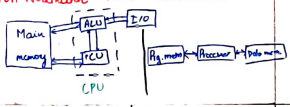
Processor Organization & Architecture

Von Neumann Archit / Harvard Archit.



- single mem. to be shared by both code & data
- processor need to fetch code in separate clock cycle & data in another clock cycle ⇒ request 2 clock cycles
- high speed → less time consumming
- simple design

- separate mem. to be shared by both code & data
- single clock cycle is sufficient as separate buses are used one used to access code & data
- slower speed → more time consumming
- complex design

Instruction Register (IR)

- use to hold current instructions
- contents of IR available to CU which generates timing signal that control various processing elements involved in executing the instructions

Program Counter (PC)

- always contains address of next instruction to be executed

Stack pointer (SP)

- used to point to the top activation record on the runtime stack
- runtime stack contains one activation record for each function that is currently unfinished in the program

Accumulator (AC)

- general purpose register
- specifically used to accumulate result of the currently running instructions.

Memory Address Register (MAR)

- handles data transfer between main memory & processor
- holds the address of main memory to or from which data is to be transferred

Memory Data Register (MDR)

- handles data transfer between main memory & processor
- contains the data to be written into or read from addressed word of main memory

_____ × _____ × _____

Addressing Modes

- ≠ ways in which the location of an operand can be specified in an instruction

- Immediate Mode
- Register mode
- Absolute mode
- Indirect mode
- Index Mode
- Base with index
- BWI & register
- Relative
- Auto increment
- Auto decrement

CPU Ao

- divided in 4 grps:
  1) Registers
  2) ALU
  3) Interrupt Control
  4) Timing & Control Circuitry.

- consist of PIPO
- scratch-pad memory → stores data & address of memory
- archit. of micro computer depends on no & types of registers used in microprocessor
- Register are classified
  - a) temporary registers
  - b) general purpose R
  - c) special purpose R

Instruction Formats

- an instruction is normally made up of a combination of an opcode and some ways of specifying an operand by its location or addressing memory

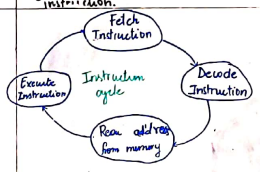
opcode	mode	Address or operand

Reverse Polish Notation

- postfix notation
- method for representing expression in which the operator symbol is placed after arg. being operated on
- eg: 5 : 2 3 +

Instruction cycle

- program in the memory unit of computer consists of a sequence of instructions.
- instructions are executed by processor by going through a cycle for each instruction.



Interrupt Cycle

- in instruction cycle, interrupt cycle is the first part
- occurs at random times during the execution of a program in response to signal from hardware

Type	Mean of communication
Interrupt	the CPU & OS Kernel
Signals	the OS kernel & OS process

Fetch Cycle

- CPU fetch some data or instructions from main memory when stored it into its internal memory → registers
  ⇒ fetch cycle

Application of microprocessing

- in realization of control unit
- in OS
- in high level language support
- in micro diagnostics
- in user tailoring
- in emulation

Pipeline Processing

Pipeline:- divide the execution of instructions into multiple steps
- overlap the exec. of ≠ instructions

into ≠ stages.

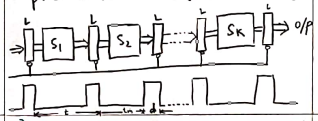
non pipeline

1) single cycle implementation
- the cycle time depends on the slowest instructions
- every instruction takes the same amount of time

II) Multicycle implementation
- divide the execution of instructions into multiple steps.
- each instruction may take variable no. of steps (clock cycles)

Asynchronous & Synchronous Linear pipelining

I) Synchronous Pipeline
- all transfer simultaneously
- one task / operation enter pipeline / cycle
- processor reservation table → diagonal



II) Asynchronous Pipeline
- Transfers performed when individual processors are ready
- Handshaking protocols between processors
- mainly used in multiprocessor systems with message-passing

Pipeline Hazards

- Instruction Hazards occur when instruction read and write register that are used by other instructions.

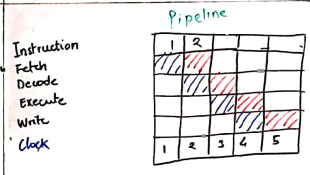
I) Structural Hazard (Resource conflicts)
- These hazards are caused when two instructions access the memory at the same time

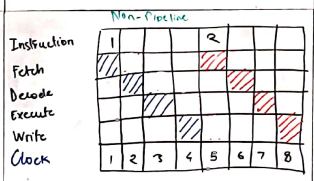
II) Data Hazards (Data dependency)
- The Hazard arises when an instruction depends on the result of a previous instruction but the result is not yet available

III) RAR Hazard (Control Hazard)
- RAR Hazard occurs when two instructions, both read from the same register. This hazard does not cause a problem for the processor because reading a register does not change register's value

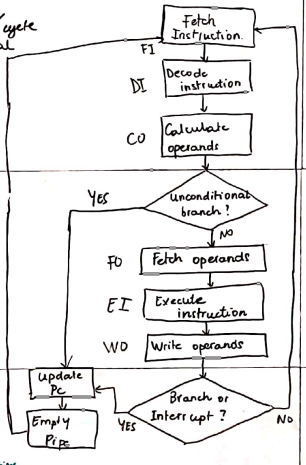
Pipeline Stalls

- The hardware inserts a special instruction → NOP
- hardware interlocks ← method.





Six stage Instruction Pipeline



Fetch Instruction

Instructions → fetched from memory → temporary buffer before it gets executed

Decode instruction

- instruction decoded by CPU so that the necessary opcodes & operands can be determined

Calculate operand

It is based on addressing scheme used, either operands are directly provided in the instructions or the effective address has to be calculated

Execute instruction

The instruction can now be executed

Write operand

Once the instruction executed, result from execution needs to be stored or return back to the memory.