Computer Architecture - Computer Architecture is a blueprint for design and implementation of a computer system. It provides the functional details and behaviour of a computer system and comes before computer organization. Computer architecture deals with 'What to do?'

Computer Organization - Computer Organization is how operational parts of a computer system are linked together. It implements the provided computer architecture. Computer organization deals with 'How to do?'

Computer Architecture	Computer Organization
Computer Architecture is concerned with the way hardware components are connected together to form a computer system.	Computer Organization is concerned with the structure and behaviour of a computer system as seen by the user.
It acts as the interface between hardware and software.	It deals with the components of a connection in a system.
Computer Architecture helps us to understand the functionalities of a system.	Computer Organization tells us how exactly all the units in the system are arranged and interconnected.
A programmer can view architecture in terms of instructions, addressing modes and registers.	Whereas Organization expresses the realization of architecture.
While designing a computer system architecture is considered first.	An organization is done on the basis of architecture.
Computer Architecture deals with high-level design issues.	Computer Organization deals with low-level design issues.
Architecture involves Logic (Instruction sets, Addressing modes, Data types, Cache optimization)	Organization involves Physical Components (Circuit design, Adders, Signals, Peripherals)

Design Principles for Modern Computers

There is a set of design principles, sometimes called the RISC design principles that architects of general-purpose CPUs do their best to follow:

- **All Instructions Are Directly Executed by Hardware**
  – eliminates a level of interpretation

- **Maximize the Rate at Which Instructions are Issued**
  – MIPS = millions of instructions per second
  – MIPS speed related to the number of instructions issued per second
  – Parallelism can play a role

- **Instructions Should be Easy to Decode**
  – a critical limit on the rate of issue of instructions
  – make instructions regular, fixed length, with a small number of fields.
  – the fewer different formats for instructions, the better.

- **Only Loads and Stores Should Reference Memory**
  – operands for most instructions should come from- and return to- registers.
  – access to memory can take a long time
  – thus, only LOAD and STORE instructions should reference memory.

- **Provide Plenty of Registers**
  – accessing memory is relatively slow, many registers (at least 32) need to be provided, so that once a word is fetched, it can be kept in a register until it is no longer needed.

Difference between Hardwired and Micro programmed Control Unit

The Hardwired and Microprogrammed control unit generates the **control signals** to fetch and execute instructions. The fundamental difference between hardwired and microprogrammed control unit is that hardwired is a **circuitry** approach whereas, the microprogram control unit is implemented by **programming**.

The hardwired control unit is designed for the **RISC** style instruction set. On the other hand, the microprogrammed control unit was designed for the **CISC** style instruction set.

These control units can be distinguished on the several parameters which we have discussed below. We will also discuss the design of both the control unit. So let us start with our discussion.

Basis of Differentiation	**Hardwired Control Unit**	**Microprogrammed Control Unit**
Basic	It is a circuitry approach.	This control unit is implemented by programming
Design	RISC style instructions	CISC style instructions
Modification	Modification is difficult as the control unit is hardwired. Modifying it will require the change in hardware.	Modifications are easy in case of microprogrammed control unit as it will require the in change in the code only.
Instructions	It works well for simple instructions.	It works well for complex instructions also.
Costing	Implementing hardwired structure requires a cost.	Implementing microprograms is not costly.
Control memory	No control memory is required	Control memory is required
Execution Speed	Faster execution	Comparatively slow

What is Control Unit?

Control Unit is the part of the computer's central processing unit (CPU), which directs the operation of the processor. It was included as part of the Von Neumann Architecture by John von Neumann. It is the responsibility of the Control Unit to tell the computer's memory, arithmetic/logic unit and input and output devices how to respond to the instructions that have been sent to the processor. It fetches internal instructions of the programs from the main memory to the processor instruction register, and based on this register contents, the control unit generates a control signal that supervises the execution of these instructions.

A control unit works by receiving input information to which it converts into control signals, which are then sent to the central processor. The computer's processor then tells the attached hardware what operations to perform. The functions that a control unit performs are dependent on the type of CPU because the architecture of CPU varies from manufacturer to manufacturer.

Functions of the Control Unit

1. It coordinates the sequence of data movements into, out of, and between a processor's many sub-units.
2. It interprets instructions.
3. It controls data flow inside the processor.
4. It receives external instructions or commands to which it converts to sequence of control signals.
5. It controls many execution units (i.e. ALU, data buffers and registers) contained within a CPU.
6. It also handles multiple tasks, such as fetching, decoding, execution handling and storing results.

