Government polytechnic kendrapara

DEPARTMENT

OF

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

LECTURE NOTES

Semester : 4th

Subject: MICROPROCESSOR & MICROCONTROLLER (TH-3)

Prepared by : Priyanka Dhal, (PTGF, Electronics &Telecommunication Engg.)

VISION OF THE DEPARTMENT-

To be a center of excellence in the field of E&TC Engineering by providing quality technical education.

MISSION OF THE DEPARTMENT-

1. To create an excellent teaching learning environment for making the students acquire the knowledge needed.
2. To inculcate self-learning attitude, entrepreneurial skill.
3. To impart knowledge required for recent and advanced engineering.

PROGRAM EDUCATIONAL OBJECTIVE (PEO)-

1. Recognize and apply the acquired fundamental knowledge in basic science and mathematics in solving E&TC Engineering problems.
2. To gain employment in public and private sector organization.
3. Involve in higher study and career enhancement.

PROGRAM SPECIFIC OUTCOME (PSO)-

1. To design, test and troubleshoot the simple analog and digital circuits.
2. An ability to solve complex E&TC Engineering problems using various tools i.e. hardware and software.
3. To pursue higher studies or get placed in various industries.

COURSE OUTCOME (CO)-

After the completion of the course the students will be able to

1. Comprehend the architecture and pin diagram of 8085, 8086 microprocessor and 8051 micro controller
2. Analyze the addressing modes and different instructions of 8085, 8086 microprocessor and 8051 micro controller.
3. Develop programming skill in assembly language using 8085, 8086 microprocessor and 8051 micro controller.
4. Draw the timing diagram of various instructions
5. Analyze electrical circuitry to the microprocessor I/O ports in order to interface the processor to external devices.

Unit-1: Microprocessor (Architecture and Programming-8 bit-8085)

1.1 Introduction to Microprocessor and Microcomputer & distinguish between them.
1.2 Concept of Address bus, data bus, control bus & System Bus
1.3 General Bus structure Block diagram.
1.4 Basic Architecture of 8085 (8 bit) Microprocessor
1.5 Signal Description (Pin diagram) of 8085 Microprocessor
1.6 Register Organizations, Distinguish between SPR & GPR, Timing & Control Module,
1.7 Stack, Stack pointer & Stack top.
1.8 Interrupts:-8085 Interrupts, Masking of Interrupt (SIM,RIM)

Unit-2: Instruction Set and Assembly Language Programming

2.1 Addressing data & Differentiate between one-byte, two-byte &three-byte instructions with examples.
2.2 Addressing modes in instructions with suitable examples.
2.3 Instruction Set of 8085(Data Transfer, Arithmetic, Logical, Branching, Stack& I/O , Machine Control)
2.4 Simple Assembly Language Programming of 8085
2.4.1 Simple Addition & Subtraction
2.4.2 Logic Operations (AND, OR, Complement 1's & 2's) & Masking of bits
2.4.3 Counters & Time delay (Single Register, Register Pair, More than Two Register)
2.4.4 Looping, Counting & Indexing (Call/JMP etc). 2.4.5 Stack & Subroutinesprogrames.
2.4.6 Code conversion, BCD Arithmetic & 16 Bit data Operation, Block Transfer.
2.4.7 Compare between two numbers.
2.4.8 Array Handling (Largest number & smallest number in the array)
2.5 Memory & I/O Addressing,

Unit-3: TIMING DIAGRAMS.

1.1 Define opcode, operand, T-State, Fetch cycle, Machine Cycle, Instruction cycle & discuss the concept of timing diagram.
1.2 Draw timing diagram for memory read, memory write, I/O read, I/O write machine cycle.
1.3 Draw a neat sketch for the timing diagram for 8085 instruction (MOV,MVI,LDA instruction).

Unit-4 Microprocessor Based System Development Aids

4.1 Concept of interfacing
4.2 Define Mapping &Data transfer mechanisms - Memory mapping & I/O Mapping
4.3 Concept of Memory Interfacing:- Interfacing EPROM & RAM Memories
4.4 Concept of Address decoding for I/O devices
4.5 Programmable Peripheral Interface: 8255

4.6 ADC & DAC with Interfacing.
4.7 Interfacing Seven Segment Displays
4.8 Generate square waves on all lines of 8255
4.9 Design Interface a traffic light control system using 8255.
4.10 Design interface for stepper motor control using 8255.

Unit-5 Microprocessor (Architecture and Programming-16 bit-8086)

5.1 Register Organisation of 8086
5.2 Internal architecture of 8086
5.3 Signal Description of 8086
5.4 General Bus Operation& Physical Memory Organisation
5.5 Minimum Mode &Timings,
5.6 Maximum Mode &Timings,
5.7 Interrupts and Interrupt Service Routines, Interrupt Cycle, Non-Maskable Interrupt, Maskable Interrupt
5.8 8086 Instruction Set & Programming: Addressing Modes, Instruction Set, Assembler Directives and Operators,
5.9 Simple Assembly language programming using 8086 instructions.

Unit-6 Microcontroller (Architecture and Programming-8 bit):-
6.1 Distinguish between Microprocessor & Microcontroller
6.2 8 bit & 16 bit microcontroller
6.3 CISC & RISC processor
6.4 Architectureof8051Microcontroller
6.5 Signal Descriptionof8051Microcontrollers
6.6 Memory Organisation-RAM structure, SFR
6.7 Registers,timers,interruptsof8051Microcontrollers
6.8 Addressing Modes of 8051
6.9 Simple 8051 Assembly Language Programming Arithmetic& Logic Instructions , JUMP, LOOP, CALL Instructions, I/O Port Programming
6.10 Interrupts, Timer & Counters
6.11 Serial Communication
6.12 Microcontroller Interrupts and Interfacing to 8255

UNIT-1:MICROPROCESSOR(ARCHITECHTUREANDPROGRAMMING -8085-8-BIT)

MICROPROCESSOR:

- AMicroprocessorisamultipurpose,Programmableclockdriven,registerbasedelectronicdevice,
- Thatreadbinaryinstructionfromastoragedevicecalledmemory,acceptsbinarydataasinputandprocessesdataaccordingtothoseinstructionsandprovidesresultsasoutputs.
- MicroprocessorisclockdrivensemiconductordevicewhichforismanufacturedbyusingLSIandVLSI technique.

MICROCOMPUTER:

- A**microcomputer**isasmall,relativelyinexpensivecomputerwithamicroprocessorasitscentral processing unit (CPU). It includes a microprocessor, memory, and input/output(I/O)facilities.

- Microcomputersbecamepopularinthe1970sand80swiththeadventofincreasinglypowerfulmicroprocessors.
- ExamplesofMicrocomputersareIntel8051controller-asingleboardcomputer,

- IBMPCand AppleMacintoshcomputer.

MICROCONTROLLER:

- A microcontroller is a small computer on a single integrated circuit containing a processorcore,memory andprogrammable input/output peripherals.
- Microcontrollers are usedinautomaticallycontrolledproductsanddevices,suchasautomobile engine control systems, implantable medical devices, remote controls, officemachines,powertools,toysandotherembeddedsystems.

DIFFERENCEBETWEENMICROCOMPUTERANDMICROPROCESSOR-
GeneralArchitectureofMicrocomputerSystem:

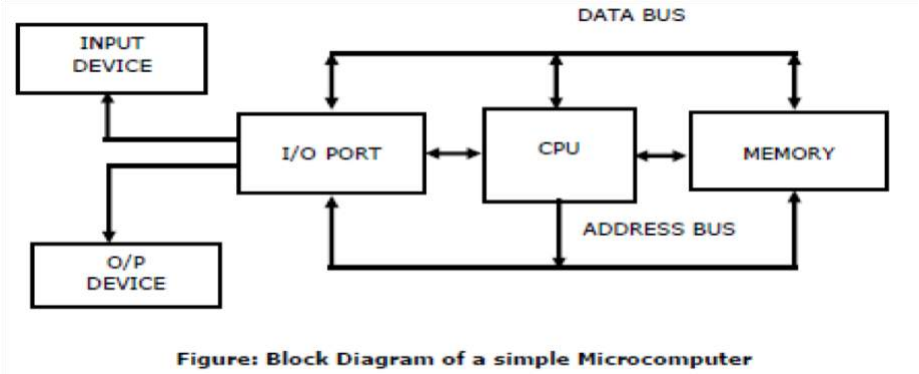


Figure: Block Diagram of a simple Microcomputer

ThemajorpartsareCPU,MemoryandI/O

Therearethreebuses,addressbus,databusandcontrolbus;

MEMORY:

- Memory consistofRAMandROM, the purposeofmemoryis tostorebinary codesforthesequencesof instructionsyou wantthe computertocarryout.
- Thesecondpurposeofthememoryistostorethebinary-codeddatawithwhichthecomputerisgoingtobe working.

INPUT/OUTPUT:

- Theinput/outputorI/OSectionallowsthecomputertotakeindatafromtheoutsideworldor senddata tothe outside world.
- Peripheralssuchaskeyboards,videodisplayterminals,printersareconnectedtoI/OPort.

CPU(CENTRALPROCESSINGUNIT):

- InamicrocomputerCPUisamicroprocessor.
- Thefetchesbinarycodedinstructionsfrommemory,decodestheinstructionsintoaseriesofsimpleactionsandcarriesouttheseactionsinasequenceof steps.
- The CPU also contains an address counter or instruction pointer register, which holds theaddressofthe nextinstructionor data itemtobefetchedfrommemory.

Architectureofmicroprocessor-

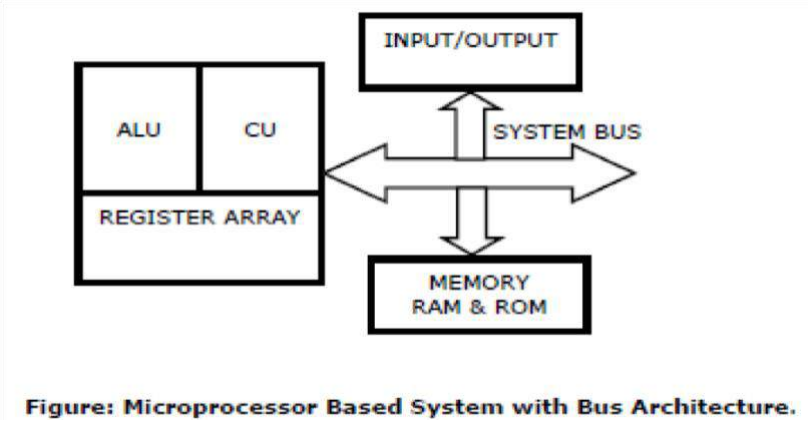


Figure: Microprocessor Based System with Bus Architecture.

Microprocessorisdividedintothreesegments-

1. ALU
2. Register

3. ControlUnit

ArithmeticLogicUnit:

- ThisistheareaofMicroprocessorwherevariouscomputingfunctionsareperformedonda ta.

- TheALUperformsoperationssuchasaddition,subtractionandlogicoperationssuchasA ND,ORandexclusive OR.

ControlUnit:

- TheControlUnitProvidesthenecessarytimingandcontrolsignalstoalltheoperationsint he Microcomputer
- ItcontrolstheflowofdatabetweentheMicroprocessorandMemoryandPeripherals.
- TheControlunitperforms2basictasks

  →Sequencing

  →Execution

Register:

- Thesearestoragedevicestostoredatatemporarily.

- Therearedifferenttypesofregistersdependinguponthemicroprocessor.
- Theseregistersareprimarilyusedtostoredatatemporarilyduringtheexecutionofaprogr amandare accessibletotheuser throughtheinstructions.

ADDRESSBUS:

- Theaddressbusconsistsof16,20,24or32 parallelsignallines.
- OntheselinestheCPUsendsouttheaddressofthememorylocationthatistobewrittento or read from. The no of memory location that the CPU can address is determined bythenumberof addresslines.

- If the CPU has N address lines, then it can directly address 2N memory locations i.e. CPUwith16addresslines canaddress216 or65536 memorylocations.

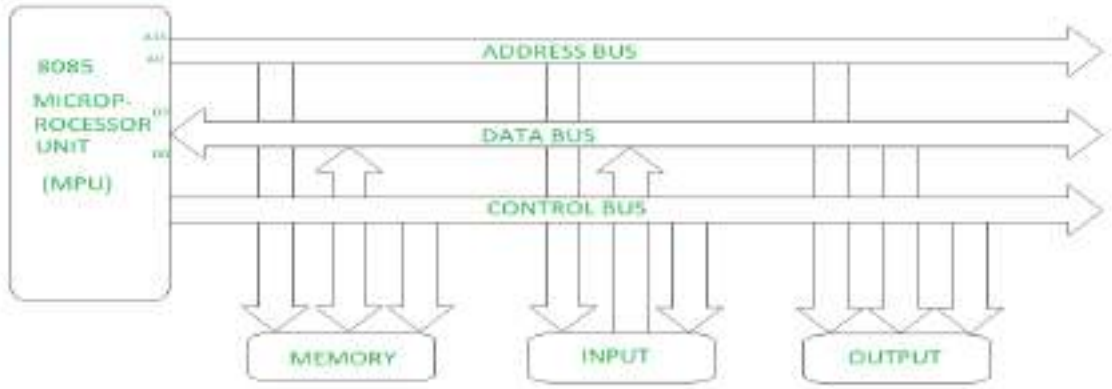
DATABUS:

- Thedatabusconsistsof8,16or32parallelsignallines.
- Thedatabuslinesarebi-directional.
- ThismeansthattheCPUcanreaddatainfrommemoryoritcansenddataouttomemory.
-

CONTROLBUS:

- Thecontrolbusconsistsof4to10parallelsignallines.
- TheCPUsendsoutsignalsonthecontrolbustoenabletheoutputofaddressedmemorydevices orportdevices.
- TypicalcontrolbussignalsareMemoryRead,MemoryWrite,I/ORea dandI/OWrite.

Busstructureblockdiagram:



Bus organization system of 8085 Microprocessor

ADDRESSBUS:

- Itisagroupofconductingwireswhichcarriesaddressonly.
- Addressbusisunidirectionalbecausedataflowinonedirection,frommicroprocessortomemoryo r from microprocessor toInput/output devices.
- LengthofAddressBusof8085microprocessoris16Bit(i.e.FourHexadecimalDigits),rangingfrom000 0 H toFFFFH,(HdenotesHexadecimal).
- Themicroprocessor8085cantransfermaximum16bitaddresswhichmeansitcanaddress65,536 different memorylocation.
- TheLengthoftheaddressbusdeterminestheamountofmemoryasystemcanaddress.
- Suchasasystemwitha32-bitaddressbuscanaddress2^32memorylocations.
- Ifeachmemorylocationholdsonebyte,theaddressablememoryspaceis4GB.However,theactuala mountofmemorythatcanbeaccessedisusuallymuchlessthanthistheoreticallimitduetochipset andmotherboardlimitations.

DATABUS:

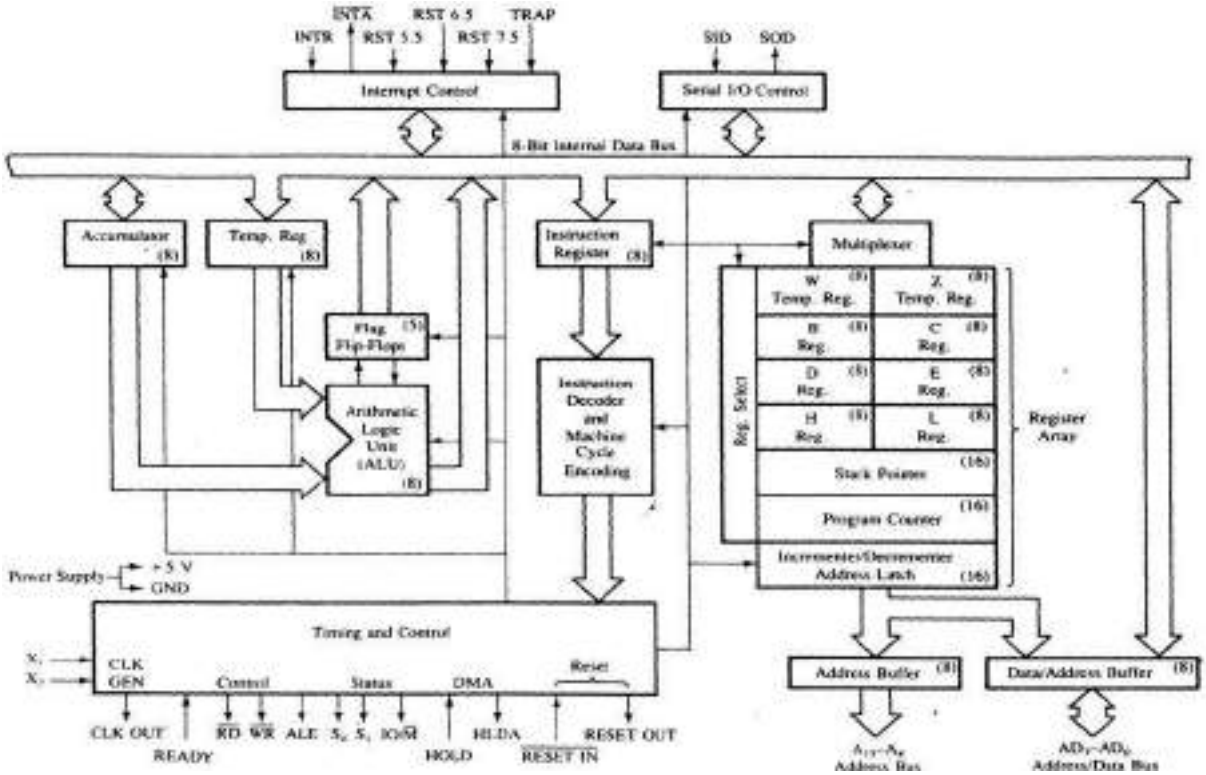
- ItisagroupofconductingwireswhichcarriesDataonly.
- Databusisbidirectionalbecausedataflowinbothdirections,frommicroprocessortomemoryorIn put/outputdevicesandfrommemoryorInput/outputdevicestomicroprocessor.
- LengthofDataBusof8085microprocessoris8Bit(Thatis,twoHexadecimalDigits),rangingfrom0 0 H toFFH.(H denotesHexadecimal).

- When it is write operation, the processor will put the data (to be written) on the data bus,when it is read operation, the memory controller will get the data from specific memoryblockand put itintothe databus.
- Thewidthofthedatabusisdirectlyrelatedtothelargestnumberthatthebuscancarry,suchas an 8 bit bus can represent 2 to the power of 8 unique values, this equates to the number0to255.A16 bit buscancarry0 to65535.

CONTROLBUS:

- It is a group of conducting wires, which is used to generate timing and control signals tocontrol all the associated peripherals, microprocessor uses control bus to process data i.e.what todowith selectedmemorylocation.Some controlsignalsare:
- Memoryread
- Memorywrite
- I/Oread
- I/OWrite
- Opcodefetch

ARCHITECTUREOF8085MICROPROCESSOR:



Accumulator:

Itisan8-bitregisterusedtoperformarithmetic,logical,I/O&load/storeoperations.Itisconnectedtointernal data bus&ALU.

Arithmeticandlogicunit:

Asthenamesuggests,itperformsarithmeticandlogicaloperationslikeAddition,Subtraction,AND,OR,etc.on8-bit data.

Generalpurposeregister:

- Thereare6generalpurposeregistersin8085processor,i.e.B,C,D,E,H&L.Eachregistercanhold8-bit data.
- Theseregisterscanworkinpairtohold16-bitdataandtheirpairingcombinationislikeB-C,D-E&H-L.

Programcounter:

- It is a 16-bit register used to store the memory address location of the next instruction to beexecuted.
- Microprocessor increments the program whenever an instruction is being executed, so thattheprogramcounterpointstothememoryaddressofthenextinstructionthatisgoingtobeexecuted.

Stackpointer:

Itis alsoa16-bitregisterworkslikestack,whichis always incremented/decrementedby 2duringpush &popoperations.

Temporaryregister:

Itisan8-bitregister,whichholdsthetemporarydataofarithmeticandlogicaloperations.

Flagregister:

Itisan8-bitregisterhavingfive1-bitflip-flops,whichholdseither0or1dependinguponthe result storedinthe accumulator.

Thesearethesetof 5flip-flops:

- Sign(S)
- Zero(Z)
- AuxiliaryCarry(AC)
- Parity(P)
- Carry(C)

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

Instructionregisteranddecoder:

- Itisan8-bitregister.
- WhenaninstructionisfetchedfrommemorythenitisstoredintheInstructionregister.
- InstructiondecoderdecodestheinformationpresentintheInstructionregister.

Timingandcontrolunit:

Itprovidestimingandcontrolsignaltothemicroprocessortoperformoperations.Followingareth etimingandcontrolsignals,whichcontrolexternalandinternalcircuits:-

- ControlSignals:READY,RD',WR',ALE
- StatusSignals:S0,S1,IO/M'
- DMASignals:HOLD, HLDA
- RESETSignals:RESETIN,RESETOUT

Interruptcontrol:

- Asthenamesuggestsitcontrolstheinterruptsduringaprocess.
- Whenamicroprocessorisexecutingamainprogramandwheneveraninterruptoccurs,themicrop rocessorshiftsthecontrolfromthemainprogramtoprocesstheincomingrequest.
- Aftertherequestiscompleted,thecontrolgoes backtothemainprogram.
- Thereare5interruptsignalsin8085microprocessor:INTR,RST7.5,RST6.5,RST5.5,andTRAP.

SerialInput/outputcontrol:

Itcontrolstheserialdatacommunicationbyusingthesetwoinstructions:SID(Serialinputdata)andS OD(Serialoutput data).

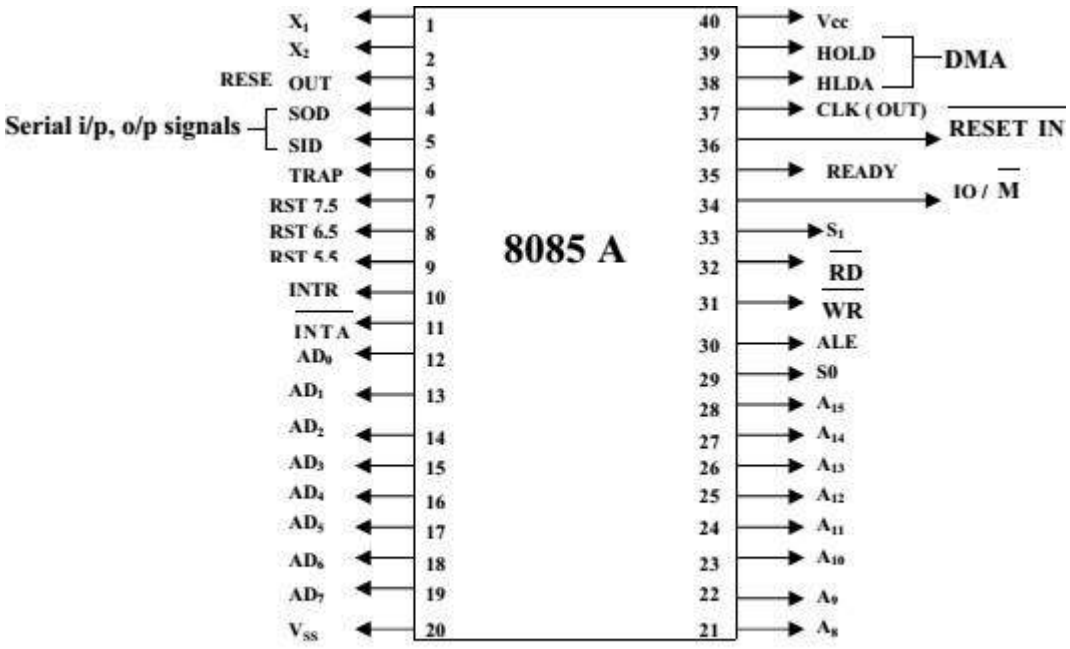
Addressbufferandaddress-databuffer:

- Thecontentstoredinthestackpointerandprogramcounterisloadedintotheaddressbufferandad dress-data buffer tocommunicate with theCPU.
- ThememoryandI/Ochipsareconnectedtothesebuses;theCPUcanexchangethedesireddatawith the memoryandI/Ochips.

Addressbusanddatabus:

Data bus carries the data to be stored. It is bidirectional, whereas address bus carries thelocationtowhereitshouldbestoredanditisunidirectional.Itisusedtotransferthedata&Addr essI/O devices.

PINDIAGRAMOF8085:



Pin Diagram of 8085

Thepinsofan8085microprocessorcanbeclassifiedintosevengroups:-

Addressbus:

A15-A8,itcarriesthemostsignificant8-bitsofmemory/IOaddress.

Databus:

AD7-AD0,itcarriestheleastsignificant8-bitaddressanddatabus.

Controlandstatussignals:

Thesesignalsareusedtoidentifythenatureofoperation.Thereare3controlsignaland3statussignals
.

ThreecontrolsignalsareRD',WR'&IO/M'.

RD':

ThissignalindicatesthattheselectedIOormemorydeviceistobereadandisreadyforacceptingdata available onthedata bus.

WR':

ThissignalindicatesthatthedataonthedatabusistobewrittenintoaselectedmemoryorIOlocati on.

IO/M':

ThissignalisusedtodifferentiatebetweenIOandMemoryoperations,i.e.whenitishighindicatesIOoperationandwhenit islowthenit indicatesmemoryoperation.

ALE:

Itisapositivegoingpulsegeneratedwhenanewoperationisstartedbythemicroprocessor.Whenthepulsegoes high,itindicatesaddress.Whenthepulsegoesdownitindicates data.

S1&S0:

Thesesignalsareusedtoidentifythetypeofcurrentoperation.

S1	S0	Operation
0	0	Halt
0	1	Write
1	0	Read
1	1	Fetch

Powersupply:

Thereare2powersupplysignalsV_{cc}&V_{ss}.VCCindicates+5vpowersupplyandVSSindicatesgroundsignal.

Clocksignals:

Thereare3clocksignals,i.e.X1,X2,CLKOUT.

X1X2:

Acrystal(RC,LCN/W)isconnectedatthesetwopinsandisusedtosetfrequencyoftheinternalclock generator. Thisfrequencyisinternallydividedby2.

CLKOUT:

Thissignalisusedasthesystemclockfordevicesconnectedwiththemicroprocessor.

Interrupts&externallyinitiatedsignals:

- Interruptsarethesignalsgeneratedbyexternaldevicestorequestthemicroprocessortoperform a task.

- There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. We will discussinterruptsindetailininterruptssection.

TRAP:

- It is a non-maskable interrupt, having the highest priority among all interrupts. By default, itisenableduntilitgetsacknowledged.Incaseoffailure,itexecutesasISRandsendsthedatatoback upmemory.Thisinterrupt transfersthecontroltothelocation0024H.

RST7.5:

- Itisamaskableinterrupt,havingthesecondhighestpriorityamongallinterrupts.Whenthisinterr upt is executed, the processor saves the content of the PC register into the stack andbranchesto003CH address.

RST6.5:

- It is a maskable interrupt, having the third highest priority among all interrupts. When thisinterrupt is executed, the processor saves the content of the PC register into the stack andbranchesto0034Haddress.

RST5.5:

- Itisamaskableinterrupt.Whenthisinterruptisexecuted,theprocessorsavesthecontentofthePC registerintothe stackandbranchesto002CH address.

INTR:

Itisamaskableinterrupt,havingthelowestpriorityamongallinterrupts.Itcanbedisabledbyreset tingthe microprocessor.

When**INTRsignalgoeshigh**,thefollowing eventscanoccur:

ThemicroprocessorchecksthestatusofINTRsignalduringtheexecutionofeachinstruction.

- WhentheINTRsignalishigh,thenthemicroprocessorcompletesitscurrentinstructionandsen dsactive lowinterrupt acknowledgesignal.
- Wheninstructionsarereceived,thenthemicroprocessorsavestheaddressofthenextinstructio nonstackandexecutesthereceived instruction.

INTA':

Itisaninterruptacknowledgmentsentbythemicroprocessorafter INTRisreceived.

RESETIN:

Thissignalisusedtoreset themicroprocessorbysettingtheprogramcountertozero.

RESETOUT:

Thissignalisusedtoresetalltheconnecteddeviceswhenthemicroprocessor isreset.

READY:

ThissignalindicatesthatthedeviceisreadytosendorreceivedataIfREADYislow,thentheCPU hastowaitfor READY togohigh.

HOLD:

Thissignalindicatesthatanothermasterisrequestingtheuseoftheaddressanddatabuses.

HLDA(HOLDAcknowledge):

Itindicatesthatthe CPUhasreceivedthe HOLDrequestanditwill relinquishthebusinthenextclock cycle.HLDAisset tolowafter theHOLD signalisremoved.

SerialI/Osignals:

Thereare2serialsignals,i.e.SIDandSODandthesesignalsareusedforserialcommunication.

SOD(Serialoutputdataline):

TheoutputSODisset/resetasspecifiedbytheSIMinstruction.

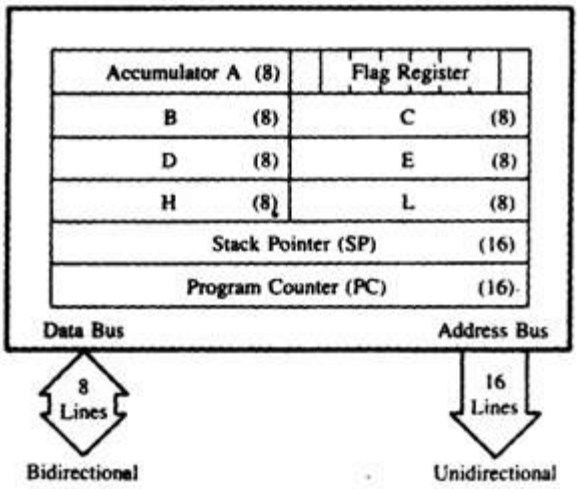
SID(Serialinputdataline):

ThedataonthislineisloadedintoaccumulatorwheneveraRIMinstructionis executed.

- When the INTR signal is high, then the microprocessor completes its current instruction andsendsactive low interruptacknowledgesignal.
- Wheninstructionsarereceived,thenthemicroprocessorsavestheaddressofthenextinstruction onstackandexecutesthereceived instruction.

REGISTERORGANIZATION:

Ithassixaddressable8-bitregisters:A,B,C,D,E,H,Landtwo16-bitregistersPCandSP.Theseregisterscanbe classifiedas:



- **GeneralPurposeRegisters**
- **TemporaryRegisters:**Temporarydataregister,WandZregisters
- **SpecialPurposeRegisters:**Accumulator,Flagregisters,Instructionregister
- **Sixteen-bitRegisters:**ProgramCounter(PC),StackPointer(SP)

1. GeneralPurposeRegisters:

- RegistersB,C,D,E,H,andLaregeneralpurposeregistersin8085Microprocessor.AlltheseGPRSare 8-bitswide.Theyarelessimportantthanthe accumulator.
- Theyareusedtostoredatatemporarilyduringtheexecutionoftheprogram.Forexample,thereisn oinstructiontoaddthe contentsofB and Eregisters.
- AtleastoneoftheoperandshastobeinA.ThustoaddBandEregisters,andtostoretheresultinBregist er,the followinghave tobedone.

➢ MovetoAregisterthecontentsofBregister.

➢ ThenaddAandEregisters.Theresultwill beinA.

➢ MovethisresultfromAregistertoBregister.

- Itispossibletousetheseregistersaspairstostore16-bitinformation.OnlyB-C,D-E,andH-Lcanform register pairs.
- Whentheyareusedas registerpairs inan instruction, theleftregisterisunderstoodto havetheMSBbyte andtherightregisterstheLSBbyte.
- Forexample,inD-Eregisterpair,thecontentoftheDregisteristreatedastheMSBbyte,andthecontent of Eregister istreatedasthe LSBbyte.

