

# Elements of Machine Instruction

- ① Operation code
- ② Source Operand reference
- ③ Result Operand reference
- ④ Next Instruction reference

① OPERATION CODE: - The Code that is used to specify the operations that will be going to be performed. It is in form of binary numbers (OPcodes).

② SOURCE OPERAND REFERENCE: This are the input for the process and it will be one or more than one operation.

③ Result OPERAND Reference: Produces a result.

④ NEXT INSTRUCTION REFERENCE: Tells the processor where to send the next instruction after the execution of the current instruction is complete.

\* Source and Result Operand can be in one of the following three areas

i) main or virtual memory

ii) Processor register

iii) Input / output device

## INSTRUCTION TYPES

Instruction can be categorized as follows:

- i) Data Processing: Arithmetic or Logic Instruction
- ii) Data Storage: Memory Instruction
- iii) Data Movement: I/O Instructions
- iv) Control: Test and branch

\* Arithmetic Instructions Provide Computational Capability for processing data.

\* Logic (Boolean) Instructions Operate on the number of bits of a word as bits rather than as numbers.

\* Memory Instructions: These operations are performed primarily on data in processor register

\* I/O Instructions: They are needed to transfer program and data into memory and the result of computation back to the user.

\* Test Instructions: They are used to test the value of a dataword or the status of a computation.

\* Branching Instructions: Are used to branch to a different set of instruction depending on the decision made

\* BRANCHING IS a process of taking one of two or more alternative Paths of Computation

# INSTRUCTION SET DESIGN

One of the most interesting and most analyzed aspect of Computer design is instruction set design.

The design of an instruction set is very complex because it affect so many aspect of the computer system.

The instruction set is the programmer's means of controlling the processor. Therefore, programmers requirements must be considered when designing the instruction set.

\* The most important fundamental design issues include the following

- 1) Operation repertoire: This indicates how many and which operation to provide and how complex operation should be.
- 2) Data types: This refers to various types of data upon which operations are performed.
- 3) Instruction format: This contains, instruction length in bits, number of address, sizes of various field
- 4) Registers: This refers to the number of registers which can be referenced by instruction and how they are used.
- 5) Addressing: This refers to the modes by which the address of an operand is specified.

## WRITE-BACK CACHE & WRITE-THROUGH CACHE

Write-through, i. data is simultaneously updated to Cache and memory

Write-back, i. data is updated only in the cache and updated into the memory at a later time

### Differences

#### WRITE THROUGH

- i) Both main memory and cache memory is updated simultaneously with every memory write operation
- ii) Main memory always contains same data as cache
- iii) Number of memory write operation in a typical program is more

#### WRITE BACK

- i) Only the cache location is updated during write operation
- ii) Main memory & cache may have different data
- iii) Number of memory write operation is less

\* **VIRTUAL MEMORY** - A storage allocation scheme in which secondary memory can be addressed as though it were part of the main memory

\* **PAGE TABLE** - A data structure used by a virtual memory in a computer O.S to store the mapping between virtual addresses and physical addresses. i.e. It defines the mappings from virtual addresses to physical ones.

\* LOOKASIDE BUFFERS acts as a cache for the pagetable

Translation Lookaside buffer is a memory cache that stores the recent translations of virtual memory to physical memory address for faster

\* Difference Between Virtual Address And Physical Address

Virtual Address	Physical Address
i) Accessed during process execution	i) Address where instruction resides.
ii) Produced by C.P.U	ii) Computed by memory management unit (MMU)
iii) Helps to obtain the physical address	iii) Helps to identify a location in the main memory
iv) It is viewable	iv) Not viewable
v) Logical address space	v) Physical address space
vi) Used to access physical address	vi) NOT directly accessed
vii) It ranges	vii) It is constant
viii) Not physically exist	viii) Physically exist.

\* How Virtual memory System interacts with Cache memory

\* Page 247, 248, 10.1, 10.2, 10.5

\* Why do vast Majority of Systems use Virtual and Physical pages that are the same size?

