The 8086 Microprocessor architecture(Chapter 2)

Adane Tadesse

Faculty of Electrical and Computer Engineering

Computer and control Stream

Jimma University

Jimma, Oromia, Ethiopia

email: addmomcomp@gmail.com

January 22, 2020

Outline

Features of 8086 Microprocessors

- It is 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
- Single +5V power supply
- Clock speed range of 5-10MHz
- Fetch stage can pref etch up to 6 bytes of instructions and stores them in the queue.
- It has 256 interrupts.
- A 40 pin dual in line package.

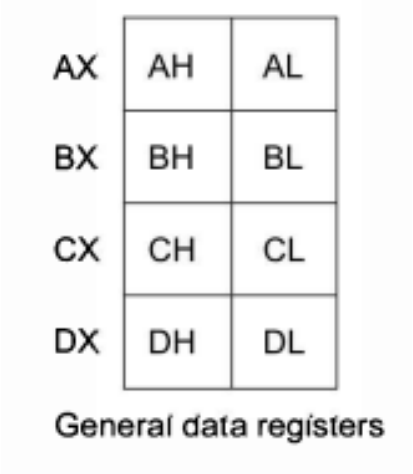
- It has a 20 bit address bus can access up to pow(2,20)=1 MB memory locations.
- It has multiplexed address and data bus AD0-AD15 and A16 –A19.
- Address ranges from 00000H to FFFFFH
- 8086 is designed to operate in two modes, Minimum and Maximum.
- Word size is 16 bits and double word size is 4 bytes.
- The 8086 has 14 registers each one with 16-bit.

Register Organization

- 8086 has a powerful set of registers known as general purpose registers and special purpose registers.
  - **general purpose registers**
    - These registers can be used as either 8-bit registers or 16-bit registers.
    - They may be either used for holding data, variables and intermediate results temporarily or for other purposes like a counter or for storing offset address for some particular addressing modes etc
  - **Special purpose register:**used as segment registers, pointers, index registers or as offset storage registers for

- The 8086 registers are classified into the following types:
  - General Data Registers
  - Segment Registers
  - Pointers and Index Registers
  - Flag Register
- **General Data Registers**
  - The 8086 has four 16-bit general purpose registers labeled **AX, BX, CX, and DX**.
  - Each 16-bit general purpose register can be split into two 8-bit registers.
  - The letters **L** and **H** specify the lower and the higher bytes of a particular register. For example, the BH means the higher byte and BL means the lower byte of BX

- The letter X is used to specify the 16-bit register.



- **Accumulator register (AX):**
  - Consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX.
  - AL in this case contains the low order byte of the word, and AH contains the high-order byte.

- The I/O instructions use the AX or AL for inputting / outputting 16 or 8 bit data to or from an **I/O port**.
- Multiplication and Division instructions also use the **AX or AL**.
- **Base Register (BX):**
  - Consists of two 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX.
  - This is the only general purpose register whose contents can be used for addressing the 8086 memory.
  - All memory references utilizing this register content for addressing use DS as the default segment register.
  - it is used to store the value of the offset.

- **Counter Register (CX):**
  - Consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX.
  - When combined, CL register contains the low order byte of the word, and CH contains the high-order byte.
  - Instructions such as **SHIFT, ROTATE** and **LOOP** use the contents of CX as a counter.
  - Example:

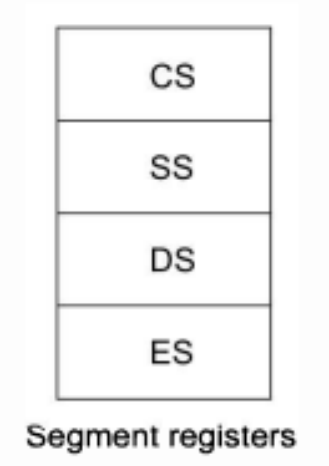
    The instruction LOOP START automatically decrements CX by 1 without affecting flags and will check if [CX] = 0.

    If it is zero, 8086 executes the next instruction; otherwise the 8086 branches to the label START.

- **Data Register (DX)**
  - Consists of two 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX.
  - When combined, DL register contains the low order byte of the word, and DH contains the high-order byte.
  - Used to hold the high 16-bit result (data) in 16 X 16 multiplication or the high 16-bit dividend (data) before a 32 / 16 division and the 16-bit reminder after division.

Segment Registers:

- 8086 has four special segment registers: cs, ds, es, and ss. These stand for Code Segment, Data Segment, Extra Segment, and Stack Segment, respectively
- These registers are all 16 bits wide. They deal with selecting blocks (segments) of main memory.



Segment registers

Code Segment Register(CS)

- 16-bit
- CS contains the base or start of the current code segment; IP contains the distance or offset from this address to the next instruction byte to be fetched.
- it computes the 20-bit physical address by logically shifting the contents of CS 4-bits to the left and then adding the 16-bit contents of IP.
- That is, all instructions of a program are relative to the contents of the CS register multiplied by 16 and then offset is added provided by the IP.

Data Segment Register(DS)

- 16-bit
- Points to the current data segment; operands for most instructions are fetched from this segment.
- The 16-bit contents of the Source Index (SI) or Destination Index (DI) or a 16-bit displacement are used as offset for computing the 20-bit physical address.

Stack Segment Register(SS)

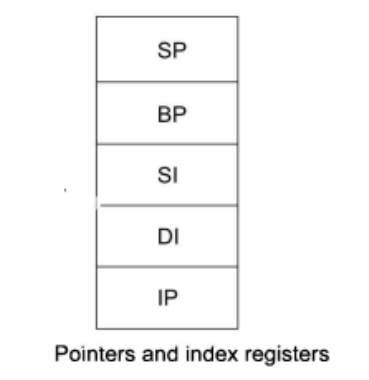
- 16-bit
- Points to the current stack.
- The 20-bit physical stack address is calculated from the Stack Segment (SS) and the Stack Pointer (SP) for stack instructions such as PUSH and POP.
- In based addressing mode, the 20-bit physical stack address is calculated from the Stack segment (SS) and the Base Pointer (BP).

Extra Segment Register(ES)

- 16-bit
- Points to the extra segment in which data (in excess of 64K pointed to by the DS) is stored.
- String instructions use the ES and DI to determine the 20-bit physical address for the destination.

Pointers and Index Registers

- Pointers and index registers contain offsets of data and instructions.
- All segment registers are 16-bit wide. But it is necessary to generate 20-bit address (physical address) on the address bus.
- To get 20-bit physical address one or more pointer or index registers are associated with each segment register.



Pointers and index registers

- Stack Pointer(SP):
  - used to accesses the stack segment
  - it points the program stack that means SP stores the base address of the Stack Segment.
- Instruction Pointer(IP)
  - contains offset within the code segment
  - The Instruction Pointer is a register that holds the address of the next instruction to be fetched from memory.

- Base Pointer(BP)
  - contains offset within the data segment
  - can be used to access data in other segments
- Source Index (SI)
  - used to store the offset of source data in data segment
  - when string operations are performed the SI register point to memory locations in the data segment which is addressed by the DS register.
- Destination Index(DI)
  - used to store the offset of destination in data or extra segment also used for string operations

Flag (Status) Register

- is a Flip-Flop (FF) which indicates some condition produced by the execution of an instruction or controls certain operations of the EU.
- The 8086 flag register contents indicate the results of computation in the ALU. It also contains some flag bits to control the CPU operations
- A 16 bit flag register is used in 8086. It is divided into two parts .
  - Condition code or status flags
  - Machine control flags

Flag Register

Auxiliary Carry Flag

This is set, if there is a carry from the lowest nibble, i.e, bit three during addition, or borrow for the lowest nibble, i.e, bit three, during subtraction.

Carry Flag

This flag is set, when there is a carry out of MSB in case of addition or a borrow in case of subtraction.

Sign Flag

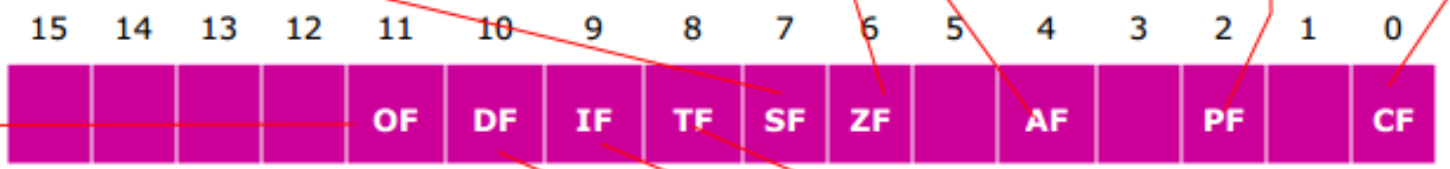
This flag is set, when the result of any computation is negative

Zero Flag

This flag is set, if the result of the computation or comparison performed by an instruction is zero

Parity Flag

This flag is set to 1, if the lower byte of the result contains even number of 1's ; for odd number of 1's set to zero.



Over flow Flag

This flag is set, if an overflow occurs, i.e, if the result of a signed operation is large enough to accommodate in a destination register. The result is of more than 7-bits in size in case of 8-bit signed operation and more than 15-bits in size in case of 16-bit sign operations, then the overflow will be set.

Tarp Flag

If this flag is set, the processor enters the single step execution mode by generating internal interrupts after the execution of each instruction

Direction Flag

This is used by string manipulation instructions. If this flag bit is '0', the string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto incrementing mode.

Interrupt Flag

Causes the 8086 to recognize external mask interrupts; clearing IF disables these interrupts.

