Micro-Operations

The operations executed on data stored in registers are called micro-operation. A micro-operation is an elementary operation Performed on the information Stored in one or more registers.

Example, shift, count, clear and Load.

The most common micro-operations Performed in a digital computer can be classified into four categories:

(i) Register transfer micro-operations

(ii) Arithmetic micro-operations

(iii) Logic micro-operations

(iv) Shift micro-operations.

(i) Register transfer micro-operation :-

These micro-operations transfers information from one register to another. The information doesnot change during these micro-operations. A register transfer micro-operation may be designed as:

$$R_1 \leftarrow R_2$$

$\leftarrow$ symbol implies that the content of register R_2 are transferred to register R_1. Here, R_2 is a source register while R_1 is a destination register.

The following important points about register transfer micro-operations are:

* For register transfer micro-operation, their must be a path for data transfer from the output of the source register to the input of the destination register.

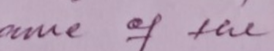
* In addition, the destination register should have a parallel load capability, as we expect the register transfer to occur in a pre-determined control condition.

* A common path for connecting various registers is through a common internal data bus of the processor. In general, the size of this data bus should be equal to the number of bits in a general register.

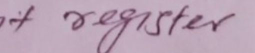
The conventions used to represnst the micro-operations are:

1. Computer register names are designated by capital letters (some time followed by numerals) to denote it's function. For example, R_1, R_2 (General purpose registers), AR (Address register), IR (Instruction Register) etc.

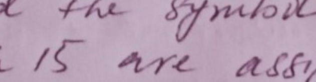
2. Individual bits within a register are numbered from 0 (right most bit) to n-1 (leftmost bit) as shown in fig.(b). Common ways of drawing the block diagram of a computer register are shown below. The name of the 16-bit register which is partitioned into two subfields (in fig. d), bits 0 through 7 are assigned the symbol L (for low byte) and bits 8 through 15 are assigned the symbol H (for high byte). The symbol PC (L) refers to low-order byte and PC(H) refers to high order byte.



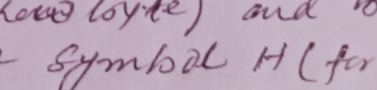
(a) Register



(b) Individual bits



(C) Numbering of bits



(d) Subfield

(3) Information transfer from one register to another is designated in symbolic notation by a replacement operator. For example, the statement $R_2 \leftarrow R_1$ denotes a transfer of all bits from source register R_1 to designation register R_2 during one ~~clock~~ clock pulse and the destination register has a parallel load capacity. However, the content of register R_1 remain unchanged after the register transfer microoperation. More than one transfer can be shown using a comma operator.

(4) If a transfer is to occure only under a pre-determined control condition, then this condition can be sepecified as a control function. For example, if 'P' is a control function then 'P' is a Boolean variable that can have a value of 0 or 1. It is terminated by colon (:) and placed in front of the actual transfer statement. The operation specified in the statement takes place only when $P=1$. Consider the statements:

$$\text{If } (P=1) \text{ then } (R_2 \leftarrow R_1)$$

$$P: R_2 \leftarrow R_1$$

where P is a control function that can be either 0 or 1.

(5) All micro-operations are written on a single line are to be executed at the same time provided the statements or a group of statements to be implemented together are free of conflict.

A conflict occurs if two different contents are being transferred to a single register at the same time. For example, the statement: new line $X: R_1 \leftarrow R_2, R_1 \leftarrow R_3$ represents a conflict because both R_2 and R_3 are trying to transfer their contents at the same time.

(6) A clock is not included explicitly in any statements discussed above. However, it is assumed that all transfer occur during the clock edge transition immediately following the period when control function is 1. All statements imply a hardware construction for implementing the micro-operation statement as shown below:

Implementation of controlled data transfer from R_2 to R_1 only when $T=1$

$$T: R_1 \leftarrow R_2 .$$