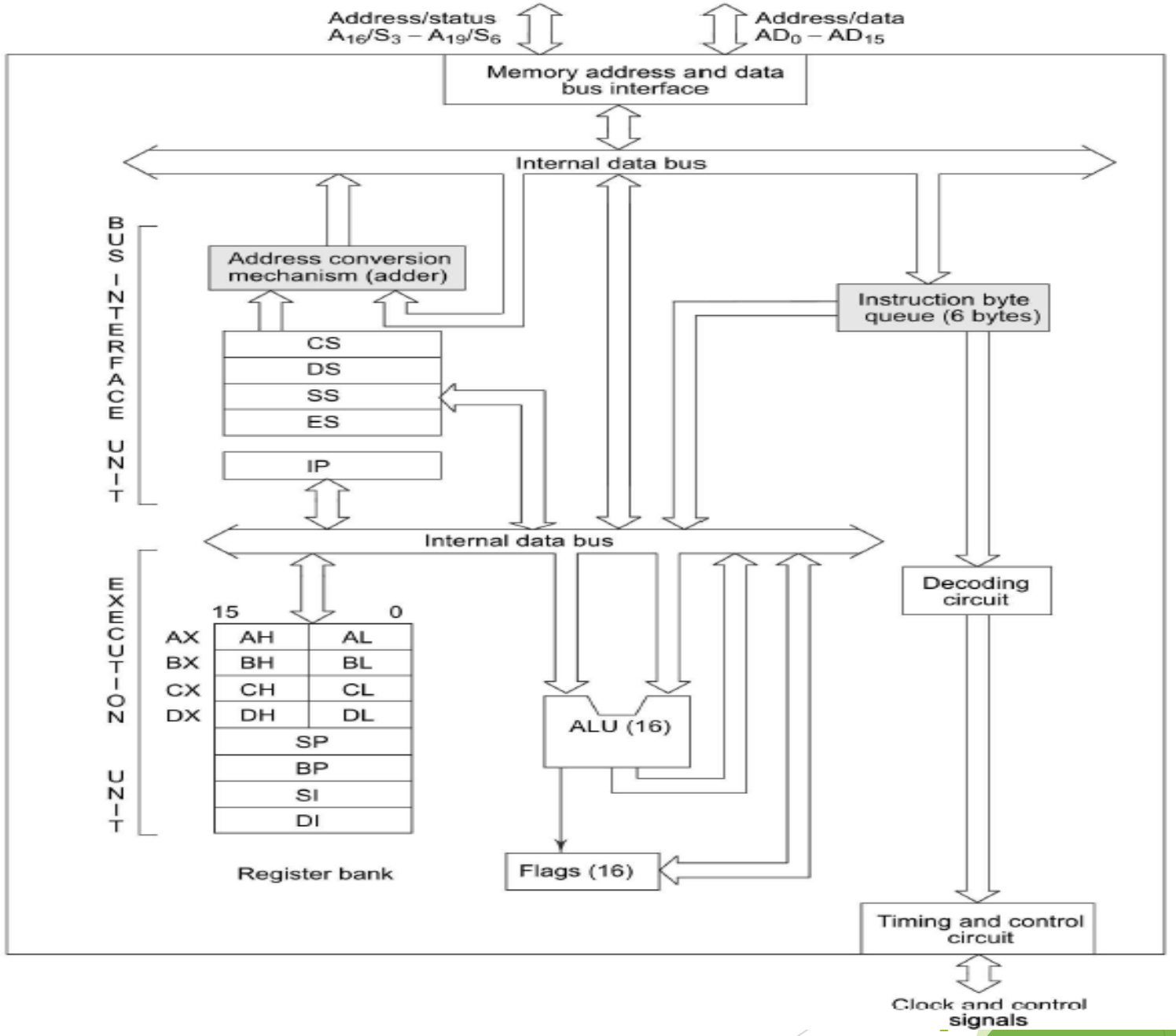
Module 2- 8086 Microprocessor

8086 Architecture



8086 Architecture

8086 Microprocessor is divided into two functional units,

- EU (Execution Unit) and
- BIU (Bus Interface Unit).
  - Contains circuit for physical address calculation and a predecoding instruction byte queue (6BYTES LONG)
  - The Bus Interface Unit (BIU) generates the 20-bit physical memory address and provides the interface with external memory (ROM/RAM).
- EU (Execution Unit)
  - Execution unit gives instructions to BIU starting from where to fetch the data and then decode and execute those instructions. Its function is to control operations on data using the instruction decoder & ALU.
- EU has no direct connection with system buses as shown in the above figure, it performs operations over data through BIU.
  - Alu is a functional unit in EU. ALU handles all arithmetic and logical operations, like +, −, ×, /, OR, AND, NOT operations.

8086 Architecture

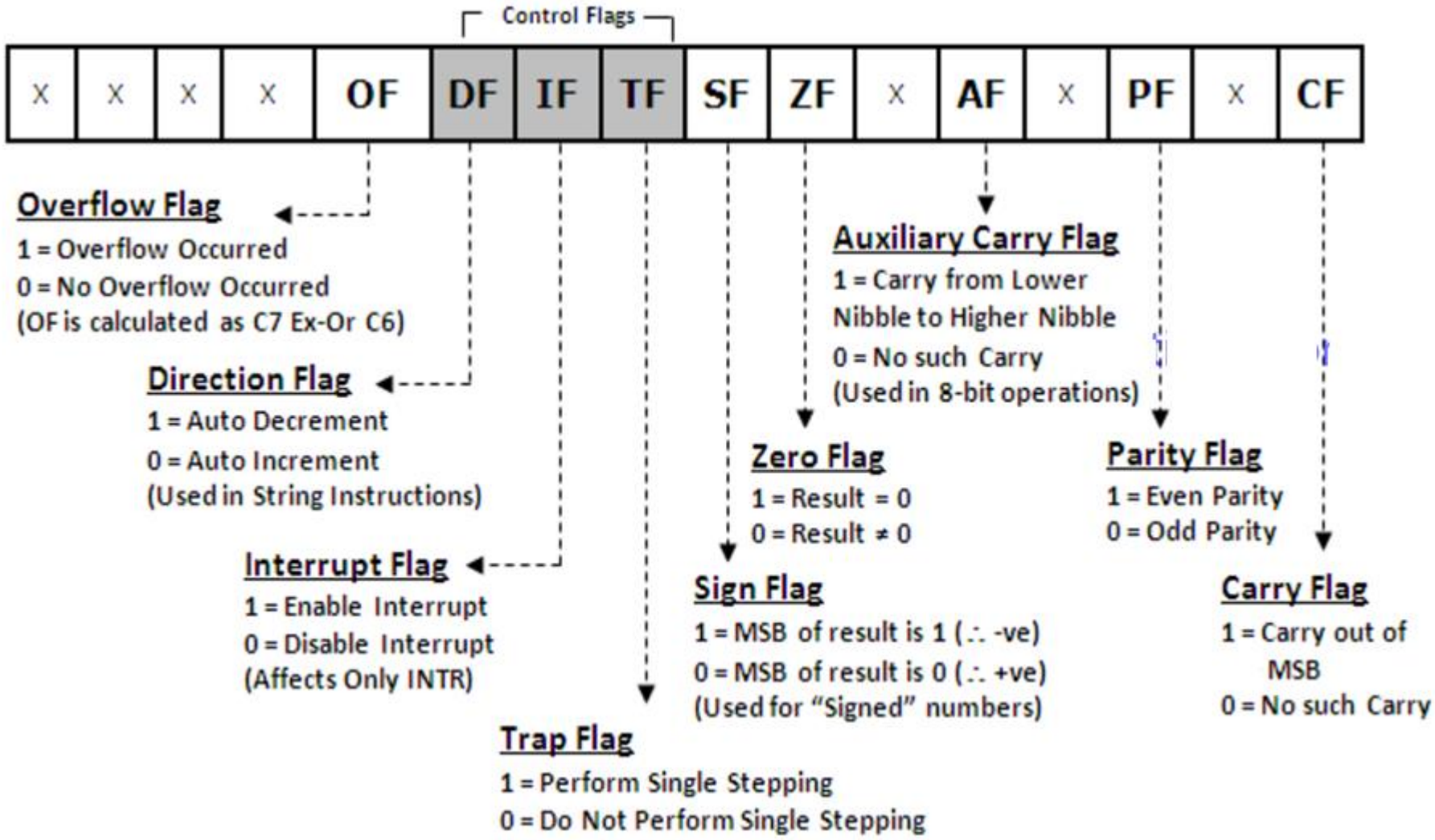
BIU (Bus Interface Unit)

- BIU takes care of all data and addresses transfers on the buses for the EU like sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory.
- EU has no direction connection with System Buses so this is possible with the BIU. EU and BIU are connected with the Internal Bus.
- It has the following functional parts –
- Instruction queue – BIU contains the instruction queue. BIU gets upto 6 bytes of next instructions and stores them in the instruction queue. When EU executes instructions and is ready for its next instruction, then it simply reads the instruction from this instruction queue resulting in increased execution speed.
- Fetching the next instruction while the current instruction executes is called pipelining.

