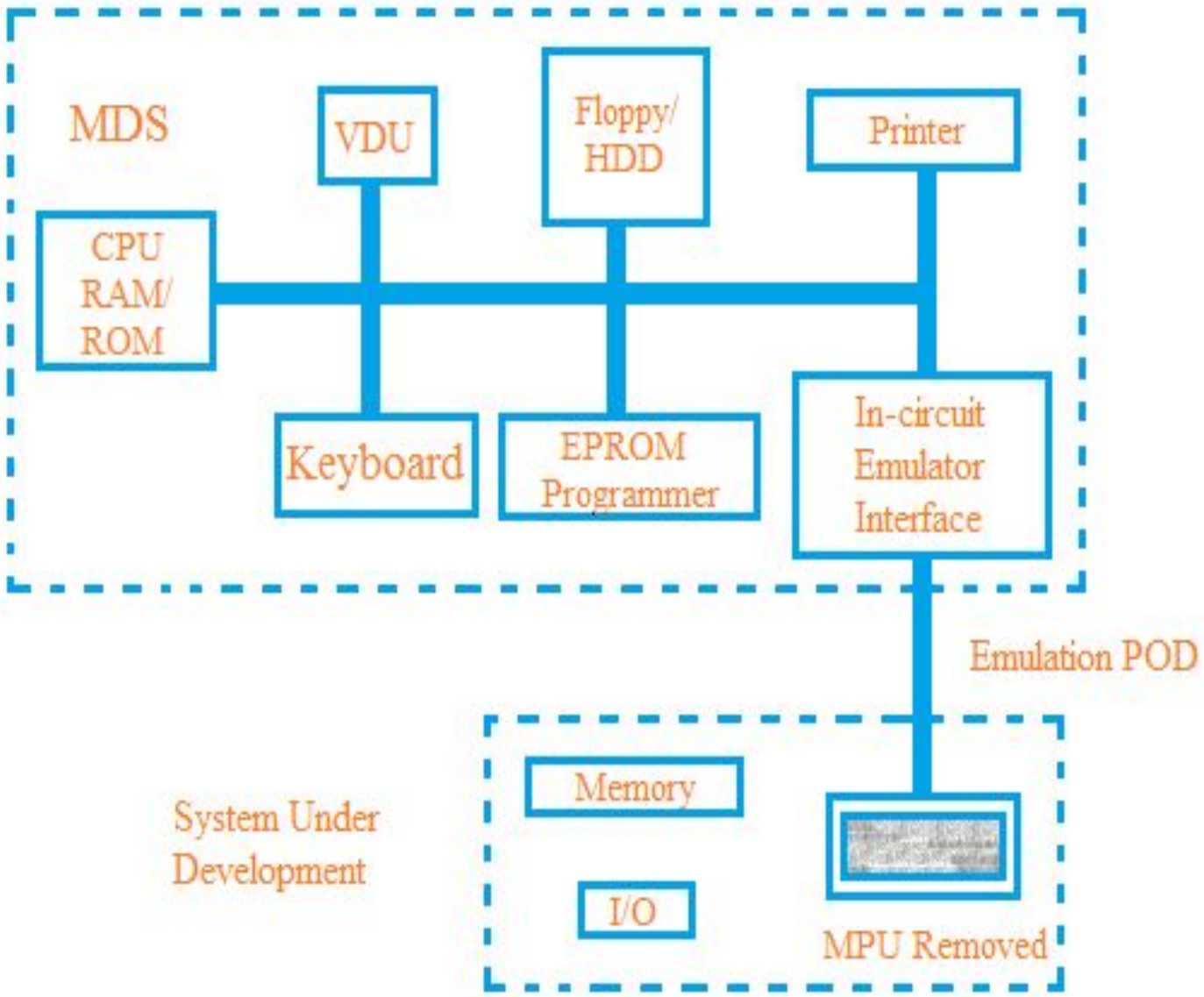
Microprocessor Development System

- MDS Hardware System
- MDS Software System
- Typical Development Steps

Microprocessor development system

- MDS is a microcomputer.
- Combination of special purpose software and hardware, which are provide to aid the design and development of MPU based system.
- Allow user to develop s/w for any given application based on the particular MPU.
- Create an environment to test and debug the target H/W.
- Facilitate integration of the S/W and H/W.
- The System under development is often referred to as target system.