2. TemporaryRegisters:

- **TemporaryDataRegister:-**
- TheALUhastwoinputs.Oneinputissuppliedbytheaccumulatorandotherfromthetemporarydata register.
- Theprogrammercannotaccessthistemporarydataregister.However,itisinternallyusedforexecutionofmost ofthe arithmeticandlogicalinstructions.
- **WandZregister:-**WandZregistersaretemporaryregisters.Theseregistersareusedtohold 8-bitdataduringtheexecutionofsomeinstructions.Theseregistersarenotavailablefortheprogrammer since8085Microprocessor Architectureusestheminternally.

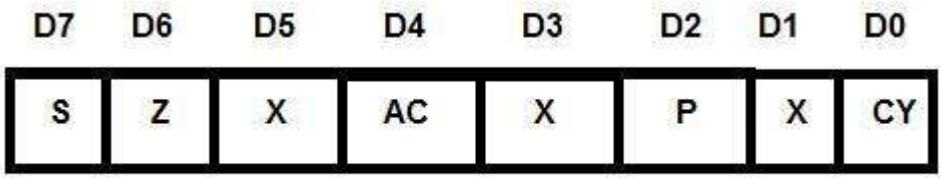
3. SpecialPurposeRegisters:

✓ **Accumulator(A):**

- Register A is an 8-bit register used in 8085 to perform arithmetic, logical, I/O & load/storeoperations.
- RegisterAisquiteoftencalledasanAccumulator.Anaccumulatorisaregisterforshort-term,intermediate storage of arithmetic and logic data in a computer's CPU (Central ProcessingUnit).
- Inanarithmeticoperationinvolvingtwooperands,oneoperandhastobeinthisregister.Andtheresultofthearithmeticoperationwillbe storedoraccumulatedinthisregister.
- Similarly,inalogicaloperationinvolvingtwooperands,oneoperandhastobeintheaccumulator. Also, some other operations, like complementing and decimal adjustment, canbeperformed onlyonthe accumulator.

✓ **FlagRegister:**

- It is a 3-bit register, in which five of the bits carry significant information in the form of flags:S(Signflag),Z(Zeroflag),AC(Auxiliarycarryflag),P(Parityflag),andCY(carryflag).



Flag Register of 8085

- **S-Signflag:-**
  Aftertheexecutionofarithmeticorlogicaloperations,ifbitD7oftheresultis1,thesignflagisset.InagivenbyteifD7is1,thenumberwillbeviewedasanegativenumber.IfD7 isU, thenumberwillbe consideredasapositive number.
- **Z-Zeroflag**:-Thezeroflagsetsiftheresultoftheoperationin ALUiszeroandflagresetsiftheresult is non-zero. The zero flags are also set if a certain register content becomes zerofollowinganincrementordecrement operationof that register.
- **AC-auxiliary Carry flag: -** This flag is set if there is an overflow out of bit 3 i.e. carry fromlower nibble to higher nibble (D3 bit to D4 bit). This flag is used for BCD operations and it isnotavailable for the programmer.
- **P-Parity flag: -** Parity is defined by the number of one's present in the accumulator. Afterarithmetic or logical operation, if the result has an even number of ones, i.e. even parity, theflagisset.If the parityisodd,the flagisreset.
- **CY-Carryflag:**-
  Thisflagissetifthereisanoverflowoutofbit7.Thecarryflagalsoservesasaborrowflagforsubtraction.Inboththeexamples shownbelow, thecarryflagisset.

✓ **InstructionRegister:**-

- In a typical processor operation, the processor first fetches the opcode of instruction frommemory (i.e. it places an address on the address bus and memory responds by placing thedatastoredat the specifiedaddressonthe databus).
- TheCPUstoresthisopcodeinaregistercalledtheinstructionregister.Thisopcodeisfurthersenttothe instructiondecodertoselect oneof the256 alternatives.

4. SixteenBitRegisters:

✓ **Programcounter(PC):-**

- Program is a sequence of instructions. Microprocessor fetches these instructions from thememoryandexecutesthem.
- The program counter is a special purpose register which, at a given time, stores the addressofthe next instructiontobe fetched.
- ProgramCounteractsasapointertothenextinstruction.
- How processor increments program counter depends on the nature of the instruction; forone-byteinstructionitincrementsprogramcounterbyone,fortwo-byteinstructionitincrements program counter by two and for three-byte instruction it increments programcounterbythreesuchthatprogramcounteralwayspointstotheaddressofthenextinstruction..

StackPointer(SP):-

ThestackisareservedareaofthememoryintheRAMwheretemporaryinformationmaybestored.A16-bitstackpointerisusedtoholdtheaddressofthemostrecentstackentry.

DISTINGUISHBETWEENGPRANDSPR:

GPR-

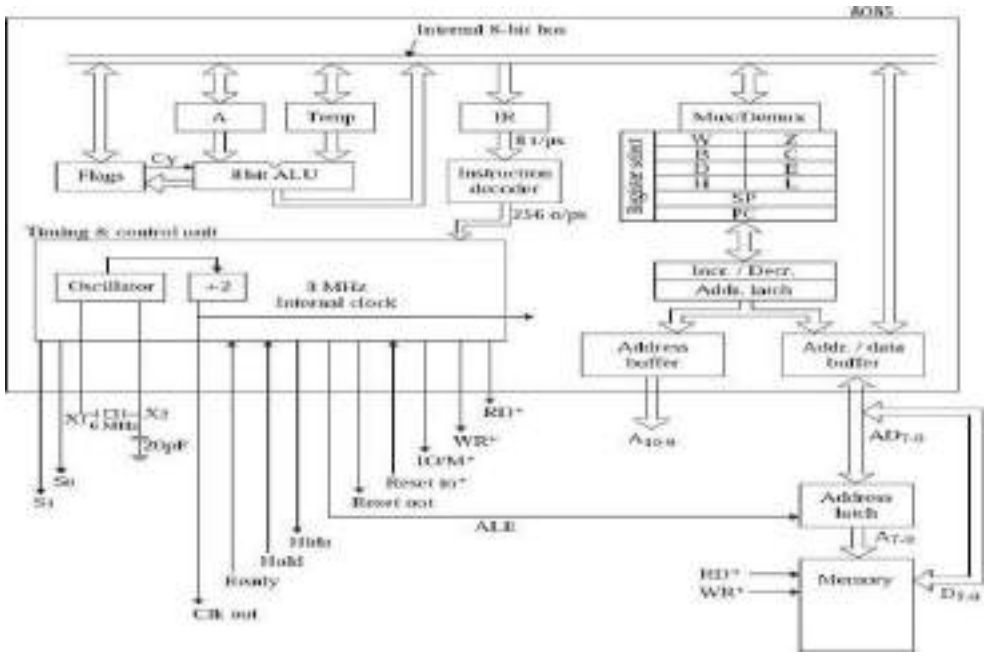
- ItstandsforGeneralpurposeregisters.
- Intheseregistersdatacanbeaccesseddirectlywithoutrequiringanyintermediate.
- ExamplesofGPRareB,C,D,E,H,andL.
- Theseregistersareof8-bit.
- Inordertohold16bitdata,two8bitregistercanbecombinedortheycanworkinpairssuchasB-C,D-EandH-L.These pairsareknownasregisterpairs.
- TheH-Lpairworksasamemorypointer.
- Amemorypointerholdstheaddressofaparticularmemorylocation.

SPR-

- SPRstandsforspecialpurposeregister.
- Inspecialpurposeregisterdatacannotaccesseddirectlyandrequiresanintermediate.
- ExamplesofSPRareAccumulator,programcounter,stackpointer.
- Theseregistersareusedonlybymicroprocessornotbyusers.

Timingandcontrolunit:

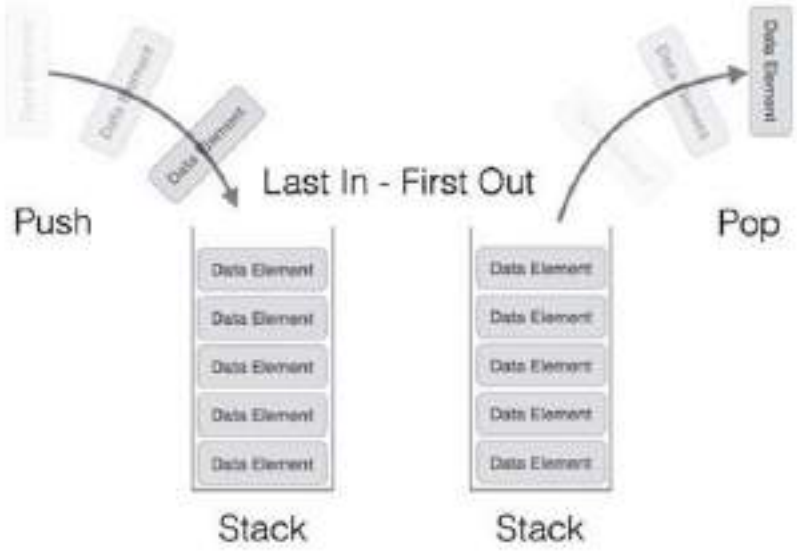
- WeuseTimingandcontrollingunitin8085forthegenerationoftimingsignalsandthesignalstocontrol.
- Alltheoperationsandfunctionsbothinteriorandexteriorofamicroprocessorarecontrolledbythisunit.
- X2andCLKoutputpins:Todoorratherperformtheoperationsoftiminginthemicrocomputersystem,wehaveagenerator calledclockgeneratorintheCUof8085.
- Other than the quartz crystal the complete circuit of the oscillator is within the chip. The twopins namely X1 and X2 are taken out from the chip to give the connection to the crystalexternally.
- We connect a capacitor of 20pF between the terminal X2 and ground just to analyze if thecrystalisgettingstarted.
- The frequency of the crystal is divided by 2 which divide the counter of the unit of control by2.
- Internally 8085A works with a frequency of 3 MHz internally with clock frequency. Hence acrystaloffrequencyof6-MHzcrystalgetsconnected betweenX1 andX2.
- Everyoperationintheentire8085systemoccurswiththegivensynchronizationprocesswiththeclock.TherearePeripheralchipslike8251USART,whichdoesnotoperateuntilasmallclocksignalisinneed.



STACK,STACKPOINTERANDSTACKTOP:

STACK:

- ThestackisaLIFO(lastin,firstout)datastructureimplementedintheRAMareaandisusedtostorea ddressesanddatawhenthemicroprocessor branchestoasubroutine.
- Thenthereturnaddressusedtogetpushedonthisstack.Alsotoswapvaluesoftworegistersandregi ster pairswe use the stackaswell.



STACKPOINTER:

- Itisaspecialpurpose16-bitregister thatstorestheaddressofthe“**topofstack**”**.**

- "8085" providesthe **"stackpointer"** whichgivestheaddressofthe"topofstack". So,wheneveryouwanttostoreanitemitstacks,youjuststoreitattheaddressprovidedbythestack pointer.

STACKoperationin8085microprocessor.

ThestackisareservedareaofthememoryinRAMwheretemporaryinformationmaybestored.An8-bitstackpointerisusedtoholdtheaddressofthemostrecentstackentry.
Thislocationwhichhasthemostrecententryiscalledasthetopofthestack.
When the information is written on the stack, the operation is called PUSH. When theinformationisreadfromthestack,theoperationiscalledPOP.Thestackworksontheprinciple of Last inFirst Out.

8085interrupts:

- Interruptisaprocesswhereanexternaldevicecangettheattentionofthemicroprocessor**.**
- Aninterruptisconsideredtobeanemergencysignalthatmaybeserviced.
- TheMicroprocessormayrespondtoitassoonaspossible.
- TheprocessstartsfromtheI/Odevice
- Theprocessisasynchronous

ClassificationofInterrupts:

Interruptscanbeclassifiedintotwotypes:

- **MaskableInterrupts** (CanbedelayedorRejected)
- **Non-MaskableInterrupts**(CannotbedelayedorRejected)

Interruptscanalsobeclassifiedinto:

- **Vectored**(theaddressoftheserviceroutineishard-wired)
- **Non-vectored**(theaddressoftheserviceroutineneedstobesuppliedexternallybythedevice)

WhathappenswhenMPisinterrupted?

- When the Microprocessor receives an interrupt signal, it suspends the currently executingprogram and jumps to an Interrupt Service Routine (ISR) to respond to the incominginterrupt.
- Eachinterruptwillmostprobablyhaveitsown ISR.
- Responding to an interrupt may be immediate or delayed depending on whether theinterruptismaskableornon-maskableandwhetherinterruptsarebeingmaskedornot.

- There are two ways of redirecting the execution to the ISR depending on whether theinterruptisvectoredor non-vectored.
- **Vectored:**TheaddressofthesubroutineisalreadyknowntotheMicroprocessor.

- **<u>NonVectored</u>:**Thedevicewillhavetosupplytheaddressofthesubroutinetothe Microprocessor.

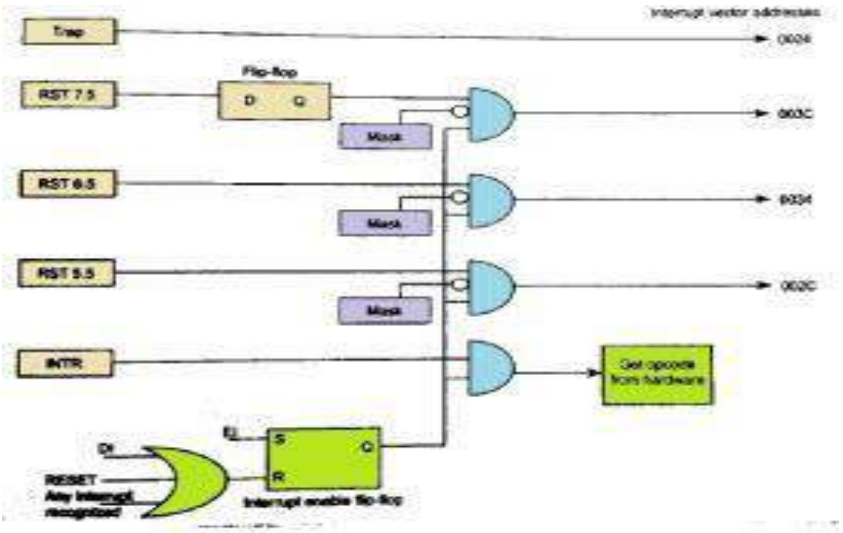
- When a device interrupts, it actually wants the MP to give a service which isequivalent to asking the MP to call a subroutine. This subroutine is called **ISR**(InterruptService Routine)
- The'EI'instructionisaonebyteinstructionandisusedtoenablethenon-maskableinterrupts.
- The'DI'instructionisaonebyteinstructionandisusedtodisablethenon-maskableinterrupts.
- The8085hasasingleNon-Maskableinterrupt.Thenon-maskableinterruptisnotaffectedbythevalue of theInterruptEnable flipflop.

The8085has5interrupt inputs.

- The**INTR**inputistheonlynon-vectoredinterrupt.INTRismaskableusingtheEI/DIinstructionpair.
- **RST5.5, RST6.5,RST7.5**areallautomaticallyvectoredandaremaskable.
- **TRAP**istheonlynon-maskableinterruptinthe8085.itisalsoautomaticallyvectored.

MaskingofinterruptSIM,RIM:

- Whenwestudyinterruptsin8085microprocessorthenweshouldknowMaskingofInterruptsin8085 microprocessor.
- In8085microprocessormaskingofinterruptcanbedoneforfourhardwareinterruptsINTR,RST5.5, RST 6.5, andRST7.5.
- Themaskingof8085interruptsisdoneatdifferentlevels.Inbellowfigureshowstheorganization ofhardwareinterrupts inthe8085microprocessor.



- Themaskableinterruptsarebydefaultmaskedbytheresetsignal.Sonointerruptisrecognized bythehardwarereset.
- TheinterruptscanbeenabledbytheEIinstruction.
- ThethreeRSTinterruptscanbeselectivelymaskedbyloadingtheappropriatewordintheaccumulatorandexecutingSIMinstruction.Thisiscalled software masking.
- Allmaskableinterruptsaredisabledwheneveraninterruptisrecognized.
- Allmaskableinterruptscanbedisabled byexecutingtheDIinstruction.
- If we talk about RST 7.5 interrupt. It alone has a flip-flop to recognize edge transition. The DIinstruction reset interrupt enable flip-flop in the processor and the interrupts are disabled.Toenable interrupts, EI instructionhastobeexecuted.

SIMInstruction:

TheSIMinstructionisusedtomaskorunmaskRSThardwareinterrupts.Whenexecuted,theSIM instruction reads the content of accumulator and accordingly mask or unmask theinterrupts. The format of control word to be stored in the accumulator before executing SIMinstructionisasshown inFig.

Bit position	D7	D6	D5	D4	D3	D2	D1	D0
Name	SOD	SDE	X	R7.5	MSE	M7.5	M6.5	M5.5
Explanation	Serial data to be sent	Serial data enable—set to 1 for sending	Not used	Reset RST 7.5 flip-flop	Mask set enable—Set to 1 to mask interrupts	Set to 1 to mask RST 7.5	Set to 1 to mask RST 6.5	Set to 1 to mask RST 5.5

- Inadditiontomaskinginterrupts,**SIM**instructioncanbeusedtosendserialdataonthe**SOD** lineoftheprocessor.
- ThedatatobesendisplacedintheMSBbitoftheaccumulatorandtheserialdataoutputisenabled bymakingD6 bitto1.

RIMInstruction:

- RIMinstructionisusedtoreadthestatusoftheinterruptmaskbits.
- When**RIM**instructionisexecuted,theaccumulatorisloadedwiththecurrentstatusoftheinterrupt masksand thependinginterrupts.
- TheformatandthemeaningofthedatastoredintheaccumulatorafterexecutionofRIMinstructionisshownin Fig.

Bit position	D7	D6	D5	D4	D3	D2	D1	D0
Name	SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
Explanation	Serial input data in the SID pin	Set to 1 if RST 7.5 is pending	Set to 1 if RST 6.5 is pending	Set to 1 if RST 5.5 is pending	Set to 1 if interrupts are enabled	Set to 1 if RST 7.5 is masked	Set to 1 if RST 6.5 is masked	Set to 1 if RST 5.5 is masked

- Inaddition**RIM**instructionisalsousedtoreadtheserialdataonthe**SID**pinoftheprocessor.
- Thedataonthe**SID**pinisstoredintheMSBoftheaccumulatoraftertheexecutionofthe**RIM** instruction.

- E.g. write an assembly language program to enables all the interrupts in 8085 after reset**.** **EI**Enable interrupts MVI A, 08H: Unmask the interrupts **SIM:** Set the mask and unmask usingSIMinstruction.

UNIT-2: INSTRUCTION SET AND ASSEMBLY LANGUAGEPROGRAMMING

INSTRUCTIONWORDSIZE:

- Thetotalmemorylocationrequiredtofeedtheinstructioninmemoryiscalledas**instructionword size**.
- Thememorylocationof8085microprocessorcanaccommodate8-bitsofdata.
- Tostore16-bitsdata, theyarestoredintwoconsecutivememorylocations(i.e.2Bytes).
- Accordingtotheinstructionwordsizein8085microprocessor,therearethreetypesofinstructions :

a. 1-Byteinstruction
b. 2-Byteinstruction
c. 3-Byteinstruction

1 –ByteInstructions:

- Theyincludeopcodeandoperandsinthesamebyte.
- Operandsare internalregistersandcodedintotheinstruction.
- Instructionsrequireonememorylocationtostorethesinglebyte inthememory.

Note:
Instructionshavingtheonlyregisterorregisterpairastheoperandis1–ByteInstructions.Instructionsintheabsenceof operandarealso1–Byte Instructions.
Examples:

MOVB,C

LDAX

BNOPHL

T

2 –ByteInstructions:

- 1st bytespecifiesopcodeand2nd bytespecifiesoperand.
- Instructionsrequiretwomemorylocationstostoreinthememory.

Note:
Instructions having the8-bitnumbereitherasanaddress ordata astheoperandis2–ByteInstructions.
Examples:

MVIB,26H

IN 56H

3 –ByteInstructions:

- Ina3-byteinstruction,thefirstbytespecifiestheopcode,andthefollowingtwobytesspecifythe16-bit address.
- The2nd byteholdsthelowerorderaddress.
- The3rd-byteholdsthehigherorderaddress.
- Instructionsrequirethreememorylocationstostorethesinglebyteinthememory.

Note:
Instructionshavingthe16-bitnumbereitherasanaddressordataastheoperandis3–ByteInstructions.
Examples:

LDA2050H

JMP2085H

ADDRESSINGMODES:

- Thevariouswaysofspecifyingdata(oroperands)forinstructionsarecalledas**addressingmodes.**
- The8085addressingmodesareclassifiedintofollowingtypes:

1. Immediateaddressingmode
2. Directaddressingmode
3. Registeraddressingmode
4. Registerindirectaddressingmode
5. Implicitaddressingmode

1. DirectAddressingmode:

- Inthisaddressingmodetheaddressoftheoperandisspecifiedintheinstructionitself.

or

- Themodeofaddressinginwhichthe16-bitaddressoftheoperandisdirectlyavailableintheinstructionitselfiscalledDirectAddressingmode.i.e.,theaddressoftheoperandisavailableinthe instructionitself. Thisisa 3-byte instruction.

Example:
LDA 9525H→Load the contents of memory location into

Accumulator.STA 8000H→Store the contents of the Accumulator in the

location 8000HIN01H→Readthedata fromportwhoseaddressis01H

2. Registeraddressingmodes:

- Inthisaddressingmodetheaddressoftheoperandisoneofthegeneralpurposeregister.

or

- Inthismodetheoperandsaremicroprocessorregistersonlyi.e.theoperationisperformedwithinvariousregistersof the microprocessor.

Example:

- MOVA,B→MovethecontentsofBregistertoAregister.
- SUBD→SubtractthecontentsofDregisterfromAccumulator.
- ADDB,C→AddthecontentsofCregistertothecontentsofBregister.

3.Registerindirectaddressingmodes:

- Inthisaddressingmodetheaddressoftheoperandisspecifiedbyaregisterpair.

or

- The 16-bit address location of the operand stored in a register pair (H-L) is given in theinstruction.Theaddressoftheoperandisgiveninanindirectwaywiththehelpofaregisterpair. Soitiscalled Register indirectaddressingmode.

Example:

- LXIH9570H→LoadimmediatetheH-Lpairwiththeaddressofthelocation 9570H
- MOVA,M→Movethe contents of the memory location pointed by the H-L pairtoaccumulator

4. ImmediateAddressingmode:

- Inthisaddressingmodetheoperandisspecifiedintheinstructionitself.

or

- In this mode operand is a part of the instruction itself is known as Immediate Addressingmode. If the immediate data is 8-bit, the instruction will be of two bytes. If the immediatedatais16 bit,the instructionisof 3 bytes.

Example:

ADI DATA →Add immediate the data to the contents of the accumulator.LXIH 8500H→Load immediate the H-L pair with the operand 8500HMVI 08H→Move the data 08 H immediately to the accumulatorSUI05H→Subtractimmediatelythe data05Hfromthe accumulator

5. ImplicitAddressingmode:

- Inthisaddressingmodetheinstructiondon'trequiretheaddressoftheoperand.

or

- Themodeofinstructionwhichdonotspecifytheoperandintheinstructionbutitisimplicated,isknowasimplicitaddressingmode.i.e.,theoperandissupposedtobepresentgenerallyinaccumulator.

Example:
CMA→complementthecontentsofAccumulatorCMC→C

omplement carry

RLC→Rotate Accumulator left by one

bitRRC→RotateAccumulatorrightbyonebitST

C→Set carry.

INSTRUCTIONSETOF8085:

- Aninstructionisabinarybitpatternwhichperformsaspecificfunctioninasystem.Theentiregroup of instructionsof a systemiscalledthe instructionset.

- Instruction set determines what functions the microprocessor can perform with a singleinstruction.

- Theinstructionsetinmicroprocessor8085canbeclassifiedintofivefunctionalcategories:

OR

- Aninstructionisacommandtothemicroprocessortoperformagiventaskonaspecifieddata.
- Eachinstructionhastwoparts:oneistasktobeperformed,calledtheoperationcode(opcode),andthesecondisthe datatobeoperatedon,called theoperand.
- The operand (or data) can be specified in various ways. It may include 8-bit (or 16-bit) data,aninternalregister,amemorylocation,or8-bit(or16-bit)address.Insomeinstructions,theoperandisimplicit.

1. Datatransfer(copy)operations
2. Arithmeticoperations
3. Logicaloperations

4. Branchingoperationsand
5. Machine-controloperations.

1.DATATRANSFERINSTRUCTION:

- Theseinstructionsmovedatabetweenregisters,orbetweenmemoryandregisters.
- Thisgroupofinstructionscopiesdatafromalocationcalledassourcetoanotherlocationcalledasdestination,withoutmodifyingthecontentsofthe source
- Theseinstructionsarenotthedatatransferinstructionsbutdatacopyinstructionbecausethesource isnot modified.

Opcode Operand **Copy from to sourcedestination**	**Description**
MOV Rd,Rs	Thisinstructioncopiesthecontentsofthesource
M,Rs	register the destination register; the into contentsof
	The source register are not altered. If one of theoperands is a memory location, its location isspecifiedbythecontentsoftheHLregisters. Example:MOVB,CorMOVB,M

Rd,M Moveimmediate8-bit

MVI Rd,data	The8-bitdataisstoredinthedestinationregisteror
M,data	Memory.If the operand is a memory location, itslocationisspecifiedbythecontentsoftheHLregisters.Example:MVIB,57HorMVIM, 57H

Loadaccumulator

LDA 16-bitaddress	The contents of a memory location, specified by a16-bit address in the operand, are copied to theaccumulator. Thecontentsofthesourcearenotaltered.Example:LDA2034H

Loadaccumulatorindirect

LDAX B/DReg.pair	The contents of the designated register pair point to amemory location. This instruction copies the contents ofthatmemorylocationintotheaccumulator.Thecontents ofeithertheregisterpairorthememorylocationare not altered. Example:LDAXB

Loadregisterpairimmediate

LXIReg.pair,16-bitdata	The instruction loads 16-bit data in the register pairdesignatedinthe operand. Example:LXIH,2034H

LoadHandLregistersdirect	
LHLD 16-bitaddress	The instruction copies the contents of the memorylocationpointedoutbythe16-bitaddressintoregisterLandcopiesthecontentsofthe nextmemory location into register H.The contents ofsourcememorylocationsarenotaltered. Example:LHLD2040H
Storeaccumulatordirect STA 16-bitaddress	The contents of the accumulator are copied into thememorylocationspecifiedbytheoperand.Thisisa3-byteinstruction,thesecondbytespecifiesthelow-order address and the third byte specifies thehigh-orderaddress.Example:STA4350H
Store accumulator indirect STAX Reg.pair	The contents of the accumulator are copied into thememory location specified by the contents of theoperand(registerpair).Thecontentsoftheaccumulator are notaltered. Example:STAXB
StoreHandLregistersdirect SHLD 16-bitaddress	ThecontentsofregisterLarestoredintothememory location specified by the 16-bit address intheoperandandthecontentsofHregisterarestoredintothenextmemorylocationbyincrementingtheoperand.Thecontentsofregisters

	HL are not altered.This is a 3-byte instruction, thesecondbytespecifiesthelow-orderaddressandthethirdbytespecifiesthe high-orderaddress. Example:SHLD2470H
ExchangeHandLwithDand E XCHG none	ThecontentsofregisterHareexchangedwiththecontents of register D, and the contents of register Lare exchanged with the contents of register E.Example:XCHG

ArithmeticOperations:

Theyperformarithmeticoperations,suchas,addition,subtraction,increment,anddecrement.

Addition:

- Additionofany8-bitnumber,orthecontentsofaregisterorthecontentsofamemorylocationisaddedtothecontents oftheaccumulatorandthesumisstoredintheaccumulator.
- Notwoother8-bitregisterscanbeadded directly.
- ForexamplethecontentsofregisterBcannotbeaddeddirectlytothecontentsoftheregister C.8085canalsoperform16-bit.ItcanalsoperformBCDaddition.

Subtraction:

- Subtraction of any 8-bit number, or the contents of a register, or the contents of a memorylocationcanbesubtractedfromthecontentsoftheaccumulatorandtheresultsstoredint heaccumulator.
- The subtraction is performed in 2's compliment, and if the results is negative. Then they areexpressedin2'scomplement.
- Notwootherregisterscanbesubtracteddirectly.8085donotperform16-bitsubtraction.

IncrementorDecrement:

- The 8-bit contents of any register or a memory location can be incremented or decrement by1.
- Similarly,the16-bitcontentsofaregisterpaircanbeincrementedordecrementby1.
- Theseincrementanddecrementoperationscanbeperformeddirectlyinthesourceitself.Itmeans without usingaccumulator.

Opcode	Operand	Meaning	Explanation
ADD	R M	Addregisteror memory, to theaccumulator	The contents of the registeror memory are added to thecontents of the accumulatorandtheresultisst oredintheaccumulator. **Example**–ADDR,ADDM
ADC	R M	Addregistertothea ccumulator withcarry	The contents of the registeror memory & M the Carry flagare added to the contents oftheaccumulatorandtheresul tisstoredintheaccumulator. **Example**–ADCR,ADDM
ADI	8-bitdata	Add the immediate totheaccumulator	The8-bitdataisaddedtothecontentso ftheaccumulatorand the result is stored in theaccumulator. **Example**–ADI55
ACI	8-bitdata	Add the immediate tothe accumulator withcarry	The 8-bit data and the Carryflag are added to the contentsoftheaccumulatorand theresultisstoredintheaccumu lator. **Example**–ACI55
LXI	Reg. pair, 16bitdata	Loadtheregisterpairi mmediate	Theinstructionstores16-bitdataintotheregisterpairdes ignatedintheoperand. **Example**–LXIH,3025H

DAD	Reg.pair	Add the register pairtoHandLregisters	The16-bitdataofthespecifiedregister pairareadded to the contents of theHLregister. **Example**–DAD
SUB	R M	Subtracttheregister or the memory fromtheaccumulator	The contents of the registeror the memory are subtractedfromthecontentsoft heaccumulator, and the result isstoredintheaccumulator. **Example**–SUBR,SUBM
SBB	R M	Subtract the sourceand borrow from theaccumulator	Thecontentsoftheregisterorth ememory&MtheBorrowflagare subtractedfromthecontentsofth eaccumulatorandtheresultispla cedintheaccumulator. **Example**–SBBR,SBBM
SUI	8-bitdata	Subtract the immediatefromtheaccu mulator	The8-bitdataissubtractedfromthecon tentsoftheaccumulator&theres ultisstoredintheaccumulator. **Example**–SUI55
SBI	8-bitdata	Subtract thei mmediatefromtheaccu mulator with borrow	The8-bitdataandborrowissubtracted from the contents ofthe accumulator & the result isstoredintheaccumulator
INR	R M	Increment the register or the memoryby1	Thecontentsofthedesignatedr egisterorthememory are incremented by 1and their result is stored at thesameplace.

			Example–INRR,INRM
INX	R	Increment register pairby1	Thecontentsofthedesignatedregisterpairareincrementedby1 andtheirresultisstoredatthesameplace. **Example**–INXR
DCR	R M	Decrement the register or the memoryby1	Thecontentsofthedesignated register or memoryare decremented by 1 and theirresultisstoredatthesameplace. **Example**–DCRR,DCRM
DCX	R	Decrement the registerpair by1	Thecontentsofthedesignatedregisterpairaredecrementedby1 andtheirresultisstoredatthesameplace. **Example**–DCXR
DAA	None	Decimal adjust accumulator	Thecontentsoftheaccumulatorarechangedfromabinary value to two 4-bit BCDdigits. Ifthevalueofthelow-order4-bitsintheaccumulatorisgreaterthan9orifACflagisset,theinstructionadds6tothelow-orderfourbits. If the value of the high-order4-bitsintheaccumulatorisgreaterthan9orif theCarryflagisset,theinstructionadds6tothehigh-orderfourbits. **Example**–DAA

LOGICALOPERATIONS:

Thesetypeinstructionsperformsvariouslogicaloperationswiththecontentsoftheaccumulator. 8085 canperform six logicaloperationwhich are:

- AND
- OR
- Exclusive-OR
- NOT
- Compare
- Rotate

A 8-bit number can be logically ANDed with the contents of the accumulator. It can also be acontent of register or of a memory location. The results are stored in the accumulator. Thecontentof the accumulator canbe complimented.

