Microprocessor ,Assembly Language & Computer Interfacing Sessional

EEE-3212

Fareha Akter Juha
Adjunct Faculty, Dep. of CSE,
Leading University, Sylhet
Email: farehaakter02@gmail.com

Lab-2: 8085 Microprocessor Instruction Set.

Introduction of Instruction

An instruction is a binary pattern designed inside a microprocessor to perform a specified function. In another word, we can say An instruction is a command given task on specified data.

1. One is the task to be performed, called the “operation code” (OPCODE).
2. Second is the data to be operated on called the “OPERAND”.

The operand (or data) may include 8-bit or 16-bit data, on the internal register, a memory location, or an 8-bit or 16-bit address.

The **instruction set** is a collection of Instructions of the Microprocessor, that determines what functions that microprocessor can perform.

Types of the Instruction set of 8085 based on Operation Modes

1. **Data Transfer Instruction set of 8085**

In this group, the data in transfer,

- From register to register
- Between memory and register
- Between I/O and accumulator
- Load an 8-bit number in a register
- Load 16-bit number in a register pair.

Instruction Set	Meaning	Example	Addressing Mode	Bytes of Instruction
MOV r_1, r_2 $[r_1] \leftarrow [r_2]$	Move content of one register to another register	MOV A, B	Register addressing mode	1-byte
MOV r, M [r] ← [M] or [r] ← [[H,L]]	Move content of memory to register	MOV B, M	Register indirect addressing mode	1-byte
MOV M, r [M] ← [r] or [[H,L]] ← [r]	Move content of register to memory	MOV M, C	Register indirect addressing mode	1-byte
MOV r, data [r] ← data	Move immediate data to register	MOV A, 05H	Immediate addressing mode	2-byte
LXI rp, 16-bit data [rp] ← 16-bit data; [rh] ← 8 MSBs of data,[rl] ← 8 LSBs of data	Load register pair immediately	LXI H, 2800H, i.e. [L] ← [00],[H] ← [28]	Immediate addressing mode	3-byte
LDA address [A] ← [[address]]	Load accumulator direct	LDA 2400H	direct addressing mode	3-byte

STA address [[address]] ← [A]	store accumulator direct	STA 2000H	direct addressing mode	3-byte
LHLD address [L] ← [[address]], [H] ← [[address + 1]]	Load H-L pair direct	LHLD 2500H	direct addressing mode	3-byte
SHLD address [[address]] ← [L],[[address + 1]] ← [H]	store H-L pair direct	SHLD 2500H	direct addressing mode	3-byte
LDAX rp [A] ← [[rp]]	Load accumulator indirect	LDAX B	Register indirect addressing mode	1-byte
STAX rp [[rp]] ← [A]	store accumulator indirect	STAX D	Register indirect addressing mode	1-byte
XCHG [H-L] ↔ [D-E]	Exchange the content of H-L pair with D-E pair	XCHG	Register addressing mode	1-byte
MVI M, data [[H-L]] ← [data] or [M] ← [data]	Move immediate data to memory	LXI H , 2400H MVI M, 08 HLT	Register indirect addressing mode	2-byte

Types of the Instruction set of 8085 based on Operation Modes

2. Arithmetic Group Instruction set of 8085

In this group, the data is performed as addition, subtraction, increment (add 1), decrement (subtract 1), etc. The results of the arithmetic operations are stored in the accumulator, thus the previous content of the accumulator is altered. In the add operation, if the sum is larger than 8-bit, CY is set.

Instruction Set	Meaning	Example	Addressing Mode	Bytes of Instruction
ADD r [A]← [A] + [r]	Add register to the accumulator	ADD B	Register addressing mode	1-byte
ADC r [A]← [A] + [r] + [C]	Add register with carry to the accumulator	ADC D	Register addressing mode	1-byte
ADD M [A]← [A] + [M] or [A]← [A] + [[H-L]]	Add memory to accumulator	ADD M	Register indirect addressing mode	1-byte

Instruction Set	**Meaning**	**Example**	**Addressing Mode**	**Bytes of Instructio n**
ADC M [A]← [A] + [M] + [C] or [A]← [A] + [[H-L]] + [C]	Add memory with carry to accumulator	ADC M	Register indirect addressing mode	1-byte
ADI data [A]← [A] + data	Add immediate data to the accumulator	ADI 08H	Immediate addressing mode	2-byte
ACI data [A]← [A] + data + [C]	Add immediate data with carry to the accumulator	ACI 08H	Immediate addressing mode	2-byte
SUB r [A]← [A] – [r]	Subtract register from the accumulator	SUB B	Register addressing mode	1-byte
SBB r [A]← [A] – [r] – [C]	Subtract register from the accumulator with borrow	SBB B	Register addressing mode	1-byte

INR r [r]← [r] + [01]	Increment register content by 1	INR D	Register addressing mode	1-byte
DCR r [r]← [r] – [01]	Decrement register content by 1	DCR D	Register addressing mode	1-byte
DAD rp [[H-L]]← [[H-L]] + [rp]	Double addition register pair	DAD H	Register addressing mode	1-byte

Types of the Instruction set of 8085 based on Operation Modes

3. Logical Group Instruction set of 8085

The instruction set of this group performs AND, OR, EXOR operations, compare, rotate or take the complement of data in register or memory. The logic operations cannot be performed directly with the content of two registers.

