TOPIC 6 – COMPUTER ARCHITECTURE

LEARNING OUTCOMES

By the end of this topics, you will be able to:

1. describe the computer operation, interconnection of buses and addressing techniques
2. explain the flags, condition codes, status registers and storage units.
3. explain the subroutine calls, interrupts, pipelined computers.
4. identify the RISC architecture, CISC architecture, security and protection.

INTRODUCTION

This topic explains the computer operation, interconnection of buses, addressing techniques, flags, condition codes, status registers, storage units, subroutine calls, interrupts, pipelined computers, RISC architecture, CISC architecture, security and protection.

6.1 PROCESSING AND CONTROL UNITS

The central processing unit (CPU) is a part of the computer architecture that performs as an electronic circuitry within a computer. It carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.

The term central processing unit (CPU) has used in the computer industry since the early 1960s. It refers to a processor, more specifically to its processing unit and control unit (CU), distinguishing these core elements of a computer from external components such as main memory and I/O circuitry.

The principal components of a central processing unit (CPU) are arithmetic logic unit (ALU) which performs arithmetic and logic operations. The processor registers supply operands to the arithmetic logic unit (ALU) and store the results of arithmetic logic unit (ALU) operations, and a control unit that orchestrates the fetching from memory and execution of instructions by directing the coordinated operations of the arithmetic logic unit (ALU), registers and other components.

The Figure 6.34 showing the components of central processing unit (CPU).

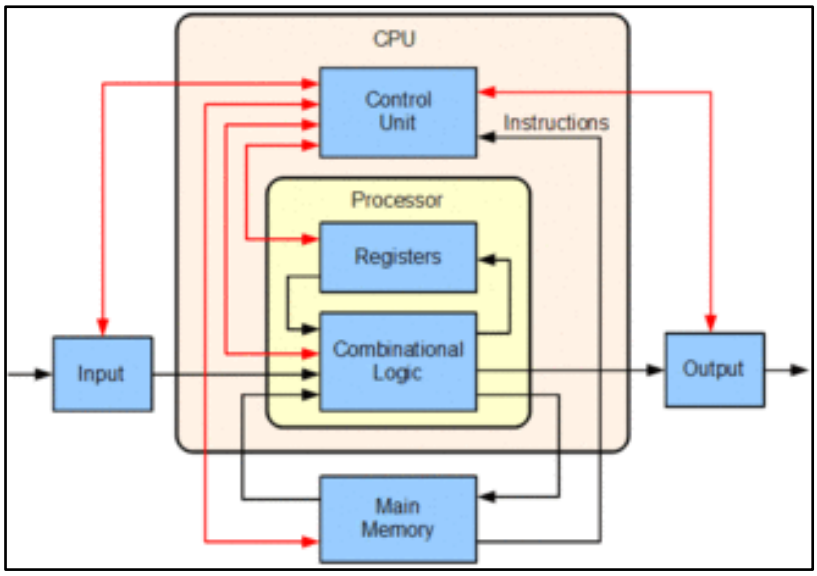

Figure 6.33

SELF CHECK 6.1

1. What id central processing unit (CPU)?
2. Explain THREE (3) components of processor.

6.2 FETCH-EXECUTE INSTRUCTION CYCLE

The basic operational process of a computer is known as an instruction cycle or also called a fetch–decode–execute cycle. It is the process where a computer retrieves a program instruction from its memory, determines what actions the instruction dictates, and carries out those actions. Figure 6.34 shown the fetch-execute instruction cycle.

The process starts when the are instructions sent for processing then the instructions will be fetched, execute. If there is no issue the process will be halted after complete the execution. However, if there is issue like an interrupt occur in the middle of execution, it will be checked (if its enabled) with the interrupt cycle then restart the cycle. Otherwise it will be sent straight away to restart the execution cycle.