Rotate:

Eachbitoftheaccumulatorcanbeshiftedeitherleftorrighttothenextposition.

Compare:

- Any8-bitnumberorthecontentofaregister,orcontentofamemorylocationcanbecomparedforequality,greaterthan,orlessthan,withthecontentsoftheaccumulator.
- Theresultisreflectedbyzeroandcarryflags.

Opcode	Operand	Meaning	Explanation
CMP	R M	Compare theregister ormemory withtheaccumulator	The contents of the operand(register or memory) are Mcomparedwiththecontentsofthe accumulator.
CPI	8-bitdata	Compareimmediate withtheaccumulator	Thesecondbytedataiscomparedwith the contents of theaccumulator.
ANA	R M	Logical ANDregister ormemory withtheaccumulator	The contents of the accumulator arelogicallyANDwithMthecontentsof the register or memory, and theresultisplacedintheaccumulator.

ANI	8-bitdata	Logical ANDimmediate withtheaccumulator	The contents of the accumulator arelogically AND with the 8-bit data andthe result is placed in theaccumulator.
XRA	R M	Exclusive ORregister ormemory withtheaccumulator	ThecontentsoftheaccumulatorareExclusive OR with M the contents ofthe register or memory, and theresultisplacedintheaccumulator.
XRI	8-bitdata	Exclusive ORimmediate withtheaccumulator	The contents of the accumulator areExclusive OR with the 8-bit data andthe result is placed in theaccumulator.
ORA	R M	Logical ORregister ormemory withtheaccumulator	The contents of the accumulator arelogically OR with M the contents oftheregisterormemory,andresultis placedinthe accumulator.
ORI	8-bitdata	Logical ORimmediate withtheaccumulator	Thecontentsoftheaccumulatorarelogically OR with the 8-bit data andthe result is placed in theaccumulator.
RLC	None	Rotate theaccumulatorleft	Eachbinarybitoftheaccumulatorisrotated left by one position. Bit D7 isplaced in the position of D0 as wellas in the Carry flag. CY is modifiedaccordingtobit D7.
RRC	None	Rotate theaccumulatorright	Each binary bit of the accumulator isrotated right by one position. Bit D0isplacedinthepositionof D7aswellas in the Carry flag. CY is modifiedaccordingtobit D0.
RAL	None	Rotate theaccumulatorleftthroughcarry	Eachbinarybitoftheaccumulatorisrotated left by one position throughthe Carry flag. Bit D7 is placed in theCarry flag, and the Carry flag isplaced in the least significantposition D0. CY is modifiedaccordingtobit D7.

RAR	None	Rotate theaccumulat orright throughcarry	Eachbinarybitoftheaccumulatorisrot ated right by one positionthrough the Carry flag. Bit D0 isplaced in the Carry flag, and theCarry flag is placed in the mostsignificant position D7. CY ismodifiedaccordingtobitD0.
CMA	None	Complement accumulator	The contents of the accumulator arecomplemented.Noflagsareaffecte d.
CMC	None	Complement carry	TheCarryflagiscomplemented.Noot herflagsareaffected.
STC	None	SetCarry	SetCarry

BRANCHINGOPERATIONS:

Thisgroupofinstructiontransfersthecontrolofmicroprocessorfromonelocationtoanotherloca tion.8085canperformfour typesofbranchingoperations.Theseare:

- JMP-Jumpwithinaprogram.
- CALL-Jumpfrommainprogramtosub-routine.
- RET-Jumpfrom sub-routinetomainprogram.
- RST-Jumpfrommainprogramtoinstructionsubroutine.

Jump:

- Conditionaljumpsaretheimportantaspectofthedecision-makingprocessintheprogrammingof a microprocessor.
- Theseinstructionstestsforacertainconditionsandaltertheprogramsequencewhentheconditio nismet.
- Forexamplezeroorcarryflag,Inaddition,theinstructionsetalsoincludesaninstructioncalledunc onditionaljump.

Call,return,andrestart:

- Thesetypeofinstructionschangesthesequenceofaprogrameitherby callingasub-routineor returningfromasub-routine.

- Theconditionalcallandreturninstructionscanalsotesttheconditionflags.

1. JumpInstructions:–

The jump instruction transfers the program sequence to the memory address given in theoperand based on the specified flag. Jump instructions are 2 types: Unconditional JumpInstructionsandConditionalJumpInstructions.

(a) UnconditionalJumpInstructions:

- Transferstheprogramsequencetothedescribedmemoryaddress.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
JMP	address	Jumpstotheaddress	JMP2050

(b) ConditionalJumpInstructions:

- Transferstheprogramsequencestothedescribedmemoryaddressonlyiftheconditioninsatisfied.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
JC	address	Jumpstotheaddressifcarryflagis1	JC2050
JNC	address	Jumpstotheaddressifcarryflagis0	JNC2050
JZ	address	Jumpstotheaddressifzeroflagis1	JZ2050
JNZ	address	Jumpstotheaddressifzeroflagis0	JNZ2050
JPE	address	Jumpstotheaddressifparityflagis1	JPE2050
JPO	address	Jumpstotheaddressifparityflagis0	JPO2050
JM	address	Jumpstotheaddressifsignflagis1	JM2050
JP	address	Jumpstotheaddressifsignflag0	JP2050

2. CallInstructions:–

The call instruction transfers the program sequence to the memory address given in theoperand. Before transferring, the address of the next instruction after CALL is pushed ontothe stack. Call instructions are 2 types: Unconditional Call Instructions and Conditional CallInstructions.

(a) UnconditionalCallInstructions:

- Ittransferstheprogramsequencetothememoryaddressgivenintheoperand.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
CALL	address	Unconditionallycalls	CALL2050

(b) ConditionalCallInstructions:

Onlyiftheconditionissatisfied,theinstructionsexecutes.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
CC	address	Callifcarryflagis1	CC2050
CNC	address	Callifcarryflagis0	CNC2050
CZ	address	Callsifzeroflagis1	CZ2050
CNZ	address	Callsifzeroflagis0	CNZ2050
CPE	address	Callsifparityflagis1	CPE2050
CPO	address	Callsifparityflagis0	CPO2050
CM	address	Callsifsignflagis1	CM2050
CP	address	Callsifsignflagis0	CP2050

3. ReturnInstructions:–

The return instruction transfers the program sequence from the subroutine to the callingprogram. Jump instructions are 2 types: Unconditional Jump Instructions and ConditionalJumpInstructions.

(a) UnconditionalReturnInstruction:

- Theprogramsequenceistransferredunconditionallyfromthesubroutinetothecallingprogram.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
RET	none	Return from the subroutine unconditionally	RET

(b) ConditionalReturnInstruction:

Theprogramsequenceistransferredunconditionallyfromthesubroutinetothecallingprogramonlyistheconditionissatisfied.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
RC	none	Returnfromthesubroutineifcarryflagis1	RC
RNC	none	Returnfromthesubroutineifcarryflagis0	RNC
RZ	none	Returnfromthesubroutineifzeroflagis1	RZ
RNZ	none	Returnfromthesubroutineifzeroflagis0	RNZ
RPE	none	Returnfromthesubroutineifparityflagis 1	RPE
RPO	none	Returnfromthesubroutineifparityflagis 0	RPO
RM	none	Returnsfromthesubroutineifsignflagis1	RM
RP	none	Returnsfromthesubroutineifsignflagis0	RP

STACK,I/O&MACHINE-CONTROLOPERATIONS:

Thesetypeofinstructionscontrolsthemachinefunctions,suchashalt,interrupt,ordonothing.

Opcode	Operand	Meaning	Explanation

NOP	None	Nooperation	Nooperationisperformed,i.e.,theinstructionisfetchedanddecoded.
HLT	None	Haltandenter wait state	TheCPUfinishesexecutingthecurrentinstructionandstopsfurtherexecution.An interruptorresetisnecessarytoexitfromthehaltstate.
DI	None	Disableinterrupts	The interrupt enable flip-flop is resetandalltheinterruptsaredisabledexcept TRAP.
EI	None	Enableinterrupts	Theinterruptenableflip-flopissetandalltheinterruptsareenabled.
RIM	None	Read interruptmask	Thisinstructionisusedtoreadthestatusofinterrupts7.5,6.5,5.5andreadserialdata inputbit.
SIM	None	Set interruptmask	This instruction is used to implementthe interrupts 7.5, 6.5, 5.5, and serialdataoutput.

Stackinstructionsareasfollows:
PUSH - Push Two bytes of Data onto the Stack**POP** - Pop Two Bytes of Data off the Stack**XTHL** - Exchange Top of Stack with H & L**SPHL**-MovecontentofH&LtoStackPointer

I/Oinstructionsareasfollows:
IN-InitiateInputOperation
OUT-InitiateOutputOperation

ASSEBLYLANGUAGEPROGRAMMINGOF8085:

WhatisAssemblyLanguageProgram?

- MachinelanguageandHexcodeinstructionsareverydifficultfortheprogrammer.
- Hence for programmer, the instructions of microprocessor are made in the form of Englishabbreviation (short form). These instructions are name as Assembly Language instructionsor mnemonics.
- The combinations of different mnemonics are known as Assembly Language Program and itisa low levellanguage.

Examplesofassemblylanguageprogram

LoadingRegisterorMemorywithData

Example1:Writea programtotransfer07HinregisterL.

Memory Address	MachineCode	Mnemonics	Operands	Comments
2000H	2E,07	MVI	L,07	Moveimmediate 07 inregisterL
2002H	76	HLT		Stoporterminatethep rogram

- TheinstructionMVIL,07willmovethedata07totheregisterL.
- Theinstructionwillstoptheprogram.
- ThemachinecodefortheinstructionMVIL,07is2E,07.
- The1stbyteofthemachinecodeis2EwhichistheHexcodefortheinstructionMVIL.
- Thesecondbyteisthedata07.Themachinecodefor HLTis76.
- The machine codes are fetch in the memory locations, starting from the memory locations2000H.
- Memory location 2000 H contains 2E, 2001 H contains 07 and memory location 2002 Hcontain 76, After the execution of a program, the contents of Register L can be examinedwhichare 07.

Memory Address	Machine Code	Mnemonics	Operands	Comments
2000H	3E,08	MVI	A,08	Get08inregisterA
2002H	4F	MOV	C,A	MovethecontentsofregisterAt oregister C
2003H	76	HLT		Halt

Example2WriteaprogramtoloadregisterAwith08HandthenmoveittoregisterC.

- InthisprogramtheinstructionMVIA,ONHwillplacethegivendata081Hintheregister A.

- TheHexcodefor MVIA,08His3E,08IHwhere3EistheHexcodefor MVIA.
- The instruction MOV C, A will move the contents of register A to the register C. Itsmachinecode is4F.
- With this instruction the data of register A is copies into the register C. It meansthe given data, is 08 H which was previously placed in register A is now copiedintothe register C.
- TheinstructionHLTwhosemachinecodeis76stopstheprogram.
- Thememorylocationsrequiredforthisprogramare2000Hto2003H.Anyothermemo ry locations can be selected. After the execution of a program, the contentsofregister C canbe examined.

Example3.Writeaprogramtoloadthecontentsofmemorylocation2050Hintoaccumulatora nd thenmovethis datainto registerB

Memory Address	MachineC ode	Mnemonics	Operands	Comments
2000H	3A,50,20	LDA	2050H	Loadthecontentsofmemorylocation 2050Hintotheaccumulator
2002H	47	MOV	B,A	MovethecontentsofregisterAtoregist erB
2004H	76	HLT		Stop

- TheinstructionLDA2050Hwillloadthecontentsofmemorylocation2050Hintotheaccumulator.
- Themachinecodefortheinstruction LDAis3A.
- TheinstructionMOVB.A(Machinecode47)willmovethecontentsofAccumulatortotheregisterB.
- Firstofalldata07isfetchinthememorylocation2050.
- Thenmemorylocations2000Hcontain3A,2001Hcontain50H,2002Hcontains20H,2003Hcontai ns47 H and2004 H contains76H.
- Afterexecutionofaprogram,thecontentsofregisterBcanbeexamined.

Example4.Writeaprogramtoaddtwo8-bitnumbers.

MEMORYAD DRESS	MACHINE CODE	MNEMONICS	OPERANDS	COMMENTS
2000	21,01,25	LXI	H,2501H	Getaddressoffirstnumberin H-Lpair.

2003	7E	MOV	A,M	1st numberinaccumulator.
2004	23	INX	H	IncrementcontentofH-Lpair.
2005	86	ADD	M	Add1stand2nd numbers.
2006	32,03,25	STA	2503H	Storesumin2503H.
2009	76	HLT		Stoptheprogram.

EXPLANATION:

- The1st numberwasstoredinthememorylocation2501H.
- 2501wasplacedinH-LpairbytheexecutionoftheinstructionLXIH,2501H.
- TheinstructionMOVA,Mmovedthecontentofthememorylocationaddressedby H-Lpairtotheaccumulator.
- Thusthe1st number49Hwhichwasinthe2501Hwasplacedintheaccumulator.
- TheINXHincreasedthecontentofH-Lpairfrom2501to2502H.
- TheinstructionADDMaddedthecontentofthememorylocationaddressedby H-Lpairwiththeaccumulator.
- Theresultgotstoredintheaccumulator.
  TheinstructionSTA2503Hstoredthesuminthememorylocation2503H.
- TheinstructionHLTendedtheprogram.

Example5.Writeaprogram tosubtracttwo8-bitnumbers.

MEMORY ADDRESS	MACHINE CODES	MNEMONICS	OPERAND	COMMENTS
2000	21,01,25	LXI	H,2501	Get addressof1stin H-Lpair.
2003	7E	MOV	A,M	1st numberinaccumulator.
2004	23	INX	H	ContentofH-Lpairincreasesfrom2501 to2502 H
2005	96	SUB	M	1st number-2nd number.
2006	23	INX	H	ContentofH-Lpairbecomes2503H.
2007	77	MOV	M,A	Storeresultin2503H.
2008	76	HLT		Stoptheprogram

EXPLANATION:

- Thefirstno.wasstoredinthememorylocation2501H.
- 2501HwasplacedinH-LpairbytheexecutionoftheinstructionLXIH,2501H.

- TheinstructionMOVA,Mmovedthecontentofthememorylocationaddressedby H-Lpairtothe accumulator.
- Thusthefirstno.49Hwhichwasinthe2501Hwasplacedintheaccumulator.
- TheINXHincreasedthecontentofH-Lpairfrom2501to2502H.
- TheinstructionSUBMsubtractedthecontentofthememorylocationaddressedbyH-Lpairfromthe accumulator.
- Thesecondno.whichwasinthememorylocation2502Hwassubtractedfromthefirstno.whichwas inthe accumulator.
- Theresultgotstoredintheaccumulator.
- TheINXHincreasedthecontentofH-Lpairfrom2502to2503H.
- TheinstructionMOVM,Amovedthecontentoftheaccumulatortothememorylocationaddressedb yH-Lpairtothe accumulator.
- Theresultwhichwasstoredintheaccumulatorgotstoredinthememorylocation2503H.
- TheinstructionHLTendedtheprogram.

Example6.Writeanassemblylanguageprogramin8085microprocessortoperformANDoperationbetweenlowerandhigherordernibbleof8bit number.

Assumption–
8bitnumberisstoredatmemorylocation2050.Finalresultisstoredatmemorylocation3050.

EXPLAINATION:

MEMORY ADDRESS	MNEMONICS	COMMENT
2000	LDA2050	A←M[2050]
2003	ANI0F	A←A(AND)0F
2005	MOVB,A	B←A
2006	LDA2050	A←M[2050]
2009	ANIF0	A←A(AND)F0
200B	RLC	Rotateaccumulatorleftbyonebitwithoutcarry
200C	RLC	Rotateaccumulatorleftbyonebitwithoutcarry
200D	RLC	Rotateaccumulatorleftbyonebitwithoutcarry
200E	RLC	Rotateaccumulatorleftbyonebitwithoutcarry

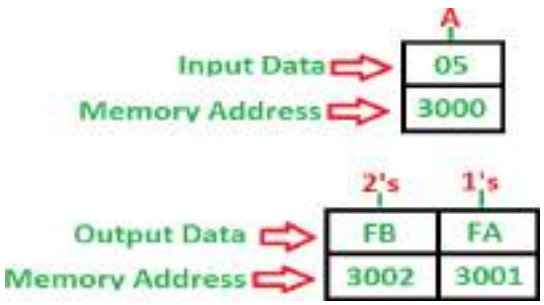
200F	ANAB	A←A(AND)B
2010	STA3050	M[3050] ←A
2013	HLT	END

EXPLANATION:

RegistersA,Bareusedforgeneralpurpose.

1. **LDA2050:** loadthecontentofmemorylocation2050inaccumulatorA.
2. **ANI0F:**perform ANDoperationinAand0F.StoretheresultinA.
3. **MOVB, A:**movesthecontentofAinregister B.
4. **LDA2050:** loadthecontentofmemorylocation2050inaccumulatorA.
5. **ANIF0:**perform ANDoperationinAandF0.StoretheresultinA.
6. **RLC:**rotatethecontentofAleftbyonebitwithoutcarry.Usethisinstruction4timestoreversethe content of A.
7. **ANAB:**performANDoperationinAandB. StoretheresultinA.
8. **STA3050:** storethecontentofAinmemorylocation3050.
9. **HLT:**stopsexecutingthe programandhaltsanyfurtherexecution.

Example 7- Write a program to find 1's and 2's complement of 8-bit number wherestarting address is 2000 and the number is stored at 3000 memory address and storeresultinto3001 and3002memoryaddress.



Program–

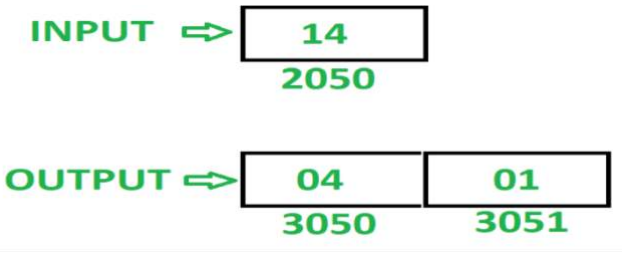
MEMORYADDRESS	MNEMONICS	OPERANDS	COMMENT
2000	LDA	[3000]	[A]←[3000]
2003	CMA		[A]←[A^]
2004	STA	[3001]	1'scomplement
2007	ADI	01	[A]←[A]+01
2009	STA	[3002]	2'scomplement
200C	HLT		Stop

EXPLANATION:

1. **A**isan8-bitaccumulator whichisusedtoloadandstorethedatadirectly

2. **LDA**isusedtoloadaccumulatordirectusing16-bitaddress(3Byteinstruction)
3. **CMA**isusedtocomplementcontentofaccumulator(1Byteinstruction)
4. **STA**isusedtostoreaccumulatordirectusing16-bitaddress(3Byteinstruction)
5. **ADI**isusedtoadddataintoaccumulatorimmediately(2Byteinstruction)
6. **HLT**isusedtohalttheprogram

Example8:– Write an assembly language program in 8085 microprocessor to showmaskingoflower and highernibbleof8bitnumber.
Example–



Assumption: - 8 bit number is stored at memory location 2050. After masking of nibbles,lower order nibble is stored at memory location 3050 and higher order nibble is stored atmemorylocation3051.

Program–

MEMORYADDRESS	MNEMONICS	COMMENT
2000	LDA2050	A←M[2050]
2003	MOVB,A	B←A
2004	ANI0F	A←A(AND)0F
2006	STA3050	M[3050] ←A
2009	MOVA,B	A←B
200A	ANI0F	A←A(AND)0F
200C	RLC	rotatecontentofAleftby1bit without carry
200D	RLC	rotatecontentofAleftby1 bitwithoutcarry
200E	RLC	rotatecontentofAleftby1bit without carry
200F	RLC	rotatecontentofAleftby1bit without carry
2010	STA3051	M[3051]←A

2013	HLT	END

EXPLANATION:

RegistersA,Bareused:

1. **LDA2050:** loadthecontentofmemorylocation2050inaccumulatorA.
2. **MOVB, A:**movesthecontentofAtoB.
3. **ANI0F:**perform ANDoperationofAwith0Fandstoretheresultbackto A.
4. **STA3050:** storecontentofAinmemorylocation3050.
5. **MOVA,B:**movesthecontentofBinA.
6. **ANI0F:**perform ANDoperationofAwith0Fandstoretheresultbackto A.
7. **RLC:**rotatecontentofAleftby1bitwithoutcarry.Usethisinstruction4timestoreversethe content of A.
8. **STA3051:** storethecontentofAinmemorylocation3051.
9. **HLT:**stopsexecutingtheprogramandhaltsanyfurtherexecution.

COUNTER:

- AcounterisdesignedsimplybyloadingappropriatenumberintooneoftheregistersandusingINRorDNRinstructions.
- Loopisestablishedtoupdatethecount.
- Eachcountischeckedtodeterminewhetherithasreachedfinalnumber;ifnot,theloopisrepeated.C

TIMEDELAY:

- Procedureusedtodesignaspecificdelay.
- Aregisterisloadedwithanumber,dependingonthetimedelayrequiredandtheregisterisdecrementeduntilitreacheszerobysettingupaloopwithconditionaljumpinstruction.

Using8-bitregisterascounter:

- Counter is another approach to generate a time delay. In this case the program size issmaller.So in this approach we can generate more time delay inlessspace. Thefollowingprogramwilldemonstratethe timedelayusing8-bit counter.

Program	Time(T-States)

• MVIB,FFH • LOOP:DCRB • JNZLOOP • RET	7 4 7/10 10

- Herethefirstinstructionwillbeexecutedonce,itwilltake7T-states.DCRCinstructiontakes4 T-states. This will be executed 255 (FF) times. The JNZ instruction takes 10 T-states whenitjumps(Itjumps254times),otherwiseitwilltake7T-States.AndtheRETinstructiontakes10T-States.
- 7+((4*255)+(10*254))+7+10=3584.Sothetimedelaywillbe3584*1/3μs=1194.66μs.So when we need some small delay, then we can use this technique with some other valuesinthe place of FF.
- This technique can also be done using some nested loops to get larger delays. The followingcodeisshowinghowwe cangetsomedelaywithoneloopintosome other

Using16-bitregister-pairascounter:

- Insteadofusing8-bitcounter,wecandothatkindoftaskusing16-bitregisterpair.Usingthismethod more time delay can be generated. This method can be used to get more than 0.5secondsdelay.Let ussee andexample.

Program	**Time(T-States)**
LXI B,FFFFHLOOP:DCXB **MOV A,BORA** **CJNZ LOOPRE T**	10 6 4 4 10(ForJump), 7(Skip) 10

- IntheabovetablewehaveplacedtheT-States.Fromthattable,ifwecalculatethetimedelay,itwillbe like this:
- 10 + (6 + 4 + 4 + 10) * 65535H – 3 + 10 = 17 + 24 * 65535H = 1572857. So the time delay willbe1572857 *1/3μs=0.52428s.Herewe aregettingnearly0.5sdelay.
- Indifferentprogram,weneed1sdelay.Forthatcase,thisprogramcanbeexecutedtwice.Wecancall theDelaysubroutinetwiceor useanotherouterloopfortwo-timeexecution.

Looping,countingandindexing(Call/JMP)

Toperformarepetitivetask,commonlyusedtechniquesarelooping,counting,andindexing.Toadd databytesstoredin memory,for example,thefollowingstepsarenecessary.

LOOPING

- Theprogrammingtechniqueusedtoinstructthemicroprocessortorepeattasksiscalledlooping.
- Thistaskisaccomplishedbyusingjumpinstructions.
- Definethetasktoberepeatediscalled**Looping**.
- Aloopissetupbyusing eitheraconditionalJumporanunconditionalJumpasillustratedinExamples.

COUNTING:

- Specifyhowmanytimesthetaskistoberepeatediscalled**Counting**.
- The counter is set by loading a count (number of times the task is to be repeated) .into aregisteroraregisterpair,andthecountingisdonebydecrementingthecounteverytimetheloop is repeated. The counter can also be set up to count from 0 to the final count usingincrementinstructions.

INDEXING:

- Specifythelocationofthedataiscalled**Indexing**.
- The starting location of the data can be specified by loading the memory address into aregisterpair andusingtheregister pairasamemorypointer orindex.

SETTINGFLAGS:

- Indicatetheendoftherepetitivetaskiscalled**SettingFlags**.
- The end of repetition is indicated by the flag of the conditional Jump instruction. When thecondition is true, the loop is repeated; when the condition is false, the loop execution isterminated,andthe executiongoestothe nextinstructioninmemory.

CLASSIFICATIONOF LOOPS:

1 Conditional
loop2.Unconditional
loop

CONTINUOUSLOOP:

- Repeatsataskcontinuously.
- Acontinuousloopissetupbyusingtheunconditionaljumpinstruction
- Aprogramwithacontinuousloop does not stoprepeatingthetasksuntilthesystemisreset.

CONDITIONALLOOP:

- Aconditionalloopissetupbyaconditionaljumpinstructions.
- Theseinstructionscheckflags(Z,CY,P,S)andrepeatthetasksiftheconditionsaresatisfied.
- Theseloopsincludecountingandindexing.

CONDITIONALLOOPANDCOUNTER:

- Acounterisatypicalapplicationoftheconditionalloop.
- Amicroprocessorneedsacounter,flagtoaccomplishtheloopingtask.
- Counterissetupbyloadinganappropriatecountinaregister.
- Countingisperformedbyeitherincrementordecrementthecounter.
- Loopissetupbyaconditionaljumpinstruction.
- Endofcountingisindicatedbyaflag.

Example:

- Stepstoaddtenbytesofdatastoredinmemorylocationsstartingatagivenlocationanddisplaythe sum.
- Themicroprocessorneeds

1. Acountertocount10databytes.
2. Anindexoramemorypointertolocatewheredatabytesarestored.
3. Totransferdatafromamemorylocationtothemicroprocessor(ALU)
4. Toperformaddition
5. Registersfortemporarystorageofpartialanswers
6. Aflagtoindicatethecompletionofthestack
7. Tostoreoroutputtheresult.

StackandSubroutinesprograms:

- Thestackisareservedareaofthememoryin RAMwherewecanstoretemporaryinformation.
- Interestingly,thestackisasharedresourceasitcanbesharedbythemicroprocessorandtheprogra mmer.
- Theprogrammercanusethestacktostoredata.Andthemicroprocessorusesthestacktoexecutesu broutines.
- The8085hasa16-bitregisterknownasthe‘StackPointer.’
- Thisregister’sfunctionistoholdthememoryaddressofthestack.Thiscontrolisgiventotheprogra mmer.

- Theprogrammercandecidethestarting address ofthestackby loadingtheaddress intothestackpointer registerat thebeginningofa program.
- ThestackworksontheprincipleofFirstinLastOut.Thememorylocationofthemostrecentdataentryonthe stack isknownasthe Stack Top.

Howdoesastackworkinassemblylanguage?

- Weusetwomaininstructionstocontrolthemovementofdataintoastackandfromastack.Thesetwoinstructionsare**PUSH** and **POP.**
- PUSH–Thisistheinstructionweusetowriteinformationonthestack.
- POP–Thisistheinstructionweusetoreadinformationfromthestack.
- Therearetwomethodstoadddatatothestack:**Directmethodandindirectmethod**

Directmethod:

Inthedirectmethod,thestackpointersaddressisloadedintothestackpointerregisterdirectly.

```
LXISP,8000H
LXI H,
1234HPUSHH
POP
DHLT
```

Explanationofthecode:

- LXI SP, 8000H – The address of the stack pointer is set to 8000H by loading the number intothestack pointer register.
- LXIH,1234H– Next,weaddanumbertotheHLpair.ThemostsignificanttwobitswillentertheHregister.Theleast significanttwobits willentertheLregister.
- PUSH H – The PUSH command will push the contents of the H register first to the stack. ThenthecontentsoftheLregisterwillbesenttothestack.Sothenewstacktopwillhold34H.
- POP D – The POP command will remove the contents of the stack and store them to the DEregister pair. The top of the stack clears first and enters the E register. The new top of thestack is 12H now. This one clears last and enters the D register. The contents of the DEregisterpair isnow 1234H.
- HLT–HLTindicatesthattheprogramexecutionneedstostop.

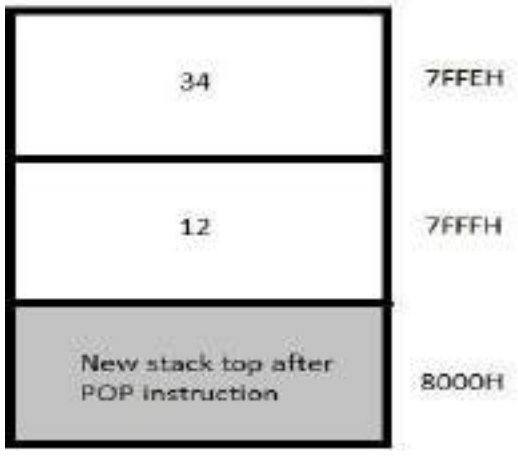
Indirectmethod:

Intheindirectmethod,thestackpointersaddressisloadedintothestackpointerregisterviaanotherr egister pair.

```
LXI H,
8000HSPHL
LXI H,
1234HPUSH
H
POP
```

Explanationofthecode

- LXI H, 8000H – The number that we wish to enter into the stack pointer, 8000H, is loadedintothe HL pair register.
- SPHL – This is a special command that we can use to transfer data from HL pair to stackpointer (SP).Now,thecontentsoftheHL pair areinthe SP.
- LXI H, 1234H – Next, we add a number to the HL pair. The most significant two bits will entertheHregister.Theleast significanttwobits willentertheLregister.
- PUSHH– ThePUSHcommandwillpushthecontentsoftheHregisterfirsttothestack.Thenthecontentsof theLregisterwillbesenttothestack.Sothenewstacktopwillhold34H.
- POP D – The POP command will remove the contents of the stack and store them to the DEregister pair. The top of the stack clears first and enters the E register. The new top of thestack is 12H now. This one clears last and enters the D register. The contents of the DEregisterpair isnow 1234H.
- HLT–HLTindicatesthattheprogramexecutionneedstostop.
- Boththemethodscanbeshowndiagrammaticallywiththefollowingdiagram.



WhatisaSubroutineisassemblylanguage?

- Asubroutineisasmallprogramwrittenseparatelyfromthemainprogramtoperformaparticular taskthatyoumayrepeatedlyrequireinthe mainprogram.
- Essentially,theconceptofasubroutineisthatitisusedtoavoidtherepetitionofsmallerprograms.
- Subroutinesarewrittenseparatelyandarestoredinamemorylocationthatisdifferentfromthe mainprogram.
- Callasubroutinemultipletimesfromthemainprogramusingasimple CALLinstruction.

BCDtobinaryconversionin8085:

(2200H)=67H
(2300H)=6xOAH+7 =3CH+ 7=43H

SourceProgram:

```
    LDA2200H
                        :GettheBCDnumberM
    OVB,A               :Save it
    ANIOFH               : Mask most significant four
    bitsMOVC,A
                        :SaveunpackedBCDIinCregisterM
    OVA,B               :GetBCDagain
    ANIFOH              :Maskleastsignificantfourbits
    RRC
                        :ConvertmostsignificantfourbitsintounpackedBCD2RR
    C
    RRC
    RRC
    MOVB,A
                         :SaveunpackedBCD2inBregisterXR
    AA                   :Clearaccumulator (sum=0)
    MVID,0AH
                        :SetDasamultiplierof10Sum:
ADD D                   :Add10 until(B)= 0
    DCRB                  :DecrementBCD2byone
    JNZSUM
                      :Ismultiplicationcomplete?iifnot,gobackandaddagainADDC
                      : Add BCD1
    STA2300H              :Storetheresult
    HLT                 :Terminateprogramexecution
```

BCDtoHEXconversionin8085Microprocessor:

Program

```
LXIH,5000
MOVA,M
            ;InitializememorypointerA
DDA         ;MSDX2
MOVB,A
            ;StoreMSDX2A
DDA         ;MSDX4
ADDA        ;MSD X
8ADDB       ;MSDX10
INXH        ;PointtoLSD
ADDM
             ;AddtoformHEXI
NXH
MOVM,A      ;Store the
resultHLT
```

Result

Input:

Data0:02Hinmemorylocation5000Da
ta1:09Hinmemorylocation5001

Output:

Data0:1DHinmemorylocation5002

ProgramtofindlargeroftwonumbersPROGRA

M:

MEMORY ADDRESS	MACHINE CODE	LABELS	MNEMONICS	OPERANDS	COMMENTS
2000	21,01,25		LXI	H,2501H	Addressof1st numberinH-Lpair.
2003	7E		MOV	A,M	1st number inaccumulator.
2004	23		INX	H	Addressof2nd numberinH-Lpair.
2005	BE		CMP	M	Compared 2nd numberwith1st number.Isthe 2nd number>1st?
2006	D2,0A,20		JNC	AHEAD	No,largernumberisinaccumulator. Go toAHEAD
2009	7E		MOV	A,M	Yes,get2nd numberinaccumulator.
200A	32,03,25	AHEAD	STA	2503H	storelargernumberin2

<u>Example-1:</u>

Data:
2501→98H
2502→87H
Result:
2503→98Handitisstoredinthememorylocation2503H.

