IDHAYA COLLEGE FOR WOMEN

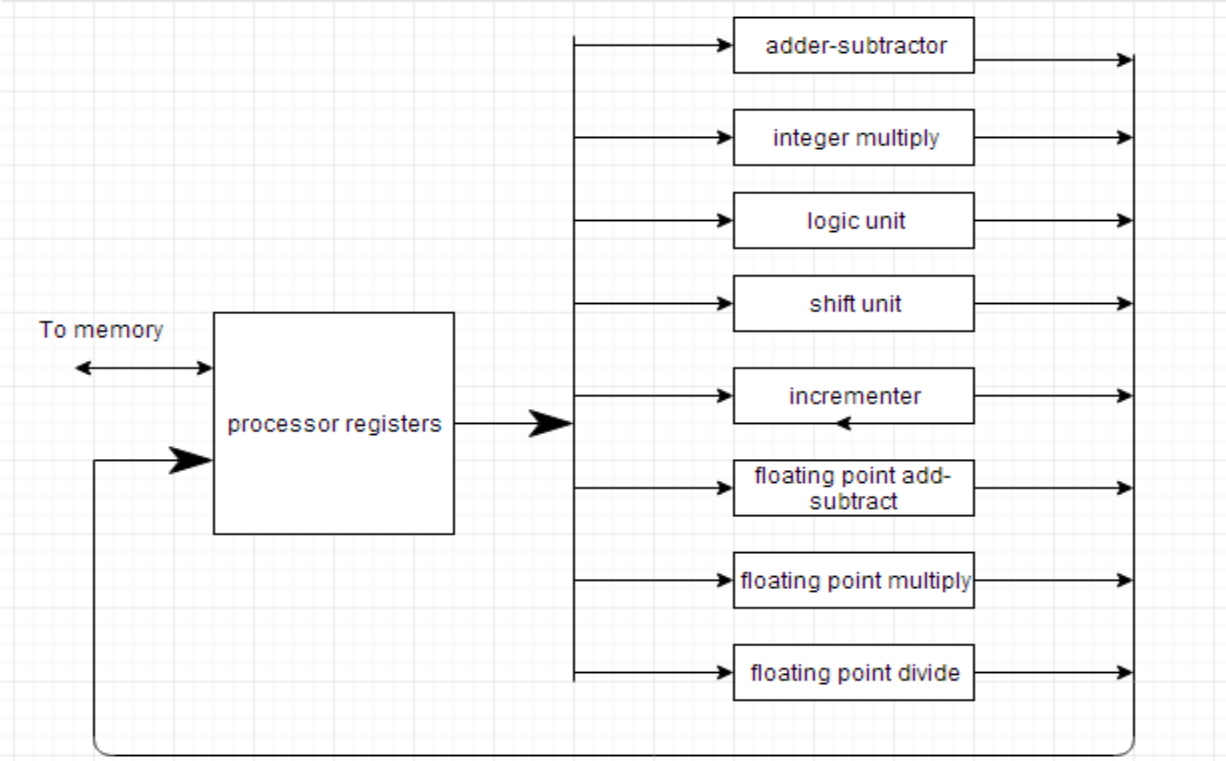
KUMBAKONAM – 612 001

DEPARTMENT OF INFORMATION TECHNOLOGY

SEMESTER	**:**	**IV**
CLASS	**:**	**III IT**
SUBJECT- INCHARGE	**:**	**Ms. G. AKILAMADHUVADHANI**
SUBJECT NAME	**:**	**Computer and Organisation Architecture**
SUBJECT CODE	**:**	**16SACIPH2**
TOPIC	**:**	**UNIT 5**

UNIT –V

- Parallel Processing and Data Transfer Modes in a Computer System
- Instead of processing each instruction sequentially, a parallel processing system provides concurrent data processing to increase the execution time.
- In this the system may have two or more ALU's and should be able to execute two or more instructions at the same time. The purpose of parallel processing is to speed up the computer processing capability and increase its throughput.
- **NOTE: Throughput is the number of instructions that can be executed in a unit of time.**
- Parallel processing can be viewed from various levels of complexity. At the lowest level, we distinguish between parallel and serial operations by the type of registers used. At the higher level of complexity, parallel processing can be achieved by using multiple functional units that perform many operations simultaneously.



PROCESSOR WITH MULTIPLE FUNCTIONAL UNITS

Data Transfer Modes of a Computer System

According to the data transfer mode, computer can be divided into 4 major groups:

1. **SISD**
2. **SIMD**
3. **MISD**
4. **MIMD**

SISD (Single Instruction Stream, Single Data Stream)

It represents the organization of a single computer containing a control unit, processor unit and a memory unit. Instructions are executed sequentially. It can be achieved by pipelining or multiple functional units.

SIMD (Single Instruction Stream, Multiple Data Stream)

- It represents an organization that includes multiple processing units under the control of a common control unit. All processors receive the same instruction from control unit but operate on different parts of the data.
- They are highly specialized computers. They are basically used for numerical problems that are expressed in the form of vector or matrix. But they are not suitable for other types of computations

MISD (Multiple Instruction Stream, Single Data Stream)

- It consists of a single computer containing multiple processors connected with multiple control units and a common memory unit.
- It is capable of processing several instructions over single data stream simultaneously. MISD structure is only of theoretical interest since no practical system has been constructed using this organization.

