

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**UNIT I-THE 8086 MICROPROCESSOR**

**PART – A**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Write the flags of 8086.** | **AU-AM 2021** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| The flags of 8086 microprocessor   * Sign Flag (S) * Zero Flag (Z) * Auxiliary Cary Flag (AC) * Parity Flag (P) * Carry Flag (CY) * Overflow Flag (O) * Directional Flag (D) * Interrupt Flag (I) * Trap Flag (T) | | | | | | |
| **2** | **List the types of interrupts in 8086** | **AU-AM 2021** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| The types of interrupts in 8086:   * Divide by zero interrupt (Type 0) * Single step interrupt (Type 1) * Non Maskable interrupt (Type 2) * Breakpoint interrupt (Type 3) * Overflow interrupt ( Type 4) * Software interrupt | | | | | | |
| **3** | **State the advantages of subroutine.** | | **AU-ND 2019** | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.4.1** |
| * [Decomposing](https://en.wikipedia.org/wiki/Decomposition_(computer_science)) a complex programming task into simpler steps. * Reducing [duplicate code](https://en.wikipedia.org/wiki/Duplicate_code) within a program. * Enabling [reuse of code](https://en.wikipedia.org/wiki/Code_reuse) across multiple programs. * Dividing a large programming task among various programmers, or various stages of a project. * [Hiding implementation details](https://en.wikipedia.org/wiki/Information_hiding) from users of the subroutine. * Improving readability of code by replacing a block of code with a function call where a [descriptive](https://books.google.com/books?id=_i6bDeoCQzsC&pg=PA39&dq=descriptive+function+name) function name serves to describe the block of code. This makes the calling code concise and readable even if the function is not meant to be reused. * Improving [traceability](https://en.wikipedia.org/wiki/Traceability#Software). | | | | | | |
| **4** | **For microprocessor, the contents of the registers are, CS=2001H, SS=6046, IP=2456, SP=2200H. Calculate the corresponding physical address for the addressed byte in a)CS b) SS.** | | **AU-ND 2019** | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **2.1.3** |
| a)Code segment =20010 H  IP + = 2456 H  **Physical address =22466 H**  b)Stack segment =60460 H  SP = 2200 H  **Physical address =62660 H** | | | | | | |
| **5** | **Give the examples for the following modes of addressing.**  **(i)Relative based indexed mode**  **(ii)Direct addressing** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **2.1.3** |
| **(i) Relative based indexed mode**  Example:  In this mode, the EA is the sum of base register, index register and displacement  MOV AL, [BX+SI+10H]  **(ii) Direct addressing**  In this type of addressing mode the effective address is directly given in the instruction as displacement. Example:  MOV AX, [DISP]  MOV AX, [0500] | | | | | | |
| **6** | **What is the need for interrupts in microprocessor operation?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.4.1** |
| * An interrupt is a condition that halts the microprocessor temporarily to work on a different task and then return to its previous task. * Interrupt is an event or signal that request to attention of CPU. This halt allows peripheral devices to access the microprocessor. * Whenever an interrupt occurs the processor completes the execution of the current instruction and starts the execution of an Interrupt Service Routine (ISR) or Interrupt Handler. | | | | | | |
| **7** | **What are Byte and String Manipulation?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| * String is a series of data byte or word available in memory at consecutive locations. * It is either referred as byte string or word string. * Their memory is always allocated in a sequential order. Instructions used to manipulate strings are called string manipulation instructions. | | | | | | |
| **8** | **Define stack pointer.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| * A stack pointer is a specialized [buffer](http://searchcio-midmarket.techtarget.com/definition/buffer) which stores data from the top to down. As new requests come in, they "push down" the older ones. * The most recently entered request always resides at the top of the stack, and the program always takes requests from the top. | | | | | | |
| **9** | **List the various addressing modes of 8086.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 2.2.4** |
| The various addressing modes of 8086 are:   * Register Addressing mode : * Immediate Addressing mode: * Register Indirect Addressing mode: * Direct Addressing mode: * Indexed Addressing mode: * Base Relative Addressing mode: * Base Indexed addressing mode: | | | | | | |
| **10** | **Define Macros.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| * Macros are small routines that are used to replace strings in the program. They can have parameters passed to them, which enhances the functionality of the micro itself. * A Macro is a group of instructions with a name. When a macro is invoked, the associated set of instructions is inserted in place in to the source, replacing the macro name. Macros provide several powerful mechanisms useful for the development of generic programs. | | | | | | |
| **11** | **The offset address of a data is (341B)H and the data segment register value is (123A)H. What is the physical address of the data?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 2.1.3** |
| * Dada segment Address is (123A)H * Base address of data segment is (123A0)H * Physical address = Base address + Offset address   = (123A0) H+ (341B) H  = (157BB) H | | | | | | |
| **12** | **List the segment registers of 8086.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.4.1** |
| The segment registers of 8086 are:   * Code Segment register * Data Segment register * Stack Segment register * Extra Segment register | | | | | | |
| **13** | **Write about the different types of interrupts supported in 8086?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.4.1** |
| * The following are the different types of interrupts supported in 8086:  1. Hardware Interrupt  * Maskable Interrupts * Non-Maskable Interrupts  1. Software Interrupts  * 256 Types of Software Interrupts | | | | | | |
| **14** | **Identify the addressing modes in the following instruction.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 2.1.2** |
| AND AL, BL - Register Addressing mode  SUB AL, 24H - Immediate Addressing mode  MOV AL, (BP) - Register Indirect Addressing mode  MOV CX, 1245H**. -** Direct Addressing mode | | | | | | |
| **15** | **What is a tristate bus?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.2.1** |
| * A three-state bus is a computer bus connected to multiple tri-state output devices, only one of which can be enabled at any point to avoid bus contention. This scheme allows for the same bus to be shared among multiple devices. | | | | | | |
| **16** | **What is an assembler?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| * An assembler is a program that takes basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations. * Assemblers need to:  1. Translate assembly instructions and pseudo-instructions into machine instructions. 2. Convert decimal numbers, etc. specified by programmer into binary.  * Typically, assemblers make two passes over the assembly file:  1. First pass: Reads each line and records labels in a symbol table. 2. Second pass: Use info in symbol table to produce actual machine code for each line. | | | | | | |
| **17** | **Why is the 8086 memory divided into odd and even banks?** | | | **REMEMBER**  **BT-L1** | **CO2** | **PI 2.2.2** |
| * The address space is physically connected to a 16-bit data bus by dividing the address space into two 8-bit banks of up to 512 k bytes each. * One bank is connected to the lower half of the 16-bit data bus (D0 – D7) and contains even address bytes, i.e., when A0 bit is low, the bank is selected. * The other bank is connected to the upper half of the data bus (D8 - D15) and contains odd address bytes, i.e., when A0 is high and BHE (Bus High Enable) is low, the odd bank is selected. A specific byte within each bank is selected by address lines A1-A19. | | | | | | |
| **18** | **What do you mean by segment override prefix?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.4.1** |
| * A segment override prefix allows any segment register (DS, ES, SS, or CS) to be used as the segment when evaluating addresses in an instruction. * An override is made by adding the segment register plus a colon to the beginning of the memory reference of the instruction as in the following examples:   mov ax, [es: 60126] ; Use es as the segment  mov ax, [cs: BX] ; Use cs as the segment  mov ax, [ss: BP+si+3] ; Use ss as the segment | | | | | | |
| **19** | **What are the 8086 instructions used for BCD arithmetic?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| * DAA(DECIMAL ADJUST AFTER BCD ADDITION)-This instruction is used to make sure the result of adding two packed BCD numbers is adjusted to a legal BCD number. * DAS (DECIMAL ADJUST AFTER BCD SUBTRACTION)-This instruction is used after subtracting one packed BCD number from another packed BCD number, to make sure the result is correct packed BCD. | | | | | | |
| **20** | **What are the contents of AL and CY after the execution of the following segment?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 2.1.2** |
| Mov BL, B5H  RCL BL, 3  Mov AL, BL  **Output:**  C=0 , BL=10110101  After 1strotation,C=1 ,BL=01101010  After 2ndrotation,C=0 ,BL=11010101  After 3rdrotation,C=1 ,BL=10101011 | | | | | | |
| **21** | **What are the 8086 instructions used for ASCII arithmetic?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| The 8086 instructions used for ASCII arithmetic are:   * AAA : ASCII adjust after addition * AAS : ASCII adjust after subtraction * AAM : ASCII adjust after multiplication * AAD: ASCII adjusts before division. | | | | | | |
| **22** | **List the various string instructions available in 8086.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 2.2.4** |
| The various string instructions of 8086 are:   1. MOVS : Move byte or word string 2. MOVSB, MOVSW : Move byte, word string 3. CMPS : Compare byte or word string 4. SCAS : Scan byte or word string (comparing to A or AX) 5. LODS, STOS : Load, store byte or word string to AL or AX | | | | | | |
| **23** | **How clock signal is generated in 8086 microprocessor?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.2.1** |
| * The clock signal of 8086 is generated using 8284 clock generator. * The crystal is connected to the crystal oscillator input of 8284 to generate the clock signal. * The maximum internal frequency of 8086 is 5 MHz. | | | | | | |
| **24** | **How will carry and zero flags reflect the result of the instruction CMP BX, CX?** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| * The carry and zero flags reflect the following result for CMP BX,CX instruction:   If BX>CX, carry flag=1  If BX=CX, zero flag=1  If BX<CX, carry flag=0 | | | | | | |
| **25** | **What is an assembler directive? Give two examples.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| * Assembler directives or pseudo instructions are to help the programmer to communicate the program requirements to the assembler. * Examples: ASSUME, DB, DD, DQ, DT, END, ENDS, ENDP, EQU etc. | | | | | | |

