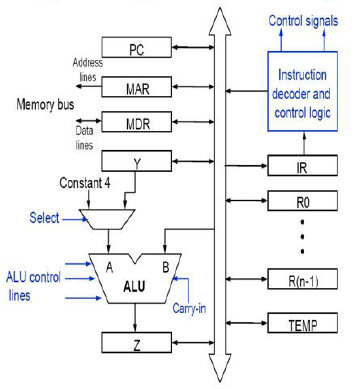
**single bus organization of data path**



Single bus organization:

* + ALU, control unit and all the registers are connected via a single common bus (Called Internal Bus)
  + Bus is internal to the processor and should not be confused with the external bus that connects the processor to the memory and I/O devices.

**Data lines** of the external memory bus are connected to the internal processor bus via MDR.

* + Register MDR has two inputs and two outputs.
  + Data may be loaded to (from) MDR from (to) internal processor bus or external memory bus.

**Address lines** of the external memory bus are connected to the internal processor bus via MAR.

* + MAR receives input from the internal processor bus.
  + MAR provides output to external memory bus.

**Instruction decoder and control logic block**, or control unit issues signals to control the operation of all units inside the processor and for interacting with the memory bus.

* + Control signals depend on the instruction loaded in the Instruction Register (IR)

Outputs from the control logic block are connected to:

* + Control lines of the memory bus.
  + ALU, to determine which operation is to be performed.
  + Select input of the multiplexer MUX to select between Register Y and constant 4.
  + Control lines of the registers, to select the registers.

**Registers Y, Z, and TEMP:**

* + Used by the processor for temporary storage during execution of some instructions.
  + Note that Registers R0 to R(n-1) are used to store data generated by one instruction for later use by another instruction.
  + Data is stored in R0 through R(n-1) after the execution of an instruction.

**Multiplexer MUX** selects either the output of register Y or a constant 4, depending upon the control input Select.

* + Constant 4 is used to increment the value of the PC.

B input of ALU is obtained directly from processor-bus.

As instruction execution progresses, data are transferred from one register to another, often passing through ALU to perform arithmetic or logic operation.

**Hardwired control Vs Microprogrammed control: DSCCFAADAIRC**

|  |  |  |
| --- | --- | --- |
| Definition | Hardwired control is a control mechanism to generate control signals by using gates, flip-flops, decoders, and other digital circuits | Microprogrammed control is a control mechanism to generate control signals using a memory called control store(CS) which contains the control signals |
| Speed | Hardwired control unit is faster when compared to microprogrammed control unit as the required control signals are generated with the help of hardwares | This is slower than the other as micro instructions are used for generating signals here |
| Cost | More costlier as everything has to be realized in terms of logic gates | Less costlier than hardwired control as only micro instructions are used for generating control signals |
| Control functions | Implemented in hardware | Implemented in software |
| Flexibility | Not flexible to accommodate new system specifications on new instructions, redesign is required | More flexible to accommodate new system specifications on new instructions |
| Ability to handle large or complex instruction sets | It cannot handle complex instructions as the circuit design for it becomes complex | It can handle complex instructions |
| Ability to support operating system & diagnostic features | Very difficult | Easy |
| Design process | Complicated | Orderly and systematic |
| Applications | Used in computer that makes use of Reduced Instruction Set Computers(RISC). | Used in computer that makes use of Complex Instruction Set Computers(CISC) |
| Instruction set size | Usually under 100instructions | Usually over 100instructions |
| ROM size | - | 2K to 10K by 20-400 bit micro instructions |
| Chip area | Uses least area | Uses more area |

**Steps involved in the execution of an instruction:**

Consider the instruction *Add (R3),R1*

Which adds he contents of a memory- location pointed by R3 to register R1.

Executing this instruction requires the following actions:

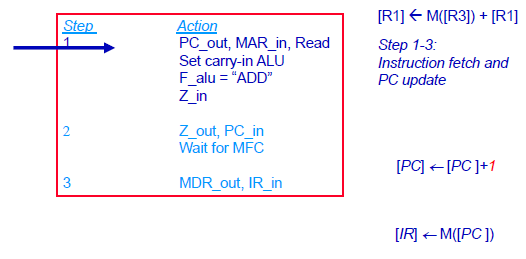
1)Fetch the instruction.