Difference between Direct and Indirect Addressing Modes:

S.N.	Direct Addressing Mode	Indirect Addressing Mode
1.	Address field contains the effective address of operand	Address field contains reference of effective address
2.	Requires only one memory reference	Requires two memory references
3.	Fast addressing	Slower than direct addressing mode
4.	No further classification	Further classified into two categories
5.	No further calculation is required to perform the operation	Require further calculation to find the effective address

System Bus Design

- The bus is a communication channel.
- The characteristic of the bus is shared transmission media.
- The limitation of a bus is only one transmission at a time.
- A bus which is used to provide communication between the major components of a computer is called a **System bus**.

System bus contains 3 categories of lines used to provide the communication between the CPU, memory and IO named as:

1. Address Lines:
- Used to carry the address to memory and IO.
- Unidirectional.
- Based on width of an address bus we can determine the capacity of a main memory

2. Data Lines:
- Used to carry the binary data between the CPU, memory and IO.
- Bidirectional.
- Based on the width of a data bus we can determine the word length of a CPU.
- Based on the word length we can determine the performance of a CPU.

3. Control Lines:
- Used to carry the control signals and timing signals
- A control signal indicates type of operation.
- Timing Signals used to synchronize the memory and IO operations with a CPU clock.

Direct Memory Access Advantages and Disadvantages

Advantages:

1. Transferring the data without the involvement of the processor will **speed up** the read-write task.
2. DMA **reduces the clock cycle** requires to read or write a block of data.
3. Implementing DMA also **reduces the overhead** of the processor.

Disadvantages

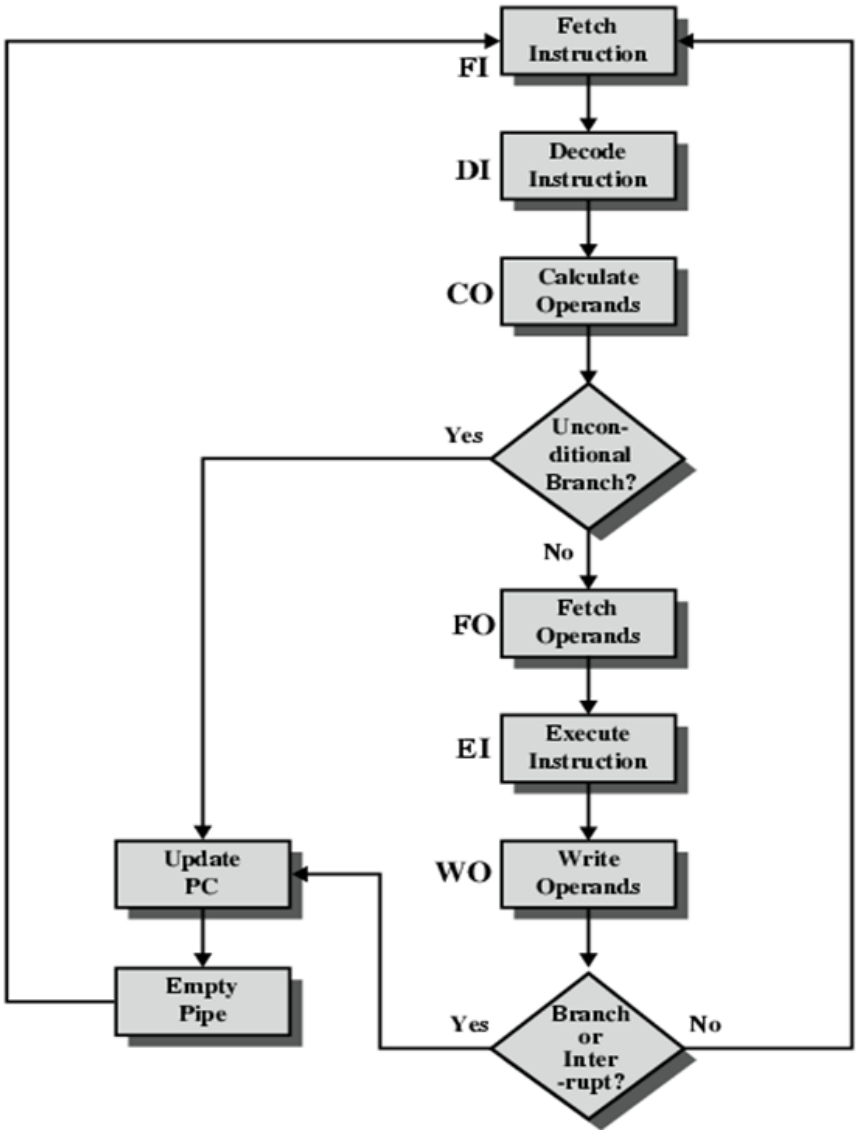
1. As it is a hardware unit, it would **cost** to implement a DMA controller in the system.
2. Cache **coherence** problem can occur while using DMA controller.

