MICROPROCESSOR INTERFACING AND ASSEMBLY LANGUAGE

Mid Term Answer Sheet

Muhammad Muneeb Qureshi
BSCS/1-18/M01021

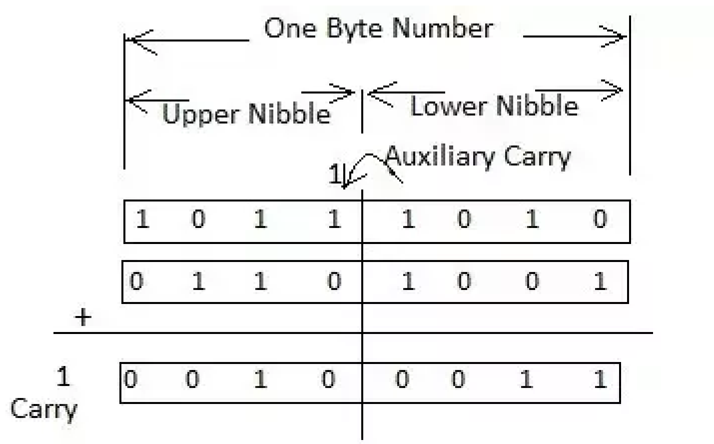
Answer# 1(a)

Displays generally show the status of arithmetic and logical operations. The parity indicator is like an even parity bit of the operation result. The parity flag is set to 1 when the result has an even number of ones and is reset when the result has an odd number of ones

AUXILIARY CARRY FLAG - Set when a carry is created from a lower bite.

This flag is set to 1 by the command that has just ended if bit 3 is transferred to bit 4 of register A while the command is being executed. Due to the relationship between the decimal point with pure BCD and the hexadecimal coding, it is possible to bring BCD values directly into the A register and to carry out mathematical operations with them. However, the result is like processing two hexadecimal characters. If the result is to be returned to the program as a BCD instead of hexadecimal, the Decimal Adjustment Accumulator (DAA) statement can perform this translation; the aid transport pavilion is provided to facilitate this process.

1. It occupies the fourth bit of the flag register. In an arithmetic operation, when the transmission is generated by the third bit and the fourth bit is passed, the auxiliary transmission flag is set (1). Otherwise, the flag is reset (0). This flag is only used for Binary Coded Decimal (BCD) operations.



Example of a binary sum in which an auxiliary carry is generated.

2. So, the auxiliary flag is set when there is a transfer from bit d3 of the data to bit d4 of the data ... that is, for example. If you add

29 —————- 0010 1001 (41) feet 10

39 —————- 0011 1001 (57) base 10

~~~~~~~~~~~~ 0110 0010 (98) base 10
~~~~~~~~~~~~

The help flag is set here

PARITY FLAG – parity flag is set if the parity is set to odd or even.

The parity is determined by the number of ones in the accumulator. If the result after an arithmetic / logic operation has an even number of ones, it is considered even parity. If the parity is odd, the flag is reset.

Example:

1. The PARITY display is 1 if several ones in the accumulator are equal after a certain process. Otherwise, it is 0 if any number of ones in the accumulator are odd.
2. Parity refers to the presence of the number one in the given 8-bit data. The parity flag is set or raised if the number of presences of one in the data is even (2,4,6,8) and if the number of presences of one in the data is odd (1,3.5, 7) then that will be Flag reset or lowered.

2nd bit of the flag register.

This flag tests a 1-bit number in the accumulator. If the battery has an even number of ones, the parity is even and the flag bit is set (1). On the other hand, if the number of ones is odd, the parity is odd and the flag is reset to zero (0).

Conclusion:

In the Intel 8085 microprocessor, the flag register is part of the program / processor status word. The PSW contains 8 bits, 5 of which are flag bits (the flip-flops store the execution information in the accumulator), which represent the execution status of the microprocessor. (Some books mention that the PSW is a 16-bit register that contains the flag register (8 bits) and the accumulator (8 bits). But here are just the answers to the auxiliary and parity flags after PSW 8-bit).

1. Auxiliary flag: This is a D4 bit of the flag register. This flag is always set when the lower nibble is executed on the upper nibble (bit 3 to bit 4) or borrows 8 bits from the upper nibble to the lower nibble (bit 4 to bit 3), otherwise it is started reset. This flag is used by decimal arithmetic instructions.

1 - The report is generated between bit 3 and bit 4

0 - other

2. Parity flag: This is bit D2 of the Tag Register. This flag is set when the result has even parity, an even 1-bit number. If the parity is odd, then PF is cleared.