**PART B**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | How would you show your understanding of internal hardware architecture of 8086 microprocessor with neat diagram? **(13)** | **AU-AM 2021** | **UNDERSTAND**  **BT-L2** | **CO1** | **PI 1.3.1** |
| **2** | What are the various addressing modes of 8086 microprocessor with examples? **(13)** | **AU-AM 2021** | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| **3** | For 8086 Microprocessor what are the instruction set and assembler directives? **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| **4** | Explain the various addressing modes of 8086 microprocessor with suitable examples. **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 2.2.4** |
| **5** | Draw and explain the architecture of 8086 with neat diagram. **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| **6** | Describe the interrupts of 8086 and its types with service routine. **(13)** | | **UNDERSTAND**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **7** | (i)Explain briefly about the internal hardware architecture of 8086 microprocessor with a neat diagram. **(08)**  (ii)Write short note about assembler directives**. (05)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| **8** | Classify the different addressing modes of 8086 microprocessor with suitable examples. **(13)** | | **ANALYZE**  **BT-L4** | **CO1** | **PI 2.2.4** |
| **9** | Define interrupts and their types. Write in detail, about interrupt service routine. **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| **10** | (i)Explain briefly about the internal hardware architecture of 8086 microprocessor with a neat diagram.  **(08)**  (ii)Write a 8086 ALP to convert BCD data to Binary data.  **(05)** | | **REMEMBER**  **BT-L1**  **APPLY**  **BT-L3** | **CO1** | **PI 1.3.1** |
| **11** | (i)The data transfer rate of I/O Device ‘A’ is considerably less than that of the microprocessor. Draw a flowchart of data transfer operation to be used. **(06)** (ii)Describe the functions of execution unit and bus interface unit. **(07)** | | **APPLY**  **BT-L3**  **REMEMBER**  **BT-L1** | **CO2** | **PI 1.3.1** |
| **12** | (i) Develop a program to transfer 10 bytes of data from memory location starting from 2000H. **(07)**  (ii)Describe program location control directives with suitable examples. **(06)** | | **APPLY**  **BT-L3**  **REMEMBER**  **BT-L1** | **CO1** | **PI 2.1.3** |
| **13** | (i) Develop a program to multiply two 16 bit numbers stored in P1 and P2. **(07)**  (ii)Explain rotate and shift instructions with suitable examples. **(08)** | | **APPLY**  **BT-L3 REMEMBER**  **BT-L1** | **CO1** | **PI 2.1.3** |
| **14** | With neat diagrams explain the bus interfacing unit and execution unit available in 8086 microprocessor. **(13)** | | **REMEMBER**  **BT-L1** | **CO2** | **PI 1.4.1** |
| **15** | Briefly explain the addressing modes in 8086 microprocessor with example. **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.4.1** |
| **16** | (i) Briefly explain the arithmetic group of instructions of 8086 processor. **(07)**  (ii)Briefly explain the assembler directives available in 8086. **(06)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| **17** | (i) Explain the 8086 interrupt types in detail. **(06)**  (ii)Describe how memory is accessed in 8086 with suitable diagram **(07)** | | **REMEMBER**  **BT-L1** | **CO2** | **PI 1.2.1** |
| **18** | Explain the architecture of 8086 microprocessor. **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |
| **19** | (i)Describe the architecture of 8086 microprocessor with neat diagram. **(08)**  (ii)What are the differences between memory mapped I/O and I/O mapped I/O. **(05)** | | **REMEMBER**  **BT-L1**  **UNDERSTAND**  **BT-L2** | **CO2** | **PI 1.3.1** |
| **20** | (i) What do you mean by assembler directives? Explain SEGEMENT, TYPE, OFFSET with suitable examples. **(06)**  (ii)Write an 8086 ALP to check whether the given string is palindrome or not. **(07)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI 1.3.1** |

**PART - C**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1.** | i)Write an 8086 ALP to find the sum of numbers in an array of 10 elements. **(08)** **(8)**  ii)Write an 8086 ALP to find the largest number and smallest number in an array.  **(07)** | **AU-AM 2021** | **APPLY**  **BT-L3** | **CO1** | **PI 2.1.3** |
| **2.** | (i) Discuss the data movement and program control instructions of 8086. **(10)**  (ii)Write an 8086 ALP to find the sum of numbers in the array of 10 elements. **(05)** | | **REMEMBER**  **BT-L1**  **APPLY**  **BT-L3** | **CO1** | **PI 2.1.3** |
| **3** | Write the 8086 assembly language program to arrange the set of numbers in ascending order and explain. **(15)** | | **APPLY**  **BT-L3** | **CO1** | **PI 2.1.3** |
| **4** | (i) Explain the data transfer, arithmetic and branch instructions with examples. **(08)**  (ii) Write an 8086 ALP to find the sum of numbers in an array of 10 elements. **(07)** | | **REMEMBER**  **BT-L1**  **APPLY**  **BT-L3** | **CO1** | **PI 2.1.3** |
| **5.** | (i)Write an 8086 ALP to separate odd and even numbers in a given array. **(07)**  (ii)Explain the data transfer group and logical group of 8086 instruction with necessary examples. **(08)** | | **APPLY**  **BT-L3**  **REMEMBER**  **BT-L1** | **CO1** | **PI 2.1.3** |