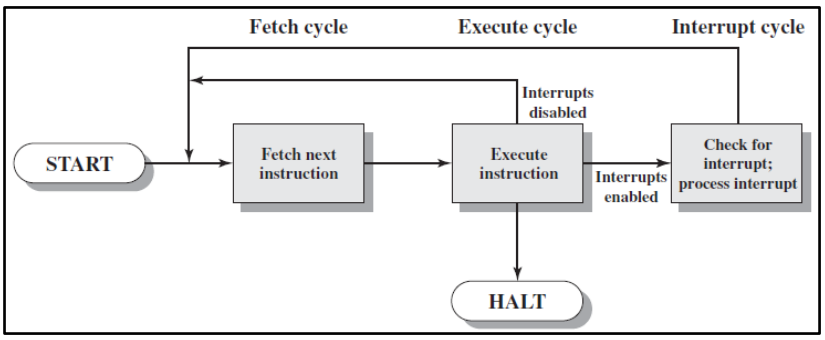

Figure 6.34

SELF CHECK 6.2

1. Describe the fetch-execute instruction cycle in processor.

6.3 INTERCONNECTION BUSES

In the computer architecture, the system bus is a single computer bus that connects the major components of a computer system. It is also combining the functions of a data bus to carry information, an address bus to determine where it should be sent, and a control bus to determine its operation.

The system bus consists of data bus, address bus and control bus (Figure 6.35). The data bus is a path that carries the data among system modules on the data lines. The address bus determines the maximum possible memory capacity of the system and also used to address input/output ports. It used the address lines to design the source or destination of the data on the data bus. The control bus controls the functioning of all other components connected to computer system. It is used to transfer the control signals from component to another component. It is also used to communicate with all connected devices via cables and printed circuits such as motherboard.

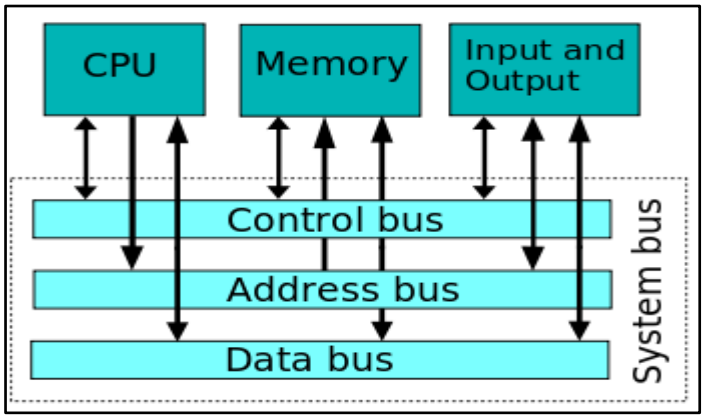

Figure 6.35

SELF CHECK 6.3

1. Explain the following types of buses in computer system:
   - Data Bus
   - Address Bus
   - Control Bus

6.4 ADDRESSING TECHNIQUES

The addressing techniques specifies the address of the operands for the operation to be performed. This operation must be executed on some data stored in computer register or the main memory. During the program execution. the operands are chosen by depending on the addressing mode of the instruction.

The addressing mode specifies a rule for interpreting or modifying the address field of the instruction between the operand is activity referenced.

The most common addressing techniques are:

- Implied and Immediate Addressing Modes
  Implied addressing also known as implicit or inherent addressing mode has no operand specified in the instruction.
  The immediate addressing mode has specified operand in the instruction. the format of the instruction as shown in Figure 6.36.

Figure 6.36

- Direct or Indirect Addressing Modes
  The direct addressing mode consists of 3-bit opcode, 12-bit address and a mode of designated as (I). The mode bit (I) is zero for Direct Address and 1 for Indirect Address. It is shown in the Figure 6.37.

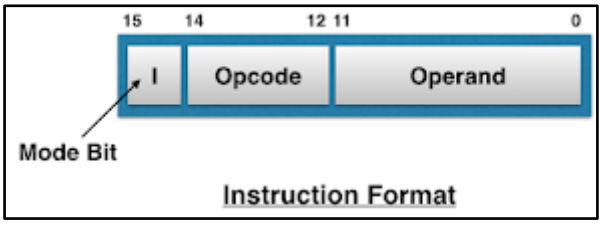

Figure 6.37

The indirect addressing mode or absolute addressing which the address of data (operand) is specified in the instruction. In this mode, the operand resides in memory and its address is given directly by the address field of the instruction. In other words, in this mode, the address field contain the Effective Address of operand. For example, the instruction **ADD (A)** which Means adds the content of cell pointed to contents of A to Accumulator. So, it can be illustrated as in Figure 6.38.