Instruction Set	Meaning	Example	Addressing Mode	Bytes of Instruction
ANA r [A] ← [A] ∧ [r]	AND register with the accumulator	ANA B	Register addressing mode	1-byte
ORA r [A] ← [A] ∨ [r]	OR register with the accumulator	ORA B	Register addressing mode	1-byte
CMA [A] → [$\bar{A}$]	Complement the accumulator	CMA	Implicit addressing mode	1-byte
CMP r	Compare register with accumulator	CMP B	Register addressing mode	1-byte

Types of the Instruction set of 8085 based on Operation Modes

4. Branch Group Instruction set of 8085

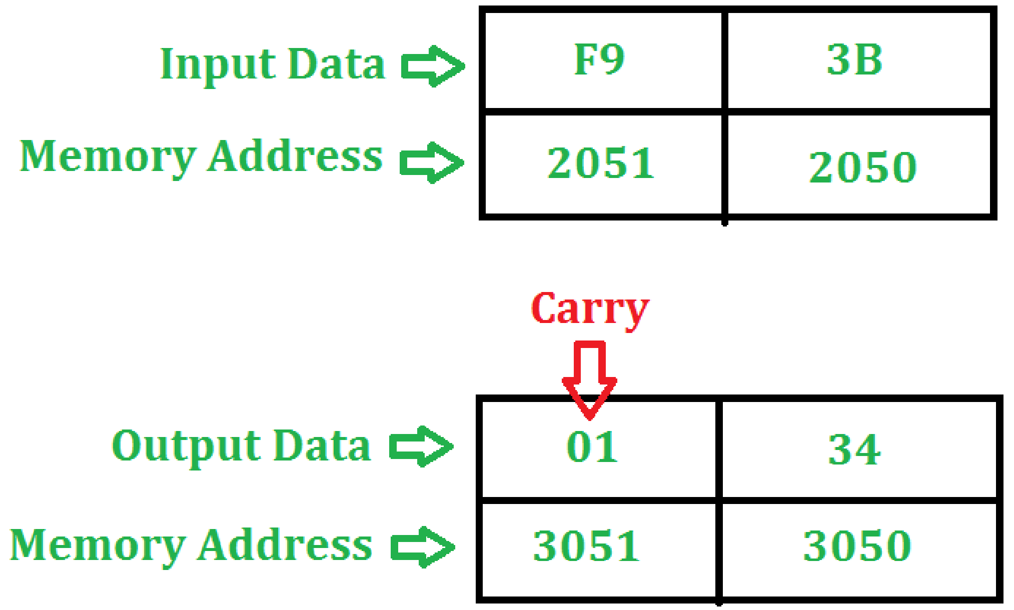
The Branch Instructions are the most powerful instructions because they allow the microprocessor to change the sequence of a program, either unconditionally or under certain test conditions.
These instructions are the key to the flexibility and versatility of a computer.
These are classified into 3 categories:

JUMP instructions
CALL and RETURN instructions
RESTART instructions.

Opcode	Operand	Description
JC	16-bit	Jump on carry (if result generates carry and CY=1)
JNC	16-bit	Jump on no carry (CY=0)
JZ	16-bit	Jump on zero (if result is zero and Z=1)
JNZ	16-bit	Jump on no zero (Z=0)

8085 program to add two 8 bit numbers

Problem – Write an assembly language program to add two 8 bit numbers stored at address 2050 and address 2051 in 8085 microprocessor. The starting address of the program is taken as 2000.
Example:



Program:

Memory Address	Mnemonics	Comment
2000	LDA 2050	A<-[2050]
2003	MOV H, A	H<-A
2004	LDA 2051	A<-[2051]
2007	ADD H	A<-A+H
2008	MOV L, A	L←A
2009	MVI A 00	A←00
200B	ADC A	A←A+A+carry
200C	MOV H, A	H←A
200D	SHLD 3050	H→3051, L→3050
2010	HLT	

Algorithm –

1. Load the first number from memory location 2050 to accumulator.
2. Move the content of accumulator to register H.
3. Load the second number from memory location 2051 to accumulator.
4. Then add the content of register H and accumulator using “ADD” instruction and storing result at 3050
5. The carry generated is recovered using “ADC” command and is stored at memory location 3051

Explanation –

1. LDA 2050 moves the contents the of 2050 memory location to the accumulator.
2. MOV H, A copies contents of Accumulator to register H to A
3. LDA 2051 moves the contents of the 2051 memory location to the accumulator.
4. ADD H adds contents of A (Accumulator) and H register (F9). The result is stored in A itself. For all arithmetic instructions A is by default an operand and A stores the result as well
5. MOV L, A copies contents of A (34) to L
6. MVI A 00 moves immediate data (i.e., 00) to A
7. ADC A adds contents of A(00), contents of register specified (i.e A) and carry (1). As ADC is also an arithmetic operation, A is by default an operand and A stores the result as well
8. MOV H, A copies contents of A (01) to H
9. SHLD 3050 moves the contents of L register (34) in 3050 memory location and contents of H register (01) in 3051 memory location
10. HLT stops executing the program and halts any further execution

Thank You