8086 Architecture

- The Bus Interface Unit (BIU) generates the 20-bit physical memory address and provides the interface with external memory (ROM/RAM).
- 8086 has a single memory interface. To speed up the execution, 6-bytes of instruction are fetched in advance and kept in a 6-byte Instruction Queue while other instructions are being executed in the Execution Unit (EU).
- Hence after the execution of an instruction, the next instruction is directly fetched from the instruction queue without having to wait for the external memory to send the instruction.
- This is called pipe-lining and is helpful for speeding up the overall execution process.
- 8086's BIU produces the 20-bit physical memory address by combining a 16-bit segment address with a 16-bit offset address.

8086 Architecture- PSW- Flag Register



8086 Architecture- Flag Register

Flag Register

- It is a 16-bit register that behaves like a flip-flop, i.e. it changes its status according to the result stored in the accumulator.
- It has 9 flags and they are divided into 2 groups – **Conditional Flags and Control Flags.**

Conditional Flags

- It represents the result of the last arithmetic or logical instruction executed. Following is the list of conditional flags –
- **Carry flag** – This flag indicates an overflow condition for arithmetic operations.

8086 Architecture- Flag Register

- **Auxiliary flag** – When an operation is performed at ALU, it results in a carry/barrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), then this flag is set, i.e. carry given by D3 bit to D4 is AF flag. The processor uses this flag to perform binary to BCD conversion.
- **Parity flag** – This flag is used to indicate the parity of the result, i.e. when the lower order 8-bits of the result contains even number of 1's, then the Parity Flag is set. For odd number of 1's, the Parity Flag is reset.
- **Zero flag** – This flag is set to 1 when the result of arithmetic or logical operation is zero else it is set to 0.
- **Sign flag** – This flag holds the sign of the result, i.e. when the result of the operation is negative, then the sign flag is set to 1 else set to 0.
- **Overflow flag** – This flag represents the result when the system capacity is exceeded.

8086 Architecture- Flag Register

Control Flags

- Control flags controls the operations of the execution unit. Following is the list of control flags –
- **Trap flag** – It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in a single step mode.
- **Interrupt flag** – It is an interrupt enable/disable flag, i.e. used to allow/prohibit the interruption of a program. It is set to 1 for interrupt enabled condition and set to 0 for interrupt disabled condition.
- **Direction flag** – It is used in string operation. As the name suggests when it is set then string bytes are accessed from the higher memory address to the lower memory address and vice-a-versa.

8086 Architecture- Functional parts

Decoding Unit

- The decoding unit decodes opcode bytes issued from the instruction byte queue.

Timing and Control Unit

- Timing and control unit derives the necessary control signals to execute the instruction opcode received from the queue, depending on information made available by decoding circuit.
- The execution unit passes the result to bus interface unit for storing them in memory.

Addressing modes

- The different ways in which a source operand is denoted in an instruction is known as addressing modes.
- There are 8 different addressing modes in 8086 programming –
- **Immediate addressing mode** The addressing mode in which the data operand is a part of the instruction itself is known as immediate addressing mode.
- **Register addressing mode** It means that the data is stored in a register and it is referred using particular register. All registers except IP may be used in this mode

```
MOV CX, 4929 H, ADD AX, 2387 H,  MOV AL, FFH
```

Example

```
MOV CX, AX    ; copies the contents of the 16-bit AX register into
              ; the 16-bit CX register),
ADD BX, AX
```

Addressing mode

Direct addressing mode

- The addressing mode in which the 16 bit memory address(offset) is directly specified in the instruction.

```
MOV AX, [1592H], MOV AL, [0300H]
```

- Data resides in data segment. Effective address computed by adding 1592h to content if Data segment which is segment address -10HxDS +1592 h

Register indirect addressing mode

- This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI & SI. The default segment is DS or ES.

Eg: MOV AX,[BX] -Data is present in data segment whose offset address is in BX. The effective address is 10H*DS+[BX]

Addressing mode

Indexed addressing mode

- In this addressing mode, OFFSET of the operand is stored in one of the index registers. DS is default segment for SI and DI. Incase of string instructions DS and ES are default segments of SI and DI respectively.

Eg: MOV AX,[SI] Effective address is 10HxDS+[SI]

Register relative

- Data is available at an effective address formed by adding 8-bit/16-bit displacement with the content of any of the registers BX, BP, SI, DI in the default (DS or ES segment)

Eg: MOV AX, 50H[BX]- Effective address is 10HXDS+50H+[BX]

Addressing mode

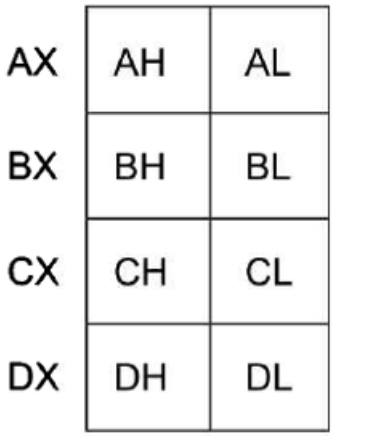
Based-index addressing mode

- In this addressing mode, the offset address of the operand is computed by summing the base register (BX or BP) to the contents of an Index register (SI or DI).
- Eg: MOV AX,[BX] [SI]- BX is base register, SI IS INDEX REGISTER. The effective address is 10HxDS +[BX]+[SI]

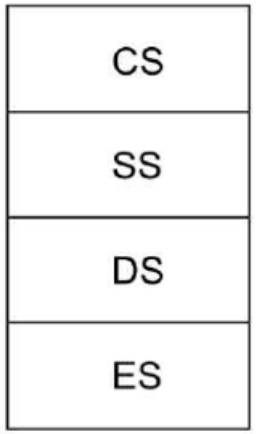
Relative based indexed

- In this addressing mode, effective address is formed by adding 8 or 16 bit displacement with sum of contents of any one of the base registers (BX or BP) and any one of the index registers in a default segment.
- Eg: MOV AX,50H [BX][SI]
- 50H is displacement. BX is base register. SI is index register. Effective address is 10H*DS+{BX]+[SI]+50H

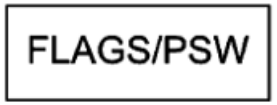
8086 register organization

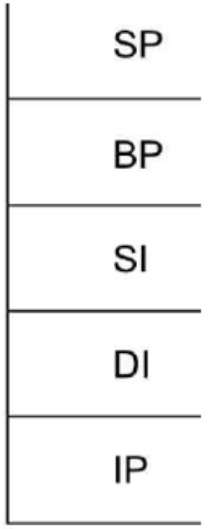


General data registers



Segment registers





Pointers and Indexing registers

8086 register organization

- **Segment register** – BIU has 4 segment buses, i.e. CS, DS, SS& ES. It holds the addresses of instructions and data in memory, which are used by the processor to access memory locations. It also contains 1 pointer register IP, which holds the address of the next instruction to executed by the EU.
- CS – It stands for Code Segment. It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.
- DS – It stands for Data Segment. It consists of data used by the program and is accessed in the data segment by an offset address or the content of other register that holds the offset address.
- SS – It stands for Stack Segment. It handles memory to store data and addresses during execution.
- ES – It stands for Extra Segment. ES is additional data segment, which is used by the string to hold the extra destination data.
- Instruction pointer (IP)– It is a 16-bit register used to hold the address of the next instruction to be executed