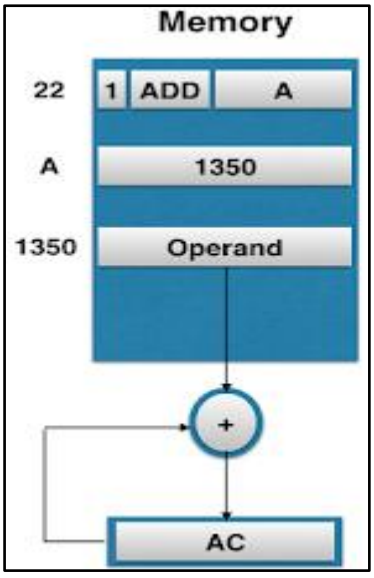

Figure 6.38

- Register Addressing Modes
  In Register Addressing Mode, the operands are in registers that reside within the CPU. In this mode, the instruction specifies a register in CPU, which contain the operand. It is like Direct Addressing Mode but the only difference is the address field refers to a register instead of memory location. Example **EA=R,** so it can be illustrated like shown in Figure 6.39.

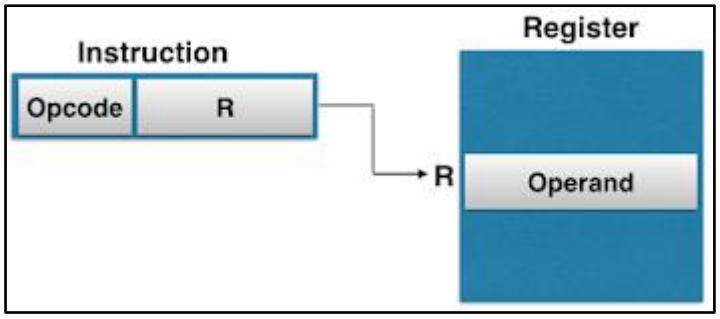

Figure 6.39

- Register Indirect Addressing Mode
  The Register Indirect Addressing Mode is where the instruction specifies a register in CPU whose contents give the operand in memory. In other words, the selected register contains the address of operand rather than the operand itself.
  For example, EA = (R) which means the control fetches instruction from memory and then uses its address to access Register and looks in Register (R) for effective address of operand in memory. It can be illustrated as shown in Figure 6.40.

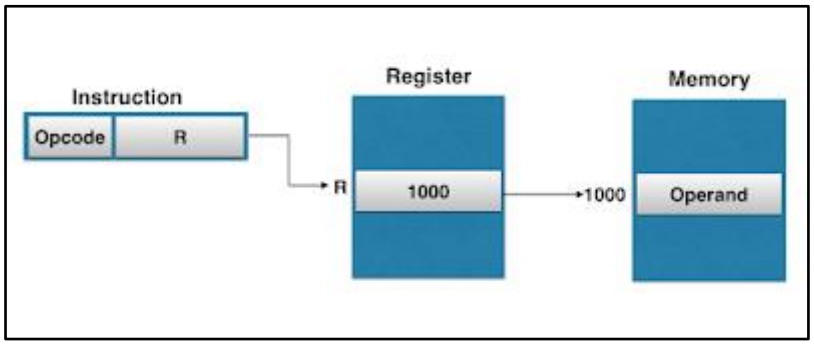

Figure 6.40

- Auto-Increment and Auto-Decrement Addressing Modes
  These addressing modes are similar to Register indirect Addressing Mode except that the register is incremented or decremented after or before its value is used to access memory.

  The modes are required because when the address stored in register refers to a table of data in memory, then it is necessary to increment or decrement the register after every access to table so that the next value is accessed from memory. Thus, these addressing modes are common requirements in computer.