Flag register example

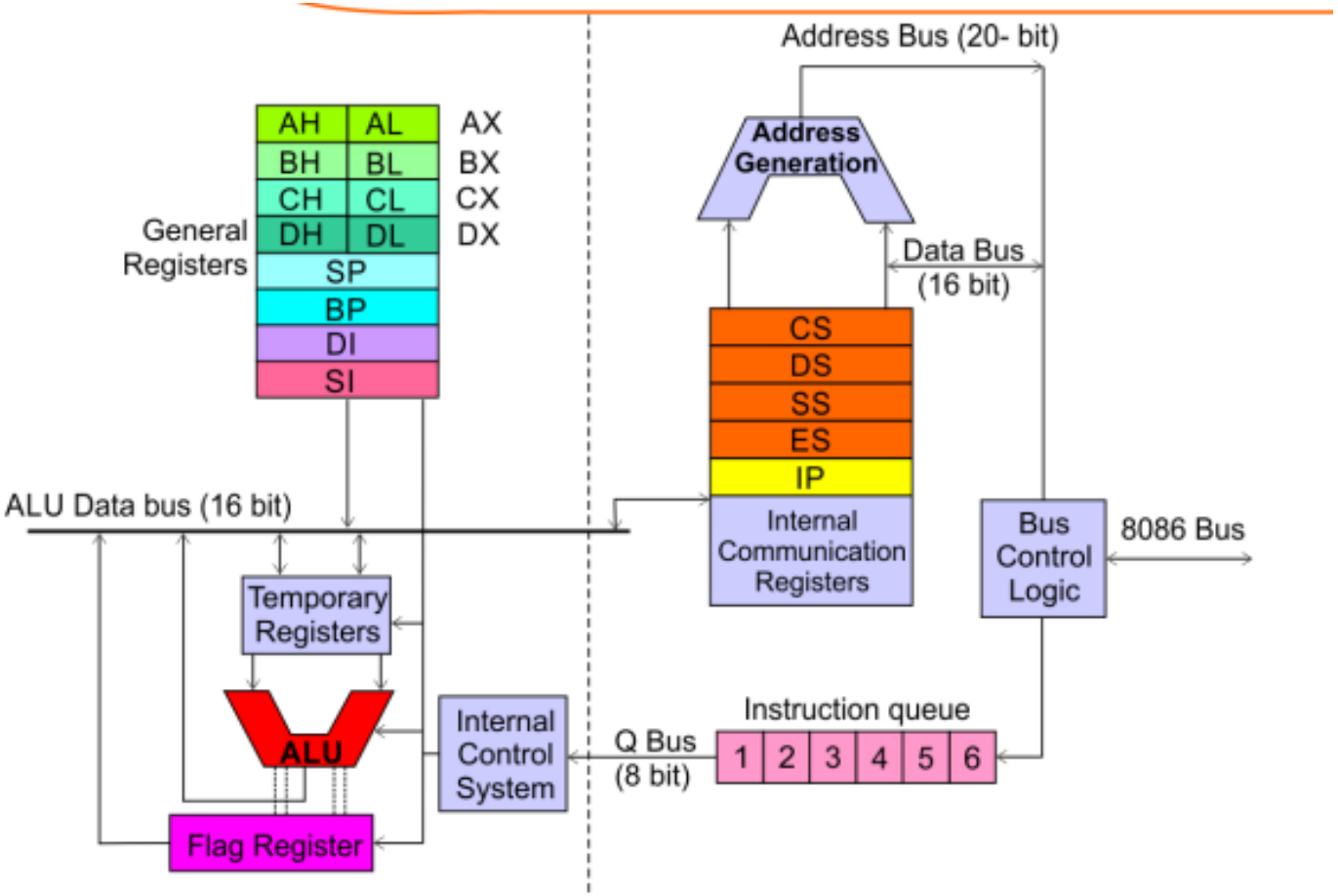
- Write the state of status flag for 8086 microprocessor
  - Add **0ffh** and **04h**
  - Add **64h** and **32h**
  - Subtract **0ffh** from **02h**
  - Add **4h** and **7fh**
  - Add **3h** and **7h**
  - subtract **1h** from **8h**

architecture of 8086

- The architecture of 8086 includes
  - Arithmetic Logic Unit (ALU)
  - Flags, General registers, Instruction byte queue, Segment registers
- It is internally divided into two separate functional units.
- These are the **Bus Interface Unit (BIU)** and the **Execution Unit (EU)**.
- These two functional units can work simultaneously to increase system and hence throughput.

- The throughput is a measure of number of instructions executed per unit time.
- The major reason for this separation is to increase the processing speed of the processor.
- The BIU has to interact with memory and input and output devices in fetching the instructions and data required by the EU
- EU is responsible for executing the instructions of the programs and to carry out the required processing.

Architecture Diagram



Execution Unit (EU)

EU executes instructions that have already been fetched by the BIU.

Bus Interface Unit (BIU)

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/ O ports.

Bus Interface Unit (BIU)

- The BIU is the 8086 interface to the outside world.
- It provides a full 16-bit bi-directional data bus and 20-bit address bus.
- performing all external bus operation like It sends address of the memory or I/O, It fetches instruction from memory, It reads data from port/memory, It writes data into port/memory, It supports instruction queuing, It provides the address relocation facility

- it contains memory address and data bus interface logic, segment registers,memory addressing logic,address conversion mechanisms and six byte instruction object code queue
- **Instruction queue:**is First-In First-Out group of registers in which up to six bytes of instruction codes are pre-fetched from memory
- **Memory Address and data bus interface logic:**of the BIU generates all the bus control signals for memory and I/O.

- **Address conversion mechanisms:**
  - which used to produce 20-bit address.
  - it generated using segment and offset registers each of the size 16-bit.
  - the content of segment register called as segment address
  - the content of an offset register also called offset address
  - to get the total physical address put the lower nibble 0H to segment address and add to offset address

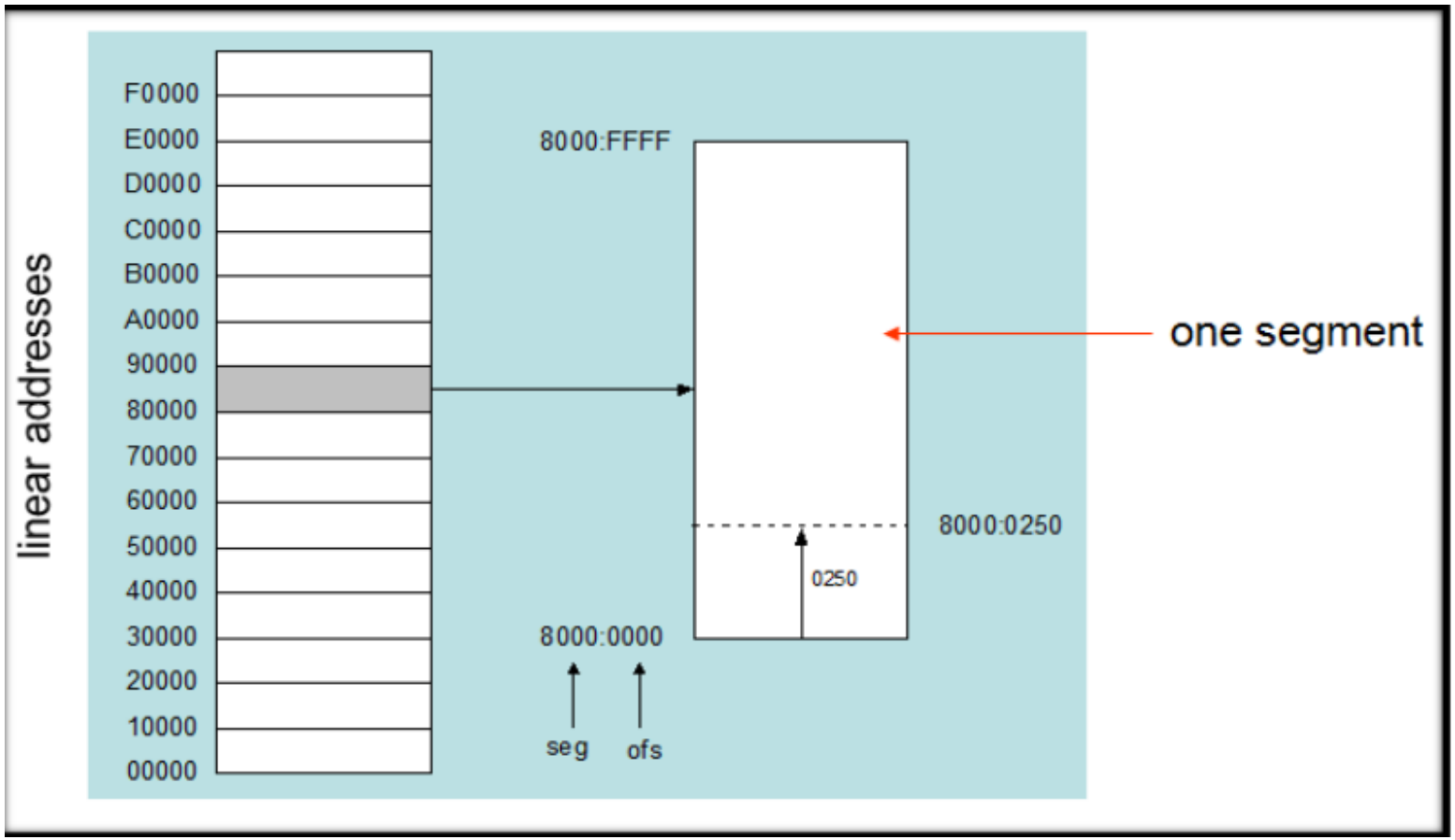
Execution Unit(EU)

- execute instructions that have already been fetched by BIU
- The EU of 8086 tells the BIU from where to fetch instructions or data, decodes instructions and execute instructions. It contains:
  - Instruction Decoder: translate the instructions
  - Arithmetic Logic Unit (ALU)
  - Flag Register
  - General Purpose Registers
  - Pointers and Index Registers
  - Timing and Control circuit: generate control signals for internal and external operations of the microprocessor

Memory segmentation

- It is the process in which the main memory of computer is divided into different segments and each segment has its own base address.
- it is used to increase the execution speed of computer system so that processor can able to fetch and execute the data from memory easily and fastly

Segmentation



cont..

