Processor organization

To understand processor organization, let us consider the requirements of CPU:

Fetch instruction: CPU reads instruction memory.

Interpret instruction: The instruction is decoded to determine what action is required.

Fetch data: The execution of an instruction may require reading of data from memory or I/O module.

Process data: The execution of an instruction may require performing arithmetic or logical operation on data.

Write data: The results of execution may require writing data to memory or an I/O module.

To perform all these tasks CPU needs to store data temporarily. It should remember the location of last instruction, so that it can fetch the next instruction. Data and instructions should be stored temporarily while execution of instruction.

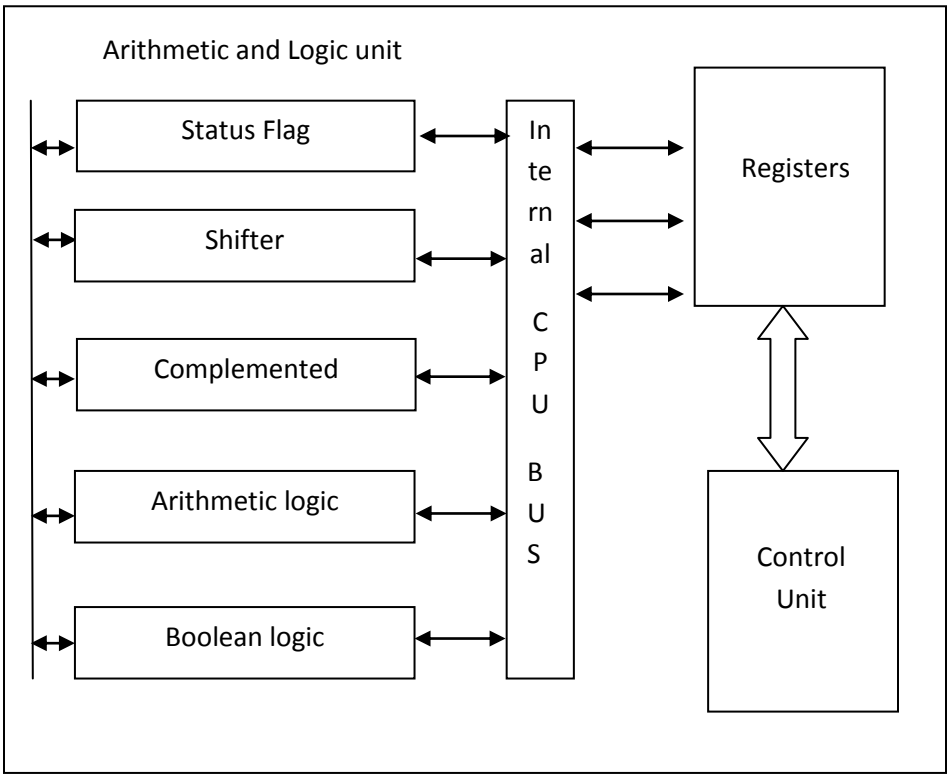
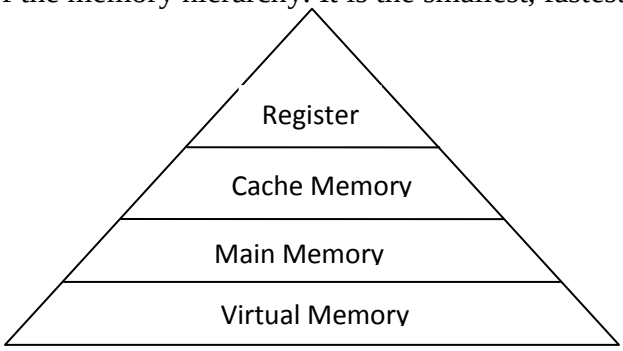


Figure: Internal structure of CPU.

The data transfer and logic control paths are indicated including an element labeled internal CPU bus. This is used to transfer data between various registers and the ALU.

Register Organization

Registers are at the top of the memory hierarchy. It is the smallest, fastest and most costly memory.



Within CPU, there is set of registers. Register in CPU performs two roles.

1. User visible registers: It minimizes the main memory reference by optimizing use of registers. Four registers are used as these type of registers:
   i. General purpose: These registers can be used for variety of functions. They can be used to hold operand. They can also be used for addressing functions.
   ii. Data: These registers are used to hold data and are not used for the calculation of an operand address.
   iii. Address: These are used for addressing modes. E.g. Segment Pointers are used to hold the base of the segment. Index registers are used for indexed addressing modes. Stack pointer is used to point top of the stack in the memory as implicit addressing.
   iv. Condition Codes: Conditional codes (flags) are bit set by CPU hardware as the result of operation. E.g. an arithmetic operation may produce a positive, negative, zero or overflow result. Condition codes are collected in one or more register. Usually, they form part of control register.
2. Control and status registers: These are used by control unit to control the operation of CPU. For instruction execution four registers are used:
   i. Program Counter (PC): Contains address of an instruction to be fetched.
   ii. Instruction register (IR): Contains instruction most recently used.
   iii. Memory Address register (MAR): Contains address of a location in memory.
   iv. Memory Buffer register (MBR): Contains a word of data to be written to memory or the word most recently used.