**UNIT II- 8086 SYSTEM BUS STRUCTURE**

**PART-A**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **How clock signal is generated in 8086? What is the maximum internal clock frequency of 8086?** | **AU-AM 2021** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| * The 8086 does not have on-chip clock generation circuit. Hence the clock generator chip, 8284 is connected to the CLK pin of 8086. The clock signal supplied by 8284 is divided by three for internal use. * The maximum internal clock frequency of 8086 is 5 MHz. | | | | | | |
| **2** | **What is the function of MN/MX pin ?** | **AU-AM 2021** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| MN/MX is an input pin used to select one of this mode.   * When MN/MX is high the 8086 operates in minimum mode. In this mode the 8086 is configured to support single processor system. * When MN/MX is low 8086 is configured to support multiprocessor system. | | | | | | |
| **3** | **State the function of ALE signal.** | **AU-ND 2019** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| ALE is the 8086 signal which is used demultiplex the AD0-AD15 into A0-A15 and D0 to D15 using external latches | | | | | | |
| **4** | **Draw the simplified diagram of co-processor based multiprocessor system.** | **AU-ND 2019** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| The simplified diagram of co-processor based multiprocessor system is:  Multiprocessor Configuration Overview - Tutorialspoint | | | | | | |
| **5** | **Define System Bus.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| * Bus is a group of wires or lines that are used to transfer the addresses of Memory or I/O devices. It us unidirectional. * In Intel 8086 microprocessor, Address bus is of 16 bits. This means that Microprocessor 8086 can transfer maximum 16 bit address which means it can address 65,536 different memory locations. | | | | | | |
| **6** | **What are pointers and index registers?** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| * IP, BP and SP are the pointers and contain offsets within the code, data and stack segments respectively.   SI and DI are the index registers, which are used as general purpose registers and also for offset storage in case of indexed, based indexed and relative based indexed addressing modes. | | | | | | |
| **7** | **List two differences between maximum mode and minimum mode configuration of 8086.** | | **UNDERSTAND**  **BT-L2** | | **CO1** | **PI**  **2.2.4** |
| |  |  |  | | --- | --- | --- | | **Sl.No** | **Minimum mode** | **Maximum mode** | | 1. | 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. | | 2. | 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. | | | | | | | |
| **8** | **What is meant by multiprogramming?** | | **UNDERSTAND**  **BT-L2** | | **CO1** | **PI**  **1.3.1** |
| * Multiprogramming is the technique of running several process at a time using timesharing. * It allows a computer to do several things at the same time. | | | | | | |
| **9** | **What is the need of LOCK signal?** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| * It is an active low pin. * It indicates that other system bus masters **have** not been allowed to gain control of the system bus while **LOCK**' is active low(0). * The **LOCK signal** will be active until the completion of the next instruction. | | | | | | |
| **10** | **Write some example for advanced processors.** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| The examples for advanced processors are:   * 80386, 80486 * Pentium-I, Pentium-II * Pentium-III | | | | | | |
| **11** | **Define machine cycle.** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| * Machine cycle is defined as the time required completing one operation. It includes 3 to 6 T-states. * The machine cycle is a 4 process cycle that includes reading and interpreting the machine language, executing the code and then storing that code. | | | | | | |
| **12** | **Differentiate External verses Internal bus.** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **2.2.3** |
| * Internal Data Bus: The internal data bus only works inside a CPU that is internally. It is able to communicate with the internal cache memories of the CPU. Since they are internally placed they are relatively quick and are now affected by the rest of the computer. * External Data bus: This type of bus is used to connect and interface the computer to its connected peripheral devices. Since they are external and do not lie within the circuitry of the CPU they are relatively slower. | | | | | | |
| **13** | **Compare closely coupled and loosely coupled configurations.** | | **UNDERSTAND**  **BT-L2** | | **CO1** | **2.2.4** |
| |  |  | | --- | --- | | **Closely coupled configurations** | **Loosely coupled configurations** | | A multiprocessor system with common shared memory is known as closely coupled System. | A multiprocessor system in which each processor has its own private local memory is known as loosely coupled system. | | Here, the information can be shared among the CPUs by placing it in the common global memory. | Here, the information is transferred from one processor to other by message-passing system. | | Parallelism can be implemented less efficiently. | Parallelism can be implemented more efficiently. | | System structure is less flexible. | System structure is more flexible. | | | | | | | |
| **14** | **Define bus. Why bus request and cycle stealing are required?** | | **UNDERSTAND**  **BT-L2** | | **CO1** | **PI**  **1.3.1** |
| * Bus is a group of wires to transfer data from source to destination. * In the cycle stealing mode, the DMA controller obtains access to the system bus using BR (Bus Request) and BG (Bus Grant) signals, which are the two signals controlling the interface between the CPU and the DMA controller. | | | | | | |
| **15** | **State the different data transfer schemes.** | | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| There are two types of data transfer scheme:   * Synchronous * It is the data method which is used when the IO and Microprocessor match in Speed. To transfer a data to or from the device, the user program issues a suitable instruction addressing the device. The data transfer is completed at the end of the execution of the instruction. * Asynchronous * It is the data transfer method which is used when the speed of the IO device does not match with the microprocessor. Asynchronous data transfer method is also known as handshaking. | | | | | | |
| **16** | **What are the advantages of memory mapped I/O over I/O-mapped I/O**? | | **UNDERSTAND**  **BT-L2** | | **CO1** | **PI**  **2.2.4** |
| |  |  |  | | --- | --- | --- | | **Sl.No.** | **Memory mapped I/O** | **I/O-mapped I/O** | | 1 | I/O is treated as memory. | I/O is treated I/O. | | 2 | 16-bit addressing. | 8- bit addressing. | | 3 | More Decoder Hardware. | Less Decoder Hardware. | | 4 | Can address 216=64k locations. | Can address 28=256 locations. | | 5 | Less memory is available. | Whole memory address space is available. | | | | | | | |
| **17** | **What is the function of LOCK and RQ/ GT signals?** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| * LOCK signal indicates that other systems bus master will be prevented from gaining the system bus, while LOCK signal is low. It is activated by the LOCK Prefix instruction and remains active until the end of the next instruction. * Request / Grant Signals: * In a maximum mode configuration, the minimum mode HOLD, HLDA interface is also changed. These two are replaced by request/grant lines RQ/ GT0 and RQ/ GT1, respectively. * It is used for bus requests and bus grants. | | | | | | |
| **18** | **State the functions of queue status lines QS0 and QS1 in 8086.** | | **UNDERSTAND**  **BT-L2** | | **CO1** | **PI**  **1.3.1** |
| and**-** Queue status: The queue status bits shows the status of the internal instruction Queue. qs1 and qs0 reflect the status of the instruction queue. This status indicates the activity in the queue during the previous clock cycle.  The encoding of these signals is as follows:   |  |  |  | | --- | --- | --- | | **QS0** | **QS1** | **Function** | | 0 | 0 | No operation, queue is idle | | 0 | 1 | First byte of opcode | | 1 | 0 | Queue is empty | | 1 | 1 | Subsequent byte of opcode | | | | | | | |
| **19** | **What is meant by loosely coupled configuration?** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| * In Multiprocessor system, two 8086’s cannot be tied directly together. * In a loosely coupled configuration a number of modules of 8086 can be through a common system bus to work as a multiprocessor system. | | | | | | |
| **20** | **When the 8086 processor is in ‘Max mode’ and ‘Min mode’?** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| The 8086 provides MN/MX pin to select the mode of operation:   * MN/MX =1 for minimum mode and * MN/MX =0 for maximum mode. | | | | | | |
| **21** | **List any four 8087 data formats.** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| The following are the few data formats of 8087:   * Word integer * Short integer * Short real * Long real | | | | | | |
| **22** | **What are the features of closely coupled multiprocessor systems?** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| The features of closely coupled multiprocessor systems are:   * A multiprocessor system with common shared memory. * Parallelism can be implemented less efficiently. * System structure is less flexible. | | | | | | |
| **23** | **What is the difference between uni-programming and multiprogramming?** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| The differences between uni-programming and multiprogramming are:   |  |  | | --- | --- | | **Uni-programming** | **Multiprogramming** | | Programming unit which performs an independent task | The code for two or more processes is in a memory at the same time | | Processes are executed in a serial fashion | Processes are executed in a multiplexed fashion. | | | | | | | |
| **24** | **What is the use of instruction queue in 8086 microprocessor?** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| In 8086, a 6-byte instruction queue is presented at the Bus Interface Unit (BIU). It is used to prefetch and store at the maximum of 6 bytes of instruction code from the memory. Due to this, overlapping instruction fetch with instruction execution increases the processing speed. | | | | | | |
| **25** | **Explain the BHE and LOCK signals of 8086.** | | **REMEMBER**  **BT-L1** | | **CO1** | **PI**  **1.3.1** |
| BHE (Bus High Enable):Low on this pin during first part of the machine cycle indicates that at least one byte of the current transfer is to be made on higher byte AD15-AD8.  LOCK:This signal indicates that an instruction with a LOCK prefix is being executed and the bus is not to be used by the other processor. | | | | | | |

**PART B**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | Draw the pin diagram of 8086 processor and explain all the signals. **(13)** | **AU-AM 2021** | **UNDERSTAND**  **BT-L2** | **CO1** | **PI**  **1.3.1** |
| **2** | i)Explain in detail about closely coupled configurations. **(06)**  ii)Discuss on loosely coupled configurations in detail. **(07)** | **AU-AM 2021** | **UNDERSTAND**  **BT-L2** | **CO1** | **PI**  **1.3.1** |
| **3** | Draw the diagrams showing address demultiplexing for 8086. Explain the use of IC in the system and the relevant pins and signals.  **(13)** | **AU-ND 2019** | **UNDERSTAND**  **BT-L2** | **CO2** | **PI**  **1.3.1** |
| **4** | Draw the timing diagram for the ‘Memory Read’ machine cycle of 8086. Explain the function of the relevant signals and discuss how each signal changes in the progress of the machine cycles.  **(13)** | **AU-ND 2019** | **UNDERSTAND**  **BT-L2** | **CO1** | **PI**  **1.3.1** |
| **5** | Distinguish between closely coupled and loosely coupled multiprocessor configurations. **(13)** | | **ANALYZE**  **BT-L4** | **CO1** | **PI**  **2.2.4** |
| **6** | Explain the following: (i)Multiprocessor system. **(04)** (ii) Coprocessor **(03)** (iii) Multiprogramming **(03)** (iv) Semaphore **(03)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **7** | Explain in detail about the system bus timing of 8086/8088 **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **8** | (i)Draw and explain the minimum mode configuration of 8086 microprocessor. **(07)** (ii)Briefly explain the architectural advancements of microprocessor. **(06)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **9** | Write notes on  (i) Maximum mode in 8086 **(07)**  (ii) Interrupt processing **(06)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **10** | (i) How the interrupt vector is handled in 8086? **(06)**  (ii) Explain the timing diagram of write cycle in 8086 in minimum mode.  **(07)** | | **UNDERSTAND**  **BT-L2** | **CO1** | **PI**  **1.3.1** |
| **11** | Describe the minimum mode 8086 system and its timing diagram**. (13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **12** | (i) Draw the pin diagram of 8086 microprocessor and list the pin details. **(07)**  (ii)Explain the bus structure of 8086 microprocessor.  **(08)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **13** | Explain Min/Max mode of 8086 microprocessor. **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **14** | Explain in detail about closely coupled configuration of multiprocessor configuration with suitable diagram. **(13)** | | **UNDERSTAND**  **BT-L2** | **CO1** | **PI**  **1.3.1** |
| **15** | Explain in detail about Loosely Coupled Multiprocessor configuration. **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **16** | Write notes on the following **(13)**   * 1. Programmed I/O   (ii) Interrupt I/O | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **17** | With a neat sketch, explain the architecture of the 8089 processor? **(13)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **18** | Discuss the maximum mode configuration of 8086 with a neat diagram. Mention the functions of various signals.  **(13)** | | **UNDERSTAND**  **BT-L2** | **CO1** | **PI**  **1.3.1** |
| **19** | What do you understand from system bus structure? Explain. **(13)** | | **UNDERSTAND**  **BT-L2** | **CO1** | **PI**  **1.3.1** |
| **20** | Differentiate between the memory mapped I/O and I/O mapped I/O. **(06)**  Draw the control word and status word format of 8087 processor. **(07)** | | **ANALYZE**  **BT-L4**  **REMEMBER**  **BT-L1** | **CO1** | **PI**  **2.2.4** |

