MIC Question Bank UT – 1

***Created by Shaikh Hassan,Oaish Qazi And Azlan Ahmed***

**Chapter 1**

|  |  |
| --- | --- |
| 1. | State the function of –BHE and Ao pins of 8086. (S-19) (2M) |
| Ans. | 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.  A0: A0 is analogous to BHE for the lower byte of the data bus, pinsD0-D7. A0 bit  is Low during T1 state when a byte is to be transferred on the lower portion of  the bus in memory or I/O operations. |
| 2. | Explain the concept of pipelining in 8086. State the advantages of pipelining (any two). (S-19) (W-19) (4M) |
| Ans. | **Pipelining:**  1. The process of fetching the next instruction when the present instruction  is being executed is called as pipelining.  2. Pipelining has become possible due to the use of queue.  3. BIU (Bus Interfacing Unit) fills in the queue until the entire queue is  full.  4. BIU restarts filling in the queue when at least two locations of queue are  vacant.  **Advantages of pipelining:**   * The execution unit always reads the next instruction byte from the   queue in BIU. This is faster than sending out an address to the memory  and waiting for the next instruction byte to come.   * More efficient use of processor. * Quicker time of execution of large number of instruction. * In short pipelining eliminates the waiting time of EU and speeds up the   processing. -The 8086 BIU will not initiate a fetch unless and until there  are two empty bytes in its queue. 8086 BIU normally obtains two instructions per fetch |
| 3. | Explain memory segmentation in 8086 and list its advantages. (any two)(S-19)(S-22)4m |
| Ans.  4.  Ans. | Memory Segmentation:   1. In 8086 available memory space is 1MByte. 2. This memory is divided into different logical segments and each   segment has its own base address and size of 64 KB.   1. It can be addressed by one of the segment registers. 2. There are four segments.       **Advantages of Segmentation:**   * 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.   * It allows memory addressing capability to be 1 MB. * gives separate space for Data, Code, Stack and Additional Data   segment as Extra segment size.   * The addresses of the segment may be assigned as 0000H to F000H   respectively.   * The offset values are from 00000H to FFFFFH * Segmentation 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 fast.  Describe the mechanism for generation of physical address in 8086 with suitable example. (S-19) (4M)    As all registers in 8086 are of 16 bit and the physical address will be in 20 bits.  For this reason the above mechanism is helpful.  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  OR  • i.e. Calculate physical Address for the given  CS= 3525H, IP= 2450H. |
| 5. | Draw architectural block diagram of 8086 and describe its register organization. (S-19) (6M) (W-19) (4M) |
| Ans. | Register Organization of 8086  1. AX (Accumulator) – Used to store the result for arithmetic / logical  Operations  2. BX – Base – used to hold the offset address or data  3. CX – acts as a counter for repeating or looping instructions.  4. DX – holds the high 16 bits of the product in multiply (also handles  divide operations)  5. CS – Code Segment – holds base address for all executable instructions  in a program   1. SS - Base address of the stack 2. DS – Data Segment – default base address for variables   8. ES – Extra Segment – additional base address for memory variables in  extra segment.  9. BP – Base Pointer – contains an assumed offset from the SS register.  10. SP – Stack Pointer – Contains the offset of the top of the stack.  11. SI – Source Index – Used in string movement instructions. The source  string is pointed to by the SI register.  12. DI – Destination Index – acts as the destination for string movement  Instructions  13. IP – Instruction Pointer – contains the offset of the next instruction to be  executed.  14. Flag Register – individual bit positions within register show status of  CPU or results of arithmetic operations. |
| 6. | State the function of READY & INTR pin of 8086. (W-19) (2M) |
| Ans. | Ready:  It is used as acknowledgement from slower I/O device or memory.  It is Active high signal, when high; it indicates that the peripheral device is  ready to transfer data.  INTR  This is a level triggered interrupt request input, checked during last clock  cycle of each instruction to determine the availability of request. If any  interrupt request is occurred, the processor enters the interrupt acknowledge  cycle. |
| 7. | Draw flag register of 8086 and explain any four flags. (W-19) (4M) |
