COMPUTER INSTRUCTIONS

MANINDER KAUR
professormaninder@gmail.com

INSTRUCTION FORMAT

- An instruction format or instruction code is a group of bits used to perform a particular operation on the data stored in computer.
- Processor fetches an instruction from memory and decodes the bits to execute the instruction.
- Different computers may have their own instruction set.

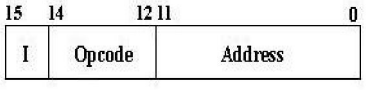
INSTRUCTION FORMAT

- Instruction code is divided into two parts namely operation code and address of data.
- Operation code consisting group of bits to define an operation such as add, subtract, multiply etc.

INSTRUCTION FORMAT

In an instruction format:
- First 12 bits (0-11) specify an address.
- Next 3 bits specify operation code (opcode).
- Left most bit specify the addressing mode I

I = 0 for direct address
I = 1 for indirect address



Instruction format

TYPES OF INSTRUCTIONS

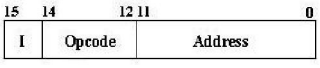
The basic computer has three 16-bit instruction code formats:

1. Memory Reference Instructions
2. Register Reference Instructions
3. Input/Output Instructions

Memory Reference Instructions

In Memory reference instruction:
- First 12 bits(0-11) specify an address.
- Next 3 bits specify operation code (opcode).
- Left most bit specify the addressing mode I

I = 0 for direct address
I = 1 for indirect address



Instruction format

(Opcode = 000 through 111)

Memory Reference Instructions

In Memory reference instruction:

- first 12 bits (0-11) specify an address.
- The address field is denoted by three x's (in hexadecimal notation) and is equivalent to 12-bit address.
- The last mode bit of the instruction represents by symbol I.
- When I = 0, the last four bits of an instruction have a hexadecimal digit equivalent from 0 to 6 since the last bit is zero (0).
- When I = 1 the last four bits of an instruction have a hexadecimal digit equivalent from 8 to E since the last bit is one (1).

Maninder Kaur
professormaninder@gmail.com

Memory Reference Instructions

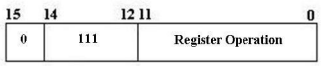
	Hexadecimal code		
Symbol	I = 0	I = 1	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	ADD memory word to AC
LDA	2xxx	Axxx	LOAD Memory word to AC
STA	3xxx	Bxxx	Store content of AC in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and Skip if zero

Maninder Kaur
professormaninder@gmail.com

Register Reference Instructions

In Register Reference Instruction:

- First 12 bits (0-11) specify the register operation.
- The next three bits equals to 111 specify opcode.
- The last mode bit of the instruction is 0.
- Therefore, left most 4 bits are always 0111 which is equal to hexadecimal 7.



Instruction format

Maninder Kaur
professormaninder@gmail.com

Register Reference Instructions

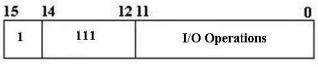
Symbol	Hexadecimal code	Description
CLA	7800	Clear AC
CLE	7400	Clear E
CMA	7200	Complement AC
CME	7100	Complement E
CIR	7080	Circulate right AC and E
CIL	7040	Circulate left AC and E
INC	7020	Increment AC
SPA	7010	Skip next instruction if AC positive
SNA	7008	Skip next instruction is AC is negative
SZA	7004	Skip next instruction is AC is 0
SZE	7002	Skip next instruction is E is 0
HLT	7001	Halt computer

Maninder Kaur
professormaninder@gmail.com

I/O Reference Instructions

In I/O Reference Instruction:

- First 12 bits (0-11) specify the I/O operation.
- The next three bits equals to 111 specify opcode.
- The last mode bit of the instruction is 1.
- Therefore, left most 4 bits are always 1111 which is equal to hexadecimal **F**.



Instruction format

Maninder Kaur
professormaninder@gmail.com

I/O Reference Instructions

Symbol	Hexadecimal code	Description
INP	F800	Input character to AC
OUT	F400	Output character from AC
SKI	F200	Skip on input flag
SKO	F100	Skip on Output flag
ION	F080	Interrupt on
IOF	F040	Interrupt off

Maninder Kaur
professormaninder@gmail.com