Explain six-stage instruction pipeline with suitable diagram

A typical instruction cycle can be split into many sub cycles like Fetch instruction, Decode instruction, Execute and Store. The instruction cycle and the corresponding sub cycles are performed for each instruction. These sub cycles for different instructions can thus be interleaved or in other words these sub cycles of many instructions can be carried out simultaneously, resulting in reduced overall execution time. This is called instruction pipelining.

The more are the stages in the pipeline, the more the throughput is of the CPU.

If the instruction processing is split into six phases, the pipelined CPU will have six different stages for the execution of the sub phases.



Fetch instruction: Instructions are fetched from the memory into a temporary buffer before it gets executed.

Decode instruction: The instruction is decoded by the CPU so that the necessary op codes and operands can be determined.

Calculate operand: Based on the addressing scheme used, either operands are directly provided in the instruction or the effective address has to be calculated.

Fetch Operand: Once the address is calculated, the operands need to be fetched from the address that was calculated. This is done in this phase.

Execute Instruction: The instruction can now be executed.

Write operand: Once the instruction is executed, the result from the execution needs to be stored or written back in the memory.

Peripherals Devices in Computer Organization

A **Peripheral Device** is defined as the device which provides input/output functions for a computer and serves as an auxiliary computer device without computing-intensive functionality.

Generally peripheral devices, however, are not essential for the computer to perform its basic tasks, they can be thought of as an enhancement to the user's experience. A peripheral device is a device that is connected to a computer system but is not part of the core computer system architecture. Generally, more people use the term peripheral more loosely to refer to a device external to the computer case.

Classification of Peripheral devices: It is generally classified into 3 basic categories which are given below:

1. **Input Devices:** The input devices are defined as it converts incoming data and instructions into a pattern of electrical signals in binary code that are comprehensible to a digital computer. **Example:** Keyboard, mouse, scanner, microphone etc.

2. **Output Devices:** An output device is generally reverse of the input process and generally translating the digitized signals into a form intelligible to the user. The output device is also performed for sending data from one computer system to another. For some time punched-card and paper-tape readers were extensively used for input, but these have now been supplanted by more efficient devices. **Example:** Monitors, headphones, printers etc.

3. **Storage Devices:** Storage devices are used to store data in the system which is required for performing any operation in the system. The storage device is one of the most requirement devices and also provides better compatibility. **Example:** Hard disk, magnetic tape, Flash memory etc.

Advantage of Peripherals Devices: Peripheral devices provide more feature due to this operation of the system is easy. These are given below:

- It is helpful for taking input very easily.
- It is also provided a specific output.
- It has a storage device for storing information or data
- It also improves the efficiency of the system.

RISC Processor

RISC stands for **Reduced Instruction Set Computer Processor**, a microprocessor architecture with a simple collection and highly customized set of instructions. It is built to minimize the instruction execution time by optimizing and limiting the number of instructions. It means each instruction cycle requires only one clock cycle, and each cycle contains three parameters: fetch, decode and execute. The RISC processor is also used to perform various complex instructions by combining them into simpler ones. RISC chips require several transistors, making it cheaper to design and reduce the execution time for instruction.

Examples of RISC processors are SUN's SPARC, PowerPC, Microchip PIC processors, RISC-V.

Advantages of RISC Processor

1. The RISC processor's performance is better due to the simple and limited number of the instruction set.
2. It requires several transistors that make it cheaper to design.
3. RISC allows the instruction to use free space on a microprocessor because of its simplicity.
4. RISC processor is simpler than a CISC processor because of its simple and quick design, and it can complete its work in one clock cycle.

Disadvantages of RISC Processor

1. The RISC processor's performance may vary according to the code executed because subsequent instructions may depend on the previous instruction for their execution in a cycle.
2. Programmers and compilers often use complex instructions.
3. RISC processors require very fast memory to save various instructions that require a large collection of cache memory to respond to the instruction in a short time.

