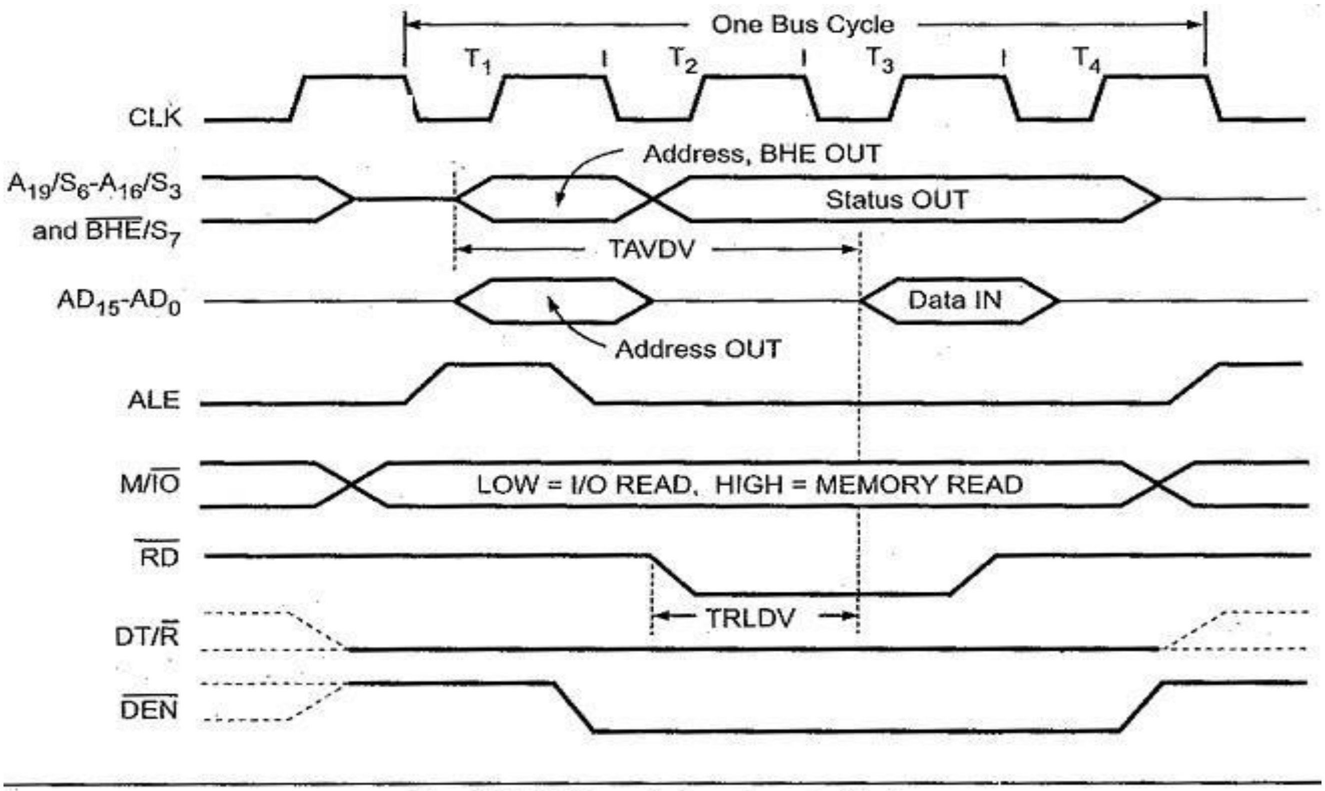


Fig. 10.7 (a) Input (read operation)

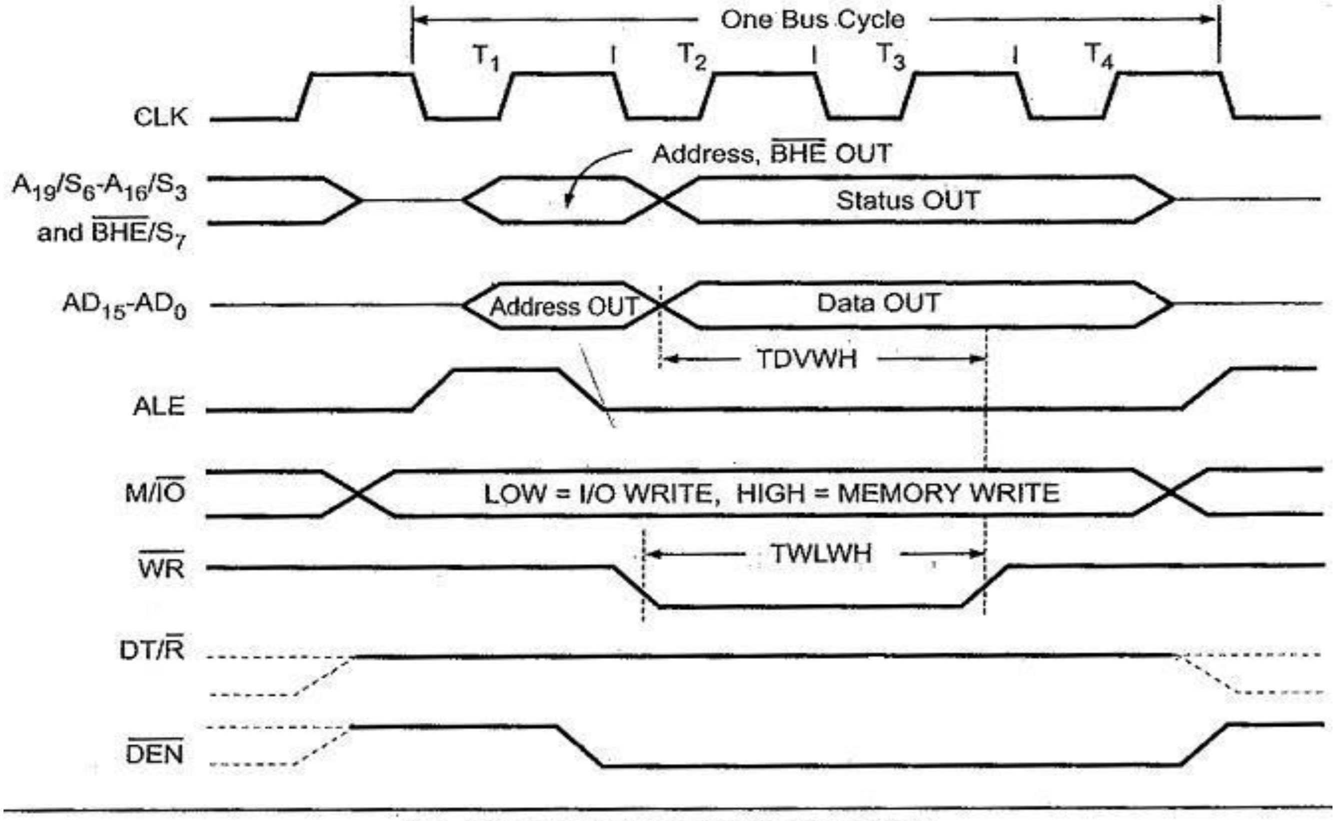


Fig. 10.7 (b) Output (write operation)

# MAXIMUMMODECONFIGURATIONOF8086MICROPROCESSOR

Formultipurposeoperation the 8086microprocessors can operatesin maximum mode and in that case the MN/MX' pin tends to logic 0 i.e. MN=0, MX'=0, MX=1.

Inmaximummodeoperationmultiple RAM,multipleROM,multipleIOdevicesareattachedto the processor to perform different memory read and memory write and IO read and IO write operation.

In maximum mode operation bus controller i.e. 8288 is used because the control signal which arerequiredtoperformdifferentreadwriteoperationsuchasRD',WR',IO/M'arenotavailable in maximum mode. So, if we are using a bus controller then all the control signal which are required to perform different read write operation are resent in this bus controller.

Through this bus controller we can connect to multiple or different RAM, ROM, IO devices to performdifferentreadwriteoperationandthisbuscontrollerisindirectlycontrolbythestatus signal pin i.e. $s_0$', $s_1$', $s_2$' of maximum mode control pins.

| $S_2$ | $S_1$ | $S_0$ | Status |
|---|---|---|---|
| 0 | 0 | 0 | Interruptacknowledgement |
| 0 | 0 | 1 | I/ORead |
| 0 | 1 | 0 | I/OWrite |
| 0 | 1 | 1 | Halt |
| 1 | 0 | 0 | Opcodefetch |
| 1 | 0 | 1 | Memoryread |

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | Memorywrite |
| 1 | 1 | 1 | Passive |

The different units which are attached to the maximum mode pins to perform different read write operations are

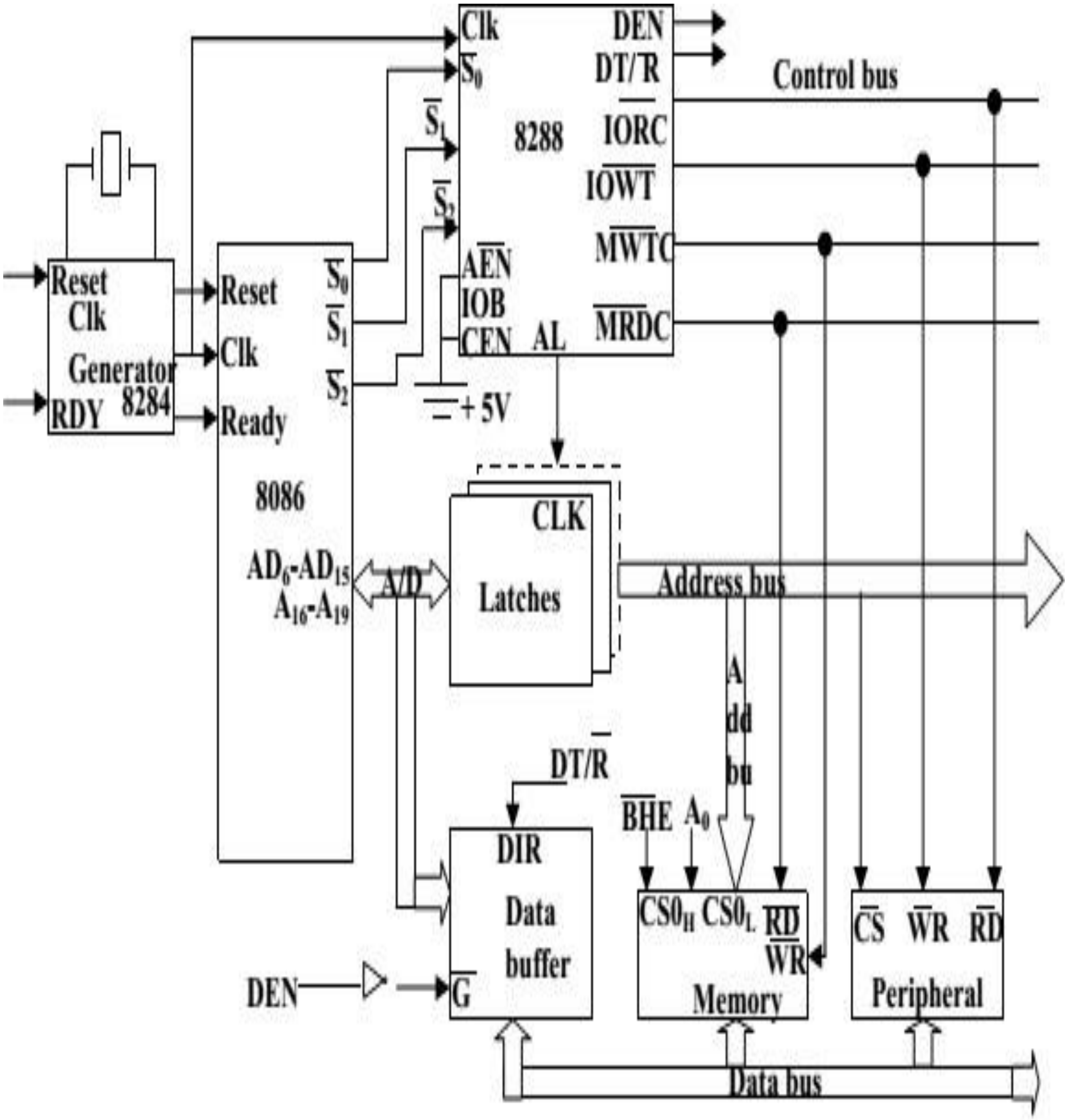
1. **Multiplelatches:**
   - Latchesaretemporarystoragedeviceandactsasa flipflop.
   - Multiplelatchesareusedtoseparatelystorestheaddress andthedata.
2. **Transreceiver:**
   - Basically, used for transfer and reception of data to multiple RAM, multiple ROM and multiple IO devices.
3. **8288buscontrollers:**
   - Thesebuscontrollersarebasicallyusedforperformingdifferentread,writeandIO operationwiththehelpofcontrolsignalsuchasMRDC',MWTC',IORC',IOWC'etc.
4. **MultipleRAM:**
   - Itisknownasrandomaccess memory.
   - Itstemporarystoresthedataanditvolatileinnaturethatmeansifthepower supply OFF the data gets deleted.
   - Herawecanperformmultiplereadand writeoperation.

5. **MultipleEPROM:**
   - EPROM-ErasableProgrammableReadOnlyMemory.
   - Itisnon-volatileinnatureandisusedto storesthelibraryfunction.
   - Itisonlyusedformultiplememoryreadoperation.

6. **MultipleIOdevices:**
   - Itisbasicallyusedforprovidingmultipleinputdatatothe processor.

- InmaximummodemultipleprocessorsisusedsoamultipleIOdeviceisconnected to processor.

7. **Clockgenerator:**
   - Itisbasicallyusedfor providinginputclockfrequency tothe processorso thatwe can check how much time the processor will take to execute an operation.
   - Itcanvaryfrom5MHz to10MHz.



Maximum Mode 8086 System.

## Timingdiagramof8086microprocessor

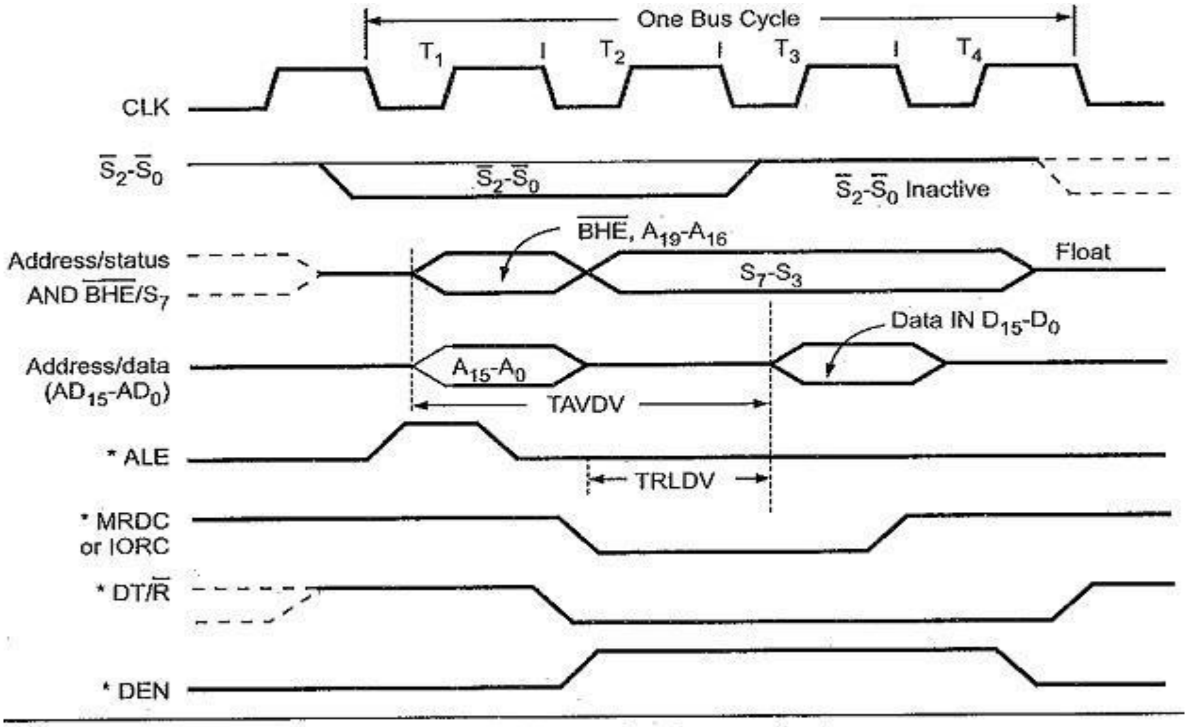


**Fig. 10.10 (a) Input (read operation)**

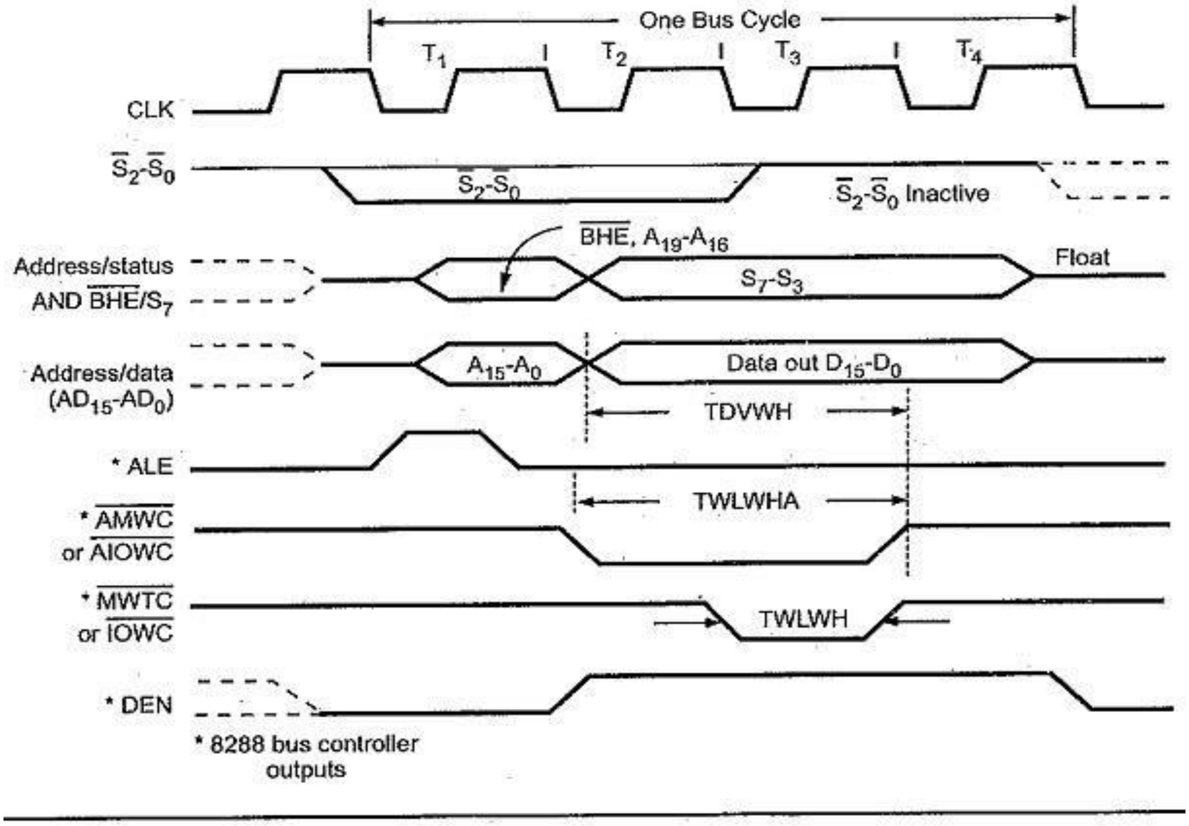


**Fig. 10.10 (b) Output (write operation)**

## INTERRUPTSTRUCTUREOF8086MICROPROCESSORS

The 8086 gets the new values of CS and IP register from four memory addresses. When it responds to an interrupt, the 8086 goes to memory locations to get the CS and IP values for the start of the interrupt service routine. In an Interrupt Structure of 8086 system the first 1 Kbyte of memory from 00000H to 003FFH is reserved for storing the starting addresses of interrupt service routines. This block of memory is often called the **Interrupt Vector Table in 8086**or the**interrupt pointer table**. Since 4 bytes are required to store the CS and IP values foreachinterruptserviceprocedure,thetablecanholdthestartingaddressesfor256interrupt service routines.