Answer# 01(b)

Difference Between microprocessors and microcontrollers:

- **Microprocessor**

In simple terms, the microprocessor is useful in very intensive processes. It only contains a processor (central processing unit), but many other parts are required to work with the processor to complete a process. All of these other parts are connected to the outside.

Microprocessors are not designed for a specific task and are useful when the tasks are complex and misleading, such as: B. in software development, in games and other memory-intensive applications where inputs and outputs are not defined.

Some examples from everyday life are:

a) Home appliances: complex home security, personal computers, video game systems and much more.
b) Transportation and industrial equipment: cars, trains, planes, computer servers, high-tech medical equipment, etc.

- **Microcontroller**

The microcontroller is designed for a specific task or to repeatedly execute the assigned task. Once the program is built into a microcontroller chip, it cannot be easily changed and you may need special tools to re-register it. Depending on the application, the process is defined in the microcontroller. Hence the output depends on the input from the user or the sensors or predefined inputs.

Some examples from everyday life are:

a) Calculator, washing machine, ATM, robotic arm, camera, microwave, oscilloscope, digital multimeter, EKG machine, printer, etc.

Microprocessor	Microcontroller
The microprocessor is the heart of the computer system.	The microcontroller is the heart of an onboard system.
It's just a processor, so memory and I / O components need to be connected externally.	The microcontroller has a processor as well as internal memory and I / O components.
The memory and I / O must be connected externally, which makes the circuit large.	Memory and I / O are already in place, and the internal circuitry is small.
It cannot be used on compact systems.	It can be use it in compact systems.
The cost of the overall system is high.	The cost of the overall system is low.
Due to the external components, the total power consumption is high. Therefore, it is not ideal for devices that use stored energy, such as batteries.	Since the external components are weak, the overall power consumption is lower. Therefore, it can be used with devices that run on stored energy, such as: Batteries.
Most microprocessors do not have energy saving functions.	Most microcontrollers offer an energy-saving mode.
It is mainly used in personal computers.	It is mainly used in washing machines, MP3 players, and car systems.

The microprocessor has fewer registers, so there are more memory-based operations.	The microcontroller has more registers. Therefore, the programs are easier to write.
The microprocessors are based on the Von Neumann model.	The microcontrollers are based on the Harvard architecture.
It is a central processing unit on a single integrated silicon-based chip.	It is a by-product of the development of CPU microprocessors and other peripheral devices.
It lacks RAM, ROMs, I / O drives, timers, and other on-chip devices.	It has a processor and RAM, ROM and other peripherals integrated into a single chip.
It uses an external bus to connect to RAM, ROM, and other peripheral devices.	It uses an internal control bus.
Microprocessor-based systems can work at a very high speed thanks to the technology used	Microcontroller-based systems work with up to 200 MHz or more, depending on the architecture.
It is used for general purposes.	Applications that you can use to manage large amounts of data. It is used for certain application systems.
It is complex and expensive, with a large number of instructions to process.	It's easy and inexpensive to process with fewer instructions.

Answer# 02(a)

Although volatile memory and non-volatile memory are related and come from the same source, there are many differences between these terminologies that can be confusing to many people new to information technology. In addition, the architectures used turn out to be different. Hence, both must consider important differences that are just as important. In this blog I am going to describe the main differences between them.

In computing, memory refers to devices that are used to store various types of data and information. The term primary storage is used for high speed (i.e., RAM) use with storage systems other than secondary storage. In general, the secondary storage has a slower access schedule and slower data storage, but contains more storage capacity. In addition, main memory can be stored in secondary memory, if necessary, using an in-memory storage technique called "virtual memory".

1. **Volatile memory**
   Volatile memories are computer memories that require electricity to maintain stored information. The most modern volatile semiconductor memory is static RAM (Random Access Memory) (i.e. SRAM) or dynamic RAM (i.e. DRAM).
   When switched on, the SRAM retains its content and can be easily plugged in. Compared to SRAM, Dynamic RAM (DRAM) is more complicated to communicate and control and requires regular update cycles to avoid loss of content.
   However, DRAM only uses one transistor and one capacitor per bit, which allows much higher densities to be achieved and is much cheaper with more bits on a memory chip per bit.
   SRAM is popular in small embedded systems that require only ten kilobytes or less. Z-RAM, TTRAM, A-RAM and ETA RAM contain new dynamic storage systems that are intended to replace or compete with SRAM and DRAM.
   It is the storage hardware that receives / stores data at high speed. It is also known as temporary storage. Data in volatile memory is stored until the system is able to do so, but after the system is turned off, the data in volatile memory is automatically erased. RAM (Random Access Memory) and cache are common examples of volatile memory. Here, data collection / storage is quick and cheap.