- The size of address bus of 8086 is 20 and is able to address 1 Mbytes ( ) of physical memory.
- The compete 1 Mbytes memory can be divided into 16 segments, each of 64 Kbytes size.
- The addresses of the segment may be assigned as 0000H to F000H respectively.
- he offset values are from 0000H to FFFFFH.

Types of Segmentation

- **overlapping segment:**
  - A segment starts at a particular address and its maximum size can go up to 64 Kbytes. But if another segment starts along this 64 Kbytes location of the first segment, the two segments are said to be overlapping segment.
  - The area of memory from the start of the second segment to the possible end of the first segment is called as overlapped segment.
- **Non Overlapped Segment:**
  - A segment starts at a particular address and its

maximum size can go up to 64 Kbytes. But if another segment starts before this 64 Kbytes location of the first segment, the two segments are said to be Non- overlapping segment.

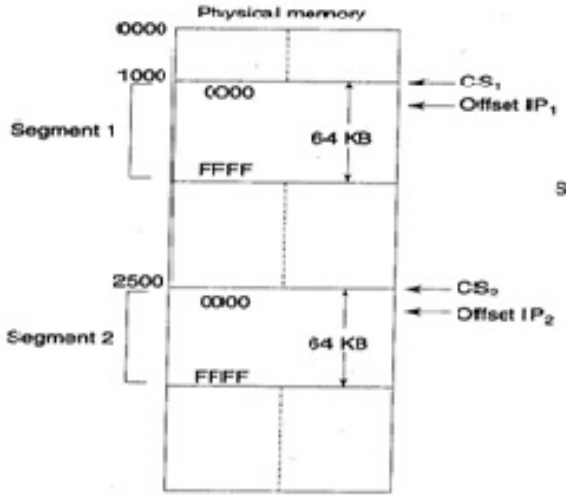


Fig: Non-overlapping Segments

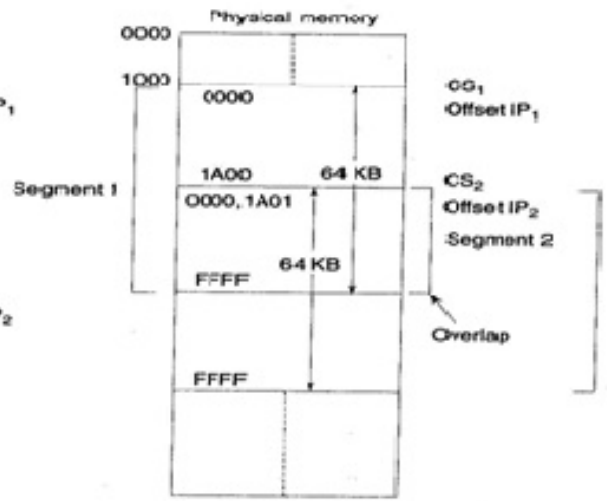


Fig: overlapping segment

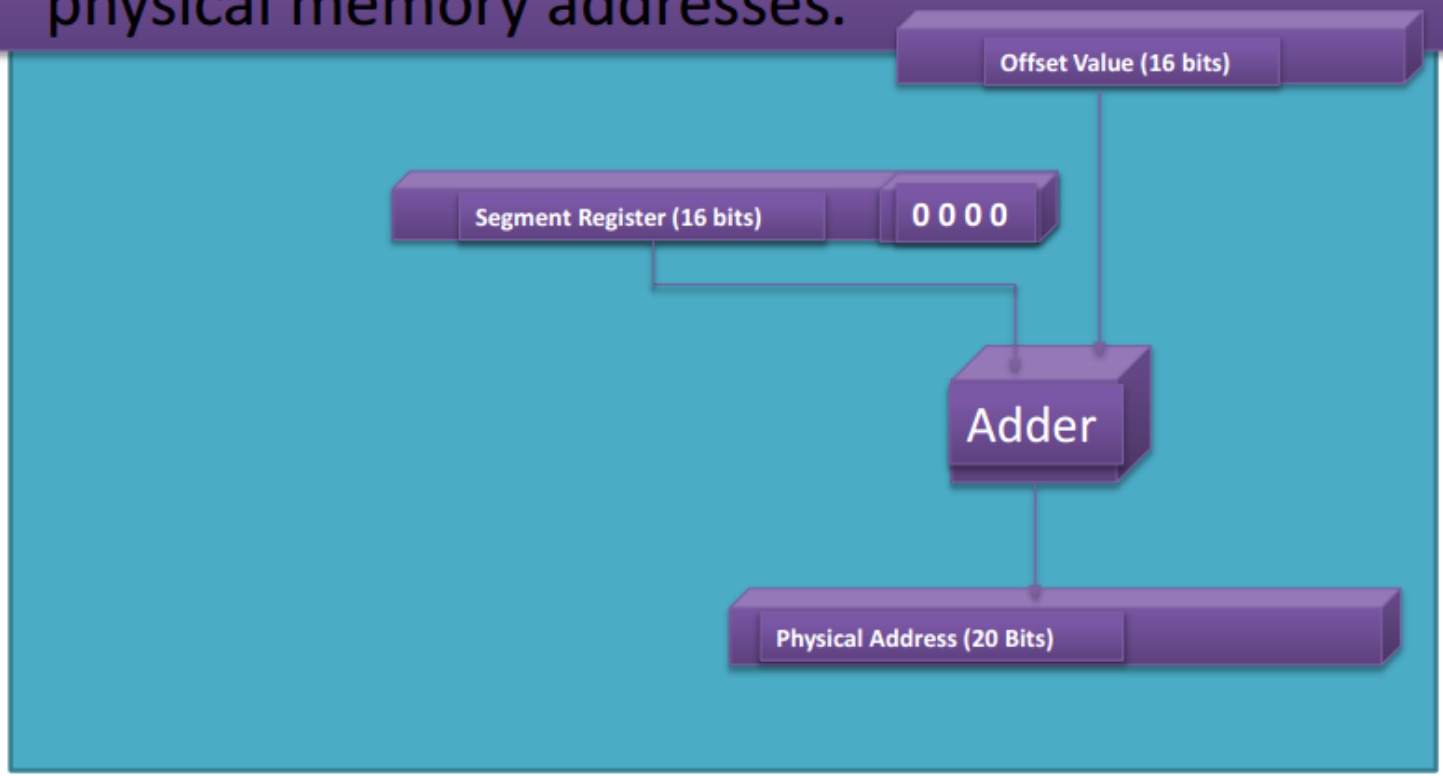
Advantages of segmentation

The main advantages of the segmented memory scheme are as follows:

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

Memory Address Generation

- The BIU has a dedicated adder for determining physical memory addresses.