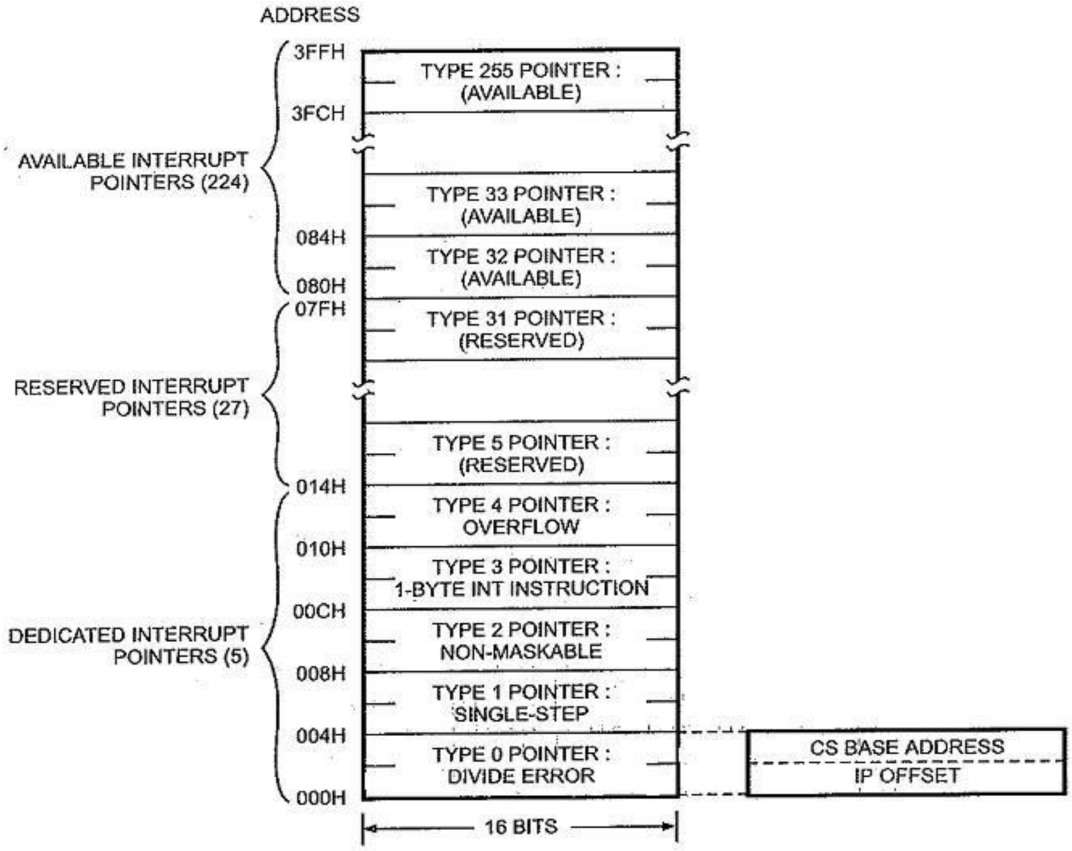

ADDRESS
3FFH
3FCH
084H
080H
07FH
014H
010H
00CH
008H
004H
000H
AVAILABLE INTERRUPT POINTERS (224)
RESERVED INTERRUPT POINTERS (27)
DEDICATED INTERRUPT POINTERS (5)
TYPE 255 POINTER : (AVAILABLE)
TYPE 33 POINTER : (AVAILABLE)
TYPE 32 POINTER : (AVAILABLE)
TYPE 31 POINTER : (RESERVED)
TYPE 5 POINTER : (RESERVED)
TYPE 4 POINTER : OVERFLOW
TYPE 3 POINTER : 1-BYTE INT INSTRUCTION
TYPE 2 POINTER : NON-MASKABLE
TYPE 1 POINTER : SINGLE-STEP
TYPE 0 POINTER : DIVIDE ERROR
CS BASE ADDRESS
IP OFFSET
16 BITS

**Fig. 9.2 8086 interrupt vector table**

Each interrupt type is given a number between 0 to 255 and the address of each interrupt is foundbymultiplyingthetypeby4e.g.fortype11,interruptaddressis11x4 =$44_{10}$=0002CH

Onlyfirstfivetypeshaveexplicitdefinitionssuchasdividebyzeroandnon-maskableinterrupt. Thenext27interrupttypes,from5to31,arereservedbyIntelforuseinfuturemicroprocessors. The upper 224 interrupt types, from 32 to 255, are available for user for hardware or software interrupts.

Fivetypesofdefinedinterrupts are:

1. TYPE-0orDividedbyzeroerrorinterrupt
2. TYPE-1orSinglestep interrupt
3. TYPE-2orNon-maskableinterrupt
4. TYPE-3orBreakpointinterrupt
5. TYPE-4orOverflowinterrupt

1. **TYPE-0orDividedbyzeroerrorinterrupt:**Incaseofdivisionoperationifthequotient value is very large or more than the destination register value or more than 16-bit so in thatcasethedividedbyzerointerruptwillgetactivatedtoindicatethattheresultismore than the destination register value and the result cannot be stored in accumulator.
2. **TYPE-1 or Single step interrupt:** If this interrupt is activated then it will check the program step by step so it acts as an interpreter.
3. **TYPE-2or Non-maskable interrupt:** If this interrupt is activatedthenincase of system power failure or AC power failure then the data will not get deleted.
4. **TYPE-3orBreakpointinterrupt:**Iftheinterruptisactivatedthentheprocessorwillstop the current program and jump tosome other program whichis required at that specific moment of time after the completion of that specific program. The processor can again return back to the original program.
5. **TYPE-4 or Overflow interrupt:** In case of any type of arithmetic operation such as multiplication,addition,subtractionetc.iftheresultismorethanthedestinationregister value or more than 16-bit or the result is very large, in that case the overflow interrupt willgetactivatedandtends tologic1 toindicatethat the resultisverylarge andcannot be stored in accumulator.

**Multiplechoicequestionanswer:**

1. The 8086 up is ________bitsprocessor.

   A) 8bits

   B) 10bits

   C) 16bits

   D) 32bits

   Answer: C

2. The8086uphasa_______byteinstructionqueue.

   A) 6byte

   B) 2byte

   C) 4byte

   D) 8byte

   Answer:A

3. Thetotalinternalarchitectureof8086upisdividedinto

   A) ALUandTimingcontrol unit

   B) Businterfaceunitandexecutionunit

   C) InstructionqueueandAlu

   D) SegmentregisterandIP.

   Answer: B

4. TheBuscontrol andaddressgenerationgenerates

   A) 20-bitaddressline.

   B) 16-bit

   C) 24-bit

   D) 12-bit.

   Answer:A

5. The no.Ofaddressingmodesin8086upis

   A) 8.

   B) 5.

   C) 6.

   D) 4.

   Answer:A

6. The no.Ofinstructionsetin8086upis

   A) 8.

   B) 6.

   C) 5.

   D) 4

Answer:A

7. The no.Ofstatusflagsin8086 upis

A) 6.

B) 8.

C) 9.

D) 5.

Answer:C

8. Thestatusflagof8086 upisdividedinto.

A) conditionalflagandcontrolflag

B) conditional flag and trap flag.

C)conditionalflagandinterruptflag.

D)controlflagandinterruptflag.

Answer: A

9. Theno.Ofconditionalflagsare.

A)4.

B) 5.

C) 8.

D) 6.

Answer:D

10. Theno.Ofcontrolflagsare.

A)3.

B) 5.

C) 6.

D) 8.

Answer:A

11. Theno. ofinterruptsin8086upis

A) 256

B) 255

C) 200

D) 5.

Answer:A

12. TheNMIinterruptis

A) TYPE 2

B) TYPE1.

C) TYPE 3

D) TYPE O.

Answer:A

13. The single step interrupt is ________.

A) TYPE 0.

B) TYPE2.

C) TYPE 1.

D) Noneofthese

Answer: C

14. The Dividebyzeroerror interrupt is ____________.

A) TYPEO

B) TYPE1

C) TYPE 2

D) TYPE 3

Answer:A

15. TheBreakpointinterruptis

A) TYPE 3.

B) TYPE0.

C) TYPE 1.

D) TYPE 2.

Answer:A

16. Theaddresslineof8086 upis

A) 20bit.

B) 16bit.

C) 24bit.

D) Noneofthese

Answer: A

17. The8086up operatesat

A) 12VDC.

B) 5V DC.

C) 220VAc.

D) Noneofthese

Answer: B

18. The minimum mode is used ______.

A) multimodeoperation

B) complexoperation.

C) morethan one processorisattached.

D) simpleinputoutputoperation.

Answer: D

19. Themaximummodeisusedfor___________.

A) simpleinputoutputoperation.

B) multipurposeoperation.

C) logicaloperation.

D) noneofthese.

Answer: B

20. Theinstructioncycleconsistof____________.

A) executioncycle.

B) opcodefetchcycleandexecutioncycle.

C) opcodefetchcycle

D) noneofthese

Answer: B

# ASSIGNMENTFULL MARKS-60

## SECTION-A

**Shortanswer type question:** **2*4=8**

a. Whatarethemaximum mode controlsignalpins?
b. Whatarethegeneral-purposeregistersof 8086microprocessor?
c. Whatisstack and stack pointer?
d. Howmanystatusflagsarein8086microprocessorsamongthem?What arethe conditional flags and control flags?

## SECTION-B

**Q.2Focusedanswertypequestion:** **6*6=36**

a. Explainthe minimummode configurationof8086microprocessor.
b. Explain the opcodefetchcycleof minimummodeconfiguration.
c. Explain themaximum mode configurationof 8086microprocessor.
d. Write theprogramtofindthelargestno.fromthedataarrayusing8086 microprocessors.
e. Explaintheinterruptsof8086microprocessor.
f. Explaintheaddressingmodesof8086microprocessorwithsuitableexamples.

## SECTION-C

**Q.3Longanswertypequestion:** **16*1=16**

a) Explain the internal architecture of8086microprocessorwithsuitablediagram.Write a program to find smallest no.from data array using 8086 microprocessors.

OR

b) Explain the instruction set of 8086 microprocessors. Draw the timing diagram of read cycle of minimum mode configuration for 1byte instruction.

# MODULE-3:INTERFACINGDEVICES

**Introduction:**

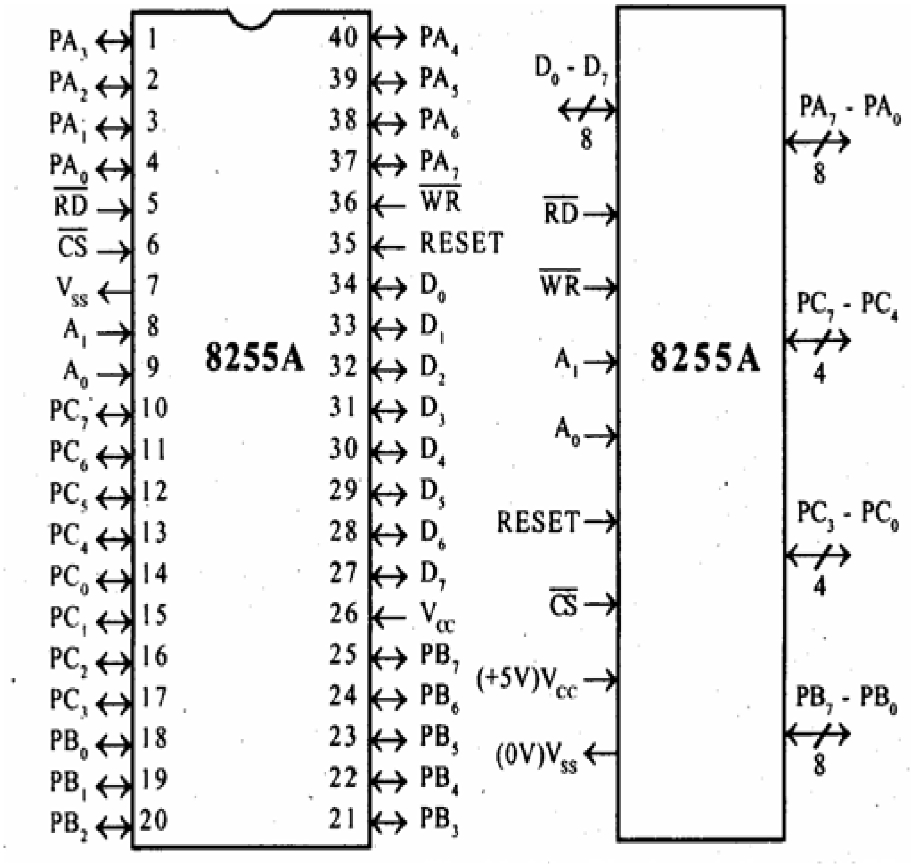
Themainfunctionofinterfacingdeviceisthat,

- Itmatchestheprocessor’sspeedwithIOdevicespeedsothatourdatawillnotgetloosed. Normallytheprocessor’sspeedisfastandIOspeedislowsoifweareusinganinterfacing device then it matches the processor speed with IO device speed.
- It obeys the ASCII (American Standard Code for Information Interchange) or alpha numeric code.
- Itactsasvoltageregulatorsothatour appliancewillnotget brunt.
- Through the interfacing device instead of transferring the data directlyto the processor we can indirectly store in the interfacing device and we can transfer it to the processor. So, in this way we can avoid over burden of the processor.
- Through the interfacing device we can connect multiple IO device so in this way we can receive data from multiple IO devices and then we can transfer into the processor.
- Thevariousinterfacingdevicesare:
    1. 8255PPI(ProgrammablePeripheralInterface)
    2. 8257DMA(DirectMemoryAccess)
    3. 8259PIC(PriorityInterruptController)
    4. 8251USART(UniversalSynchronousAsynchronousReceiverTransmitter)

## 1. 8255PPI (ProgrammablePeripheralInterface):

- Itconsistsof40 pinICchipandoperates at+5VDC.
- It is known as programmable peripheral interface because through certain control word bit which is present inside the read write logic section of PPI.
- We can control the ports and mode of operation that means we can make any port as input or output port and through this port we can transfer the data from the processor to the external device or we can receive the data from the IO device to the processor.

**Pindiagramof8255 PPI:**



## Pindescriptionof8255PPI:

**Port:**Outof40pins,24pinsareusedforports.Thereare3portsthroughwecantransferand receives data.

a) **PortA($P_{A0}$-$P_{A7}$):**Pinno.-3,2,1,40,39,38,37,36.
b) **PortB($P_{B0}$-$P_{B7}$):**Pinno-18topinno-25
c) **PortC:**
   i. PortCupper($P_{C4}$-$P_{C7}$):Pinno-13toPinno-10
   ii. PortClower($P_{C0}$-$P_{C3}$):Pinno-14toPinno-17

**RD':** Pinno-5isusedforreadoperationif RD'=0,RD=1.

**CS':** Pin no-6 is used as cheap selection pin. If CS'=0, CS=1, then the control signal pins are get activated.

**GND:**Pinno-7isusedasgroundpin.

**$A_1$& $A_0$:** Pin no-8 & pin no-9 is used as $A_1$&$A_0$ respectively. These 2 pins are controlsignal pins used for controlling the ports and modes operations.

