Microprocessors and Microcontrollers

Evolution of Microprocessors:

A microprocessor with a 4-bit ALU will be referred to as a 4-bit microprocessor.

- Intel 4004 (1971)-first commercially available microprocessor.
  - Contained 2300 PMOS transistors.

- Intel 8008 (1972)-8 bit words.
  - Required 20 or more additional device to form a functional CPU.

- Intel 8080 (1974)-larger instruction set.
  - Required only two additional device to form a functional CPU.
  - Used NMOS transistors (faster).
  - Referred to a second generation microprocessor.

- MC 6800 (Motorola)-8 bit general purpose CPU.
  - Required only +5v supply

Evolution of Microprocessors in three major directions:

1. Dedicated or Embedded Controllers:
   - Used to control “smart” machines such as microwave ovens, cloth washers, sewing machines etc.

     Eg: TMS-1000 family of 4 bit microprocessors produced by Texas instruments.
     - Intel 8048-8 bit CPU, RAM, ROM &some I/O ports.
       - Referred to as microcontrollers.

     Eg: intel 8051, Motorola MC6801, INTEL 8096.

2. Bit slice processor:

- Device which can be used to build a custom CPU.

  Eg: AMD 2900 family of devices.

  - Includes 4 bit ALUs, multiplexers, sequencers and other parts needed for custom building a CPU.

GENERAL PURPOSE CPUs :

- INTEL 8085 -8 BIT PROCESSOR
- MOTOROLA MC6809 - 8 BIT PROCESSOR
- INTEL 8086(1978) - BIT PROCESSOR
- MOTOROLA MC68000 - 16 BIT PROCESSOR
- INTEL 8088, 80186, 80188, 80286, 80386, 80486 ETC.

The main limitations of 8 bit microprocessors were their low speed, low memory addressing capability, limited number of general purpose registers and a less powerful instruction set.

The 8086 microprocessor has a much more powerful instruction set along with the architectural developments which imparts substantial programming flexibility and improvement in speed over the 8 bit microprocessor.

REGISTER ORGANISATION OF 8086

1. General Data Registers

- AX, BX, CX and DX are the general purpose 16-Bit registers.
- AH→ Higher 8 Bits
- AL→ Lower 8 Bits
- AL→ Used as an 8 Bit accumulator for 8-Bit operations.

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

2. Segment Registers

The complete 1 MB memory is divided into 16 logical segments.

Each segments contains 64 KB of memory.

4 segments registers –

- Code Segment Register (CS)
- Data Segment Register (DS)
- Extra Segment Register (ES)
- Stack Segment Register (SS)

CS Register → Used for addressing a memory location in the code segment of the memory, where the executable program is stored.

DS Register → Points to the data segment of the memory, where data is stored.

ES Register → ES also contains data.

SS Register → Used for addressing stack segment of the memory. Memory used to store stack data.

The CPU uses the stack for temporarily storing important data.

Example: Contents of the CPU registers which will be required at a later stage.

AH AL BH BL CH CL DH DL

- While addressing any location in the memory bank, the physical address is calculated from two parts.
    a) Segment address

b) Offset

- The segment register contain 16-Bit segment base addresses, related to different segments.
- Any of the pointers and index registers or BX may contain offset of the location to be addressed.

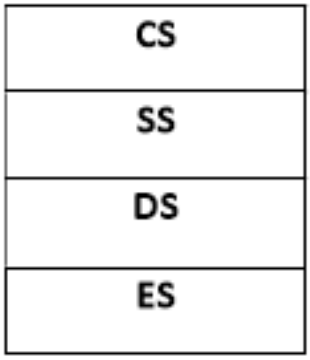


Fig: Segment Register

Pointers and Index Registers

- The index register usually contain offset within the code and stack segments. IP (code), BP& SP (stack segments)
- The index registers are used as general purpose registers as well as for offset storage in case of indexed, based indexed and relative-based indexed addressing modes.
- The register SI is generally used to store the offset of source data in data segment while the register DI is used to store the offset of determination in data or extra segment