ProgramtofindsmalleroftwonumbersPR
OGRAM:-

ADDRESS	MACHINECODES	LABELS	MNEMONICS	OPERANDS	COMMENTS
2000	21,01,25		LXI	H,2501H	Addressofthe1st number in H-Lpair
2003	7E		MOV	A,M	1st number inaccumulator
2004	23		INX	H	Address of the 2ndnumberinH-Lpair.
2005	BE		CMP	M	Compare 2nd numberwith1st .Is1stnumber<2nd number?
2006	DA,0A,20		JNC	AHEAD	Yes, smaller numberisinaccumulator.GotoAHEAD.
2009	7E		MOV	A,M	No ,get 2nd number in accumulator
200A	32,03,25	AHEAD	STA	2503H	Storesmallernumber in 2503H.
200D	76		HLT		stop

EXAMPLE:
DATA:250
1-84H
2502-
99H**RESU**
LT:2503-
84H

ProgramtofindthelargestnumberinadataarrayPROG

RAM:

MEMORY ADDRESS	MACHINE CODES	LABELS	MNEMONICS	OPERANDS	COMMENTS
2000	21,00,25		LXI	H,2500H	Addressforcountin H-Lpair.
2003	4E		MOV	C,M	CountinregisterC.
2004	23		INX	H	Addressofthe1st numberinH-Lpair.
2005	7E		MOV	A,M	1st number in accumulator.
2006	OD		DCR	C	Decrementcount.
2007	23		INX	H	Address of next number.
2008	BE		CMP	M	Compare next number with previous maximum.Isnextnumber>previousmaximum.
2009	D2,0D,20		JNC	AHEAD	NO, Larger numberisinaccumulator.GOtothelabel AHEAD.
200C	7E		MOV	A,M	Yes, get largernumber in accumulator.
200D	0D		DCR	C	DecrementCount.
200E	C2,07,20		JNZ	LOOP	
2011	32,04,25		STA	2504H	Store result in 2504H.
2014	76		HLT		StoptheProgram.

Example-
1:Data:250
0→03
2501→98
2502→75
2503→99
Result:2504→99

Programtofindthesmallestnumberinadataarray

PROGRAM:

MEMORY ADDRESS	MACHINE CODES	LABLES	MNEMONICS	OPERANDS	COMMENTS
2000	21,00,25		LXI	H,2500H	Gettheaddressforcountintheh-Lpair
2003	4E		MOV	C,M	CountinregisterC.
2004	23		INX	H	Getaddressof1st numberinH-Lpair.
2005	7E		MOV	A,M	1st number in accumulator.
2006	0D		DCR	C	Decrementcount.
2007	23	LOOP	INX	H	Address of nextnumberinH-Lpair.
2008	BE		CMP	M	Compare next number with previous smallest. Ispreviousssmallest< nextno?
2009	DA,0D,20		JC	AHEAD	Yes, smaller numberintheaccumulator .GotoAHEAD.
200C	7E		MOV	A,M	No,getnextnumberinaccumulator.
200D	0D	AHEAD	DCR	C	Decrementcount.
200E	C2,07,20		JNZ	LOOP	
2011	32,50,24		STA	2450H	Store smallest numberin2450H.
2014	76		HLT		Stoptheprogram.

- Amemoryaddressisauniqueidentifierusedbyadeviceor CPU fordatatracking.
- This binary address is defined by an ordered and finite sequence allowing the CPU to trackthelocationof each memorybyte.
- Modern computers are addressed by bytes which are assigned to memory addresses – binarynumbersassignedtoarandomaccessmemory(RAM)cellthatholdsuptoonebyte.Datagreaterthanonebyteisconsecutivelysegmentedintomultiplebyteswithaseriesofcorrespondingaddresses.

- HardwaredevicesandCPUstrackstoreddatabyaccessingmemoryaddressesviadatabuses.
- BeforeCPUprocessing,dataandprogramsmustbestoredinuniquememoryaddresslocations.

OR

MemoryAddress:

- The bus determines a fixed number of CPU memory addresses assigned according to CPUrequirements.TheCPU thenprocessesphysicalmemoryinindividualsegments.
- Theoperatingsystem'sread-onlymemory(ROM)basicinput/outputsystem(BIOS)programsanddevicedriversrequirememoryaddresses.Beforeprocessing,inputdevice/keyboard data, stored software or secondary storage must be copied to RAM withassignedmemoryaddresses.
- Memory addresses are usually allocated during the boot process. This initiates the startupBIOS on the ROM BIOS chip, which becomes the assigned address. To enable immediate videocapability, the first memory addresses are assigned to video ROM and RAM, followed by thefollowingassigned memoryaddresses:
    - ExpansioncardROMandRAMchips
    - Motherboarddualinlinememorymodules,singleinlinememorymodulesorRambusinlinememorymodules
    - Otherdevices

I/Oaddressing:

- Input/output(I/O)portaddressesareusedtocommunicatebetweendevicesandsoftware.
- TheI/Oportaddressisusedtosendandreceivedataforacomponent.
- AswithIRQs,eachcomponentwillhaveauniqueI/Oportassigned.
- Thereare65,535I/Oportsinacomputer,andtheyarereferencedbyahexadecimaladdressinthe range of 0000h toFFFFH.

UNIT-3:TIMINGDIAGRAMS

TimingDiagram:

- TimingDiagramisagraphicalrepresentation.Itrepresentstheexecutiontimetakenbyeachinstructioninagraphicalformat.The executiontimeisrepresentedinT-states.

InstructionCycle:

- Thetimerequiredtoexecuteaninstructioniscalledinstructioncycle.or
- Thetimetakenbytheprocessortocompletetheexecutionofaninstruction.Aninstructioncycleconsistsof onetosixmachinecycles.

MachineCycle:

- Thetimerequiredtoaccessthememoryorinput/outputdevicesiscalledmachinecycle.or

- Thetimerequiredtocompleteoneoperation;accessingeitherthememoryorI/Odevice.Amachinecycleconsistsof three tosixT-states.

T-State:

- Themachinecycleandinstructioncycletakesmultipleclockperiods.
- AportionofanoperationcarriedoutinonesystemclockperiodiscalledasT-state.or
- Timecorrespondingtooneclockperiod.Itisthebasicunittocalculateexecutionofinstructionsorprogramsinaprocessor.

Fetchcycle:

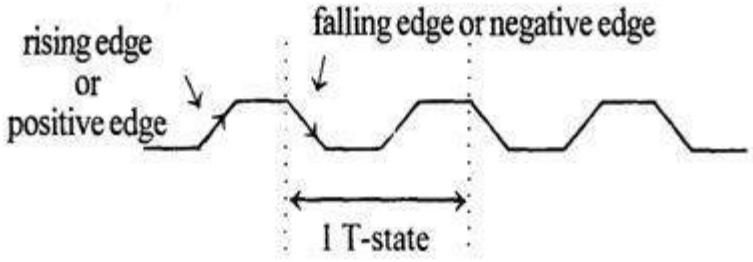
- The fetch cycle in a microprocessor comprises of several time states during which the nextinstructiontobeexecutediscopied(fetched)fromthememorylocation(whoseaddressisinthe Program Counter)tothe InstructionRegister.

MACHINECYCLESOF8085:

The8085microprocessorhas5(seven)basicmachinecycles.Theyare

1. Opcodefetchcycle(4T)
2. Memoryreadcycle(3 T)
3. Memorywritecycle(3T)
4. I/Oreadcycle(3T)
5. I/Owritecycle(3T)

Note : Time period, T = 1/f ; where f = Internal clock frequency



- Each instruction of the 8085 processor consists of one to five machine cycles, i.e., when the8085processorexecutesan instruction,it willexecutesomeofthemachinecyclesinaspecificorder.

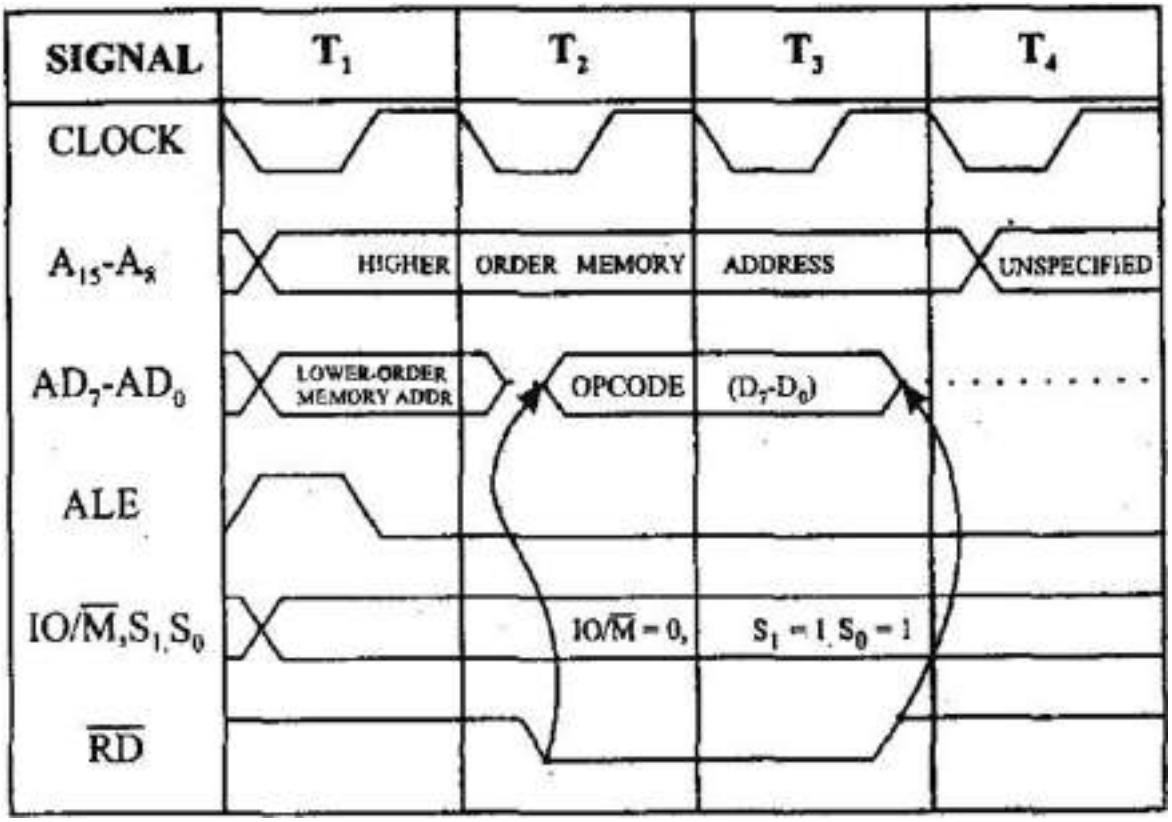
- Theprocessortakesadefinitetimetoexecutethemachinecycles.Thetimetakenbytheprocess or toexecute amachinecycleisexpressedinT-states.

- OneT-stateisequaltothetimeperiodoftheinternalclocksignaloftheprocessor.

- TheT-statestartsatthefallingedge ofaclock.

OpcodeFetchMachineCycle:

- ItisthefirststepintheexecutionofanyinstructionThetimingdiagramofthiscycleisgiveninFig.7.



- Thefollowingpointsexplainthevariousoperationsthattakeplaceandthesignalsthatarechanged duringtheexecutionofopcode fetchmachinecycle:

T1clockcycle:

- ThecontentofPCisplacedintheaddressbus;AD0-AD7linescontainslowerbitaddressandA8 – A15containshigherbit address.
- **IO/M'**signalislowindicatingthatamemorylocationisbeingaccessed.S1andS0alsochangedtoth e levels.
- ALEishigh,indicatesthatmultiplexedAD0–AD7actaslowerorderbus.

T2clockcycle:

- Multiplexedaddressbusisnowchangedtodatabus.
- The**(RD)'**signalismadelowbytheprocessor.Thissignalmakesthememorydeviceloadthedatabu swith thecontentsofthelocationaddressedbytheprocessor.

T3clockcycle:

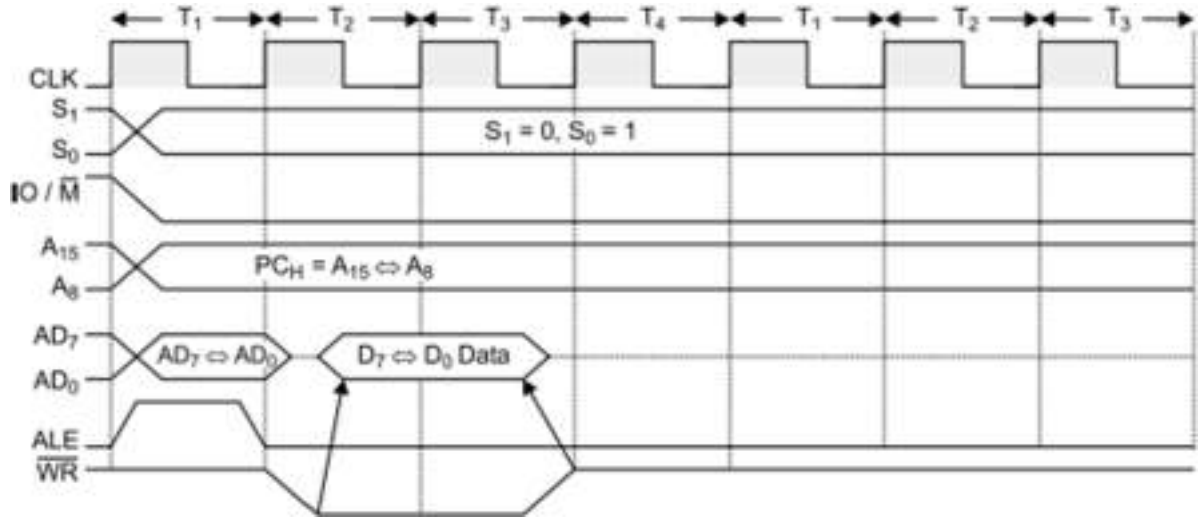
- Theopcodeavailableonthedatabusisreadbytheprocessorandmovedtotheinstructionregister.
- The**(RD)'**signalisdeactivatedbymakingitlogic1.

T4clockcycle:

- The processor decode the instruction in the instruction register and generate the necessarycontrol signals to execute the instruction. Based on the instruction further operations suchasfetching,writingintomemoryetc.takesplace.

<u>MemoryReadMachineCycle:</u>

- The memory read cycle is executed by the processor to read a data byte from memory. Themachinecycleisexactlysametoopcodefetchexcept:a)IthasthreeT-statesb)TheS0signalisset to0.



T1state:

- Thehigherorderaddressbus(A8-A15)andlowerorderaddressanddatamultiplexed(AD0-AD7)bus.
- ALEgoeshighsothatthememorylatchesthe(AD0-AD7)sothatcomplete16-bitaddressareavailable.
- Themicroprocessoridentifiesthememoryreadmachinecyclefromthestatussignals IO/M'=0,S1=1,S0=0.Thisconditionindicatesthememoryreadcycle.

T2state:

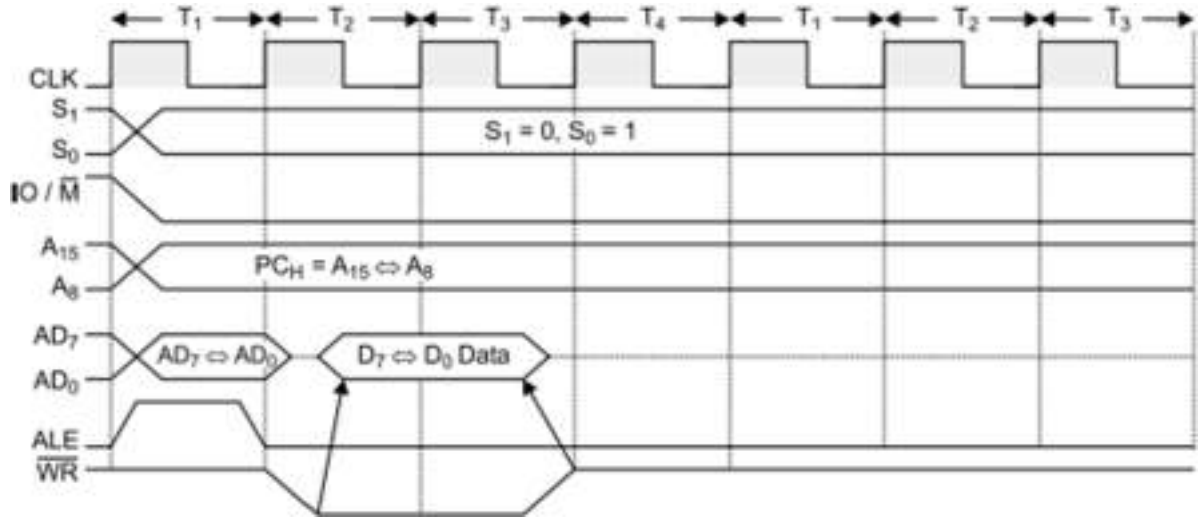
- Selectedmemorylocationisplacedonthe(D0-D7)oftheA/Dmultiplexedbus.RD'goes LOW

T3State:

- Thedatawhichwasloadedonthepreviousstateistransferredtothemicroprocessor.
- InthemiddleoftheT3stateRD'goeshighanddisablesthememoryreadoperation.
- Thedatawhichwasobtainedfromthememoryisthendecoded.

<u>MemoryWriteMachineCycle:</u>

- Thememorywritecycleisexecutedbytheprocessortowriteadatabyteinamemorylocation.The processortakesthreeT-statesand(**WR)'**signalismadelow.



T1state:

- Thehigherorderaddressbus(A8-A15)andlowerorderaddressanddatamultiplexed(AD0-AD7)bus.
- ALEgoeshighsothatthememorylatchesthe(AD0-AD7)sothatcomplete16-bitaddressareavailable.
- Themicroprocessoridentifiesthememoryreadmachinecyclefromthestatussignals IO/M'=0,S1=0,S0=1.Thisconditionindicatesthememoryreadcycle.

T2state:

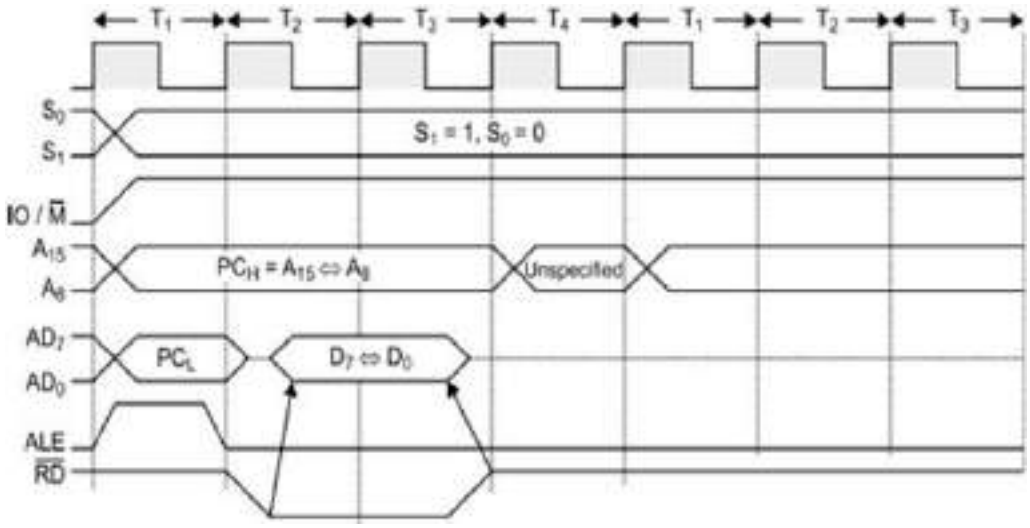
- Selectedmemorylocationisplacedonthe(D0-D7)oftheA/Dmultiplexedbus.WR'goes LOW

T3State:

- InthemiddleoftheT3stateWR'goeshighanddisablesthememorywriteoperation.The datawhichwasobtainedfromthememoryisthendecoded.

<u>I/OReadCycle:</u>

The I/O read cycle is executed by the processor to read a data byte from I/O port or fromperipheral,whichisI/Omappedinthesystem.The8-bitportaddressisplacedbothinthelowerandhigherorderaddressbus.TheprocessortakesthreeT-statestoexecutethismachinecycle.



T1state:

- Thehigherorderaddressbus(A8-A15)andlowerorderaddressanddatamultiplexed(AD0-AD7)bus.
- ALEgoeshighsothatthememorylatchesthe(AD0-AD7)sothatcomplete16-bitaddressareavailable.
- ThemicroprocessoridentifiestheI/Oreadmachinecyclefromthestatussignals IO/M'=1, S1=1,S0=0.ThisconditionindicatestheI/Oreadcycle.

T2state:

- Selectedmemorylocationisplacedonthe(D0-D7)oftheA/Dmultiplexedbus.RD'goes LOW

T3State:

- Thedatawhichwasloadedonthepreviousstateistransferredtothemicroprocessor.
- InthemiddleoftheT3stateRD'goeshighanddisablestheI/Oreadoperation.
- ThedatawhichwasobtainedfromtheI/Oisthendecoded.

I/OWriteCycle:

TheI/Owritecycleisexecutedbytheprocessortowriteadatabytetoa I/Oportortoaperipheral,which is I/O mapped in the system. The processor takes three T-states to execute this machinecycle.

T1state:

- The higher order address bus (A8-A15) and lower order address and data multiplexed(AD0-AD7)bus.
- ALE goes high so that the memory latches the (AD0-AD7) so that complete 16-bit addressareavailable.

- ThemicroprocessoridentifiestheI/Oreadmachinecyclefromthestatussignals IO/M'=1, S1=0,S0=1.ThisconditionindicatestheI/Oreadcycle.

T2state:

- Selectedmemorylocationisplacedonthe(D0-D7)oftheA/Dmultiplexedbus.WR'goes LOW

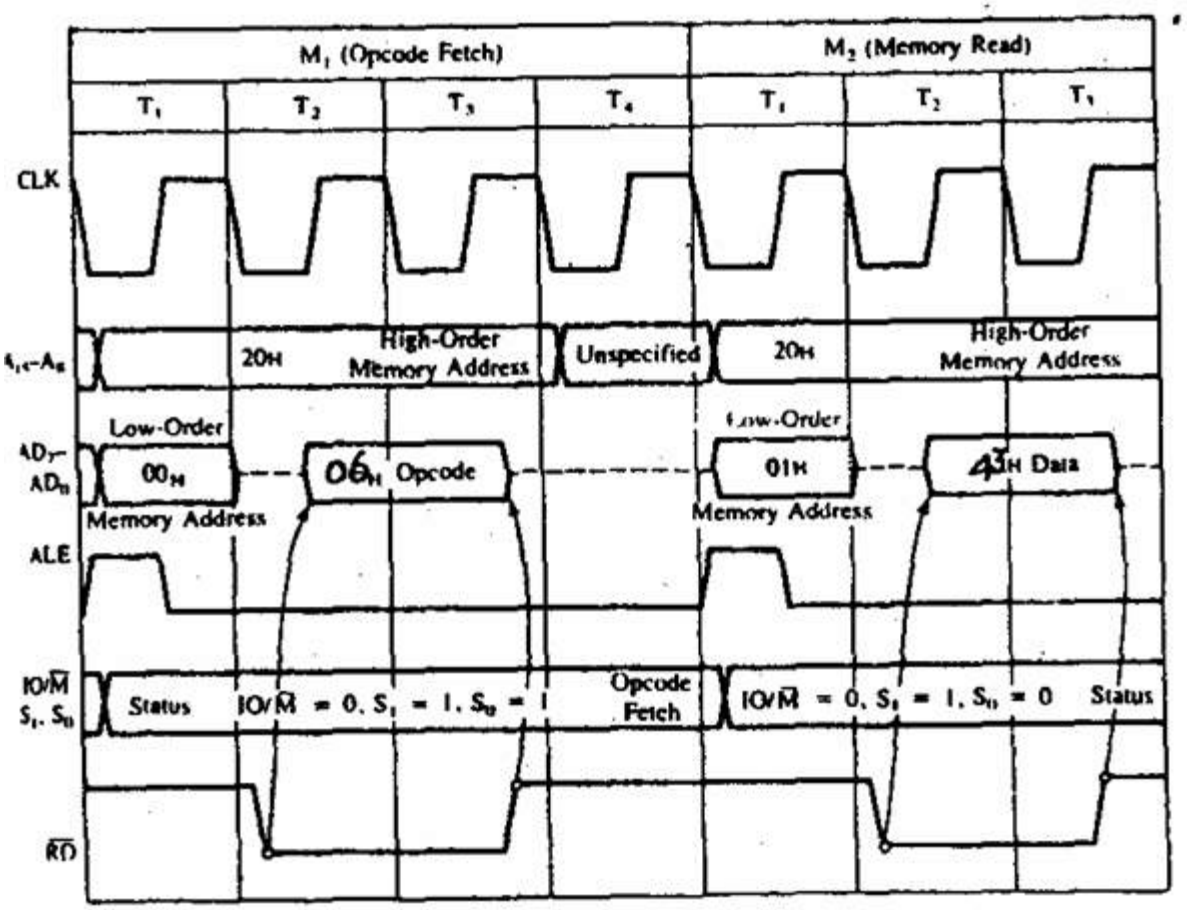
T3State:

- InthemiddleoftheT3stateWR'goeshighanddisablestheI/Owriteoperation.Thedata whichwasobtainedfromtheI/Oisthendecoded.

TimingdiagramforMVIB,43H.

- FetchingtheOpcode06Hfromthememory2000H.(OFmachinecycle)
- Read(move)thedata43Hfrommemory2001H.(memoryread)

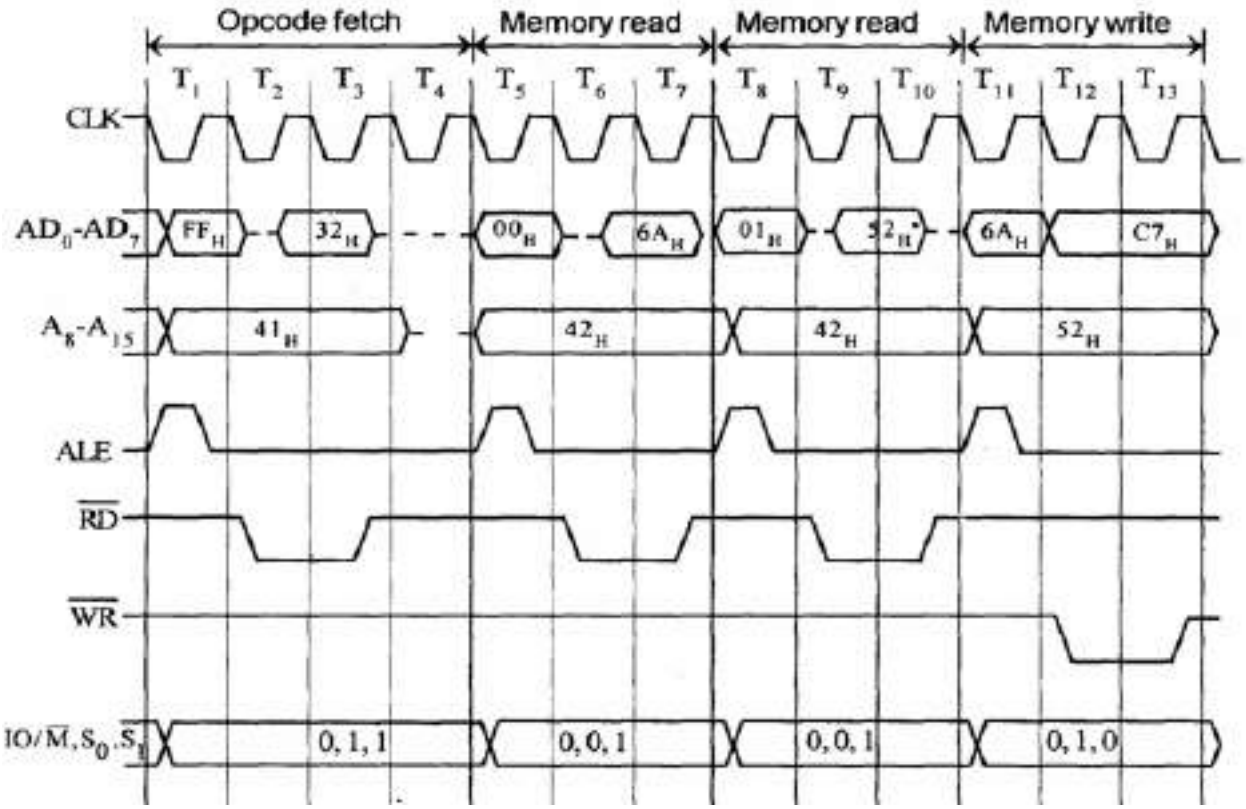
Address	Mnemonics	Opcode
2000	MVI B, 43_H	06_H
2001		43_H



TimingdiagramforSTA526AH.

- STAmeansStoreAccumulator-Thecontentsoftheaccumulatorisstoredinthespecifiedaddress(526A).
- TheopcodeoftheSTAinstructionissaidtobe32H.Itisfetchedfromthememory41FFH-OFmachinecycle
- Thenthelowerordermemoryaddressisread(6A).-MemoryReadMachineCycle
- Readthehigherordermemoryaddress(52).-MemoryReadMachineCycle
- Thecombinationofboththeaddressesareconsideredandthecontentfromaccumulatoriswritenin526A.-MemoryWrite MachineCycle
- AssumethememoryaddressfortheinstructionandletthecontentofaccumulatorisC7H.So,C7H from accumulatorisnowstoredin526A.