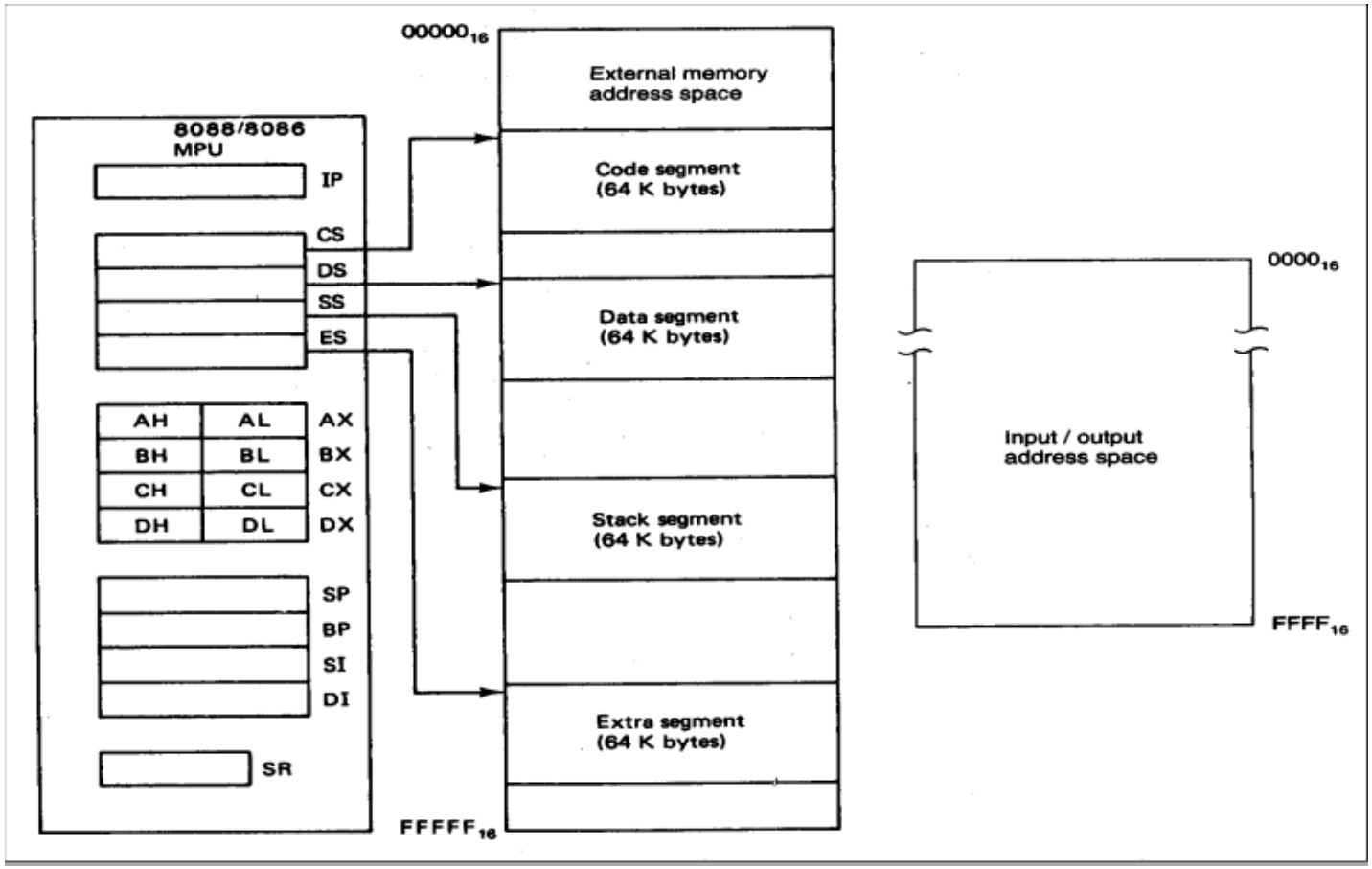


Segment : Offset Address

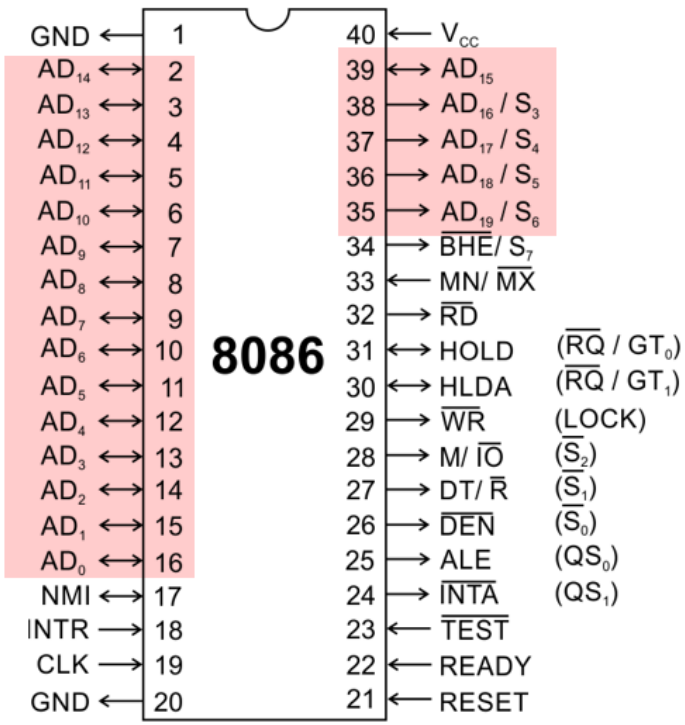
- Logical Address is specified as **segment: offset**
- Physical address is obtained by shifting the segment address 4 bits to the left and adding the offset address.
- Thus the physical address of the logical address **A4FB:4872** is:

```
  A4FB0
+ 4872
  A9822
```

Segmented memory representation



Signal Description of 8086



AD_0-AD_{15} (Bidirectional)

Address/Data bus

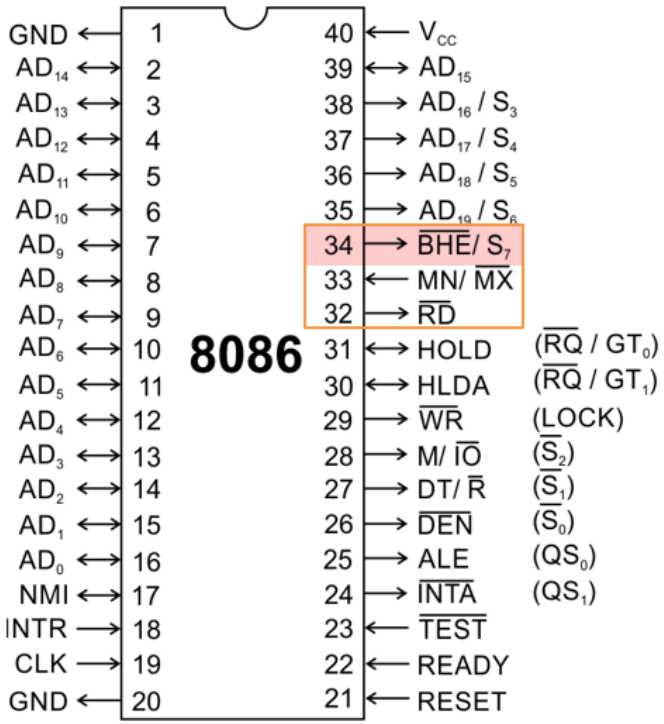
Low order address bus; these are multiplexed with data.

When AD lines are used to transmit memory address the symbol A is used instead of AD, for example A_0-A_{15}.

When data are transmitted over AD lines the symbol D is used in place of AD, for example D_0-D_7, D_8-D_{15} or D_0-D_{15}.

A_{16}/S_3, A_{17}/S_4, A_{18}/S_5, A_{19}/S_6

High order address bus. These are multiplexed with status signals



BHE (Active Low)/S_7 (Output)

Bus High Enable/Status

It is used to enable data onto the most significant half of data bus, D_8-D_{15}. 8-bit device connected to upper half of the data bus use BHE (Active Low) signal. It is multiplexed with status signal S_7.

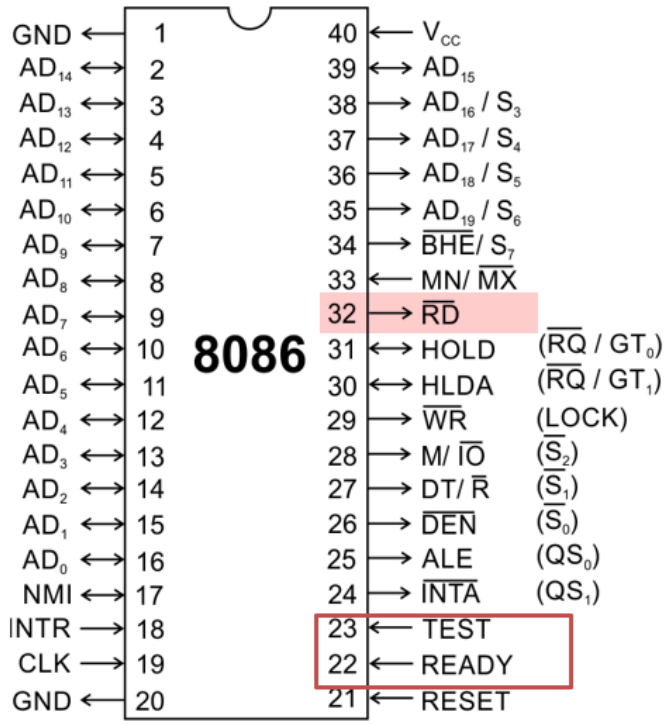
MN/ MX

MINIMUM / MAXIMUM

This pin signal indicates what mode the processor is to operate in.

RD (Read) (Active Low)

The signal is used for read operation.
It is an output signal.
It is active when low.



TEST

TEST input is tested by the 'WAIT' instruction.

8086 will enter a wait state after execution of the WAIT instruction and will resume execution only when the TEST is made low by an active hardware.

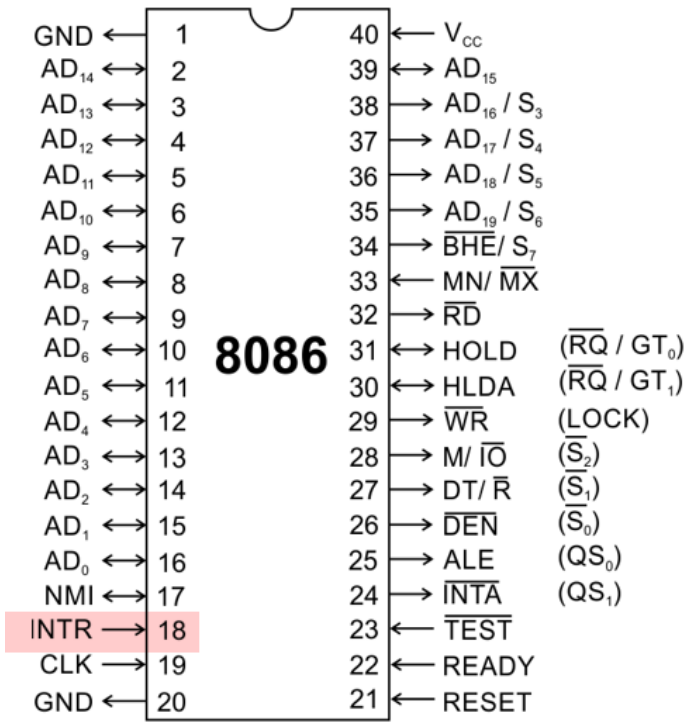
This is used to synchronize an external activity to the processor internal operation.

READY

This is the acknowledgement from the slow device or memory that they have completed the data transfer.

The signal made available by the devices is synchronized by the 8284A clock generator to provide ready input to the 8086.

The signal is active high.



RESET (Input)

Causes the processor to immediately terminate its present activity.

The signal must be active HIGH for at least four clock cycles.