  Auto-increment Addressing Mode are similar to Register Indirect Addressing Mode except that the register is incremented after its value is loaded or accessed at another location like accumulator (AC). Let say, the Effective Address is equal to EA = (R) but, after accessing operand, register is incremented by 1. It is like shown in Figure 6.41.

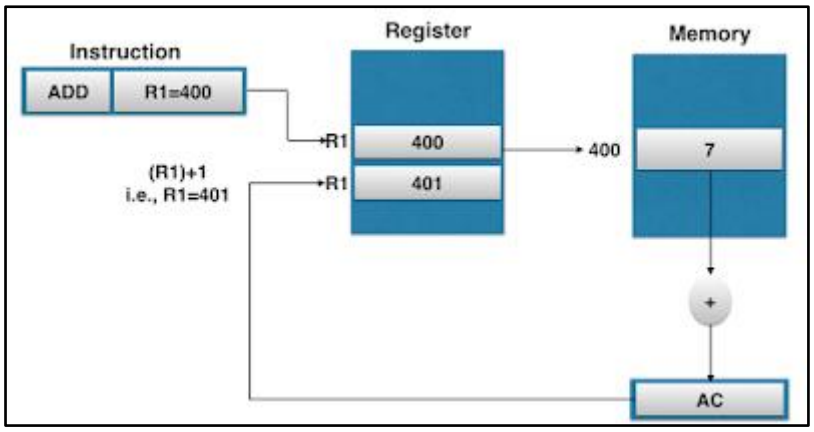

Figure 6.40

The effective address is (R) = 400 and operand in AC is 7. After loading R1 is incremented by 1. So, it becomes 401. It means, in the Auto-increment mode, the R1 register is increment to 401 after execution of instruction.

The auto-decrement Addressing Mode is reverse of auto-increment which the register is decrement before the execution of the instruction. In this case, an effective address is equal to EA= (R) - 1. It is shown in Figure 6.41.

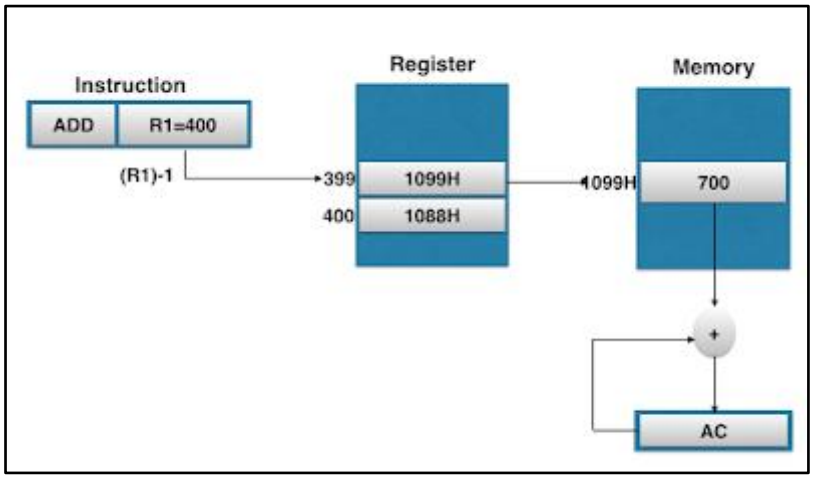

Figure 6.41

In the Auto-decrement mode, the register R1 is decremented to 399 prior to execution of the instruction, means the operand is loaded to accumulator, is of address 1099H in memory, instead of 1088H. In this case effective address is 1099H and contents loaded into accumulator is 700.

- Displacement Based Addressing Modes

  The Displacement Based Addressing Modes is a powerful addressing mode due to it is a combination with direct addressing or register indirect addressing mode. For example, EA=A+ (R) which means a Displacement Addressing Modes requires the instruction with two address fields; explicit which one of address field indicates direct address and the other one indicates indirect address. The value contained in one addressing field is A, which is used directly and the value in other address field is R, which refers to a register whose contents are to be added to produce effective address (Figure 6.42).