2)Fetch the first operand.

3)Perform the addition.

4)Load the result into R1.

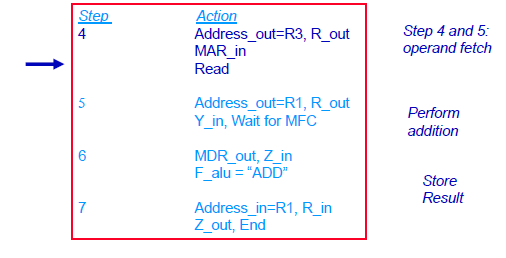
Instruction execution proceeds as follows: R1← R1 + [R3]

****

**Step1**:The instruction-fetch operation is initiated by loading contents of PC into MAR & sending a Read request to memory. The Select signal is set to Select4, which causes the Mux to select constant 4. This value is added to operand at input B (PC‟s content), and the result is stored in Z

**Step2:** Updated value in Z is moved to PC.

**Step3:** Fetched instruction is moved into MDR and then to IR.

****

**Step4**: Contents of R3 are loaded into MAR & a memory read signal is issued.

**Step5:** Contents of R1 are transferred to Y to Prepare for addition.

**Step6:** When Read operation is completed, memory-operand is available in MDR, and the

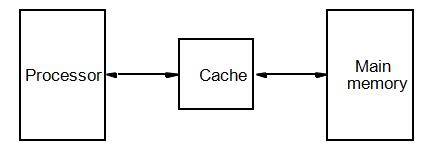
addition is performed.

**Step7:** Sum is stored in Z, then transferred to R1.

The End signal causes a new instruction fetch cycle to begin by returning to step1.

More explanation w.r.t single bus organization

**Cache Hit & Miss:**

****

***Cache has a replica of the contents of the main memory.***

***Contents of the cache and the main memory may be updated simultaneously.***

The performance of cache memory is measured in terms of a quantity called **hit ratio**.

When the CPU refers to memory and finds the word in cache it is said to produce a **hit**.

If the word is not found in cache, it is in main memory then it counts as a **miss**.

**Types of Read-Only Memory (ROMs):**

Data are written into a ROM when it is manufactured.

* **Programmable Read-Only Memory (PROM):**
  + Allow the data to be loaded by a user.
  + Process of inserting the data is irreversible.
  + Storing information specific to a user in a ROM is expensive.
  + Providing programming capability to a user may be better.
* **Erasable Programmable Read-Only Memory (EPROM):** 
  + Stored data to be erased and new data to be loaded.
  + Flexibility, useful during the development phase of digital systems.
  + Erasable, reprogrammable ROM.
  + Erasure requires exposing the ROM to UV light.
* **Electrically Erasable Programmable Read-Only Memory (EEPROM):**
  + To erase the contents of EPROMs, they have to be exposed to ultraviolet light.
  + Physically removed from the circuit.
  + EEPROMs the contents can be stored and erased electrically.
* **Flash memory:**
  + Has similar approach to EEPROM.
  + Read the contents of a single cell, but write the contents of an entire block of cells.
  + Flash devices have greater density.
    - Higher capacity and low storage cost per bit.

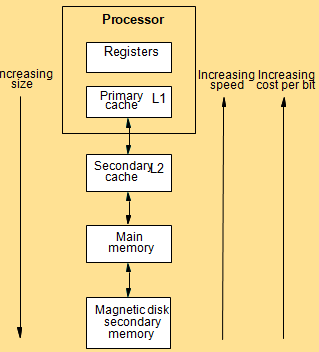
Power consumption of flash memory is very low, making it attractive for use

in equipment that is battery-driven

**Memory Hierarchy:**

****

**Or**

****

* Fastest access is to the data held in processor registers. Registers are at the top of the memory hierarchy.
* Relatively small amount of memory that can be implemented on the processorchip. This is processor cache.
* Two levels of cache. Level 1 (L1) cache is on the processor chip. Level 2 (L2) cache is in between main memory and processor.
* Next level is main memory. Much larger, but much slower than cache memory.
* Next level is magnetic disks. Huge amount of inexepensive storage.