Computer Organization and Embedded System

Control and Central Processing Units (ACtE0401)

Ishwar Kumar Singh
Computer Engineer
Government of Nepal

4.1 Control and central processing units (ACtE0401) :

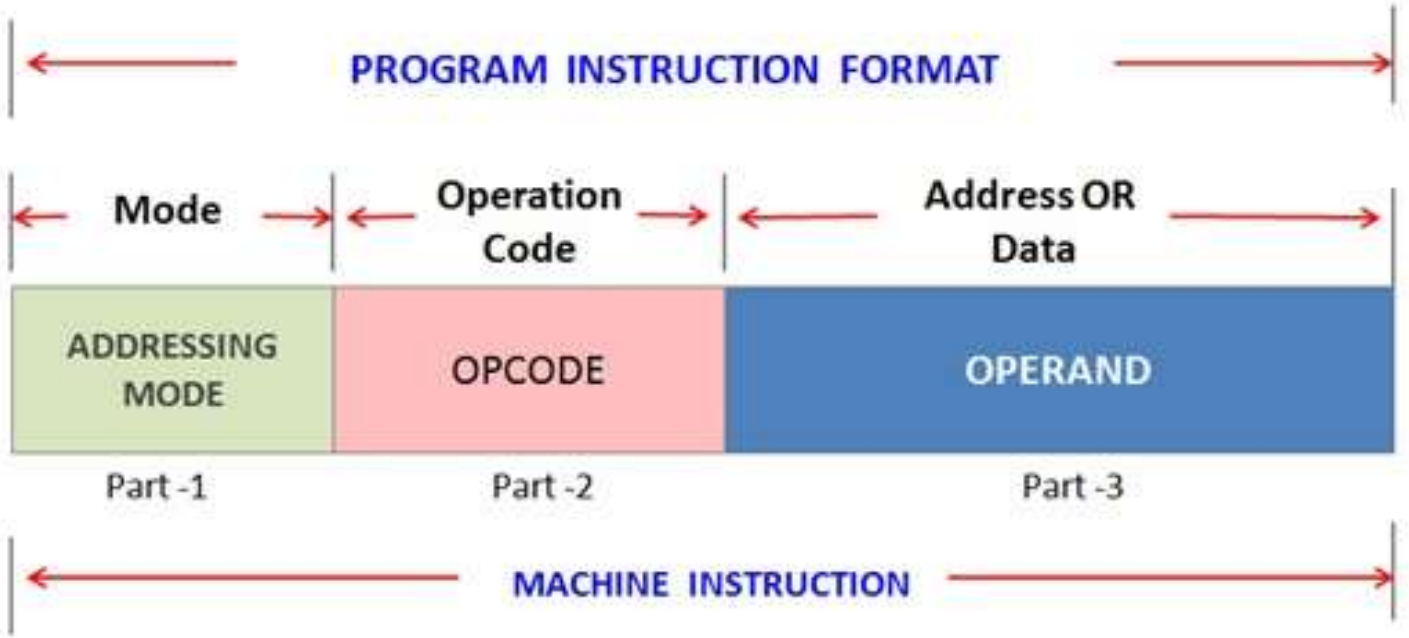
- Control Memory,
- Addressing Sequencing,
- Computer Configuration,
- Microinstruction Format,
- Design of Control Unit,
- CPU Structure and Function,
- Arithmetic and Logic Unit,
- Instruction Formats,
- Addressing Modes,
- Data Transfer and Manipulation,
- RISC and CISC Pipelining
- Parallel Processing.

Instruction Formats

- A computer usually has a variety of Instruction Code Formats.
- It is the function of the control unit within the CPU to interpret each instruction code and provide the necessary control functions needed to process the instruction.
- An ‘n’ bit instruction that has ‘k’ bits in the address field and ‘m’ bits in the operation code field come addressed ‘2^k’ location directly and specify ‘2^m’ different operation.
- The bits of the instruction are divided into groups called fields.
- The most common fields in instruction formats are: Operation Code, Address Field and Mode Field.

Instruction Formats

- An Operation code (Op-code) field that specifies the operation to be performed.
- An Address field that designates a memory address or a processor register.
- A Mode field that specifies the way the operand or the effective address is determined.



