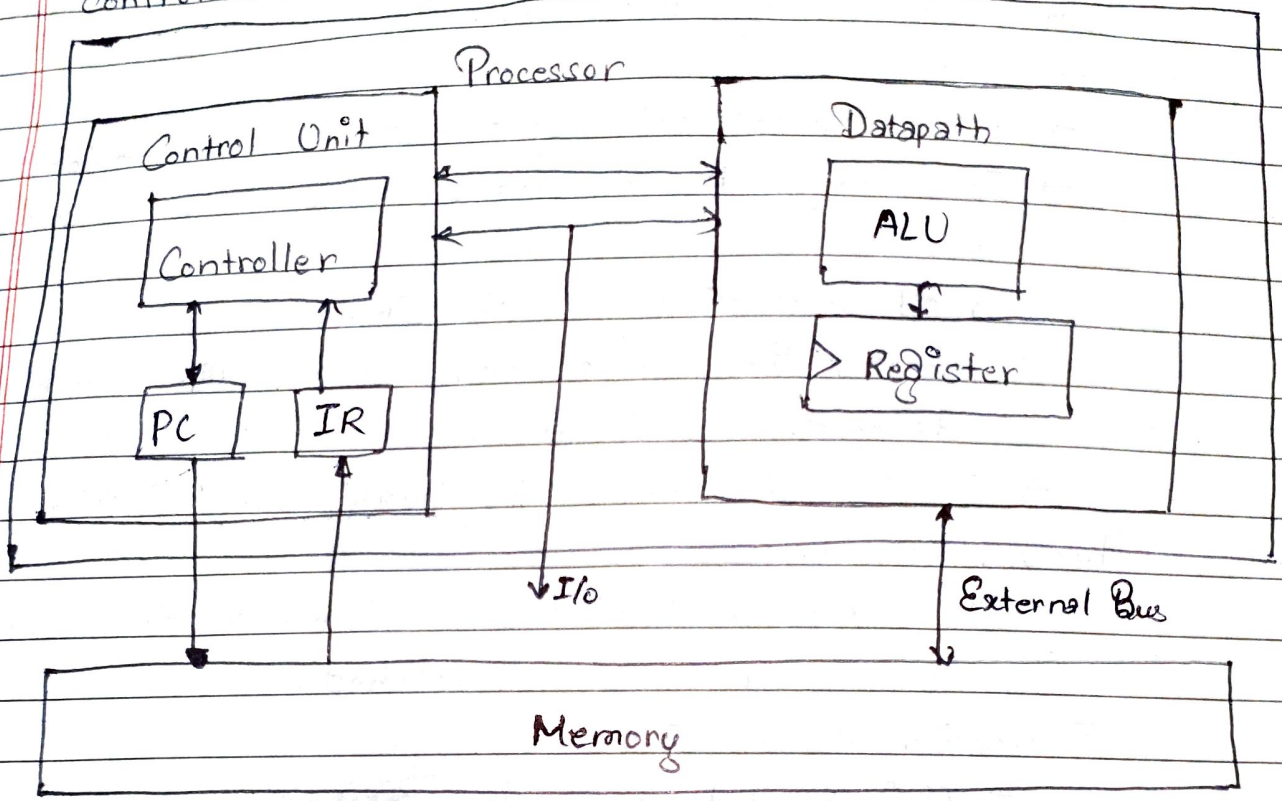
GOOD MORNING
PAGE NO. :
DATE : __/__/____

Week #4:

① Explain datapath & control unit with block diagram of basic architecture of general purpose processor.

Ans: In general purpose processor, the dapath and control unit are connected to the memory.



Basic Architechure of General Purpose Processor.

Datapath :

- It consist of circuitry to manipulate data and temporary data storage.
- It contains ALU which manipulates data through operations such as add, subtract, shifting etc.
  ALU also generates status signal to represent various conditions such as carry, zero, sign, priority.
- Status signals ~~is~~ are stored in status register.
- Datapath contains registers to store temporary data and different status information.
- It also has internal bus for data transfer with datapath & external bus for movement of data from & to memory.

