# Model Viva Questions for Advanced Microprocessor and Microcontroller lab

## Class: ET&T (V Semester)

## Title of the Practical: Identify different components/unit in 8086 kit

Q1: What is 8086?
A1: The 8086 (also called iAPX86) is a 16-bit microprocessor chip designed by Intel between early 1976 and mid-1978, when it was released. The 8088, released in 1979, was a slightly modified chip with an external 8-bit data bus (allowing the use of cheaper and fewer supporting logic chips and is notable as the processor used in the original IBM PC.
Q2: what is buses and operation?
A2: All internal registers as well as internal and external data buses were 16 bits wide, firmly establishing the "16-bit microprocessor" identity of the 8086. A 20-bit external address bus gave a 1 MB physical address space ($2^{20}$ = 1,048,576). This address space was addressed by means of internal 'segmentation'. The data bus was multiplexed with the address bus in order to fit a standard 40-pin dual in-line package. 16-bit I/O addresses meant 64 KB of separate I/O space ($2^{16}$ = 65,536). The maximum linear address space was limited to 64 KB, simply because internal registers were only 16 bits wide. Programming over 64 KB boundaries involved adjusting segment registers (see below) and was therefore fairly awkward (and remained so until the 80386).
Q3 how many register 8086 contain?
A3 The 8086 had eight (more or less general) 16-bit registers including the stack pointer, but excluding the instruction pointer, flag register and segment registers. Four of them, AX, BX, CX, DX, could also be accessed as twice as many 8-bit registers (see figure) while the other four, BP, SI, DI, SP, were 16-bit only.
Q4what is interrupt?
A4 In computing, an interrupt is an asynchronous signal indicating the need for attention or a synchronous event in software indicating the need for a change in execution.
Q5 what is stack?
A5 In computer science, a stack is a last in, first out (LIFO) abstract data type and data structure. A stack can have any abstract data type as an element, but is characterized by only two fundamental operations: push and pop. The push operation adds an item to the top of the stack, hiding any items already on the stack, or initializing the stack if it is empty. The pop operation removes an item from the top of the stack, and returns this value to the caller. A pop either reveals previously concealed items, or results in an empty stack.
Q6 what is return address?
A6 In postal mail, a return address is an explicit inclusion of the address of the person sending the message. It provides the recipient (and sometimes authorized intermediaries) with a means to determine how to respond to the sender of the message if needed.
Q7 what is instruction set?
A7 An instruction set, or instruction set architecture (ISA), is the part of the computer architecture related to programming, including the native data types, instructions, registers, addressing modes, memory architecture, interrupt and exception handling, and external I/O.
Q8 what kind of flags is used in 8086?
A9 8086 has a 16 bit flag register. Out of these, 9 are active, and indicate the current state of the processor. These are — Carry flag, Parity flag, Auxiliary flag, Zero flag, Sign flag, Trap flag , Interrupt flag , Direction flag and Overflow flag.
Q9 what are the two main components in 8086?
A9. 1 program memory 2. Data memory 3. Microprocessor 4. Input port 5. Output port 6. Clock generator

Q10 what is driver interface?
A10 Drivers or buffers are also used in microprocessor interface design. However, they are usually only needed when the loading on the microprocessor data, address or control lines is too high. And that is usually because the microprocessor address, data or control lines are connected in parallel to large number of external electronic components. Another situation that requires drivers is when the microprocessor is connected to long cables. Drivers are needed in both cases so that delay times, rise and fall times and noise levels do not adversely affect the reliability of the microprocessor's data and calculations.