Block Diagram Of MDS System



MDS Hardware System

- MDS H/W system consist of following subsystem –

i. CPU
ii. Console Unit
iii. Main Memory
iv. Secondary and Backup Memory
v. Printer
vi. EPROM Programmer
vii. In- Circuit Emulator

MDS Hardware System(contd.)

- **CPU:**

I. Central Processing Unit.
II. Heart of the System.
III. Responsible for the overall control of the devices.

MDS Hardware System(contd.)

- **Console Unit:**

I. Consisting of a visual display Unit (VDU) and a keyboard.
II. VDU and keyboard provides an interface between the user and MDS.
III. User enters data through the keyboard and it displayed by the VDU.

MDS Hardware System (contd.)

- **Main Memory:**

I. Divided into three functionally different parts.
II. ROM stores the loader/operating system.
II. RAM store system s/w and other temporary program loaded.
V. Emulation memory consist of fast RAM used for storing the user's program of the target system.
V. Size of the three different memories varies from each other.

MDS Hardware System (contd.)

- **Secondary and Backup Memory:**

I. Winchester disk
II. Floppy disks systems
II. Solid State disk

MDS Hardware System

- **Printer:**

I. Used in MDS to get hardcopy of user programs.
II. Allows development and documentation to go hand-in-hand
III. Dot matrix printers are currently used because of low cost

MDS Hardware System

- **EPROM programmer:**

I. allows direct loading of the software under development
II. made as an optional item with MDS
II. made as universal in configuration

MDS Hardware System

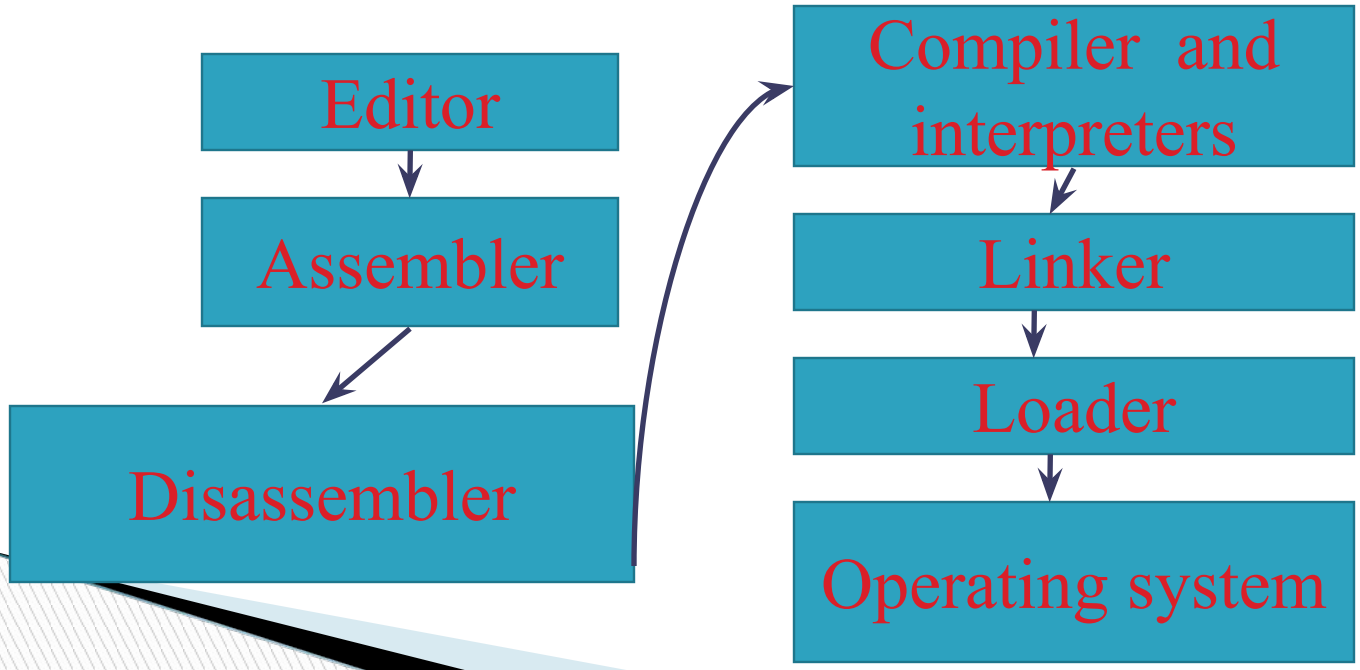
In-circuit emulator:

I. Consists of special purpose hardware along with necessary software.
II. Enables user to emulate close to target system allowing testing of the target software
II. Provides an interface between MDS and target system giving window .