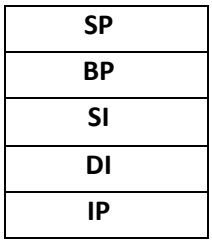


Fig: Pointers and Index Register

Architecture of 8086

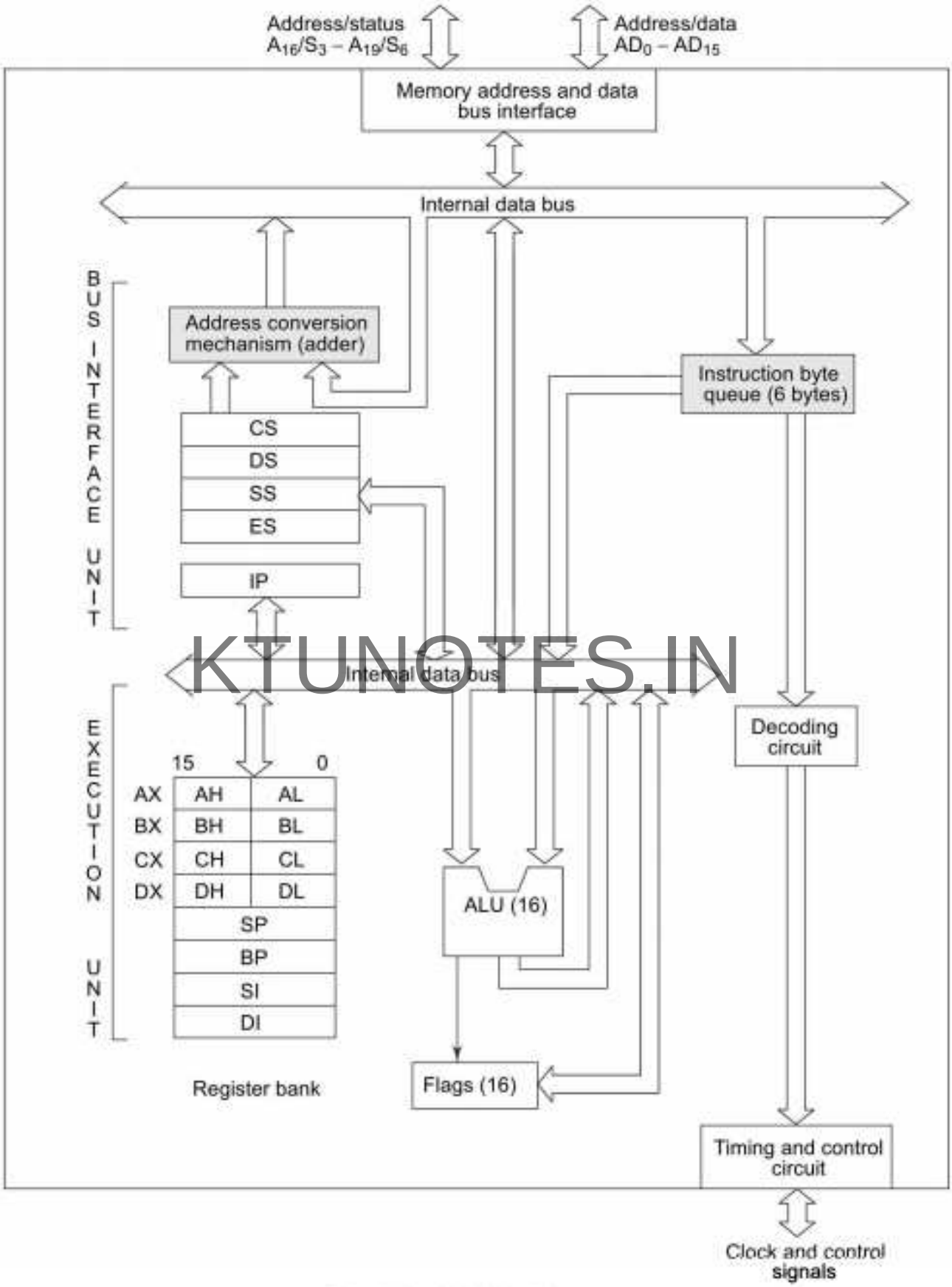


Fig. 1.2 *8086 Architecture*

The architecture of 8086 can be divided into 2 parts as follows:

- Bus Interface Unit (BIU)
- Execution Unit (EU)

BIU

BIU contains the circuit for physicial address calculation and a predecoding instuction byte queue(6 bytes long). BIU makes systems bus signal available for external interfacing devices. BIU is responsible for establishing communications with external devices and perhapials including memory via bus communications.