## Title of the Practical: Execute a sample program

Q1 what are the common concept in programming?
A1 Learning any imperative programming language involves mastering a number of common concepts: Variables: declaration/definition Assignment: assigning values to variables Input/Output: Displaying messages Displaying variable values Control flow: if-then Loops Subprograms: Definition and Usage Programming
Q2 what is variables?
A2 VariablesFor the moment we will skip details of variable declaration and simply use the 8086 registers as the variables in our programs. Registers have predefined names and do not need to be declared.
Q3 how many 16 bit register s are used in 8086?
A3 The 8086 has 14 registers. Each of these is a 16-bit register. Initially, we will use four of them – the so called the general-purpose registers: ax, bx, cx, dx
Q4 give example to Store the ASCII code for the letter A in register bx.
A4 Example: Store the ASCII code for the letter A in register bx. mov bx, ‘A’ the mov instruction also allows you to copy the contents of one register into another register.
Q5 give example by which we can loads the value 2 into bx
A5 Example: mov bx, 2
mov cx, bx The first instruction loads the value 2 into bx where it is stored as a binary number. [A number such as 2 is called an integer constant]
Q6 Why the comma is essential?
A6 It is used to separate the two operands. A missing comma is a common syntax error.
Q7what is comment?
A7 Anything that follows semi-colon (;) is ignored by the assembler. It is called a comment. Comments are used to make your programs readable. You use them to explain what we are doing in English.
Q8 what are add, inc, dec and sub instructions?
A8 The 8086 provides a variety of arithmetic instructions The add instruction adds the source operand to the destination operand, leaving the result in the destination operand The inc instruction takes one operand and adds 1 to it. It is provided because of the frequency of adding 1 to an operand in programming. The dec instruction like inc takes one operand and subtracts 1from it. This is also a frequent operation in programming. The sub instruction subtracts the source operand from the destination operand leaving the result in the destination operand.
Q 9 Write a code fragment to display the character ’a’
On the screen:

```
A 9    mov dl, ‘a‘; dl = ‘a‘
       mov ah, 2h; character output subprogram
        int 21h; call ms-dos output character
```

Q 10 Write a code fragment to read a character from the keyboard:

```
A 10            mov ah, 1h; keyboard input subprogram
                int 21h; character input
                         ; Character is stored in al]
```

## Title of practical: Develop an assembly language program for addition and subtraction of 8 bit numbers.

Q1: What is Intel 8051?
A1: The Intel 8051 microcontroller is one of the most popular general purpose microcontrollers in use today. The success of the Intel 8051 spawned a number of clones which are collectively referred to as the MCS-51 family of microcontrollers, which includes chips from vendors such as Atmel, Philips, Infineon, and Texas Instruments.The Intel 8051 s a Harvard architecture
Q2: What is Harvard architecture?
A2: The Harvard architecture is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters.
Q3 Enhanced 8051 devices are manufactured by?
A3 Enhanced 8051-compatible devices manufactured by more than 20 independent manufacturers including Atmel, Infineon Technologies (formerly Siemens AG), Maxim Integrated Products (via its Dallas Semiconductor subsidiary), NXP (formerly Philips Semiconductor), Nuvoton (formerly Winbond), ST Microelectronics, Silicon Laboratories (formerly Cygnal), Texas Instruments, Ramtron International, Silicon Storage Technology, and Cypress Semiconductor.
Q4: What are the functions provide by 8051?
A4: The 8051 architecture provides many functions CPU, RAM, ROM, I/O, interrupt logic, timer, etc, in a single package
Q5: What are the features of 8051?
A5: List the features of 8051 micro controllers:-

1) Single supply +5v operation using HMOS technology.
2) 4096 bytes program memory on-chip.
3) 128 data memory on chip
4) 4 register banks
5) 2 multiple modes, 16 bit timer/counter
6) Extensive Boolean processing capabilities
7) 64KB external RAM size.
8) 32 bi-directional I/O lines.