**PART – C**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | Examine the effectiveness of the minimum mode and maximum mode of operations in 8086 in detail. **(15)** | **AU-AM 2021** | **ANALYZE**  **BT-L4** | **CO1** | **PI**  **2.2.2** |
| **2** | Discuss about the multiprocessor configurations of 8086. **(15)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **3** | Explain the system bus structure of 8086. Draw the timing diagram for interrupt acknowledgement cycle.  **(15)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **4** | Explain in detail about the system bus timing of 8086.  **(15)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |
| **5** | Describe the maximum mode signals, bus cycles, and maximum mode system configuration of 8086 Microprocessor in detail. **(15)** | | **REMEMBER**  **BT-L1** | **CO1** | **PI**  **1.3.1** |

**unit-iii – I/O INTERFACING**

**PART-A**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **What are the signals available for serial communication** | **AU-AM 2021** | **REMEMBER**  **BT-L1** | **CO3** | **PI**  **1.3.1** |
| Transmitter signals:   * TxD-Transmit Data * TxRDY- Transmitter Ready * TxE- Transmitter Empty * TxC- Transmitter Clock   Receiver Signals:   * RxD-Receiver Signals * RxRDY- Receiver Ready * RxC-Receiver Clock * SYNDET (Syn Detect)/BRKDET (Break Detect) | | | | | |
| **2** | **Define conversion time.** | **AU-AM 2021** | **REMEMBER**  **BT-L1** | **CO3** | **PI**  **1.3.1** |
| * Conversion time of an ADC is the time required by the ADC to perform a complete conversion process. | | | | | |
| **3** | **What is the function of the following signals in an ADC (while interfacing a microprocessor)? i) EOC ii) SOC** | **AU-ND 2019** | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 1.3.1** |
| **EOC-**End of Conversion  After the conversion is over, the ADC sends end of conversion (EOC) signal to inform the microprocessor that the conversion is over and the result is ready at the output buffer of the ADC.  **SOC –** Start of Conversion  The start of conversion signal is a pulse of a specific duration. The process of analog to digital conversion is a slow process, and the microprocessor has to wait for the digital data till the conversion is over. | | | | | |
| **4** | **List the function of HOLD and HLDA in 8086.** | **AU-ND 2019** | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| * A HIGH on HOLD pin indicates that another master is requesting to take over the system bus. * On receiving HOLD signal processor outputs HLDA signal HIGH as an acknowledgement. * After "HOLD" is detected as being Low, the processor will lower the HLDA, it will again drive the local bus and control lines. | | | | | |
| **5** | **Give the various modes and applications of 8254 timer?** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| The various modes and applications of 8254 timer are,  Mode 0: Interrupt On Terminal Count  Mode 1: Hardware Retriggerable one shot  Mode 2: Rate Generator  Mode 3: Square Wave Generator  Mode 4: Software Triggered Strobe  Mode 5: Hardware Triggered Strobe | | | | | |
| **6** | **What is the handshake signals used in Mode-2 configuration of 8255?** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| * In mode2 only port A can be used as a bidirectional port. * The handshaking signals are provided on five lines of port C (PC3-PC7). Port B can be used in mode0 or in mode1. | | | | | |
| **7** | **Draw the format of read back command register of 8254.** | | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 1.4.1** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | | **1** | **1** | **COUNT** | **STATUS** | **CNT2** | **CNT1** | **CNT0** | **0** |  * D7-D6 = 1 for read back command register * D5 Latch count * D4 Latch status * D3 select counter 2 * D2 select counter 1 * D1 select counter 0 | | | | | |
| **8** | **How the DMA operation performed with 8086?** | | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 1.3.1** |
| * The direct memory access DMA interface of the 8086 minimum mode consist of the HOLD and HLDA signals. * When an external device wants to take control of the system bus, it signals to the 8086 by switching HOLD to the logic 1 level. | | | | | |
| **9** | **What is the drawback of memory mapped I/O?** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 2.4.3** |
| Drawbacks of memory mapped I/O are:   * Memory mapped I/O scheme utilizes memory reference instructions, which are three byte instructions. * Due to wider port address, the interface of hardware is also complicated. * The complexity of the program is large. | | | | | |
| **10** | **How DMA is initiated?** | | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 2.4.4** |
| * The device requests the CPU through a DMA controller to hold its data, address and control bus, so that the device may transfer data directly to/from memory. * The DMA data transfer is initiated only after receiving HLDA signal form the CPU. | | | | | |
| **11.** | **What is key debouncing?** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| * The push button keys when pressed, bounces a few times, closing and opening the contacts before providing a steady reading. * So reading taken during bouncing may be faulty. * Therefore the microprocessor must wait until the key reach to steady state. This is known as key debounce. | | | | | |
| **12.** | **What is direct memory access?** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| * Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations. * The process is managed by a chip known as a DMA controller (DMAC) | | | | | |
| **13.** | **What is a sample and hold circuit?** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| * The sample and Hold circuit is used to convert the analog signal into digital signal by using the RC circuit. * The capacitor holds the value of the signal for a particular time interval. * The analog signal is sampled once for the RC time constant value. | | | | | |
| **14.** | **State the applications of Programmable Interval Timer.** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| Applications of PIT 8254 are:   * Real time clock Event-counter * Digital one-shot Programmable rate generator * Square wave generator Binary rate multiplier * Complex waveform generator Complex motor controller | | | | | |
| **15.** | **What are the basic modes of operation of 8255**? | | **REMEMBER**  **BT-L1** | **CO2** | **P 1.2.1** |
| There are two basic modes of operation of 8255. They are:   * I/O mode, There are three modes. * Mode 0 - Simple input/output   - Outputs are latched. Inputs are buffered. Do not have handshake.   * Mode 1 - Input/output with handshake   - Input or output data transfer is controlled by handshake signals.   * Mode 2 - Bidirectional I/O data transfer   - Bidirectional. Both inputs and outputs are latched.   * BSR mode * The individual bits of Port C can be set or reset by sending out a single OUT instruction to the control register. * When Port C is used for control/status operation, this feature can be used to set or reset individual bits. | | | | | |
| **16.** | **What are the advantages of Programmable Interval Timer / Counter IC?** | | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 2.2.4** |
| The 8253 is a programmable counter/timer chip designed for use as an Intel Microcontroller peripheral. The main uses of 8253 are as follows:   * Interrupt a time sharing operating system at evenly spaced intervals so that it can switch a program. * Programmable on shot generator * Serves as a programmable baud rate generator. * Measure time delays between external events * Count the number of times an event occurs. * Causes the processer to be interrupted after a programmed number of external events have occurred. | | | | | |
| **17.** | **List the display modes of 8279 keyboard/ display controller.** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| The display modes of 8279 are:   * Scanned keyboard mode * Scanned sensor matrix * Strobed input. | | | | | |
| **18.** | **Define resolution of A/D converter.** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.4.1** |
| Resolution of A/D converter is defined as a ratio of change in value of input voltage Vi, needed to change the digital output by 1 LSB. If the full scale input voltage required to cause a digital output of all 1’s is ViFS, then the resolution can be given as,  Resolution =ViFS / (2n-1). | | | | | |
| **19.** | **Draw the sample and hold circuit.** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
|  | | | | | |
| **20.** | **State the advantages of DMA.** | | **REMEMBER**  **BT-L1** | **CO2** | **PI 2.2.4** |
| The advantages of DMA are:   * It allows the transfer between I/O and memory without CPU intervention. * It is a hardware control transfer and hence it is very fast. * Increased throughput * Reduced power requirements | | | | | |
| **21** | **List the advantages and disadvantages of parallel communication over serial communication.** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 2.2.4** |
| |  |  |  | | --- | --- | --- | | Basis For Comparison | Serial Transmission | Parallel Transmission | | Meaning | Data flows in bi-direction,  bit by bit. | Multiple lines are used to send data i.e. 8 bits or 1 byte at a time. | | Cost | Economical | Expensive | | Bits transferred at 1 clock pulse | 1 bit | 8 bits or 1 byte | | Speed | Slow | Fast | | Applications | Used for long distance communication. Ex.PC to PC. | Short distance. Ex. Computer to printer. | | | | | | |
| **22** | **What is key bouncing?** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| * Bouncing is the tendency of any two metal contacts in an electronic device to generate multiple signals as the contacts close or open. * Mechanical switches are used as keys in most of the keyboards. * When a key is pressed the contact bounce back and forth and settle down only after a small time delay (about 20ms). * Even though a key is actuated once, it will appear to have been actuated several times. This problem is called Key Bouncing. | | | | | |
| **23** | **What are the modes used in keyboard mode?** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 2.2.4** |
| The modes used in keyboard mode are:   * Scanned keyboard mode with 2 key lockout * Scanned keyboard with N – key rollover * Scanned key board with special error mode * Sensor matrix mode * 2-key lock out: Simultaneous key depression is not allowed. * N-key rollover: Each key depression is treated independently from all others | | | | | |
| **24** | **Name the peripheral ICs used for parallel and serial data transfer** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.4.1** |
| The following are the peripheral ICs used for parallel and serial data transfer,   * For Parallel communication : 8255 – Programmable Peripheral Interface * For Serial Communication : 8251 – Serial Communication Interface. | | | | | |
| **25** | **Give the salient features of Programmable Interval Timer**. | | **REMEMBER**  **BT-L1** | **CO3** | **PI 2.2.4** |
| The Programmable Interval Timer is used as:   * Programmable rate generator * Real time clock * Digital one shot * Complex motor controller | | | | | |