Physical Address Calculations

8086 addresses a segmented memory. The complete physicial address which is 20 bits long is generated using segment and offset registers. The content of a segment register (segment address) is shifted left bitwise four times and added with contents of offset register (offset address) to generate 20 bit physical address

Segment Address + Offset Address = Physical Address(Left Shifted)

Eg : If Segment address is 1005H and offset address is 5555H then the physical address will be generated as

Segment Address = 1005H = 0001 0000 0000 0101

Segment Address = 0001 0000 0000 0101 0000 (Left Shifted 4 times)

Offset Address = 5555H = 0101 0101 0101 0101

Physicial Address = Segment Address +Offset Address (Left Shifted 4 Times)

= 0001 0000 0000 0101 0000 + 0101 0101 0101 0101

= 0001 0101 0101 1010 0101

= 155A5

Segment addressed by the segment value 1005H can have offset value from 0000H to FFFFH within it i.e maximum 64K locations can be allocated for segment

Segment Address : base address of a particular segment

Offset Address : indicates the distance of required memory locations in the segment from base address

Offset is a 16-bit number, each segment can have a maximum of 216 (64K) locations

BIU has a seprate adder to perform physical address calculation by adding segment and offset address.

- Segment address value is content of code,data,stack, or extra segment register.
- Offset may be the content of IP,AX,BX,SI,DI,SP,BP or an immediate 16-bit value.
- The external bus is used to fetch the machine code of the next instruction and arrange it in a queue known as precoded instruction byte queue. It is 6 bytes long and FIFO structure.
- The instructions from the queue are taken from the queue are taken for decoding sequentially.
- Once a byte is decoded, the queue is rearranged by pushing it out and queue status is checked for possibilty of the next opcode fetch cycle.
- While BIU fetches opcode, the Execution Unit executes previously decoded instruction concurrently, BIU & EU together forms a pipeline.
- BIU manages complete interface of execution unit with memory and I/O devices,under the control of timing & control unit.

Execution Unit

- Contains the register set of 8086 except segment register and IP
- It has a 16-bit ALU, which performs arithmetic and logic functions.
- It has 16-bit flag register reflects the result of execution.
- The decoding unit decodesthe opcode bytes issued from the insruction byte queue.

- The timing and control circuit unit derives nessaray control signals to execute instruction opcode reccived from the queue, depending upon the information from the decoding circuit.
- The EU may pass the results to the BIU for storing them in memory.

MEMORY SEGMENTATION IN 8086

- In segmentation memory scheme ,the complete physically available memory may be divided into a number of logical segments .
- Each segment is 64k Bytes in size and is addressed by one of the segment registers.
- The 16 bit contents of the segment register actually point to the starting location of a particular segments.
- An offset address is needed to address a specific memory location within the segments.
- The offset address is 16 bit long so that maximum offset value can be FFFF H.
- The CPU 8086 is able to address 1M bytes of physical memory .
- The complete 1M bytes of memory can be divided into 16 segments , each of 64K bytes size .
- Addresses of the segments maybe assigned as 0000H to F000H .
- Offset address values are from 0000H to FFFFH .
- Physically address range 00000H to FFFFFH .