Q6: What is a particularly useful feature of the 8051?
A6: A particularly useful feature of the 8051 core is the inclusion of a Boolean processing engine which allows bit-level Boolean logic operations to be carried out directly and efficiently on internal registers and RAM. This feature helped cement the 8051's popularity in industrial control applications. Another valued feature is that it has four separate register sets, which can be used to greatly reduce interrupt latency compared to the more common method of storing interrupt context on a stack.
Q7: What is MCS-51 UARTs?
A7: MCS-51 UARTs make it simple to use the chip as a serial communications interface. External pins can be configured to connect to internal shift registers in a variety of ways, and the internal timers can also be used, allowing serial communications in a number of modes, both synchronous and asynchronous.
Q8: What is the function of UART and timer?
A8: Once a UART, and a timer if necessary, have been configured, the programmer needs only to write a simple interrupt routine to refill the send shift register whenever the last bit is shifted out by the UART and/or empty the full receive shift register (copy the data somewhere else). The main program then performs serial reads and writes simply by reading and writing 8-bit data to stacks.
Q9: What are the components of MCS-51 based microcontrollers?
A9: MCS-51 based microcontrollers typically include one or two UARTs, two or three timers, 128 or 256 bytes of internal data RAM (16 bytes of which are bit-addressable), up to 128 bytes of I/O, 512 bytes to 64 kB of internal program memory, and sometimes a quantity of extended data RAM (ERAM) located in the external data space.

Q10: In 8051 how many clock cycles per machine cycle?
A10: The original 8051 core ran at 12 clock cycles per machine cycle. With most instructions executing in one or two machine cycles.

## Title of practical: Develop a assembly language program for bit logical operation

Q1: What are the features of modern 8051 based microcontrollers?
A1: Common features included in modern 8051 based microcontrollers include built-in reset timers with brown-out detection, on-chip oscillators, self-programmable Flash ROM program memory, boot loader code in ROM, EEPROM non-volatile data storage, I²C, SPI, and USB host interfaces, CAN or LIN bus, PWM generators, analog comparators, A/D and D/A converters, RTCs, extra counters and timers, in-circuit debugging facilities, more interrupt sources, and extra power saving modes.
Q2: Can you explain Memory Architecture of 8051?
A2: The MCS-51 has four distinct types of memory – internal RAM, special function registers, program memory, and external data memory.
Q3: What is Internal RAM (IRAM)?
A3: Internal RAM (IRAM) is located from address 0 to address 0xFF. IRAM from 0x00 to 0x7F can be accessed directly, and the bytes from 0x20 to 0x2F are also bit-addressable. IRAM from 0x80 to 0xFF must be accessed indirectly, using the @R0 or @R1 syntax, with the address to access loaded in R0 or R1.
Q4: What is Special function registers (SFR)?
A4: Special function registers (SFR) are located from address 0x80 to 0xFF, and are accessed directly using the same instructions as for the lower half of IRAM. Some of the SFR's are also bit-addressable.
Q5: What is Program memory (PMEM)?
A5: Program memory (PMEM, though less common in usage than IRAM and XRAM) is located starting at address 0. It may be on- or off-chip, depending on the particular model of chip being used. Program memory is read-only, though some variants of the 8051 use on-chip flash memory and provide a method of re-programming the memory in-system or in-application. Aside from storing code, program memory can also store tables of constants that can be accessed by MOVC A, @DPTR, using the 16-bit special function register DPTR.
Q6: What is External data memory (XRAM)?
A6: External data memory (XRAM) also starts at address 0. It can also be on- or off-chip; what makes it "external" is that it must be accessed using the MOVX (Move eXternal) instruction. Many variants of the 8051 include the standard 256 bytes of IRAM plus a few KB of XRAM on the chip. If more XRAM is required by an application, the internal XRAM can be disabled, and all MOVX instructions will fetch from the external bus.
Q7: What are the hardware features of 8051?
A7: 8051 has specific hardware features such as the multiple register banks and bit manipulation instructions. There are many commercial C compilers. SDCC is a popular open source C compiler. Other high level languages such as Forth, BASIC, Pascal/Object Pascal, PL/M and Modula-2 are available for the 8051, but they are less widely used than C and assembly.
Q8: What are the addressing modes of 8051?
A8: List the addressing modes of 8051:-
Direct addressing
Register addressing
Register indirect addressing
Implicit addressing
Immediate addressing
Index addressing
Bit addressing
Q9: What type of operations allows any addressing mode?
A9: Many of the operations allow any addressing mode for the source or the destination, for example, MOV 020h; 03fh will copy the value in memory location 0x3f in the internal RAM to the memory location 0x20, also in internal RAM.
Q10: Why in the 8051 all arithmetic operations must use the accumulator?
A10: Because the 8051 is an accumulator-based architecture, all arithmetic operations must use the accumulator, e.g. ADD A, 020h will add the value in memory location 0x20 in the internal RAM to the accumulator.