CISC Processor

The CISC Stands for **Complex Instruction Set Computer**, developed by the Intel. It has a large collection of complex instructions that range from simple to very complex and specialized in the assembly language level, which takes a long time to execute the instructions. So, CISC approaches reducing the number of instruction on each program and ignoring the number of cycles per instruction. It emphasizes to build complex instructions directly in

the hardware because the hardware is always faster than software. However, CISC chips are relatively slower as compared to RISC chips but use little instruction than RISC. Examples of CISC processors are VAX, AMD, Intel x86 and the System/360.

Characteristics of CISC Processor

Following are the main characteristics of the RISC processor:

1. The length of the code is shorts, so it requires very little RAM.
2. CISC or complex instructions may take longer than a single clock cycle to execute the code.
3. Less instruction is needed to write an application.
4. It provides easier programming in assembly language.
5. Support for complex data structure and easy compilation of high-level languages.
6. It is composed of fewer registers and more addressing nodes, typically 5 to 20.
7. Instructions can be larger than a single word.
8. It emphasizes the building of instruction on hardware because it is faster to create than the software.

Advantages of CISC Processors

1. The compiler requires little effort to translate high-level programs or statement languages into assembly or machine language in CISC processors.
2. The code length is quite short, which minimizes the memory requirement.
3. To store the instruction on each CISC, it requires very less RAM.
4. Execution of a single instruction requires several low-level tasks.
5. CISC creates a process to manage power usage that adjusts clock speed and voltage.
6. It uses fewer instructions set to perform the same instruction as the RISC.

Disadvantages of CISC Processors

1. CISC chips are slower than RSIC chips to execute per instruction cycle on each program.
2. The performance of the machine decreases due to the slowness of the clock speed.
3. Executing the pipeline in the CISC processor makes it complicated to use.
4. The CISC chips require more transistors as compared to RISC design.
5. In CISC it uses only 20% of existing instructions in a programming event.

RISC	**CISC**
It is a Reduced Instruction Set Computer.	It is a Complex Instruction Set Computer.
It emphasizes on software to optimize the instruction set.	It emphasizes on hardware to optimize the instruction set.
It is a hard wired unit of programming in the RISC Processor.	Microprogramming unit in CISC Processor.
It requires multiple register sets to store the instruction.	It requires a single register set to store the instruction.
RISC has simple decoding of instruction.	CISC has complex decoding of instruction.
Uses of the pipeline are simple in RISC.	Uses of the pipeline are difficult in CISC.
It uses a limited number of instruction that requires less time to execute the instructions.	It uses a large number of instruction that requires more time to execute the instructions.
It uses LOAD and STORE that are independent instructions in the register-to-register a program's interaction.	It uses LOAD and STORE instruction in the memory-to-memory interaction of a program.
RISC has more transistors on memory registers.	CISC has transistors to store complex instructions.
The execution time of RISC is very short.	The execution time of CISC is longer.
RISC architecture can be used with high-end applications like telecommunication, image processing,	CISC architecture can be used with low-end applications like home automation, security system, etc.

video processing, etc.	
It has fixed format instruction.	It has variable format instruction.
The program written for RISC architecture needs to take more space in memory.	Program written for CISC architecture tends to take less space in memory.
Example of RISC: ARM, PA-RISC, Power Architecture, Alpha, AVR, ARC and the SPARC.	Examples of CISC: VAX, Motorola 68000 family, System/360, AMD and the Intel x86 CPUs.

Difference between Programmed and Interrupt Initiated I/O:

Programmed I/O	**Interrupt Initiated I/O**
Data transfer is initiated by the means of instructions stored in the computer program. Whenever there is a request for I/O transfer the instructions are executed from the program.	The I/O transfer is initiated by the interrupt command issued to the CPU.
The CPU stays in the loop to know if the device is ready for transfer and has to continuously monitor the peripheral device.	There is no need for the CPU to stay in the loop as the interrupt command interrupts the CPU when the device is ready for data transfer.
This leads to the wastage of CPU cycles as CPU remains busy needlessly and thus the efficiency of system gets reduced.	The CPU cycles are not wasted as CPU continues with other work during this time and hence this method is more efficient.
CPU cannot do any work until the transfer is complete as it has to stay in the loop to continuously monitor the peripheral device.	CPU can do any other work until it is interrupted by the command indicating the readiness of device for data transfer
Its module is treated as a slow module.	Its module is faster than programmed I/O module.
It is quite easy to program and understand.	It can be tricky and complicated to understand if one uses low level language.
The performance of the system is severely degraded.	The performance of the system is enhanced to some extent.