**$V_{CC}$:**Pinno-26isused assupplypin.+5VDCissupplytotheICthrough$V_{CC}$pin.

**D0toD7:**Pinno-34topinno-27isusedasdatalinepins(D0toD7)whichisof8-bitsand these pins are bidirectional through which we can transfer and receive the data.

**Reset:**Pinno-35isusedasresetpin.Itisbasicallyusedtorestarttheprocessor.

**WR':**Pin no-36isusedforwriteoperationifWR'=0,WR=1.

**<u>INTERNALARCHITECTUREOF8255PPI:</u>**

- Itisknownas8255programmableperipheralinterfacesbecausethroughcertaincontrol word bits we can control the ports and modes of operation.
- Thetotalinternalarchitectureof 8255PPIcanbedividedinto3differentunitsthey are:
  a) Ports
  b) Databus buffer
  c) Readwritecontrollogicsection

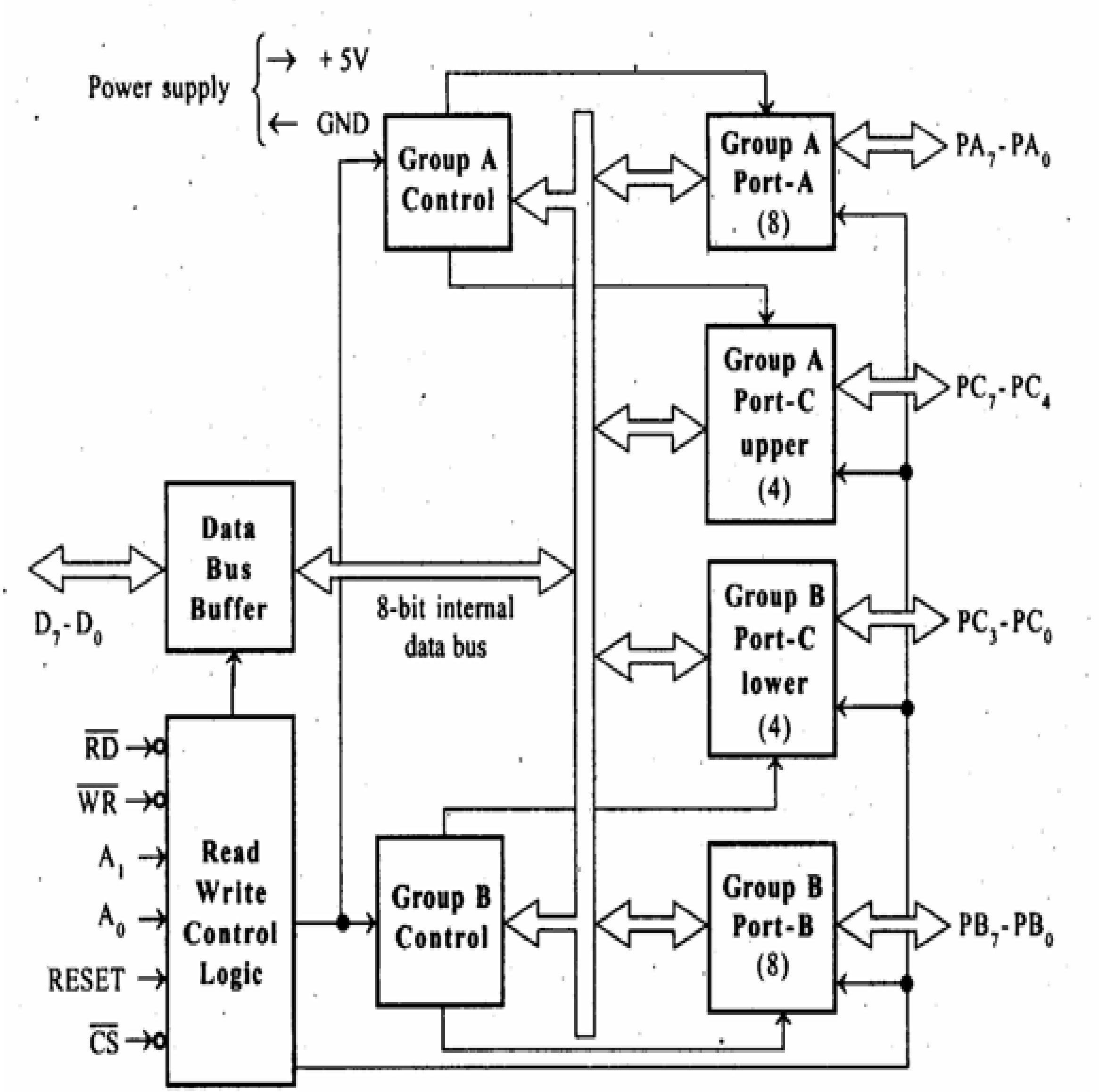
**a) Ports:** In 8255 PPI there are 3 parts i.e. Port A, Port B & Port C and the port C is alsodividedinto2typesi.e.Port Clowerand Port Cupper.Eachport can transfer or receive 8-bits of data and can operated in 3 different modes.

**i. Mode-0:** In this mode of operation there no combination of ports takes place and all the ports behaves as simple input output ports i.e. Port A can transfer individually 8-bit of data, Port B can transfer 8-bit of data and Port C upper & Port C lower can transfer individually 4 bits of data.

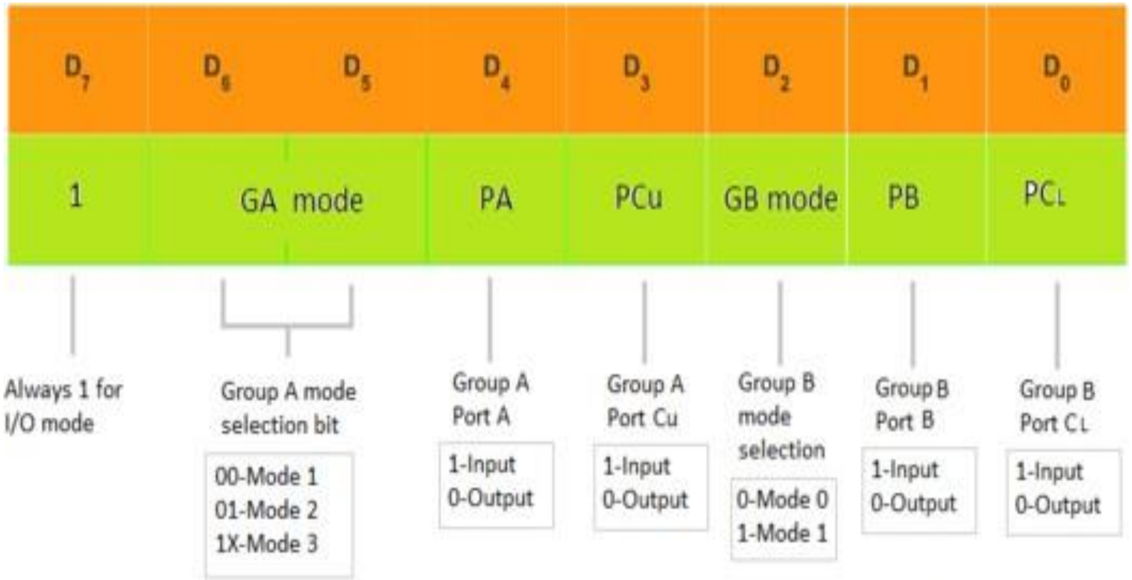
ii. **Mode-1:**Inthismodeofoperationthereiscombinationofcodestakesplace i.e.PortAcombineswithPortCupperandPortBcombinewithPortClower to transmit 12-bits of data. So, mode-1 is also known as “*Handshaking Mode”, where the combination of ports takes place.*

iii. **Mode-2:** In this mode only Port-A get activated and all other ports get deactivated so, Port A behaves as bidirectional port through which we can transfer or receive the data. This port is controlled by group control that meansgroupcontrolsPortAandPortCupperandthemodeofoperation. Similarly,groupcontrolthePortBandPort Clowerandmodeofoperation.

b) **Databusbuffer:**Itisknownasbidirectionalthatmeansitcantransferandreceive data and it consists of 8-bit of data.



c) **Read write control logic section:** It is basically use for controlling the ports and modeofoperationwiththehelpofcontrolwordbitwhichispresentinsidethe read write logic section and these control word bits are connected to different portsthroughwhichitcontrolstheportsthatmeansitcanmakeanyportasinput or output through the control word data which is of 8-bit.



PCu-Port C upper
PCL-Port C lower

8255 Control Word For I/O mode

Q. Make a control word bit for mode-0 operation where port A and port B behaves asinput port & port C upper and port C lower output port.

Ans.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

= $92_H$

Q.Makea controlwordbitformode-1operation,portAinput,portBoutput,portC upper input, port C lower output.

Ans.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

$=BC_H$

Q.Makeacontrolwordbitformode-2operationportAbehavesasinputallotherports are output ports.

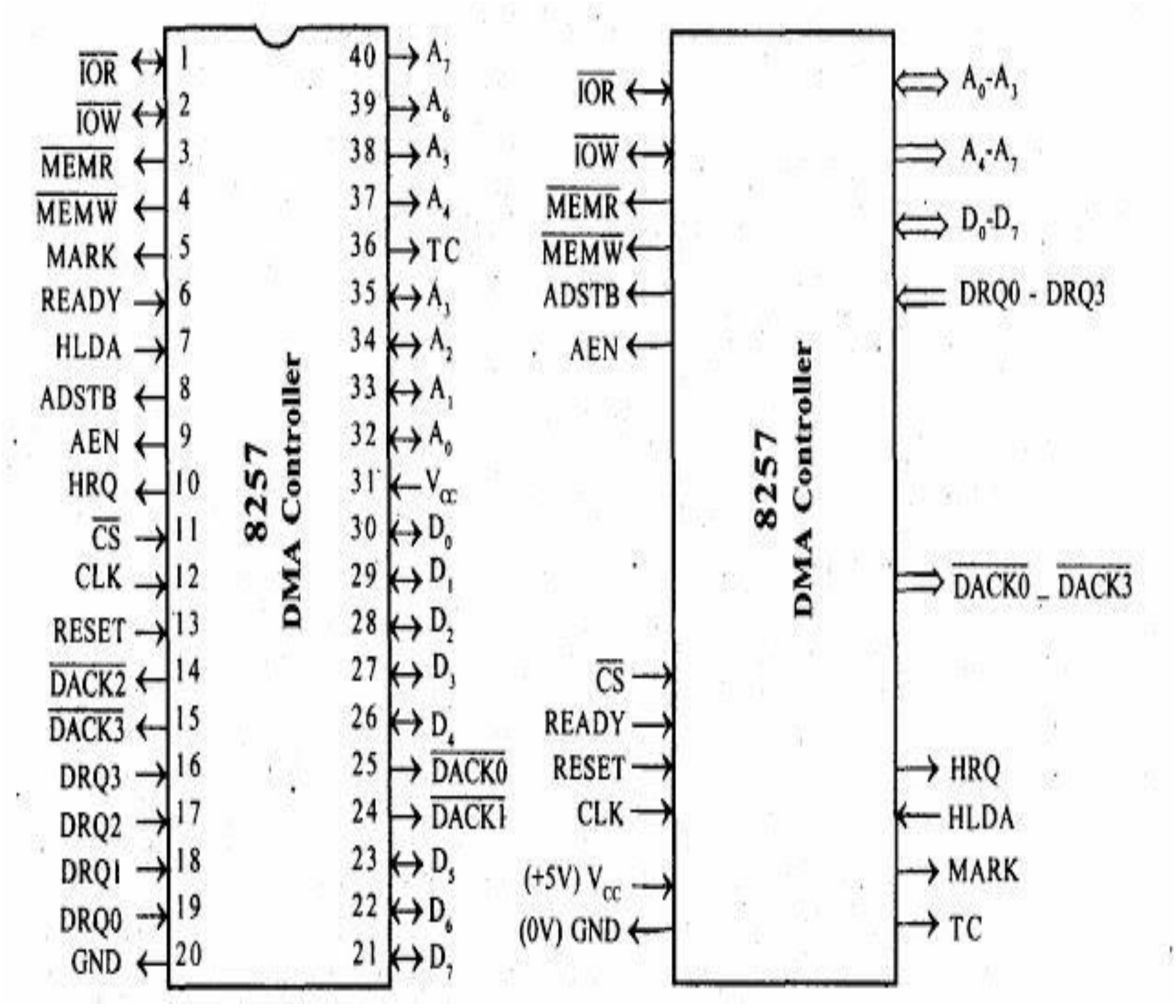
Ans.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

$= D0_H$

## 2. 8257DMA (DirectMemory Access):

### Pindiagramof8257DMA



### Pindescriptionof8257DMA

Itconsistsof40 pinICchipandoperates at+5V DC.

**IORandIOW’**

- Pinno-1 andpinno-2isusedasIORandIOW’operationrespectively.

- IORisusedforIOreadfromIOdeviceandIOW'isusedforIOwritefor transferring the data from the processor to the IO device.

**MEMR'andMEMW'**

- Pinno-3andpinno-4isusedasmemoryreadandmemorywriteoperation respectively.
- IfMEMR'=0ANDMEMW'=1thenmemoryreadoperationoccurs.
- IfMEMR'=1andMEMW'=0thenmemorywriteoperationoccurs.

**MARK**

- Pinno-5isusedformemoryacknowledgementpin.
- Through this pin the processor sends an acknowledgement message if a read write operation is performed.

**READY**

- Pinno-6isusedas readypin.
- Itkeepstheprocessorinactivestate.
- NormallytheprocessorspeedisfastandIOdevicespeedisslow.So,some time we are transferring the data to the processor but the data has not reachto the processor. So, inthat case the ready pintends tologic 0 and the processor goes to wait state.

**HLDA**

- Pinno-7is usedasHLDApin.
- TheprocessorwillsendanacknowledgemessagethroughtheHLDApin.

**ADSTB**

- Pinno-8isusedasaddressstrobepin.
- ItsfunctionissameasthatofALE pinof8085microprocessor.
- Itisbasicallyusedtoseparatetheaddressbusanddatabus.

**AEN**

- Pinno-9isusedasAddressEnablepin.
- Itisbasicallyusedtocheckthevalidityoftheaddress.

**HRQ**

- Pinno-10is usedasHoldRequestpin.

- Through this pin the request message for HOLD operation is send to the processor.

**CS’**

- Pin no-11 is used as cheap selection pin. If CS’=0, CS=1, then the control signal pins are get activated.

**CLK**

- Pinno-12is usedas CLK pin.
- Thispintellsabouttheclock pulse.
- Through this pin we can connect to other digital IC pins and basically use to provide square wave pulse or clock pulse or clock frequencies.

**RESET**

- Pinno.-13isusedasRESET pin.
- Byusingthispin,theprogramcontrolreturnstoFFFF$0_H$.
- Basically,itisusedtorestarttheprocessoriftheprogramhangsinbetween.

**DACK’**

- Thereare4DACK’pinin8257DMAsuchasDACK’$_0$(pinno-25),DACK’$_1$(pin no-24), DACK’$_2$(pin no-14) and DACK’$_3$(pin no-15)
- TheuseofDACK’pinistosendanacknowledgemessagewhenthedatais successfully accepted by the processor.

**DRQ**

- Thereare4different DMARequestpinsuchasDRQ$_3$(pinno-16),DRQ$_2$(pin no-17), DRQ$_1$(pin no-18), DRQ$_0$(pin no-19).
- Throughthese pin4differentIOdevicesareconnecttotheIC chip.

**GND**

- Pinno-20isusedastheGNDpin.

**$D_0$-$D_7$**

- Pinno-30,29,28,27,26,23,22,21isusedas$D_0$,$D_1$,$D_2$,$D_3$,$D_4$,$D_5$,$D_6$,$D_7$ respectively.
- Thesepinsaredatalinepinswhichisof8bitandbidirectionalinnaturei.e. it can transmit and receive the data.

### VCC

- Pinno-31isusedasVCCpin.
- ThroughVCCpin+5VDCsupplyisprovidedtotheIC.

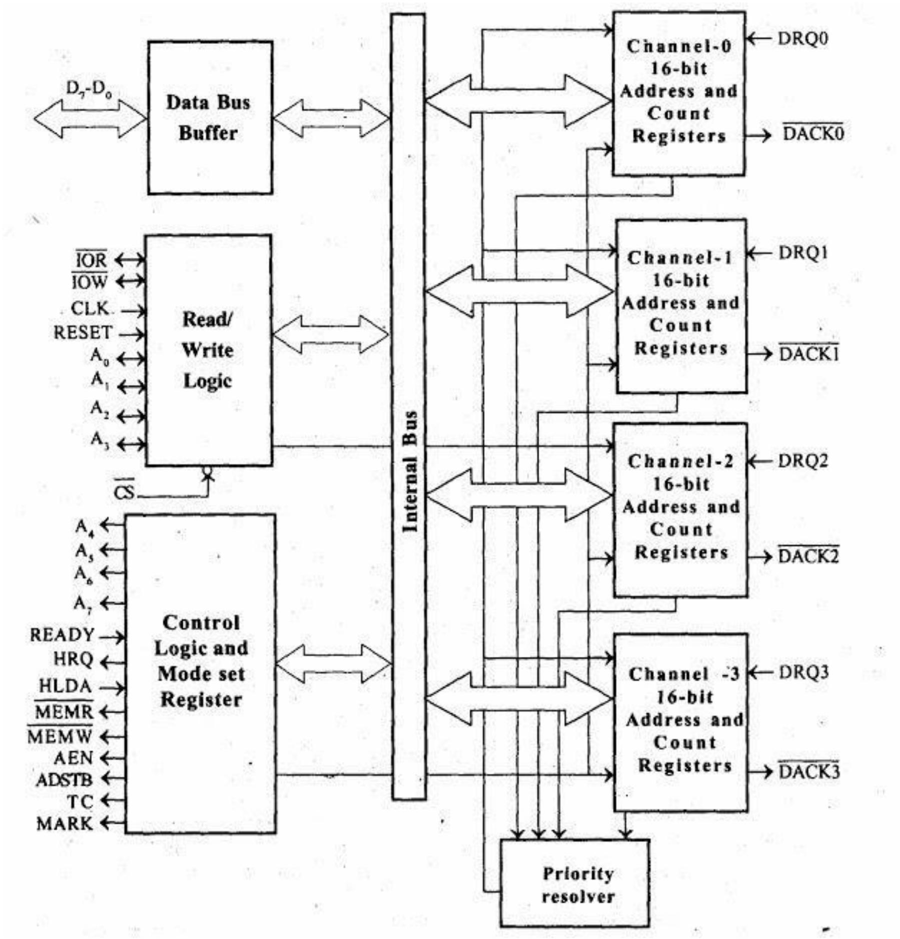
### $A_0$-$A_7$

- Pinno-32,33,34,35,37,38,39,40isused as$A_0$,$A_1$,$A_2$,$A_3$,$A_4$,$A_5$,$A_6$,$A_7$.
- Thesepins areaddresslinepinwhichisunidirectionalinnature.

