Reg. No: (BSCS/3-17/M01006)

Student Name: SHAH FARAZ

Course: Microprocessor Interfacing & Assembly Language

Faculty: Miss Sania Urooj

Submission Date: 09-July-2021

MID TERM EXAM

Q. No. 1 (a): Explain the difference between assembly and machine language.

Answer:

Machine code is a computer program written in machine language instructions that can be executed directly by a computer's central processing unit (CPU). Conversely, assembly language is a low-level programming language in which there is a strong correspondence between the program's statements and the architecture's machine code instructions. Hence, this is the fundamental difference between machine code and assembly language.

Syntax

Machine Language consists of binaries, which are zeros and ones. Assembly language, on the other hand, follows a syntax similar to the English Language. Therefore, this is a major difference between machine code and assembly language.

Comprehensibility

Only the CPU understands the machine code; however, the programmer understands the assembly language.

Dependency

Another difference between machine language and assembly language is that the machine code depends on the platform or the operating system. But assembly language consists of a set of standard instructions.

Usage

Considering the usage, the CPU can directly execute the machine language to perform the defined tasks in the computer program. On the other hand, real-time systems, and microcontroller-based embedded systems are some examples of applications using assembly language.

Conclusion

In brief, assembly language is one level ahead of machine language. The main difference between machine language and assembly language is that the machine code is a language consisting of binaries that can be directly executed by a computer while an assembly language is a low-level programming language that requires a software called an assembler to convert it into machine code.

Q. No. 1 (b): What are the major differences between microprocessors and microcontrollers?

Answer:

Differences between Microprocessors and Microcontrollers

- ## Microprocessors:

The **microprocessor** is used in the very intensive processes. It only contains a CPU (central processing unit) but there are many other parts needed to work with the CPU to complete a process. These all other parts are connected externally. The microprocessor chip is not containing all these parts internally. The number of external parts and the size of the external parts depends on the application. Generally, it connected with memory elements like RAM and ROM, I/O ports, timers, serial interface, etc. The advantage of the microprocessor is that it has a flexible structure. It means you can decide the size of RAM, ROM, number of I/O ports and can modify all the things which are connected externally according to the application.

- ## Microcontrollers:

Microcontrollers are used to do the same assigned task repeatedly. Hence, the number of I/O ports and the amount of memory required is less compared to the microprocessor. As told earlier, in microcontroller external parts are integrated with CPU in a single chip and because of this integrated structure the overall size of the microcontroller is smaller compared to the microprocessor. In

microcontroller you cannot modify the size of RAM, ROM and other components. Once a controller is designed the structure is fixed. So, the structure of the microcontroller is not flexible.

Q. No. 2 (a): Provide the name of pins that are used as control signals and also write down their functions?

Answer:

These signals are used to identify the nature of operation. Three control signals are RD, WR & ALE.

RD − This signal indicates that the selected IO or memory device is to be read and is ready for accepting data available on the data bus.

WR − This signal indicates that the data on the data bus is to be written into a selected memory or IO location.

ALE − It is a positive going pulse generated when a new operation is started by the microprocessor. When the pulse goes high, it indicates address. When the pulse goes down it indicates data.

Q. No. 2 (b): What is the purpose of MN/Mx pin? Explain
Define the minimum and maximum mode of 8086 microprocessor.

Answer:

- **MN/Mx pin**

This pin indicates what mode the processor to operate in. In minimum mode, the 8086 itself generates all bus control signals. In maximum mode the three status signals are to be decoded to generate all the bus control signals.

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 small single processor system using a few devices that the system bus. When MN/MX is low 8086 is configured to support multiprocessor system.

- **Minimum and Maximum mode of 8086 Microprocessor**

Minimum mode:

- The 8086 microprocessor operates in minimum mode when MN/MX' = 1.
- In minimum mode,8086 is the only processor in the system which provides all the control signals which are needed for memory operations and I/O interfacing.
- Here the circuit is simple but it does not support multiprocessing.
- The other components which are transreceivers, latches, 8284 clock generator, 74138 decoder, memory and i/o devices are also present in the system.
- The address bus of 8086 is 20 bits long. By this we can access 220 byte memory i.e. 1MB. Out of 20 bits, 16 bits A0 to A15(or 16 lines) are multiplexed with a data bus. By multiplexing, it means they will act as address lines during the first T state of the machine cycle and in the rest, they act as data lines. A16 to A19 are multiplexed S3 to S6 and BHE' is multiplexed with S7.

Maximum mode:

- In this we can connect more processors to 8086 (8087/8089).
- 8086 max mode is basically for implementation of allocation of global resources and passing bus control to other coprocessor (i.e. second processor

in the system), because two processors can not access system bus at same instant.

- All processors execute their own program.
- The resources which are common to all processors are known as global resources.
- The resources which are allocated to a particular processor are known as local or private resources.

Q. No. 3 (a): Define the purpose of BIU and Execution Unit.

Answer:

- BIU (Bus Interface Unit):

To increase the processing speed of the processor 8086 microprocessor has been divided into EU and BIU functional units. As we have mentioned above that the EU codes and decodes the data, BIU interacts with the memory, input and output devices in order to fetch the instructions and data thereby transfer of all data and addresses on the buses for the EU execution. This unit helps in sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory. EU does not have any direct connection with System Buses so this is possible with the BIU. Fetching and execution works simultaneously to increase the performance of the microprocessor.

- EU (Execution Unit):

Execution unit is a functional unit consists of CPU which helps in performing operations and arithmetical calculations instructed by the computer program. In machine language we could say that it decodes and executes instructions. Apart from other internal units it has its own control unit which helps to perform functions along with other basic internal units. The data which is required by the EU is fetched from the BIU thus maintaining interaction with the memory and input and output devices.

Q. No. 3 (b): What's the purpose of BHE pin of 8086 Microprocessor?

Answer:

BHE stands for Bus High Enable. It is available at pin 34 and used to indicate the transfer of data using data bus D8-D15. This signal is low during the first clock cycle, thereafter it is active.

The bus high enable signal is used to indicate the transfer of data over the higher order (D15-D8) data bus. It goes low for the data transfers over D15-D8 and is used to derive chip selects of odd address memory bank or peripherals. BHE is low during T1 for read, write and interrupt acknowledge cycles, when- ever a byte is to be transferred on the higher byte of the data bus.