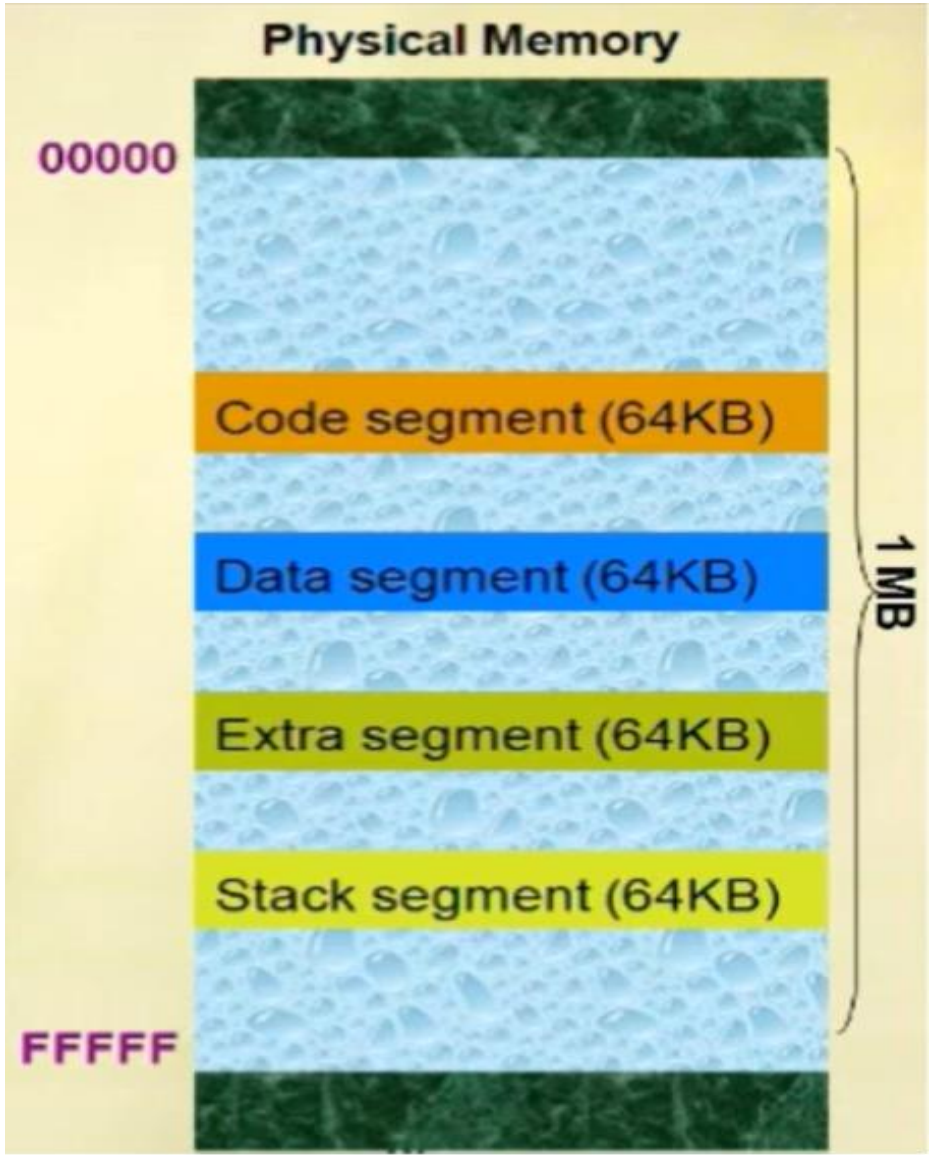
8086 General purpose registers

- There are 8 general purpose registers, i.e., AH, AL, BH, BL, CH, CL, DH, and DL.
- These registers can be used individually to store 8-bit data and can be used in pairs to store 16bit data. The valid register pairs are AH and AL, BH and BL, CH and CL, and DH and DL.
- It is referred to the AX, BX, CX, and DX respectively.
- AX register – It is also known as accumulator register. It is used to store operands for arithmetic operations.
- BX register – It is used as a base register. It is used as an offset storage to form memory address in certain addressing modes.
- CX register – It is referred to as counter. It is used in loop instruction to store the loop counter.
- DX register – This register is used as a general purpose register which may be used as an implicit operand or destination in case of few instructions.

8086 Pointer and index registers

- IP-Instruction Pointer-offset with in code segment.
- BP-Base pointer-Offset within stack segment.
- SP-Stack pointer-Offset within stack segment
- SI-Source Index register
- DI-Destination Index
- The index registers are used as general purpose registers as well as for offset storage in case of indexed ,base indexed and relative base indexed addressing modes.
- SI is used to store offset of source data in data segment
- DI is used to store offset of destination data or extra segment.
- The index registers are particularly useful for string manipulations.

8086 Memory segmentation



8086 Memory segmentation

- The number of address lines in 8086 is 20, 8086 BIU will send 20bit address, so as to access one of the 1MB memory locations.
- 8086 has a segmented memory.
- Complete physical memory may be divided into no : of logical segments. Each segment is 64 Kbytes in Size and is addressed by one of the segments.
- The 16 bit content of segment register actually point to starting location of a particular segment.
- To address a specific memory location within a segment we need an offset address. The offset address is also 16 bits long.
- Complete 1 MB memory can be divided into 16 segments each of 64KBytes.

8086 Memory segmentation

Advantages of segmented memory scheme

- Allows the memory capacity to be 1 MB although the actual address to be handled are of 16 bit size.
- Allows placing of code data and stack portions of same program in different parts of memory for data and code protection
- Permits a program and/or its data to be put to different areas of memory each time program is executed. ie provision for relocation is done

8086 Memory segmentation

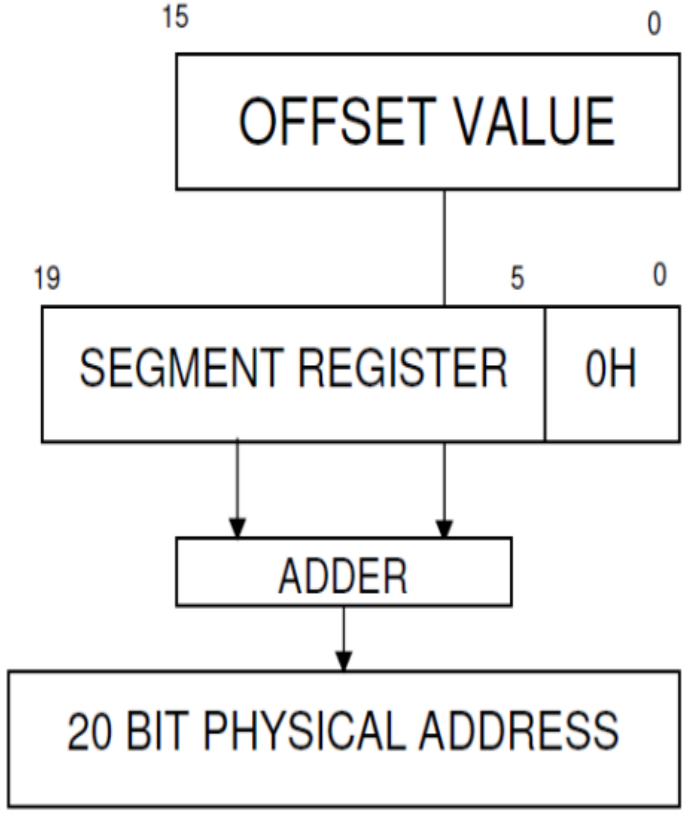
- Each of these segments are addressed by an address stored in corresponding segment register.
- These registers are of 16-bit in size.
- Each register stores the base address (starting address) of the corresponding segment.
- Because the segment registers cannot store 20 bits, they only store the upper 16 bits

8086 Memory segmentation

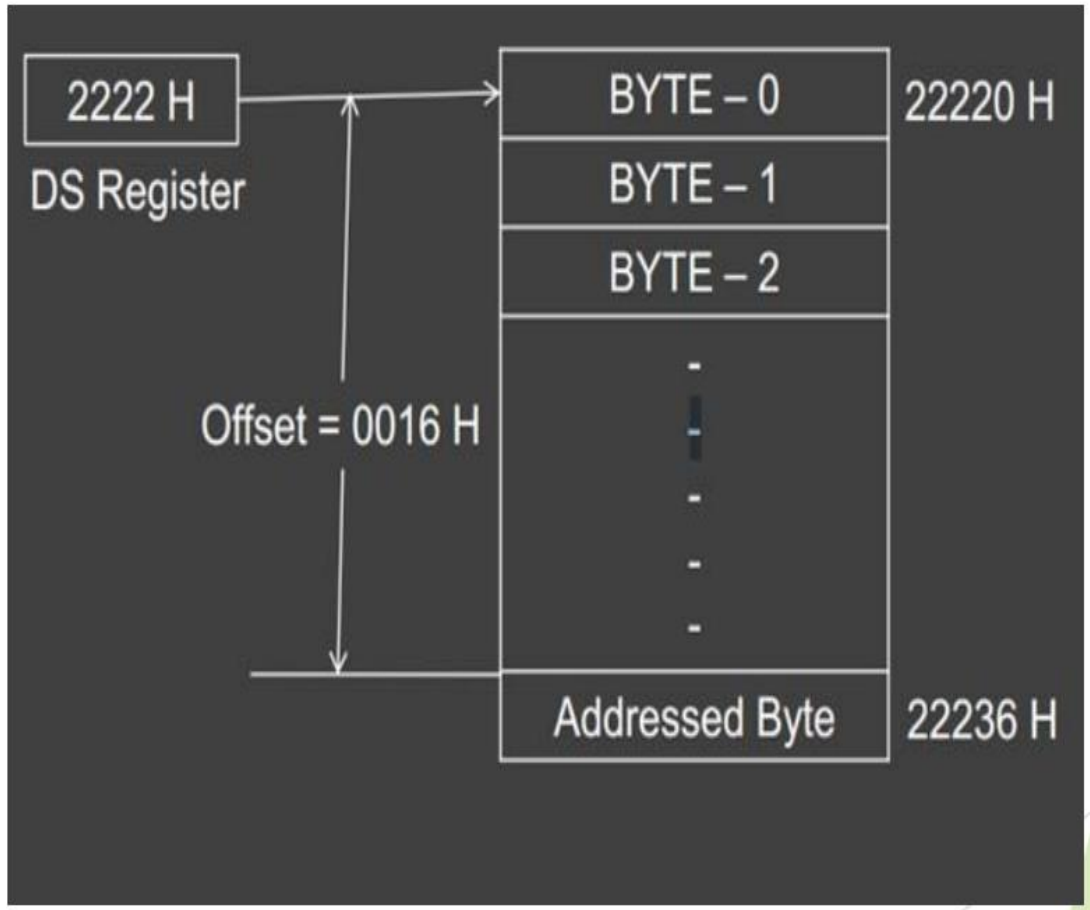
- Rules of Segmentation
- Segmentation process follows some rules as follows:
- The starting address of a segment should be such that it can be evenly divided by 16.
- Minimum size of a segment can be 16 bytes and the maximum can be 64 kB.