## Title of practical: Develop 8051 assembly language program on trainer kit for multiplication and division of two 8 bit numbers

Q1Define DA A?

A1This is a decimal adjust instruction

It adjusts the 8-bit value in ACC resulting from operations like ADD or ADDC and produces two 4-bit digits (in packed Binary Coded Decimal (BCD) format Effectively, this instruction performs the decimal conversion by adding 00H, 06H, 60H or 66H to the accumulator,depending on the initial value of ACC and PSW

If ACC bits A3-0 are greater than 9 (xxxx1010-xxxx1111), or if AC=1, then a value 6 is added to the accumulator to produce a correct BCD digit in the lower order nibble.

If CY=1, because the high order bits A7-4 is now exceeding 9 (1010xxxx-1111xxxx), then these high order bits will be increased by 6 to produce a correct proper BCD in the high order nibble but not clear the carry.

Q2 Define Logical EX OR?

A2 This instruction performs the logical XOR (Exclusive OR) operation on the source and destination operands and stores the result in the destination variable

No flags are affected

Example:XRL A,R0

If ACC=C3H (11000011) and R0=AAH (10101010), then the instruction results in ACC=69H (01101001)

Q3 Define CLR A instruction?

A3 CLR A This instruction clears the accumulator (all bits set to 0)

No flags are affected If ACC=C3H, then the instruction results in ACC=00H.

Q4 Define RRC A instruction?

A4 The instruction rotates the accumulator contents one bit to the right through the carry flag

The original value of carry flag will move into Bit 7 of the accumulator and Bit 0 rotated into carry flag. No other flags are affected If ACC=C3H (11000011), and the carry flag is 0, the instruction results in ACC=61H (01100001) with the carry.

Q5 Define PUSH Direct?

A5 This instruction increments the stack pointer (SP) by 1The contents of Direct, which is an internal memory location or a SFR, are copied into the internal RAM location addressed by the stack pointer No flags are affected Example: PUSH 22H PUSH 23H

Initially the SP points to memory location 4FH and the contents of memory locations 22H and 23H are 11H and 12H respectively. After the above instructions, SP=51H, and the internal RAM locations 50H and 51H will store 11H and 12H respectively.

Q6 Define POP instruction?

A6 This instruction reads the contents of the internal RAM location addressed by the stack pointer (SP) and decrements the stack pointer by 1. The data read is then transferred to the Direct address which is an internal memory or a SFR. No flags are affected.

Example:

POP DPH

POP DPL

 If SP=51H originally and internal RAM locations 4FH, 50H and 51H contain the values 30H, 11H and 12H respectively, the instructions above leave SP=4FH and DPTR=1211H

POP SP If the above line of instruction follows, then SP=30H. In this case, SP is decremented to 4EH before being loaded with the value popped (30H).

Q7Define XCH instruction?

A7 This instruction swaps the contents of ACC with the contents of the indicated data byte

Example: XCH A,@R0

Suppose R0=2EH, ACC=F3H (11110011) and internal RAM location 2EH=76H (01110110). The result of the above instruction leaves RAM location 2EH=F3H and ACC=76H.

Q8 Define CLR<bit> instruction?

A8 This operation clears (reset to 0) the specified bit indicated in the instruction  No other flags are affected  CLR instruction can operate on the carry flag or any directly addressable .

Q9What is the operation performed by SETB<bit>?

A9This operation sets the specified bit to 1 SETB instruction can operate on the carry flag or any directly-addressable bit.No other flags are affected

Q10 Define CPL <bit> instruction?

A10 This operation complements the bit indicated by the operand No other flags are affected
 CPL instruction can operate on the carry flag or any directly addressable.