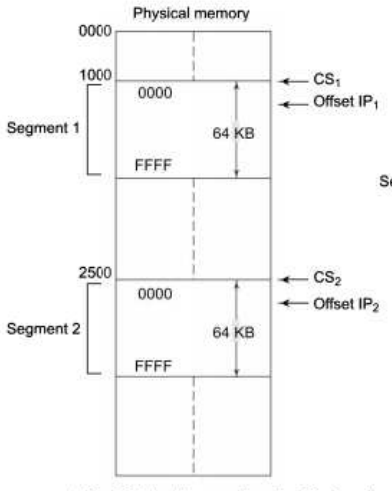


Fig. 1.3(a) *Non-overlapping Segments*

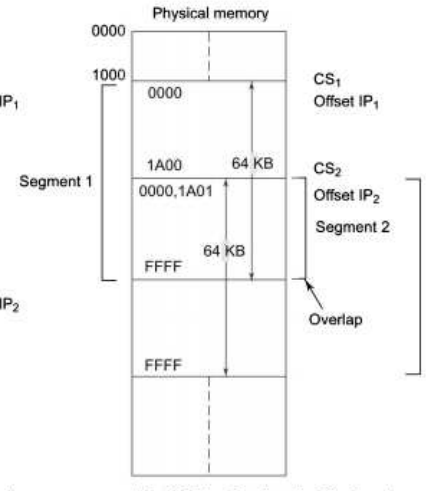


Fig. 1.3(b) *Overlapping Segments*

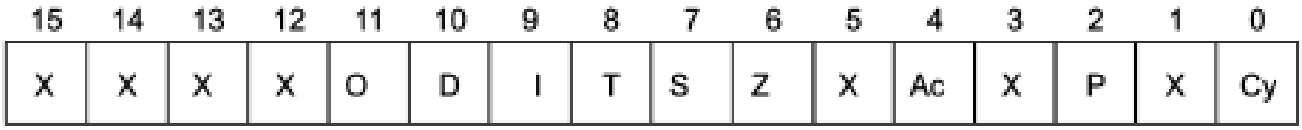
- Segments may be overlapping . suppose a segment starts at a particular address and its maximum size can be 64K bytes. But if another segments starts before this 64K bytes locations of the first segment , the two segments are said to be overlapping .
- The area of memory from the start of the second segment to the possible end of the first segment is called an overlapped segment area
- Physical address in = CS,+IP,=CS2+IP2 overlapped area locations

ADVANTAGES OF SEGMENTED MEMORY SCHEME

1. Allows the memory capacity to be 1MB although the actual addresses to be handled are of 16 bit size.
2. Allows the placing of code,data and stack portions of the same program in different parts(segments) of memory each time the programme is executed i.e provision for relocation is done.
3. Permits a program and data to be put into different area of memory each time the program is executed i.e. provision for relocation is done

Flag Register - 8086

- Indicates the results of computation in the ALU
- Also contains some flag bits to control the CPU operations



O — Overflow flag
D — Direction flag
I — Interrupt flag
T — Trap flag
S — Sign flag
Z — Zero flag
Ac — Auxiliary carry flag
P — Parity flag
Cy — Carry flag
X — Not used

Fig. 1.4 *Flag Register of 8086*

S Sign Flag – This flag is set when the result of any computation is negative.For signed computations the sign flag equals the MSB of the result.

Z Zero Flag – This flag is set if the result of the computation or comparison performed by the previous instruction is zero.

Parity Flag – This flag is set to 1 if the lower byte of the result contains even number of 1's.

Carry Flag – This flag is set when there is a carry out of MSB in the case of addition or borrow in the case of subtraction.

Trap Flag – If this flag is set a trap interrupt is generated after execution of each instruction. The processor executes the current instruction and the control is transferred to the trap interrupt service routine.

Interrupt Flag – If this flag is set the mask-able interrupts are recognized by the CPU.

Direction Flag – This is used by string manipulation instruction. If this flag bit is 0, the string is processed beginning from the lowest address to the highest address. i.e., auto decrementing mode otherwise the string is processed in the auto increment mode.

AC – Auxiliary carry flag

This flag is set if there is a carry from the lowest nibble (bit three) during addition or borrow for the lowest nibble during subtraction.

Overflow flag -This flag is set if an overflow occur i.e if the result of signed operation is large enough to accommodated in a destination register.

> **example**: In case of addition of two signed numbers, if the result overflows onto the sign bit i.e the result is of more than 7-bits in size in case of 8-bits signed operations and more than 15-bits in size in case of 16-bits signed operations, then overflow flag will be set.

SIGNAL DESCRIPTION OF 8086

- 8086 is available in three clock rates -5 MHz ,8 MHz and 10MHz
- Packaged in a 40 pin CERDIP/plastic package.

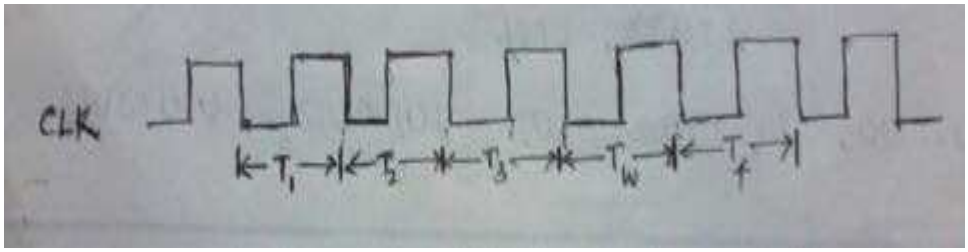
8086 signals can be categorized in three groups

- Signals having common functions in minimum as well as maximum mode.
- Signals having special functions for minimum mode.
- Signals having special functions for maximum mode.