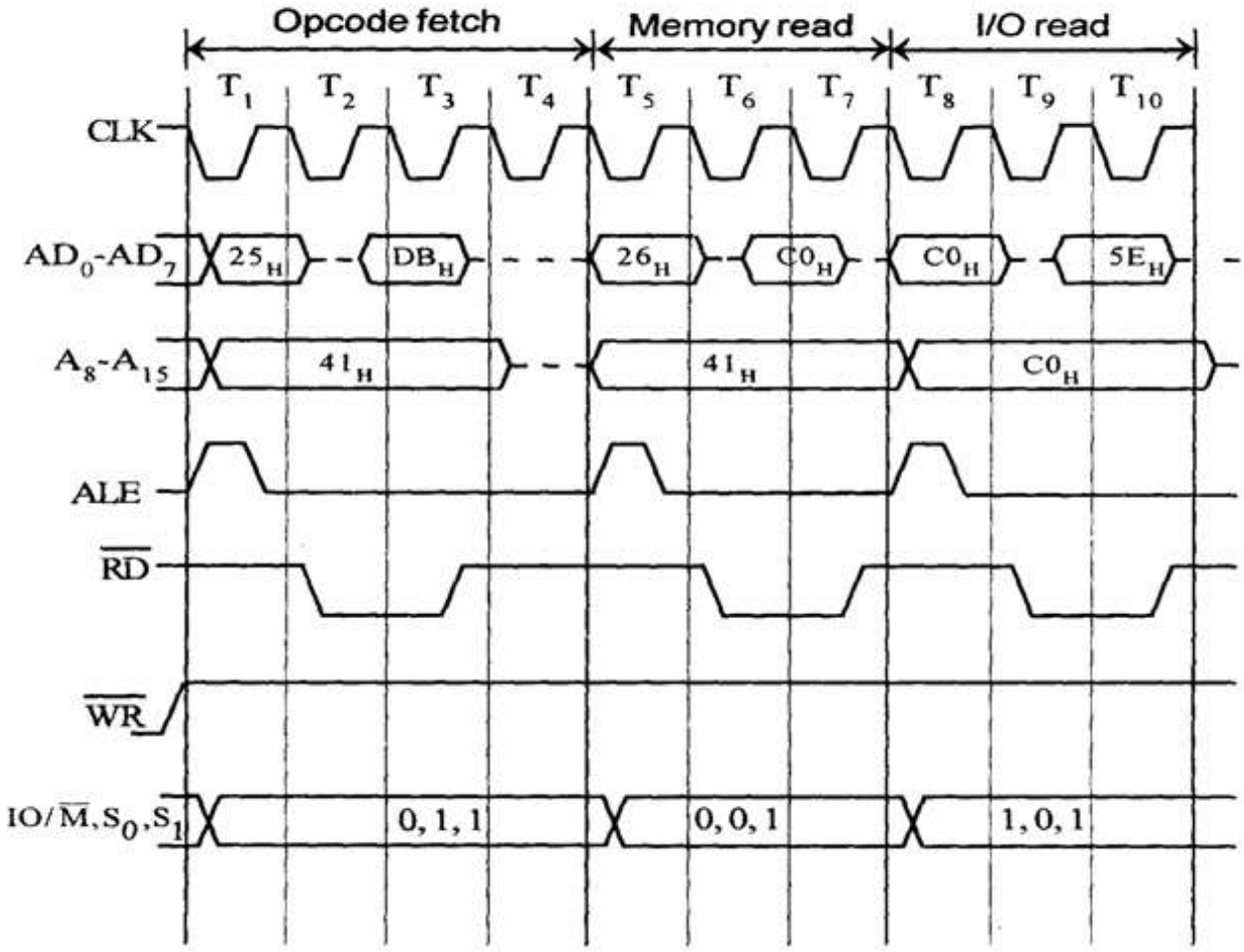
Address	Mnemonics	Opcode
41FF	STA 526A$_H$	32$_H$
4200		6A$_H$
4201		52$_H$



TimingdiagramforINC0H.

- FetchingtheOpcodeDBHfromthememory4125H.
- ReadtheportaddressC0Hfrom4126H.
- ReadthecontentofportC0Handsendittotheaccumulator.
- Letthecontentofportis5EH.

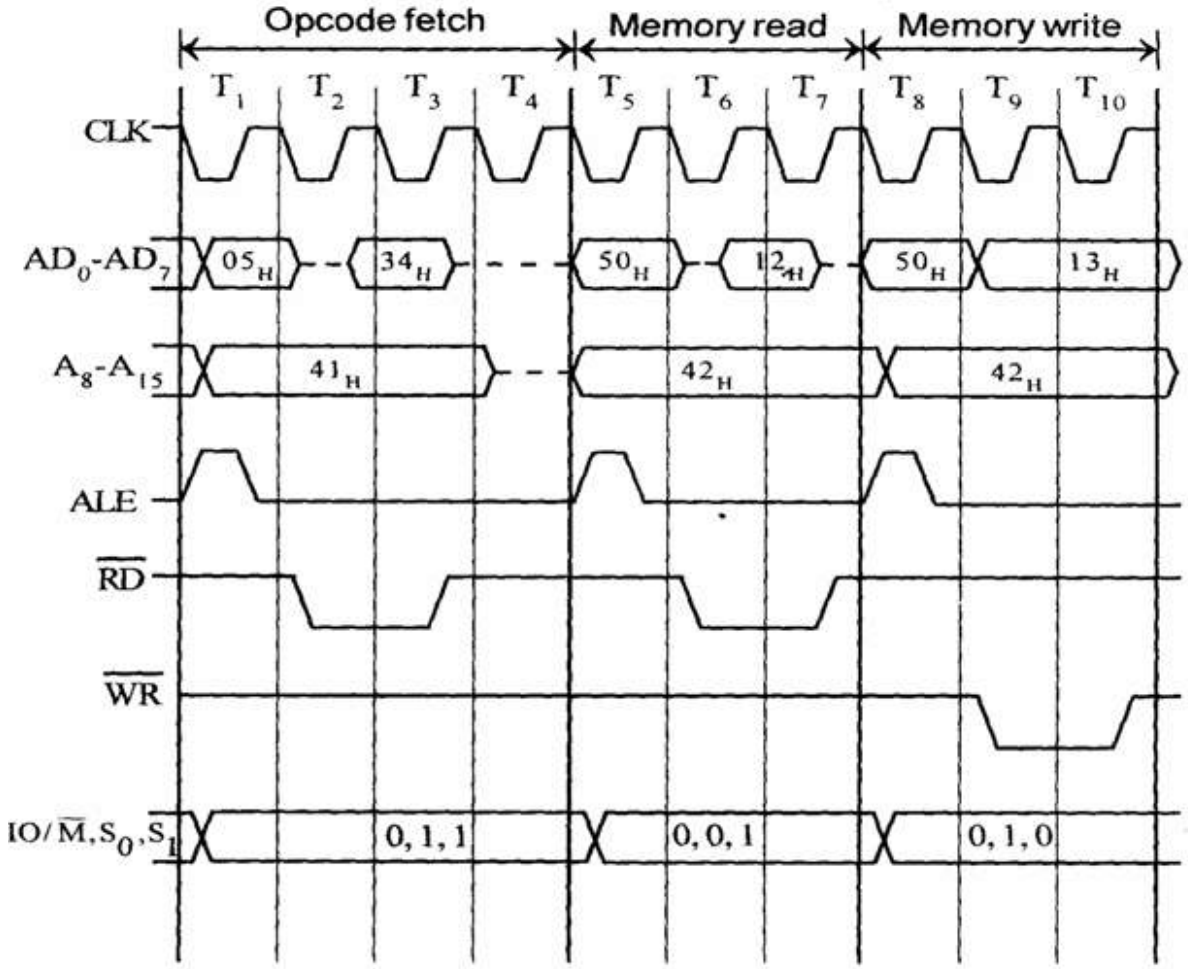
Address	Mnemonics	Opcode
4125	IN C0$_H$	DB$_H$
4126		C0$_H$



TimingdiagramforINRM

- FetchingtheOpcode34Hfromthememory4105H.(OFcycle)
- Letthememoryaddress(M)be4250H.(MRcycle-ToreadMemoryaddressanddata)
- Letthecontentofthatmemoryis12H.
- Incrementthememorycontentfrom12Hto13H.(MWmachinecycle)

Address	Mnemonics	Opcode
4105	INR M	34_H


Opcode fetch
Memory read
Memory write
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
CLK
AD0-AD7
05H
34H
50H
12H
50H
13H
A8-A15
41H
42H
42H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1
0, 0, 1
0, 1, 0

UNIT-4:MICROPROCESSORBASEDSYSTEMDEVELOPMENTAIDS

INTRODUCTIONTOINTERFACING:

- We know that a microprocessor is the CPU of a computer. A microprocessor can performsomeoperationonadataandgivetheoutput.Buttoperformtheoperationweneedaninputto enter the data and an output to display the results of the operation. So we are using akeyboard and monitor as Input and output along with the processor.Microprocessorsengineering involves a lot of other concepts and we also interface memory elements likeROM,EPROMtoaccessthe memory.
- Interfacingamicroprocessoristoconnectitwithvariousperipheralstoperformvariousoperations toobtainadesiredoutput.

INTERFACINGTYPES:

Therearetwotypesofinterfacingin 8085processor.

- MemoryInterfacing.
- I/Ointerfacing.

Purposeofinterfacing:

- The interfacing process involves matching the memory requirements with themicroprocessor signals.
- Theinterfacingcircuitthereforeshouldbedesignedinsuchawaythatitmatchesthememorysignalrequirementswith the signalsof themicroprocessor.
- ForexampleforcarryingoutaREADprocess,themicroprocessorshouldinitiateareadsignalwhichthe memoryrequirestoreadadata.
- Insimplewords,theprimaryfunctionofamemoryinterfacingcircuitistoaidthemicroprocessor inreadingandwritinga datatothe givenregisterof amemorychip.

Disadvantagesofinterfacing:

- Themaindisadvantagewiththisinterfacingisthatthemicroprocessorcanperform onlyonefunction.
- Itfunctionsasaninputdeviceifitisconnectedtobuffer.
- Itfunctionasanoutputdeviceifitisconnectedtolatch.
- Thusthecapabilityisverylimitedinthistypeofinterfacing.

TypesofCommunicationInterface

There are two ways in which a microprocessor can connect with outside world or othermemorysystems.

1. SerialCommunicationInterface
2. ParallelCommunicationinterface

SerialCommunicationInterface:

- Inserialcommunicationinterface,theinterfacegetsasinglebyte ofdatafromthemicroprocessor andsendsit bit bybit toother systemserially

- Theinterfacealsoreceivesdatabitbybitseriallyfromtheexternalsystemsandconvertsthedata intoa singlebyte andtransfersittothemicroprocessor.

ParallelCommunicationInterface:

- Thisinterfacegetsabyteofdatafrommicroprocessorandsendsitbitbybittotheothersystemsinsimultaneousor parallel.

- Theinterfacealsoreceivesdatabitbybitsimultaneouslyfromtheexternalsystemandconvertsthedata intoa singlebyte andtransfersit tomicroprocessor.

MEMORYMAPPING:

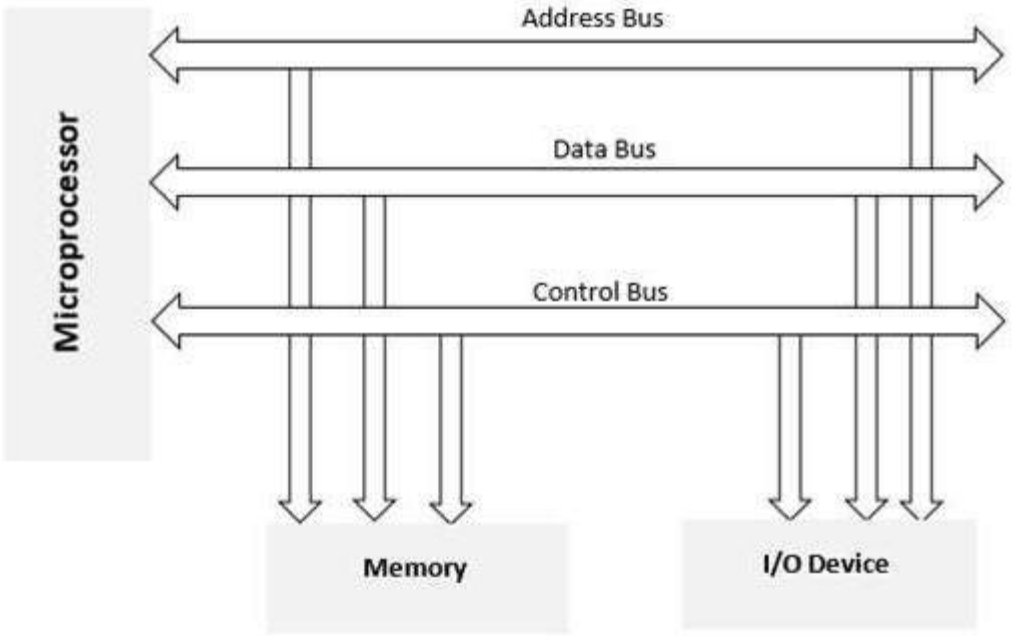
- Memorymappingisamethodtoexpandthememoryofthemicroprocessor.
- Beinglimitedinmemoryresources,microprocessorneedstobeconnectedtoexternalmemorydeviceslikeRAM/ROM/EEPROM.
-

- Theinterfacingbetweenthemicroprocessorandthememorydevicebyconnectingthedataanddressbusiscalled memorymapping.

I/OINTERFACING:

- I/O Interfacing is achieved by connecting keyboard (input) and display monitors (output)withthemicroprocessor.
- We know that keyboard and Displays are used as communication channel with outsideworld. So it is necessary that we interface keyboard and displays with the microprocessor.This is called I/O interfacing. In this type of interfacing we use latches and buffers forinterfacingthe keyboardsanddisplayswith the microprocessor.

BlockdiagramofmemoryandI/Ointerfacing



I/OMappingin8085Microprocessor:

I/Ointerfacing:

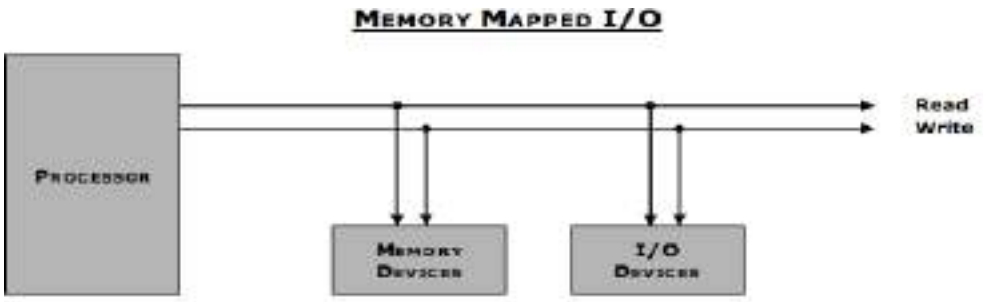
TherearetwomethodsofinterfacingtheInput/Outputdeviceswiththemicroprocessor.Theyare ,

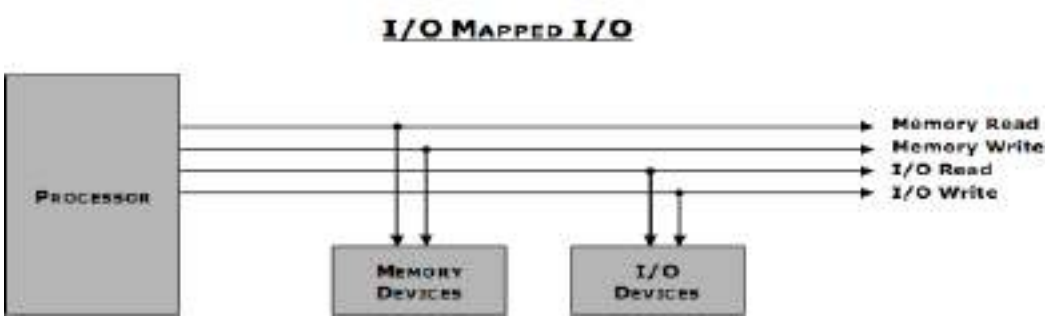
1) MemorymappedI/O and

2) I/OmappedI/O.

	MEMORYMAPPEDI/O	I/OMAPPEDI/O
1	**I/Odevicesaremappedintom emory space.**	**I/OdevicesaremappedintoI/Osp ace.**
2	I/Odevicesareallotted **memoryaddresses**.	I/Odevicesareallotted **I/Oaddresses**.
3	Processor **does not differentiate**betweenmemoryandI/ O.TreatsI/Odevicesalsolikememoryd evices.	Processor**differentiatesbetweenI/Od evices and memory**. It isolates I/Odevices.
4	I/Oaddressesareasbigasmemory addresses. E.g.in 8085, **I/Oaddresseswillbe16bit** asmemory	I/Oaddressesaresmallerthanmemory addresses. E.g. in 8085, **I/O addresseswillbe8bit**thoughmemorya ddresses

	addressesarealso16-bit.	are16-bit.
5	Thisallowsustoincreasethenumberof I/O devices. E.g. in 8085, we canaccessupto**2^{16}=65536I/O devices**.	Thisallowsustoaccesslimitednumberof I/O devices. E.g. in 8085, we canaccessonlyup to **2^{8}=256I/Odevices**.
6	We can transfer data from I/Odevicesusing**anyinstructionli keMOV**etc.	We can transfer data from I/O deviceusing**dedicatedI/Oinstruction slikeINandOUTONLY**.
7	Datacanbetransferredusing**anyre gister**of theprocessor.	Datacanbetransferredonlyusingafix edregister.E.g.in8085only**"A" register**.
8	Weneed**onlytwocontrol**signalsin thesystem:Read andWrite.	Weneed**fourcontrolsignals**:MemoryRe ad, Memory Write and I/O Read andI/OWrite
9	Memoryaddressesarebigso **addressdecodingwill beslower**.	I/Oaddressesaresmallerso**addressd ecodingwill befaster**.
10	Address decoding will be **morecomplexand costly**.	Addressdecodingwillbe **simplerandcheaper**.





MEMORYMAPPEDI/O:

InthismethodtheI/Odevicesaretreatedlikethememory.Apartofthememoryaddress space is used for the I/O devices. The memory mapped I/O scheme isshowninfigure.

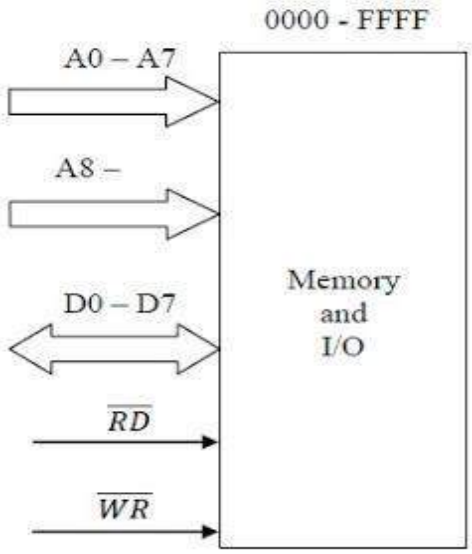


Figure:MemorymappedI/Oscheme

• Inmemorymapped I/Oscheme,thesameaddressspaceisusedforbothmemoryandI/Odevices.

• The microprocessor uses the sixteen address line A_0 – A_7 and A_8– A_{15} for the memoryaswellasforthe I/O devices.

• The I/O devices share the address space with the memory. All the memory relatedinstructionsare usedfor addressingI/Odevicesalso.

• NoseparateINandOUTinstructionsarerequiredinmemorymappedI/Oscheme.

• IO/M'pinisnotrequired.

Stepsformemoryoperations(memoryreadandmemorywrite):

- When the memory related instructions like LDA and STA are used, themicroprocessor placesthe16-bitaddressontheaddressbus.
- *RD*'isactivatedforreadoperationand*WR*' isactivatedforwriteoperation.

StepsforI/Ooperations(I/OreadandI/Owrite):

- When the memory related instructions like LDA and STA are used, themicroprocessor placesthe16-bitaddressontheaddressbus.
- *RD*'isactivatedforreadoperationand*WR*' isactivatedforwriteoperation.

I/OMAPPEDI/O:

In this method, I/O devices are treated as I/O devices and memory is treated asmemory.Separateaddressspaceisusedformemoryand I/O.TheI/OmappedI/Osche meisshowninfigure.

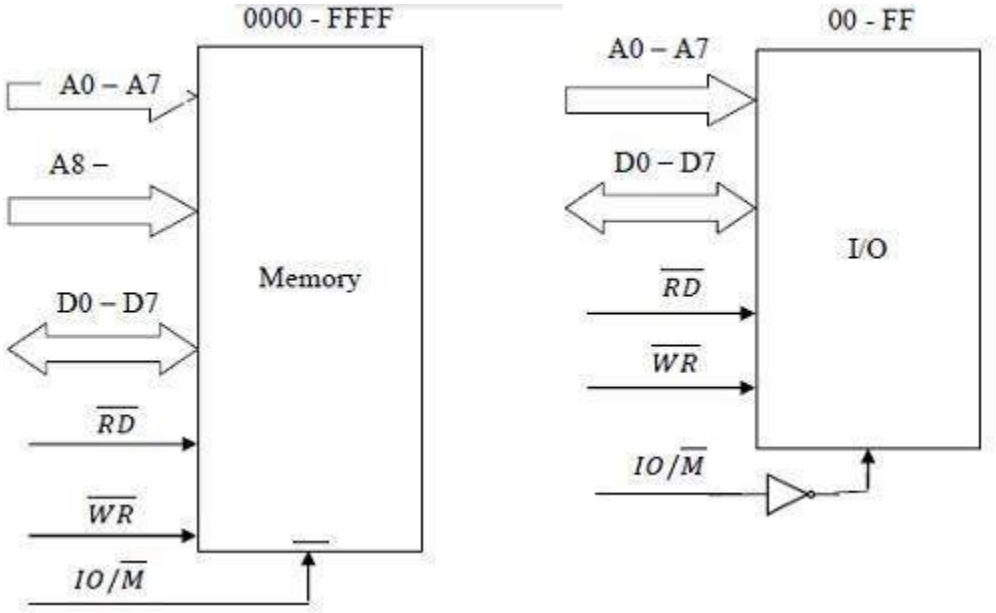


Figure:I/OmappedI/Oscheme

• In I/O mapped I/O scheme, the microprocessor uses the sixteen address lines A_0 – A_7 and A_8 – A_{15} for the memory and eight address lines A_0 to A_7 to identify an input /outputdevice.

• Here,thefulladdressspace0000–FFFFisusedfor thememoryandaseparateaddressspace 00–FFisusedforthe I/Odevices.

• Hence, the microprocessor can address 65536 (2^{16}) memory locations 256 (2^8)inputdevicesand256(2^8)output devicesseparately.

• INandOUTinstructionsareusedtoactivatetheIO/*M*'signal.

• WhenIO/*M*'islow,thememoryisselectedfor readingandwritingoperations.

• WhenIO/*M*'ishigh, theI/Oportisselectedforreadingandwritingoperations.

Stepsformemoryoperations(memoryreadandmemorywrite):

- WhenthememoryrelatedinstructionslikeLDAandSTAareused,themicr oprocessor placesthe16-bitaddressontheaddressbus.
- ThemicroprocessormakestheIO/*M*'linelow.
- The microprocessor makes the *RD*' low for read operation and *WR*' low forwriteoperation.

StepsforI/Ooperations(I/OreadandI/Owrite):

- 1 When the I/O related instructions like IN and OUT are used, themicroprocessorplacesthe8-bitaddresson theaddressbusA_0–A_7aswellasA_8–A_{15}.
- IO/M'lineismadehigh.
- The microprocessor makes the *RD*' low for read operation and *WR*' low forwrite operation.

MEMORYINTERFACING:

- Whileexecutinganinstruction,thereisanecessityforthemicroprocessortoacces s memory frequently for reading various instruction codes and datastoredinthe memory.
- The read/write operations are monitored by control signals. Themicroprocessoractivatesthesesignalswhenitwantstoreadfromandwritei ntomemory.
- The interfacing circuit aids in accessing the memory requires some signals toreadfromandwritetoregisters.Similarlythemicroprocessortransmitssomesig nalsforreadingorwritinga data.

  OR
- MemoryInterfacingisusedwhenthemicroprocessorneedstoaccessmemoryfreq uently for reading and writing data stored in the memory. It is used whenreading/writingtoaspecific register of amemorychip.

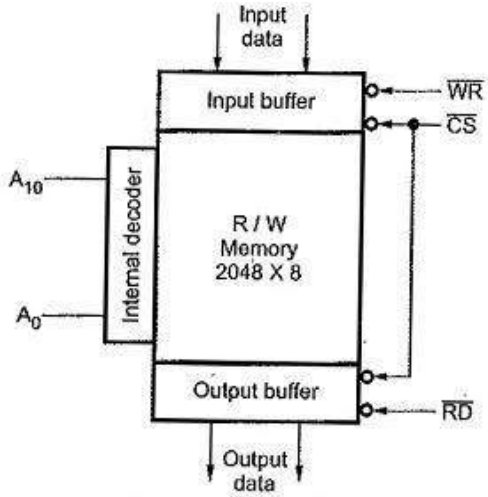
<u>Thememoryinterfacingrequiresto:</u>

- Selectthechip
- Identifytheregister
- Enabletheappropriatebuffer.

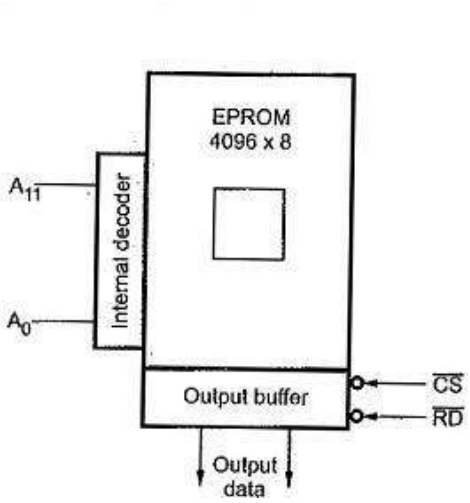
- Microprocessor system includes memory devices and I/O devices. It isimportanttonotethatmicroprocessorcancommunicate(read/write)witho nly one device at a time, since the data, address and control buses arecommonfor allthe devices.
- InordertocommunicatewithmemoryorI/Odevices,itisnecessarytodecodethead dressfromthemicroprocessor.
- Duetothiseachdevice(memoryorI/O)canbeaccessedindependently.Thefollo wingsectiondescribescommonaddressdecodingtechniques.

MemoryStructureanditsRequirements:

Asmentionedearlier,read/writememoriesconsistofanarrayof registers,inwhicheach register has unique address. The size of the memory is N x M, where N is thenumber ofregistersandMistheword length,innumber ofbits.



LogicdiagramforRAM



LogicdiagramforEPROMI

NTERFACINGEPROM&RAMMEMORIES:

- Microprocessor 8085 can access 64Kbytes memory since address bus is 16-bit.But it is not always necessary to use full 64Kbytes address space. The totalmemorysize depends uponthe application.
- Generally EPROM (or EPROMs) is used as a program memory and RAM (orRAMs) as a data memory. When both, EPROM and RAM are used, the totaladdressspace 64Kbytesissharedbythem.

- Thecapacityofprogrammemoryanddatamemorydependsontheapplication.

- Itisnotalwaysnecessarytoselect1EPROMand1RAM.WecanhavemultipleEPROM sandmultiple RAMsasper therequirementofapplication.

- We can place EPROM/RAM anywhere in full 64 Kbytes address space. Butprogrammemory(EPROM)shouldbelocatedfromaddress0000Hsincereseta ddressof 8085 microprocessoris0000H.

- ItisnotalwaysnecessarytolocateEPROMandRAMinconsecutivememory.

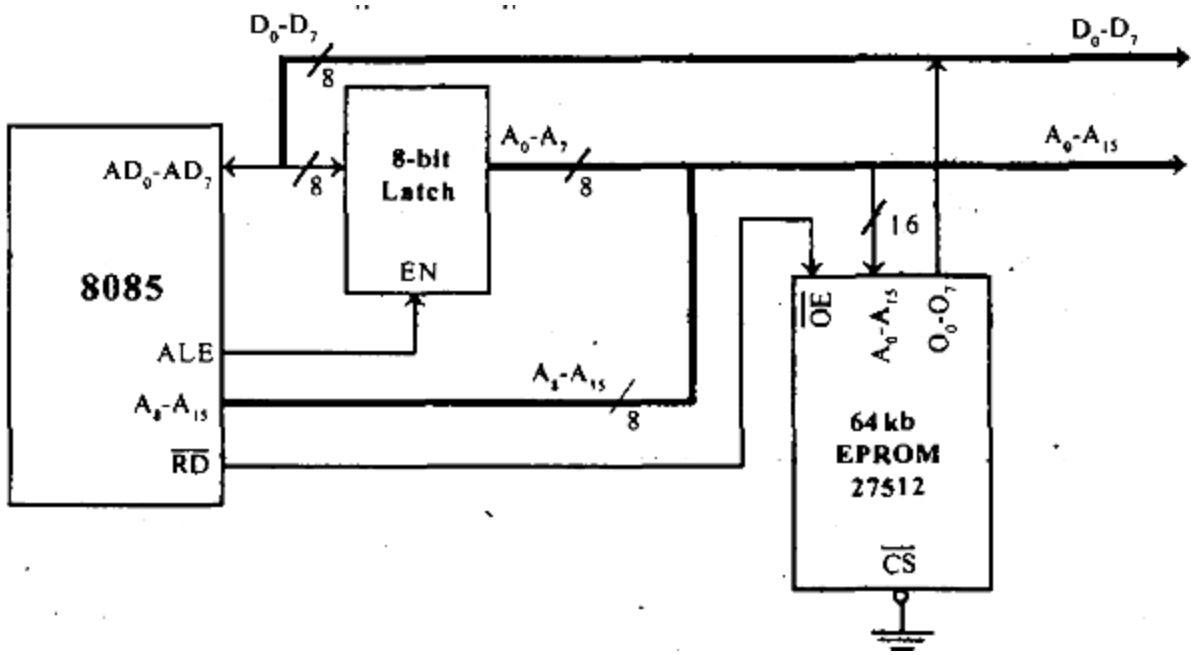
- **For example**: If the mapping of EPROM is from 0000H to OFFFH, it is not mustto locate RAM from 1000H. We can locate it anywhere between 1000H andFFFFH.Wheretolocatememorycomponenttotallydependsontheapplication

EXAMPLESOFMEMORYINTERFACING

EXAMPLE-1

Considerasysteminwhichthefullmemoryspace64kbisutilizedforEPROMmemory.Int erfacethe EPROMwith8085processor.

- Thememorycapacityis64Kbytes.i.e.
- 2^n=64x1000bytes wheren=addresslines.
- So,n=16.
- In this system the entire 16 address lines of the processor are connected toaddressinputpinsofmemoryICin ordertoaddresstheinternal locationsofmemory.
- Thechipselect(CS)pinofEPROMispermanentlytiedtologiclow(i.e.,tiedtoground).
- SincetheprocessorisconnectedtoEPROM,theactivelowRDpinisconnectedtoacti ve low outputenablepinofEPROM.
- TherangeofaddressforEPROMis0000HtoFFFFH.

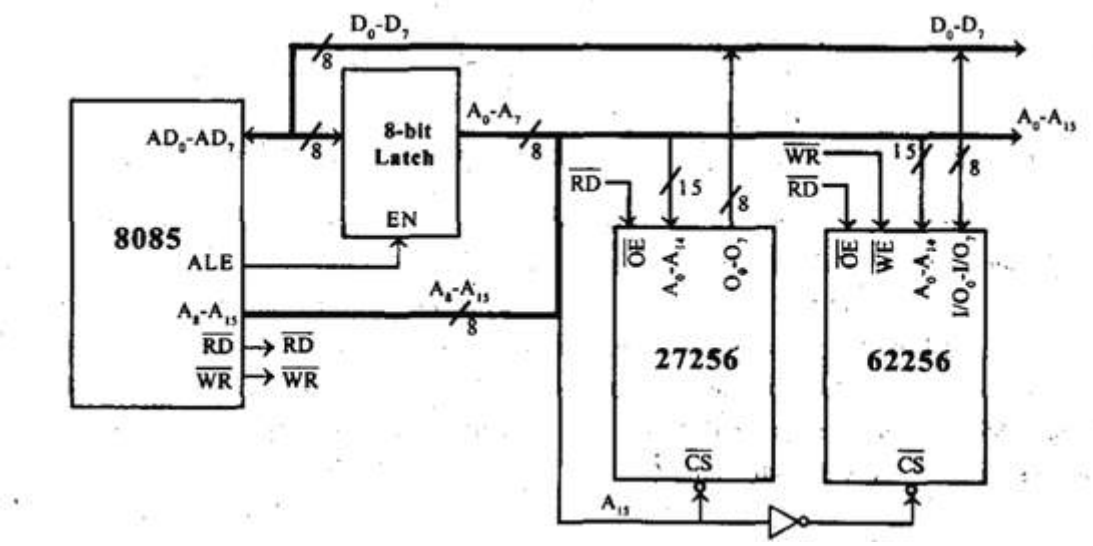


Interfacing64KbEPROMwith8085E

XAMPLE-2

Consider a system in which the available 64kb memory space is equallydividedbetweenEPROMandRAM.InterfacetheEPROMand RAMwith8085processor.