2. **Non-volatile memory**
   Non-volatile memory is computer memory that retains the information stored even when it is not turned on or when there is no power.
   Examples of non-volatile memory include ROM (Read-Only Memory), flash memory, most types of magnetic computer storage devices (such as hard drives, floppy disks, and magnetic tapes), optical discs, and early computer storage methods such as paper tape and punch cards.
   Coming non-volatile storage technologies include FeRAM, CBRAM, PRAM, SONOS, RRAM, NRAM, Racetrack, and Millipede.
   It is not cheap or slow to retrieve / store when compared to volatile memory, but it does store a larger amount of data. All information that has to be retained over a longer period of time is stored in a non-volatile memory. Non-volatile memory has a major impact on the storage capacity of a system.

In short, volatile and non-volatile memory is here simply that volatile memory requires electricity or electricity to store data. In contrast, non-volatile memory does not need any of the requirements of volatile memory.

Answer# 02(b)

MN / MX *:

This pin indicates which mode the processor is working in. In the minimum mode, the 8086 generates all bus control signals. In the maximum mode, all three status signals must be decoded in order to generate all bus control signals.

It is available at pin 33. It indicates in which mode the processor is operated; when it is high, it works in minimum mode and vice versa.

MN / MX is an input pin used to select one of these models. When MN / MX is high, the 8086 is operating in minimum mode. In this mode, the 8086 is configured to support a small, single processor system that uses peripheral devices other than the system bus. When MN / MX is low, the 8086 is configured to support a multiprocessor system.

<u>Features of the 8086 microprocessors:</u>

a) It contains a 20-bit address bus.
b) It contains a 16-bit data bus, which is why the 8086 is referred to as a 16-bit microprocessor.
c) This is a 2-stage channel processor. You can pull 6 bytes from memory and put them in a queue for faster execution.
d) The control bus carries signals for performing operations such as reading, writing, etc.
e) It has memory banks. with 2 banks of 512 KB each. These banks are called lower (even) banks and higher (odd) banks.
f) In 8086 all memory is divided into four memory segments: code, stack, data and additional segment.
g) 8086 has a 16-bit IO address.
h) It has 256 pauses.
i) 8086 has two modes of operation:

- Minimum mode
- Maximum mode

Minimum mode:

This 8086 is the only processor in the system. In an 8086 system in minimal mode.

8086 works in minimum mode when the MN / MX spindle is in logical 1.

In this mode, all control signals are output by the 8086 itself.

Maximum mode:

Here we can connect additional processors to the 8086 (8087/8089).

The 8086-Max mode is basically used to implement global resource allocation and to pass control of the bus to other coprocessors (i.e. the second processor in the system) since two processors cannot access the system bus at the same time.

All processors execute their own program.

The resources common to all processors are called global resources.

The resources allocated to a particular processor are called local or private resources.

Advantages of the 8086-maximum mode:

Helps you connect additional devices such as the 8087. This interface is also known as a tightly coupled coprocessor configuration. In this 8086 it is called the host and 8087 as the coprocessor.

It supports multiprocessing and thus contributes to increasing efficiency.

The 8087 was the first floating point coprocessor for the 8086 series of microprocessors. The purpose of the 8087 was to improve the calculations for floating point operations like addition, subdivision, multiplication, division, and square root.

Disadvantages of the maximum mode compared to the minimum mode:

It has a more complex circuit than the minus mode.

Application of 8086:

- Microcomputers are built with 8086. For example, the IBM PC used the Intel 8088, a version of the 8086 with the 8-bit data bus.
- It is used in calculators.
- It is used for control purposes such as signage (it uses microcontrollers that are nothing more than one or more processor, as well as memories and programmable I / O devices).

Minimum mode 8086	Maximum mode 8086
There can be only one processor.	There can be multiple processors
ALE - the address lock for the lock is activated by 8086	ALE for the lock is given by the 8288 bus controller
The signals HOLD and HLDA are used for the bus request.	The signals RQ, GT, RQ–, GT– etc. are used for the bus request.
The circuit is simpler.	The circuit is more complex
Multiprocessing not possible	Multiprocessing is possible
The performance is slower.	High efficiency.

Answer# 03(a)

CPU Architecture of 8086

Microprocessors act as CPUs in the stored program model of the digital computer. Its job is to generate all synchronization signals in the system and to synchronize the data transfer between memory, I / O and itself. It fulfills this task through the architecture of the three-bus system.