### TC

- Pinno-36isusedastrans connectedpin.
- This pin is generally keptblank so thatthrough this pin we can connectother digital IC.

## INTERNALARCHITECTUREOF8257DMA:



ItisknownasDMA(DirectMemoryAccess)becauseinsteadoftransmittingthedatadirectlyto the processor, we can indirectly store the data from different IO devices and then we can transfer the data to processor to avoid over burden of the processor.

Thetotalinternalarchitectureof8257/8237DMAisbasicallydividedinto5differentunits

**a) 4 Channel**

- Thereare4channelsIDMAthroughwhichwecanconnectedto4differentIOdevices.
- Eachchannelhasgot2pinsi.e.DQRpinandDACK’ pin.

**b) Priority resolver**

- Themainfunctionofpriorityresolveristocheckwhichdatahasgone1st from4different channels through FIFO sequence and then according to priority basis, the data is transferred to the processor.

**c) Databusbuffer**

- Itisbidirectional,throughwhichwecantransferandreceive8bitsofdata.

**d) Read/writelogicsection**

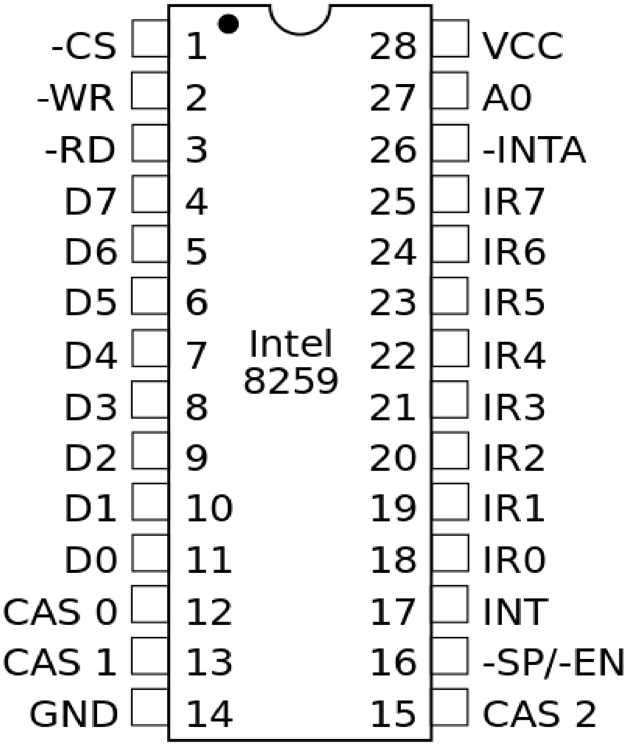
- It has the different control signal such as IOW’, IOR’ etc. through which it can perform different read write operation.

**e) Modesetandstatuswordregister**

- ThisunithasgotdifferentcontrolsignalsuchasADSTBpin,HLDA,RD’,WR’etc.through whichitcancontrolthedeviceandcanperformdifferentmemoryreadandmemorywrite operations.

## 3. 8259PIC(PriorityInterruptController)

**Pin diagram of 8259 PIC**



**Pindescriptionof8259PIC** It

consists of 28 pin IC chip. **CS'**

**PIN**

- Pinno-1isusedas CS'pin.
- Thispinisusedforchipselectionpurpose.
- IfCS'=0,CS=1,thencontrolsignalsareactivated.

**WR' PIN**

- Pinno-2isusedasmemorywrite pin.
- IfWR'=0,WR=1,thenchipperformthewriteoperation.

**RD'PIN**

- Pinno-3is usedas memoryreadpin.
- IfRD'=0,RD=1,thenchipperformthe memorywriteoperation.

**$D_7$-$D_0$PIN**

- Pinno-4to pinno-11are usedfordata linepini.e.$D_7$-$D_0$.

- Thedatalinepins areof8 bitandisbidirectionalinnaturei.e.wecantransfer orreceive the data.

**CASPIN**

- Pinno-12,13&15isusedasCAS$_0$,CAS$_1$&CAS$_2$respectively.
- Thesepinsarecascadebuffer pin.
- Basically,itisusedtoincreasetheinterruptlabel.
- Normallyasingle8259PICcanconnectto8differentIOdevicesbutthroughthese cascade buffer pin we can increase the interrupt label to 64.

**GND**

- Pinno-14isusedastheGNDpin.

**SP/EN'**

- Pinno-16is usedas SP/EN'pin.
- Whenasingle8259PICisoperatingtheSP/EN'pintendtologic0i.e.SP=0,EN'=0,EN=1 that means the processor is operating in enable mode
- When8259PICisconnectedtootherPICthenitoperatesinslavemodethatmeansSP=1, EN'=1, EN=0.

**INTPIN**

- Pinno-17is usedas interruptpin.
- Throughthispinthedataissendtotheprocessorandifthedataissuccessfullyaccepted by the processor then the processor sends an acknowledge message through the INTA' pin (Pin no-26) or interrupt acknowledgement pin.

**IR$_0$-IR$_7$**

- Pinno-18to pinno-25isusedasinterruptrequestline pin.
- Throughthis pintheprocessor canreceive8 no. ofinterrupt request from 8 different IO device at the same time.

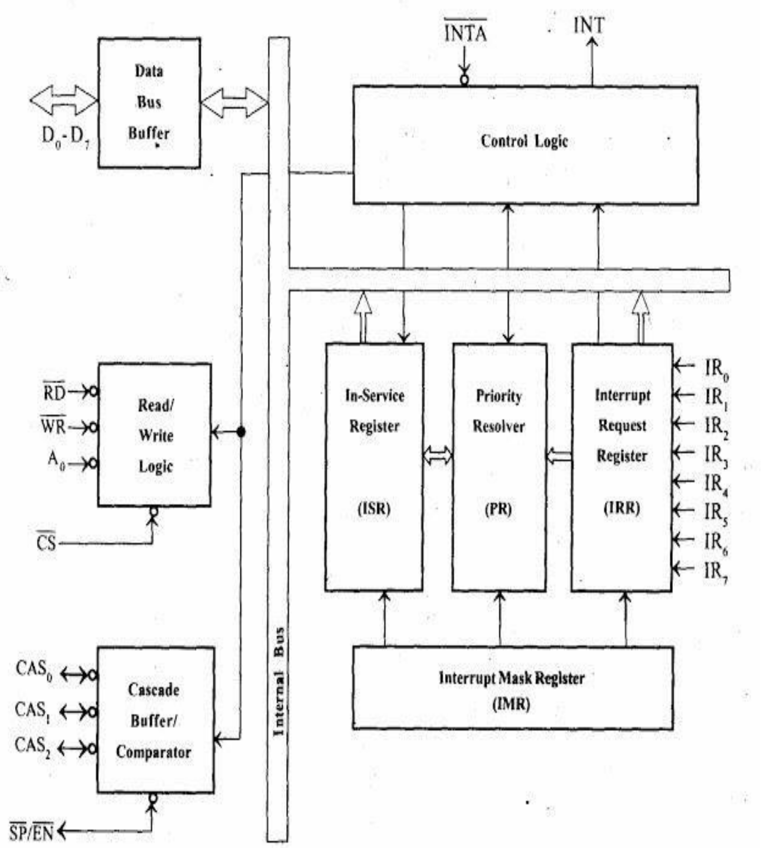
**A$_0$PIN**

- Pinno-27is usedasA$_0$.
- Itisacontrolsignalpinformemoryreadormemorywriteoperation.
- IfA$_0$=0thenprocessorperformmemorywriteoperationandifA$_0$=1thenprocessor perform memory read operation.

### VCC

- Pinno-28isusedasVCCpin.
- ThroughVCCpin+5VDCsupplyisprovidedtotheIC.

### INTERNALARCHITECTUREOF8259PIC



It is basically used for interrupt driven operation where a single 8259 PIC can receive 8 no. of interrupt request from 8 different IO devices but in cascade form the interrupt label can be increase to 64.

Thetotalinternalarchitectureof8259PIC canbedividedinto4differentunits.They are

1. Interruptandcontrollogicsection
2. Databusbuffersection
3. Readwritelogicsection
4. Cascadebuffersection

1. **Interruptandcontrollogicsection**
   Underthissectionwehave5differentunits,they are,

a. **InterruptRequestRegister(IRR)**

Throughthisregisterwecanreceive8no.ofinterruptrequestfrom8differentIOdevice through the interrupt request line i.e. $IR_0$-$IR_7$ and the data is stored in this interrupt request register.

b. **Priority resolver**

Themainfunction ofpriorityresolveris tocheck whichdatahas got thehighest priority by 3 different modes. They are

i. **FIFOsequence**

Inthissequencethedatafrom8differentIO deviceswhichisstoredinIRRissendto the priority resolver and the priority resolver checks which data has come first depending uponFirst InFirst Out(FIFO) sequenceand that dataissend toin service register and other data are being blocked or marked by interrupt mask register.

ii. **RotatingPrioritymode**

In this mode a particular sequence is provided to the interrupt request line so that the data from the IO device is transmitted in that sequence to the interrupt request register and then this data are transmitted to the priority resolver.

iii. **FixedPrioritymode**

In this mode a particular interrupt request line is provided with highest priority so that the data from these interrupt request lines is transmitted first to the in-service register and other data are provided with lower priority.

iv. **Interruptmaskregister**

Itsmainfunctionistoblockormaskthedatawhichishavinglowerpriority.

v. **In-serviceregister**

The data which is having highest priority is send to in-service register and then this data is send to the processor through the control logic unit.

c. **Databusbuffer**

Itisbidirectionalanditisof8bits.

d. **Readwritelogicsection**

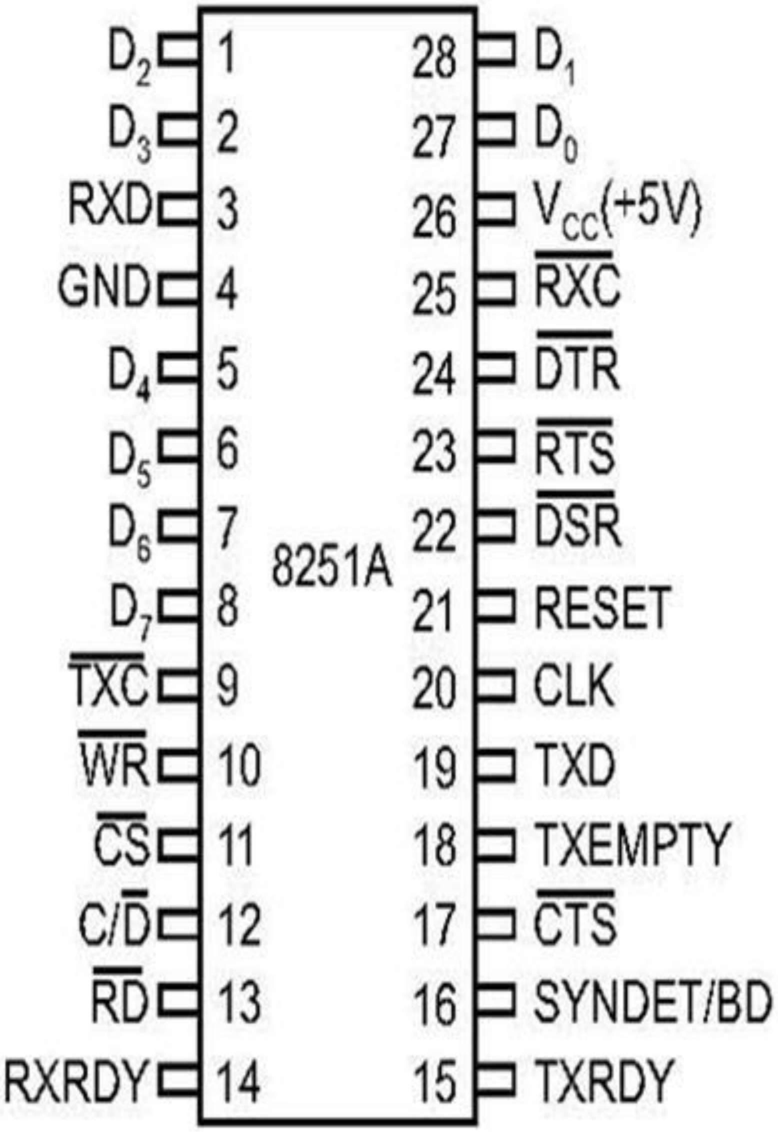
Ithasgotdifferentcontrolsignalpinsuchas RD′,WR′,$A_0$,CS′pinwhichisbasicallyused for read write and cheap selection operation.

e. **Cascadebufferandcomparator**

Throughthisunitwecanincreasetheinterruptlevelupto64.ThedifferentpinsareCAS0, CAS1, CAS2 or cascade buffer pin which we can be connected to other 8259 PIC and SP/EN′ pin is used to check whether the processor is operation in slave mode or enable mode.

**8251USART(UniversalSynchronousAsynchronousReceiverTransmitter) Pin**

**diagram of 8251 USART**



**Pindescriptionof8251 USART**

Itconsistsof28 pinICchipandoperates at+5V DC.

**$D_0$-$D_7$**

- Pinno-27,28,1,2,5,6,7,8 isusedas$D_0$,$D_1$,$D_2$,$D_3$,$D_4$,$D_5$,$D_6$,$D_7$respectively.
- Thesepinsaredatalinepinandbidirectionalinnature.

**RXDpin**

- Pinno-3isusedasRXDpin.
- Itisaninputpinorreceiverdata pin.
- Through this pin the data is received by the processor from the external IO device and then stored in receiver buffer section.

**GNDpin**

- Pinno-4isusedastheGNDpin.

**TXC′pin**

- Pinno-9is usedas transmitterclockpin.
- It is used to control the rate at which the characters are transmitted and it is connected to transmitter control unit and it always keeps it in active mode i.e. TXC′=0, TXC=1.

**WR′ Pin**

- Pinno-10isusedasmemorywritepin.
- IfWR′=0,WR=1,thenchipperformthewriteoperation.

**CS′PIN**

- Pinno-11isusedasCS′pin.
- Thispinisusedforchipselectionpurpose.
- IfCS′=0,CS=1,thencontrolsignalsareactivated.

**C/D′pin**

- Pinno-12is usedas control/datapin.
- Thispinisusedtodifferentiatebetweenthecontrolsignaland data.
- IfC/D′=0thenthedatalinesareactivatedandifC/D′=1thenthecontrolsignalsare activated.

**RD′**

- Pinno-13is usedasmemoryreadpin.
- IfRD′=0,RD=1,thenchipperformthe memorywriteoperation.

**RXRDY**

- Pinno-14isusedasreceiverready pin.
- thesepinsareusedtoinformtheprocessorthatthedataisreadyfortransfertothe processor.
- Thispinisconnectedtothereceivercontrolunit.

- **TXRDY**
- Pinno-15is usedastransmitterreadypin.
- Itisanoutputpinanditsfunctionaretoinformtheprocessorthatthetransmitterbuffer is ready to transmit the data to the external IO device.

**SYNDET/BRKDET**

- Pinno-16isusedassynchronousdetect/baudratepin.
- Thispinisaninputoutputpinthroughwhichwecandetectthesynchronousdatatransfer and also, we can measure the baud rate (*The no. of data bits transmitted per unit time*).

**CTS'**

- Pinno-17is usedas controlanddatatransfer pin.
- Itisusedinmodemcontrolfor checkingserialorparalleldatatransmission.

**TXE**

- Pinno-18is usedastransmitteremptypin.
- Itisanoutputpinanditisusedtoinformtheprocessorthatthetransmittercontrol section is empty and it can receive new data.

**TXD**

- Pinno-19is usedas transmitterdatapin.
- It is an output pin which is connected to the transmitter buffer pin and is basically used to transfer the data to the external IO device.

**CLK**

- Pinno-20isusedas clockpin.
- Thispintellsabouttheclock pulse.
- ThroughthispinwecanconnecttootherdigitalICpinsandbasicallyusetoprovide square wave pulse or clock pulse or clock frequencies.

**RESET**

- Pinno-21isusedas RESETpin.
- Byusingthispin,theprogramcontrolreturnstoFFFF$0_H$.
- Basically,itisusedtorestarttheprocessor iftheprogramhangsin between.

**DSR'**

- Pinno-22isusedasdatasetready pin.

- It is aninput pinand it is basically used tocheck whether the datais ready for receiving at the modem unit.

**DTS’**

- Pinno-23isusedasdatarequesttosendpin.
- Itisanoutputpinandisconnectedtothemodemanditindicatesthatthetransmitteris ready to transmit the data through themodem.

**DTR’**

- Pinno-24is usedasdatatransferandreceiverpin.