**PART B**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | Draw the block diagram of 8279 and explain the function of each. **(13)** | **AU-AM 2021** | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **2** | Explain the operation of DMA controller 8237 with neat diagrams. **(13)** | **AU-AM 2021** | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **3** | (i)Draw the block diagram of the PPI 8255 and explain the ports and modes of the chip.  **(06)**  (ii)Write a program in assembly language to set/reset the following bits of Port C. Use the BSR feature of the chip.  1) PC0 to be set 2) PC7 to be reset 3) PC1 to be set. **(07)** | **AU-ND 2019** | **REMEMBER**  **BT-L1** | **CO3** | **PI 2.1.3** |
| **4** | (i)Draw the connections between an ADC and 8086, using 8255 as an interface. Write a program to generate a triangular waveform using this setup. **(06)**  (ii)Draw the block diagram of the 8251 and discuss how it caters to serial communication. Write the steps in transmitting one byte of data serially. **(07)** | **AU-ND 2019** | **APPLY**  **BT-L3** | **CO3** | **PI 2.1.3** |
| **5** | Explain in detail about interfacing of four LCD digits to 8086. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **6** | How are D/A and A/D interfaces used? Explain. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 2.1.2** |
| **7** | What are interrupt controller and DMA controller? Explain. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **8** | Draw the block diagram and explain the operations of USART. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **9** | Explain in detail about DMA controller with its diagram. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **10** | Draw and explain the functional diagram of keyboard and display controller. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **11** | Draw and explain the functional diagram of parallel communication interfacing chip. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **12** | Discuss in detail about memory mapped I/O and I/O mapped I/O with necessary diagram. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **13** | (i) Explain the function of programmable peripheral interface – Intel 8255. **(07)**  (ii)Draw the block diagram to interface a analog to digital converter with a microprocessor and explain its working. **(06)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **14** | (i) Draw a schematic to interface keyboard and display with 8086 using 8255 and explain. **(07)** (ii)Write notes on Programmable Interval Timers (PIT) 8253 and 8254. **(06)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **15** | With a neat block diagram explain the key board and display controller IC 8279.  **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **16** | (i) With neat block diagram explain the 8251 and its operating modes. **(07)**  (ii)Draw the block diagram of I/O interface & explain in detail. **(06)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **17** | Draw and explain the block diagram of traffic light control system. **(13)** | | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 1.3.1** |
| **18** | Write short notes on  (i)LED display **(08)**  (ii) LCD display **(08)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **19** | Explain in detail about the asynchronous communication. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |
| **20** | Explain in detail about the synchronous communication. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |

**PART –C**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | Draw and explain the block diagram of alarm controller. **(15)** | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 1.3.1** |
| **2** | Draw the block diagram of traffic light control system using 8086. Write the algorithm an ALP for traffic light control system. **(15)** | **APPLY**  **BT-L3** | **CO3** | **PI 2.1.2** |
| **3** | Draw the diagram to interface a stepper motor with 8086 microcontroller and write ALP to run a stepper motor in both forward and reverse direction with delay. **(15)** | **APPLY**  **BT-L3** | **CO3** | **PI 2.1.2** |
| **4** | Describe the different modes of operation of timers/counters in 8086 with its associated registers. **(15)** | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 2.2.4** |
| **5** | Draw the block diagram to interface a analog to digital converter with a microprocessor and explain its working. **(15)** | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.3.1** |