Types of Instruction

- Computers may have instructions of several different lengths containing varying number of addresses
- The number of address fields in the instruction format of a computer depends on the internal organization of its registers.
- Most computers fall into one of the three common types of CPU organizations:
  - **Single accumulator organization**
  - **Stack organization**
  - **General register organization**

Types of Instruction

- Based on the operation performed by the instructions, the instruction can be categorized into following types:
  - **Data Transfer Instruction**
  - **Data Manipulation Instruction**
    - **Arithmetic Instruction**
    - **Logical and Bit Manipulation Instruction**
    - **Shift Instruction**
  - **Program Control Instruction**

Single Accumulator Organization

- It was generally used in earlier machine with the implied address as a CPU register known as accumulator.
- With this type of instruction, the accumulator contains one of the operand and is used to store the result.
- The instruction used in this organization is known as One Address Instruction which uses an implied accumulator (AC) register for all data manipulation.
- All operations are done between the AC register and a memory operand.
- LOAD and STORE instructions are used for transferring the values to and from memory and Ac register.
- Format: Op X; Ac ← Ac Op X
- Example: MUL X; Ac ← Ac MUL X

Stack Organization

- The instruction in a stack computer consists of an operation code with no address field due to which the instructions are known as Zero Address Instruction
- This operation has the effect of popping the top numbers from the stack, operating the numbers and pushing the result into the stack. For example: ADD
- The name "Zero" address is given because of the absence of an address field in the computational instruction.
- Format: Op; TOS ← TOS OP (TOS-1)
- Example: DIV; TOS ← TOS DIV (TOS-1)

General Register Organization

- The instruction format in this type of computer needs more than one register address fields.
- Following are the types of instructions based on the number of address:
  - **Three address Instruction**
  - **Two address Instruction**

Three address Instruction

- With this type of instruction, each instruction specifies two operand location and a result location.
- A temporary location T is used to store some intermediate result so as not to alter any of the operand location.
- The three address instruction format requires a very complex design to hold the three address references.
- Format: Op X, Y, Z; X ← Y Op Z
- Example ADD X, Y, Z; X ← Y ADD Z
- ADVANTAGE: It results in short programs when evaluating arithmetic expressions.
- DISADVANTAGE: The instructions requires too many bits to addresses.

Two address Instruction

- Two-address instructions are the most common in commercial computers.
- With this type of instruction, each address field can specify either a processor register, or a memory word.
- One address must do double duty as both operand and result.
- The two address instruction format reduces the space requirement.
- To avoid altering the value of an operand, a MOV instruction is used to move one of the values to a result or temporary location T, before performing the operation.
- Format: Op X, Y; X ← X Op Y
- Example: SUB X, Y; X ← X SUB Y

Data Transfer Instruction

- Data transfer instructions cause transfer of data from one location to another without changing the binary information.
- The most common transfer are between the
  - Memory and Processor registers
  - Processor registers and input output devices
  - Processor registers themselves

Data Transfer Instructions	
Name	Mnemonic
Load	LD
Store	ST
Move	MOV
Exchange	XCH
Input	IN
Output	OUT
Push	PUSH
Pop	POP

Data Manipulation Instruction

- Data manipulation instructions perform operations on data and provide the computational capabilities for the computer. These instructions perform arithmetic, logic and shift operations.
- These instruction can be categorized into following types:
  - **Arithmetic Instruction**
  - **Logical and Bit Manipulation Instruction**
  - **Shift Instruction**

Arithmetic Instruction

- Arithmetic instructions perform basic mathematical operations such as Addition, Subtraction, Division, Multiplication.

Arithmetic Instructions	
Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with carry	ADDC
Subtract with borrow	SUBB
Negate	NEG

Logical and Bit Manipulation Instruction

- These instructions perform binary operations on the strings of bits stored in register.
- Useful for manipulating individual bits or group of bits that represents binary coed information.

Logical & Bit Manipulation Instructions	
Name	Mnemonic
Clear	CLR
Complement	COM
AND	AND
OR	OR
Exclusive OR	XOR
Clear Carry	CLRC
Set Carry	SETC
Enable Interrupt	EI
Disable Interrupt	DI

Shift Instruction

- These instructions are used to shift the content of an operand.
- Bits of a word are moved to the right or left as per the requirement.