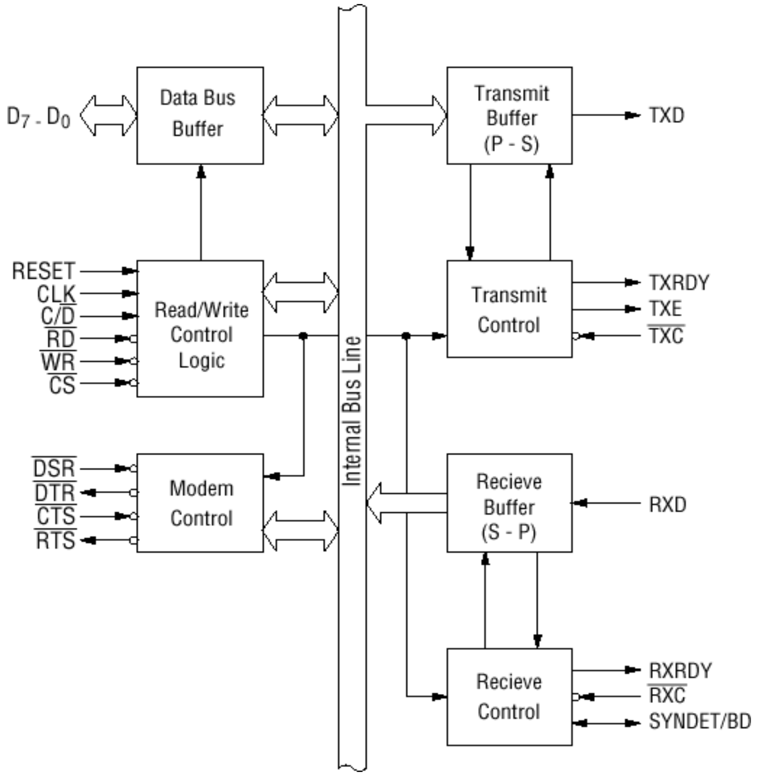
**RXD’**

- Pinno-25isusedasreceiverdatapin.

**VCC**

- Pinno-26isusedaspowersupplypin.
- ThroughVCCpin+5VDCsupplyisprovidedtotheIC

**INTERNALARCHITECTUREOF8251USART**



Itsupportsbothsynchronousand asynchronousmodesof operation.

The8251USARTreceivesparallel datafromtheprocessorandtransferitseriallyand alsoit receives serial data from the IO device and transmit them parallelly to the processor.

Thetotalinternalarchitectureof8251USARTcanbedividedinto4differentunitssuchas

1. **Transmitterandreceiverbuffercontrolunit**
   - Thesearebasicallyusedforreceptionandtransmissionofdata.
   - Thereceiverunithasgotreceivercontrolandreceiverbufferunit.
   - The receiver control has got different control signals such as RXD for receiving data, RXRDY it indicates that the receiver is ready toreceived data from the IO device and to transmit it to the processor through the receiver buffer unit.
   - TheSYNDTindicatesthatthereceiverisreadytoreceiveagroupofcharactersordata at a time.
   - Similarly, the transmitter section has got 2 units; they are control buffer and transmitter buffer.
   - The transmitter control has got different control signals through which it checks whether the transmitter is empty and to transmit data to the transmitter buffer. (Control signals are TXRDY', TXE).
2. **Databusbuffer**
   - Itisbidirectionalinnatureandthroughthedatabusbufferwecantransferandreceive 8-bit of data.
3. **Readwritelogicunit**
   - It is basicallyused for read writeoperationand for control-oriented activities suchas RD' for read operation, WR' for write operation.
   - IftheC/D'=0thenthedatalineis selected.
   - IftheC/D'=1thenthecontrolsignalisselected
4. **Modem control**
   - Itisbasicallyusedtoconvertthe digitaltoanalogdataandviceversa.
   - Itisalsousedfordatatransmission.

## SERIALDATATRANSFERFORMAT

1. **Synchronousserialdatatransfer**

- Inthismethodablockofdataistransmittedseriallyatthesametime.
- Theno. ofdatabytesisnot limited.
- Thedatabytestransmittedoneaftertheother.
- Herethedata speedorbitrateismorethan20$_{KB/sec}$.
- Inthiscasethedatawhichistransmittedareseriallybetweenthetransmitterandreceiver and some synchronous characters are added to the data bits and these synchronous characterscanbeconnectedtomultiplereceivers.So,insynchronousdatatransferserial toparallel datais transmitted and thereis nostart and stop bit and datatransmissionis unlimited normally 256 bits.

2. **Asynchronousserialdatatransfer**

- Inthismethod1bytesofdataistransferseriallyatatime.
- Aftereachdatatransferthereisa startandstopbit.
- Thedatatransferrate issloworlessthan20$_{KB/Sec}$.
- Herethedatatransferrateislimitedascomparetosynchronousdatatransferwherethe word length is unlimited.
- Here 8 bits of data is transferred at a time after which there is a stop bit but there is no start or stop bit in synchronous data transfer.

| SERIALDATATRANSFER | PARALLELDATATRANSFER |
|---|---|
| 1. 1bitofdataistransferredatatime.<br>2. Datatransferis slow.<br>3. This method is used for long distance communication. | 1. 8bit/16bitofdataistransferredat a time.<br>2. Datatransferis fast.<br>3. This method is used for short distance communication. |

**Q.Connectthe32KbyteRAMwith8086microprocessorsinmaximummodewiththe starting address from 60000$_H$.**

**Solution:**

Total capacity of RAM= 32KB

Evenmemorybanksize=16KB

Oddmemorybanksize=16KB

Now16Kbytes=16*1024=$2^4$*$2^{10}$=$2^{14}$i.e.here14pinsareusedfor addressoperation.

| A 19 | A 18 | A 17 | A 16 | A 15 | A 14 | A 13 | A 12 | A 11 | A 10 | A 9 | A 8 | A 7 | A 6 | A 5 | A 4 | A 3 | A 2 | A 1 | A 0 | BHE’ | ADDRESS | RAM /ROM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 60000 | 16K*8 EVEN |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 67FFE | |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 60001 | 16K*8 ODD |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 67FFF | |

[In8086microprocessoraddressline=20bit Accordingtoquestionwehavetoconnect32KbyteRAMwith8086Microprocessor so 32kbyte= 16kbyte + 16kbyte

EVEN ODD

16kbyte= = 16*1024= $2^4$*$2^{10}$= $2^{14}$i.e. here 14 pins are used for address operation and among from rest 6pin, 5 (A19-A15 pins) are used for control signal and 1 (A0 pin) is used for even or odd ram selection pin.

**Q. Connect the 64Kbyte RAM with 8086 microprocessors in maximum mode with the starting address from 00000$_H$.**

**Solution:**

Total capacity of RAM= 64KB

Evenmemorybanksize=32KB

Oddmemorybanksize=32KB

Now16Kbytes=32*1024=$2^5$*$2^{10}$=$2^{15}$i.e.here15pinsareusedfor addressoperation.

| A 19 | A 18 | A 17 | A 16 | A 15 | A 14 | A 13 | A 12 | A 11 | A 10 | A 9 | A 8 | A 7 | A 6 | A 5 | A 4 | A 3 | A 2 | A 1 | A 0 | BHE’ | ADDRESS | RAM /ROM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00000 | 32K*8 EVEN |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0FFFE | |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 00001 | |

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0FFFF | 32K*8 ODD |

[In8086microprocessoraddressline=20bit Accordingtoquestionwehavetoconnect32KbyteRAMwith8086Microprocessor so 64kbyte= $\underline{\text{32kbyte}}$ + $\underline{\text{32kbyte}}$

EVEN ODD

32kbyte= = 32*1024= $2^5$*$2^{10}$= $2^{15}$i.e. here 15 pins are used for address operation and among from rest 5 pin, 4 (A19-A16 pins) are used for control signal and 1 (A0 pin) is used for even or odd ram selection pin.

**Q. Usinga3:8decoderinterfacesa32KbyteRAMwith8086microprocessorsinminimum mode with the starting address from 00000$_H$.**

**Solution:**

Total capacity of RAM= 32KB

Evenmemorybanksize=16KB

Oddmemorybanksize=16KB

Now16Kbytes=16*1024=$2^4$*$2^{10}$=$2^{14}$i.e.here14pinsareusedfor addressoperation.

| A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | BHE' | ADDRESS | RAM /ROM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00000 | 16K*8 EVEN |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 07FFE | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 00001 | 16K*8 ODD |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 07FFF | |

[In8086microprocessoraddressline=20bit Accordingtoquestionwehavetoconnect32KbyteRAMwith8086Microprocessor so 32kbyte= $\underline{\text{16kbyte}}$ + $\underline{\text{16kbyte}}$

EVEN ODD

16kbyte= = 16*1024= $2^4$*$2^{10}$= $2^{14}$i.e. here 14 pins are used for address operation and among from rest 6 pin, 5 (A19-A15 pins) are used for control signal and 1 (A0 pin) is used for even or odd ram selection pin.

**<u>Multiplechoicequestionanswer:</u>**

1. Theno.Ofportsin8255PPIis
   A)2
   B)3
   C)4
   D) 5
   Answer:B
2. whichportin8255PPIisdividedintoupperandlowerport.
   A) portc
   B) portA
   C) portB
   D) noneofthese.
   Answer: A
3. InMode0operation.
   A) Alltheportsbehave asbidirectional port
   B) Alltheportsbehaveassimpleinputoutput port.
   C) PortAbehavesasbidirectionalport
   D) Noneofthese.
   Answer: B
4. Thefunctionofinterfacingdevice i.e.
   A) speedmatching
   B) impedancematching
   C) reducingspeed
   D) Disconnectingports
   Answer: A
5. Mode1operation
   A) HandshakingMode
   B) BidirectionalMode

C) UnidirectionalMode

D) WaitMode

Answer: A

6. InMode2operation

A) OnlyportAbehavesas bidirectional Mode

B) unidirectionalMode

C) wait Mode

D) InactiveMode

Answer: A

7. 8255PPIisknownasprogrammableperipheraldevicebecause

A) Itisaninterfacingdevice

B) connectingports

C) ThroughcontrolwordbitwecancontroltheportsandModeof Operation.

D) connectingports

Answer: C

8. Ifcontrolwordbitis80Hthenitoperatesin

A) Mode0

B) Mode1

C) Mode2

D) Mode3

Answer: A

9. In8257DMA thereare

A) 2channel

B) 1channel

C) 3channel

D) 4channel.

Answer: D

10. TheFIFOsequence

A) Thedatawhichhascomefirstwilltransferfirst.

B) Thedatawhichhascomefirstwilltransferlast.

C) Fixeddatatransfer.

D) Rotatingdatatransfer

Answer: A

11. Thefunctionofpriorityresolveris

A) Thedatahavinghighestpriority.

B) lowestpriority.

C) interfacing

D) connectingexternaldevice.

Answer: A

12. Asingle8259PIChas

A) 8interruptrequestline

B) 7interruptrequest line

C) 3interruptrequest line

D) 4interruptrequestline

Answer: A

13. Incascadeformtheinterruptlevelcanbeincreasedto

A) 52

B) 60

C) 64

D) Noneofthese

Answer: C

14. Thefunctionofinterruptmask register

A) Blockdatahavinglowerpriority.

B) highestpriority

C) unblockdata

D) Transferdata

Answer: A

15. Thefunctionofin-serviceregister

A) storedatahavinghighestpriority

B) lowestpriority

C) Nopriority

D) unblockdata

Answer: A

16. Themainfunctionofcontrollogicunitof8259PICis

A) Transferdatatotheprocessor

B) blockdatatotheprocessor

C) Unblockdata

D) Noneofthese

Answer: A

17. Thefunctionof8251 USART

A) synchronousdatatransfer.

B) synchronousandAsynchronousdatatransfer

C) Asynchronousdatatransfer.

D) Noneofthese

Answer: B

18. AfunctionofMODEM
    A) Modulation
    B) Demodulation
    C) Encoding
    D) ModulationandDemodulation.
    Answer: D
19. ThefunctionofBusHighEnablepin
    A) checkthevalidityofdata
    B) Checkthevalidityof address.
    C) connectingtoports
    D) Noneofthese.
    Answer: B
20. Thenumberof cascadebufferpinin8259PICis
    A)2
    B)3
    C)4
    D) 5
    Answer:B
21. In8255PPIhas
    A) 28 pin
    B) 30
    C) 32
    D) 40
    Answer:D
22. HowmanyModesaretherein8255PPI?
    A)1
    B)2
    C)4
    D )3
    Answer:D
23. TheunitwhichispresentinmaximumModebutnotinminimum Mode
    A) 8288Buscontroller
    B) latches
    C) Transreceiver
    D) Noneofthese
    Answer: A
24. Howmanypinsaretherein8257DMA
    A) 124

B) 28

C) 30

D) 40

Answer:D

25. Ifthecontrolwordbitis98Hthen8255PPIwilloperatein

A) Mode2

B) Mode3

C) Mode1

D) Mode0

Answer: D

26. Ifthecontrolwordbitis(BC)Hthe8255PPIwilloperatein

A) Mode0

B) Mode1

C) Mode2

D) noneofthese

Answer: B

27. Ifthecontrolwordbitis(DO)Hthen8255PPIwilloperatein

A) Mode2

B) Mode1

C) Mode3

D) noneofthese

Answer: A

28. Microprocessorisusedfor

A) programoriented

B) interfacing

C) controloriented

D) noneofthese

Answer: A

29. Thefunctionof latches

A) storingresult

B) connectingports

C) Data

D) Dataandaddress

Answer: D

30. Insynchronousdatatransferthedatatransferrateismorethan

A) 10kbps

B) 15kbps

C) 20Kbps

D) 5kbps

Answer: C

## ASSIGNMENTFULL MARKS-100

### SECTION-A

### (ANSWERALLQUESTIONS)

**Shortanswer type question: 2*8=16**

a) Whatismeantbyinterfacingdevice?
b) Howmanyports aretherein8255 PPI?
c) why8255isknownasprogrammableperipheralinterface.
d) Make a controlword bitfor modeoperation where port AandB behaves asinput port.
e) whatismeantby8257 DMA?
f) Howmanychannelsaretherein8257 DMA?
g) whatisthefunctionofpriorityresolver?
h) whatisthefunctionof8259 PIC?

### SECTION-B

**Q.2Focusedanswertype question: 6*6=36**

a) Explainthefunctionof8255 PPI.
b) Explainthefunctionof8257 DMA.
c) Explainthepindiagramof8555PPI.
d) Explainthepindiagramof8257DMA
e) Explainthefunctionof8259 PIC.
f) Explainthepindiagramof8259PIC.
g) Explainthevariousmodes of operationof8255 PPI.

h) Makeacontrolwordbitformodeoandmode1operationwhereportAbehavesasinput port, port B as output port, port $C_{upper}$ as input and port $C_{lower}$ output.

**SECTION-C**

**Q.3Longanswertypequestion: 16*3=48**

a) Explaintheinternalarchitectureandpindiagramof8255PPI.
b) Explaintheinternalarchitectureandpindiagramof8257DMA.
c) Explaintheinternalarchitectureandpindiagramof8259PIC.

d)

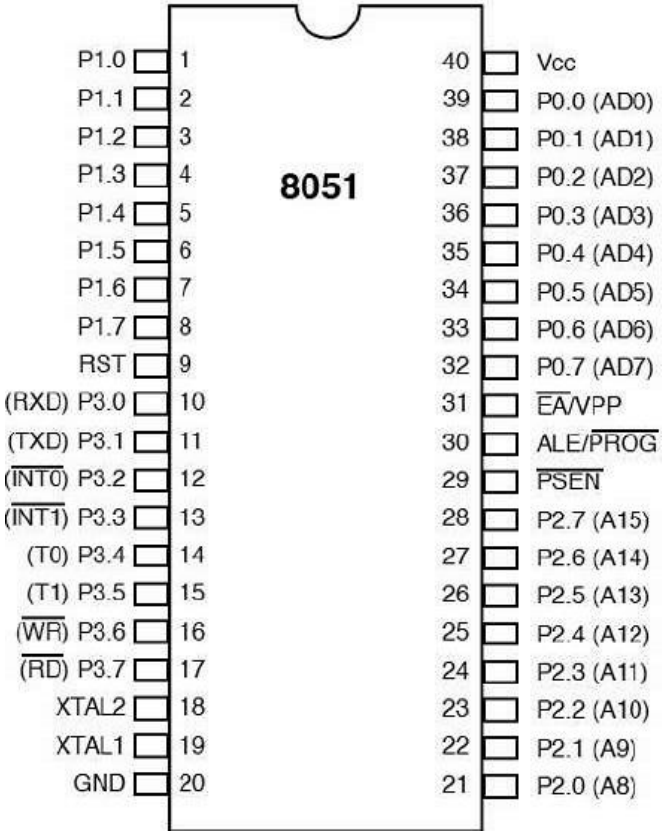
# MODULE-4:8051MICROCONTROLLER

Microcontroller is like a mini computer with a CPU along with RAM, ROM, serial ports, timers, andIO peripherals allembeddedon asingle chip.It'sdesignedto perform applicationspecific tasks that require a certain degree of control such as a TV remote, LED display panel, smart watches, vehicles, traffic light control, temperature control, etc. It's a high-end device with a microprocessor, memory, and input/output ports all on a single chip. It's the brains of a computersystemwhichcontainsenoughcircuitrytoperformspecificfunctionswithoutexternal memory.Sinceitlacksexternalcomponents,thepowerconsumptionislesswhichmakesitideal for devices running on batteries. Simple speaking, a microcontroller is complete computer system with less external hardware.