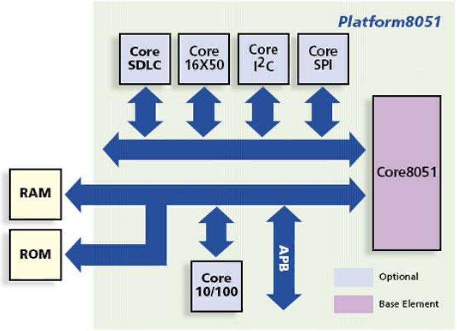
## Title of the Practical: Interface ELC matrix display

Q1 what is interface?
A1 In the field of computer science, an interface refers to a point of interaction between components, and is applicable at the level of both hardware and software. This allows a component, whether a piece of hardware such as a graphics card or a piece of software such as an internet browser, to function independently while using interfaces to communicate with other components via an input/output system and an associated protocol.
Q2 what is hardware and software interface?
A2 In addition to hardware and software interfaces, a computing interface may refer to the means of communication between the computer and the user by means of peripheral devices such a monitor or a keyboard, an interface with the internet via Internet Protocol, and any other point of communication involving a computer.
Q3 explain software interface?
A3 software interface may refer to a range of different types of interface at different "levels": an operating system may interface with pieces of hardware, applications or programs running on the operating system may need to interact via streams, and in object oriented programs, objects within an application may need to interact via methods.
Q4 what is constants?
A4 In computer programming, a constant is a special kind of variable whose value cannot typically be altered by the program during its execution (though in some cases this can be circumvented, e.g. using self-modifying code). Many programming languages make an explicit syntactic distinction between constant and variable symbols.
Q5 what is programming against software interface?
A5 The use of interfaces allows implementation of a programming style called programming against interfaces. The idea behind this is to base the logic one develops on the sole interface definition of the objects one uses and not to make the code depend on the internal details. This allows the programmer the ability to later change the behavior of the system by simply swapping the object used with another implementing the same interface. Pushing this idea to the limit one can introduce the inversion of control which means leaving the context to inject the code with the specific implementations of the interface that will be used to perform the work.
Q6 explain hardware interface?
A6 Hardware interfaces exist in computing systems between many of the components such as the various buses, storage devices, other Input/Output devices, etc. A hardware interface is described by the mechanical, electrical and logical signals at the interface and the protocol for sequencing them (sometimes called signaling).[1] A standard interface, such as SCSI, decouples the design and introduction of computing hardware, such as I/O devices, from the design and introduction of other components of a computing system, thereby allowing users and manufacturers great flexibility in the implementation of computing systems.[1]
Q7 what is protocol?
A7 The protocol is a description of 1the messages that are understood by the object,2 the arguments that these messages may be supplied with, and3 the types of results that these messages return.4the invariants that are preserved despite modifications to the state of an object5the exceptional situations that will be required to be handled by clients to the object.
Q8 what is display device?
A8 A display device is an output device for presentation of information for visual, tactile or auditive reception, acquired, stored, or transmitted in various forms. When the input information is supplied as an electrical signal, the display is called electronic display.
Q9 what is dot matrix display?
A9 A dot matrix display is a display device used to display information on machines, clocks, railway departure indicators and many and other devices requiring a simple display device of limited resolution. The display consists of a matrix of lights or mechanical indicators arranged in a rectangular configuration (other shapes are also possible, although not common) such that by switching on or off selected lights, text or graphics can be displayed. A dot matrix controller

converts instructions from a processor into signals which turns on or off lights in the matrix so that the required display is produced.
Q10 what are the common size of dot matrix display?
A10 Common sizes of dot matrix displays: (1)128×16 (Two lined),(2)128×32 (Four lined)(3) 192×64 (Eight lined)

# Title of practical: Interfacing design for A/D and D/A converter

Q1: Draw the block diagram of 8051?
A1:



*Platform8051* Block Diagram

Q2: What are the applications of 8051 chip?
A2: 8051 chips are used in a wide variety of control systems, telecom applications, robotics as well as in the automotive industry.
Q3: Can you explain PIN 9 for 8051?
A3: PIN 9: PIN 9 is the reset pin which is used reset the microcontroller's internal registers and ports upon starting up. (Pin should be held high for 2 machine cycles.)
Q4: What are the function of PIN 40 and 20?
A4: PIN 40 and 20: Pins 40 and 20 are VCC and ground respectively. The 8051 chip needs +5V 500mA to function properly, although there are lower powered versions like the Atmel 2051 which is a scaled down version of the 8051 which runs on +3V.
Q5: What is the program start address of 8051?
A5: The 8051 starts executing program instructions from address 0x00 in the program memory.
Q6: What is meant by micro controller?
A6: A device which contains the microprocessor with integrated peripherals like memory, serial ports, parallel ports, timer/counter, interrupt controller, data acquisition interfaces like ADC, DAC is called micro controller
Q7: Explain the operating mode 0 of 8051 serial port?
A7: In this mode serial data enters and exists through RXD, TXD outputs the shift clock. 8-bits are transmitted or received: 8-data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.
Q8: Explain the mode 3 of 8051 serial port?
A8: In this mode, 11 bits are transmitted (through TXD) or (received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1).It is same as mode 2 except the baud rate. The baud rate in mode 3 is variable.
Q9: Explain the interrupts of 8051 micro controller?
A9: External interrupt 0 (IE0) – Highest priority
Timers interrupt 0 (TF0)
External interrupt 1 (IE1)
Timers interrupt 1 (TF1)
Serial port Interrupt
Receive interrupt (RI) - lowest priority
Transmit interrupt (TI)
Q10: How many bytes of internal RAM and ROM supported by 8051 micro controller?
A10: 128 bytes of internal RAM and 4 bytes of ROM.

## Title of practical: Develop a 8051 assembly language program to AND/ OR 8 bit numbers

Q1 What are addressing modes of 8051 microcontroller?
A1 The MCS-51 instruction set offers several addressing modes, including
1.direct register, using ACC (the accumulator) and R0-R7
2.direct memory, which access the internal RAM or the SFR's, depending on the address
3.indirect memory, using R0, R1, or DPTR to hold the memory address. The instruction used may vary to access internal RAM, external RAM, or program memory.
4individual bits of a range of IRAM and some of the SFR's
Q2 Define ADD instruction for 8051 microcontroller?
A2 Description: ADD and ADDC both add the value operand to the value of the Accumulator, leaving the resulting value in the Accumulator. The value operand is not affected. ADD and ADDC function identically except that ADDC adds the value of operand as well as the value of the Carry flag whereas ADD does not add the Carry flag to the result.
The Carry bit (C) is set if there is a carry-out of bit 7. In other words, if the unsigned summed value of the Accumulator, operand and (in the case of ADDC) the Carry flag exceeds 255 Carry is set. Otherwise, the Carry bit is cleared.
The Auxillary Carry (AC) bit is set if there is a carry-out of bit 3. In other words, if the unsigned summed value of the low nibble of the Accumulator, operand and (in the case of ADDC) the Carry flag exceeds 15 the Auxillary Carry flag is set. Otherwise, the Auxillary Carry flag is cleared.
The Overflow (OV) bit is set if there is a carry-out of bit 6 or out of bit 7, but not both. In other words, if the addition of the Accumulator, operand and (in the case of ADDC) the Carry flag treated as signed values results in a value that is out of the range of a signed byte (-128 through +127) the Overflow flag is set. Otherwise, the Overflow flag is cleared.
Q3 Define AJMP instruction in 8051 microcontroller?
A3AJMP unconditionally jumps to the indicated code address. The new value for the Program Counter is calculated by replacing the least-significant-byte of the Program Counter with the second byte of the AJMP instruction, and replacing bits 0-2 of the most-significant-byte of the Program Counter with 3 bits that indicate the page of the byte following the AJMP instruction. Bits 3-7 of the most-significant-byte of the Program Counter remain unchaged. Since only 11 bits of the Program Counter are affected by AJMP, jumps may only be made to code located within the same 2k block as the first byte that follows AJMP.
Q4 Define ANL instruction in 8051 microcontroller?
A4 ANL does a bitwise "AND" operation between operand1 and operand2, leaving the resulting value in operand1. The value of operand2 is not affected. A logical "AND" compares the bits of each operand and sets the corresponding bit in the resulting byte only if the bit was set in both of the original operands, otherwise the resulting bit is cleared.
Q5 Define CLR instruction?
A5 CLR clears (sets to 0) all the bit(s) of the indicated register. If the register is a bit (including the carry bit), only the specified bit is affected. Clearing the Accumulator sets the Accumulator's value to 0.
Q6 Define CPL instruction?
A6 CPL complements operand, leaving the result in operand. If operand is a single bit then the state of the bit will be reversed. If operand is the Accumulator then all the bits in the Accumulator will be reversed. This can be thought of as "Accumulator Logical Exclusive OR 255" or as "255-Accumulator." If the operand refers to a bit of an output Port, the value that will be complemented is based on the last value written to that bit, not the last value read from it.
Q7 Define DA instruction?
A7DA adjusts the contents of the Accumulator to correspond to a BCD (Binary Coded Decimal) number after two BCD numbers have been added by the ADD or ADDC instruction. If the carry bit is set or if the value of bits 0-3 exceed 9, 0x06 is added to the accumulator. If the carry bit was set