PIN CONFIGURATION – Refer diagram in textbook/next page

AD_{15} – AD_0:-

- Multiplexed memory I/O address and data lines.
- Address remains on the lines during T_1 state, while data is available on the data bus during T_2 ,T_3 , Tw and T_4.
- T_1,T_2,T_3,T_4 and Tw are the clock states of a machine cycle.
- Tw is a wait state.
- These lines are active high and float to a tri-state (high impedance state) during interrupt acknowledge and local bus acknowledge cycles.



A_{19}/S_6 ,A_{18}/S_5, $A_{17}/S4$, A_{16}/S_3

- Time multiplexed address and status lines.
- During T_1 , these are the most significant address lines for memory operations.
- During I/O operations, these lines are low.
- During memory or I/O operations, status information is available on those lines for T_2,T_3,Tw and T_4.
- S_6 is always low.
- S_5 denotes the status of interrupt enable flag bit and is updated at the beginning of each clock cycle.
- S_4 and S_3 together indicate which segment register is presently being used for memory accesses.
- These lines float to tri-state off during the local bus acknowledge.
- The address bits are separated from the status bits using latches controlled by the ALE signal.

S_4	S_3	Indications
0	0	Alternate data
0	1	Stack
1	0	Code
1	1	Data

$\overline{BHE}$ / S_7:- Bus High Enable / Status

- Indicate the transfer of data over the higher order (D_{15} - D_8)data.
- It goes low for the data transfers over D_{15}-D_8 and is used to derive chip select of odd address memory bank or peripherals.

- $\overline{BHE}$ is low during T_1 for read ,write and interrupt acknowledge cycles , whenever a byte is to be transferred on the higher byte of the data bus.
- The status information is available during T_2,T_3,and T_4.
- Status is active low and is tri-stated during 'hold'.

$\overline{BHE}$	A_0	Indication
0	0	Whole word(2 byte)
0	1	Upper byte from or to odd address
1	0	Lower byte from or to even address
1	1	None

$\overline{RD}$ – Read :-

- Indicates the peripheral that the processor is performing a memory or I/O operation.
- Active low –T_2,T_3,Tw of read cycle.
- Tri-stated during hold acknowledge.

READY:-

- Acknowledgement from the slow devices or memory that they have completed data transfer.
- Signal is active high.

INTR:- Interrupt Request

- Sampled during last cycle of each instruction to determine the availability of the request.
- If any interrupt request is pending, the processor enters the interrupt acknowledge cycle.

- Internally marked by resetting the interrupt enable flag.

$\overline{\text{TEST}}$:-

- This input is examined by a WAIT instruction.
- If the $\overline{\text{TEST}}$ input goes low, execution will continue, else the processor remains in an idle state.

NMI:-Non Maskable Interrupt

- Causes Type-2 interrupt.
- Not maskable internally by software.
- Transition from low to high initiates the interrupt response at the end of each instruction.

RESET:-

- Causes the processor to terminate the current activity and start execution from FFFFOH.
- Active high for atleast 4 clock cycles.
- Restarts execution when the RESET returns low.

CLK- Clock input:-

- Provide the basic timing for processor operation and bus control activity.
- Asymmetric square wave with 33% duty cycle.

V_{CC}:- +5 power supply for the operation of the internal circuit.

GND:- ground for the internal circuit.

MN/$\overline{MX}$:- minimum / maximum mode operation .

Minimum Mode Operation- Pin Function

M/$\overline{I/O}$:-

- Memory I/O (logically equivalent to S_2 in maximum mode).
- Low indicates that CPU is having an I/O operation.
- High indicates that CPU is having a memory operation.
- Active in the previous T_4 and remains active till final T_4 of the current cycle.
- Tri-stated during local bus “hold acknowledge”.

$\overline{INTA}$:- Interrupt Acknowledge

- Read strobe for interrupt acknowledge cycles.
- Low means processor has accepted the interrupt.
- Active low during T_2,T_3 and Tw of each interrupt acknowledge cycle.

ALE:-

- Indicate the availability of the valid address on address/data lines.
- Connected to latch enable input of latches.
- Active high and is never tri-stated.

DT/$\overline{R}$:- Data Transmit/Receive

- To decide the direction of data flow through transreceivers (bi-directional buffers).
- High when processor sends out data.
- Low when processor receives data.
- Tri-stated during ‘hold acknowledge’.