A microprocessor is an integrated circuit with all the functions of a CPU; However, it cannot be used independently because, unlike a microcontroller, it has no memory or peripherals.

The microprocessor also has a S / W function and must recognize, decode and execute program instructions fetched from the memory unit. This requires an arithmetic and logic unit (ALU) in the CPU to perform arithmetic and logic functions (AND, OR, NOT, compare, etc.).

The 8086 processor is organized into two separate processors called the Bus Interface Unit (BIU) and the Execution Unit (UE). The BIU provides H / W functions including memory generation and I / O addresses for data transfer between the outside world, i.e., H. outside the CPU, and the UE.

8086 has no RAM or ROM inside. However, it has internal registers for storing intermediate and final results and is connected to the external memory via the system bus.

In the case of the 8086, this is a 16-bit integer processor on an integrated 40-pin dual in-line circuit.

The size of the internal registers (available in the chip) indicates how much information the processor can use at the same time (in this case 16-bit registers) and how this data moves internally within the chip. internal data bus.

8086 provides the programmer with 14 internal registers, each 16 bits or 2 bytes wide.

Memory segmentation:

- To increase execution speed and recovery speed, 8086 memory segments.
- The 20-bit address bus can address 1 MB memory and divide it into 16 64 KB segments.
- 8086 works with only four 64K segments in full 1MB memory.
- The internal architecture of Intel 8086 is divided into 2 units: the Bus Interface Unit (BIU) and the Execution Unit (UE). These are explained below.

1. The Bus Interface Unit (BIU):

Provides the 8086 interfaces for external memory and I / O devices via the system bus. Run different machine cycles, e.g., B. Read memory, read I / O, etc. to transfer data between memory and I / O devices.

BIU performs the following functions:

- Generates the physical 20-bit address for memory access.
- Get directions from memory.
- Transfer data to and from memory and I / O.
- Maintains a 6-byte prefetch command queue (supports pipeline).
- BIU mainly contains the 4 segment registers, the instruction pointer, a prefetch queue and an address generation circuit.

Instruction Pointer (IP):

- It's a 16-bit register. Causes subsequent instructions to scroll further in the code segment.
- The IP address is incremented after receiving each command byte.
- IP is given a new value every time a branch instruction occurs.
- CS is multiplied by 10H to obtain the 20-bit physical address of the code segment.
- The address of the next command is calculated as CS x 10H + IP.

Example:

CS = 4321H IP = 1000H

then CS x 10H = 43210H + displacement = 44210H

This is the meaning of teaching.

- Code segment register:
  CS contains the base address of the code segment. All programs are stored in the code segment and are accessible via IP.

- Register of data segments:
  DS has the base address for the data segment.

- Stack segment register:
  SS contains the base address of the stack segment.

- Extra segment register:
  ES contains the base address of the additional segment.

- Address Generation Circuit:
  - The BIU has a circuit for generating physical addresses.
  - Generate the 20-bit physical address using segment and offset addresses using the formula:

    physical direction

    = Segment direction x 10H + compensation direction

6-byte prefetch queue:

- This is a 6-byte (FIFO) queue.
  - The fetching of the next instruction (by CS BIU) while the current instruction is executing is called a pipeline.
  - It is flushed when a branch instruction occurs.

2. The Execution Unit (EU):

The main components of the UE are general purpose registers, ALU, special registers, command registers and command decoders and flag / status registers.

1. Retrieves, decodes, and performs arithmetic and logical operations using the ALU from the queue in BIU.
2. Sends control signals for internal data transfers within the microprocessor.
3. Sends request signals to the BIU to access the remote module.
4. It works in terms of T states (clock cycles) and not machine cycles.

8086 has four 16-bit general purpose registers AX, BX, CX and DX. Save intermediate values during runtime. Each of them has two 8-bit parts (upper and lower).

- AX register:
  Contains operands and results for multiplication and division. Also, an accumulator during chain operation.
- BX recording:
  Retains the memory address (offset address) in indirect addressing modes.
- CX entry:
  Keep track of the number of instructions, e.g., B. Loop, rotate, move, and string operations.
- DX recording:
  Used with AX to hold 32-bit values during multiplication and division.

1. Arithmetic Logical Unit (16 bit):

Performs 8- and 16-bit arithmetic and logic operations.

Special purpose register (16 bit):

- Stack pointer:
  Aim for the top of the pile. The stack is located in the stack segment and is used for commands such as PUSH, POP, CALL, RET, etc.
- Base pointer:
  BP can contain offset addresses from any position in the stack segment. Used to access random locations on the stack.
- Source Index:
  Maintains the offset direction in the data segment for string operations.
- Target index:
  Maintains the direction of travel in the extra segment for chain operation.