when the instruction began, or if 0x06 was added to the accumulator in the first step, 0x60 is added to the accumulator.

The Carry bit (C) is set if the resulting value is greater than 0x99, otherwise it is cleared.

Q8 Define CJNE?

A8CJNE compares the value of operand1 and operand2 and branches to the indicated relative address if operand1 and operand2 are not equal. If the two operands are equal program flow continues with the instruction following the CJNE instruction.

The Carry bit (C) is set if operand1 is less than operand2, otherwise it is cleared.

Q9Define JB instruction?

A9 JB branches to the address indicated by reladdr if the bit indicated by bit addr is set. If the bit is not set program execution continues with the instruction following the JB instruction.

Q10 Define JC instruction?

A10JC will branch to the address indicated by reladdr if the Carry Bit is set. If the Carry Bit is not set program execution continues with the instruction following the JC instruction.

## Title of practical: Develop a program on trainer kit for inverse ANDing and EXORing of 8 bit numbers

Q1 What are different addressing modes?
A1Eight modes of addressing are available with the C8051
The different addressing modes determine how the operand byte is selected

| Addressing Modes | Instruction |
|---|---|
| Register | MOV A, B |
| Direct | MOV 30H, A |
| Indirect | ADD A,@R0 |
| Immediate Constant | ADD A,#80H |
| Relative* | SJMP +127/-128 of PC |
| Absolute* | AJMP within 2K |
| Long* | LJMP FAR |
| Indexed | MOVC A,@A+PC |

Q2 Define register addressing mode?
A2The register addressing instruction involves information transfer between registers
Example:
MOV R0, A
The instruction transfers the accumulator content into the R0 register. The register bank (Bank 0, 1, 2 or 3) must be specified prior to this instruction. In the Register Addressing mode, the instruction involves transfer of information between registers. The accumulator is referred to as the A register.
Q3 Define direct addressing?
A3 This mode allows you to specify the operand by giving its actual memory address (typically specified in hexadecimal format) or by giving its abbreviated name (e.g. P3) Used for SFR accesses
Example:
MOV A, P3 ;Transfer the contents of ;Port 3 to the accumulator
MOV A, 020H ;Transfer the contents of RAM ;location 20H to the accumulator.
Q4Define indirect addressing mode?
A4 This mode uses a pointer to hold the effective address of the operand  Only registers R0, R1 and DPTR can be used as the pointer registers The R0 and R1 registers can hold an 8-bit address, whereas DPTR can hold a 16-bit address Used for the upper data memory area
Examples: 1.MOV @R0,A ;Store the content of ;accumulator into the memory ;location pointed to by the contents ;of register R0. R0 could have an ;8-bit address, such as 60H.
2.MOVX A,@DPTR ;Transfer the contents from ;the memory location ;pointed to by DPTR into the ;accumulator. DPTR could have a ;16-bit address, such as1234H.
Q5Define immediate constrant addressing mode?
A5  This mode of addressing uses either an 8- or 16-bit constant value as the source operand This constant is specified in the instruction, rather than in a register or a memory location The destination register should hold the same data size which is specified by the source operand Examples:ADD A,#030H ;Add 8-bit value of 30H to ;the accumulator register ;(which is an 8-bit register).MOV DPTR,#0FE00H ;Move 16-bit data constant ;FE00H into the 16-bit Data ;Pointer Register.
Q6Define ACALL Instruction?
A6 ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the stack pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, op code bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2K block of program memory as the first byte of the instruction following ACALL. No flags are affected.