$\overline{DEN}$:-Data Enable

- Indicates the availability of valid data over the address/data lines.
- Enable the data transreceivers to separate the data from the multiplexed address/data signal.
- Active from middle of T_2 to T_4.
- Tri-stated during ‘hold acknowledge’ cycle.

HOLD, HLDA:- Hold/Hold Acknowledge

- Hold line goes high indicates to the processor that another master is requesting the bus access.
- The processor, after receiving the HOLD request, issues the HLDA signal in the middle of the next clock cycle after completing the current bus cycle.
- At the same time, the processor floats the local bus and control lines.
- When the processor deletes the HOLD line low, it lowers the HLDA signal.

❖ If a DMA request is made while the CPU is performing a memory or I/O cycle, it will release the local bus during T_4 provided.

1. The request occurs on or before T2 state of the current cycle.
2. The current cycle is not operating over the lower byte of the word (or operating on an odd address).
3. The current cycle is not the first acknowledge of an interrupt acknowledge sequence.
4. A lock instruction is not being executed.

MAXIMUM MODE: SIGNAL DESCRIPTION

S_2, S_1, S_0 – status lines

- Indicates the type of operation carried out by the processor.
- Becomes active during T_4 of the previous cycle and remain active during T_1 and T_2 of the current bus cycle.
- The status lines return to passive state during T_3 of the current bus cycle so that they may again become active for the next bus cycle during T_4.
- Any change in these lines during T_3 indicates the starting of a new cycle and return to passive state indicates end of the bus cycle.

S2	S1	S0	Characteristics
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

Lock

- indicates that other system bus masters will be prevented from gaining the system bus, while the lock signal is low.
- Activated by the 'lock' prefix instruction and remains active until the completion of the next instruction.
- This floats to tri state off during hold acknowledge.

QS_1, QS_0- queue status

- Status of the code - prefetch queue.
- Active during the clk cycle after which the queue operation is performed.

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

RQ/GT_0, RQ/GT_1----Request/grant

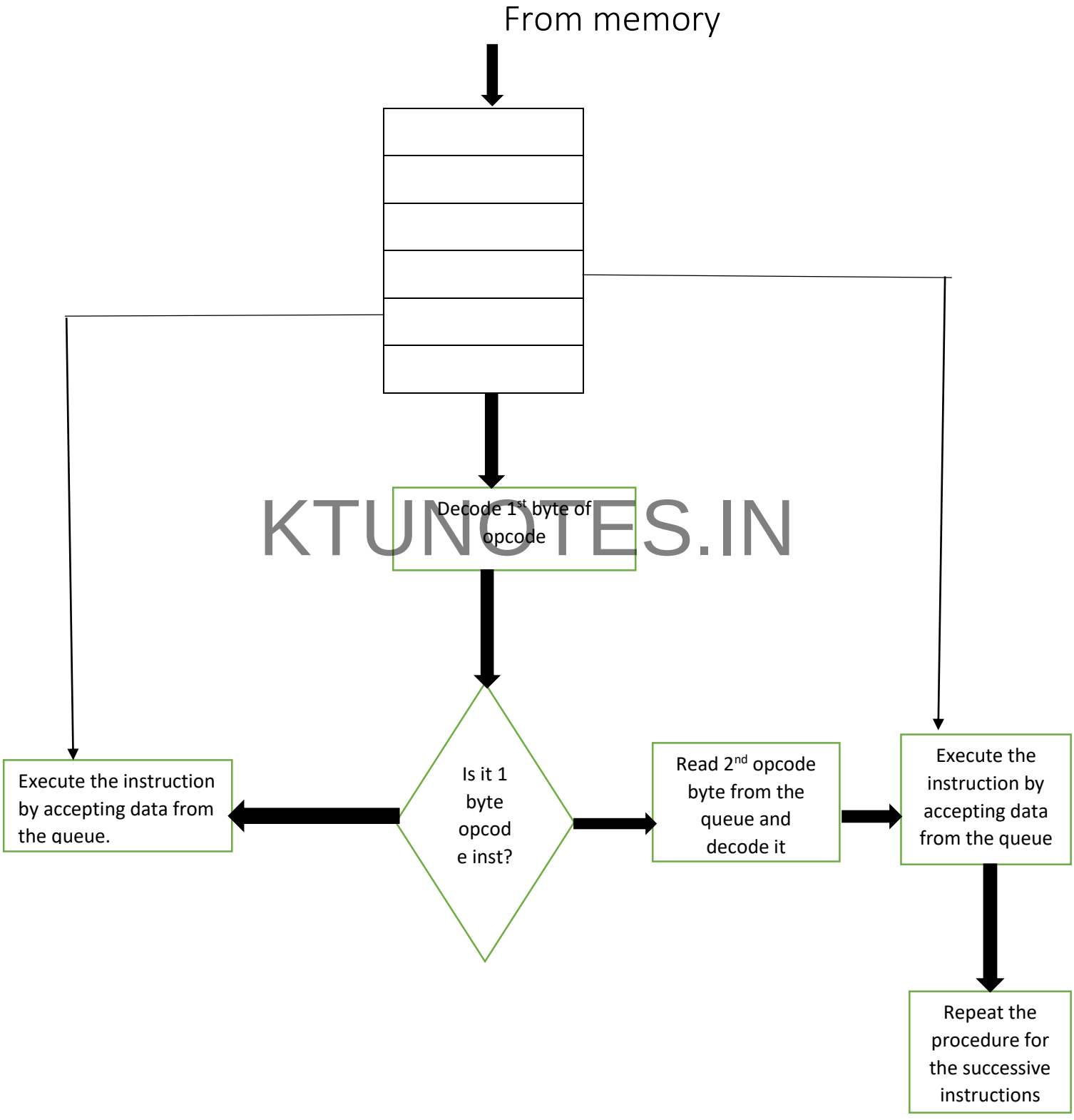
- Used by other local bus masters,to force the processor to release the local bus at the end of the processors current bus cycle.
- Bidirectional.
- RQ/GT_0 has higher priority than RQ/GT_1.