Characteristics of Multiprocessors

- A multiprocessor system is an interconnection of two or more CPUs with memory and input-output equipment.
- The term "processor" in multiprocessor can mean either a central processing unit (CPU) or an input-output processor (IOP).
- Multiprocessors are classified as multiple instruction stream, multiple data stream (MIMD) systems
- Multiprocessing improves the reliability of the system.
- The benefit derived from a multiprocessor organization is an improved system performance.
- Multiple independent jobs can be made to operate in parallel.
- A single job can be partitioned into multiple parallel tasks.
- Multiprocessing can improve performance by decomposing a program into parallel executable tasks.
- Multiprocessor are classified by the way their memory is organized: Shared memory and Distributed memory

Parallel Computer Architecture

In the last 50 years, there has been huge developments in the performance and capability of a computer system. This has been possible with the help of Very Large Scale Integration (VLSI) technology. VLSI technology allows a large number of components to be accommodated on a single chip and clock rates to increase. Therefore, more operations can be performed at a time, in parallel.

Parallel processing is also associated with data locality and data communication. **Parallel Computer Architecture** is the method of organizing all the resources to maximize the performance and the programmability within the limits given by technology and the cost at any instance of time.

Why Parallel Architecture?

Parallel computer architecture adds a new dimension in the development of computer system by using more and more number of processors. In principle, performance achieved by utilizing large number of processors is higher than the performance of a single processor at a given point of time.

Application Trends - With the advancement of hardware capacity, the demand for a well-performing application also increased, which in turn placed a demand on the development of the computer architecture.

Before the microprocessor era, high-performing computer system was obtained by exotic circuit technology and machine organization, which made them expensive. Now, highly performing computer system is obtained by using multiple processors, and most important and demanding applications are written as parallel programs. Thus, for higher performance both parallel architectures and parallel applications are needed to be developed.

Scientific and Engineering Computing - Parallel architecture has become indispensable in scientific computing (like physics, chemistry, biology, astronomy, etc.) and engineering applications (like reservoir modeling, airflow analysis, combustion efficiency, etc.). In almost all applications, there is a huge demand for visualization of computational output resulting in the demand for development of parallel computing to increase the computational speed.

Commercial Computing - In commercial computing (like video, graphics, databases, OLTP, etc.) also high speed computers are needed to process huge amount of data within a specified time. Desktop uses multithreaded programs that are almost like the parallel programs. This in turn demands to develop parallel architecture.

Technology Trends - With the development of technology and architecture, there is a strong demand for the development of high-performing applications. Experiments show that parallel computers can work much faster than utmost developed single processor. Moreover, parallel computers can be developed within the limit of technology and the cost.

The primary technology used here is VLSI technology. Therefore, nowadays more and more transistors, gates and circuits can be fitted in the same area. With the reduction of the basic VLSI feature size, clock rate also improves in proportion to it, while the number of transistors grows as the square. The use of many transistors at once (parallelism) can be expected to perform much better than by increasing the clock rate

Technology trends suggest that the basic single chip building block will give increasingly large capacity. Therefore, the possibility of placing multiple processors on a single chip increases.

What is Auxiliary memory (secondary storage)?

Definition: Auxiliary memory (also referred to as *secondary storage*) is the non-volatile memory lowest-cost, highest-capacity, and slowest-access storage in a computer system. It is where programs and data kept for long-term storage or when not in immediate use.

Such memories tend to occur in two types-sequential access (data must access in a linear sequence) and direct access (data may access in any sequence). The most common sequential storage device is the hard disk drives, whereas direct-access devices include rotating drums, disks, CD-ROMs, and DVD-ROMs. It used as permanent storage of data in mainframes and supercomputers.

Auxiliary memory may also refer to as auxiliary storage, secondary storage, secondary memory, external storage or external memory. **Auxiliary memory** is not directly accessible by the CPU; instead, it stores noncritical system data like large data files, documents, programs and other back up information that supplied to primary memory from

auxiliary memory over a high-bandwidth channel, which will use whenever necessary. Auxiliary memory holds data for future use, and that retains information even the power fails.