Operation: ACALL
(PC) ¬ (PC) + 2
(SP) ¬ (SP) + 1
((SP)) ¬ (PC7-0)
(SP) ¬ (SP) + 1
((SP)) ¬ (PC15-8)
(PC10-0) ¬ page address
Q7 What are different instruction types?
A7The C8051 instructions are divided into five functional groups:
1 Arithmetic operations
2 Logical operations
3Data transfer operations
4 Boolean variable operations
5 Program branching operations
Q8Which flags are affected by arithmetic instruction?
A8 Arithmetic operations effect the
flags, such as Carry Flag (CY), Overflow Flag (OV) etc, in the PSW register.
Q9 What are logical operation?
A9 Logical instructions perform Boolean operations (AND,
OR, XOR, and NOT) on data bytes on a bit-by-bit basis.
Q10 What are data traqnsfer instruction?
A10Data transfer instructions can be used to transfer data between an internal RAM location and an SFR location without going through the accumulator  It is also possible to transfer data between the internal and external RAM by using indirect addressing  The upper 128 bytes of data RAM
are accessed only by indirect addressing and the SFRs are accessed only by direct addressing.

## Title of practical: Develop a assembly language program for finding 1's and 2's complement of 8 bit numbers.

Q1: Define machine cycle of 8051?
A1: 8051 machine cycle consists of 6 states, S1 through S7. One state is made up of 2 clock pulses. Thus 12 clock periods constitute one machine cycle. Two clock periods in a state is termed as phase 1 and phase 2.
Q2: What are the special functions of port 0 of 8051?
A2: Port 0 is used as a multiplexed low order address/data bus during the external memory access. When ALE is enabled, the address on port 0 pins are latched and bus is ready to act as a data bus when ALE is low.
Q3: What are the alternative functions of port 3 of 8051?
A3: Serial data input (P3.0), serial data output (P3.1), external interrupt 0 (P3.2), external interrupt 1 (P3.3), external input for timer 0(P3.4), external input for timer 1 (P3.5), external memory write pulse (P3.6), external memory read (P3.7) are the alternative functions of port 3.
Q4: What are the uses of scratch pad area of internal RAM of 8051?
A4: In internal RAM 80 bytes constitutes the scratch pad area. The scratch pad bytes can be programmed as a general purpose registers.
Q5: What are the flags supported by 8051 controller?
A5: The flags supported by 8051 controller are:-
Carry flag
Auxiliary carry flag
Over flow flag
General purpose user flag
Register bank select bit one
Register bank select bit zero
Parity flag
Q6: What are the different groups of instructions supported by 8051?
A6: The different groups of instructions supported by 8051 are:-
Data Transfer Group
Arithmetic Group
Logical Group
Branching Group
Bit manipulation Group
Q7: Write about CALL statement in 8051?
A7: There are two CALL instructions. They are
LCALL (Long call)
ACALL (Absolute call)
Q8: Write about the jump statement?
A8: There are three forms of jump. They are
__LJMP (Long jump) – 16 bit address
__AJMP (Absolute jump) – 11 bit address
__SJMP (Short jump) – relative address
Q9: Write a program to find the 2's complement using 8051?
A9: MOV A, R0
CPL A
INC A
Q9: Write a program to swap two numbers using 8051?
A9: MOV A, # data
SWAP A
Q10: What is the significance of TXD and RXD pins in 8051?
A10: TXD – Transmit data pin for serial port in UART mode. Clock output in shift register mode.
RXD – Receive data pin for serial port in UART mode. Data I/O pin in shift register mode.