Shift Instructions	
Name	Mnemonic
Logical Shift Right	SHR
Logical Shift Left	SHL
Arithmetic Shift Right	SHRA
Arithmetic Shift Left	SHLA
Rotate Right	ROR
Rotate Left	ROL
Rotate Right through Carry	RORC
Rotate Left through Carry	ROLC

Program Control Instruction

- The program control Instructions provide decision making capabilities and change the path taken by the program when executed in computer.
- These instructions specify conditions for altering the content of the program counter.
- The change in value of program counter as result of execution of program control instruction causes break in sequence of instruction execution.

Program Control Instructions	
Name	Mnemonic
Branch	BR
Jump	JMP
Skip	SKP
Call	CALL
Return	RET
Compare (by Subtraction)	CMP

Addressing Mode

- The way the operands are chosen during program execution is dependent on the addressing mode of instruction.
- It specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually referenced.
- Computers use addressing mode techniques for the purpose of accommodating the following purposes:
  - To give programming versatility to the user
  - To reduce the number of bits in the addressing field of the instructions.

Effective Address (EA)

- The effective address is defined to be the memory address obtained from the computation dictated by the given addressing mode.
- The effective address is the address of the operand in a computational-type instruction.

Types of Addressing Mode

- Implied Addressing Mode.
- Immediate Addressing Mode
- Register Direct Addressing Mode
- Register Indirect Addressing Mode
- Auto-increment or Auto-decrement Addressing Mode
- Direct Addressing Mode
- Indirect Addressing Mode
- Displacement Addressing Mode
- Relative Addressing Mode
- Index Addressing Mode
- Stack Addressing Mode

Implied Addressing Mode

- In this mode the operands are specified implicitly in the definition of the instruction.
- For example:- CMA - "complement accumulator" is an implied-mode instruction because the operand in the accumulator register is implied in the definition of the instruction.
- All the register reference instructions that use an accumulator are implied-mode instructions.
- Instruction Format: Opcode;
- Advantage: no memory reference.
- Disadvantage: limited operand

Immediate Addressing Mode

- In this mode, the operand is specified in the instruction itself.
- This instruction has an operand field rather than an address field.
- The operand field contains the actual operand to be used in conjunction with the operation specified in the instruction.
- These instructions are useful for initializing register to a constant value.
- Instruction Format: Opcode, Operand;
- For example MVI B, 50H;
- Advantage: no memory reference.
- Disadvantage: limited operand

Register Direct Addressing Mode

- In this mode, the operands are in registers that reside within the CPU.
- The particular register is selected from the register field in the instruction which contains the operand.
- Instruction Format: Opcode, Register; Register → Operand.
- For example MOV A, B;
- Advantage: no memory reference.
- Disadvantage: limited address space

Register Indirect Addressing Mode

- In this mode the instruction specifies a register in the CPU whose contents give the address of the operand in the memory.
- In other words, the selected register contains the address of the operand rather than the operand itself.
- To use register indirect mode instruction, the programmer must ensure that the memory address of the operand is placed in the processor register with a previous instruction.
- Instruction Format: Opcode, Register; Register → Memory → Operand.
- For example: LDAX B;
- Advantage: Large address space.
- Disadvantage: Extra memory reference

Direct Addressing Mode

- In this mode the effective address is equal to the address part of the instruction.
- The operand resides in memory and its address is given directly by the address field of the instruction.
- Instruction Format: Opcode, Address; Address → Operand.
- For example: LDA 4000H;
- Advantage: Simple.
- Disadvantage: limited address field

Indirect Addressing Mode

- In this mode the address field of the instruction gives the address where the effective address is stored in memory.
- Control unit fetches the instruction from the memory and uses its address part to access memory again to read the effective address.
- Instruction Format: Opcode, Address;
- Address → Memory → Operand.
- Advantage: Flexibility.
- Disadvantage: Complexity

Auto increment or Auto decrement Addressing Mode

- This is similar to register indirect mode except that the register is incremented or decremented after (or before) its value is used to access memory.
- When the address stored in the registers refers to a table of data in memory, it is necessary to increment or decrement the registers after every access to the table.
- This can be achieved by using the increment or decrement instruction.