| Ans. | **Conditional /Status Flags**  C-Carry Flag : It is set when carry/borrow is generated out of MSB of  result. (i.e D7 bit for 8-bit operation, D15 bit for a 16 bit operation).  P-Parity Flag This flag is set to 1 if the lower byte of the result contains even  number of 1’s otherwise it is reset.  AC-Auxiliary Carry Flag This is set if a carry is generated out of the lower  nibble, (i.e. From D3 to D4 bit)to the higher nibble  Z-Zero Flag This flag is set if the result is zero after performing ALU  operations. Otherwise it is reset.  S-Sign Flag This flag is set if the MSB of the result is equal to 1 after  performing ALU operation , otherwise it is reset.  O-Overflow Flag This flag is set if an overflow occurs, i.e. if the result of  a signed operation is large enough to be accommodated in destination  register.  **Control Flags**  T-Trap Flag If this flag is set ,the processor enters the single step execution  mode.  I-Interrupt Flag it is used to mask(disable) or unmask(enable)the INTR  interrupt. When this flag is set,8086 recognizes interrupt INTR. When it  is reset INTR is masked.  D-Direction Flag It selects either increment or decrement mode for DI &/or  SI register during string instructions. |
| 8. | Define logical and effective address. Describe physical address generation process in 8086. If DS = 345A H and SI = 13DC H. Calculate physical address. (W-19) (6M) |
| Ans. | **A logical address** is the address at which an item (memory cell, storage  element) appears to reside from the perspective of an executing application  program. A logical address may be different from the physical address due  to the operation of an address translator or mapping function.  **Effective Address or Offset Address:** The offset for a memory operand is  called the operand's effective address or EA. It is an unassigned 16 bit  number that expresses the operand's distance in bytes from the beginning of  the segment in which it resides. In 8086 we have base registers and index  registers.  Generation of 20 bit physical address in 8086:-  1. Segment registers carry 16 bit data, which is also known as base address.  2. BIU appends four 0 bits to LSB of the base address. This address becomes  20-bit address.  3. Any base/pointer or index register carries 16 bit offset.  4. Offset address is added into 20-bit base address which finally forms 20 bit  physical address of memory location    DS=345AH and SI=13DCH  Physical adress = DS\*10H + SI  = 345AH \* 10H + 13DCH  = 345A0+13DC  = 3597CH |
| 9. | Draw the labelled format of 8086 flag register. (S-22) (2M) |
| Ans. |  |
| 10. | Write the function of following pins of 8086 : (i) BHE (ii) ALE (iii) READY (iv) RESET (S-22) (4M) |
| Ans. | **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.  **ALE** : Address Latch Enable : This output signal indicates the availability of the valid address on the address/data lines, and is connected to latch enable input of latches.  **Ready:** It is used as acknowledgement from slower I/O device or memory.  It is Active high signal, when high; it indicates that the peripheral device is  ready to transfer data.  **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. |
| 11. | Describe how 20 bit Physical address is generated in 8086 microprocessor with suitable example. (S-22) (4M) |
| Ans. | The BIT) always inserts zeros for the lowest 4 bits (nibble) in the contents of segment register to generate 20-bit base address. For example, if the code segment register contains 348AH, then code segment will start at address 348A0H. |
| 12. | Calculate the physical address if : (S-22) (6M)  (i) CS = 1200H and IP = DE00H  (ii) SS = FF00H and SP = 0123H  (iii) DS = 1F00H and BX = 1A00H for MOV AX, [BX] |
| Ans. |  |
| 13. | State any four features of 8086. (Sample paper) |
| Ans. | 1. It was the first 16-bit miroprocessor having 16-bit multiplexed address and data bus AD0-AD15 to minimize the numbers of pin on IC 2. It provide 20-bit address line, 220 = 1MB of memory can be addressed 3. 8086 has a 16bit data bus. It can read or write data to a memory/port either 16bits or 8 bit at a time. 4. Operating clock Frequencies are 5 MHz ,8 MHz ,10 MHz 5. A 40 pin dual in line package. 6. 8086 is designed to operate in two modes, Minimum mode (single microprocessor configuration) and Maximum mode (multi micro processors configuration). 7. It can support up to 64K I/O ports. 8. It can prefetch upto 6 instruction bytes from memory and queues them in order to speed up instruction execution. 9. It requires +5V power supply. 10. It provides 14, 16 -bit registers. 11. It support multi programming 12. 24 addressing mode 13. Seperate instructions for string manuplation   **Chapter 2** |
| 1) State the role of Debugger in assembly language programming (S-19) (2M) | |
| 2) Demonstrate in detail the program development steps in assembly language programming. (S-19) (6M) | |
| 3) Explain any four assembler directives of 8086 with example. (S-19) (S-22) (4M) | |