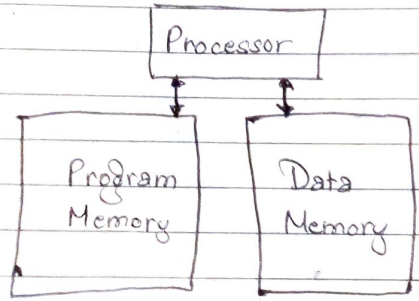
Control unit:

- It contains circuitry to generate control signals to carry out various operations.
- It consist of controller, program counter (PC) and Instruction register (IR).
- Controller:
  - → It consists of state register and control logic.
  - → It sequences through states and generates the control signals to:
    - ~ read instructions into the IR.
    - ~ control the flow of data between ALU, register of datapath and memory.
  - → It also determines the next value of PC.
- Program Counter:
  - → It is used to hold the address of the next program instruction to be fetched.
- Instruction Register (IR):
  - → It is used to hold the fetched instruction

(Q2) Classify memory architecture based on program & data storage:

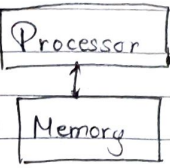
Ans: The memory architecture based on program & data storage can be classified as:

i) Harduard Architecture:



- It has distinct data & program memory space.
- It has more connections.
- The fetch of instruction and data is simultaneous. So, *Improved performance.

ii) Princeton Architecture



- Data and program share memory space.
- It has lesser connection.
- Data and instructions can't be fetched simultaneously.

(3) Write short notes on instruction execution (Fetch Instruction, Decode, Fetch Operand, Execute operations & Store).

Ans: Instructions are set of code that carry out particular function. The various stages of instruction execution are:

(i) Fetch Instruction:

Execution starts with fetching instruction from main memory. The instruction at the current Program counter (PC) will be fetched and stored in Instruction register (IR).

(ii) Decode Instruction

During this stage, the encoded instruction in the IR is interpreted by the decoder.

(iii) Fetch Operand:

The operand can be register or memory for the given operation. In operations including registers,

data required are loaded into register as specified by the instruction.

(iv) Execute Operation:
The ALU handles the ALU operations. The loaded registers are fed to the i/p of ALU to carry out the operation.

(v) Store results.
The output is then stored into register or memory as defined in IR.

---

(4) Why assembly level programmer (embedded system designer) needs architecture of processor. Write short notes on the following:
A. Instruction Set
B. Program & Data Memory Space
C. Available Registers
D. Input Output Facility.
E. Interrupts
F. Operating System.

Ans: Assembly level programmer needs to know about architecture of processor because assembly language is quite close to the particular instruction that the CPU understands.

A. Instruction Set:
Instruction represents bit configurations for operation that can be carried by the processor. In Embedded system, design need some portion of assembly code; hence, need to know the instruction set available for that particular processor they are working.
Each instruction consist of op-code and operand i/e in general. Opcode field specifies the operation to be carried. Operand field specifies the data or location of actual data for the operation specified in the opcode.

B Program & Data Memory Space
Embedded system programmer must be aware of size available for:
(i) Program &
(ii) Data
Program has defined memory space limit. So, programmer should write program within limit.
Example: On-chip memory for program and data are fixed in microcontrollers. So, efficient coding is need to write code so as not to exceed the limit.

C. Available Registers:
Programmer should have knowledge of number of registers available for both general & specific purpose.
Eg: Multiplication in 8051 microcontroller can be done using accumulator and register B. For structured programming, knowledge of accumulator and register B is not needed. However, various special function registers reused for configuring timer, serial communication & interrupts

D. Input/Output Facility.
All processors are ~~facilitated~~ facilitated with input output pins to communicate peripherals.
Embedded system programmers should be alert about the number of i/o pins and their functions. In parallel i/o port can be read or written to use specific function register.

E. Interrupts:

It is a facility provided to programmer (user) in which processor serves the device which requires urgent attention. It causes processor to suspend execution of current program and starts executing interrupt service routine. Programmers should have knowledge on the types of interrupts supported by the processor and must write service ~~rego~~ routine when required.