CLK

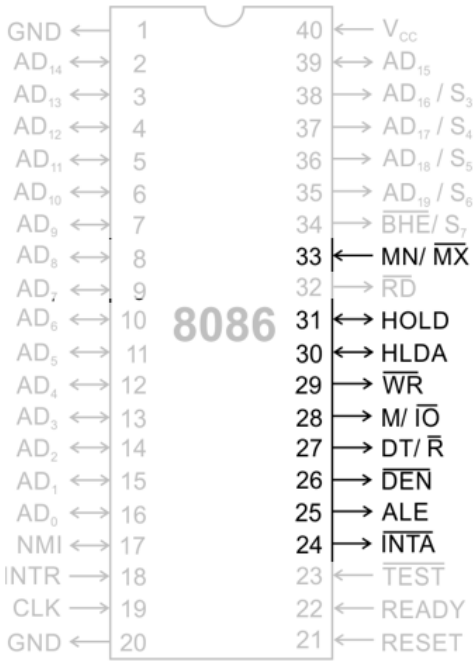
The clock input provides the basic timing for processor operation and bus control activity. Its an asymmetric square wave with 33% duty cycle.

INTR Interrupt Request

This is a triggered input. This is sampled during the last clock cycles of each instruction to determine the availability of the request. If any interrupt request is pending, the processor enters the interrupt acknowledge cycle.

This signal is active high and internally synchronized.

Min/ Max Pins



The 8086 microprocessor can work in two modes of operations : Minimum mode and Maximum mode.

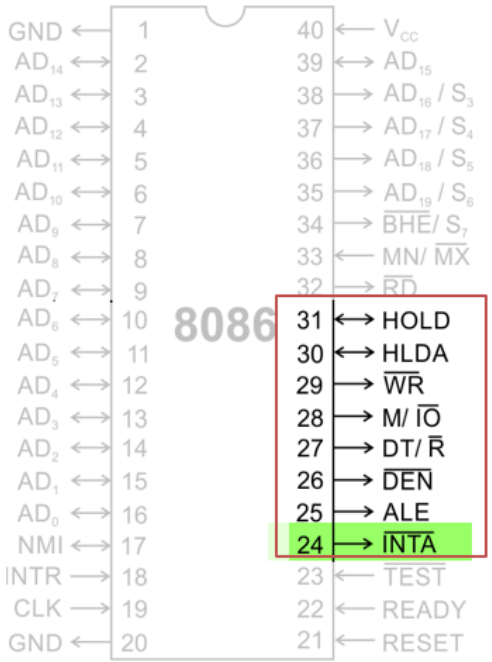
In the minimum mode of operation the microprocessor do not associate with any co-processors and can not be used for multiprocessor systems.

In the maximum mode the 8086 can work in multi-processor or co-processor configuration.

Minimum or maximum mode operations are decided by the pin MN/ MX(Active low).

When this pin is high 8086 operates in minimum mode otherwise it operates in Maximum mode.

Min Pins



Pins 24 -31

For minimum mode operation, the MN/ $\overline{MX}$ is tied to VCC (logic high)

8086 itself generates all the bus control signals

DT/$\overline{R}$ — **(Data Transmit/ Receive) Output signal from the processor to control the direction of data flow through the data transceivers**

$\overline{DEN}$ — **(Data Enable) Output signal from the processor used as out put enable for the transceivers**

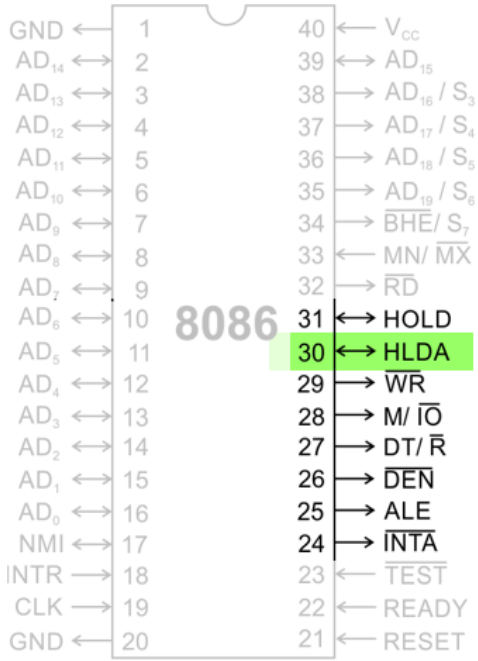
ALE — **(Address Latch Enable) Used to demultiplex the address and data lines using external latches**

M/$\overline{IO}$ — **Used to differentiate memory access and I/O access. For memory reference instructions, it is high. For IN and OUT instructions, it is low.**

$\overline{WR}$ — **Write control signal; asserted low Whenever processor writes data to memory or I/O port**

$\overline{INTA}$ — **(Interrupt Acknowledge) When the interrupt request is accepted by the processor, the output is low on this line.**

Min Pins



Pins 24 -31

For minimum mode operation, the MN/ $\overline{MX}$ is tied to VCC (logic high)

8086 itself generates all the bus control signals

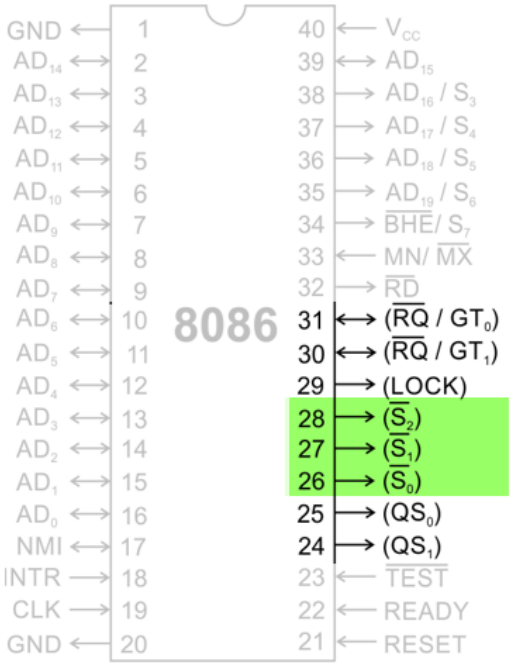
HOLD — **Input signal to the processor form the bus masters as a request to grant the control of the bus.**

Usually used by the DMA controller to get the control of the bus.

HLDA — **(Hold Acknowledge) Acknowledge signal by the processor to the bus master requesting the control of the bus through HOLD.**

The acknowledge is asserted high, when the processor accepts HOLD.

Max Pins



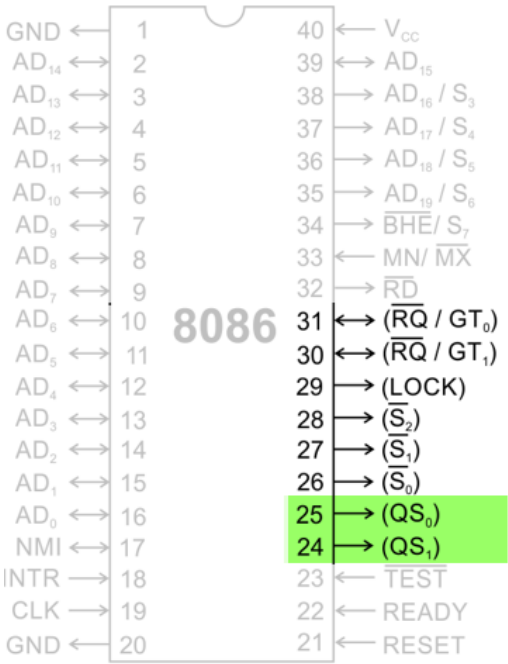
During maximum mode operation, the MN/ $\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned

$\overline{S_0}$, $\overline{S_1}$, $\overline{S_2}$ **Status signals; used by the 8086 bus controller to generate bus timing and control signals. These are decoded as shown.**

Status Signal			Machine Cycle
$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	
0	0	0	Interrupt acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive/Inactive

Max Pins



During maximum mode operation, the MN/ $\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned

$\overline{QS_0}$, $\overline{QS_1}$ **(Queue Status) The processor provides the status of queue in these lines.**

The queue status can be used by external device to track the internal status of the queue in 8086.

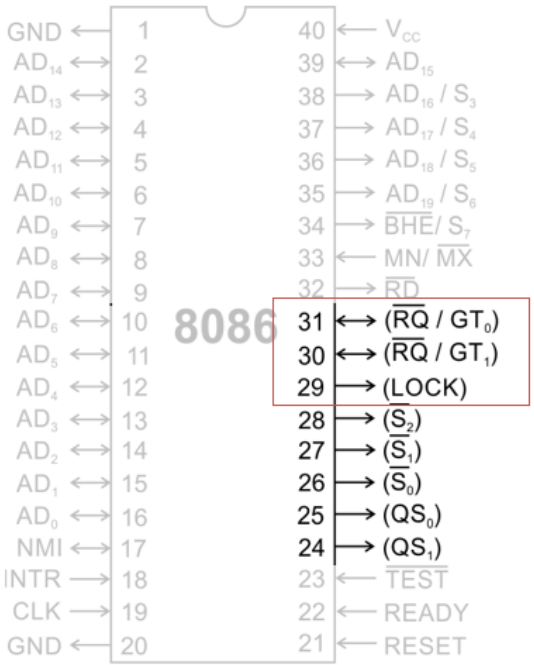
The output on QS_0 and QS_1 can be interpreted as shown in the table.

Queue status		Queue operation
QS_1	QS_0	
0	0	No operation
0	1	First byte of an opcode from queue
1	0	Empty the queue
1	1	Subsequent byte from queue

Max Pins

During maximum mode operation, the MN/ $\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned



$\overline{RQ}/\overline{GT_0}$, $\overline{RQ}/\overline{GT_1}$

(Bus Request/ Bus Grant) These requests are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle.

These pins are bidirectional.

The request on $\overline{GT_0}$ will have higher priority than $\overline{GT_1}$

$\overline{LOCK}$

An output signal activated by the LOCK prefix instruction.

Remains active until the completion of the instruction prefixed by LOCK.

The 8086 output low on the $\overline{LOCK}$ pin while executing an instruction prefixed by LOCK to prevent other bus masters from gaining control of the system bus.