THE REQUEST/GRANT SEQUENCE IS AS FOLLOWS:

1. A pulse one clock wide from another bus master requests the bus access to 8086.
2. During T_4(current) or T_1(next) clock cycle, a pulse one clock wide from 8086 to the requesting master, indicates that the 8086 has allowed the local bus to float and that it will enter the "hold acknowledge" state in the next clock cycle. The cpu ' s bus interface unit is likely to be disconnected from the local bus of the system.
3. A one clock wide pulse from the another master indicates to 8086 that the 'hold

’request is about to end and 8086 may regain control of the local bus at the next clock cycle.

THE QUEUE OPERATION IN 8086



STEPS

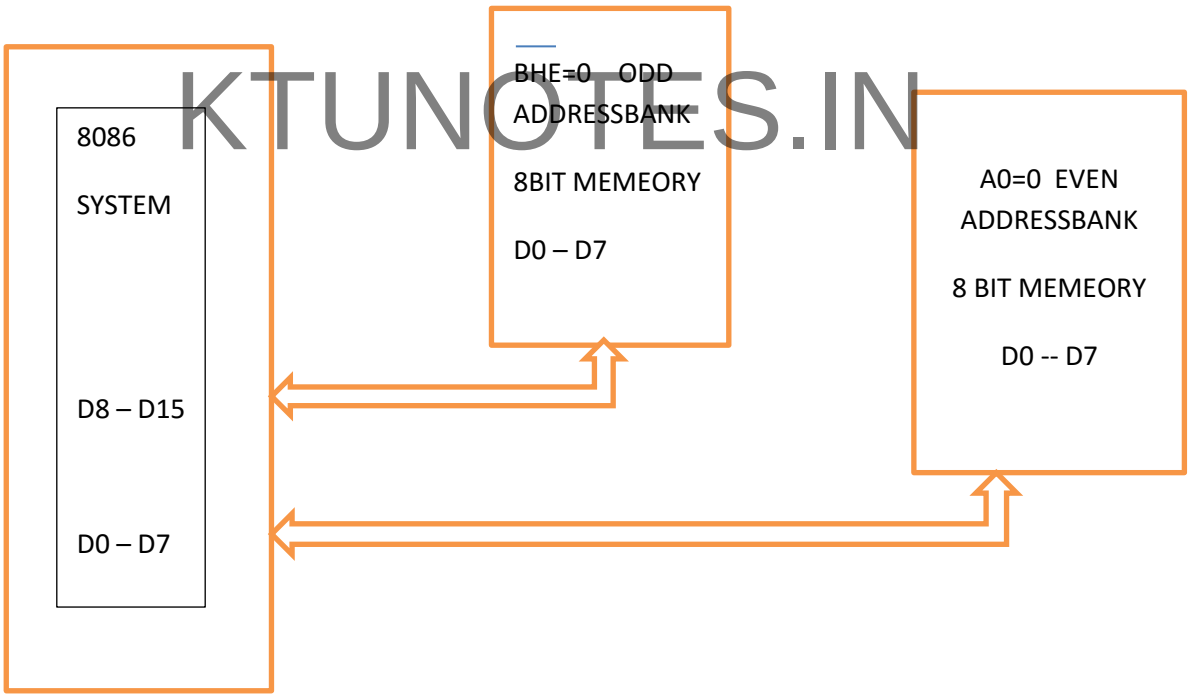
- The cs: ip is loaded with the required address from which the execution is to be started.
- Initially ,the queue will be empty and the microprocessor starts a fetch operation to bring one byte of (first byte)instruction code ,if the cs:ip address is even.
- The first byte is a complete opcode in case of some instructions(one byte opcode)and it is a part of opcode in case of two byte long opcode instruction,and remaining part of opcode lie in 2nd byte.
- Opcodes along with data are fetched in the queue .when the first byte from the queue goes for decoding, and interpretation, one byte in the queue becomes empty and subsequently the queue is updated.
- The microprocessor does not perform the next fetch operation till atleast two bytes of the instruction queue are emptied.
- After decoding the first byte ,the decoding circuit decides whether the instruction is of single opcode byte or double opcode byte.
- If it is single opcode byte, the next bytes in the queue are treated as data bytes. otherwise the next