Itbasicallyconsistsof40pinICchipandoperatesat12MHzclockfrequencyandsupplyvoltage is +5V DC.

| **MICROPROCESSOR** | **MICROCONTROLLER** |
|---|---|
| 1. A microprocessor is a general-purpose device which is called a CPU.<br>2. It is basically used for program-oriented activity.<br>3. WehavetoattachexternalRAM, ROM, oscillator with microprocessor.<br>4. Microprocessors are most commonly used as the CPU in microcomputer systems.<br>5. Microprocessor instructions are mainly nibble or byte addressable. | 1. A microcontroller is a dedicated chip which is also called single chip computer.<br>2. It is basically used for control-oriented activity or controlling the device.<br>3. Microcontroller consists of microprocessor and all other units such as RAM, ROM, input crystal oscillator etc. for complete input output operation.<br>4. Microcontrollers are used in small, minimum component designs performing control-oriented applications.<br>5. Microcontroller instructions are both bit and byte addressable. |

**PINDIAGRAMOF8051MICROCONTROLLER**



**PINDESCRIPTIONOF8051MICROCONTROLLER**

**Pins1to8:**ThesepinsareknownasPort1.Thisportdoesn'tserveanyotherfunctions.Itis internallypulledup,bi-directionalI/Oport.

**Pin9:**ItisaRESETpin,whichisusedto reset themicrocontrollerto itsinitial values.

**Pins 10 to 17:** These pins are known as Port 3. This port serves some functions like interrupts, timer input, control signals, serial communication signals RXD and TXD, etc.

**Pins18&19:**These pinsare usedforinterfacinganexternalcrystaltogetthesystemclock.

**Pin20:**Thispinprovidesthepowersupplyto thecircuit.

**Pins 21 to 28**: These pins are known as Port 2. It serves as I/O port. Higher order address bus signals are also multiplexed using this port.

**Pin29:**ThisisPSENpinwhichstandsforProgramStoreEnable.Itisusedtoreadasignalfrom the external program memory.

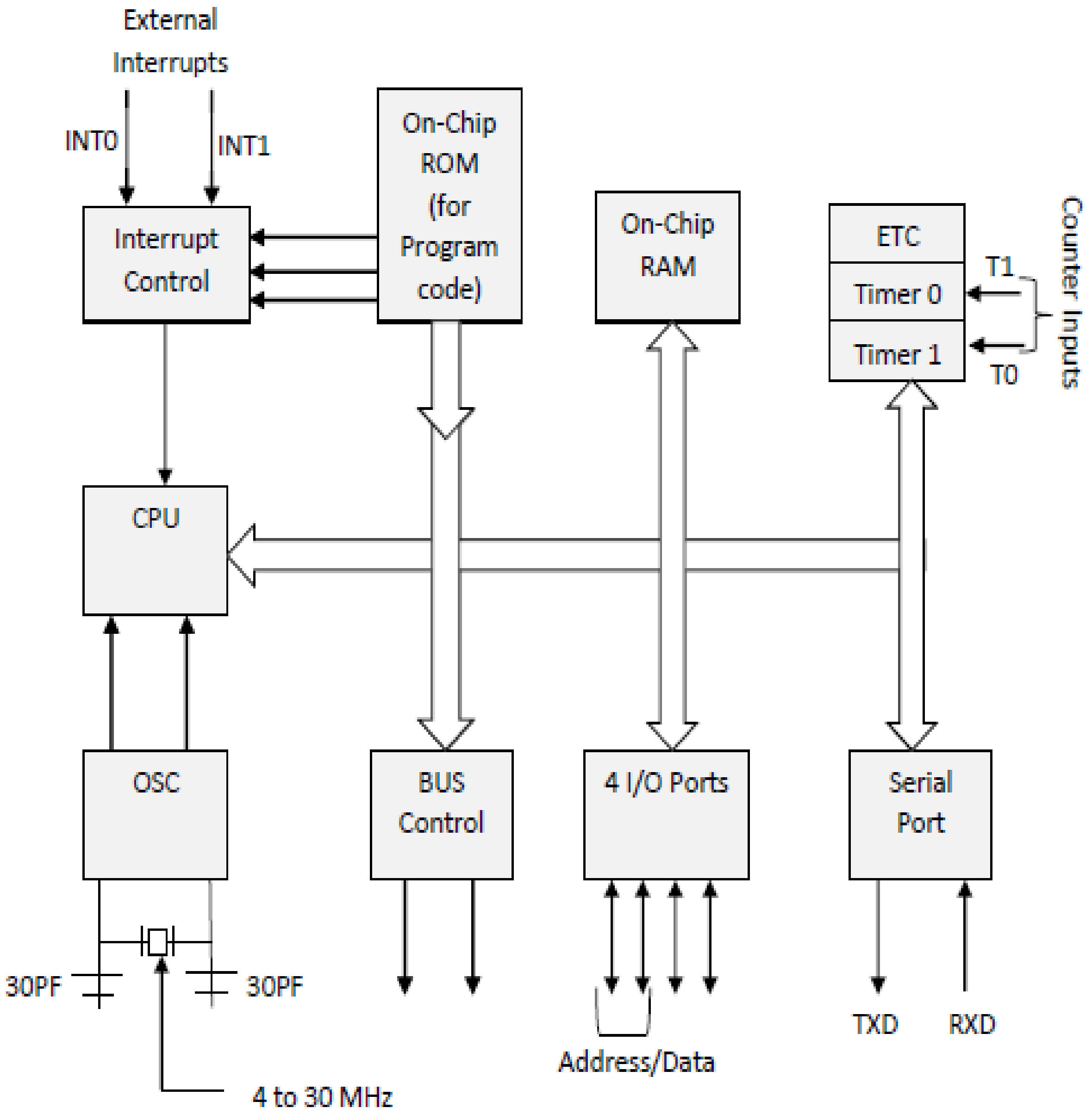
**Pin 30**:This is EApin which stands for ExternalAccess input. Itis usedtoenable/disable the external memory interfacing.

**Pin31:**ThisisALEpinwhichstandsforAddressLatchEnable.Itisusedtodemultiplexthe address-data signal of port.

**Pins32to39:** ThesepinsareknownasPort0.ItservesasI/Oport.Lowerorderaddressanddatabus signals are multiplexed using this port.

**Pin40:**Thispinisusedto providepowersupplyto the circuit.

## INTERNALARCHITECTUREOF8051MICROCONTROLLER



Thetotalinternalarchitectureof8051microcontrollerbasicallyconsistsof followingunitssuch as

1. **RAM**
   - It consists of 128 bytes of RAM out of which 32 bytes are used for register bank selection.
   - Thereare4registerbanksi.e.Bank-0,Bank-1,Bank-2,Bank-3andtheseregisterbanks canbe selected withthe help of status control word i.e. $RS_0$ and $RS_1$ whichis present inside the status flags 8051 microcontroller.
   - So, with the help of logic bits we can select a particular register bank and the mode of operation.
2. **ROM**
   - it is basically used to store the predefine data or library function and its memory capacity is 4kB.
3. **CLKfrequency/ Oscillator**
   - Inbuilt crystal oscillator is present inside the microcontroller which provide internal CLK frequency up-to 12MHz.
4. **Ports**
   - Thereare4portssuchasPort-0,Port-1,Port-2,Port-3andeachportcantransferand receives 8-bit of data.
   - Itisaconnectingpointorinterfacebetweentheprocessorandtheexternal device.
5. **Timerandcounter**
   - There are 2 timers i.e. timer-0 and timer-1 which consists of 16 bits registers and is basically used for time delay operation and providing a matching clock frequency.
6. **Processor**
   - Itisbasicallyusedforarithmeticadlogicaloperationandforstoringtheresult,tothe processor status flag is attached which is basically used to check the status of the output program or for checking the result.
   - In8051microcontrollerthestatusflagconsistsof8-bitsoutofwhich6aredefineand 2 are undefined pins.
   - ThedefinedpinsareCarry flag,Auxiliarycarry,$RS_0$,$RS_1$,Overflowflag,Parityflag.
   - $RS_0$& $RS_1$ are basically used for register bank selection or for selecting a particular bank for storing the data. If no bank is selected the by default Bank-0 is selected.

| **$RS_1$** | **$RS_0$** | **REGISTERBANK** |
|---|---|---|
| 0 | 0 | BANK-0 |
| 0 | 1 | BANK-1 |
| 1 | 0 | BANK-2 |
| 1 | 1 | BANK-3 |

   - In case of any arithmetic operation if the result is more than the destination register valuethenthedatacannot bestored inaccumulator sointhis casetheoverflow flag will tends to logic-1.

7. **Interruptcontrol**
    - Basically, it is used for interrupt operation. It has got different interrupt pins such as $INT_0$, $INT_1$ etc. through which we perform interrupt operation.

## INSTRUCTIONSETSOF8051MICROCONTROLLERS

Instructions are sets of commands given to the processor to perform a specific operation accordingly in 8051 microcontrollers the total instruction set can be divided into 5 different types according to type of operation it performs. The instructions are;

1. Datatransforminstruction
2. Arithmeticinstruction
3. Logicalinstruction
4. Booleanvariableandmanipulationinstruction
5. Programmingbranchinginstruction

1. **Datatransforminstruction:**Itisbasicallyusedfortransferringthedatafromone register to another register or register to memory without changing the content.

    E.g.MOVA,$R_0$-Move thecontentofregister$R_0$to accumulator.

    MOVA,#20-Moveimmediatelythedata$20_H$totheaccumulator. (Here# indicates a data)

2. **Arithmetic instruction:** These are basically used for arithmetic operation such as addition, subtraction, multiplication, division, increment, decrement etc. In this case the final result may change.

    E.g. SUB A, $R_0$- Subtract the content of $R_0$ from the accumulator and the result is stored in accumulator.

    ADD A, #20- Add immediately the data 20 with accumulator and the result will stored in accumulator.

3. **Logical instruction:** These instructions are basically used for logical operation such as ANA (AND), ORA (OR), XRA (XOR), CMP (COMPARE) etc.

4. **Booleanvariablemanipulationinstruction:** Theinstructiononthisgroup arebasically for et or of the status bit and for status flag manipulation.

    E.g. CLC- Clear the Carry bit.
    STC-Setthecarrybittologic1.
    CMC-Complementofcarrybit.
    JC- Jump with carry.
    JNC-Jumpwithnocarry.
    B- Borrow
    JNB-Jumpwithnoborrow

5. **Programbranchinginstruction:** Theinstructionunderthisgrouparebasicallyusedfor conditionalorunconditionaljumpoperationortocalltoaparticularmemoryaddressor to return to the particular memory address.

   E.g.CALL,JMPetc.

## ADDRESSINGMODEOF8051 MICROCONTROLLER

Itisthetechniquethroughwhichwearespecifyingdataforoperationorhowtheoperanddata is specified accordingly in 8086 microprocessors. There are 8 different addressing mode according to the type of operation it performs.

1. Registeraddressingmode
2. Immediateaddressing mode
3. Directaddressingmode
4. Registerindirectaddressingmode
5. Baseregister+Indexregisteraddressingmode

1. **Registeraddressingmode:**Inthistypeofaddressingmodetheoperanddataisnot directly specified in the instruction itself but it is specified by some register.

   E.g.*MOVA,$R_0$*-Movethecontentof register$R_0$to accumulator.

2. **Immediateaddressingmode:**Inthistypeofaddressingmodetheoperanddataisdirectly specified in the instruction itself.

   E.g.MOVA,#20-Moveimmediatelythedata $20_H$totheaccumulator.

3. **Directaddressingmode:**Indirectaddressingmodetheoperandaddressisdirectly specified in the instruction itself.

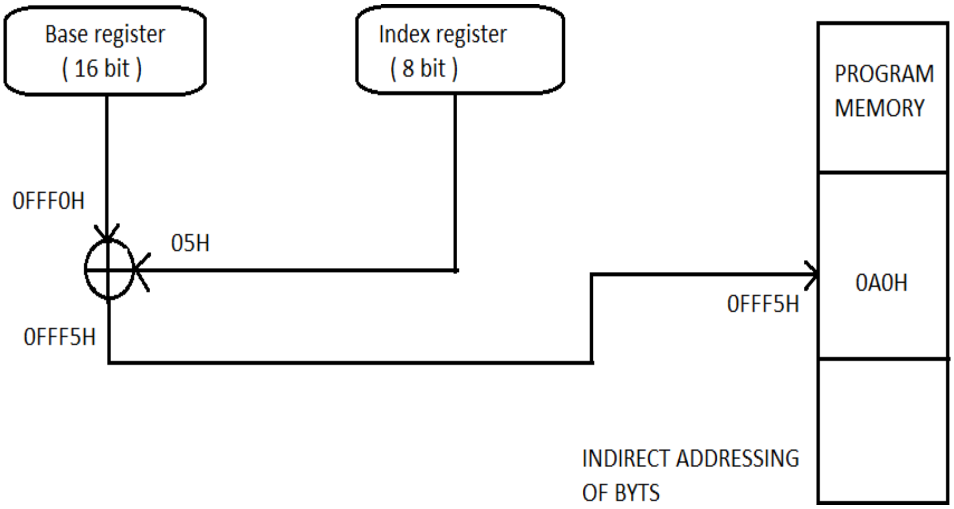
   E.g.

   *MOVA,$54_H$*-Movethecontentofmemoryaddress$54_H$toaccumulator.

4. **Registerindirectaddressingmode:** Inthistypeofaddressingmodetheoperanddatais not directlytransferred tothe accumulator; at first it stored insomememoryaddress and the transferred to the accumulator.

   E.g.*MOVC,$54_H$*-Movethecontentofmemoryaddress$54_H$ toregisterC. MOV A, C- Move the content of register C to the accumulator.

5. **Baseregister+Indexregisteraddressingmode:**Itisthecombinationofbaseaddressing modeandindexaddressing mode.



## Multiplechoicequestionanswer:

1. RAMis
   A) non-volatile
   B) Interfacing
   C) Volatile
   D) Noneofthese
   Answer: C
2. ROMis
   A) Volatile
   B) interfacing
   C) non-volatile
   D) Noneofthese
   Answer: C
3. TheMicrocontrollerisusedfor
   A) Controlorientedactivity
   B) program-orientedactivity.
   C) Transferorientedactivity.
   D) Noneofthese
   Answer: A
4. Howmanypinsaretherein8051 microcontrollers.
   A) 124
   B) 28
   C) 30
   D) 40

Answer:D

5. Theinternalclockfrequencyof8051microcontrolleris

A) 10 MHz

B) 30

C) 10

D) 12

Answer:D

6. Outof 40pinsin8051microcontrollerthe no.Ofpinsusedforportis

A) 16

B) 32

C) 24

D) noneofthese

Answer: B

7. In8051microcontrollertheexternalclockfrequencycanbeincreasedto

A)20MHZ

B)12

C)10

D)noneofthese

Answer: A

8. Microprocessorisusedfor

A) programoriented

B) interfacing

C) controloriented

D) Noneofthese

Answer: A

9. Thefunctionof latches

A) storingresult

B) connectingports

C) Data

D) Dataandaddress

Answer: D

10. In8051microcontrollerthetimers are

A) Timer0

B) Timer1

C) Timer0 andTimer1

D) Noneofthese

Answer: C

# ASSIGNMENTFULLMARKS-100

**SECTION-A**

**(ANSWERALLQUESTIONS)**

**Shortanswer type question: 2*8=16**

a) What are the different ports are in 8051 microcontroller and what is the function of the ports in 8051 microcontroller?

b) Writethedifferencebetweenthemicroprocessorandmicrocontroller.

c) WhatisthefunctionofXTAL$_1$andXTAL$_2$in8051microcontroller.

d) HowmuchRAMareusedfortheregisterbankselectionoption?

e) Howmanytypes of addressingmodesaretherein8051microcontroller?

f) Whichcontrolwordbitisusedforregisterbank selection?

g) Whatisthefunction oftimeranditisofhowmanybits?

h) WhatisthefunctionofTMO register?

**SECTION-B**

**Focusedanswertypequestion: 6*6=36**

a) Explainthedifferentinstructionsetsof8051 microcontroller.

b) Whataretheregisterbanksandexplaintheselectionofregisterbankin8051 microcontroller?

c) Drawthepindiagramof8051microcontroller.

d) Explainthedifferentaddressingmodesof8051microcontroller.

e) Describetheimportanceof8051microcontroller.

f) Explainthepindescriptionof8051microcontroller.

**SECTION-C**

**Longanswertypequestion: 16*3=48**

a) Describetheinternal architectureof8051microcontroller.

b) Explainthefunctionoftimerand TMODregistergivesuitableexample.Writeaprogram to generate a square wave of 2KHz frequency.

c) (i)Explainthedifferentinstructioncyclesin8051microcontroller.

(ii)Explainthestatusflagin8051 microcontroller.

# 5thSemesterRegular/BackExamination2019-20

# MICROPROCESSOR & MICROCONTROLLER

## BRANCH : ELECTRICAL

## Max Marks : 100

## Time : 3 Hours

## Q.CODE:HRB163

**AnswerQuestionNo.1(Part-1)whichiscompulsory,anyEIGHTfromPart-IIandanyTWOfromPart-III. The figure in the right hand margin indicate marks.**

### Part-I

**Q1OnlyShortAnswerTypeQuestions(AnswerAll-10) (2×10)**

a) List outthecontrolandstatussignallinesavailablein8085.
b) What isthedifferencebetweenRETandRETI instruction?
c) OneusertransfersanASCIIcharacter"E"(45H)withnoparitybit,onestart bit,onestopbit.Findthetimetaketo transfer 1000 characters using 9600bps.
d) If carry=1&A=75HandB=3FHpriortoexecution ofSUBA,B, thenwhat will bethecontent ofAafterexecution.
e) Assumethecontentofaccumulatorare71HandCY=0.IllustratetheaccumulatorcontentafterRRCandRAR instructions.
f) Distinguishbetweeninterruptandpolling.
g) What isthesignificanceofALEpinin8051?
h) DistinguishbetweenRISCprocessorandCSICprocessor.
i) Toget a20µsdelay,whichvalueshould beloadedintoTHregisterusingmode1,where XTAL=11.0592MHz.
j) Determinethecontrolwordsfor 8255 PPI,when port A=output,port B=output,port $C_{lower}$=input,port $C_{upper}$= input.