Physical Memory Organization

- 8086 supports 2pow(20) = 1,048,576 bytes (1Mbytes) of memory over the address range 0000016 to FFFFF16 (00000H to FFFFFH)
- Two consecutive bytes can be accessed as one word (16-bits)
- The lower-addressed byte is the least significant byte of the word, and the higher-addressed byte is its most significant byte

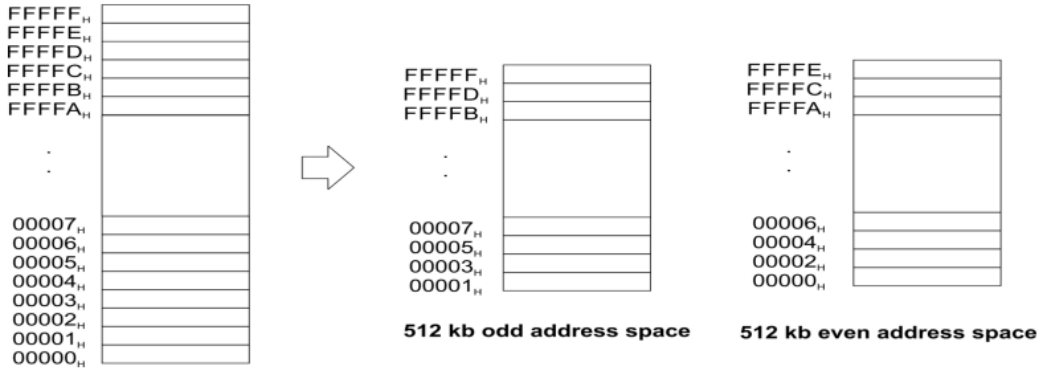
cont..

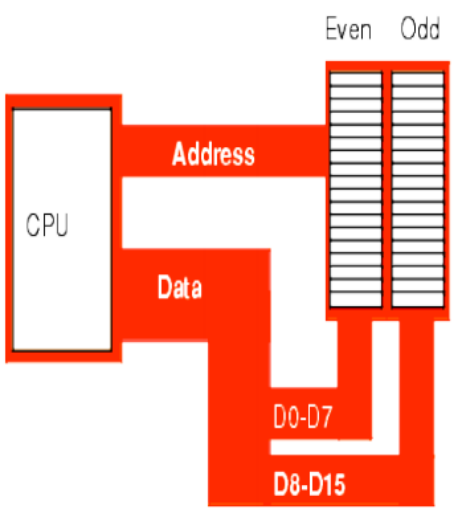
- Address of lower byte of word is called address of the full word
- Memory space of 1MB is divided into two chips (called banks) of 512KB each having even & odd addresses
- This is done because most memories are ‘byte-oriented’ (1 byte read/write at a time), but as 8086 is capable to read/write 16-bit (2 bytes) at a time, we need to use two chips for 16-bit word operations

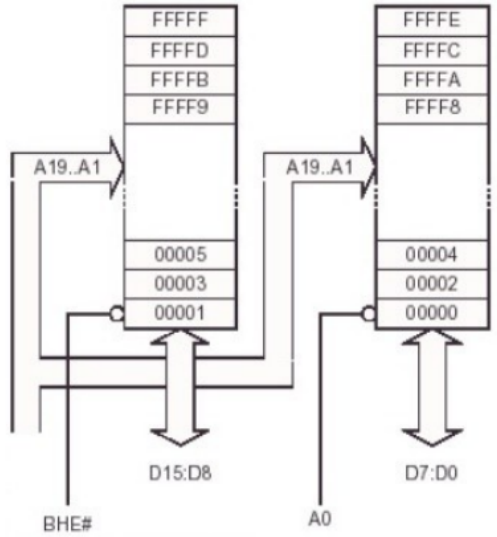
cont..

- Two banks have alternate addresses because 16-bit words are stored in consecutive locations & at the same time 8086 must access both banks simultaneously for 16-bit operations
- If we had used only 1 chip of 1MB, 8086 could read only 1 byte at a time & thus would need two operations (bus cycles) for every 16-bit word operation. Thus process would have been slower.

cont..


FFFFF$_H$
FFFFE$_H$
FFFFD$_H$
FFFFC$_H$
FFFFB$_H$
FFFFA$_H$
00007$_H$
00006$_H$
00005$_H$
00004$_H$
00003$_H$
00002$_H$
00001$_H$
00000$_H$
512 kb odd address space
512 kb even address space

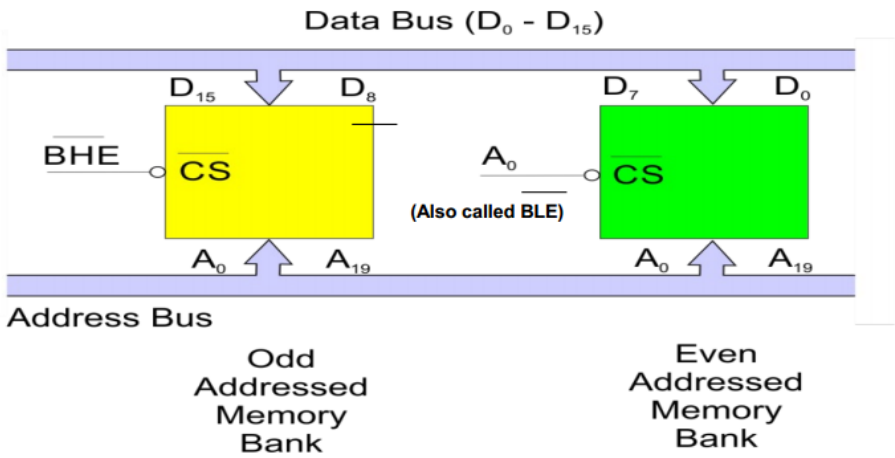

Even
Odd
CPU
Address
Data
D0-D7
D8-D15


FFFFF
FFFFD
FFFFB
FFFF9
FFFFE
FFFFC
FFFFA
FFFF8
A19..A1
A19..A1
00005
00003
00001
00004
00002
00000
D15:D8
D7:D0
BHE#
A0

cont..

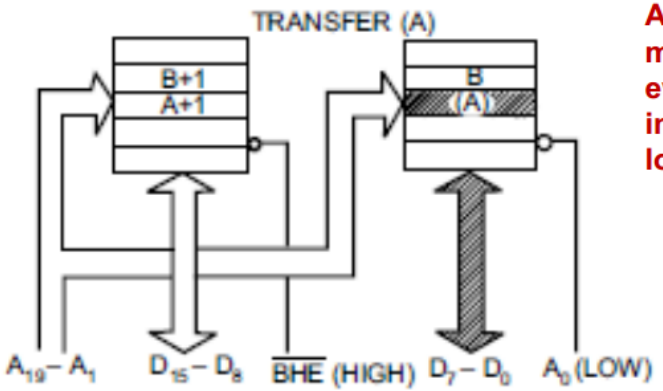
8086 MEMORY ORGANIZATION

	Operation	$\overline{BHE}$	A_0	Data Lines Used
1	Read/ Write byte at an even address	1	0	$D_7 - D_0$
2	Read/ Write byte at an odd address	0	1	$D_{15} - D_8$
3	Read/ Write word at an even address	0	0	$D_{15} - D_0$
4	Read/ Write word at an odd address	0	1	$D_{15} - D_0$ in first operation byte from odd bank is transferred
		1	0	$D_7 - D_0$ in first operation byte from odd bank is transferred



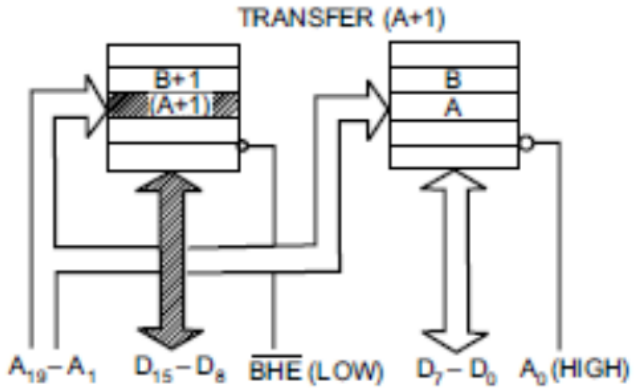
cont..