Characteristics of Auxiliary Memory are following:

- **Non-volatile memory** − Data is not lost when power is cut off.
- **Reusable** − The data stays in the secondary storage on permanent basis until it is not overwritten or deleted by the user.
- **Reliable** − Data in secondary storage is safe because of high physical stability of secondary storage device.
- **Convenience** − With the help of a computer software, authorized people can locate and access the data quickly.
- **Capacity** − Secondary storage can store large volumes of data in sets of multiple disks.
- **Cost** − It is much lesser expensive to store data on a tape or disk than primary memory.

Expression Parsing

The way to write arithmetic expression is known as a **notation**. An arithmetic expression can be written in three different but equivalent notations, i.e., without changing the essence or output of an expression. These notations are named as how they use operator in expression.

Infix Notation - We write expression in **infix** notation, e.g. a - b + c, where operators are used **in**-between operands. It is easy for us humans to read, write, and speak in infix notation but the same does not go well with computing devices. An algorithm to process infix notation could be difficult and costly in terms of time and space consumption.

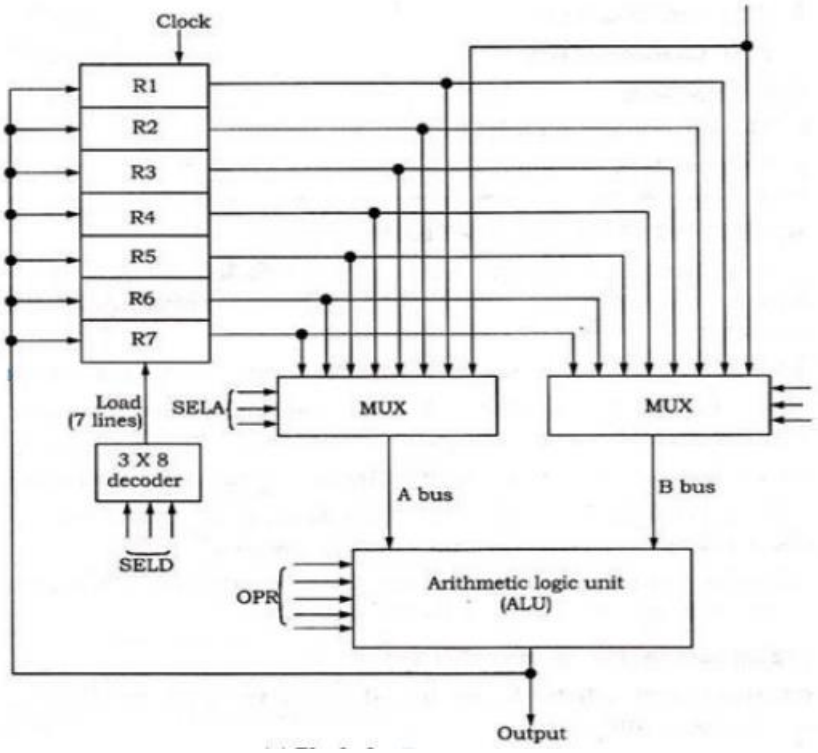
Prefix Notation - In this notation, operator is **prefix**ed to operands, i.e. operator is written ahead of operands. For example, **+ab**. This is equivalent to its infix notation **a + b**. Prefix notation is also known as **Polish Notation**.

Postfix Notation - This notation style is known as **Reversed Polish Notation**. In this notation style, the operator is **postfix**ed to the operands i.e., the operator is written after the operands. For example, **ab+**. This is equivalent to its infix notation **a + b**.

Discuss the bus organization for seven CPU registers with diagram.

- The output of each register is connected to two multiplexers (MUX) to form the two buses A and B.
- The selection lines in each multiplexer-select one register or the input data for the particular bus.
- The A and B buses form the inputs to a common ALU. The ALU determines the arithmetic or logic micro operation.
- The result of the microoperation is available for output data and also goes into the inputs of all 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, providing a transfer path between the data in the output bus and the inputs of the selected destination register.

  For example, to perform the operation: R1 <- R2 + R3
  MUX A selector (SELA): Bus A <- R2
  MUX B selector (SELB): Bus B <- R3
  ALU operation selector (OPR): ALU to ADD
  Decoder destination selector (SELD): R1 <- Out Bus


Clock
R1
R2
R3
R4
R5
R6
R7
Load
(7 lines)
SELA
MUX
MUX
3 X 8
decoder
A bus
B bus
SELD
OPR
Arithmetic logic unit
(ALU)
Output