### Part-II

**Q2OnlyFocused-ShortAnswerTypeQuestions-(AnswerAnyEightoutofTwelve) (6×8)**

**a)** Explainthevariousstepsofinstructiondecoding andexecutionin 8085.
**b)** WritetheprogramtotransferthebytesofROMspaceintoRAMlocationstarting at50H.
**c)** Drawthetimingdiagramforexecution oftheinstructionMVIA, 54H.
**d)** Writeasubroutinetogeneratedelayof220ms. Assumecrystalfrequency= 12MHz.
**e)** Howdoesdatatransferfrommemorytomicroprocessoroccurs?Explainindetail.
**f)** Describeallstepsofinterrupt processof 8085.
**g)** Dotheschematicdiagramtoshowthe minimuminterfacebetween acomputeranda peripheral.
**h)** Whatdoyoumeanbystackandbank1conflictwith referencetotheiraddressin8051?Whatstepsarebeing followed to overcome this problem?
**i)** Assumethat wehave4 bytesof hexadecimal data:35H,42H, 3FHand 52H.
   **a.** Findthechecksum byte.
   **b.** Performthechecksumoperationtoensuredata integrity.
   **c.** Ifthesecondbyte 42Hhasbeenchangedto22H, showhowchecksumdetectsthe error.

**j)** Assumethatthelowerthreebitsof P1 areconnectedtothreeswitches. WriteaprogramtosendtheASCII characters 0,1,2,3,45,6,7 based on the status of the switches.

**k)** Howdirect memorydatatransferoccursin8085?Discussusing suitablefigure.

**l)** Writeaprogramtoread200bytesofdatafromP1andsavethedatainexternalRAMstartingatRAM location 3000H.

**Part-III**

**OnlyLongAnswerTypeQuestions(AnswerAnyTwooutofFour)**

**Q3**Drawtheblockdiagramof the8259 and explainhowit canbeused forincreasingtheinterrupt capabilitiesof 8085. Explainhow8259readthestatusandchangetheinterruptmodeduringaprogramexecution.

**(16)**

**Q4**Describetheinternalhardwarearchitectureofintel8086 in detailsusing suitableschematics. **(16)**

**Q5**Writeanassemblylanguage programtodivideone16bitnumber with an8-bitnumber in8085. **(16)**

**Q6**Drawand explainthearchitecturedetailsof8051 and discussthedifferent addressing modesof 8051. **(16)**

# SOLUTION

**Q1**

**a)**Thesesignalsareusedtoidentifythenature ofoperation.Thereare3controlsignaland3statussignals.Three control signals are RD, WR &ALE and three status signals are IO/M, S0 & S1.

**j)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

= $89_H$

**Q2**

**a)** Theinstructions whicharetobeexecuted by microprocessor arefirst stored inthememory of the processor and then executed. But the processor does not execute the instructions directly. It reads the instruction byte by byte and then executes it.

*Consider MVI A, 18H. when the instruction* is to be executed, the microprocessor gets the OpcodeforMVIA andperformsthe necessaryoperationonthedatawhichis18Hinthis case. The Opcode for MVI A is 3EH. So the microprocessor first reads this Opcode from the instruction and then performs the operation specified by Opcode over the data given.

Nowletusassumewewantto storethe aboveinstructionin aspecificaddresssay5500H.We know that in 8085 processor only one byte can be stored in each address location. Therefore the Opcode 3EH is stored at the location 5500H and the data 18H is stored at the next location 5501H.

Nowforexecutionofthisinstructiontheprocessor hastosendthe addresstothememoryfor reading. Then the MEMR' signal is activated. As soon as this signal is activated the memory places the Opcode byte (3EH) on the data bus.

TheaboveprocessisconsideredasasinglecycleandiscalledtheOPCODEFETCHCYCLE.The period during which the Opcode is fetched from address to the data bus is called as Opcode fetch cycle.

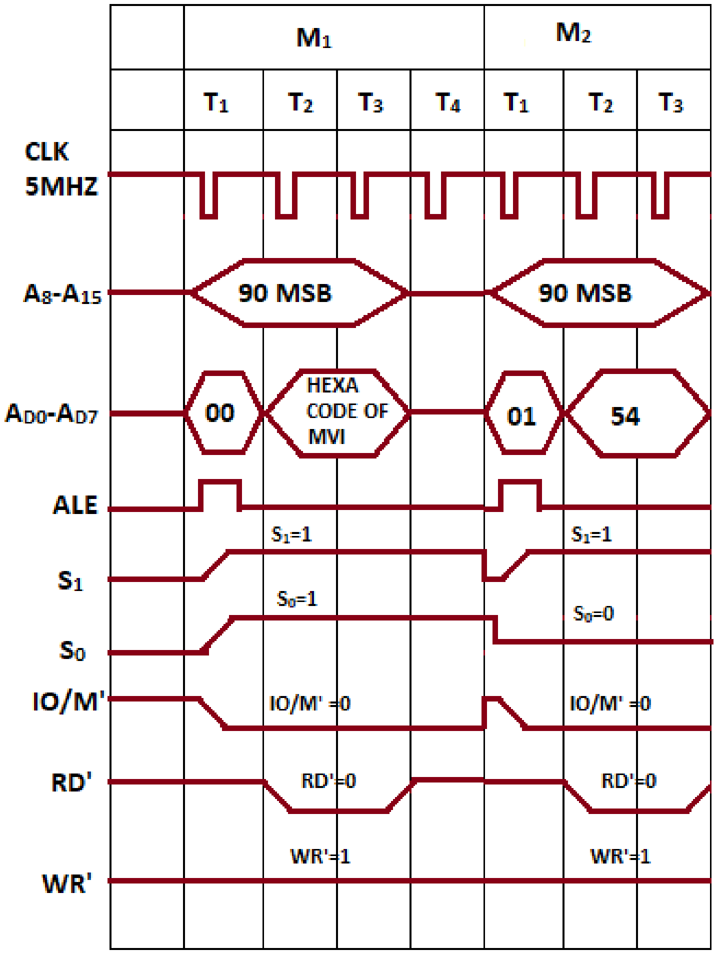
A microprocessor by default knows that the first byte which is under execution is always Opcode.The internaldatabussendstheOpcodeto theinstructiondecoder.Theinstruction decoder decodes the Opcode and identifies it as MVI A instruction. As soon as this information is obtained, the microprocessor searches for the data on which this operation should be performed.

To find the data for performing the operation the microprocessor instructs the timing and control unit to generate a proper timing signal to obtain the data. As a result of the timing signal the program counter is increased by 1. So, the address bus moves from 5500H to 5501H. Now we know that the data 18H is placed at 5501H. So, when the address bus is placed at 5501H, it identifies the data and the MEMR' signal is activated. After the activation of this signal the data is placed on the internal address bus and then it is moved to the accumulator.ThentheMVIAoperationisperformedonthe data18H andthe resultissentto the respective registers.

Thisprocess ofplacingthe addressandreadingthedataisconsideredasasinglecycleand thiscycleiscalledMemoryReadcycle.Ingeneral,thesecycles arecalledasmachinecycles.

**b)**

c)



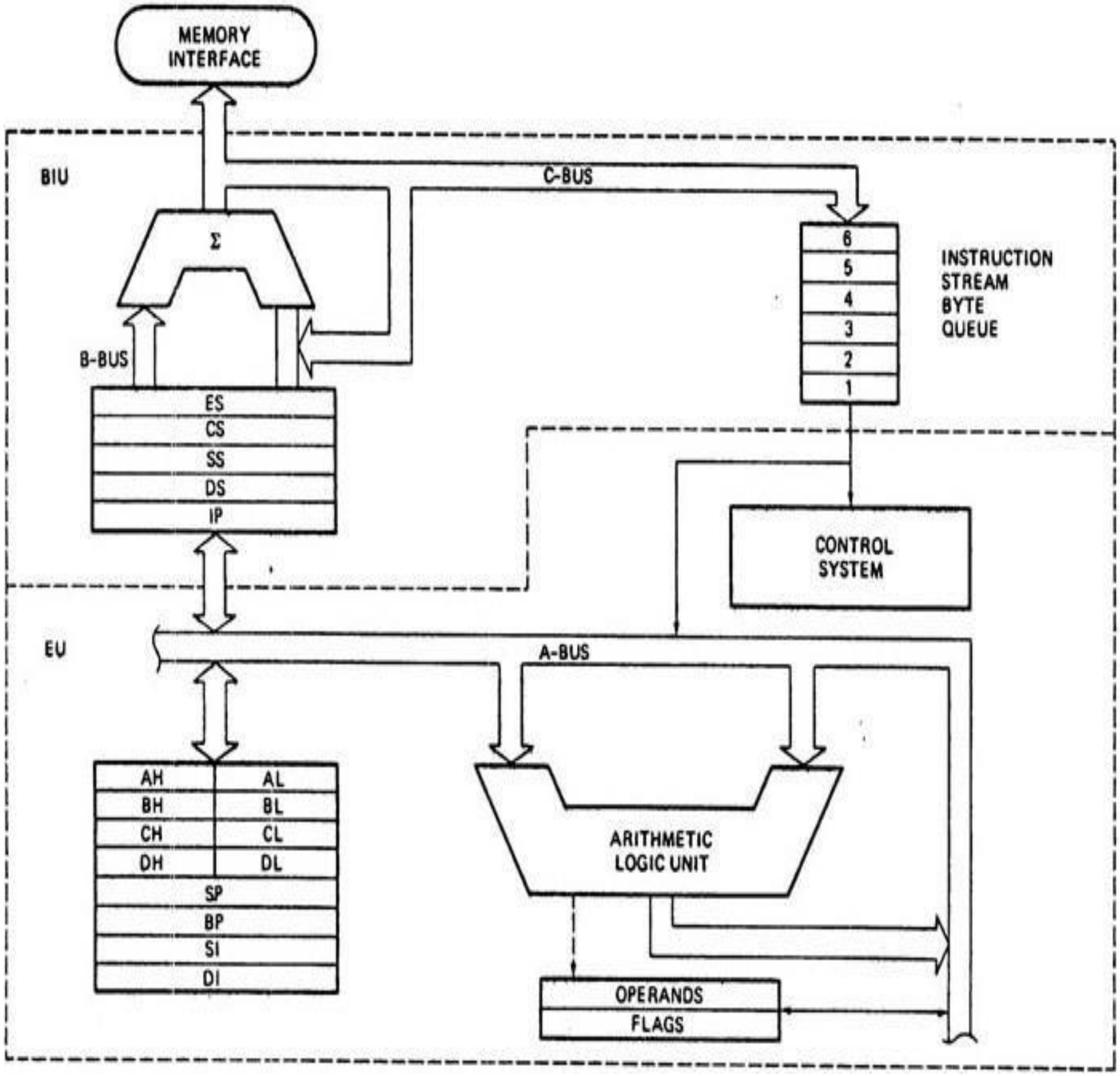
f)

**Q4**

Thetotalinternalarchitectureof8086microprocessorcanbebasicallydividedintotwodifferent units.

***1.BusInterfacesUnit(BIU)***

***2.Execution Unit (EU)***



## 1.BusInterfacesUnit (BIU):

- Itisresponsiblefortransferofdataandaddressbetweentheprocessor,memoryand input output device.
- It receives the data from the IO device and stored the data in a 6 bytes instruction queues in FIFO sequence and this data is transferred to the execution unit for arithmetic and logical operation.
- Thefunctionofdifferentunitsofbusinterfaceunitare

**6bytesinstructionqueue**

- Itsfunctionistoreceive6no.of8bitdataatatimeandstoresthedatainitand then this data can be transferred to execution unit for performing arithmetic and logical operation i.e. execution operation.
- The data is received from IO device to the 6 bytes instruction queue in FIFO sequence.

**Segmentregister**

- Thereare4segmentregister
  - Code segment register (CS): It basically used to store the opcode of aninstruction.
  - Data segment register (DS): It basically used to store the operand of an instruction.
  - Extra segment register (ES): It is basically used to store the character or string instruction such as consonant, vowel, character etc.
  - Stacksegmentregister(SS):Stackisasetofmemorylocationwhose address is different from main memory address.
    - ✓ We can transfer the data from main memory to stack memory by pushinstructionandwecanreceivethedatafromstackmemoryto main memory by pop instruction. So, segment register is basically used to store the stack memory value.
    - ✓ To locate a particular memory address, we take the help of stack pointer and given by the commandLXI SP $9605_H$.

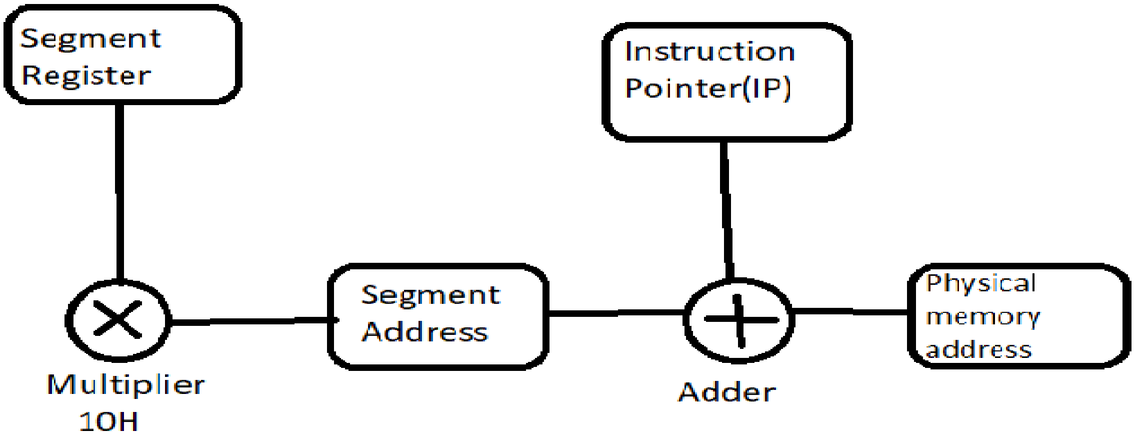
**Instructionpointer(IP)**

- Its functionis sameas that ofprogram counter of8085microprocessor and is basicallyused tocheck whethertheaddressfornext instructionisavailable or not. So, it stores the OFF-SET address.

**Buscontrolandaddressgeneration**

- It is basically used to generate 20 bits effective memory address or physical memory address.
- One address is generated from the segment register which is of 16 bit and when the address goes to the bus control and address generation it gets multiplied by the multiplier circuit of value $10_H$. So, at the output we get a segment address of 20 bit.
- Another address is generated from the instruction pointer which is of 16 bit and is known as OFF-SET address or assembly line address.

- So, whenthis address goes tothe bus control and address generation, it gets added ofwiththehelp ofadder circuit whichis present insidethebus control and address generation.
- Hencethe 20-bitsegmentaddressisaddedupwiththe16-bitIPaddresswith the help of adder circuit and at the output we get a 20-bit effective memory address or physical memory address.
- ***Effectivememoryaddress(EMA)orPhysical memoryaddress(PMA)=Segment address*10$_H$ +Instruction pointer***
- Theblockdiagramofphysicaladdressgenerationisshownasfollows.



***[Blockdiagramofphysical address generations]***

## 2. ExecutionUnit(EU):

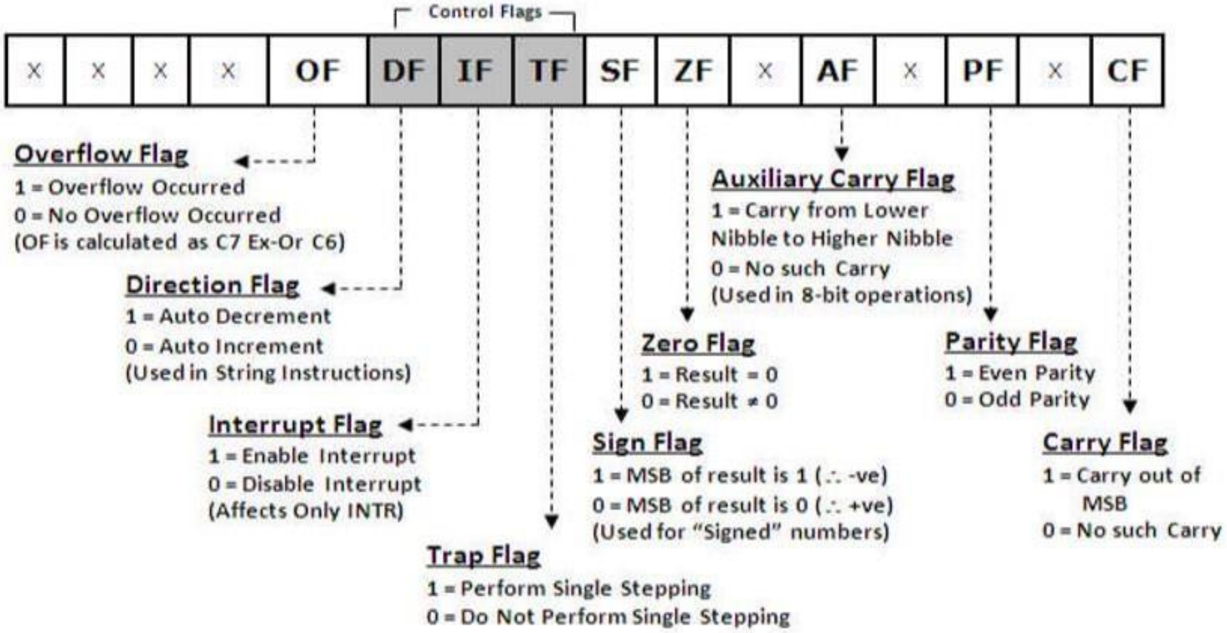
- Theexecutionunitreceivestheopcodeofaninstructionfromthe6byteinstruction queuedecodesitandperform thearithmeticandlogicaloperationandstores the result.
- Thefunctionofdifferentunitsofexecutionunitsare

**ALU**

- It is basically used to perform the arithmetic and logical operation such as addition, subtraction, multiplication, division, increment, decrement, comparison.
- Afterthearithmeticandlogicaloperation,theresultischeckbythestatusflag.

