



DATA TRANSFER
AND
DATA MANIPULATION



DATA TRANSFER AND DATA MANIPULATION

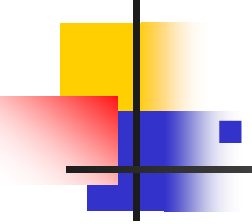
- There is a basic set of operations that most computers include in their instruction set
- The opcode and/or symbolic code may differ for the same instruction among different computers
- There are three main categories of computer instructions:
 - Data transfer
 - Data manipulation
 - Program control



Data Transfer Instruction

- **Data Transfer instructions move data from one place in the computer to another without changing the content**
- **The transfers are between memory & processor registers, b/w processor registers & i/p or o/p & b/w processor register themselves.**

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

- 
- Load instruction transfers the data from memory to a processor (accumulator).
 - Store instruction transfers from a processor register into memory.
 - Move instruction transfers the data from one register to another registers.
 - The exchange instruction swaps information between two registers or between register and memory.
 - Input and output instructions transfer data among processor registers and input or output terminals.
 - Push and pop instruction transfer data among processor registers and a memory stack.



DATA MANIPULATION INSTRUCTIONS

- Data manipulation instructions perform operations on data and provide the computational capabilities for the computer.
- The data manipulation instruction in a computer are usually divided into three basic types:
 1. Arithmetic instructions
 2. Logical and Bit manipulation instructions
 3. Shift instructions.

Arithmetic Instructions:

The four basic arithmetic operations are addition, subtraction, multiplication & division.

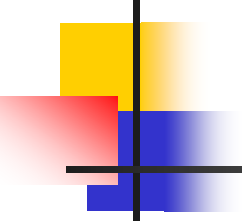
- Some computers have only addition & subtraction instruction .The multiplication & division must then be generated by means of Software Subroutines

Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with carry	ADDC
Subtract with borrow	SUBB
Negate (2's complement)	NEG



A list of Typical arithmetic instruction are as follows:-

- 1. Increment (INC):**-this instruction adds 1 to the value stored in a register or memory word.
 - One common characteristic of increment operations when executed in processor registers is that a binary number of all 1's when incremented produces a result of all 0's.
- 2. Decrement (DEC):**-this instruction subtracts 1 from the value stored in a register or memory word.
 - One common characteristic of decrement operations when executed in processor registers is that a binary number of all 0's when decremented, produces a result of all 1's.

- 
-
3. **Addition (ADD)**
 4. **Subtract (SUB)**
 5. **Multiply (MUL)**
 6. **Divide (DIV)**

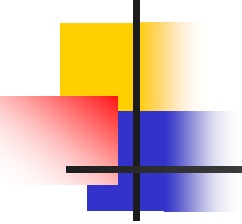
These instructions may be available for different types of data .it may be binary, decimal, floating-point data.

The mnemonics for three add instructions that specify different data types are shown below:

ADDI:- Add two binary integer numbers.

ADDF:-Add two floating-point numbers.

ADDD:-Add two decimal numbers in BCD.

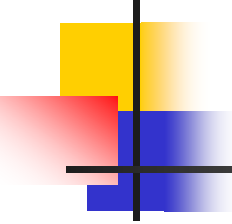
- 
-
7. **Add with Carry (ADDC):-** A special carry flip- flop is used to store the carry from an operation. The instruction “add with carry” performs the addition on two operands plus the value of the carry from the previous computational.
 8. **Subtract with borrow(SUBB):-** subtracts two words and a borrow which may have resulted from a previous subtract operation.
 9. **Negate (2’s complement):-**the negate instruction forms the 2’s complement of a number effectively.



Logical and Manipulation instructions

- They are useful for manipulating individual bits or a group of bits that represent binary-coded information.
- The logical instruction consider each bit of the operand separately and treat it as a Boolean variable.

Name	Mnemonic
Clear	CLR
Complement	COM
AND	AND
OR	OR
Exclusive-OR	XOR
Clear carry	CLRC
Set carry	SETC
Complement carry	COMC
Enable interrupt	EI
Disable interrupt	DI

- 
-
1. **Clear (CLR)**:- the clear instruction causes the specified operand to be replaced by 0's.
 2. **Complement (COM)**:-the complement instruction produces the 1's complement by inverting all the bits of the operands.
 3. **AND (AND)**
 4. **OR (OR)**
 5. **Exclusive-OR (XOR)**
- The AND, OR, XOR instructions produces the corresponding logical operations on individual bits of the operands.



6. **Clear Carry (CLRC):-**

7. **Set Carry (SETC):-**

8. **Complement Carry (COMC):**

Individual bits such as a carry can be cleared, set, or complement with appropriate instructions.

9. **Enable interrupts (EI):-**

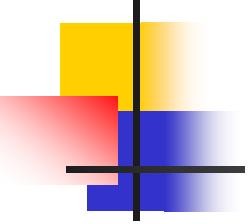
10. **Disable interrupts (DI):-**

Flips flops that controls the interrupts facility and is either enabled or disabled by means of bit manipulation instructions.



Shift instructions:-

- Instructions to shift the content of an operand.
- Shifts are operations in which the bits of a word are moved to the left or right, the bit shifted in a end of the word determines the type of shift used.
- Shift instruction may specify either Logical shifts, Arithmetic shifts ,or rotate –type operations.
- Instructions are as follows:



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

- **Logical shift right (SHR)**
- **Logical shift LEFT(SHL)**

Logical shift inserts 0 to the end bit position. the end position is the leftmost bit for shift right and the rightmost bit position for the shift left.



3. **Arithmetic Shift right (SHRA)**

4. **Arithmetic Shift left (SHLA)**

Arithmetic shift right instruction must preserve the sign bit in leftmost position. the sign is sifted to the right together with the rest of the number, but the sign bit itself remain unchanged.

5. **Rotate right(ROR)**

6. **Rotate left (ROL)**

Rotate instructions produces a circular shift. Bits shifted out at one end of the word are not lost as in a logical shift but are circulated back into the other end.



7. Rotate right through carry (RORC)

8. Rotate left through carry (ROLC)

The rotate through carry instruction transfers the carry bit into the rightmost bit position of the register, transfers the leftmost bit position into the carry, and at the same time, shifts the entire register to the left.

Some computer have a multiple-field format for the shift instructions

OP REG TYPE RL COUNT

Op is the operation code field, REG is a register address that specify the location of the operand, TYPE is a 2-bit field specifying the four different types of shifts, RL is a 1-bit field specifying a shift right or left, COUNT is a no. of shifts.