Segment	Offset Registers	Function
CS	IP	Address of the next instruction
DS	BX, DI, SI	Address of data
SS	SP, BP	Address in the stack
ES	BX, DI, SI	Address of destination data (for string operations)

8086 Creating physical address



Example

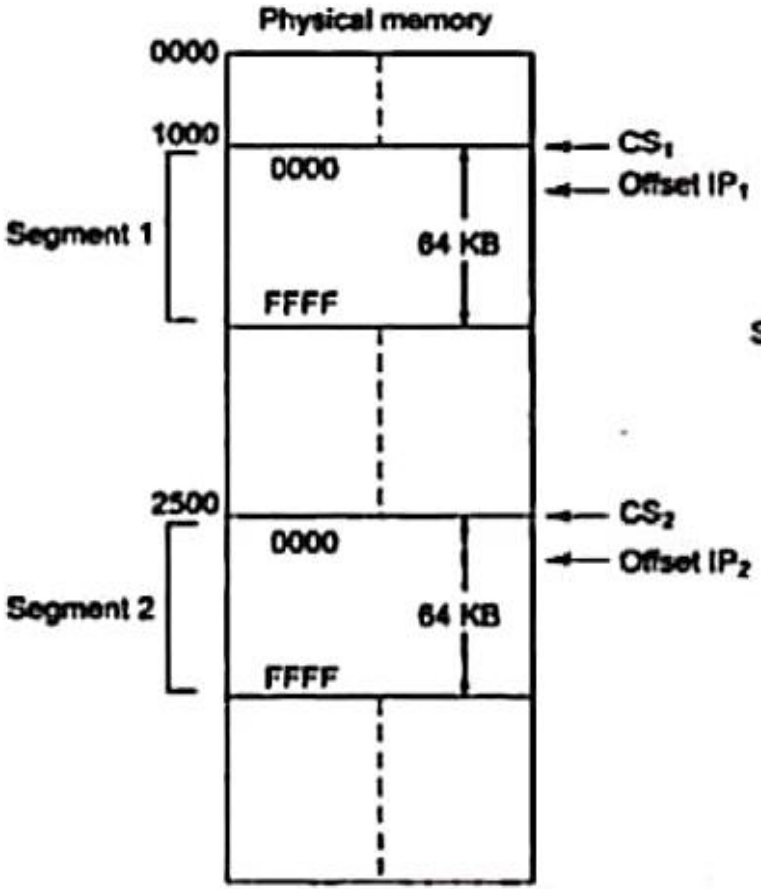


8086 Physical address calculation

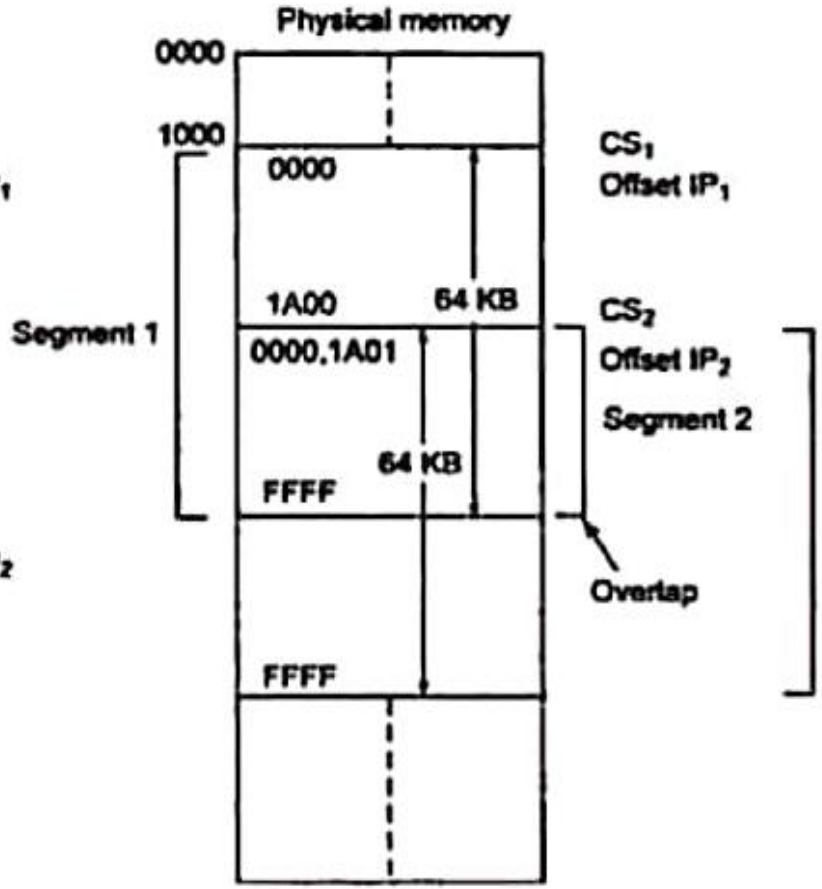
- For Example:
- Code segment Register CS holds the segment address which is 4569 H
- Instruction pointer IP holds the offset address which is 10A0 H
- The physical 20-bit address is calculated as follows.
- Segment address: 45690 H
- Offset address :+ 10A0 H
- Physical address : 46730 H

```
45690
 10A0
46730
```

8086 Segmentation



Non overlapping

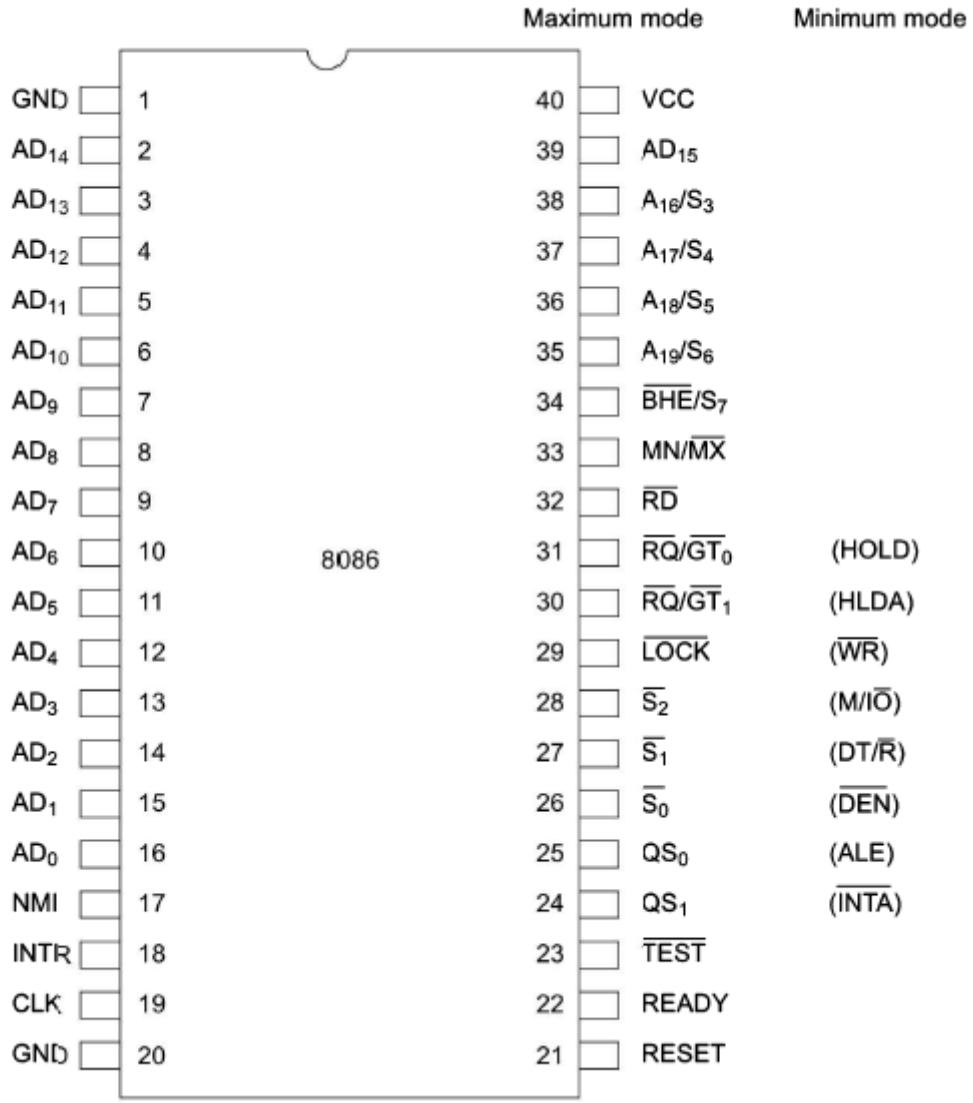


Overlapping

MINIMUM MODE AND MAXIMUM MODE

- MINIMUM MODE-SINGLE PROCESSOR MODE
- MAXIMUM MODE-MULTIPROCESSOR MODE

Pin configuration



Pin configuration

Power supply and frequency signals

- It uses 5V DC supply at VCC pin 40, and uses ground at VSS pin 1 and 20 for its operation.