F. Operating System:

An operating system is a layer of software that provides low-level services to the application layer, a set of one or more programs executing on the CPU consuming and producing input and output data. The task of managing the application layer involves the loading and executing of programs, sharing and allocating system resources to these programs, and protecting these allocated resources from corruption by non-owner programs.

The OS abstracts away the details of the underlying hardware and provides the application layer an interface to the hardware through the system call mechanism.

(5.) Explain commonly used addressing modes.

Ans: The commonly used addressing modes are:

i. Immediate addressing modes:

This is the simplest way of addressing. When it executes the instruction will operate on immediate hexadecimal number. The operand is present in instruction in this mode. This mode is used to define and use constants of set initial values of variables. The operand may be 8 bit or 16 bit

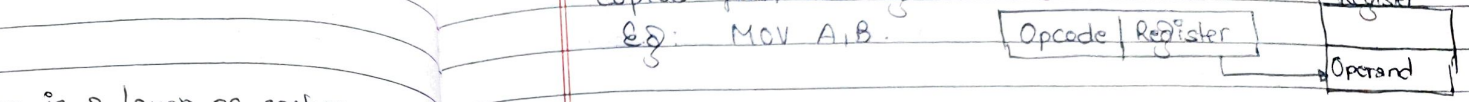
| Operand | Data |

Eg: MVI B, 05H
LXI B, 7A21H

ii Register direct addressing

In register direct addressing mode, the data is copied from one register to another.

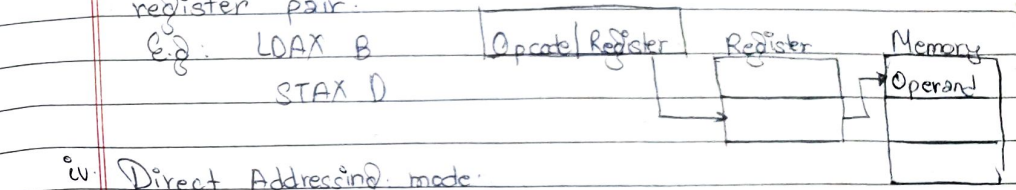
Eg: MOV A,B



iii Register Indirect Addressing

The address of the operand is specified by register pair.

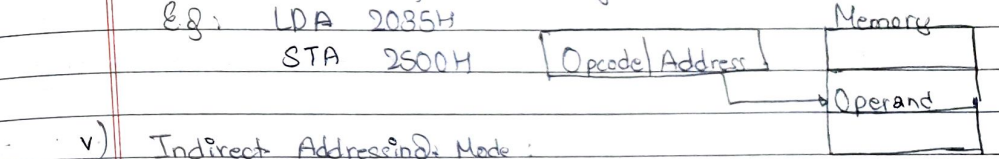
Eg: LDAX B
STAX D



iv Direct Addressing mode:

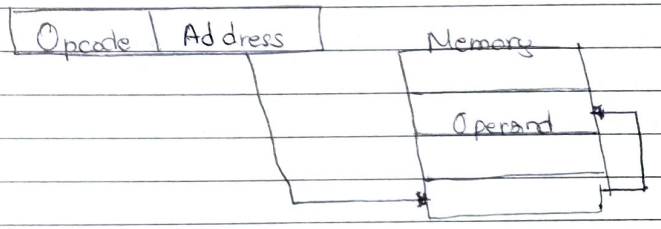
In this mode, the effective address of the operand is specified directly in the instruction.

Eg: LDA 2035H
STA 2500H



v) Indirect Addressing Mode:

In this mode the address field of the instruction gives the address where the effective address is stored in memory.



(6) Write notes on development environment for structured programming and Assembly Language programmer.

Ans: Processors along with different development tools are used for development of software or an embedded system.

The processors used are:

a) Development processor:

It is used to write and ~~delay~~ ~~the~~ debug the program. ~~It~~ may not be part of embedded system implementation.