2. Instruction registers and Instruction decoder:

The EU receives an opcode from the queue in the command register. The instruction decoder decodes them and sends the information to the control circuit for execution.

Flag / status register (16 bit):

It has 9 flags that help change or recognize the status of the microprocessor.

<u>6 national flags:</u>

1. carry flag (CF)
2. parity flag (PF)
3. auxiliary carry Flag (AF)
4. zero flag (Z)
5. sign flag (S)
6. overflow flag (O)

Status displays are updated after every arithmetic and logical link.

<u>3 control flags:</u>

1. trap flag (TF)
2. interruption Notification (IF)
3. direction Flag (DF)

These flags can be set or reset by control commands such as CLC, STC, CLD, STD, CLI, STI, etc.

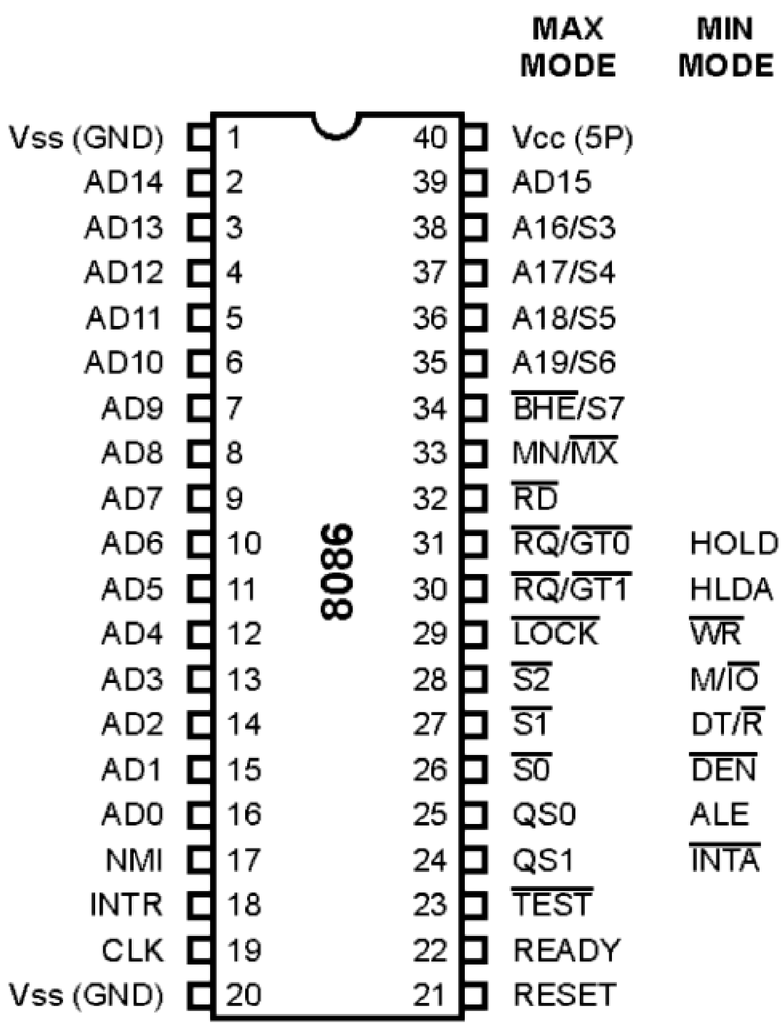
Control flags are used to control certain operations.

<u>Answer# 03(b)</u>

BHE (Bus High Enable)

BHE / S7 - Pin number 34 - BHE stands for Bus High Enable. The combination of signal BHE and status S7 provides information about the presence of data on the bus. The various combinations also indicate whether the bus contains a 16-bit general byte, a high byte or a low data byte.

BHE / S7 34 - STATUS ENABLE / BUS HIGH: During T1, the Bus High Enable (BHE) signal must be used to enable data on the main half of the data bus, pins D15 ± D8. Eight bit oriented devices connected to the top half of the bus would normally use BHE to condition the chip select functions.



BHE is low during T1 for the read, write and interrupt acknowledge cycles when a byte is to be transferred to the high side of the bus. S7 status information is available during T2, T3 and T4. The signal is active LOW and is in `` hold '' in 3 OFF states. It is LOW during T1 for the first interrupt acknowledge cycle.

BHE is used to enable the high order bus to distinguish between a word operation and a byte operation.

It is available on pin 34 and is used to indicate data transmission over the D8-D15 data bus. This signal is low for the first clock cycle, then it is active.