Clock signal

- Clock signal is provided through Pin-19. It provides timing to the processor for operations. Its frequency is different for different versions, i.e. 5MHz, 8MHz and 10MHz.

Address/data bus AD0-AD15.

- These are 16 address/data bus. AD0-AD7 carries low order byte data and AD8AD15 carries higher order byte data. During the first clock cycle, it carries 16-bit address and after that it carries 16-bit data.

Address/status bus

- A16-A19/S3-S6. These are the 4 address/status buses. During the first clock cycle, it carries 4-bit address and later it carries status signals.

S7/BHE

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

Pin configuration

Read

- It is available at pin 32 and is used to read signal for Read operation.

Ready

- It is available at pin 32. It is an acknowledgement signal from I/O devices that data is transferred. It is an active high signal. When it is high, it indicates that the device is ready to transfer data. When it is low, it indicates wait state.

RESET

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

INTR

- It is available at pin 18. It is an interrupt request signal, which is sampled during the last clock cycle of each instruction to determine if the processor considered this as an interrupt or not.

NMI

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

Pin configuration

TEST

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

MN/MX

- It stands for Minimum/Maximum and is available at pin 33. It indicates what mode the processor is to operate in; when it is high, it works in the minimum mode and vice-versa.

INTA

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

ALE

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

Pin configuration

DEN

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

DT/R

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

M/IO

- This signal is used to distinguish between memory and I/O operations.
- When it is high, it indicates I/O operation and when it is low indicates the memory operation. It is available at pin 28.

Pin configuration

WR

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

HLDA

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

HOLD

- This signal indicates to the processor that external devices are requesting to access the address/data buses. It is available at pin 31.

QS1 and QS0

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

Pin configuration

QS1 and QS0

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

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

Pin configuration

S0, S1, S2

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

S_2	S_1	S_0	Status
0	0	0	Interrupt acknowledgement
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive

Pin configuration

LOCK

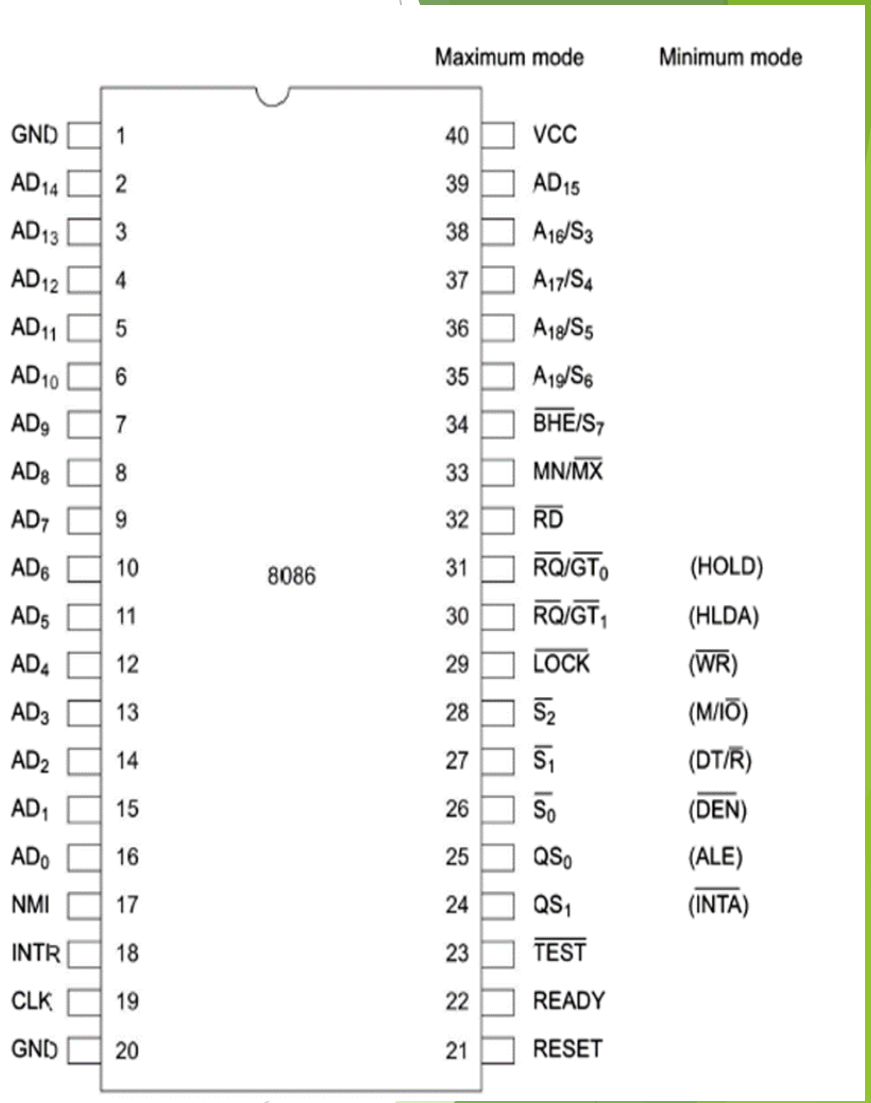
- When this signal is active, it indicates to the other processors not to ask the CPU to leave the system bus. It is activated using the LOCK prefix on any instruction and is available at pin 29.

RQ/GT1 and RQ/GT0

- These are the Request/Grant signals used by the other processors requesting the CPU to release the system bus.
- When the signal is received by CPU, then it sends acknowledgment. RQ/GT0 has a higher priority than RQ/GT1.

Minimum and Maximum mode

Minimum mode	Maximum mode
In minimum mode there can be only one processor i.e. 8086.	In maximum mode there can be multiple processors with 8086, like 8087 and 8089.
$MN/\overline{MX}$ is 1 to indicate minimum mode.	$MN/\overline{MX}$ is 0 to indicate maximum mode.
ALE for the latch is given by 8086 as it is the only processor in the circuit.	ALE for the latch is given by 8288 bus controller as there can be multiple processors in the circuit.
$\overline{DEN}$ and $DT/\overline{R}$ for the trans-receivers are given by 8086 itself.	and$DT/\overline{R}$ for the trans-receivers are given by 8288 bus controller.
Direct control signals $M/\overline{IO}$, $\overline{RD}$ and $\overline{WR}$ are given by 8086.	Instead of control signals, each processor generates status signals called $\overline{S_2}$, $\overline{S_1}$ and $\overline{S_0}$.
Control signals $M/\overline{IO}$, $\overline{RD}$ and $\overline{WR}$ are decoded by a 3:8 decoder like 74138.	Status signals $\overline{S_2}$, $\overline{S_1}$ and $\overline{S_0}$ are decoded by a bus controller like 8288 to produce control signals.
$\overline{INTA}$ is given by 8086 in response to an interrupt on INTR line.	$\overline{INTA}$ is given by 8288 bus controller in response to an interrupt on INTR line.
HOLD and HLDA signals are used for bus request with a DMA controller like 8237.	$\overline{RQ}/\overline{GT}$,lines are used for bus requests by other processors like 8087 or 8089.
The circuit is simpler.	The circuit is more complex.