MIMD (Multiple Instruction Stream, Multiple Data Stream

- It represents the organization which is capable of processing several programs at same time.
- It is the organization of a single computer containing multiple processors connected with multiple control units and a shared memory unit.

- The shared memory unit contains multiple modules to communicate with all processors simultaneously. Multiprocessors and multicomputer are the examples of MIMD. It fulfills the demand of large scale computations.

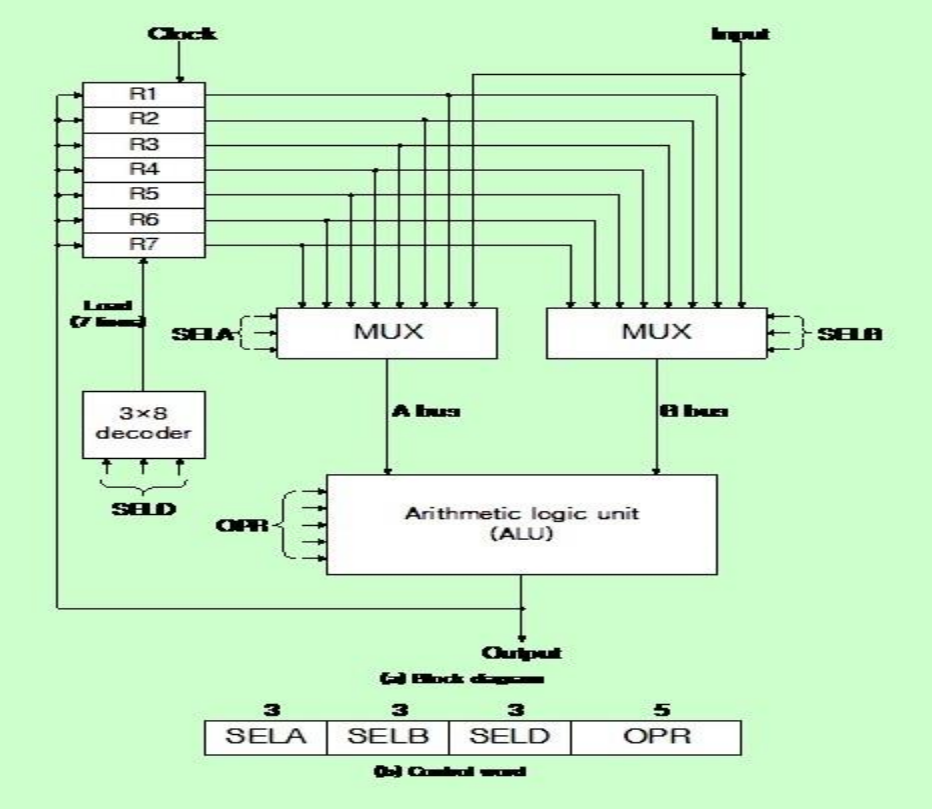
General Register Organization:

The number of registers in a processor unit may vary from just one processor register to as many as 64 registers or more.

1. **One of the CPU registers is called as an accumulator AC or 'A' register. It is the main operand register of the ALU.**
2. **The data register (DR) acts as a buffer between the CPU and main memory. It is used as an input operand register with the accumulator.**
3. **The instruction register (IR) holds the opcode of the current instruction.**
4. **The address register (AR) holds the address of the memory in which the operand resides.**

- The program counter (PC) holds the address of the next instruction to be fetched for execution.
- Additional addressable registers can be provided for storing operands and address. This can be viewed as replacing the single accumulator by a set of registers.
- If the registers are used for many purposes, the resulting computer is said to have general register organization.
- In the case of processor registers, a registers is selected by the multiplexers that form the buses.
- When a large number of registers are included in the CPU, it is most efficient to connect them through a common bus system.
- The registers communicate with each other not only for direct data transfers, but also while performing various micro-operations.
- Hence it is necessary to provide a common unit that can perform all the arithmetic, logic and shift micro-operation in the processor.

A Bus organization for seven CPU registers:



- The output of each register is connected to true multiplexer (mux) to form the two buses A & B. The selection lines in each multiplexer select one register or the input data for the particular bus.
- The A and B buses forms the input to a common ALU. The operation selected in the ALU determines the arithmetic or logic micro-operation that is to be performed.
- The result of the micro-operation is available for output and also goes into the inputs of the registers.
- The register that receives the information from the output bus is selected by a decoder.
- The decoder activates one of the register load inputs, thus providing a transfer both between the data in the output bus and the inputs of the selected destination register.
- The control unit that operates the CPU bus system directs the information flow through the registers and ALU by selecting the various components in the systems.

R1 ® R2 + R3

(1) MUX A selection (SEC A): to place the content of R2 into bus A

(2) MUX B selection (sec B): to place the content of R3 into bus B

(3) ALU operation selection (OPR): to provide the arithmetic addition (A + B)

(4) Decoder destination selection (SEC D): to transfer the content of the output bus into R1

These form the control selection variables are generated in the control unit and must be available at the beginning of a clock cycle. The data from the two source registers propagate through the gates in the multiplexer and the ALU, to the output bus, and the destination registers, all during the clock cycle intervals.