   Program Status Word (PSW) contains status information. PSW contains condition codes plus other status information. Common flags are:

   Sign: Contains sign bit of the last arithmetic operation.
   Zero: Sets when result is zero.

   Carry: Sets when an operation resulted in a carry or borrows out of a high-order bit.

   Equal: Set if a logical compare result is equal.

Overflow: Used to indicate arithmetic overflow.

Interrupt enable/disable: Used to enable or disable interrupts.

Supervisor: Indicates whether the CPU is executed in supervisor or user mode.

Instruction Cycle

Instruction cycle is made up of a number of smaller units. One subdivision is fetch, indirect, execute and interrupt. Execution of a program consists of sequential execution of instructions.

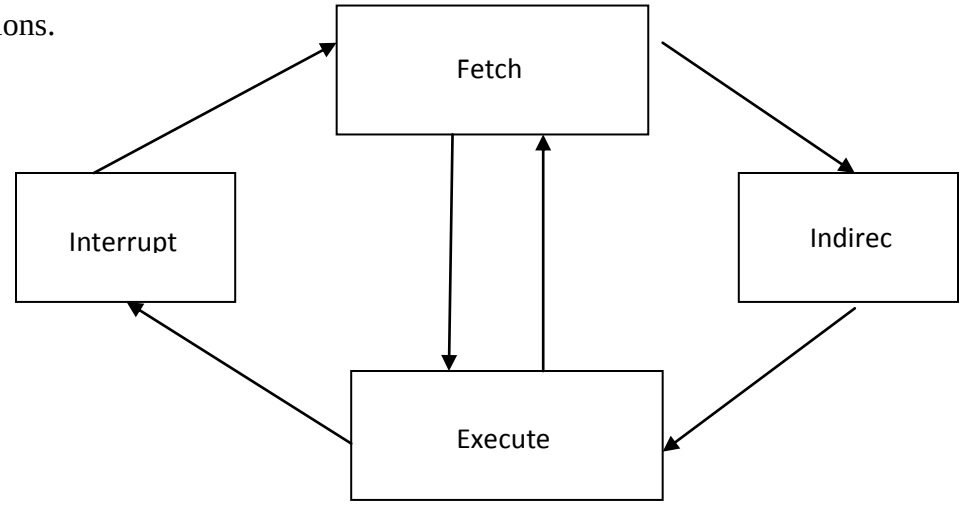


Figure: The Instruction Cycle

Instruction is executed during an instruction cycle made up of shorter sub cycles. The performance of each sub cycles involves micro-operations.