**UNIT IV- MICROCONTROLLER**

**PART A**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **Write a program to mask the 0th and 7th bit using 8051.** | **AU-AM 2021** | **APPLY**  **BT-L3** | **CO4** | **PI 2.1.2** |
| ORG 00H  MOV A, #FFH ; A = FFH = 255D = 11111111B  ANL A, #7E H ; 126 D= 7E H= 01111110B  END | | | | | |
| **2** | **Write a program to find the 2’s complement using 8051.** | **AU-AM 2021** | **APPLY**  **BT-L3** | **CO4** | **PI 2.1.2** |
| ORG 0000H  MOV DPTR, #0030H  MOVX A, @DPTR  CPL A  ADD A, #01H  MOV DPTR, #0090H  MOVX @DPTR, A  MOV PCON, #02H | | | | | |
| **3** | **What are addressing modes for a microcontroller?** | **AU-ND 2019** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.3.1** |
| The addressing modes for a microcontroller are:  **1)** [Immediate addressing mode](http://www.circuitstoday.com/8051-addressing-modes#immediate)  **2)** [Direct addressing mode](http://www.circuitstoday.com/8051-addressing-modes#direct)  **3)**[Register direct addressing mode](http://www.circuitstoday.com/8051-addressing-modes#register-direct)  **4)** [Register indirect addressing mode](http://www.circuitstoday.com/8051-addressing-modes#register-indirect)  **5)**[Indexed addressing mode](http://www.circuitstoday.com/8051-addressing-modes#indexed). | | | | | |
| **4** | **Give the format of the register PSW of 8051 and name each bit.** | **AU-ND 2019** | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| The format of the register PSW of 8051 and name each bit is:   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **CY** | **AC** | **F0** | **RS1** | **RS0** | **OV** | **-** | **P** | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |   CY – Carry Flag; AC – Auxiliary Flag; F0 – Status Flag  OV – Overflow Flag; P – Parity F  RS1 - Register Bank Select Bit 1  RS2 – Register Bank Select Bit 2 | | | | | |
| **5** | **What are the different ways of operand addressing in 8051?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.2.4** |
| **MOV A,#6AH**  Here the data 6A is the operand, often known as source data. When this instruction is executed, the data 6AH is moved to accumulator A. There are 5 different ways to execute this instruction by using 5 addressing modes for 8051. They are (i) [Immediate addressing mode](http://www.circuitstoday.com/8051-addressing-modes#immediate) (ii)[Direct addressing mode](http://www.circuitstoday.com/8051-addressing-modes#direct)(iii)[Register direct addressing mode](http://www.circuitstoday.com/8051-addressing-modes#register-direct) (iv) [Register indirect addressing mode](http://www.circuitstoday.com/8051-addressing-modes#register-indirect)(v)[Indexed addressing mode](http://www.circuitstoday.com/8051-addressing-modes#indexed). | | | | | |
| **6** | **Illustrate the DJNZ instruction.** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.3.1** |
| DJNZ <dest-byte>,<scr-byte>,rel  The DJNZ instruction decrements the byte indicated by the first operand and if the resulting value is not zero, branches to the address specified in the second operand. | | | | | |
| **7** | **Draw the pin diagram of 8051** | | **REMEMBER**  **BT-L1** | **CO4** | **P 1.4.1** |
| The pin diagram for 8051 is: | | | | | |
| **8** | **Which port used as multifunction port? List the signals.** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| * All port pins of port 3 are multifunctional. * Each pin of port 3 can be programmed to use as I/O or as one of the alternate function. * The signals are RD,WR,T1,T0,INT1, INT0,TXD and RXD. | | | | | |
| **9.** | **How microcontroller is different from microprocessor?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.2.4** |
| The difference between microprocessor and microcontroller are given below,   |  |  | | --- | --- | | **Microprocessors** | **Microcontrollers** | | A microprocessor is a single chip CPU | Microcontroller is a single IC, contains a CPU and much of remaining circuitry of a complete computer (e.g., RAM, ROM, serial interface, parallel interface, timer, interrupt handling circuit) | | Microprocessors are commonly used as a CPU in computers | Microcontrollers are found in small, minimum component designs performing control oriented activities | | | | | | |
| **10.** | **How to set 8051 in idle mode?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| **PCON Register: Power control register**  PCON (Power control) register is used to force the 8051 microcontroller into power saving mode. Power control register of 8051 contains two power saving mode bits   * **Bit 1 – PD:**Power Down   **1** = Enable Power Down mode. In this mode, Oscillator clock turned OFF and both CPU and peripherals clock stopped. Hardware reset can cancel this mode. **0** = Disable Power down mode.   * **Bit 0 – IDL:**Idle   **1** = Enable Idle mode. CPU clock turned off whereas internal peripheral module such as timer, serial port, interrupts works normally. Interrupt and H/W reset can cancel this mode.**0** = Disable Idle mode. | | | | | |
| **11.** | **How the processor 8051 knows whether on chip ROM or External program memory is used?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| **Internal ROM**  The 8051 has 4k (4096 locations) of on-chip ROM. This is used for storing the system program. 212 = 4096, therefore the internal ROM address bus is 12 bits wide and internal ROM locations go from 000H to FFFH.  **Internal RAM**  There are 256 bytes of internal RAM on the 8051. 28 = 256, therefore the internal RAM address bus is 8 bits wide and internal RAM locations go from 00H to FFH. The first 128 locations (00H to 7FH) of internal RAM are used by the programmer for storing data while the second 128 locations (80H to FFH) are the Special Function Registers (SFRs). EA-bar - the external access, on pin 31, is used for enabling or disabling the on-chip ROM.   * When tied high (5V), the 8051 executes instructions in internal ROM when executing in the lower 4k (8k for the 8052) of memory. * If tied low the 8051 will always execute instructions in external memory. | | | | | |
| **12.** | **Why it is necessary to have external pull-up for port 0 in 8051** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| * Port 0 needs pull-up resistors to be connected externally if it is to be used as output port, because it has no in built pull up circuit so it can't give +5v as output. | | | | | |
| **14.** | **How the selection of particular register bank is done in 8051?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| The selection is done using the SFR, Program Status Word (PSW) Register.  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **C** | **AC** | **F0** | **RS1** | **RS0** | **OV** | **F1** | **P** |  * RS means Register bank Select bits. RS0 and RS1 are used to select the register banks. * 00 - Register bank 0 * 01 - Register bank 1 * 10 - Register bank 2 * 11 - Register bank 3 | | | | | |
| **13.** | **List the SFRs involved in interrupt programming of 8051.** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| The SFRs involved in interrupt programming of 8051:  (i) SFR/ IE - Interrupts Enable Control  (ii) SFR/ IP - Interrupt Priority control | | | | | |
| **14.** | **How do you place a specific value in DPTR register?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| * DPTR consists of two separate registers: DPH (Data Pointer High) and (Data Pointer Low). * The Data Pointer (DPTR) is the 8051s only user-accessible 16-bit (2-byte) register. The Accumulator, "R" registers, and "B" register are all 1-byte values. * DPTR, as the name suggests, is used to point to data. It is used by a number of commands which allow the 8051 to access external memory. When the 8051 accesses external memory it will access external memory at the address indicated by DPTR. | | | | | |
| **15.** | **What is difference between AJMP and LJMP instruction?** | | **UNDERSTAND**  **BT-L1** | **CO4** | **P2 2.2.4** |
| The difference between the AJMP and LJMP instructions are given below**,**   |  |  | | --- | --- | | **AJMP** | **LJMP** | | Description: Absolute Jump | Description**:** Long jump | | Jumps unconditionally to the indicated code address within 2k Block. | Jumps unconditionally to the specified code address | | | | | | |
| **16.** | **How do you place a specific value in DPTR register?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| * DPTR consists of two separate registers: DPH (Data Pointer High) and (Data Pointer Low). * The Data Pointer (DPTR) is the 8051s only user-accessible 16-bit (2-byte) register. The Accumulator, "R" registers, and "B" register are all 1-byte values. * DPTR, as the name suggests, is used to point to data. It is used by a number of commands which allow the 8051 to access external memory. When the 8051 accesses external memory it will access external memory at the address indicated by DPTR. | | | | | |
| **17.** | **Which of the 8051 ports need pull-up resisters to function as an I/O port?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| Port 0 needs the pull-up resistor to function as an I/O port:   * It can be used for input or output. * To use the pins of port 0 as both input and output each pin must be connected externally to a 10 k pull-up resistor. * This is due to the fact that P0 is an open drain, unlike P1, P2, and P3 * With resistors connected to port 0, in order to make it an input, the port must be programmed by writing 1 to all the bits.   Dual role of port 0:   * Port 0 is also designated as AD0 - AD7, allowing it to be used for both address and data. * When connecting an 8051/31 to an external memory, port 0 provides both address and data. * The 8051 multiplexes address and data through port 0 to save pins. | | | | | |
| **18.** | **What is the difference between MOVX and MOV?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.2.4** |
| **MOV:** Move Memory  **Description:** MOV- copies the value of operand2 into operand1. The value of operand2 is not affected. Both operand1 and operand2 must be in Internal RAM. No flags are affected unless the instruction is moving the value of a bit into the carry bit in which case the carry bit is affected or unless the instruction is moving a value into the PSW register.  **MOVX**: Move Data To/From External Memory (XRAM)   * **Description:** MOVX moves a byte to or from External Memory into or from the Accumulator. | | | | | |
| **19.** | **What is the special function registers used for port operation in 8051?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| The following ports have been used as Bit addressable ports in 8051:   * **PORT P0:** When there is no external memory present, this port acts as a general purpose input/output port. In the presence of external memory, it functions as a multiplexed address and data bus. It performs a dual role. * **PORT P1:** This port is used for various interfacing activities. This 8-bit port is a normal I/O port i.e. it does not perform dual functions. * **PORT P2:** Similar to PORT P0, this port can be used as a general purpose port when there is no external memory but when external memory is present it works in conjunction with PORT PO as an address bus. This is an 8-bit port and performs dual functions. * **PORT P3**: PORT P3 behaves as a dedicated I/O port | | | | | |
| **20.** | **What are the different ways of operand addressing in 8051?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| The addressing modes in 8051 are:   * Immediate Addressing mode. * Register Addressing mode. * Direct Addressing mode. * Indirect Addressing mode. * Relative Addressing mode. * Absolute Addressing mode. * Indexed Addressing mode. | | | | | |
| **21** | **Illustrate the CJNE instruction.** | | **UNDERSTAND**  **BT-L2** | **CO4** | **P 1.4.1** |
| CJNE <dest-byte>,<scr-byte>,rel  The CJNE instruction compares the first two operands and braches to the specified destination if their values are not equal. If the values are the same, execution continues with the next instruction. | | | | | |
| **21** | **Write an 8051 ALP to toggle P1 a total of 200 times. Use RAM location 32H to hold your counter value instead of registers R0-R7.** | | **UNDERSTAND**  **BT-L2** | **CO4** | **P 1.4.1** |
| MOV P1, #55H ; P1=55H  MOV 32H, #200 ; load counter value into RAM loc 32H toggle P1  LOOP: CPL P1  ACALL DELAY  DJNZ 32H, LOOP ; repeat 200 times | | | | | |
| **23** | **What is the function of RET instruction in 8051?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **P 1.4.1** |
| RET: Return from Subroutine  **Description:** RET is used to return from a subroutine previously called by LCALL or ACALL. Program execution continues at the address that is calculated by popping the topmost 2 bytes off the stack. The most-significant-byte is popped off the stack first, followed by the least-significant-byte. | | | | | |
| **24** | **What are the advantages of using a microcontroller in place of a microprocessor?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **P 1.4.1** |
| The advantages of using a microcontroller in place of a microprocessor are:   * Flexibility * Faster speed of execution * Inexpensive * Rigid | | | | | |
| **25** | **What is the hardware and software interrupts of 8051µc? Mention its vector addresses?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **P 1.4.1** |
| The hardware and software interrupts of 8051 and their vector addresses are given below:  **Interrupts Vector address**   * INT0-External hardware interrupt0 - 0003H * T0-Timer 0 interrupt - 000BH * INT1-External hardware interrupt 1 - 001BH * T1-Timer 1 interrupt - 001BH | | | | | |