## Chapter 20 CONTROL UNIT OPERATION

Things needed to specify the function of a processor emerges!

- ① Operations (opcode)
- ② Addressing modes
- ③ Registers
- ④ I/O module interface
- ⑤ Memory module interface
- ⑥ Interrupt.

### Micro Operations

These are the functional, or atomic operations of a processor.

They are detailed low-level instructions used in some designs to implement complex machine instructions

#### \* THE FETCH CYCLE

It occurs at the beginning of each instruction cycle and causes an instruction to be fetched from memory

Four registers are involved in fetch cycle

- i) Memory Address register (MAR) : It specifies the address in memory for a read or write operation
- ii) Memory Buffer register (MBR) : Contains the last value read from memory
- iii) Program Counter (PC) : Holds the address of the next instruction to be fetched
- iv) Instruction Register (IR) : Holds the last instruction fetched

#### \* THE INDIRECT CYCLE

#### \* THE INTERRUPT CYCLE

#### \* THE EXECUTE CYCLE

#### \* THE INSTRUCTION CYCLE

## ADDRESSING MODE

An addressing mode refers to different ways in which a source operand is denoted in an instruction.

The most common addressing techniques

- ① Immediate    ② Direct    ③ Indirect    ④ Register
- ⑤ Register-Indirect    ⑥ Displacement    ⑦ Stack.

\* **Immediate Addressing :-** The operand is directly included in the instruction e.g.  $MOV \text{ EAX}, 10$

\* **Register Addressing :-** A register containing the operand. Depending upon the instruction, the register may be the first operand, the second operand or both.

e.g.  $MOV \text{ DX}, \text{TAX\_RATE}$ ; Register in first operand  
 $MOV \text{ Count}, \text{ CX}$ ; : = second operand  
 $MOV \text{ EAX}, \text{ EBX}$ ; Both operands are in registers

\* **Displacement Addressing :-** If combines the capability of direct addressing and register indirect addressing

$$EA = A + (R)$$

## CHAPTER 20 REVIEW Question

20.1: Explain the distinction between the written sequence and the time sequence of an instruction.

ANS: This sequence of instruction cycles is not necessarily the same as the written sequence of instructions that makes up the program, because of the existing branching instructions. The actual execution of instructions follows a time sequence of instruction.

20.2: Relationship Between Instructions & Micro-Operations

A micro-operation is an elementary C.P.U operation performed during one clock pulse. An instruction consists of a sequence of micro-operations.

20.3: Overall function of a Processor Control unit

- 1) It causes the processor to execute micro-operations in the proper sequence, determined by the program being executed.
- 2) It generates the control signals that cause each micro-operation to be executed.

20.4: Outline a three-step process that leads to a characterization of the control unit.

- ANS:
- 1) Define the basic elements of the processor
  - 2) Describe the micro-operation that the processor performs
  - 3) Determine the functions that the control unit must perform to cause the micro-operation to be performed

20.5: Provide a typical list of the inputs and outputs of a control unit.

ANS: The inputs

- i) Clock: This is how the control unit keeps time
- ii) Instruction register
- iii) Flags
- iv) Control signals from bus

The Outputs

- i) Control signals within the processor
- ii) Control signals to control bus : to memory & I/O module

20.6: What basic task does a Control unit perform?

- i) Sequencing: It causes the processor to step through a series of micro-operations in the proper sequence, based on the program being executed
- ii) Execution: The control unit causes each micro-operation to be performed.

20.7: List three types of Control Signal

- ANS:
- i) Those that activate an ALU function
  - ii) Those that activate a data path
  - iii) Those that are signals on the external system bus or other external interface.

20.8: Briefly explain what is meant by a hardwired implementation of a control unit

ANS: In a hardwired implementation, the control unit is essentially a combinational circuit. Its input logic signal are transformed into a set of output logic signals, which are the control signals.