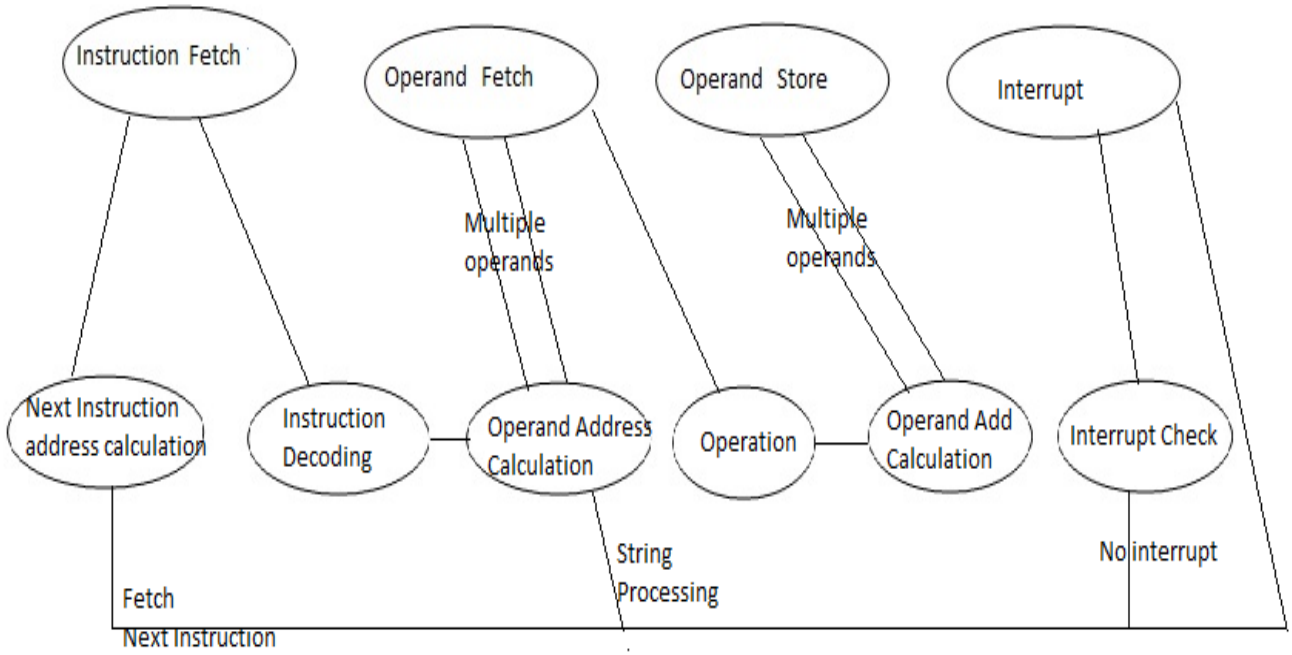


Figure: Instruction Cycle State Diagram

The Arithmetic and Logic unit

The ALU is the part of computer that actually performs arithmetic and logical operations on data. All other elements of computer system bring data into the ALU for it to process. An ALU is based on the use of simple digital logic devices that can store binary digits and perform simple Boolean logic operations.

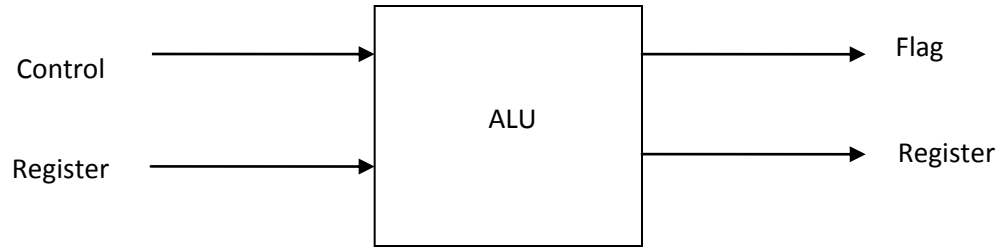


Figure: ALU inputs and Outputs

Figure above shows how ALU is interconnected with the rest of the processor. Data are present in registers and also the results are stored in registers. These registers are temporary storage locations within processor connected by signal paths. ALU also sets flag as the result of operation. e.g An overflow flag is set to 1 if the result of a computation exceeds the length of register to which it is to be stored. The control unit provides signals that control the operation of the ALU and the movement of data into and out of the ALU.

Design Principles for Modern System

The continuing technological revolution has enabled the development of modern complex and powerful system. E.g. applications that require micro-processor based system includes

Image Processing

Speech recognition

Video Conferencing

Multimedia Authoring

Voice and video annotation of files

Simulation Modeling

Today's systems support engineering and scientific applications as well as simulation systems. They have ability to support image and video applications. Business also relies on servers to handle database processing, transaction and to support client server networks. This serves as design principle for modern systems. Driving factors that is required to achieve this design principle is:

Microprocessor speed:

According to Moore's law chip makers gave new generation of chips every third year, with four times as many transistors. In, microprocessors the addition of new circuits boosts up the speed

reducing the distance between them. This improved the performance four or five times every third year.

The raw speed of microprocessor achieves its potential, when it has work to do in the form of computer instruction. To build the processor techniques used were: Branch prediction, Data flow analysis and speculative execution.

In Branch prediction processor looks the instruction fetched form memory and predicts the branches or group of instructions to be processed next. If processor guesses right, it prefetches the correct instruction and buffer them so that processor is busy.

The processor analyzes which instructions are dependent on each other's results or data. This prevents unnecessary delay.

Using branch prediction and data flow analysis processor can perform speculative execution of instructions before their actual appearance holding results in temporary locations. This makes processor execution engine as busy as possible.

Performance Balance

Performance balance is adjusting of organization and architecture to compensate for mismatch among the capabilities of various components. E.g the mismatch speed of DRAM and processor. To recover this fastest memory cache memory is used in between main memory and processor. Cache is used to bring data into processor.

Another area focuses on handling of I/O devices. The problem is data transfer between processor and peripherals. Strategies here used include caching and buffering and also the use of high speed interconnection buses. Also the use of multiprocessor can satisfy I/O devices.

Designers have to balance the throughput and processing demands on the processor components, main memory, I/O devices, and the interconnection structures. Designers need to consider:

i. The rate at which performance is changing in various technology areas (processor, buses, memory, peripherals) differs greatly from one type of element to other.
ii. New applications and new peripherals constantly change the nature of the demand on the system in terms of typical instruction profile and data access patterns.