byte in the queue is treated as the second byte of the instruction opcode.

- The second byte is then ,decoded in continuation with the first byte to decide the instruction length and no of subsequent bytes to be treated as instruction data.
- The queue is updated after every byte is read from the queue but the fetch cycle is initiated by BIU only if atleast two bytes of the queue are empty and EU may concurrently executing the fetched instructions.

PHYSICAL MEMEORY ORGANIZATION

- In 8086 based system,1MB memory is physically organized as an odd bank and an even bank, each of 512kbytes,addressed in parallel by processor.
- Byte data with an even bank address is transferred on D7-D0, while the byte data with odd address is transferred on D15-D8 bus lines.
- The processor provides two enable signals, $\overline{BHE}$ and A0 for selection of either even or odd or both the banks.
- Commercially available memory chips are only 1 Byte size that is they can store only one byte in one memory location.
- To store 16bit data, two successive memory locations are used and the lower byte of 16 bit data can be stored in first memory location while the second byte is stored in next location.
- In a 16 read or write operation both of these bytes will be read or written in a single machine cycle.



- Bits D0-D7 of a 16 bit data will be transferred over D0 - D7 (lower byte)of 16 bit data bus to/from 8bit memory.
- Bits D8-D15 of the 16 bit data will be transferred over D8 - D15 (higher byte)of the 16 bit data bus to/from 8 bit memory.
- The lower byte of a 16 bit data is stored at the first address of the map 00000H and is to be transferred over D0-D7 of the microprocessor bus.so 00000H must be in 8 bit memory.
- Higher byte of the 16 bit data is stored in the next address 00001H is to be transferred over D8-D15 of the microprocessor bus.so the address 00001H must be in 8 bit memory.
- The complete memory map of 8086 is filled with 16 bit data, all the lower bytes(D0-D7)will be stored in the 8 bit odd memory bank and all the higher bytes(D8-D15) will be stored in 8 bit even memory bank .
- Thus the complete memory map of 8086 system is divided into even and odd address memory banks.
- If 8086 transfer a 16 bit data to/from memory, both of these banks must be selected for 16 bit operation.
- To maintain the upward compactibility with 8085, 8088 must be able to implement 8 bit operations.
    - The first 8 bit operation with an even memory bank and second one is 8 bit operation with odd memory bank.
    - The two signals A0 and $\overline{BHE}$ solve the problem of selection of appropriate memory banks.

- Certain memory locations are preserved for specific cpu operations. The locations from FFFF0H to FFFFFH are reserved for operations including jump to initialization program and I/O processor initialization.
- The location 00000H to 003FFH are reserved for interrupt vector table.
- The interrupt structure provides space for a total of 256 interrupt vectors(1KB locations).