## StatusFlag

- Thestatusflagsarebasicallyusedtocheckwhethertheresultiswritingorwrong.
- Accordingly,in8086microprocessorthereare9activestatusflagsoutofwhich6are conditional flag and 3 are control flag.



- **Conditionalflag:**Conditionalflagsarethosewheretheoutputdependsuponthe input.Theconditionalflagsare*Signflag,zeroflag,Auxiliaryflag,Parityflag,Carry flag and overflow flag.*
    - **Sineflag:**Afterthearithmeticoperationiftheresultisnegativethensign flag is tends to logic 1 otherwise it will tend to logic 0.
    - **Zeroflag:**Afterthearithmeticoperationiftheresultiszerothenzeroflag will tend to logic 1 otherwise it will tend to logic 0.
    - **Auxiliaryflag:**Afterthearithmeticoperationifthereisacarryfrom3rdto 4thbitthenauxiliarycarrywilltendstologic1otherwiseitwilltendtologic 0.
    - **Parityflag:**Afterthearithmeticoperationiftheresultofthesumcontains evenno. of1'sthenparityflagwilltendtologic1otherwiseitwilltendto logic 0.
    - **Carry flag :** After the arithmetic operation if the result is more than 8 bit thentherewillbeacarryfrom7-8bitsocarryflagwilltendstologic1and incaseof16bitoperationifthereisacarryfrom15to16bitthecarryflag will tends to logic 1 otherwise it will tends to logic 0.

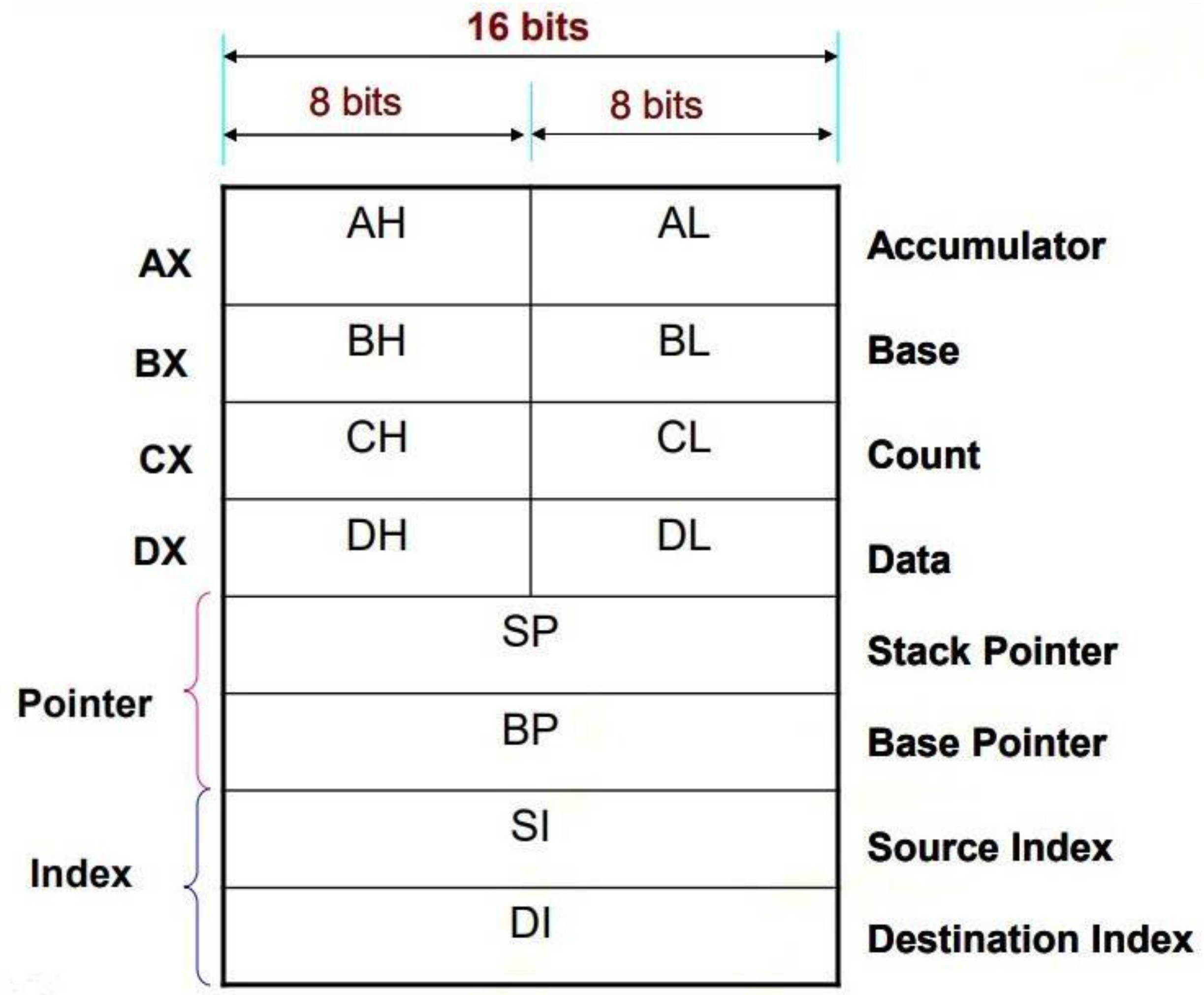
- **Overflowflag:**Afterthearithmeticoperationiftheresultismorethan16 bits,in that case the data cannot be stored in accumulator or destination register.So,inthatcasetheoverflowflagwilltendstologic1otherwiseit will tend to logic 0.

So,inconditionalflagthe outputisdependsontheinput

- **Controlflag:**Thecontrolflagsarebasicallyusedforcontrol-orientedactivitysuch as to stop the program, set or reset the operation, status flag manipulation, interruptoperation.So,inthiscaseitisusedforcontrol-orientedactivityandhere theoutputisdoesnotdependsuponinput. ThecontrolflagsareDirectionalflag, Interrupt flag, Trap flag.
    - **Directionalflag:**Incaseofcharacterorstringoperationinthatcasethe directional flag will tends logic 1 otherwise it will tend to logic 0.
    - **Interrupt flag and Trap flag:** These two flags are basically used as interrupt operation.

**Resistors**

- There are two types of register in 8086 microprocessor such as General-purposeregister and Special purpose register.



- **General-purpose register:** The general-purpose register are $A_H$ and $A_L$ i.e. A higher order and A lower order. Similarly, $B_H$ and $B_L$, $C_H$ and $C_L$, $D_H$ and $D_L$ respectively. Eachregister canstoreindividually 8bit of dataand combine form it can store 16bit of data so,

***$A_H$+$A_L$=$A_X$(16bit) $B_H$***
***+ $B_L$ = $B_{X(}$16 bit) $C_H$***
***+ $C_L$ = $C_{X(}$16 bit)***
***$D_H$+$D_L$=$D_{X(}$16bit)***

e.g.***MOV AH,08$_H$***-Moveimmediatelythedata08to$A_H$register.

***MOVAL,08$_H$***-Moveimmediatelythedata08to$A_L$ register.

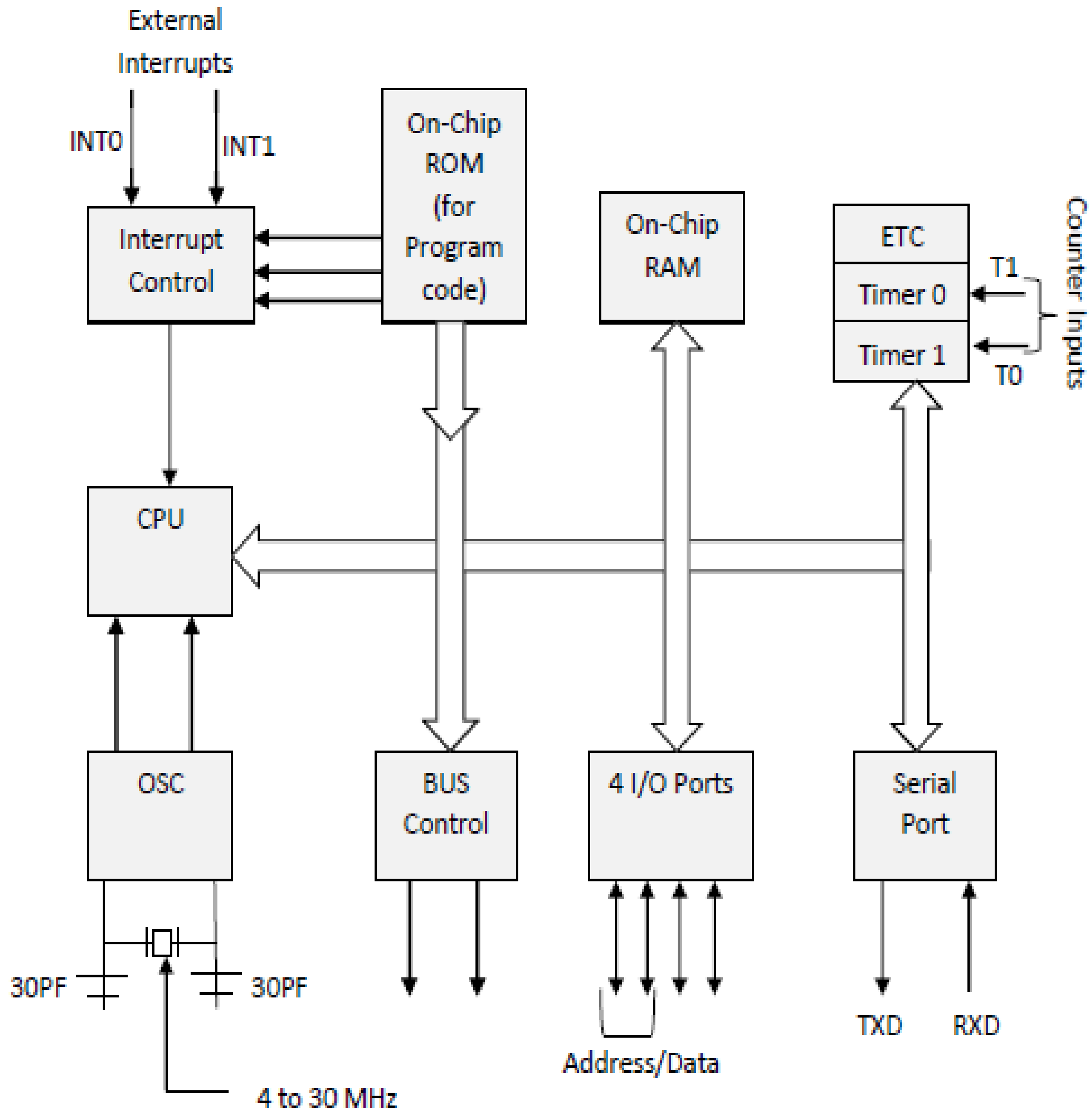
***MOVAX, 1264$_H$***-Movethe16-bitdata1264HtoA$_x$register.

- Since these registers are commonly used for storing the data temporarily for any arithmetic and logical operation so these are known as general-purpose register.

- **Special purpose register:** The special purpose registers are Stack pointer (SP), Base pointer (BI), Source index register (SI),Destination index register (DI). These are known as special purpose register because they are used for some specific operation and these are 16-bit registers.

  - **Stack pointer (SP):** Through stack pointer we can locate to a particularstack pointer address e.g**. LXI SP, 9505$_H$.**
  - **Base pointer (BP):** It is basically use to store the OFF-SET address (value of instruction pointer address).
  - **Sourceindexregister(SI):**Itisbasicallyusedtostorethestringaddress e.g.***MOV SI,[2000$_H$]***-MovetheOFF-SET address2000$_H$toSI register.
  - **Destination index register (DI):** It is basically used to store the end address e.g. **MOV DI, [2005$_H$]**-Move immediately the OFF-SET address 2005$_H$ to DI register.

**Q6**



Thetotalinternalarchitectureof8051microcontroller basicallyconsistsoffollowingunitssuch as

### 8. RAM

- Itconsistsof128bytesofRAMoutofwhich32bytesareusedforregisterbank selection.

- Thereare4registerbanksi.e.Bank-0,Bank-1,Bank-2,Bank-3andtheseregisterbanks canbe selected with the help of status control word i.e. $RS_0$ and $RS_1$ whichis present inside the status flags 8051 microcontroller.
- So, with the help of logic bits we can select a particular register bank and the mode of operation.

9. **ROM**

- it is basically used to store the predefine data or library function and its memory capacity is 4kB.

10. **CLKfrequency/ Oscillator**

- Inbuilt crystal oscillator is present inside the microcontroller which provide internal CLK frequency up-to 12MHz.

11. **Ports**

- Thereare4portssuchasPort-0,Port-1,Port-2,Port-3andeachportcantransferand receives 8-bit of data.
- Itisaconnectingpointorinterfacebetweentheprocessorandtheexternal device.

12. **Timerandcounter**

- There are 2 timers i.e. timer-0 and timer-1 which consists of 16 bits registers and is basically used for time delay operation and providing a matching clock frequency.

13. **Processor**

- Itisbasicallyusedforarithmeticadlogicaloperationandforstoringtheresult,tothe processor status flag is attached which is basically used to check the status of the output program or for checking the result.
- In8051microcontrollerthestatusflagconsistsof8-bitsoutofwhich6aredefineand 2 are undefined pins.
- ThedefinedpinsareCarry flag,Auxiliarycarry,$RS_0$,$RS_1$,Overflowflag,Parity flag.
- $RS_0$& $RS_1$ are basically used for register bank selection or for selecting a particular bank for storing the data. If no bank is selected the by default Bank-0 is selected.

| **$RS_1$** | **$RS_0$** | **REGISTERBANK** |
|---|---|---|
| 0 | 0 | BANK-0 |
| 0 | 1 | BANK-1 |
| 1 | 0 | BANK-2 |
| 1 | 1 | BANK-3 |

- In case of any arithmetic operation if the result is more than the destination register valuethenthedatacannot bestored inaccumulator sointhis casetheoverflow flag will tends to logic-1.

14. **Interruptcontrol**

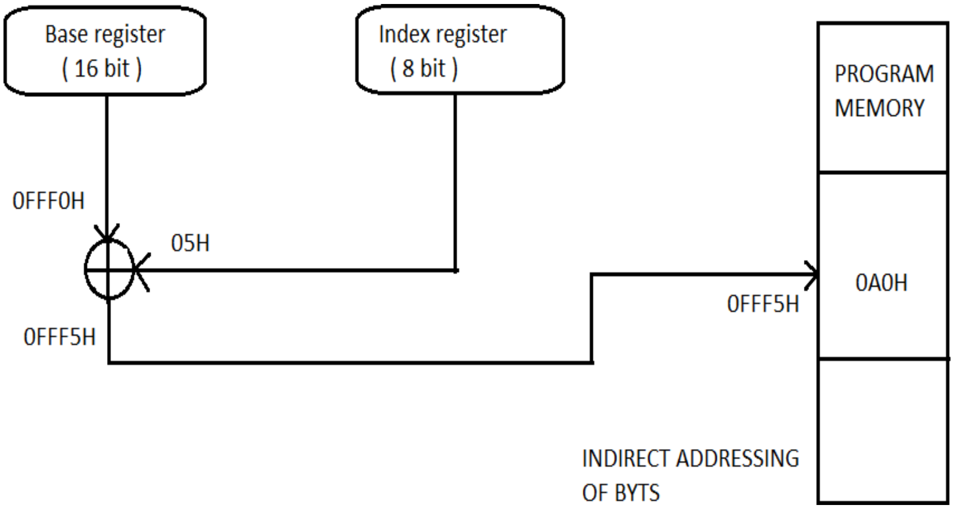
- Basically, it is used for interrupt operation. It has got different interrupt pins such as $INT_0$, $INT_1$ etc. through which we perform interrupt operation.

**ADDRESSINGMODEOF8081 MICRCONTROLLER**

Itisthetechniquethroughwhichwearespecifyingdatafor operationorhowtheoperanddata is specified accordingly in 8086 microprocessors. There are 8 different addressing mode according to the type of operation it performs.

6. Registeraddressingmode
7. Immediateaddressing mode
8. Directaddressingmode
9. Registerindirectaddressingmode
10. Baseregister+Indexregisteraddressingmode

6. **Register addressingmode:** Inthistypeof addressing modetheoperand datais not directly specified in the instruction itself but it is specified by some register. E.g.*MOVA,$R_0$*-Movethecontentof register$R_0$to accumulator.
7. **Immediateaddressingmode:**Inthistypeofaddressingmodetheoperanddata is directly specified in the instruction itself. E.g.MOVA,#20-Moveimmediatelythedata $20_H$totheaccumulator.
8. **Directaddressingmode:**Indirectaddressingmodetheoperandaddressis directly specified in the instruction itself. E.g. *MOVA,$54_H$*-Movethecontentofmemoryaddress$54_H$toaccumulator.
9. **Registerindirectaddressingmode:**Inthistypeofaddressingmodetheoperand data is not directly transferred to the accumulator; at first it stored in some memory address and the transferred to the accumulator. E.g.*MOVC,$54_H$*-Movethecontentofmemoryaddress$54_H$ toregisterC. MOV A, C- Move the content of register C to the accumulator.
10. **Baseregister+Indexregister addressingmode:**Itisthecombinationof baseaddressingmodeandindexaddressingmode.


Base register
( 16 bit )
Index register
( 8 bit )
PROGRAM
MEMORY
0FFF0H
05H
0FFF5H
0FFF5H
0A0H
INDIRECT ADDRESSING
OF BYTS