Minimum mode

(HOLD)

(HLDA)

($\overline{WR}$)

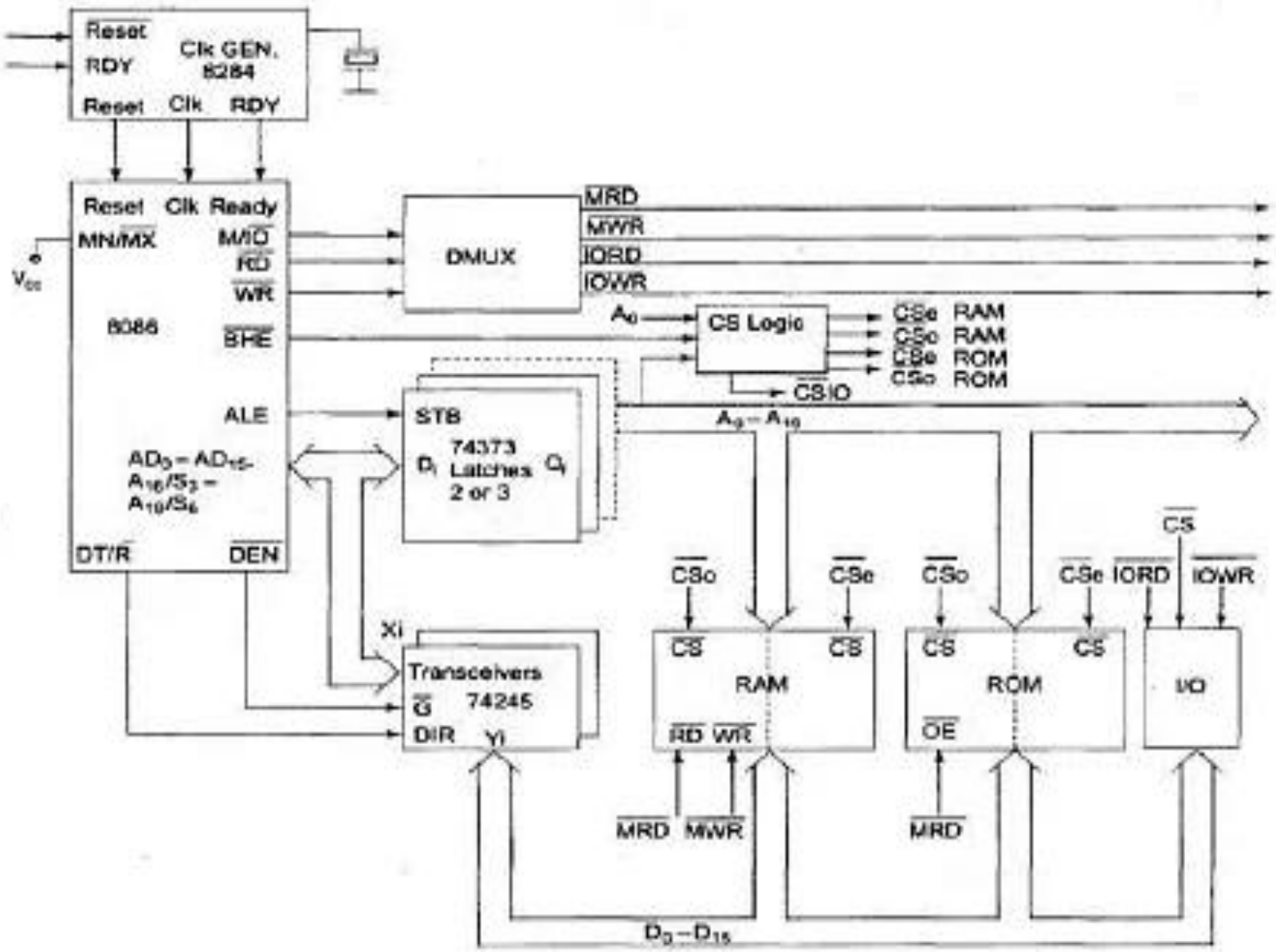
(M/$\overline{IO}$)

(DT/$\overline{R}$)

($\overline{DEN}$)

(ALE)

($\overline{INTA}$)



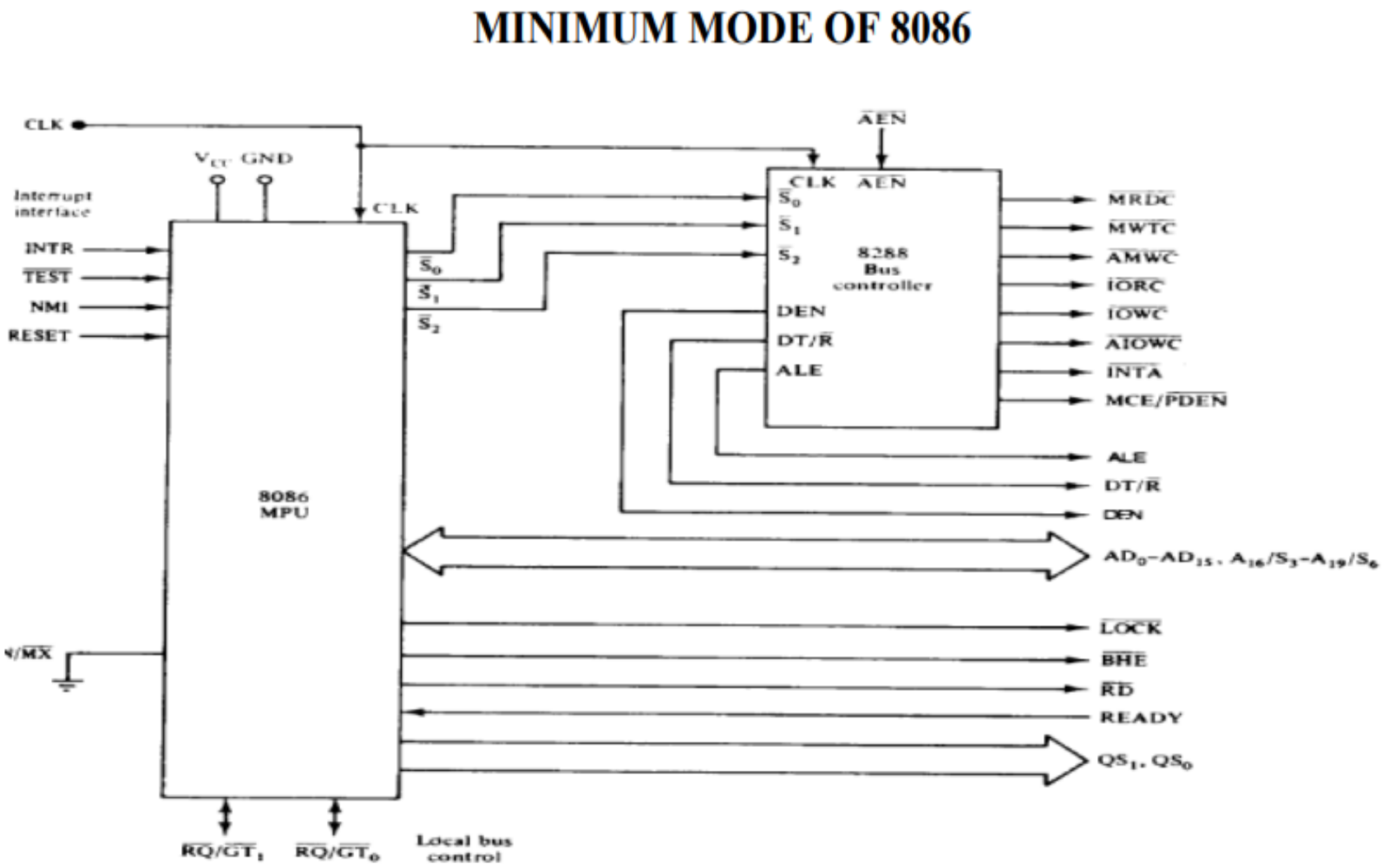
Minimum mode

- In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN//MX pin to logic 1.
- In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system.
- The remaining components in the system are latches, Transreceivers, clock generator, memory and I/O devices.
- Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.
- Latches are generally buffered output D-type flip-flops like 74LS373 or 8282.
- They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086.

Minimum mode

- In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN//MX pin to logic 1.
- In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system.
- The remaining components in the system are latches, Tran receivers, clock generator, memory and I/O devices.
- Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.
- Latches are generally buffered output D-type flip-flops like 74LS373 or 8282.
- They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086.