**PART B**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | With neat sketch explain the architecture of 8051 microcontroller. **(13)** | **AU-AM 2021** | **REMEMBER**  **BT- L1** | **CO4** | **PI 1.4.1** |
| **2** | (i)Explain the different addressing modes of 8051. **(06)**  (ii)List the various instructions available in 8051 microcontroller. **(07)** | **AU-AM 2021** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.1.2** |
| **3** | What are special function registers? Explain. **(13)** | **AU-ND 2019** | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **4** | Discuss on the different addressing modes of 8051 with suitable examples. **(13)** | **AU-ND 2019** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.2.2** |
| **5** | Write the available special function registers in 8051. Explain each register with its format and functions. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **6** | Discuss the ports and its circuits of 8051. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 3.2.2** |
| **7** | Write an ALP using 8051 instructions to receive byte of data serially and put them in P1. Set the baud rate at 4800, 8-bit data, and 1 stop bit. **(13)** | | **EVALUATE**  **BT-L5** | **CO4** | **PI 2.1.3** |
| **8** | Explain the architecture of 8051 with its diagram. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **9** | Write an 8051 ALP to create a square wave of 66 % duty cycle on bit 3 of port 1. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 2.1.3** |
| **10** | (i) Explain the architecture of 8051 microcontroller with neat diagram. **(08)**  (ii)Explain the TMOD function register and its timer modes of operations. **(05)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **11** | (i) Enumerate about the ports available in 8051 microcontroller. **(07)**  (ii)Write an assembly language program for 8051 to find the largest of three numbers.  **(06)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 2.1.3** |
| **12** | (i) Describe the serial interface with 8051 microcontroller. **(06)**  (ii)Write an assembly language program for 8051, to send 20 output lines at P2.0 vary the duration of pulse using NOP. **(07)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 2.1.3** |
| **13** | (i) Explain in detail the memory organization of 8051 microcontroller. **(13)** | | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.4.1** |
| **14** | (i) Draw the pin diagram of 8051 microcontroller and explain the function of each pin. **(08)**  (ii) Discuss briefly the various registers in 8051 microcontroller. **(05)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **15** | (i)Explain the working of program control transfer instructions of 8051 microcontroller**. (07)**  (ii)Write an assembly language program to generate square wave form using on-chip timer. **(06)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 2.1.3** |
| **16** | List the features of 8051 microcontroller. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 2.2.4** |
| **17** | Explain the architecture of 8051 microcontroller with a neat diagram. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **18** | Describe the different mode of operation of timers/counters in 8051 with its associated register.  **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **19** | (i) Explain the different types of addressing modes in 8051. **(07)**  (ii)Explain the arithmetic instructions of 8051. **(06)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 2.1.2** |
| **20** | (i) Explain the memory organization of 8051 microcontroller. **(07)**  (ii) Explain the parallel I/O and serial I/O of 8051. **(06)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **21** | Draw the functional block diagram of 8051 and explain. **(16)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |

**PART - C**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | Describe the architecture of 8051 with neat diagram. **(15)** | **EVALUATE**  **BT-L5** | **CO4** | **PI 1.4.1** |
| **2** | (i) Write a brief notes on external data move operations in 8051. **(05)**  (ii)Write an 8051 assembly language program to add three BCD numbers stored in internal RAM locations 25H, 26H, and 27H and put the result in RAM locations 31H (MSB) and 30 H (LSB). Use register R0 to store the intermediate result. **(10)** | **APPLY**  **BT-L3** | **CO4** | **PI 2.1.2** |
| **3** | Write an assembly language program using 8051 instructions to arrange an array of numbers in ascending order. **(15)** | **APPLY**  **BT-L3** | **CO4** | **PI 2.1.2** |
| **4** | (i) Briefly explain the data transfer instructions available in 8051 Microcontroller **(08)**  (ii) Using timers in 8051, write a program to generate square wave for 100ms with 50% duty cycle. **(07)** | **APPLY**  **BT-L3** | **CO4** | **PI 2.1.2** |
| **5** | (i) How are the timers of 8051 used to produce time delay in timer mode? **(08)**  (ii) Explain the interrupt structure of 8051 microcontroller. **(07)** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.4.1** |

**UNIT V- INTERFACING MICROCONTROLLER**

**PART –A**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **Define the operating model 0 of 8051 serial ports.** | **AU-AM 2021** | **REMEMBER**  **BT-L1** | **CO3** | **PI 1.4.1** |
| Serial port mode :**Mode 0:**   * Only synchronous mode * Data transferred on RXD clock on TXD * Clock is fixed at 1/12 of the oscillator frequency | | | | | |
| **2** | **Give the different types of ADC.** | **AU-AM 2021-2** | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 2.2.4** |
| There are really five major types of ADCs in use today:   * [Successive Approximation (SAR) ADC](https://dewesoft.com/daq/types-of-adc-converters#sar-adc) * [Delta-sigma (ΔΣ) ADC](https://dewesoft.com/daq/types-of-adc-converters#delta-sigma-adc) * [Dual Slope ADC](https://dewesoft.com/daq/types-of-adc-converters#dual-slope-adc) * [Pipelined ADC](https://dewesoft.com/daq/types-of-adc-converters#pipelined-adc) * [Flash ADC](https://dewesoft.com/daq/types-of-adc-converters#flash-adc) | | | | | |
| **3** | **How to program 8051 Timers?** | **AU-ND 2019** | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| * The timers in the 8051 have both start and stop signal.. * The start and stop of the timer are controlled by way of software by the TR (timer start) bits TRO and TR1. * And it is achieved by the instructions “SETB TR1″ and “CLR TR1″ for Timer 1, and “SETB TRO” and “CLR TRO” for Timer 0. | | | | | |
| **4** | **What are the types of sensor used for interfacing?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.2.4** |
| The types of sensor used for interfacing are   * Temperature Sensor * IR Sensor * Ultrasonic Sensor * Touch Sensor, Proximity Sensors * Pressure Sensor, Level Sensors * Smoke and Gas Sensors | | | | | |
| **5.** | **Give two examples of sensors and state its uses.** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.2.4** |
| The examples of sensors are:   * Thermocouples have the widest temperature range of all the temperature sensors from below -200oC to well over 2000oC. * LM35 Temperature sensors are used in measuring temperature of a particular environment and HVAC applications, providing thermal shutdown [for a component](http://www.efxkits.us/what-are-the-electrical-components-used-in-electronic-projects/)/ circuit and checking battery temperature. | | | | | |
| **6** | **What is PWM?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| * A modulation technique that uses a digital circuit to create a variable analog signal PWM is a simple concept open and close a switch at uniform repeatable intervals. * For a given fixed load to maintain a steady speed by using pulse width modulation. * By changing the width of the pulse applied to the DC motor we can increase or decrease the amount of power provided to the motor, thereby increasing or decreasing the motor speed. | | | | | |
| **7.** | **List the modes of timer in 8051.** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| Four timer modes are available in 8051:   |  |  |  |  | | --- | --- | --- | --- | | **M1** | **M2** | **Mode** | **Description** | | 0 | 0 | Mode 0 | 13-bit Timer | | 0 | 1 | Mode 1 | 16-bit Timer | | 1 | 0 | Mode 2 | 8-bit auto reload | | 1 | 1 | Mode 3 | Split Timer mode | | | | | | |
| **8.** | **State how baud rate is calculated for serial data transfer in mode 1.** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.4.1** |
| * Using Timer/Counter 1 to generate Baud rates,   Baud rate = K x Oscillator frequency/ 32 x 12 x [(256-TH1]  If SMOD=0 then K=1  If SMOD=1 then K=2 (SMOD is the PCON register)   * Using Timer/Counter 1 to generate Baud rates,   Baud rate = Timer 2 overflow rate/16 | | | | | |
| **9** | **Which register is used for serial programming in 8051 microcontroller?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| * The SCON register is used for serial programming in 8051. It is an 8-bit register used to program the start bit, stop bit, and data bits of data framing, among other things. * SMO and SM1 are D7 and D6 of the SCON register, respectively. These two bits determine the framing of data by specifying the number of bits per character, and the start and stop bits. They take the following combinations.  |  |  |  | | --- | --- | --- | | **SM0** | **SM1** | **Description** | | 0 | 0 | Serial mode 0 | | 0 | 1 | Serial mode 1, 8-bit data, 1 stop bit , 1 start bit | | 1 | 0 | Serial mode 2 | | 1 | 1 | Serial mode 2 | | | | | | |
| **10.** | **Give the schematic to interface a relay with microcontroller?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| * **The schematic to interface a relay with microcontroller is:** | | | | | |
| **11.** | **List the 8051 interrupts with its priority.** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.2.4** |
| The 8051 interrupts with its high to low priority are,   * External Interrupt 0 (INT0) * Timer Interrupt 0 (TFO) * External Interrupt1 (INT1) * Timer Interrupt1 (TF1) * Serial communication (RI+TI) | | | | | |
| **12.** | **Write about the design steps involved in using microcontroller for stepper motor.** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 3.1.6** |
| * Motor has two phases, with center-tap winding. The center taps of these windings are connected to the 12V supply. Due to this, motor can be excited by grounding four terminals of the two windings. * Motor can be rotated in steps by giving proper excitation sequence of these windings. * The lower nibble of port A of the 8255 is used to generate excitation signals in the proper sequence. * Microcontroller is programmed to rotate the stepper motor. | | | | | |
| **13.** | **State the importance of relay coil.** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| * A relay is usually an electromechanical device that is actuated by an electrical current. The current flowing in one circuit causes the opening or closing of another circuit. * Relays are like remote control switches and are used in many applications because of their relative simplicity, long life, and proven high reliability. * Relays are used in a wide variety of applications throughout industry, such as in telephone exchanges, digital computers and automation systems * All relays contain a sensing unit, the electric coil, which is powered by AC or DC current. | | | | | |
| **14.** | **Why the relays are called electromagnetic relays?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.3.1** |
| * An iron core is surrounded by a control coil. As shown, the power source is given to the electromagnet through a control switch and through contacts to the load.      * When current starts flowing through the control coil, the electromagnet starts energizing and thus intensifies the magnetic field. Thus the upper contact arm starts to be attracted to the lower fixed arm and thus closes the contacts causing a short circuit for the power to the load. * On the other hand, if the relay was already de-energized when the contacts were closed, then the contact move oppositely and make an open circuit. | | | | | |
| **15.** | **What is meant by I2C standard?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| In I2C only two bi-directional lines Serial Data (SDA) & Serial Clock (SCL) are required to carry information between the devices connected to the bus. Each I2C device is recognized by a unique 7-bit address.  The device that initiates the communication is called MASTER. The master controls the clock signal. Whereas the device being addressed by the Master is called as SLAVE. Generation of clock signals on the I2C-bus is always the responsibility of master devices; each master generates its own clock signals when transferring data on the bus. | | | | | |
| **16.** | **Write an ALP to receive input from port P1.5 and if it is high then an output 35H is sent to Port 0.** | | **APPLY**  **BT-L3** | **CO4** | **PI 2.2.1** |
| L1: JNB P1.5, L1 ; Jump if no bit P1.5 is set  MOV A, #35H; Move 35H to Accumulator  MOV P0, A ; Move the Contents of Accumulator to Port P0 | | | | | |
| **17.** | **What are precautions required while interfacing microprocessor with motors?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| While interfacing microprocessor with motors, the following precautions are made:   * Care should be taken to assure lifting in the direction intended in the design of the lifting means. * Data lines for microprocessor should be connected to the data lines of the Interfacing a device which is connected to the motor. | | | | | |
| **18.** | **Why do we need relay interface? What is SPDT relay**? | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| * Relay interface allows the isolation of two separate section of a system with two different voltage sources. * SPDT relay is a single pole double throw relay. | | | | | |
| **19.** | **What are the sources of interrupts in 8051?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.3.1** |
| In general, the five sources of interrupts in 8051are:   * Timer 0 overflow interrupt - TF0, Timer 1 overflow interrupt - TF1 * External hardware interrupt - INT0,INT1 * Serial communication interrupt - RI/TI * The timer and serial interrupts are internally generated by the microcontroller. * External interrupts are generated by additional [interfacing devices](https://www.elprocus.com/peripherals-interfacing-to-the-microcontroller-8051-in-electronics/) or switches that are externally connected to the microcontroller. These external interrupts can be edge triggered or level triggered. | | | | | |
| **20.** | **How is the microcontroller used for the traffic light control application.** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.4.1** |
| * When the power is applied to the circuit microcontroller initializes its ports for driver, status LED indicator timer on/off switch jumpers for variable pass time and enters in infinite loop where it checks for the timer on/off switch. * If it is enabled, timer is on and signal lamps switches in sequence if the timer is not enabled all the roads are cleared. * Every time, it checks for the jumper to determine the pass time for vehicles | | | | | |
| **21.** | **List the applications of Microcontroller.** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| The applications of microcontroller are listed below:   * Light sensing & controlling devices * Temperature sensing and controlling devices * Fire detection & safety devices * Industrial instrumentation devices * Process control devices | | | | | |
| **22.** | **What is the function of SM2 bit in the SCON register of 8051?** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.4.1** |
| * SM2 is a flag for "Multiprocessor communication." Generally, whenever a byte has been received the 8051 will set the "RI" (Receive Interrupt) flag. This lets the program know that a byte has been received and that it needs to be processed. * However, when SM2 is set the "RI" flag will only be triggered if the 9th bit received was a "1". That is to say, if SM2 is set and a byte is received whose 9th bit is clear, the RI flag will never be set. This can be useful in certain advanced serial applications. | | | | | |
| **23.** | **What are the functions performed by 8279?** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| Hence the following are the functions are performed by 8279   * Key board scanning * Key code generation and * Display refreshing | | | | | |
| **24.** | * + - * **Draw Draw the TCON Register Format.** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| TCON is bit addressable. The address of TCON is 88H. It is partly related to Timer and partly to interrupt.   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **TF1** | **TR1** | **TF0** | **TR0** | **IE1** | **IT1** | **IE0** | **IT0** |   **TCON Register Format**   * **TF1**:Timer1 overflow flag. It is set when timer rolls from all 1s to 0s. It is cleared when   processor vectors to execute ISR located at address 001BH.   * **TR1**:Timer1 run control bit. Set to 1 to start the timer / counter. * **TF0**:Timer0 overflow flag. (Similar to TF1) * **TR0**:Timer0 run control bit. * **IE1**:Interrupt1 edge flag. Set by hardware when an external interrupt edge is detected. It is cleared when interrupt is processed. * **IE0**:Interrupt0 edge flag. (Similar to IE1) * **IT1**:Interrupt1 type control bit. Set/ cleared by software to specify falling edge / low level triggered external interrupt. **IT0**:Interrupt0 type control bit. (Similar to IT1) | | | | | |
| **25.** | **Give some example of input devices to microprocessor - based systems.** | | **REMEMBER**  **BT-L1** | **CO2** | **PI 1.4.1** |
| The input devices used in the microprocessor- based system are:   * Keyboards * DIP switches * ADC * Floppy disc, etc. | | | | | |