E.g: Desktop computers

b) Target processor:

It is the one in which program are loaded. It is always part of system implementation.

E.g: AVR, 8051, PIC microcontroller, etc.

The tools used are:

(a) Tools for implementation phase:

i) Assembler:

It converts assembly instructions into binary machine instruction.

It replaces opcode and operand mnemonic with binary equivalent.

ii) Compiler:

It converts high level program into machine program. Each high-level constructs may be translated to several machine instructions.

iii) Linker:

It combines object files into a single executable file. It combines machine instruction from user code and instructions from standard library.

(b) Tools for verification phase:

i) Debugger:

Debugger are programs that are used to test and debug the targeted program. Debuggers are program that runs on development processor but execute code designed for target processor.

Debugger simulates the function of target processor and allow evaluation and correction of program in development processor.

ii) Emulator:

Emulators are hardware or software that enables one system to behave like another system. It consists of debugger coupled with a board connected to development processor.

iii) Device programmers:

These are devices to load binary machine programs into target processor's memory. With this device, program can be tested in its ~~release~~ realistic form with higher accuracy on actual system.

Design Flows:

All system development process includes implementation and verification phase.

During implementation:

Various tools such as assembler, compiler, linker are used.

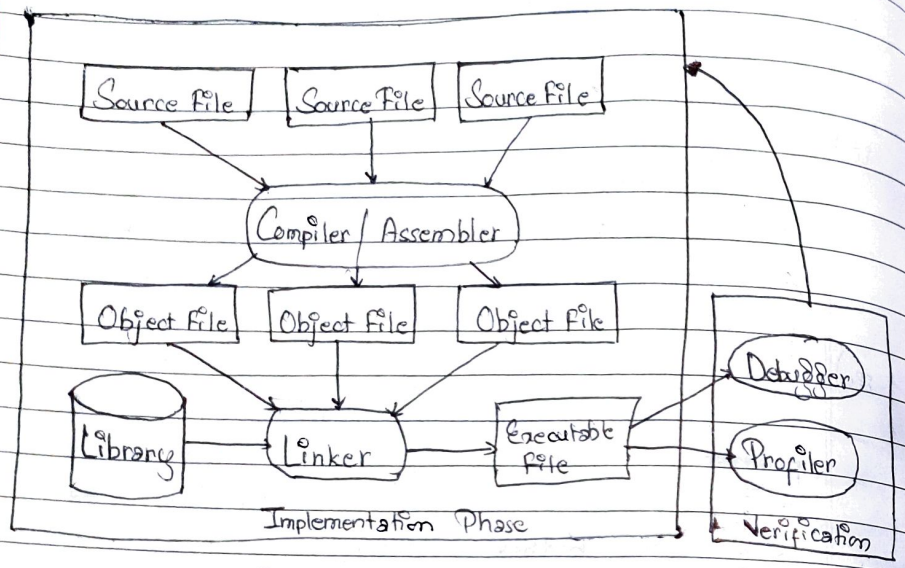
During verification phase:

Debugger, emulator, device programmer are used.

(7) Describe (with block diagram) tools for implementation phase and verification phase for both software development process and embedded system development process.

Ans:

(a) Software development process:



Software development process

The tools used for implementation phase are:

i) Compiler/ Assembler

Source code is written using an editor. Compiler converts high level program into machine level.

ii. Linker:

Linker combines all required files into final executable file.

The tools used for verification phase are:

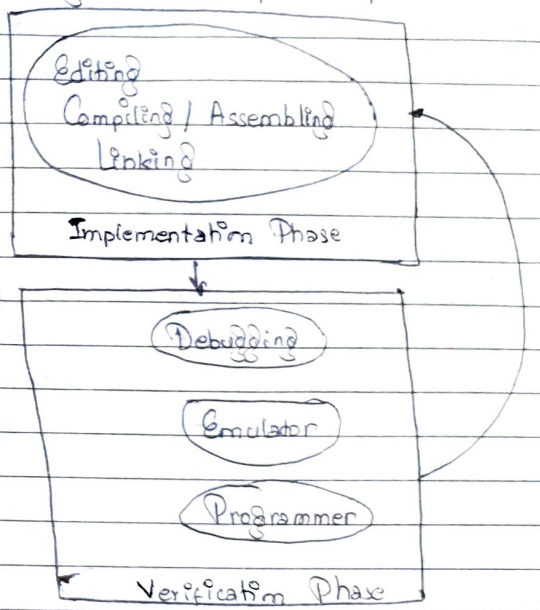
i) Debugger.