- Implement32kbmemorycapacityofEPROMusingsingleIC27256.
- 32kbRAMcapacityisimplementedusingsingleIC62256.
- The 32kb memory requires 15 address lines and so the address lines A0 - A14oftheprocessorareconnectedto15addresspinsofbothEPROMandRAM.
- TheunusedaddresslineA15isusedastochipselect.IfA15is1, itselectRAMandifA15is0, itselect EPROM.
- InverterisusedforselectingthememoryM.
- ThememoryusedisbothRamandEPROM,sothelowRDandWRpinsofprocess orareconnectedtolowWEandOEpinsofmemoryrespectively.
- TheaddressrangeofEPROMwillbe0000Hto7FFFHandthatofRAMwillbe7FFFHt oFFFFH.

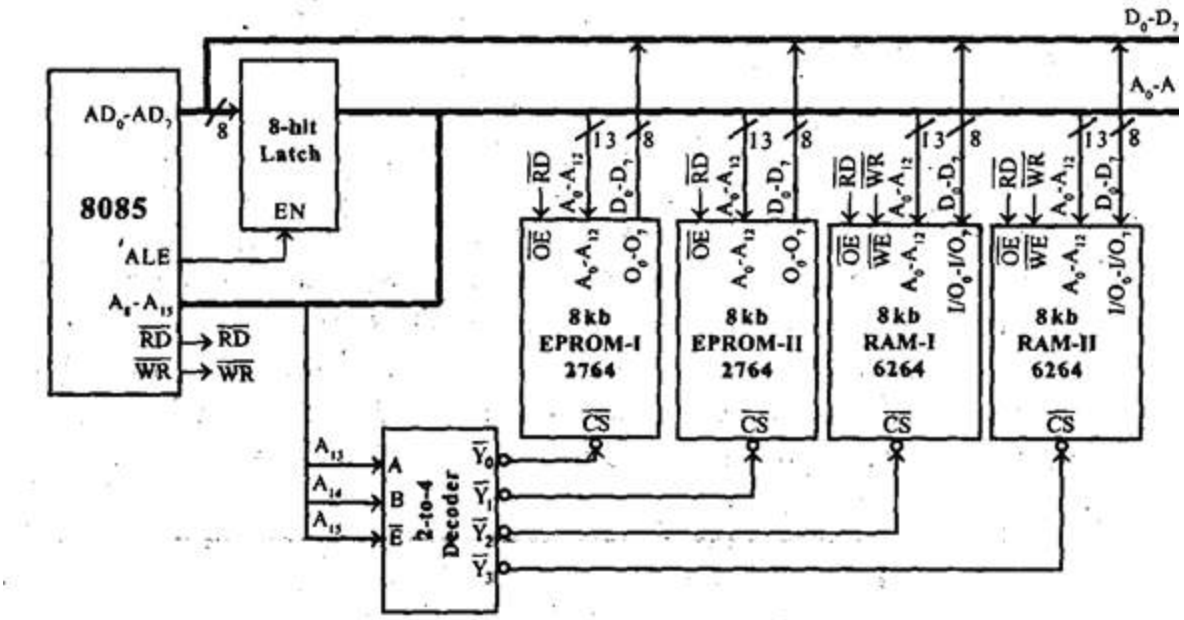


Interfacing32KbEPROMand32KbRAMwith8085

EXAMPLE-3

Consider a system in which 32kb memory space is implemented using fournumbersof8kbmemory.InterfacetheEPROMandRAMwith8085processor.

- Thetotalmemorycapacityis32Kb.So, lettwonumberof8kbn memorybeEPROMandtheremainingtwonumbersbeRAM.
- Each8kbmemoryrequires13addresslinesandsotheaddresslinesA0-A12ofthe processor areconnectedto13addresspinsofallthememory.
- The address lines and A13 - A14 can be decoded using a 2-to-4 decoder togeneratefourchipselect signals.
- Thesefourchipselectsignalscan beusedtoselectoneofthefourmemoryICat anyonetime.
- Theaddressline A15isusedasenablefordecoder.
- Thesimplifiedschematicmemoryorganizationisshown.



Interfacing16KbEPROMand16KbRAMwith8085

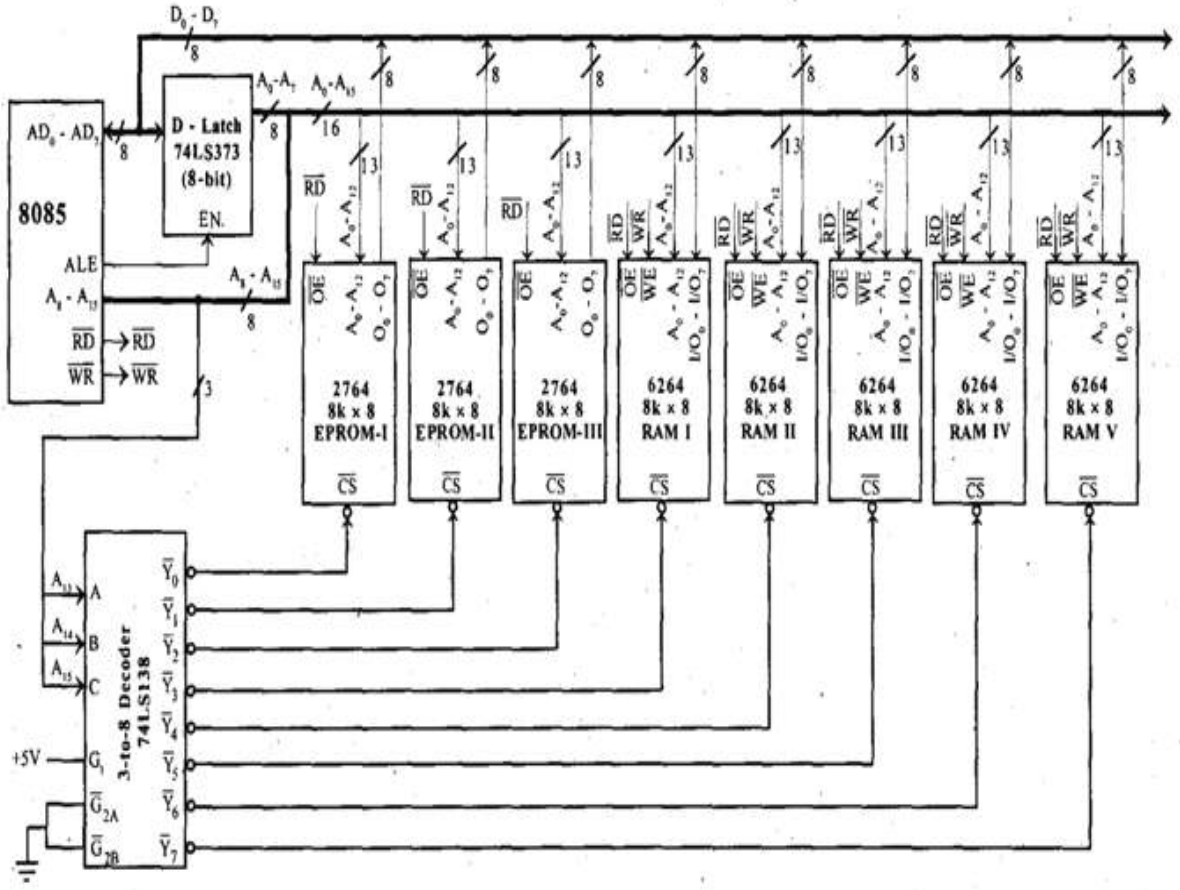
- Theaddressallottedtoeachmemory ICisshowninfollowingtable.

Device	Binary address																Hexa address
	Decoder enable/input			Input to address pins of memory IC													
	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	
8kb EPROM - I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF
8kb EPROM - II	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	2001
	0	0	1	0	0	0	0.	0	0	0	0	0	0	0	1	0	2002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF
8kb RAM - I	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4001
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5FFF
8kb RAM -II	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000
	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	6001
	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	6002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7FFF

EXAMPLE-4

Considerasysteminwhichthe64kbmemoryspaceisimplementedusingeight numbers of 8kb memory. Interface the EPROM and RAM with 8085processor.

- Thetotalmemorycapacityis64Kb.So, let4numbersof8KbEPROMand4numbersof8KbRAM.
- Each 8kb memory requires 13 address lines. So the address line A0 - A12 oftheprocessor areconnectedto13addresspinsofallthememory lCs.
- The address lines A13, A14 and A15 are decoded using a 3-to-8 coder togenerate eight chip select signals. These eight chip select signals can be usedtoselect one ofthe eightmemoriesat any onetime.
- Thememoryinterfacingisshowninfollowingfigure.



Interfacing4no.8KbEPROMand 4no.8KbRAMwith8085

- The address allocation for Interfacing 4 no. 8Kb EPROM and 4 no. 8Kb RAMwith8085 is

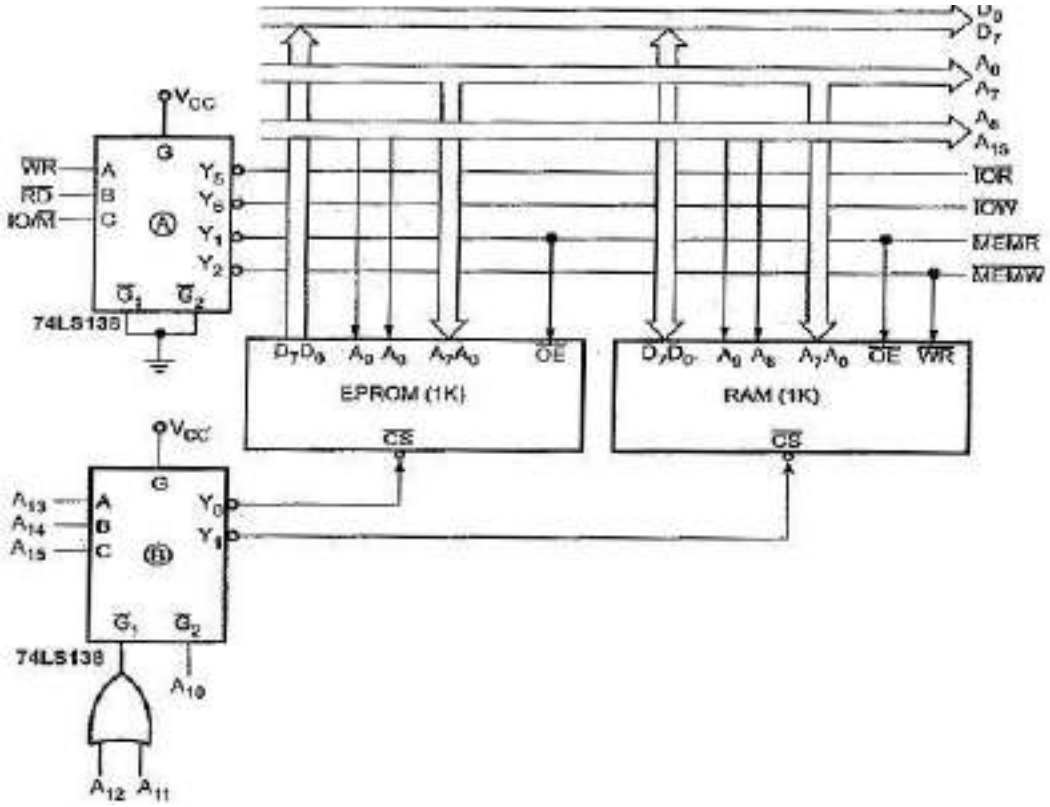
Memory IC chip	Binary address																Hexa address
	Decoder input			input to memory address pins													
	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
EPROM I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF
EPROM II	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	2001
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF
EPROM III	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4001
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5FFF
RAM I	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000
	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	6001
	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	6002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7FFF
RAM II	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8001
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	8002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	9FFF
RAM III	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	A000
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	A001
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	A002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	BFFF
RAM IV	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000
	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	C001
	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	C002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	DFFF
RAM V	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	E000
	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	E001
	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	E002
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF

AddressDecodingTechniques:

- Absolutedecoding/FullDecoding
- Lineardecoding/PartialDecoding

Absolutedecoding:

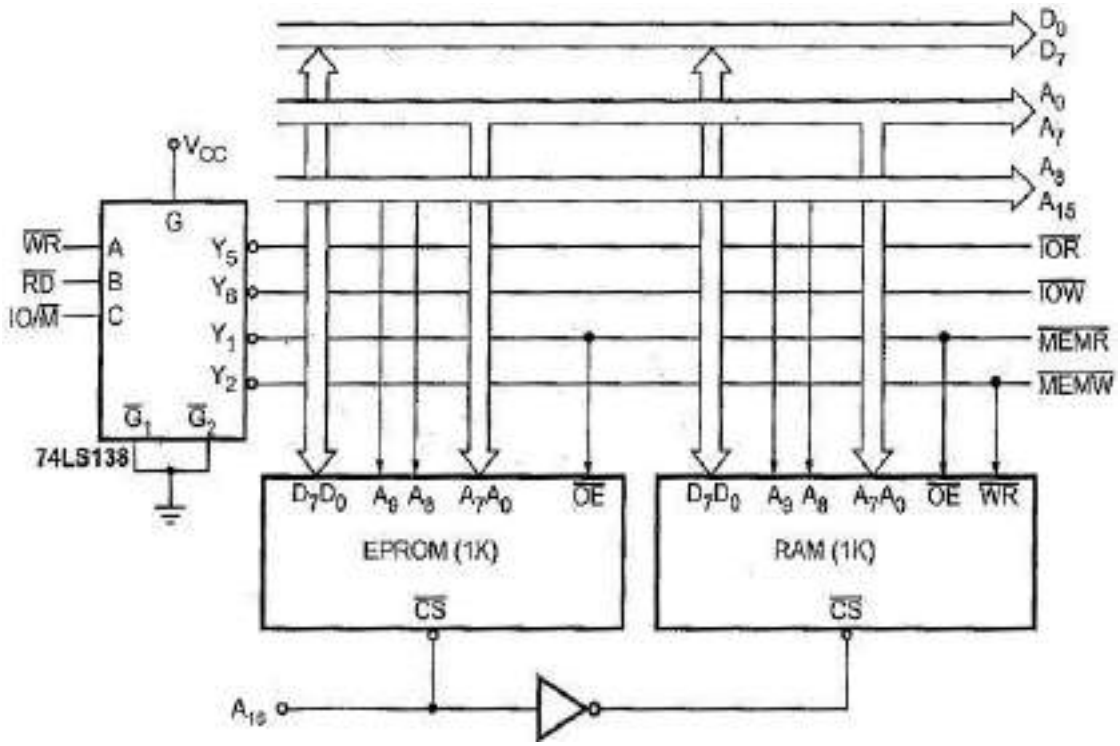
- In absolute decoding technique, all the higher address lines are decoded toselectthememorychip,andthememorychipisselectedonlyforthespecifiedlogic levels on these high-order address lines; no other logic levels can selectthechip.
- ThisfigureshowstheMemoryInterfacingin8085withabsolutedecoding.Thisaddressingtechniqueisnormallyused inlargememorysystem



Absolutedecodingtechniquediagram:

Lineardecoding:

- In small systems, hardware for the decoding logic can be eliminated by usingindividualhigh-order addresslinestoselectmemorychips. Thisisreferredtoaslineardecoding.
- ThisfigureshowstheaddressingofRAMwithlineardecodingtechnique.
- Thistechniqueisalsocalled**partialdecoding**.Itreducesthecostofdecodingcircuit,butithasadrawbackofmultipleaddresses(shadowaddresses).



Lineardecodingtechniquediagram

- It shows the addressing of RAM with linear decoding technique. A_{15} addressline, is directly connected to the chip select signal of EPROM and afterinversionitisconnectedtothe chipselectsignaloftheRAM.
- Therefore,whenthe statusofA_{15}lineis'zero',EPROMgetsselectedandwhen thestatusofA_{15}lineis'one'RAMgetsselected.
- Thestatusoftheotheraddresslinesisnot considered,sincethoseaddresslinesare notused for generationofchipselectsignals.

8255PPI:

- 8255isaprogrammableI/Odevicethatactsasinterfacebetweenperipheraldevic esandthemicroprocessorfor parallel datatransfer.
- 8255PPI(programmableperipheralinterface)isprogrammedinawaysoasto have transfer of data in different conditions according to the need of thesystem.
- In 8255, 24 pins are assigned to the I/O ports. Basically it has three, 8-bitportsthat areused for simpleorinterruptI/Ooperations.

- Thethreeportsare**PortA,PortBand PortC**andaseachporthas8lines,but the 8bitsofportCisdividedinto2groupsof4-bit each.
- Thesearegivenasport Cloweri.e.,PC_3–PC_0andportCupperi.e.,PC_7–PC_4.

- Andarearrangedingroupof12pinseachthusdesignatedasGroupAandGroup B.

Thetwomodesinwhich8255canbeprogrammedareasfollows:

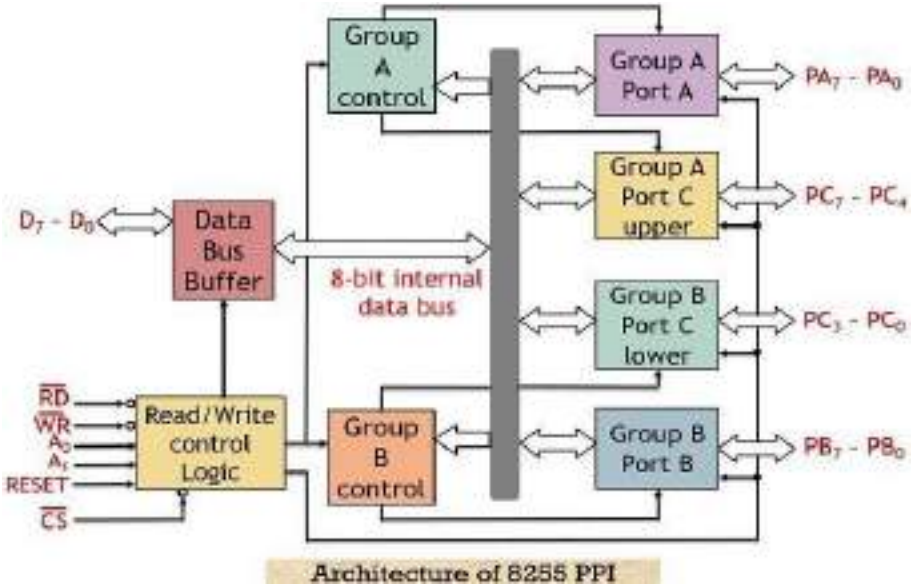
1. Bitset/resetmode
2. I/Omode

- ThebitsofportCgetssetorreset intheBSRmode.Theothermodeof8255i.e.,I/Omode isfurther classifiedinto.

Mode0:Simpleinput/output
Mode1:Inputoutputwithhandshaking
Mode2:BidirectionalI/Ohandshaking

- **Mode1andMode2**botharesamebuttheonlydifferenceismode1doesnotsupport bidirectionalhandshaking.
- Thismeansif8255isprogrammedto**mode1**input,thenitwillparticularlybeconnec ted to an input device and performs the input handshaking with theprocessor.
- Butifitisprogrammedto**mode2**thenduetobidirectionalnature,thePPIwillperfor m both input and output operation with the processor according to thecommandreceived.

Architectureof8255PPI:



Architecture of 8255 PPI

Thefigureaboverepresentsthearchitecturalrepresentationof8255PPI:Letus

understandtheoperationperformedbyeachunitseparately.

Databusbuffer:

- Itisusedtoconnecttheinternalbusof8255withthesystembussoastoestablis hproperinterfacingbetweenthe two.
- Thedatabusbufferallowstheread/writeoperationtobeperformedfrom/totheC PU.
- The buffer allows the passing of data from ports or control register to CPU incase of write operation and from CPU to ports or status register in case of readoperation.

Read/Writecontrollogic:

- This unit manages the internal operations of the system. This unit holds theabilitytocontrolthetransferofdataandcontrolorstatuswordsbothinternallyan dexternally.
- Whenever thereexistsaneedfordatafetchthenitacceptstheaddressprovidedby the processor through the bus and immediately generates command to the 2control groupsfortheparticular operation.

GroupAandGroupBcontrol:

- These two groups are handled by the CPU and functions according to thecommandgenerated bythe CPU.
- TheCPUsendscontrolwordstothegroup AandgroupBcontrolandtheyinturnsendsthe appropriatecommandtotheirrespectiveport.
- **GroupA**theaccessofthe**portA**andhigherorderbitsof**portC**.While **groupB** controls port B with the lower order bits of **port**

C.PinDiagram of8255PPI

Thefigurebelowrepresentsthe40pinconfigurationof8255programmableperipher alinterface:

Pin Diagram of 8255 PPI

CS:

- Itstandsfor chipselect.Alowsignalatthispinshowstheenablingofcommunication betweenthe8255 andtheprocessor.
- Morespecificallywecansaythatthedatatransferoperationgetsenabledbyanactivel ow signalatthispin.

RD:

- Itisthesignalusedforreadoperation.
- Alowsignalat thispinshowsthat CPUisperformingreadoperationattheportsor statusword.
- Orwecansaythat8255isprovidingdataorinformationtotheCPUthroughdatabuffer.

WR:

- It shows write operation. A low signal at this pin allows the CPU to perform writeoperationovertheportsorcontrolregisterof8255usingthedatabusbuffer.

A_0andA_1:

- Thesearebasicallyusedtoselectthedesiredportamongalltheportsofthe8255a nditdosobyformingconjunctionwith RDandWR.
- ItformsconnectionwiththeLSBoftheaddressbus.

Thetablebelowshowstheoperationofthecontrolsignals:

A_1	A_0	RD'	WR'	CS'	Input/OutputOperation
0	0	0	1	0	PortA-DataBus
0	1	0	1	0	PortB-DataBus
1	0	0	1	0	PortC-DataBus
0	0	1	0	0	DataBus-PortA
0	1	1	0	0	DataBus-PortB
1	0	1	0	0	DataBus-PortC
1	1	1	0	0	DataBus-Controlregister

Reset:

- Itisanactivehighsignalthatshowstheresettingofthe PPI.
- A high signal at this pin clears the control registers and the ports are set in theinputmode.
- Initializingtheportstoinputmodeisdonetopreventcircuitbreakdown.

- Asincaseofreset condition,iftheportsareinitializedtooutputmodethenthereexistchancesofdestruct ionof8255 along with theprocessor.

PA7-PA0:

Theseareeight**portA**linesthatactsaseitherlatchedoutputorbufferedinputlinesdep endinguponthecontrolwordloadedintothecontrolwordregister.

PC7-PC4:

- Upper nibble of port C lines. They may act as either output latches or inputbufferslines.Thisportalsocanbeusedfor generationofhandshakelinesinmode1 ormode 2.

PC3-PC0:

ThesearethelowerportClines,otherdetailsarethesameasPC7-PC4lines.

PB0-PB7:

ThesearetheeightportBlineswhichareusedaslatchedoutputlinesorbufferedinput linesinthe same wayasportA.

D0-D7:Thesearethedatabuslinesthosecarrydataorcontrolwordto/fromthemicroprocessor.

VCC:

Itisasupplyvoltage.The8255requires+5Vsupplytooperate.

GND:

Itisthegroundreference.Ifthereisexcessivepowersupplythen itpassedtoground.

MODESOFOPERATION:

Aswehavealreadydiscussedthat8255hastwomodesofoperation.Theseareasfollows:

1. Bitset/resetmode
2. I/Omode

BitSet-Resetmode:

WhenportCisutilizedforcontrolor statusoperation,thenbysendinganOUTinstruction,each individualbit ofportCcanbesetorreset.

I/Omode:

AsweknowthatI/Omodeissub-classified into3modes.So,letusnowdiscussthe3modeshere.

Mode0:Input/outputmode:

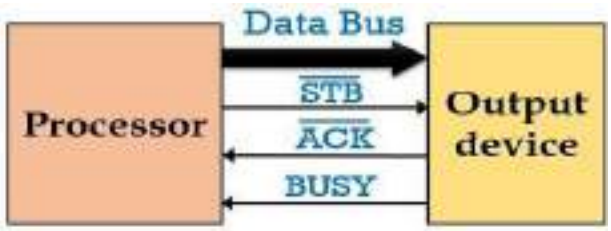
Thismodeisthesimpleinputoutputmodeof8255whichallowstheprogrammingof each port as either input or output port. The input/output feature of mode 0includes:

- Itdoesnot supporthandshakingorinterruptcapability.
- Theinputportsarebufferedwhileoutputsarelatched.

Mode1:Input/outputwithhandshaking:

Mode 1 of 8255 supports handshaking with the ports programmed as either inputor output mode. We know that it is not necessary that all the time the data istransferredbetweentwodevicesoperatingatsamespeed.So,handshakingsignalsare used to synchronize the data transfer between two devices that operates atdifferentspeeds.

ThefigurebelowshowsthedatatransferringbetweenCPUandanoutputdevicehavingdifferent operatingspeeds:



Data Transfer using handshaking signals

- HereSTBsignalisusedtoinformtheoutputdevicethatdataisavailableonthedata busbythe processor.
- HereportAandportBcanbeseparatelyconfiguredaseitherinputoroutputport.
- Both the port utilizes 3-3 lines of port C for handshaking signals. The rest twolinesoperatesasinput/output port.
- Itsupportsinterruptlogic.
- Thedataattheinputoroutputportsarelatched.

Mode2:BidirectionalI/Oportwithhandshaking:

- Inthismode,theportscanbeutilizedforthebidirectionalflow ofinformationbyhandshakingsignals.
- ThepinsofgroupAcanbeprogrammedtoactsasbidirectionaldatabusandtheport Cupper (PC_7–PC_4)are usedbythehandshakingsignal.
- Therest4lowerportCbitsareutilizedforI/Ooperations.
- Asthedatabusexhibitsbidirectionalnaturethuswhentheperipheraldevicerequ estforadatainputonlythentheprocessorloadthedatainthedatabus.
- PortBcanbeprogrammedinmode0and 1.Andinmode1thelowerbitsofportCof groupBare used for handshaking signals.

DACINTERFACING:

- Digital-to-AnalogConversionorsimplyDAC,isadevicethatisusedtoconvertadigital (usually binary) code into an analog signal (current, voltage, or electriccharge).
- Digital-to-analog conversion is the primary means by which digital equipmentsuch as computer-based systems are able to translate digital data into real-worldsignals that are more understandable to or useable by humans, such as music,speech,pictures,video.
- Italsoallowsdigitalcontrolofmachines,equipment,householdappliances.
- When data is in binary form, the 0's and 1's may be of several forms such as theTTLformwherethelogiczeromaybeavalueupto0.8voltsandthe1maybeavoltag efrom2 to5 volts.
- Thedatacanbeconvertedtocleandigitalformusinggateswhicharedesignedtobeono roffdependingonthe valueof theincomingsignal.
- Dataincleanbinarydigitalformcanbeconvertedtoananalogformbyusingasummi ngamplifier.
- Hereisasimplifiedfunctionaldiagramofan8- bitDAC.Therearemainlytwotechniquesused fordigitaltoanalogconversion

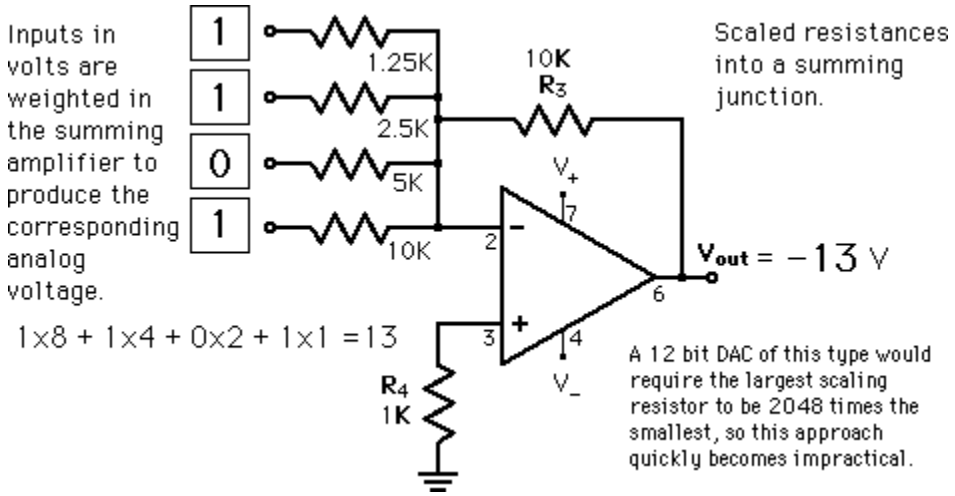
  1. WeightedSummingAmplifier

  2. R-2RNetwork

WeightedSumDAC:

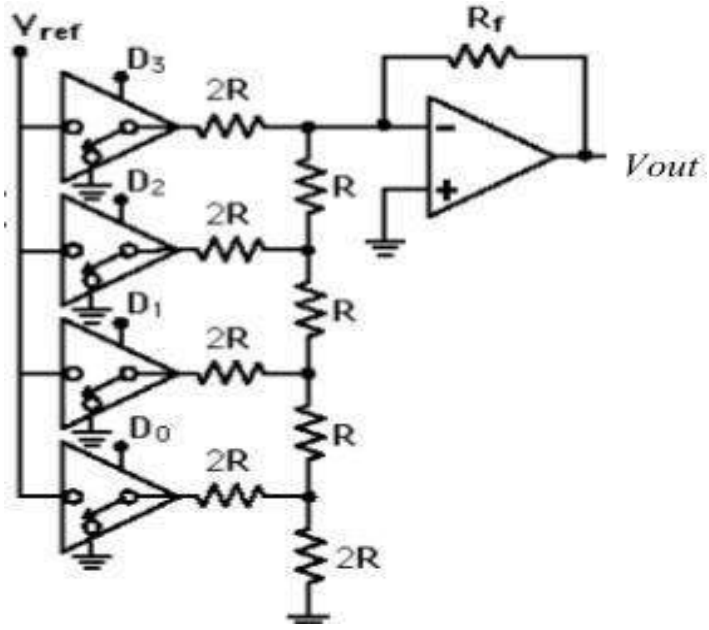
- OnewaytoachieveD/Aconversionistouseasummingamplifier.

- Thisapproachisnotsatisfactoryforalargenumberofbitsbecauseitrequirestoomuch precisioninthesummingresistors.
- ThisproblemisovercomeintheR-2RnetworkDAC.

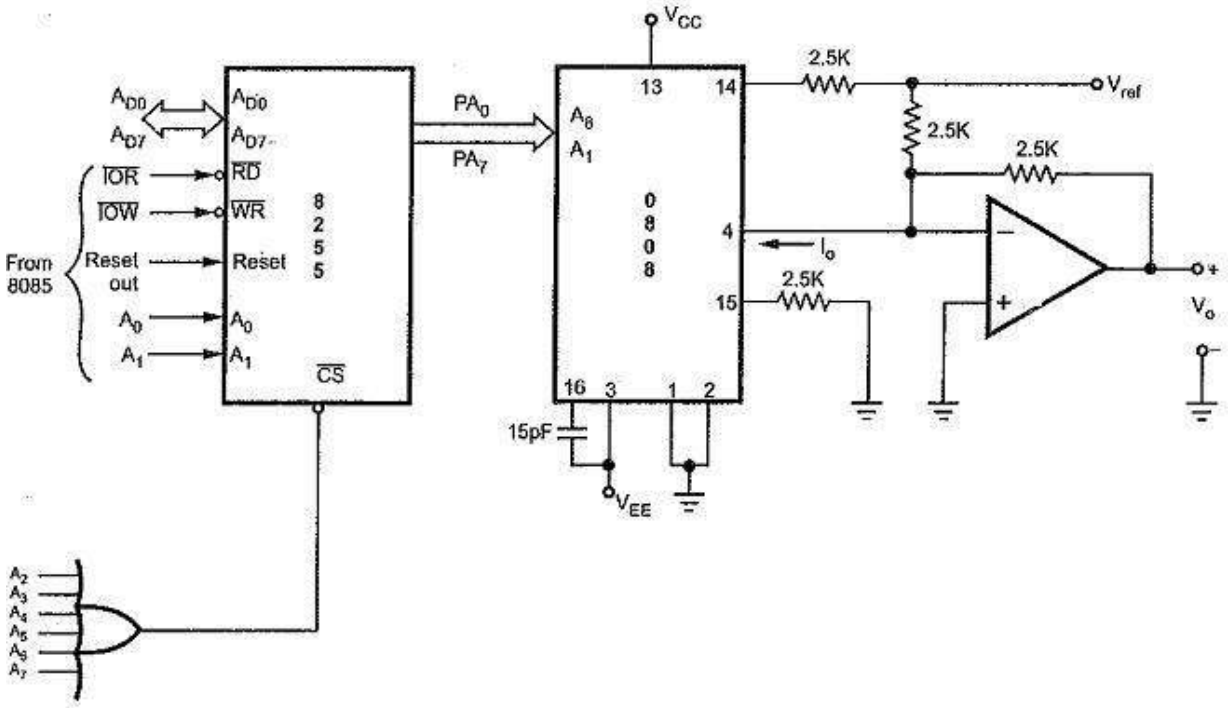


R-2RLadderDAC:

- ThesummingamplifierwiththeR-2Rladderofresistancesshownproducestheoutputwherethe D'stakethe value 0or1.
- ThedigitalinputscouldbeTTLvoltageswhichclosetheswitchesonalogical1andleave it groundedfora logical0.
- This is illustrated for 4 bits, but can be extended to any number with just theresistancevaluesR and2R.