Minimum mode



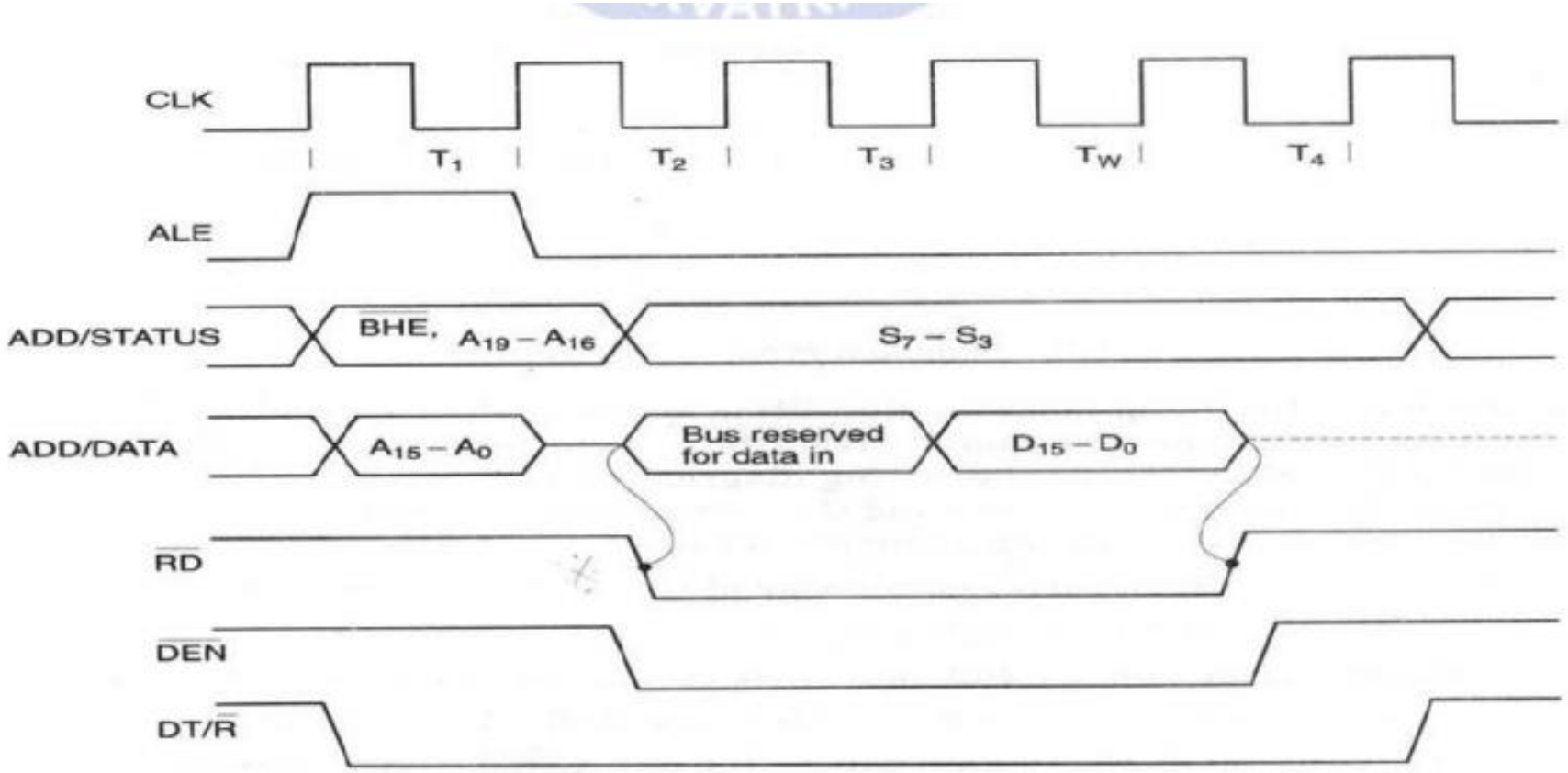
Minimum mode

- Usually, EPROMs are used for monitor storage, while RAM for users program storage. A system may contain I/O devices.
- The opcode fetch and read cycles are similar. Hence the timing diagram can be categorized in two parts, the first is the timing diagram for read cycle and the second is the timing diagram for write cycle.
- The read cycle begins in T1 with the assertion of address latch enable (ALE) signal and also M / IO signal.
- During the negative going edge of this signal, the valid address is latched on the local bus.
- The BHE and A0 signals address low, high or both bytes. From T1 to T4, the M/IO signal indicates a memory or I/O operation.
- At T2, the address is removed from the local bus and is sent to the output. The bus is then tristate.
- The read (RD) control signal is also activated in T2.

Minimum mode

- The read (RD) signal causes the address device to enable its data bus drivers. After RD goes low, the valid data is available on the data bus.
- The addressed device will drive the READY line high. When the processor returns the read signal to high level, the addressed device will again tristate its bus drivers.
- A write cycle also begins with the assertion of ALE and the emission of the address. The M/IO signal is again asserted to indicate a memory or I/O operation.
- In T2, after sending the address in T1, the processor sends the data to be written to the addressed location.
- The data remains on the bus until middle of T4 state.
- The WR becomes active at the beginning of T2 (unlike RD is somewhat delayed in T2 to provide time for floating).
- The BHE and A0 signals are used to select the proper byte or bytes of memory or I/O word to be read or write

Minimum mode



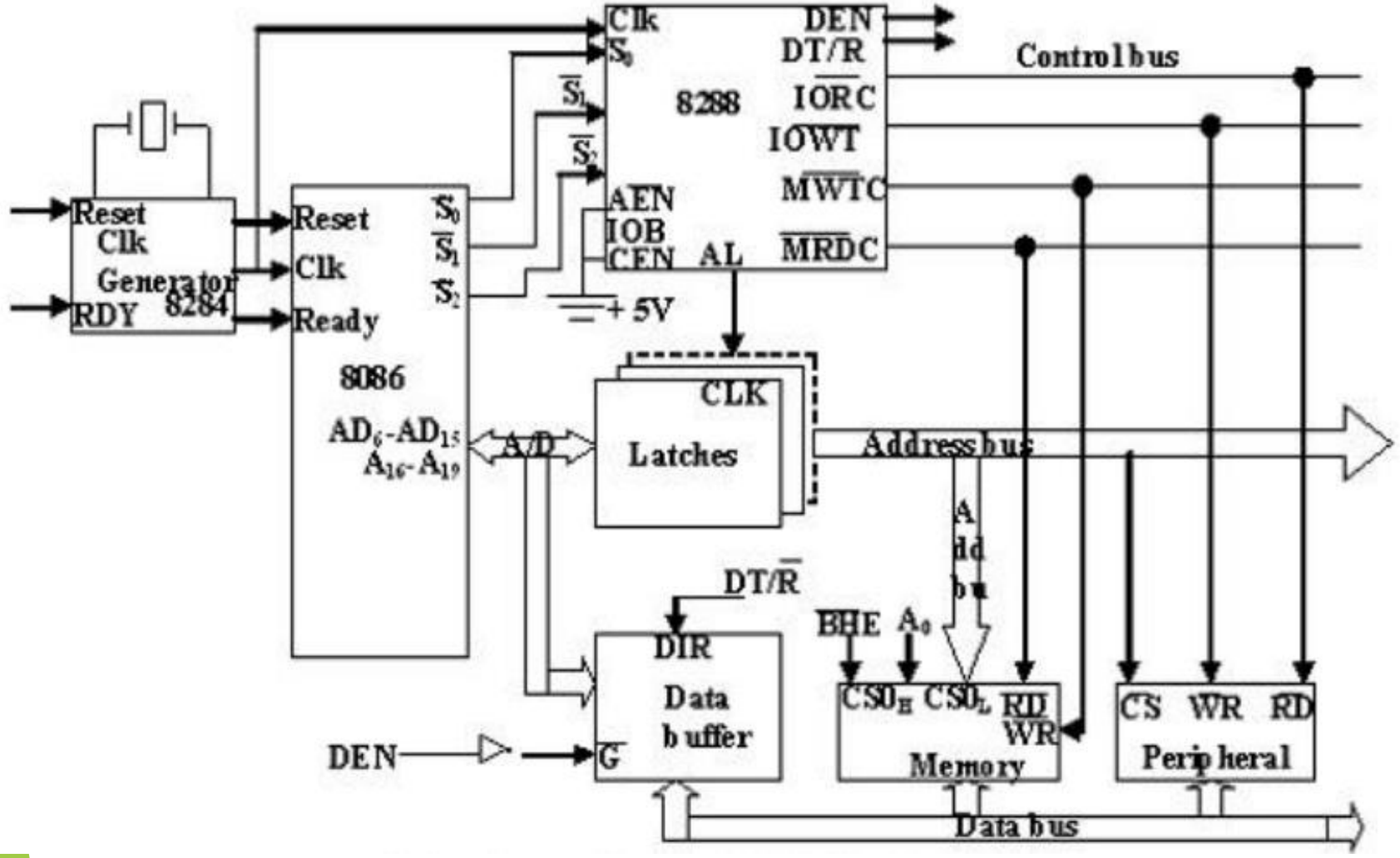
MEMORY READ CYCLE FOR 8086 IN MINIMUM MODE