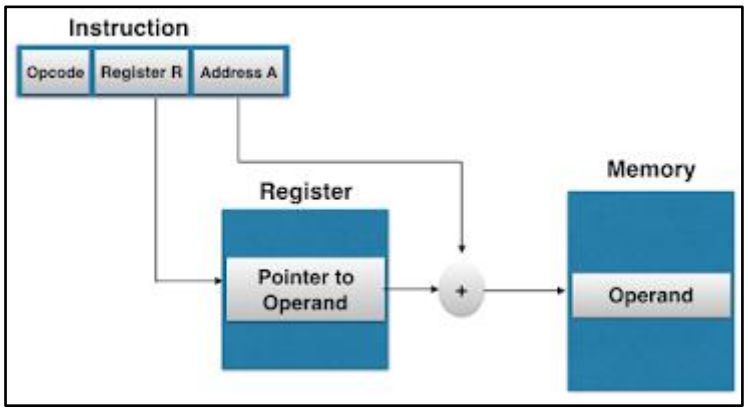

Figure 6.42

SELF CHECK 6.4

1. Explain any FOUR (4) addressing techniques.

6.5 FLAGS, CONDITION CODES, AND STATUS REGISTERS

A status register, flag register, or condition code register is a collection of status flag bits for a processor. For example, the FLAGS register of the x86 architecture or flags in a program status word (PSW) register.

The status register is a hardware register that contains information about the state of the processor. Individual bits are implicitly or explicitly read and/or written by the machine code instructions executing on the processor. The status register lets an instruction to take action contingent on the outcome of a previous instruction. Typically, flags in the status register are modified as effects of arithmetic and bit manipulation operations. For example, a Z bit may be set if the result of the operation is zero and cleared if it is nonzero. Other classes of instructions may also modify the flags to indicate status.

The flags are read by a subsequent conditional instruction so that the specified action depending on the processor, a jump, call, return, or so on. It occurs only if the flags indicate a specified result.

Table 6.11

Flag	Name	Description
Z	Zero Flag	Indicates that the result of an arithmetic or logical operation (or, sometimes a load) was zero.
C	Carry Flag	Carry/borrow. Set with a carry from an addition or borrow from a subtraction.
S/N	Sign Flag Negative Flag	Indicates that the result of a mathematical operation is negative.
V/O/W	Overflow Flag	Overflow. Set if a 2's complement overflow condition is generated.

SELF CHECK 6.5

1. What are flags, condition codes and status register?
2. Example FOUR (4) types of flags.

6.6 STORAGE UNITS-MEMORY, CACHE MEMORY, VIRTUAL MEMORY

Memory is the basic element of a semiconductor memory. The semiconductor memory cells share three properties; first, exhibit two stable or semi stable states that can be used to represent binary 1 and 0, second the capability to write to set the state and third the capability of being read to sense the state.

Another part of the memory is cache memory also called central processing unit (CPU) memory. It is a random-access memory (RAM) that a computer microprocessor can access more quickly than it can access regular random-access memory (RAM). This memory is typically integrated directly with the central processing unit (CPU) chip or placed on a separate chip that has a separate bus interconnect with the central processing unit (CPU).

The basic purpose of cache memory is to store program instructions that are frequently re-referenced by software during operation. Fast access to these instructions increases the overall speed of the software program.

The virtual memory is an area of a computer system's secondary memory storage space such as a hard disk or solid state drive. It acts as if it were a part of the system's random-access memory (RAM) or primary memory and will be used once the random-access memory (RAM) is full.

The virtual memory is a memory management technique that is implemented using both hardware and software. It maps memory addresses used by a program, called virtual addresses, into physical addresses in computer memory. The main storage as seen by a process or task appears as a contiguous address space or collection of contiguous segments.