| 4) List assembly language programming tools. (W-19) (2M) | |
| 5) Explain assembly language program development steps. (W-19) (4M) | |
| 6) Explain the use of assembler directives : (i) DW (ii) EQU (iii) ASSUME (iv) OFFSET (v) SEGMENT (vi) EVEN (W-19) (6M) | |
| 7) State the function of editor and assembler (S-22) (2M) | |
| 8) Describe how an assembly language program is developed and debugging using program developments tools. (S-22) (6M) | |

|  |  |
| --- | --- |
| Chp 3 | |
| 1. | Describe any six addressing modes of 8086 with suitable diagram. (S-19) (W-19) (6M) |
| A | 1. Immediate addressing mode:  An instruction in which 8-bit or 16-bit operand (data) is specified in the instruction, then the addressing mode of such instruction is known as Immediate addressing mode.  Example:  MOV AX,67D3H  2. Register addressing mode  An instruction in which an operand (data) is specified in general purpose registers, then the addressing mode is known as register addressing mode.  Example:  MOV AX,CX  3. Direct addressing mode  An instruction in which 16 bit effective address of an operand is specified in the instruction, then the addressing mode of such instruction is known as direct addressing mode.  Example:  MOV CL,[2000H]  4. Register Indirect addressing mode  An instruction in which address of an operand is specified in pointer register or in index register or in BX, then the addressing mode is known as register indirect addressing mode.  Example:  MOV AX, [BX]  5. Indexed addressing mode  An instruction in which the offset address of an operand is stored in index registers (SI or DI) then the addressing mode of such instruction is known as indexed addressing mode.  DS is the default segment for SI and DI. For string instructions DS and ES are the default segments for SI and DI resp. this is a special case of register indirect addressing mode.  Example:  MOV AX,[SI]  6. Based Indexed addressing mode:  An instruction in which the address of an operand is obtained by adding the content of base register (BX or BP) to the content of an index register (SI or DI) The default segment register may be DS or ES  Example:  MOV AX, [BX][SI]  7. Register relative addressing mode: An instruction in which the address of the operand is obtained by adding the displacement (8-bit or 16 bit) with the contents of base registers or index registers (BX, BP, SI, DI). The default segment register is DS or ES.  Example:  MOV AX, 50H[BX]  8. Relative Based Indexed addressing mode  An instruction in which the address of the operand is obtained by adding the displacement (8 bit or 16 bit) with the base registers (BX or BP) and index registers (SI or DI) to the default segment.  Example:  MOV AX, 50H [BX][SI] |
| 2. | Draw Machine language instruction format for Register-to-Register transfer. (W-19) (2M) |
| A |  |
| 3. | Define immediate addressing mode with suitable example. (S-22) (2M) |
| A | 1. Immediate addressing mode:  An instruction in which 8-bit or 16-bit operand (data) is specified in the instruction, then the addressing mode of such instruction is known as Immediate addressing mode.  Example:  MOV AX,67D3H |
| 4. | State the addressing mode of following instructions : (S-22) (6M) |
|  | 1. MOV AX, 3456H - Immediate Addressing Mode 2. ADD BX, [2000H] - Direct Addressing Mode 3. DAA - Implied Addressing Mode 4. MOV AX, [Si] - Indexed Addressing Mode 5. MOV AX, BX - Register Addressing Mode 6. SUB AX, [BX + SI + 80H] - Relative Based Indexed Addressing Mode |
| Chp 4 | |
| 1. | Write ALP for addition of two 8 bit numbers. Assume suitable data. (S-19) (S-22) (2M) |
| A | .MODEL SMALL  .STACK 100H  .DATA  NUM1 DB 23H  NUM2 DB 17H  RESULT DB ?  .CODE  MAIN PROC  MOV AX, @DATA  MOV DS, AX  MOV AL, NUM1  ADD AL, NUM2  MOV RESULT, AL  MOV AH, 4CH  INT 21H  MAIN ENDP  END MAIN |
| 2. | Write an ALP to add two 16-bit numbers. (W-19) (4M) |
| A | DATA SEGMENT  NUMBER1 DW 6753H  NUMBER2 DW 5856H  SUM DW 0  DATA ENDS  CODE SEGMENT  ASSUME CS: CODE, DS: DATA    START: MOV AX, DATA  MOV DS, AX  MOV AX, NUMBER1  MOV BX, NUMBER2  ADD AX, BX  MOV SUM, AX  MOV AH, 4CH  INT 21H  CODE ENDS  END START |
| 3. | Write an ALP to multiply two 16 bit signed numbers. (S-22) (4M) |
| A | .MODEL SMALL  .STACK 100H  .DATA  NUM1 DW -0004h  NUM2 DW 0008h  RESULT DW 0000h  .CODE  MAIN PROC  MOV AX, @DATA  MOV DS, AX  MOV AX, NUM1  IMUL NUM2  MOV RESULT, AX  MOV AH, 4CH  INT 21H  MAIN ENDP  END MAIN |
| 4. | Write ALP for subtraction of two 8 bit numbers and 16 bit numbers |
| A | .MODEL SMALL  .STACK 100h  .DATA  num1 DB 10h  num2 DB 5h  result DB ?    n1 DW 1234h  n2 DW 5678h  rez DW ?  .CODE  MOV AX, @DATA  MOV DS, AX    ;8-Bit Subtraction  MOV AL, num1  SUB AL, num2  MOV result, AL  ;16-Bit Subtraction  MOV AX, n1  SUB AX, n2  MOV rez, AX  MOV AH, 4Ch  INT 21h  END |