Displacement Addressing Mode

- An addressing mode which combines the capabilities of direct addressing and register indirect addressing.
- The address field of instruction is added to the content of specific register in the CPU.
- Instruction Format: Opcode, Register, Address;
- Register → Address
- Effective Address= Value of Register + Address;
- Advantage: Flexibility.
- Disadvantage: Complexity

Relative Addressing Mode

- In this mode the content of the program counter (PC) is added to the address part of the instruction in order to obtain the effective address.
- The address part of the instruction is usually a signed number (either a +ve or a -ve number).
- When the number is added to the content of the program counter, the result produces an effective address whose position in memory is relative to the address of the next instruction.
- Effective Address (EA)= PC + A

Indexed Addressing Mode

- In this mode the content of an index register (XR) is added to the address part of the instruction to obtain the effective address.
- The index register is a special CPU register that contains an index value.
- If an index-type instruction does not include an address field in its format, the instruction is automatically converted to the register indirect mode of operation.
- Effective Address (EA)= XR+A

Base Register Addressing Mode

- In this mode the content of a base register (BR) is added to the address part of instruction to obtain the effective address.
- The base register addressing mode is used in computers to facilitate the relocation of programs in memory i.e. when programs and data are moved from one segment of memory to another.
- Effective Address (EA) = BR + A

Stack Addressing Mode

- The stack is the linear array of locations which is known as Last In First Out (LIFO). The stack pointer is maintained in register.
- Effective Address (EA)= TOS

RISC and CISC

- Important aspect of computer is the design of the instruction set for processor.
- Instruction set determines the way that machine language programs are constructed.
- Early computers were using simple and small instruction set due to the need to minimize the hardware used.
- Advent of IC brought the cheaper digital software and instruction indented to increase the number of complexity.

CISC

- A computer with a large number of instructions is classified as a complex instruction set computer (CISC).
- It simplifies the compilation and improve the overall computer performance.
- The essential goal of CISC architecture is to attempt to provide a single machine instruction for each statement that is written in a high-level language.
- Examples of CISC architecture are the DEC VAX computer and the IBM 370 computer. Other are 8085, 8086, 80x86 etc.

Characteristics of CISC

- A large number of instructions-typically from 100 to 250 instructions
- Some instructions that perform specialized tasks and are used infrequently
- A large variety of addressing modes typically from 5 to 20 different modes
- Variable-length instruction formats
- Instructions that manipulate operands in memory
- Reduced speed due to memory read/write operations

Characteristics of CISC

- Use of microprogram which is a special program in control memory of a computer to perform the timing and sequencing of the micro-operations - fetch, decode, execute etc.
- Major complexity in the design of microprogram
- No large number of registers. Single register is used as a set of general purpose and is low cost.

RISC

- A computer which uses fewer instructions and has simple construction due to which execution becomes much faster within the CPU without having to use memory as often is classified as a reduced instruction set computer (RISC).
- The essential goal of RISC architecture is to attempt the reduction of the execution cycle by simplifying the instruction set
- Small set of instructions which are mostly register to register operations and uses simple load/store operations for memory access

RISC

- Each operand is brought into register using load instruction,
- Computations are done among data in registers
- And results is transferred to memory using store instruction
- Simplify instruction set and encourages the optimization of register manipulation.

Characteristics of RISC

- Relatively few instructions
- Relatively few addressing modes
- Memory access limited to load and store instructions
- All operations are done within the registers of the CPU
- Fixed length, easily decoded instruction format
- Single-cycle instruction execution
- Hardwired rather than microprogrammed control

Characteristics of RISC

- A relatively large number of registers in the processor unit
- Use of overlapped register windows to speed-up procedure call and return
- Efficient instruction pipeline i.e. fetch, decode and execute overlap
- Compiler support for efficient translation of high-level language programs into machine language programs
- A large number of registers in the processing unit are sometimes associated with RISC processors.
- RISC processors are simpler than corresponding CISC processors, they can be designed more quickly.

Differences

S.N.	RISC	CISC
1	Simple instructions taking one cycle	Complex instructions taking multiple cycles
2	Only load and store memory references	Any instructions may reference memory
3	Heavily pipelined	Not/less pipelined
4	Multiple register sets	Single register set
5	Complexity is in compiler	Complexity is in micro-programming
6	Instructions executed by hardware	Instructions interpreted by micro programming
7	Fixed format instructions	Variable format instructions
8	Few instructions and modes	Large instructions and modes

Thank You.