MDS Software Systems.

MDS software system:

Along with hard ware subsystem, a number of system software are provided in MDS. Followings things are found in MDS software:



Editor:

- A program written by the user either assembly language or in a high level language is known as a source program. The source program can be entered into the MDS via the keyboard.
- The response of the MDS is displayed on the VDU.
- The editor allows the user to enter the source code and facilities the modification of the source code whenever necessary.
- A user comes in contact with the editor very frequently.

- Editors with different capabilities and features are available.
- The simple editor is known as line-oriented editors.
- Another type of editor is recently used known as character-oriented editors.

Assembler:

- The assembler performs the complete translation of a user program in assembly language into the machine language of the microprocessor of the target system.
- The translation process consists of two distinct steps : analysis & synthesis.
- In the analysis phase, the assembler scans through the source program and identifies the various fields; label,opcode and operands of the assembly language instructions.
- In the synthesis phase,machine language code corresponding to each assembly language instruction is generated.

Disassembler:

Another useful software aid is the disassembler, which can generate an assembly language program corresponding to a machine language program.

Compiler and interpreters:

- The translation from a high level language program statements to executable machine codes is performed by a system software known as compiler.
  - The program development time is considerably shortened.
  - High level language are not processor dependent , which makes them portable for use with any system.
  - High level language are also easier to write.

- There is alternative systems program known as the interpreter , which also translates a high-level language program into a machine language program.
  - ❖ Interpreters are usually less sophisticated than compilers.
  - ❖ Although interpreters provide much lesser facilities than compilers and execute a source program rather slowly , they are nevertheless very useful software aids.

Linker:

- The object files are generated in relocatable form and the linker converts the object file to executable absolute file , which consists of actual machine code at correct addresses.
- Therefore , linkers provide more flexibility , allow software modules to be developed and translated independently and integrate them by linking at a later stage.

Loaders:

- The linked object file is actually loaded into the main memory of the MDS by loaders of execution.

- Operating system:

  The operating system provides the file and memory management routines and other utility programs to maintain the user's program in secondary and main memory in an orderly manner.

Typical Development Steps

- **Step 1:**

  Obtain problem specification.

- **Step 2a:**

  Make a block diagram of hardware system satisfying the problem requirements.

Typical Development Steps

- **Step 2b:**

  Make a flowchart of the necessary software.

- **Step 3a:**

  Obtain detailed circuit implementation based on availability, cost, performance.

Typical Development Steps

- **Step 3b:**

  Translate the flowchart into an assembly or higher-level language code.

- **Step 4a:**

Enter the source code module into MDS using the editor.

Typical Development Steps

- **Step 4b:**

  fabricate the prototype hardware circuit.

- **Step 5:**

  Assemble or compile source code module to produce the corresponding machine language using assembler or compiler.

Typical Development Steps

- **Step 6:**

  Link the absolute machine language code modules using the linker and load them into the memory by using the loader.

Typical Development Steps

- **Step 7:**

  Test the absolute object codes modules by running them in the emulation mode on the MDS

- **Step 8:**

  If necessary Debug the object program module

Typical Development Steps

And-

go back to step 4(a) to edit the source code modules if mistakes.

Step 9:

Connect the prototype target system to MDS

And

Use the emulation mode to test software

Typical Development Steps

- **Step 10:**

  Debug the prototype hardware

  - using the trace analysis facility

  And rectify the hardware faults.

Typical Development Steps

- **Step 11:**

  Program EPROM in the PROM programming mode.

Step 12:

Disconnect the target system from the MDS

Plug-in the MPU and EPROM to get the complete working model and test its operation.

Various types of MDS

- Hewlett Packard's HP6400
- Tektronix
- Philips PMDS-1 and PMDS-2
- Intel MDS
- IBM-PC based MDS

Reference:

- Microprocessor :Principles and Applications
  -By Ajit Pal

Chapter 8