- TheinterfacingofDAC0808withmicroprocessor8085isshownbelow.Here,programmable peripheral interface, 8255 is used as parallel port to send thedigitaldata toDAC.



Interfacingof0808withmicroprocessor

InterfacingDigital-To-Analogconverterto8085using8255"

- Figure below shows the interfacing of DAC 0808 with microprocessor 8085.Here,programmableperipheralinterface,8255isusedasparallelporttosendthedigitaldata toDAC.
- **I/OMapfor8255**

PORT/REGISTER	ADDRESS
PortA	00
PortB	01
PortC	02
ControlRegister	03

Program:

MVIA,80H;Initialization-controlwordfor8255toconfigureallportsasoutputPorts

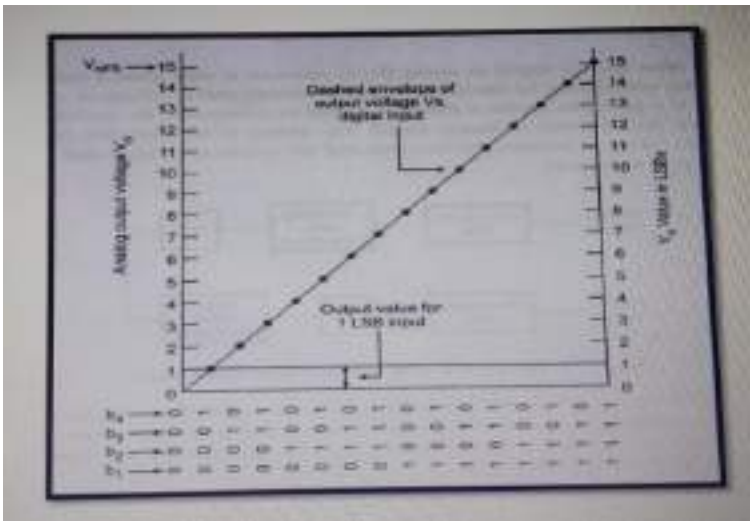
OUT03

MVIA,DATA;Load8-bitdatatobesentattheinputof0808DAC

OUT00; senddataonport A.

ACircuitDescriptionofDACmodule:

- WhenchipselectofDACisenabledthenDACwillconvertdigitalinputvaluegivent hrough portlinessPB0-PB7toanalogvalue.
- TheanalogoutputfromDACisacurrentquantity.Thiscurrentisconvertedtovoltag eusingOPAMPbasedcurrent-to-voltage converter.
- Thevoltage outputs(+/-8Vforbipolar0to8Vforunipolarmode)ofOPAMPare connected to CRO to see the wave form. Port A & Port B are connected tochannel1 and channel2respectively.
- Areferencevoltageof 8Visgeneratedusing723andisgiventoVerifypointsofthe DAC 0800. The standard output voltage will be 7.98V when FF is outputtedandwillbe 0V when00 isoutputted.
- The Output of DAC-0800 is fed to the operational amplifier to get the finaloutputasX OUTandYOUT.



Figureshowsanalogoutputvoltage v0isplottedagainstall16possibledigitalinputwords.

PerformanceParametersofDAC:

TheperformanceparametersofDACare:

1. **<u>Resolution</u>**:

- Resolutionisdefinedintwoways.
  - Resolutionisthenumberofdifferentanalogoutput valuesthatcanbe providedbya DAC.

- Forann-bitDAC**Resolution=2^n……..(1)**

  or
- Resolutionisalsodefinedastheratioofachangeinoutputvoltageresultingfroma changeof 1LSBat the digitalinputs.
- Forann-bit DACitcanbegivenas:**Resolution=$V_oF_s/2^n$-1……..(2)**
- Where,**V_oF_s**=Fullscaleoutputvoltagefromequation(1),wecansaythat,theresoluti oncanbedeterminedbythenumberofbitsintheinputbinaryword.
- Foran8-bitresolutioncanbegivenasresolution=2^n=2^8=256
- If the full scale output voltage is 10.2 V then by second definition the resolutionfor an 8-bit can be given as Resolution= = **V_o F_s /2^n -1 =10.2/2^8-1 =10.2/255 =40mV/LSB**
- Therefore,wecansaythataninputchangeof1LSBcausestheoutputtochange by40 mv

2. **<u>Accuracy</u>**:
   - lt is a comparison of actual output voltage with expected output. It isexpressedinpercentage.Ideally,theaccuracyofDACshouldbe,atworst, ±1/2,ofitsLSB.
   - Ifthefullscaleoutputvoltageis10.2Vthen foran8-bitDACaccuracycanbegivenas **Accuracy=VoFs /(2^n-1)2=10.2/255x2=20mV**

ADCCONVERTER:

- Itisaconverterwhichconvertsanalogquantityintodigitalquantity.
- TherearemanytypesofADCavailablesuchas
  1. RAMPtypeADC
  2. DualslopeADC
  3. FlashtypeADC
  4. SuccessiveapproximationtypeADC
- Mostlythesuccessiveapproximationtype ADCareused.

SPECIFICATIONOF ADC:

<u>Inputvoltagerange:</u>

- Theanaloginputcanbeeitherunipolarorbipolar.
- Unipolarmeansthevoltagehaveonepolarityi.e.(0to+5Vor-5Vto0).

- Bipolarmeansthevoltagehavetherangefromonepolaritytootherpolarity i.e.(+5Vto-5Vor -10Vto+10V)

Outputvoltagerange:

- TheresolutionofADCisdefinedasthesmallestchangeinputvoltagecanbesense dordetectedatthe output.Which isgivenby
  Resolution= range of input voltage
  rangeofoutput voltage

  Example→iftheinput voltagerangefrom0to+5Vandoutputhas8bitthentheresolution=5/2^8=19.5mv

Conversiontime:
Itisthetimerequiredtoconverttheanaloginputintodigital outputbyADCchipisknownas conversiontime.

ExampleofADCchip:

- ADC0800IC
- ADC0804IC
- ADC0808IC
- ADC0816IC

ADCINTERFACING:
The Analog to Digital Conversion is a quantizing process. Here the analog signal isrepresentedbyequivalentbinarystates.TheA/Dconverterscanbeclassifiedintotwo groupsbasedontheirconversiontechniques.

- Inthefirsttechniqueitcomparesgivenanalogsignalwiththeinitiallygene rated equivalent signal. In this technique, it includes successiveapproximation,counter andflashtype converters.
- Inanothertechniqueitdeterminesthechangingofanalogsignalsintotimeorfrequ ency. This process includes integrator-converters and voltage-to-frequencyconverters.
- The first process is faster but less accurate, the second one is more accurate. Asthe first process uses flash type, so it is expensive and difficult to design forhighaccuracy.

TheADC0808/0809Chip

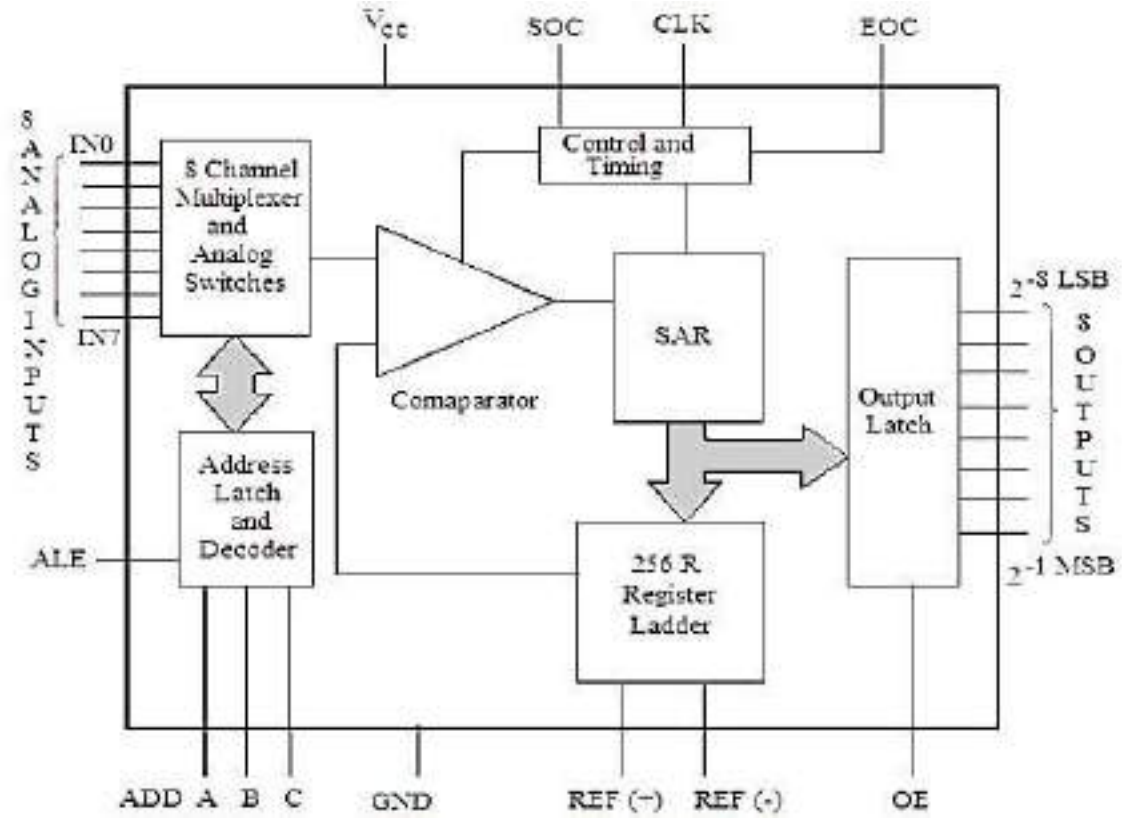
- TheADC0808/0809isan8-bitanalogtodigitalconverter.
- Ithas8channelmultiplexertointerfacewiththemicroprocessor.
- ThischipispopularandwidelyusedADC.
- ADC0808/0809isamonolithicCMOSdevice.Thisdeviceusessuccessiveappr oximationtechniquetoconvertanalogsignaltodigitalform.
- Oneofthemainadvantageofthischipisthatitdoesnotrequireanyexternalzeroan dfullscaleadjustment,only+5VDCsupplyissufficient.

LetusseesomegoodfeaturesofADC0808/0809

- Theconversionspeedismuchhigher
- Theaccuracyisalsohigh
- Ithasminimaltemperaturedependence
- Excellentlongtermaccuracyandrepeatability

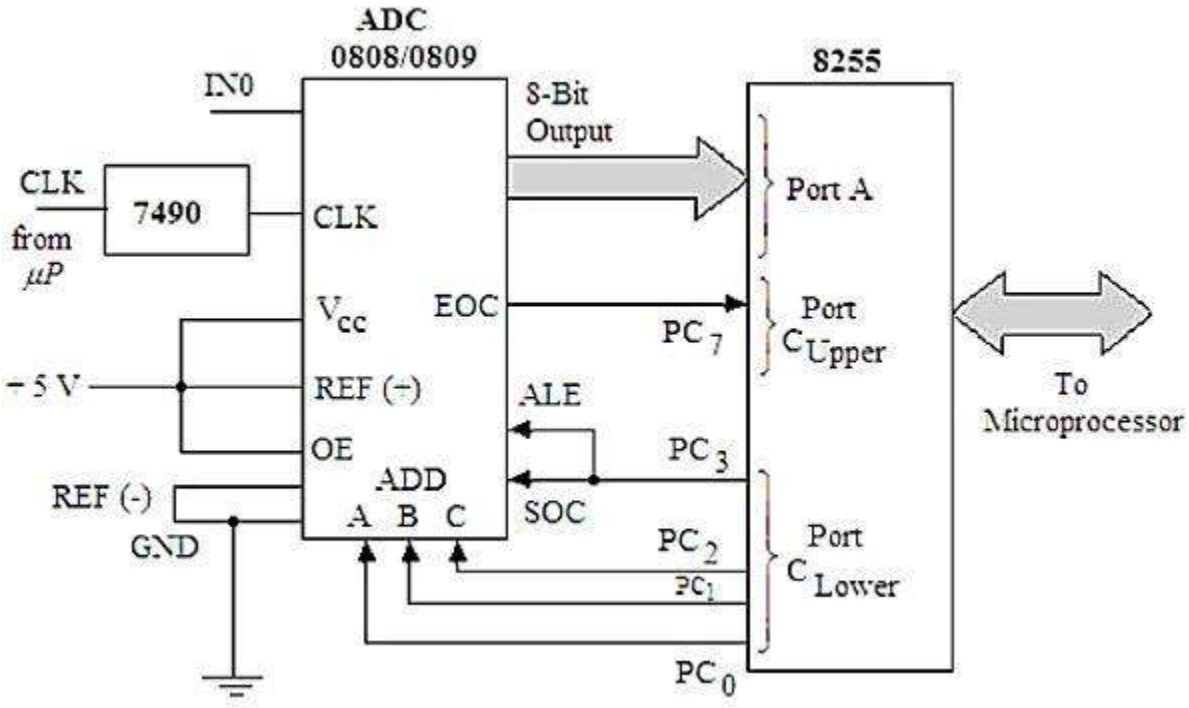
- Lesspowerconsumption

Thefunctionalblockdiagramofthischipislikethis



InterfacingADCwith8085Microprocessor:

TointerfacetheADCwith8085,weneed8255ProgrammablePeripheralInterfacechip with it. Let us see the circuit diagram of connecting 8085, 8255 and the ADCconverter.



- ThePort Aof8255chipisusedastheinputport.ThePC_7pin ofPort C_{upper}isconnected to the End of Conversion (EOC) Pin of the analog to digitalconverter.Thisportisalsousedasinputport.
- TheC_{lower}portisusedasoutputport.ThePC_{2-0}linesareconnectedtothreeaddresspinsof thischiptoselect inputchannels.
- The PC_3 pin is connected to the Start of Conversion (SOC) pin and ALE pin ofADC0808/0809.

Nowletusseeaprogramtogeneratedigitalsignalfromanalogdata.WeareusingIN0asinput pin,sothe pinselectionvaluewillbe 00H.

MVIA,98H;SetPort AandCupperasinput,C_{Lower}asoutput

OUT03H;Writecontrolword8255-ItocontrolWord register

XRAA;Cleartheaccumulator

OUT02H;sendthecontentofaccumulatortoPortC_{lower}toselect

IN0

MVIA,08H;Loadtheaccumulatorwith08H

OUT02H;ALEandSOCwillbe0

XRAA;Cleartheaccumulator

OUT02H;ALEandSOCwillbelow.

READ:IN02H;ReadfromEOC(PC7)

RAL: RotatelefttocheckC7is

1.**JNCREAD;**ifC7isnot1,gotoREAD**IN**

00H: Read digital output of

ADC**STA 8000H:** Save result at

8000H**HLT:**Stoptheprogram

SEVENSEGMENTDISPLAY:

- A seven-segment display is a form of electronic display device for displayingdecimalnumeralsthatisanalternativetothemorecomplexdotmatrixdisplays.
- Seven-segment displays are widely used in digital clocks, electronic meters, basiccalculators,andotherelectronicdevicesthatdisplaynumericalinformation.
- The binary information can be displayed in the form of decimal using this sevensegmentdisplay.Itswiderangeofapplicationsisinmicrowaveovens,calculators,washingmachines,radios,digitalclocksetc.

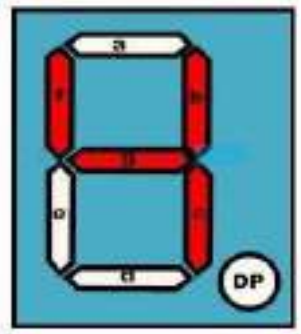
- The seven segment displays are made up of either LEDs (Light emitting diode) orLCDs (Liquid crystal display). LED or light emitting diode is P-N junction diodewhich emits the energy in the form of light, differing from normal P-N junctiondiodewhich emitsinthe form of heat.

- Liquid crystal displays (LCD) use the properties of liquid crystal for displaying.LCD will not emit the light directly. These LED's or LCD are used to display therequirednumeraloralphabet.Singlesevensegmentornumberofsegmentsarrangedinanordermeetsour requirements.

WorkingofSevenSegmentDisplay:

- Seven segment display works, by glowing the required respective LEDS in thenumeral. The display is controlled using pins that are left freely. Forwardbiasing of these pins in a sequence will display the particular numeral oralphabet.Dependingonthetypeofsevensegmentthesegmentpinsareappliedwithlogichighorlogiczeroandinthesimilarwaytothecommonpinsalso.

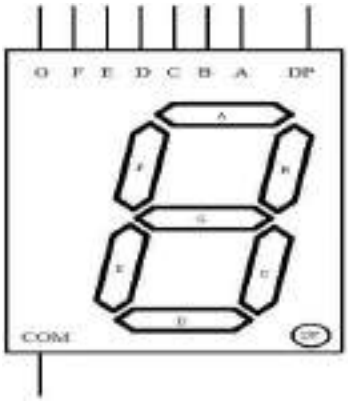
- For example to display numeral '1' segments b and c are to be switched on andtheremainingsegmentsarerequiredtobeswitchedoff.Inorder todisplaytwodigitstwosevensegmentsare used.



- Dependingoneitherthecommonpinisanodeorcathode,sevensegmentsarediviedintofollowingtypes.

INTERFACINGSEVENSEGMENTDISPLAY:

- Seven Segment Display Interfacing are generally used as numerical indicatorsandconsistsofanumberofLEDsarranged insevensegmentsasshownintheFigure



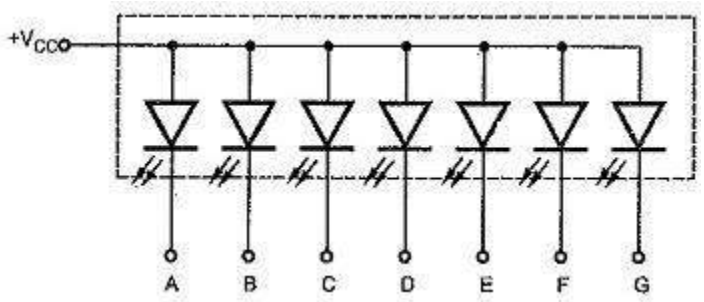
SEVENSEGMENTDISPLAY

- Anynumber between0and9canbeindicatedbylightingtheappropriatesegments.The 7-segmentdisplaysareof twotypes:

1. Commonanodetype
2. Commoncathodetype.

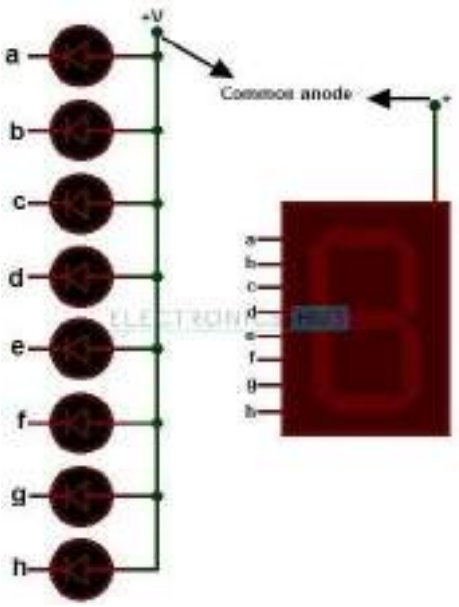
CommonAnodetype:

- IncommonanodeallanodesofLEDsareconnectedtogetherasshowninFig.



Commonanodetype

or



- IncommonanodetypealltheanodesofLED'sareconnectedtothecommonterminal and cathodes are left free. Thus, in order to glow the LED, thesecathodeshavetobeconnectedtothelogic'0'andanodetothelogic'1'.

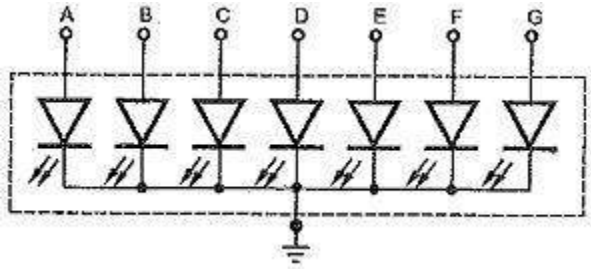
- Belowtruthtablegivestheinformationrequiredfordrivingthecommonanodesevensegments.

Segments Inputs							7 Segment Display Output
a	b	c	d	e	f	g	
0	0	0	0	0	0	1	0
1	0	0	1	1	1	1	1
0	0	1	0	0	1	0	2
0	0	0	0	1	1	0	3
1	0	0	1	1	0	0	4
0	1	0	0	1	0	0	5
0	1	0	0	0	0	0	6
0	0	0	1	1	1	1	7
0	0	0	0	0	0	0	8
0	0	0	0	1	1	0	9

- Inordertodisplayzeroonthissegmentoneshouldenablelogic highona,b,c,d, e and f segments and logic low on segment 'g'. Thus, the above tableprovidesdata onsevensegmentsfordisplayingnumeralsfrom0-9.

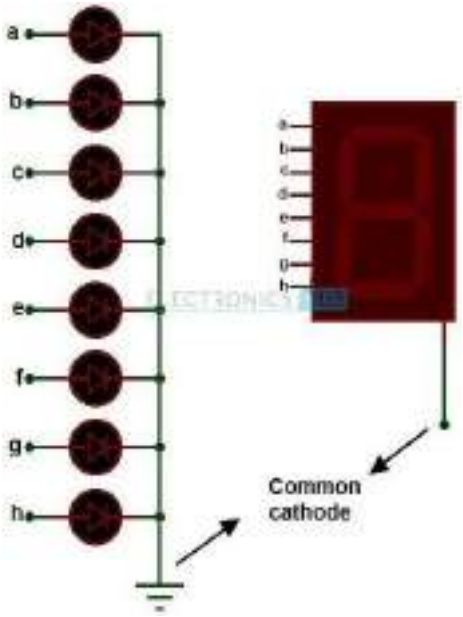
Commoncathodetype:

- As the name indicates cathode is the common pin for this type of sevensegmentsandremaining8pinsareleftfree.Here,logiclowisappliedtothecommonpinandlogichigh totheremainingpins.
- incommoncathode,allcathodesareconnectedtogether,asshowninFig



Commoncathodetype

Or



Thetruthtableofsevensegmentdisplayisshownbelow.

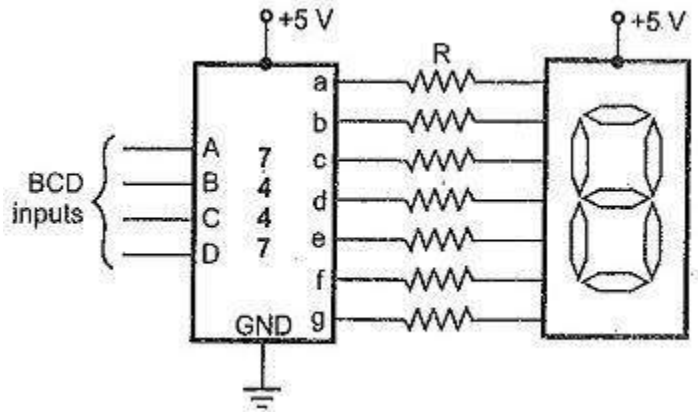
Segments Inputs							7 Segment Display Output
a	b	c	d	e	f	g	
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	0	1	9

- Abovetruthtableshowsthedatatobeappliedtothesevensegmentsto displaythedigits.Inordertodisplaydigit‘0’ onsevensegment,segmentsa,b ,c,d,eandfareappliedwithlogichighandsegmentgisappliedwithlogiclow.

DrivingaSevenSegmentDisplay:

- Itshowsacircuittodriveasingle,SevenSegmentDisplayInterfacing,com monanode LEDdisplay.

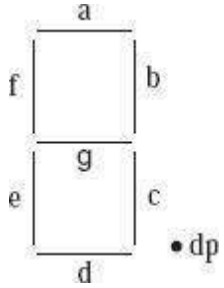
- Forcommonanode,whenanodeisconnectedtopositivesupply,alowvoltageisappl ied toa cathode toturniton.
- Here,BCDtosevensegmentdecoder,IC7447isusedtoapplylowvoltagesatcathod esaccordingtoBCDinputappliedto7447.
- TolimitthecurrentthroughLEDsegmentsresistorsareconnectedinserieswitht hesegments.
- Thiscircuitconnectionisreferredtoasastaticdisplaybecausecurrentisbeing passed throughthedisplayat alltimes.

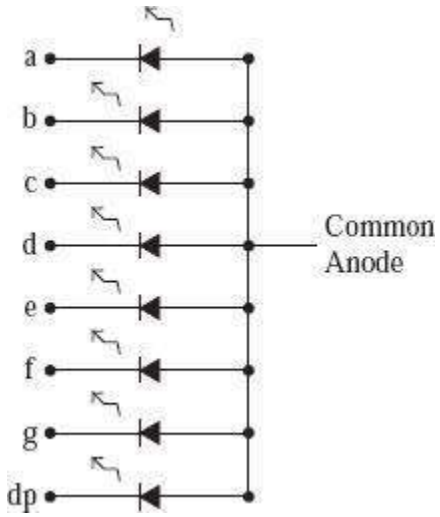


CircuitfordrivingsinglesevensegmentLEDdisplayINTERFAC

ING:

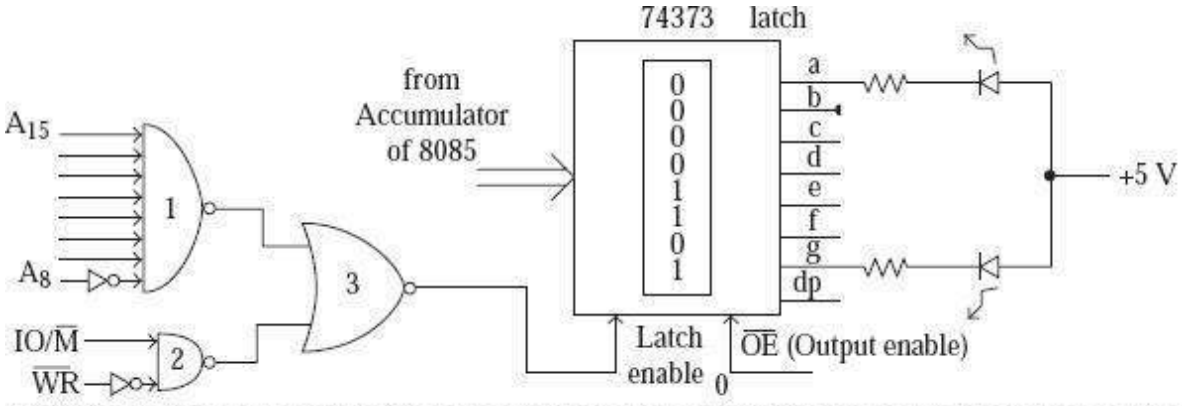
- Anoutputdevicewhichisverycommonis, especiallyinthekitof8085microprocessor and it is the Light Emitting Diode consisting of sevensegments.
- Moreover,wehaveeightsegments inaLEDdisplayconsistingof7segmentswhich,consistofcharacter8andhavinga decimalpointjustnexttoit.
- Wedenotethesegmentsas‘a,b,c,d,e,f, g,anddp’wheredpsignifies‘.’which isthedecimalpoint.
- Moreover,theseareLEDsortogetheraseriesofLightEmittingDiodes.Wehaves howntheinternalcircuitcomprisingofadisplayofseven segment





- Wehavediscussedthecommonanode-typewhichis7segmentedLightEmittingDiode.
- In the LED which is common anode and is 7-segmented, here we connect alltheeightLEDanodestogetherandtheeightexternalpinisbroughttodisplay.
- AndthispingetsconnectedtoaDCsupplyof+5Volt.
- Thecathodeendsoftheeightsegmentsarebroughtoutonthepinsofthedisplay.

The use of 74373 latch for interfacing a 7-segment display is shown in thefollowingFig.



Note: To keep the figure simple, only two of the eight segment connections are shown. The other six segment connections are similar.

- Inthe74373latchisusedasanI/OmappedI/Oportwiththeportaddressas FEH.
- Thiscouldbeeasilyverifiedfromthechipselectcircuitusedinthefigure.
- The following instructions are to be executed to display character '3' on the 7-segmentdisplay.
- Thecorrespondingprogramtosend0DHtotheportFEHwillbe-

MVIA,0DH

OUTFEH

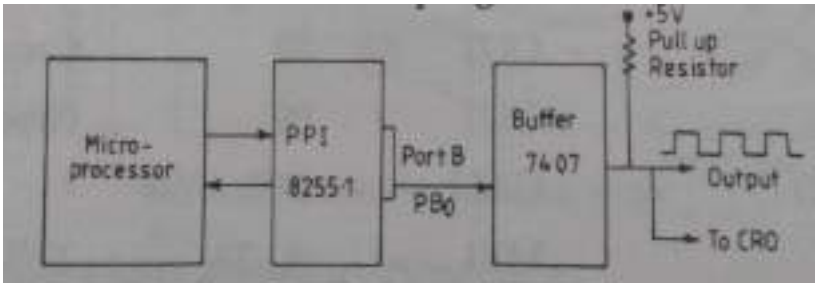
- UsingMVIinstructionweareinitializingAccumulator(A)withByte0DHi.e.0000 1101.
- ThenitwillbesenttotheportFEHbytheinstructionOUT.

GENERATESQUAREWAVESONALLLINESOF8255:

- Asquarewaveorpulsecaneasilybegeneratedbymicroprocessor.
- Themicroprocessors sendshighandthenlowsignalstogeneratesquarewaveorpulse.
- ApulseorsquarewavecanbegeneratedusingI/OportorSODlineortimer/ counter (Intel8253).

TogeneratesquarewaveorpulseusingI/Oport:

- Togeneratesquarewaveconnectionsaremadeasshowninfigurebelow.



Togeneratesquarewaveusingmicroprocessor

- The pin PB_0 of the port B is used for taking output. This is connected to abuffer 7407.Thefinaloutputistakenfromthebufferterminal.
- The8255hasbeendesignedasgeneralpurposeprogrammableI/Odevices,com patiblewith Intelmicroprocessor.

- It contain three 8-bit port which can be configured by software means toproveanyoneof3programmabledatatransfermodesavailablewith8255.
- Thecontrolwordusedintheprogramis98HtomakeportBanoutputport.