BYTE/WORD TRANFER IN 8086



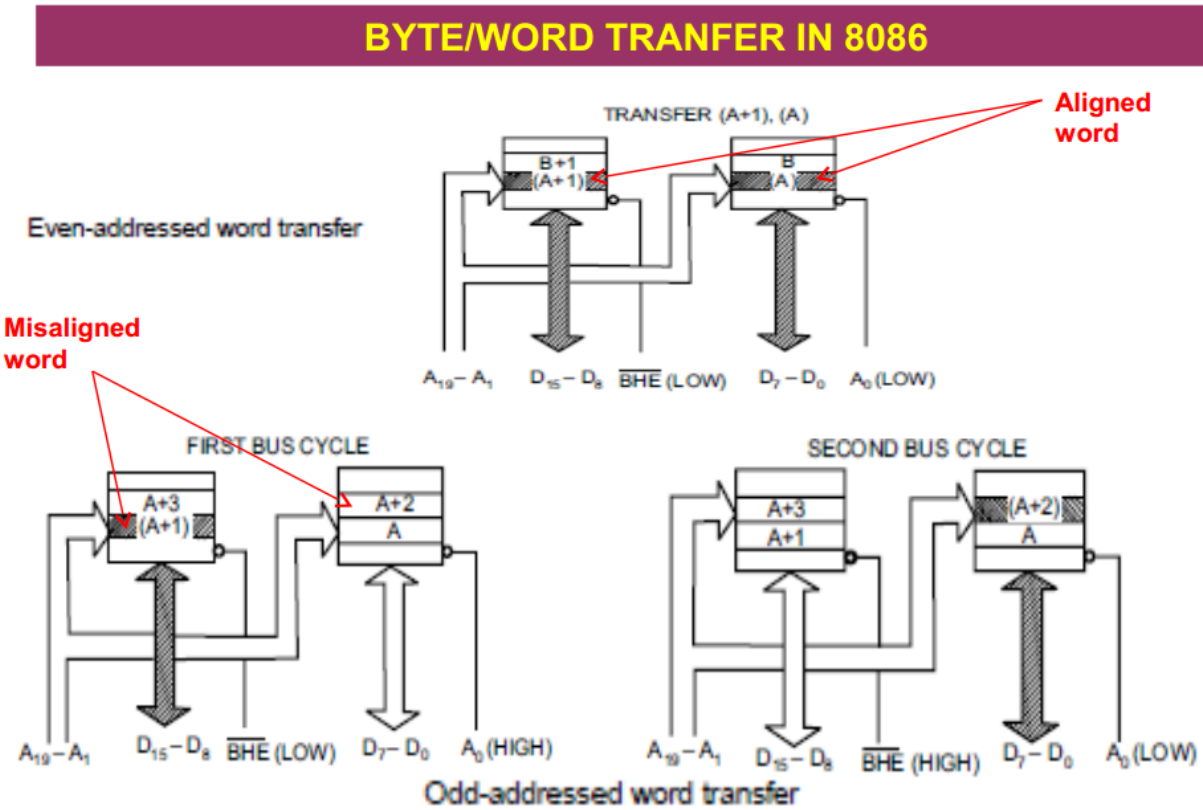
A, B = any assumed memory location address in even, odd banks. Brackets indicate contents of location & not address

Even-addressed byte transfer



Odd-addressed byte transfer

cont..

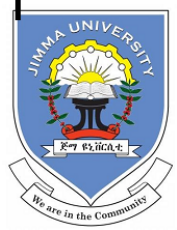


A word stored at an even-address boundary (00000H, 00002H, 00004H etc) is said to be an aligned word while a word stored at an odd address boundary (00001H, 00003H, 00005H etc) is called misaligned word. A misaligned word requires two read operations (or bus cycles) & thus double time

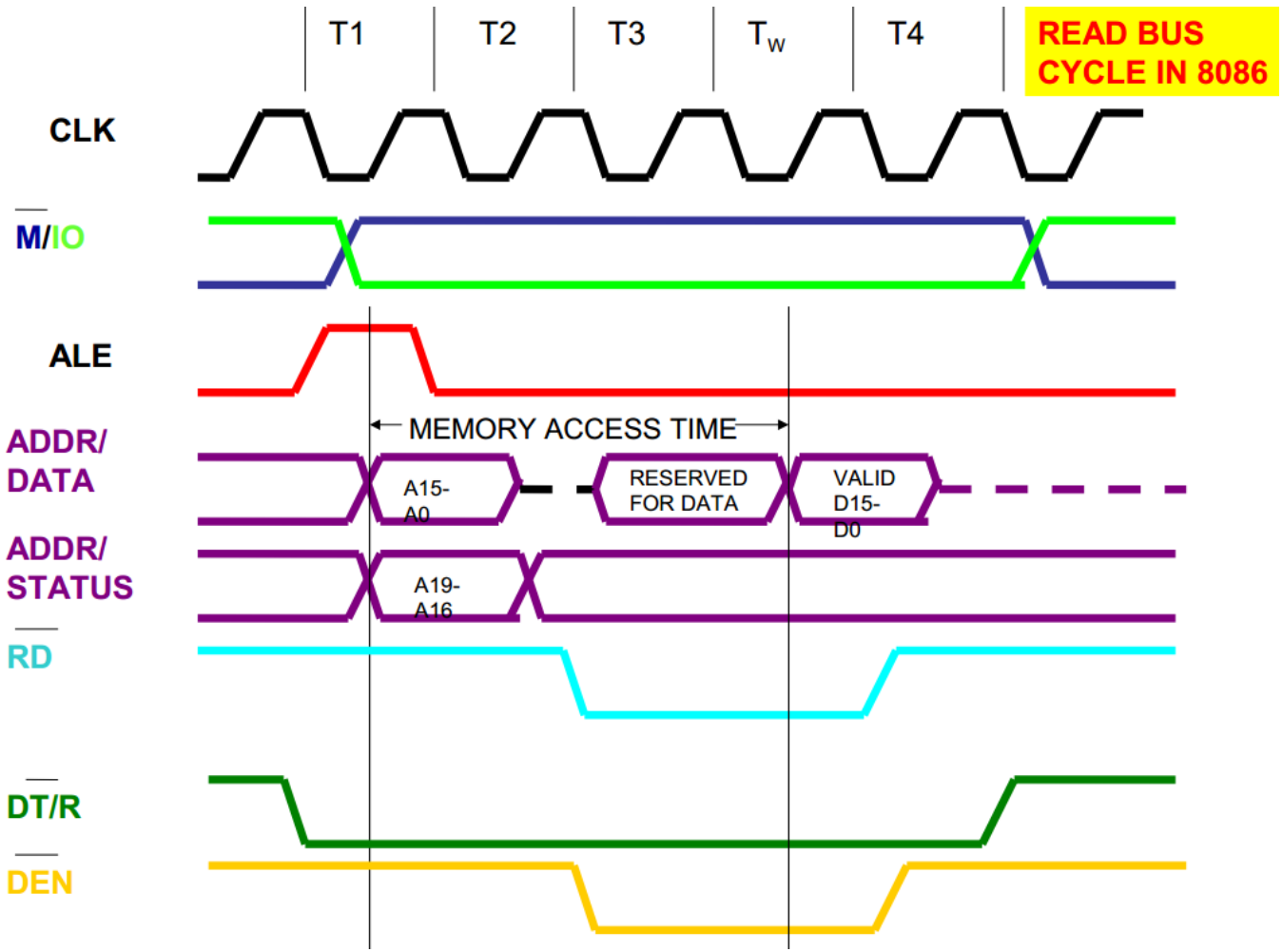
General Bus Operation

- The 8086 has a combined address and data bus commonly referred as a time multiplexed address and data bus.
- The main reason behind multiplexing address and data over the same pins is the maximum utilization of processor pins and it facilitates the use of 40 pin standard DIP package.
- The bus can be demultiplexed using a few latches and transceivers, whenever required.

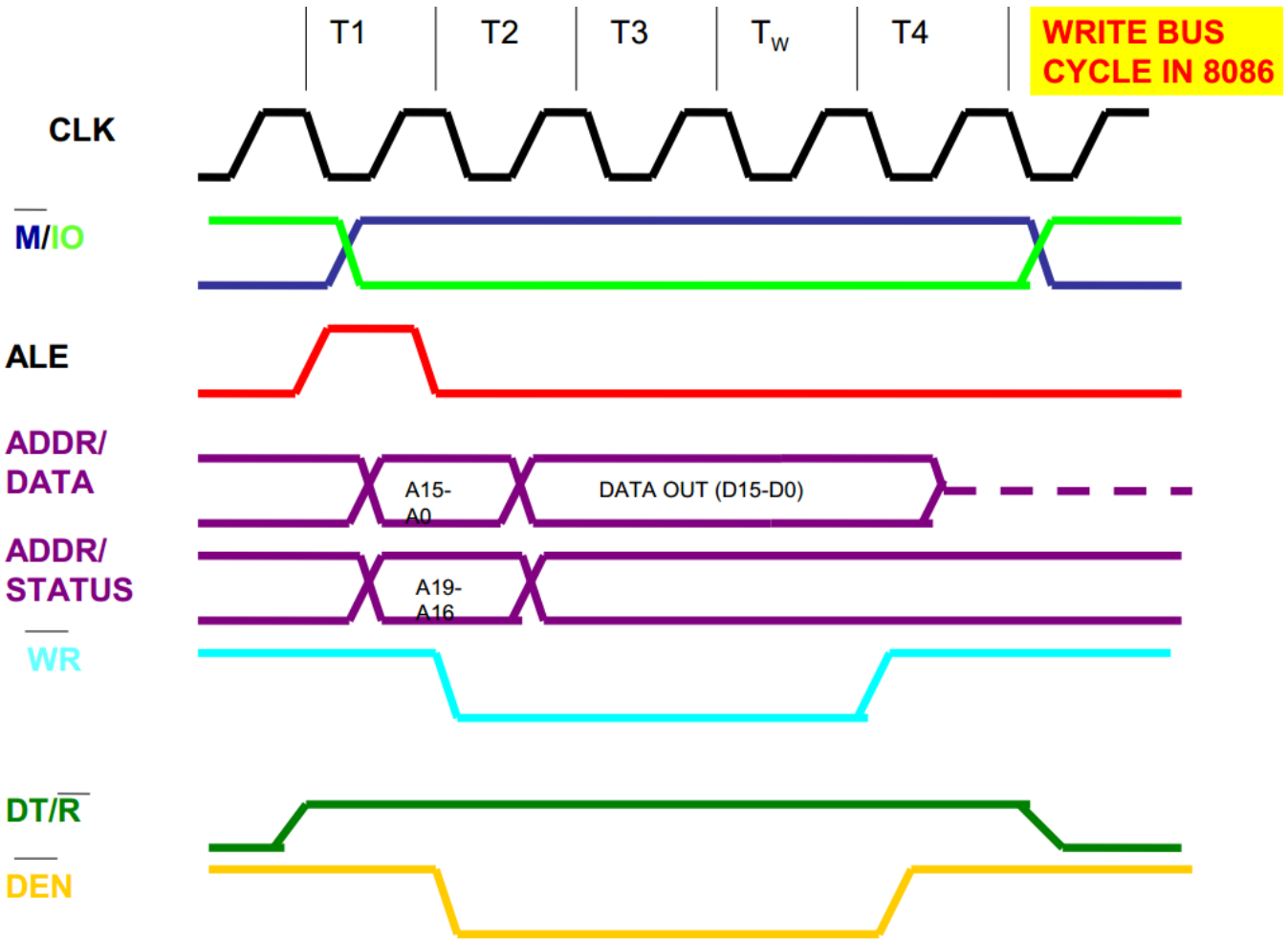
- Basically, all the processor bus cycles consist of at least four clock cycles. These are referred to as T1, T2, T3, and T4. The address is transmitted by the processor during T1. It is present on the bus only for one cycle.
- T2, i.e. the next cycle, is the data read cycle
- The data transfer takes place during T3 and T4. In case, an addressed device is slow and shows NOT READY status the wait states Tw are inserted between T3 and T4.
- The address latch enable (ALE) signal is emitted during T1 by the processor (min mode) or the bus controller (max mode) depending upon the status of the MN/MX input.