Maximum mode of 8086

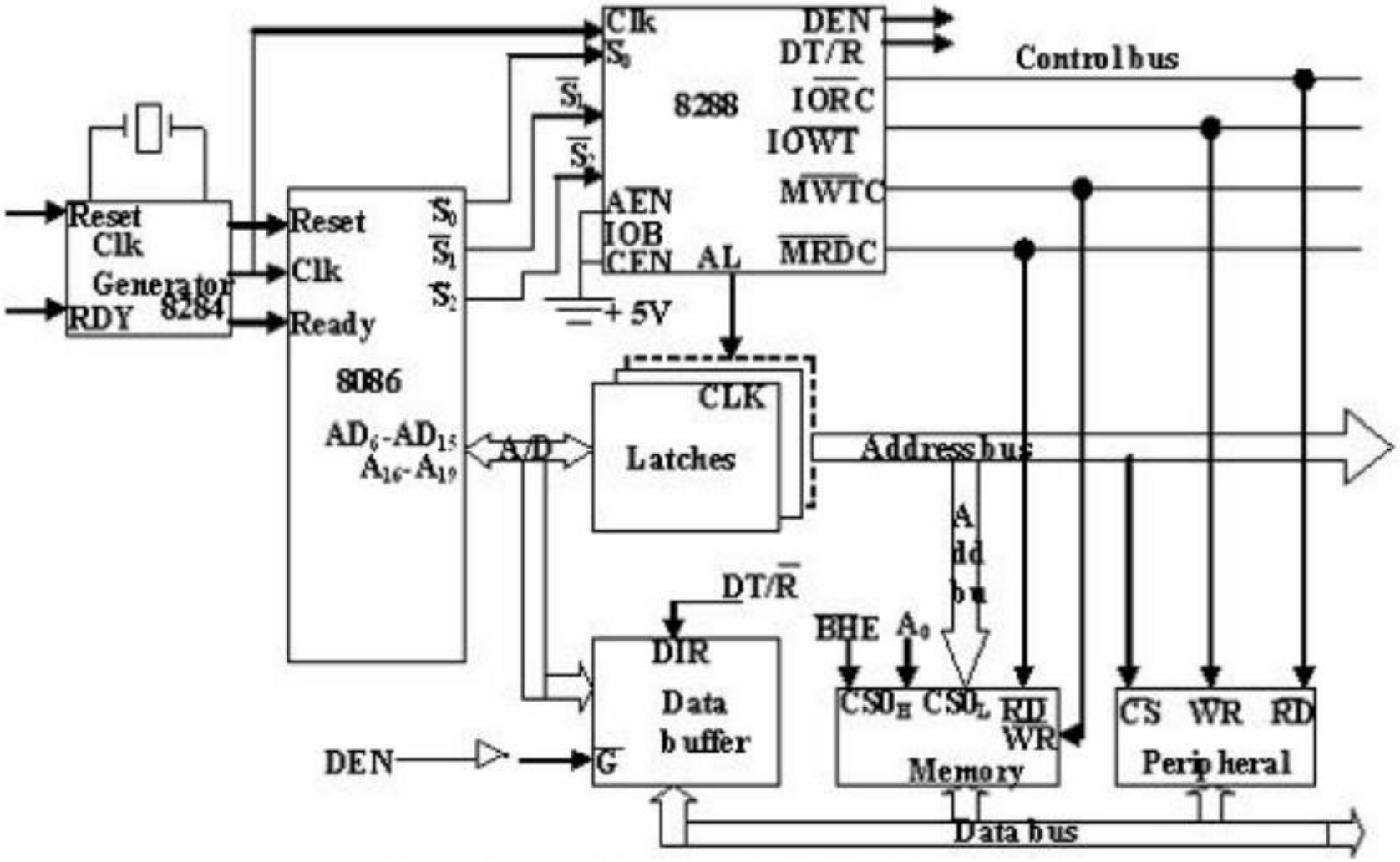
In the maximum mode, the 8086 is operated by strapping the MN/MX pin to ground.

- In this mode, the processor derives the status signal S2, S1, S0. Another chip called bus controller derives the control signal using this status information.
- In the maximum mode, there may be more than one microprocessor in the system configuration.
- The components in the system are same as in the minimum mode system.
- The basic function of the bus controller chip IC8288, is to derive control signals like RD and WR (for memory and I/O devices), DEN, DT/R, ALE etc. using the information by the processor on the status lines.
- The bus controller chip has input lines S2, S1, S0 and CLK. These inputs to 8288 are driven by CPU.
- It derives the outputs ALE, DEN, DT/R, MRDC, MWTC, AMWC, IORC, IOWC and AIOWC. The AEN, IOB and CEN pins are specially useful for multiprocessor systems.

Maximum mode of 8086



Maximum mode of 8086



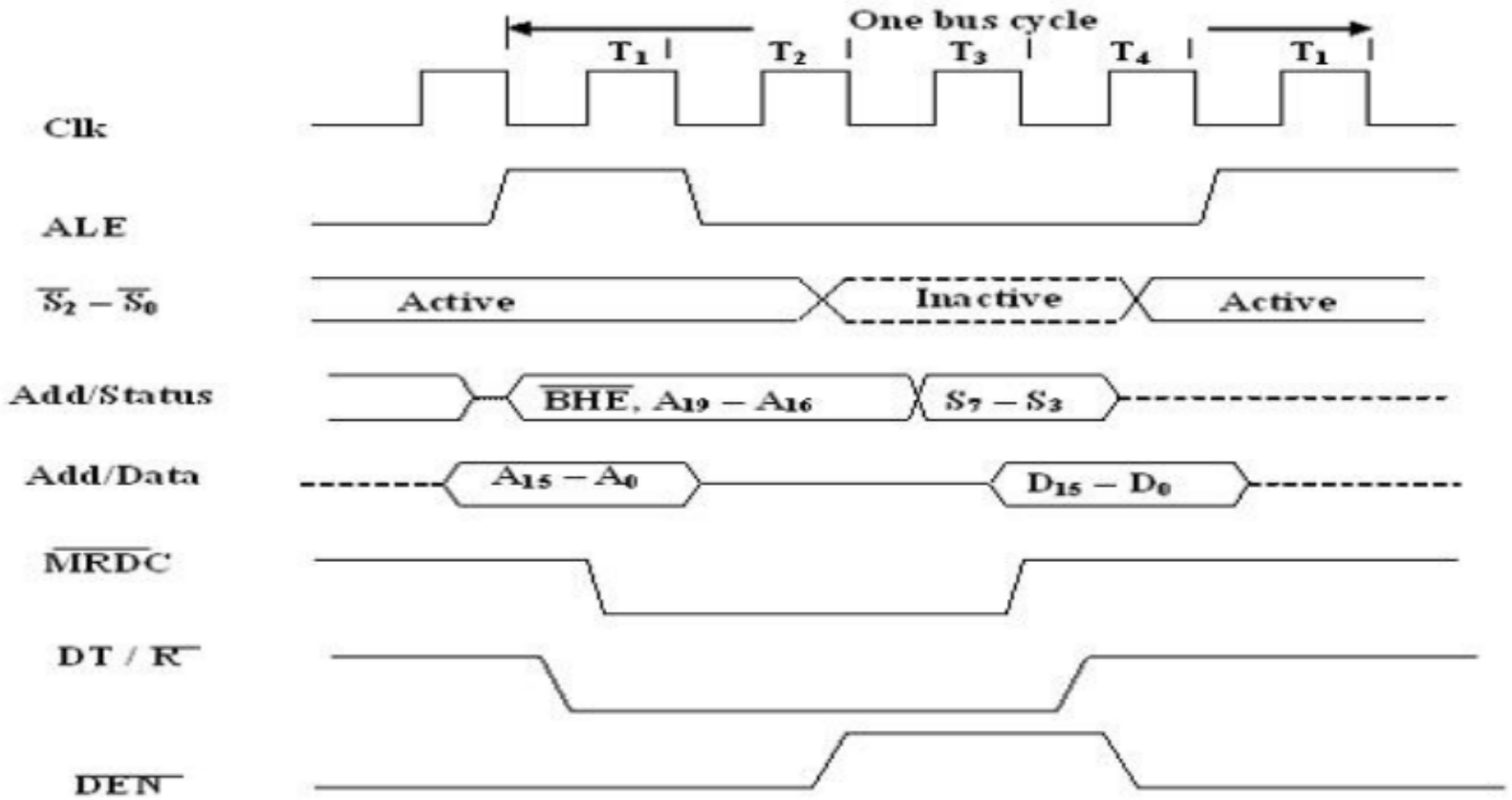
Maximum mode of 8086

- INTA pin used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device.
- IORC, IOWC are I/O read command and I/O write command signals respectively.
- These signals enable an IO interface to read or write the data from or to the address port.
- The MRDC, MWTC are memory read command and memory write command signals respectively and may be used as memory read or write signals.
- All these command signals instructs the memory to accept or send data from or to the bus.
- For both of these write command signals, the advanced signals namely AIOWC and AMWTC are available.
- Here the only difference between in timing diagram between minimum mode and maximum mode is the status signals used and the available control and advanced command signals.

Maximum mode of 8086

- R0, S1, S2 are set at the beginning of bus cycle.8288 bus controller will output a pulse as on the ALE and apply a required signal to its DT / R pin during T1.
- In T2, 8288 will set DEN=1 thus enabling transceivers, and for an input it will activate MRDC or IORC. These signals are activated until T4.
- For an output, the AMWC or AIOWC is activated from T2 to T4 and MWTC or IOWC is activated from T3 to T4.
- The status bit S0 to S2 remains active until T3 and become passive during T3 and T4.
- If reader input is not activated before T3, wait state will be inserted between T3 and T4.

Maximum mode of 8086



Memory Read Timing in Maximum Mode