Debuggers are programs that are used to test and debug the targeted program. Debuggers are program that runs on development processor but execute code designed for target processor. Debugger simulates the function of target processor and allows evaluation and correction of program in development processor.

ii) Profiler

Profiler are used for performance analysis of the program. Time and space complexity can be analysed. Time complexity includes duration of program whereas space complexity includes memory usage.

(b) Embedded system development processor:



The tools used for Implementation phase are:

i) Editing:

It is the tool used for writing the program.

ii) Assembler/ Compiler:

It is the tool used to convert high level progr to machine level.

iii) Linker:
It combines all required file to final executable file.

The tools used for verification phase are:

i) Debugger:
Debugger are programs that are used to test and debug the targeted program.

ii) Emulator:
Emulator are hardware or software that enables one system to behave like another.

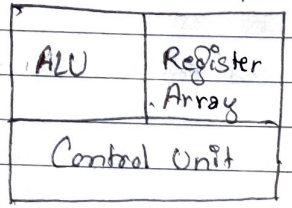
iii) Programmer:
Programmers are devices to load binary machine program into target processor's memory.

---

8. Short notes on microprocessors, digital signal processor. Describe methods to study speed of any processor.

Ans: Microprocessor:

A microprocessor is a multipurpose programmable, clock driven, register based electronic device that reads binary instruction from a storage device called memory, accepts binary data as input, and processes data according to those instructions and provides results as output. The microprocessor operates in binary 0 and 1 known as bits are represented in terms of electrical voltages in the machine that means the 0 represents low voltage level and 1 represents high voltage level.



Microprocessor.

Digital Signal Processor:

They are intended to process large amount of data. Source includes image captured by a camera, voice packet through a network routers. It contains many register files, memory blocks, memory blocks and other Arithmetic unit.

DSPs often provide instructions that are central to digital signal processing such as filtering and transforming vectors or metrics of data. In a DSP, frequently used arithmetic units are implemented in hardware and thus executes orders of magnitude faster than a software implementation.

The various method to study speed of any processor are:

a) Clock Speed
When same number of instruction per cycle, speed comparison based on clock speed is reasonable.

b) Instruction per speed
It is an evaluation based on instruction executed per second.

c) Dhrystone benchmark
It is a benchmark software program used to test a computer's processor's integer performance. Each Dhrystone is a measurement of how many times the program can run each second.

d) Millions of instruction per second (MIPS)
It is a measure of machine instructions that a computer can execute in one second.

GOOD MORNING
PAGE NO. :
DATE : / /

(9.) Write complete design steps for General purpose processor.

Ans: The steps for General purpose processor are.

I. Instruction Set Design.
- Various operations are defined. The number of registers and size of memory are defined.
- Number and kinds of operation are considered.
- Number, type and location of operands must be selected.
- Size & format of instruction must be selected.

II. Creating a FSMD.
FSMD represents state diagram of given instructions and functionality.
- Reset state is defined to clear various registers
- Fetch state is used to load instruction from memory to IR. PC is incremented after fetch.
- Decode state is used as ~~transmission~~ transition between instruction execution and fetch.
- Execute state is defined based on the operation represented by opcode.

III. Build Datapath.
- For each declared variable, we need to initiate a storage device.
- Initiate functional unit to carry out FSMD operation. Eg: General purpose ALU.
- Connect different components with datapath.

IV. Development of FSM / Controller.
- Rewrite the FSMD states without instruction or operations.
- Equivalent binary operations on control signal must be written in each state rather than the operation.
- Each operation with binary operation.