General Bus read operation cycle



General Bus write operation cycle

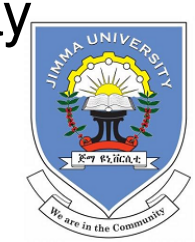


Minimum Mode Operation

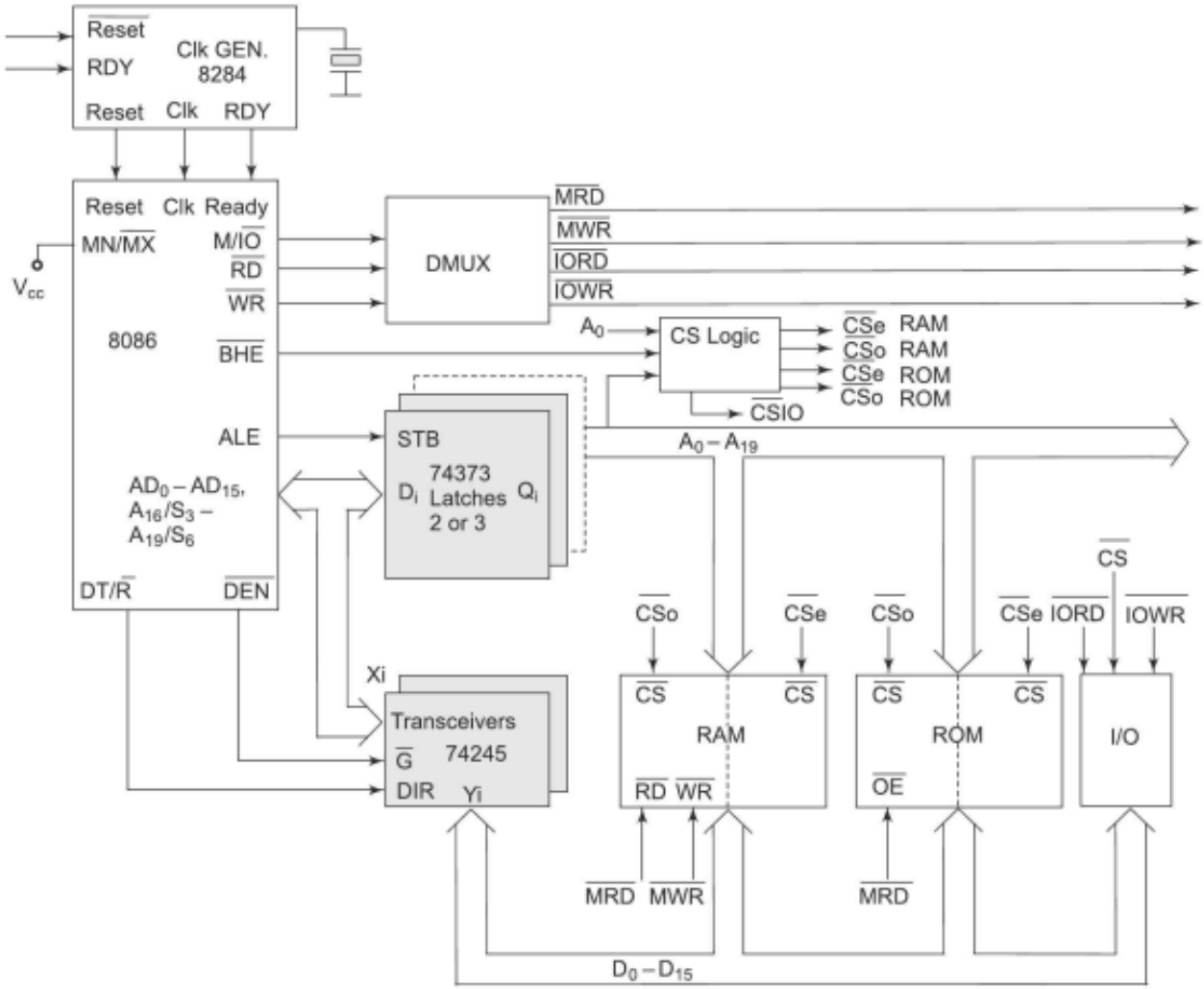
- This is a single microprocessor configuration selected by applying logic 1 to the MN / MX input pin.
- The remaining components in the system are latches, transceivers, clock generator, memory and I/O devices.
  - Latches are used to separate valid address from address/data signals controlled by ALE
  - Transceivers are bidirectional buffers Also termed as data amplifiers Controlled by DEN or DT/R
  - No interfacing or master/slave signals is required.

cont..

- No bus controller required.
- The DEN signal indicates the direction of data, i.e. from or to the processor.
- The system contains memory for the monitor and users program storage. Usually, EPROM are used for monitoring storage, while RAM users program storage. a system may contain I/O devices



minimum mode system

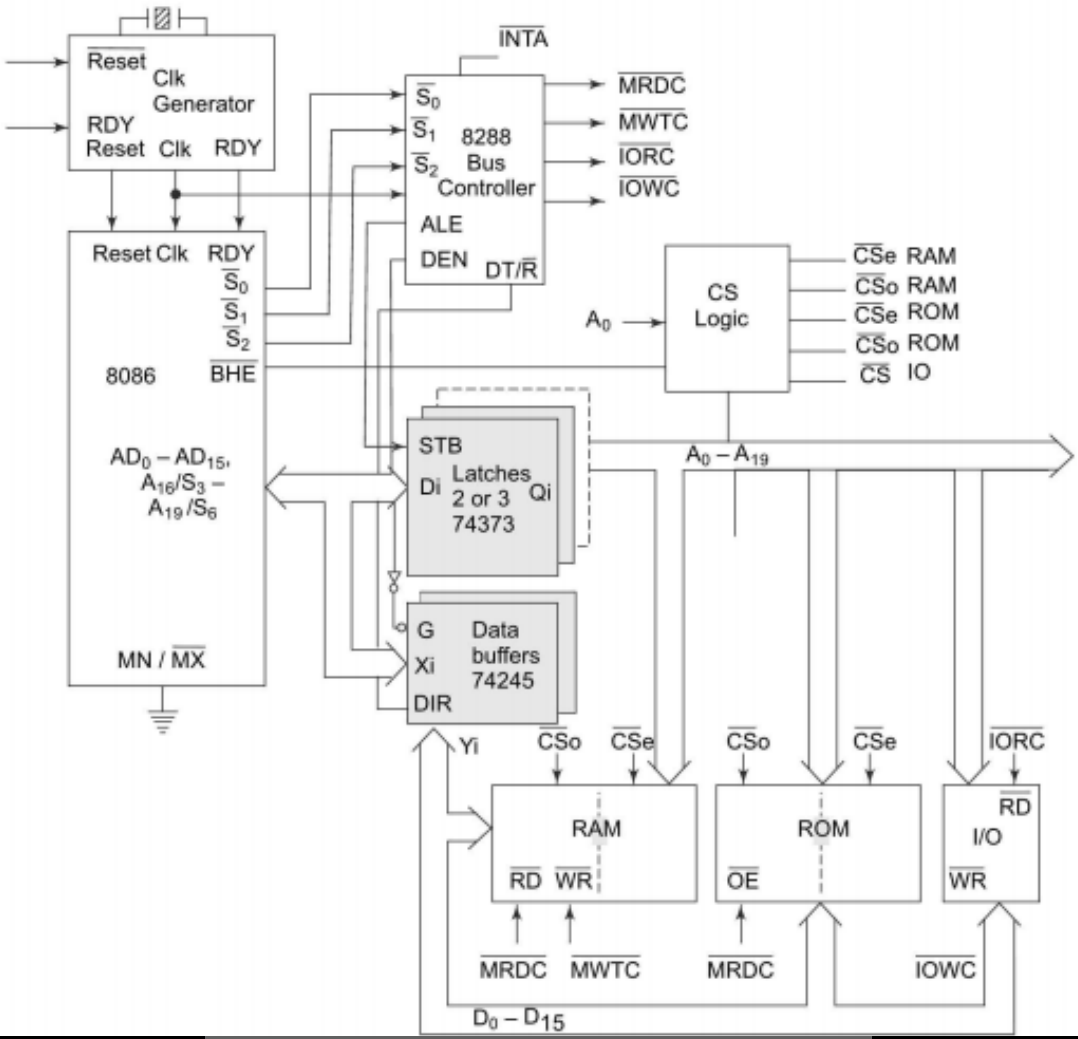


Maximum mode Operations

- MN/MX connect to Ground
- Some control signals are generated externally by the 8288 bus controller chip
- 8086 generates QS1, QS0, S0, S1, S2, LOCK, RQ/GT1, RQ/GT0 control signals.
- Max mode is used when math processor is used.
- Master/slave, multiplexing and several such control signals are required.
- 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.
  - 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 ports.
  - The MRDC, MWTC are memory read command and memory write command signals respectively and may be used as memory read or write signals.

max mode system



Thank you