PROGRAM:

MEMORY ADDRESS	**MACHINE CODES**	**LABLES**	**MNEMONICS**	**OPERANDS**	**COMMENTS**
2400	3E,98		MVI	A,98H	Getcontrolword.
2402	D3,0B		OUT	08	Initializeports.
2404	3E,00	LOOP	MVI	A,00	Move‘00’into accumulator
2406	D3,09		OUT	09	MakePB_0LOW
2408	CD,00,25		CALL	DELAY1	CalltheDELAY1subroutine.
240B	3E,01		MVI	A,01	Move‘01’into accumulator
240D	D3,09		OUT	09	MakePB_0HIGH
240F	CD,09,25		CALL	DELAY2	CalltheDELAY2subroutine.
2412	C3,04,24		JMP	LOOP	JumptoLOOP.
SUBROUTINES DELAY1					
2500	06,02		MVI	B,02	Getcountfordelay.
2502	05	GO	DCR	B	Decrement registerBby1.
2503	C3,02,25		JNZ	GO	IsB=0?No,thenjumptoGO.
2506	C9		RET		
DELAY2					
2509	0E,02		MVI	C,02	Getcountfordelay
250B	0D	BACK	DCR	C	Decrementregister Cby1.
250C	C2,0B,25		JNZ	BACK	IsC=0?No,thengoto BACK.
250F	C9		RET		

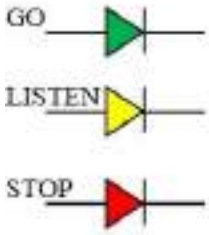
DESIGNINTERFACEATRAFFICLIGHTCONTROLSYSTEMUSING8255:

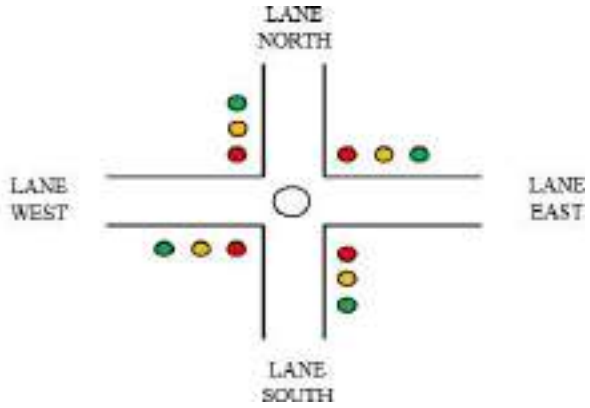
TRAFFICLIGHTCONTROL

Traffic lights, which may also be known as stoplights, traffic lamps,trafficsignals, signal lights, robots or semaphore, are signaling devices positioned at roadintersections, pedestrian crossings and other locations to control competing flows oftraffic.

ABOUTTHECOLORSOFTRAFFICLIGHTCONTROL

- Traffic lights alternate the right of way of road users by displaying lights of astandard color (red, yellow/amber, and green), using a universal color code (andaprecisesequencetoenablecomprehensionbythosewhoarecolor blind).
- Illumination of the red signal prohibits any traffic from proceeding. Usually, thered light contains some orange in its hue, and the green light contains some blue,for the benefit of people with red-green color blindness, and "green" lights in manyareasareinfactbluelensesonayellowlight(whichtogetherappeargreen).

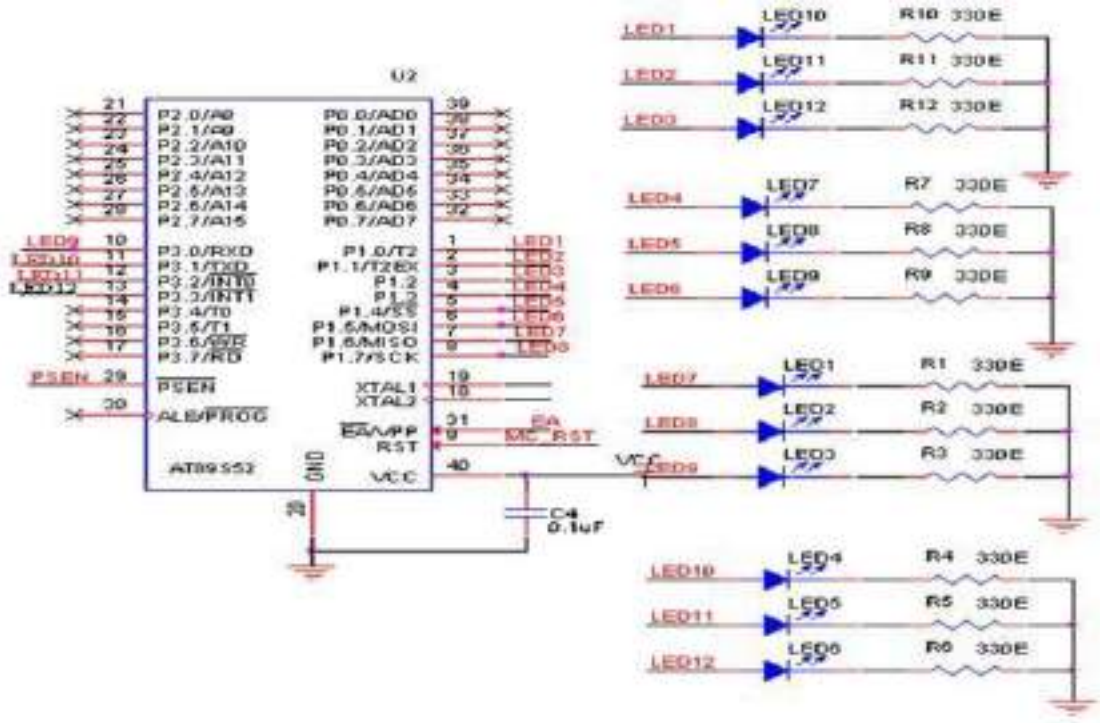




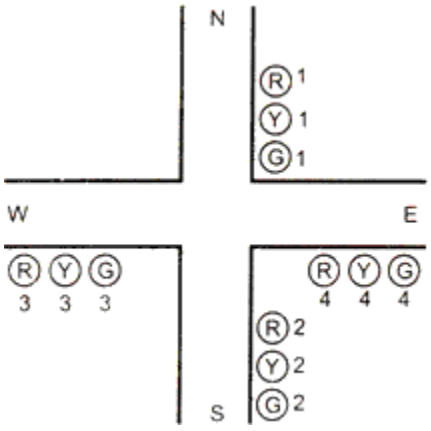
INTERFACINGTRAFFICLIGHTWITH8085

The Traffic light controller section consists of 12 Nos. point LEDS are arranged by4LanesinTrafficlight interfacecard.EachlanehasGo(Green),Listen(Yellow)andStop(Red)LEDisbeingplaced.

CIRCUITDIAGRAMTOINTERFACETRAFFICLIGHTWITH8085



HARDWAREFORTRAFFICLIGHTCONTROL



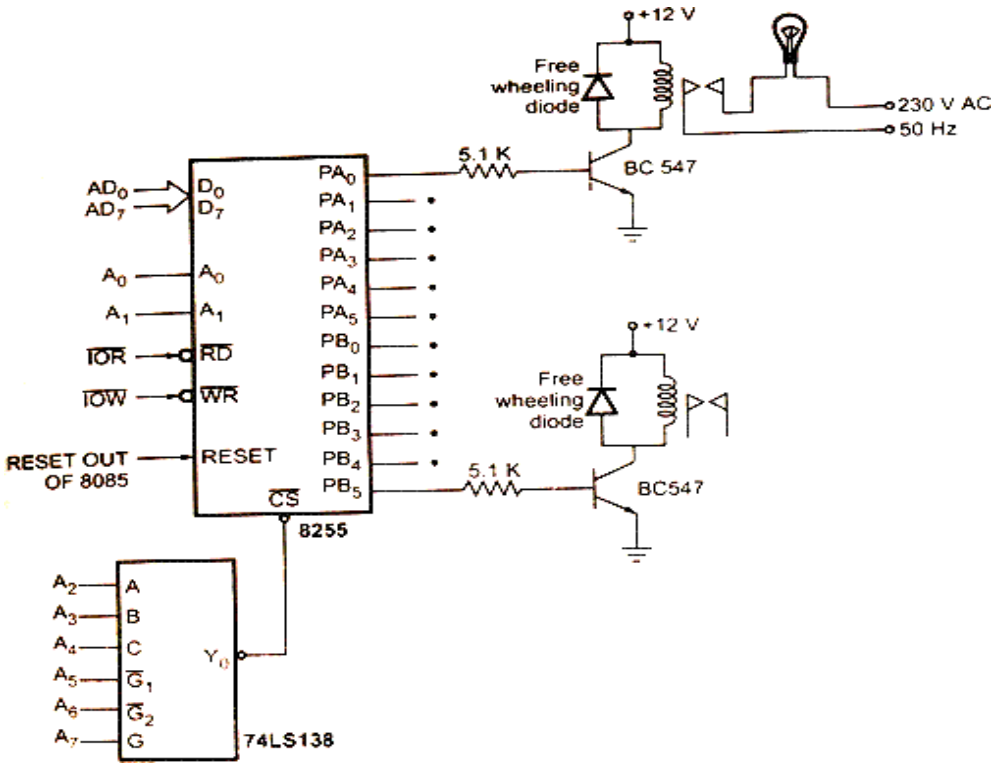
- The interfacing diagram to control 12 electric bulbs. Port A is used to controllights on N-S road and Port B is used to control lights on W-E road. Actual pinconnectionsare listedinTable 1 below.

Pins	Light	Pins	Light
PA_0	R_1	PB_0	R_3
PA_1	Y_1	PB_1	Y_3
PA_2	G_1	PB_2	G_3
PA_3	R_2	PB_3	R_4
PA_4	Y_2	PB_4	Y_4
PA_5	G_2	PB_5	G_4

Table 1

- The electric bulbs are controlled by relays. The 8255 pins are used to controlrelay on-off action with the help of relay driver circuits. The driver circuitincludes12transistorstodrive12relays.Fig.alsoshowstheinterfacingof8255

INTERFACINGDIAGRAM:



I/O MAP:

Ports / Control Register	Address lines								Address
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Port A	1	0	0	0	0	0	0	0	80H
Port B	1	0	0	0	0	0	0	1	81H
Port C	1	0	0	0	0	0	1	0	82H
Control Register	1	0	0	0	0	0	1	1	83H

Table 2

SOFTWAREFORTRAFFICLIGHTCONTROL:

Control word : For initialization of 8255.

BSR/IO	$MODE_A$		P_A	PC_H	MODE B	P_B	PC_L
1	0	0	0	X	0	0	X

= 80H

Fig. Control word

Table shows the data bytes to be sent for specific combinations.

To glow	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0	Port B Output	Port A Output
R_1,R_2,G_3 and G_4	X	X	1	0	0	1	0	0	X	X	0	0	1	0	0	1	24H	09H
Y_1,Y_2,Y_3 and Y_4	X	X	0	1	0	0	1	0	X	X	0	1	0	0	1	0	12H	12H
R_3,R_4,G_1 and G_2	X	X	0	0	1	0	0	1	X	X	1	0	0	1	0	0	09H	24H

PROGRAM:

```
MVIA,80H            :Initialize8255,portAandportB
OUT83H(CR)
                    :inoutputmodeS
TART:MVIA,09H
OUT80H(PA)
                    :SenddataonPAtoglowR1andR2MVI
A,24H
OUT81H(PB)          : Send data on PB to glow G3 and
G4MVIC,28H          :Loadmultipliercount(40ıo)fordelay
CALLDELAY
                     :Calldelaysubroutine
MVIA,12H
OUT(81H)PA
                    :SenddataonPortAtoglowY1andY2OUT(
81H)PB               : Send data on port B to glow Y3 and
Y4MVIC,0AH          :Loadmultipliercount(10ıo)fordelay
CALL:DELAY           : Call delay
subroutineMVIA,24H
OUT(80H)PA
                     :SenddataonportAtoglowG1andG2MVI
A,09H
OUT(81H)PB
                    :SenddataonportBtoglowR3andR4MVIC,
28H                 :Loadmultipliercount(40ıo)fordelay
```

```
CALLDELAY                : Call delay
subroutineMVIA,12H
OUTPA                    :Senddataonport AtoglowY1andY2
```

```
OUTPB          :SenddataonportBtoglowY3andY4
MVIC,0AH       :Loadmultipliercount(10ıo)fordelay
CALLDELAY
                :CalldelaysubroutineJ
MPSTART
```

DelaySubroutine:

```
DELAY:LXID,Count
                :Loadcounttogive0.5secdelayBAC
K:DCXD          :Decrementcounter
MOVA, D
ORAE            :Checkwhethercountis0
JNZBACK         :Ifnotzero,repeat
DCRC
                :CheckifmultiplierzerootherwiserepeatJN
ZDELAY
RET             :Returntomainprogram
```

DESIGN INTERFACE FOR STEPPER MOTOR CONTROL USING

8255:<u>STEPPERMOTOR:</u>

- Asteppermotorisadevicethattranslateselectricalpulsesintomechanicalmove mentinstepsoffixedstep angle.
  - ✓ Thesteppermotorrotatesinstepsinresponsetotheappliedsignals.
  - ✓ Itismainlyusedforpositioncontrol.
  - ✓ Itisusedindiskdrives,dotmatrixprinters,plottersandroboticsandprocesscontro lcircuits.

<u>Structure:</u>

- Steppermotors haveapermanentmagnetcalledrotor(alsocalledtheshaft)surroundedbya stator.
- The most common stepper motors have four stator windings that are paired withacenter-tap.
- This typeofsteppermotor is commonly referred to as a four-phase orunipolarstepper motor.
- Thecentertapallowsachangeofcurrentdirectionineachoftwocoilswhenawindingis grounded, therebyresultinginapolaritychangeofthestator.

<u>Interfacing:</u>

- Evenasmallsteppermotorrequireacurrentof400mAforitsoperation.Buttheport softhemicrocontrollercannotsourcethismuchamountofcurrent.
- Ifsuchamotorisdirectlyconnectedtothemicroprocessor/microcontrollerports,t hemotormaydraw largecurrentfromtheportsanddamageit.

- Soasuitabledrivercircuitisusedwiththemicroprocessor/microcontrollertooperatethe motor.

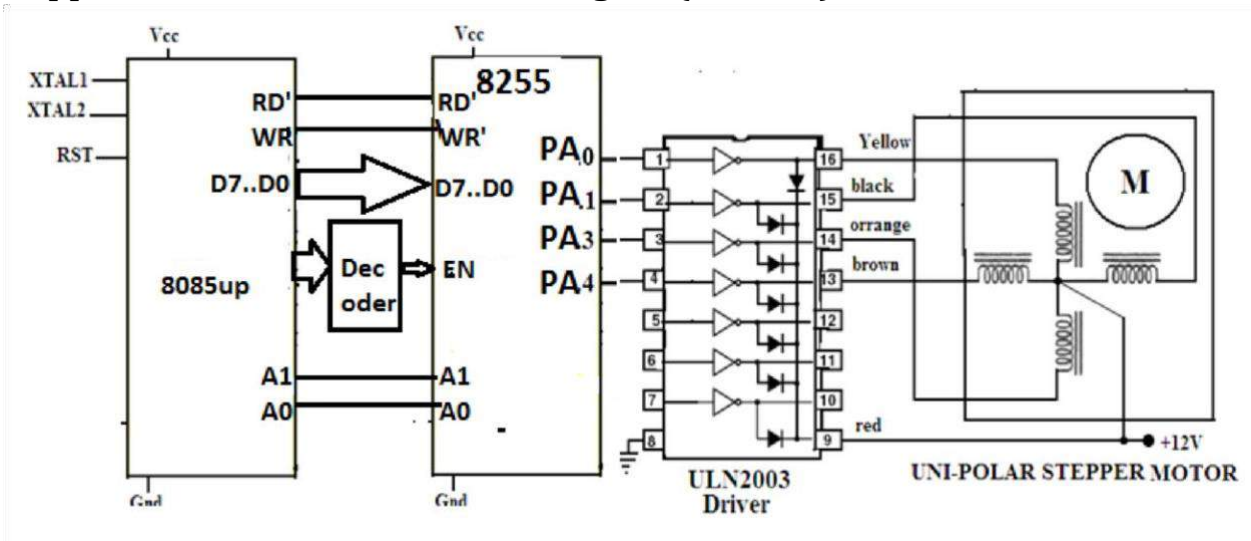
MotorDriverCircuit(ULN2003)

- SteppermotordrivercircuitsareavailablereadilyintheformofICs.
- ULN2003isonesuchdriverICwhichisaHigh-VoltageHigh-CurrentDarlingtontransistor arrayandcangive acurrent of500mA.
- This current is sufficient to drive a small stepper motor. Internally, it hasprotectiondiodesusedtoprotectthemotorfromdamageduetobackemfandlargeeddycurrents.
- So, this ULN2003 is used as a driver to interface the stepper motor to themicrocontroller.

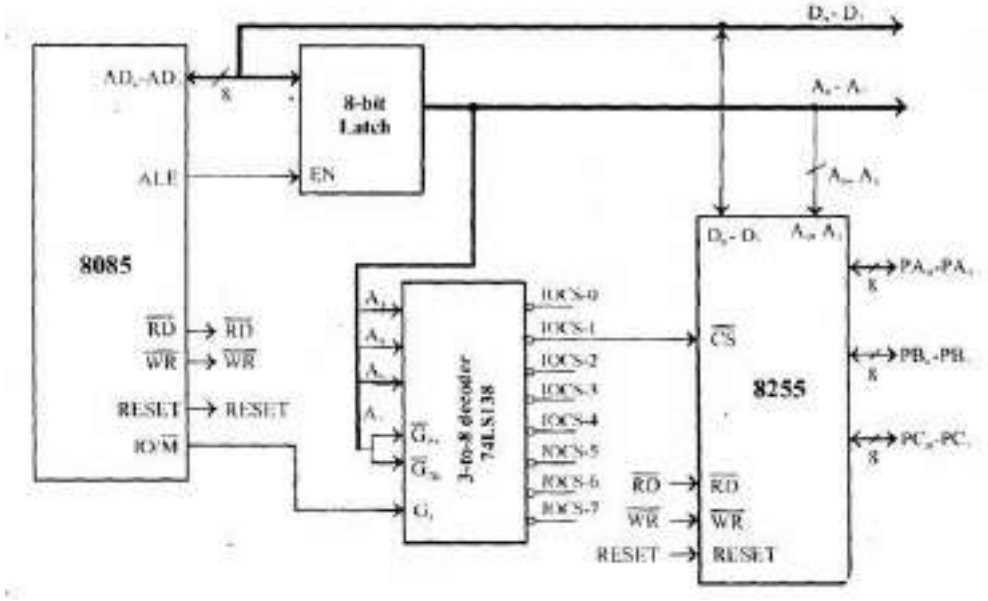
Operation:

- Theimportantparameterofasteppermotoristhe**stepangle**.
- Itistheminimumanglethroughwhichthemotorrotatesinresponsetoeach **excitationpulse**.
- Inafourphasemotoriftherearе200stepsinonecompleterotationthenthenthe step angle is360/200= 1.8^{0} .
- Sotorotatethesteppermotorwehavetoapplytheexcitationpulse.Forthisthecontrollershouldsendahexadecimalcodethroughoneofitsports.
- **Thehexcodemainlydependsontheconstructionofthesteppermotor**.So, all the stepper motors do not have the same Hex code for their rotation.(refer theoperationmanualsuppliedbythemanufacturer.)
- For example, let us consider the hex code for a stepper motor to rotate inclockwisedirectionis77H,BBH,DDHandEEH.Thishexcodewillbeappliedto the input terminals of the driver through the assembly language program.To rotate the stepper motor in anti-clockwise direction the same code isappliedinthe reverseorder.

StepperMotorinterface-SchematicDiagram(for8085):



DetailedConnectiondiagrambetween8085and8255:



ASSEMBLYLANGUAGEPROGRAM(8085)

Main :MVIA,80	; 80H→ControlwordtoconfigurePA,PB,PCin O/P
OUTCWR_Address	;WritecontrolwordinCWRof8255
MVIA,77	;Codefor thePhase1
OUTPortA_Address	;senttomotor viaport Aof8255 ;
CALLDELAY	;Delaysubroutine

```
MVIA,BB                 ;CodeforthePhaseII
OUTPortA_Address        ;senttomotor viaport Aof8255
CALLDELAY               ;Delaysubroutine.
MVIA,DD                 ;CodeforthePhaseIII
OUTPortA_Address        ;senttomotor viaport Aof8255;
CALLDELAY               ;Delaysubroutine
MVIA,EEH                ;CodeforthePhase1
OUTPortA_Address        ;senttomotor viaport Aof8255 ;
CALLDELAY               ;Delaysubroutine
JMPMAIN                 ;Keepthemotorrotatingcontinuously.
```

DELAYSubroutine

```
MVIC,FF                 ;Load C with FF-- Change it for the speedvariation

LOOP1:MVID,FF           ;Load Dwith FF
LOOP2:DCRD
JNZLOOP2
DCRC
JNZLOOP1
RET                     ;Returntomainprogram.
```

DMACONTROLLER:(8257)

- DMAstandsforDirectMemoryAccess.ItisdesignedbyInteltotransferdataatthe fastest rate. It allows the device to transfer the data directly to/frommemorywithout any interference of theCPU.
- UsingaDMAcontroller,thedevicerequeststheCPUtoholditsdata,addressand control bus, so the device is free to transfer data directly to/from thememory. The DMA data transfer is initiated only after receiving HLDA signalfromthe CPU.

HowDMAOperationsareperformed?

FollowingisthesequenceofoperationsperformedbyaDMA–

- Initially, when any device has to send data between the device and thememory,thedevicehastosendDMArequest(DRQ)toDMAcontroller.

- TheDMAcontrollersendsHoldrequest(HRQ)totheCPUandwaitsfortheCPUto assert the HLDA.
- Then the microprocessor tri-states all the data bus, address bus, and controlbus.TheCPUleavesthecontroloverbusandacknowledgestheHOLDrequestthroughHLDAsignal.
- NowtheCPUisinHOLDstateandtheDMAcontrollerhastomanagetheoperationsoverbusesbetweentheCPU,memory,andI/Odevices.

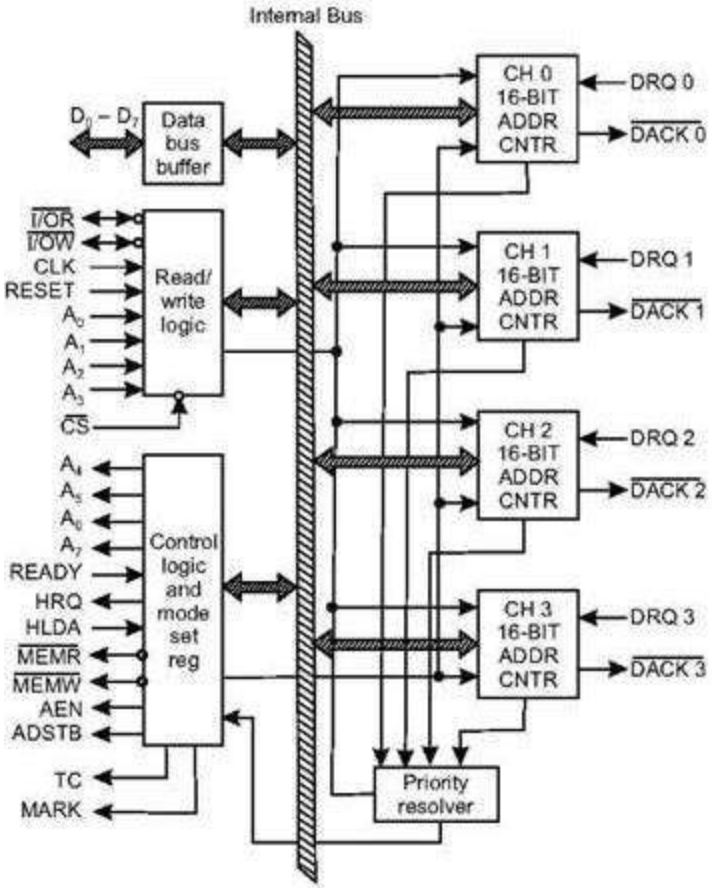
Featuresof8257:

Hereisalistofsomeoftheprominentfeaturesof8257–

- Ithasfourchannelswhichcanbeusedoverfour I/Odevices.
- Eachchannelhas16-bitaddressand14-bitcounter.
- Eachchannelcantransferdataupto64kb.
- Eachchannelcanbeprogrammedindependently.
- Eachchannelcanperformreadtransfer,writetransferandverifytransferoperations.
  - ItgeneratesMARKsignaltotheperipheraldevicethat128byteshavebeentransferred.
  - Itrequiresasinglephaseclock.
  - Itsfrequencyrangesfrom250Hzto3MHz.
  - Itoperatesin2modes,i.e.,**Mastermode**and**Slavemode**.

8257Architecture:

Thefollowingimageshowsthearchitectureof8257–



8257PinDescription:

Thefollowingimageshowsthepindiagram ofan8257DMAcontroller–

DRQ_0–DRQ_3

Thesearethefourindividualchannel DMArequestinputs,whichareusedbytheperipheral devices for using DMA services. When the fixed priority mode isselected, then DRQ_0has the highest priority and DRQ_3has the lowest priorityamongthem.

$DACK_o$–$DACK_3$

These are the active-low DMA acknowledge lines, which updates the requestingperipheralaboutthestatusoftheirrequestbytheCPU.Theselinescanalsoact asstrobelinesforthe requestingdevices.

D_o–D_7

Thesearebidirectional,datalineswhichareusedtointerfacethesystembuswiththe internal data bus of DMA controller. In the Slave mode, it carries commandwords to 8257 and status word from 8257. In the master mode, these lines areused to send higher byte of the generated address to the latch. This address isfurtherlatched usingADSTBsignal.

IOR

It is an active-low bidirectional tri-state input line, which is used by the CPU to readinternal registers of 8257 in the Slave mode. In the master mode, it is used to readdatafrom theperipheraldevicesduringamemorywritecycle.

IOW

It is an active low bi-direction tri-state line, which is used to load the contents ofthe data bus to the 8-bit mode register or upper/lower byte of a 16-bit DMAaddressregisterorterminalcountregister.Inthemastermode,itisusedtoloadthed atatotheperipheraldevicesduring DMAmemoryreadcycle.

CLK

Itisaclockfrequencysignalwhichisrequiredfortheinternaloperationof8257.

RESET

ThissignalisusedtoRESETtheDMAcontrollerbydisablingalltheDMAchannels.

A_o- A_3

These are the four least significant address lines. In the slave mode, they act as aninput, which selects one of the registers to be read or written. In the master mode,theyarethefourleastsignificantmemoryaddressoutputlinesgeneratedby8257.

CS

It is an active-low chip select line. In the Slave mode, it enables the read/writeoperationsto/from8257.Inthemastermode,itdisablestheread/writeoperation sto/from8257.

A_4-A_7

Thesearethehighernibbleofthelowerbyteaddressgeneratedby DMAinthemastermode.

READY

Itisanactive-highasynchronousinputsignal,whichmakesDMAreadybyinsertingwait states.

HRQ

Thissignalisusedtoreceivetheholdrequestsignalfromtheoutputdevice.Intheslave mode, it is connected with a DRQ input line 8257. In Master mode, it isconnectedwith HOLD inputof the CPU.

HLDA

ItistheholdacknowledgementsignalwhichindicatestheDMAcontrollerthatthebushas beengrantedtotherequestingperipheralbytheCPUwhenit issetto1.

MEMR

It is the low memory read signal, which is used to read the data from the addressedmemorylocationsduringDMAread cycles.

MEMW

Itistheactive-lowthreestatesignalwhichisusedtowritethedatatotheaddressedmemorylocation duringDMAwriteoperation.

ADST

ThissignalisusedtoconvertthehigherbyteofthememoryaddressgeneratedbytheDMA controller intothe latches.

AEN

Thissignalisusedtodisabletheaddressbus/databus.

TC

Itstandsfor'TerminalCount',whichindicatesthepresentDMAcycletothepresent peripheraldevices.

MARK

Themark willbeactivatedaftereach128cyclesorintegralmultiplesofitfromthebeginning. It indicates the current DMA cycle is the 128th cycle since the previousMARKoutput totheselected peripheraldevice.

V_{cc}

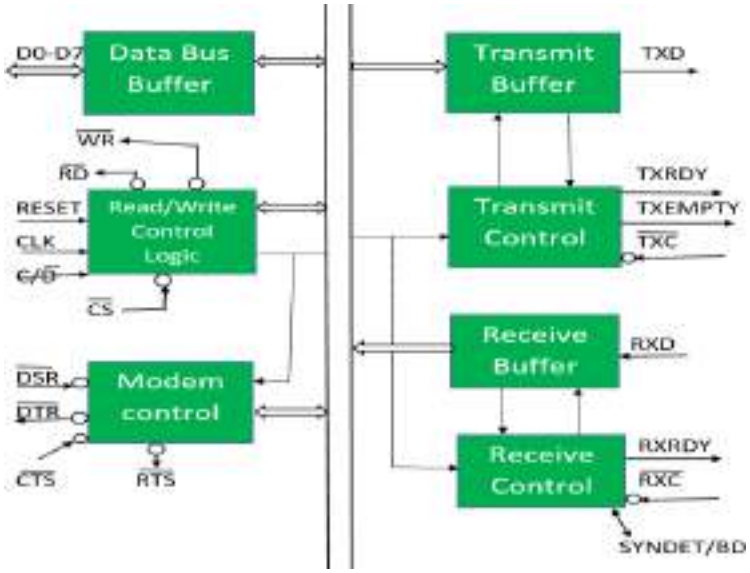
Itisthepowersignalwhichisrequiredfor theoperationofthecircuit

USART:(8251)

8251universalsynchronousasynchronousreceivertransmitter(USART)actsasamediator between microprocessor and peripheral to transmit serial data intoparallelform andviceversa.

1. Ittakesdataseriallyfromperipheral(outsidedevices)andconvertsintoparallеldata.
2. Afterconvertingthedataintoparallelform, ittransmitsittotheCPU.
3. Similarly,itreceivesparalleldatafrommicroprocessorandconvertsitintoserialform.
4. Afterconvertingdataintoserialform,ittransmitsittooutsidedevice(peripheral).

BlockDiagramof8251USART–



Databusbuffer:
Thisblockhelpsininterfacingtheinternaldatabusof8251tothesystemdatabus.The data transmission is possible between 8251 and CPU by the data bus bufferblock.

Read/Writecontrollogic:
Itisacontrolblockforoveralldevice.Itcontrolstheoverallworkingbyselectingtheoperationtobedone.Theoperationselectiondependsuponinputsignalsas:

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	Operation
1	X	X	X	Invalid
0	0	0	1	data CPU< ----- 8251
0	0	1	0	data CPU -----> 8251
0	1	0	1	Status word CPU <----------8251
0	1	1	0	Control word CPU----------> 8251

Inthisway,thisunitselectsoneofthethreeregisters-databufferregister,controlregister,statusregister.

Modemcontrol(modulator/demodulator):

A device converts analog signals to digital signals and vice-versa and helps thecomputerstocommunicateovertelephonelinesorcablewires.Thefollowingareactive-lowpinsof Modem.

- **DSR:**DataSetReadysignalisaninputsignal.
- **DTR:**DataterminalReadyisanoutputsignal.
- **CTS:**Itisaninputsignalwhichcontrolsthedatatransmitcircuit.
  RTS: Itisan outputsignalwhichisusedtosetthestatusRTS.

Transmitbuffer:

This block is used for parallel to serial converter that receives a parallel byte forconversionintoserialsignalandfurthertransmissionontothecommonchannel.

- **TXD:**Itisan outputsignal,ifitsvalueisone,meanstransmitterwilltransmit the data.

Transmitcontrol:

Thisblockisusedtocontrolthedatatransmissionwiththehelpoffollowingpins:

- **TXRDY:**Itmeanstransmitterisreadytotransmitdatacharacter.
- **TXEMPTY:**Anoutput signalwhichindicatesthatTXEMPTYpinhastransmittedallthedatacharactersandtransmitterisemptynow.
- **TXC:**Anactive-lowinputpinwhichcontrolsthedatatransmissionrateoftransmitteddata.

Receivebuffer:
Thisblockactsasabufferforthereceiveddata.

- **RXD:**Aninputsignalwhichreceivesthedata.

Receivecontrol:
Thisblockcontrolsthereceivingdata.

- **RXRDY:**Aninputsignalindicatesthatitisreadytoreceivethedata.
- **RXC:**Anactive-lowinputsignalwhichcontrolsthedatatransmissionrateofreceived data.
- **SYNDET/BD:** An input or output terminal. External synchronous mode-inputterminalandasynchronousmode-outputterminal.