**PART B**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | With neat sketch block diagram of interfacing 64KB of External RAM and 64KB of External ROM with the 8051 Microcontroller. **(13)** | **AU-AM 2021** | **UNDERSTAND**  **BT-L2** | **CO3** | **PI 3.1.6** |
| **2** | Draw and explain the ADC interfacing using 8051. **(13)** | **AU-AM 2021** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 3.1.6** |
| **3** | Write and explain. What is known as Serial Port Programming? **(13)** | **AU-ND 2019** | **REMEMBER**  **BT-L1** | **CO4** | **PI 2.1.2** |
| **4** | What are sensor interfacing and external memory interfacing? Explain. **(13)** | **AU-AM 2019** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 3.1.6** |
| **5** | Draw and explain the block diagram of alarm controller. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **6** | (i)Interface the ADC converter with 8051 and explain with neat diagram. **(07)**  (ii)Write the assembly language program to execute the ADC conversion. **(06)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **7** | Write a program for generation of unipolar square waveform of 1 kHz frequency using Timer0 in mode0. Consider the system frequency as 12 MHz. **(13)** | | **APPLY**  **BT-L3** | **CO4** | **PI 2.1.2** |
| **8** | Demonstrate the interfacing of the stepper motor with 8051 and explain its interfacing diagram and develop program to rotate the motor in clock wise direction. **(13)** | | **CREATE**  **BT-L6** | **CO4** | **PI 2.4.2** |
| **9** | Explain 8051 serial port programming with examples. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **10** | (i) Explain stepper motor control using 8051 microcontroller. **(07)**  (ii) Explain in detail about the interrupt programming in 8051. **(06)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **11** | (i) Explain the interfacing of 4 x 4 matrix keyboard with 8051 microcontroller. **(08)**  (ii) Write shortly on the various operating modes for serial port of 8051. **(05)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **12** | Explain the interfacing of keyboard with 8015 microcontroller. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **13** | Write a program for counter 1 in mode 2 to count the pulses and display the state of TL1Count on PORT 2. Assume that clock input is connected to T1 pin (P 3.5).  **(13)** | | **APPLY**  **BT-L3** | **CO4** | **PI 1.4.1** |
| **14** | Describe in detail the microcontroller based system design with an Example. **(13)** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.1.2** |
| **15** | List any four application of stepper motor? Explain how to interface stepper motor with µc. **(13)** | | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 2.2.4** |
| **16** | What is the need for PWM in motor control application? Explain. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **17** | With a neat functional diagram show how an 8051 microcontroller is in controlling a stepper motor. **(13)** | | **REMEMBER**  **BT-L1** | **CO4** | **PI 1.4.1** |
| **18** | Explain the different jump instructions in 8051. **(13)** | | **UNDERSTAND**  **BT-L2** |  | **PI 1.4.1** |
| **19** | Discuss timers of 8051 microcontroller. **(13)** | | **REMEMBER**  **BT-L1** |  | **PI 2.2.4** |

**PART C**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | Write an assembly language program using 8051 instructions to arrange an array of numbers in ascending order. **(15)** | **APPLY**  **BT-L3** | **CO4** | **PI 2.2.1** |
| **2** | With diagram explain the interfacing of a stepper motor with a 8051 microcontroller and write an 8051 ALP to run the stepper motor in both forward and reverse direction with delay. **(15)** | **APPLY**  **BT-L3** | **CO4** | **PI 2.2.1** |
| **3** | (i)Describe the different modes of operation of timer/counters in 8051 with its associated register. **(08)**  (ii)How does one interface a 16 x 2 LCD display using 8051 microcontroller? **(07)** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.4.1** |
| **4** | Describe the different modes of operation of timer/counters in 8051 with its associated register. **(15)** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.4.1** |
| **5** | Illustrate the serial communication in 8051, with its special function register. **(15)** | **UNDERSTAND**  **BT-L2** | **CO4** | **PI 1.4.1** |