The operating system manages virtual address spaces and the assignment of real memory to virtual memory. Address translation hardware in the central processing unit (CPU), often referred to as a memory management unit or memory management unit (MMU), automatically translates virtual addresses to physical addresses. Software within the operating system may

extend these capabilities to provide a virtual address space that can exceed the capacity of real memory and thus reference more memory than is physically present in the computer

SELF CHECK 6.6

1. Discuss the function of memory, cache memory and virtual memory.

6.7 SUBROUTINE CALLS

A subroutine is often coded so that it can be started or called several times and from several places during one execution of the program, including from other subroutines. Once the subroutine's task is done, it is branch back or return to the next instruction after the call.

A subroutine is a sequence of program instructions that perform a specific task, packaged as a unit. This unit can then be used in programs wherever that particular task should be performed.

Subprograms may be defined within programs, or separately in libraries that can be used by multiple programs. In different programming languages, a subroutine may be called a procedure, a function, a routine, a method, or a subprogram.

SELF CHECK 6.7

1. Describe the subroutine calls.

6.8 INTERRUPTS

A software interrupt is caused either by an exceptional condition in the processor itself, or a special instruction in the instruction set which causes an interrupt when it is executed. It is a signal that request the processor to suspend the current execution and service the occurred interrupt.

For example, a divide-by-zero exception will be thrown if the processor's arithmetic logic unit is commanded to divide a number by zero as this instruction is in error and impossible. The operating system will catch this exception, and can choose to abort the instruction. Software interrupt instructions can also function similarly to subroutine calls and be used for a variety of purposes

SELF CHECK 6.8

1. Explain the interrupts in computer architecture.

6.9 PIPELINED COMPUTERS

The pipeline is a set of data processing elements that connected in series. The output of one element is the input of the next one. The elements of a pipeline are often executed in parallel or in time-sliced fashion where some amount of buffer storage is often inserted between elements.

The instruction pipelines are the classic Reduced Instruction Set Computer (RISC) pipeline, which are used in central processing units (CPUs). It allows an overlapping execution of multiple instructions with the same circuitry. The circuitry is usually divided up into stages;

instruction decoding, arithmetic, and register fetching stages. Each stage processes one instruction at a time

SELF CHECK 6.9

1. Discuss the functions of pipelined.

6.10 REDUCED INSTRUCTION SET COMPUTER (RISC) AND CISC ARCHITECTURES

Reduced instruction set computer (RISC) and Complex instruction set computer (CISC) are the types of instruction set for the processors. These approaches increase the CPU performance. RISC makes hardware simpler by using an instruction set composed of a few basic steps for loading, evaluating, and storing operations just like a load command will load data, store command will store the data. It reduces the cycles per instruction at the cost of the number of instructions per program. CISC makes a single instruction does all loading, evaluating, and storing operations just like a multiplication command will do stuff like loading data, evaluating, and storing it, hence it's complex. It attempts to minimize the number of instructions per program but at the cost of increase in number of cycles per instruction. Earlier, processors with RISC architecture are having a smaller number of instructions as compared to processors with CISC architecture.

SELF CHECK 6.10

1. Explain the function of CISC.
2. Explain the function of RICS.

6.11 SECURITY AND PROTECTION

Computer security is the protection of computer systems from the theft or damage to the hardware, software or the information on them, as well as from disruption or misdirection of the services they provide. Security by design, or alternately secure by design, means that the software has been designed from the ground up to be secure. In this case, security is considered as a main feature.

The Open Security Architecture organization defines information technology (IT) security architecture as the design artifacts that describe how the security controls are positioned, and how they relate to the overall information technology architecture. These controls serve the purpose to maintain the system's quality attributes: confidentiality, integrity, availability, accountability and assurance services.

SELF CHECK 6.11

1. Define computer security and protection.

SUMMARY

In this topic you have learnt that:

The computer operation, interconnection of buses, addressing techniques, flags, condition codes, status registers, storage units, subroutine calls, interrupts, pipelined computers, RISC architecture, CISC architecture, security and protection.

KEY TERMS

System Bus	A bus is a high-speed internal connection. Buses are used to send control signals and data between the processor and other components.
Computer Architecture	Consists of rules and methods or procedures which describe the implementation, functionality of the computer systems.

REFERENCES

Stallings, W., (2019). *Computer Organization and Architecture Designing for Performance.* 11th ed. New York: Pearson.