DARSHINI.R.U
RA2011026020040
CSE AI & ML 'A' Sec

UNIT-II

26/10/2021

COA ASSIGNMENT-I

2 marks:

1. What are the types of basic instruction formats? Explain it with example.

There are 4 types of basic instruction formats. They are: (i) Zero address instruction
(ii) One address instruction
(iii) Two address instruction
(iv) Three address instruction

(i) **Zero Address Instruction:**

The absolute address of the operand is held in a special register that is automatically incremented or decremented to point to the top of the stack.

Eg:
```
PUSH T-A
PUSH T-B
ADD  T(A+B)
```

(ii) **One Address Instruction:**

This uses an implied accumulator register for data manipulation. One operand is in accumulator and the other is in the register or memory location.

Eg:
```
LOAD  A
ADD   B
STORE T
```

(iii) **Two Address Instruction:**

It has one OP code and two address fields. One address field is common and can be used either for source or destination and the other address field for source. Eg:
```
MOV R1, A
ADD R1, B
```

(iv) **Three Address Instruction:**

It has one OP code and three address field. One address field is used for destination and the other two address fields for source.

Eg: ADD R1, A, B

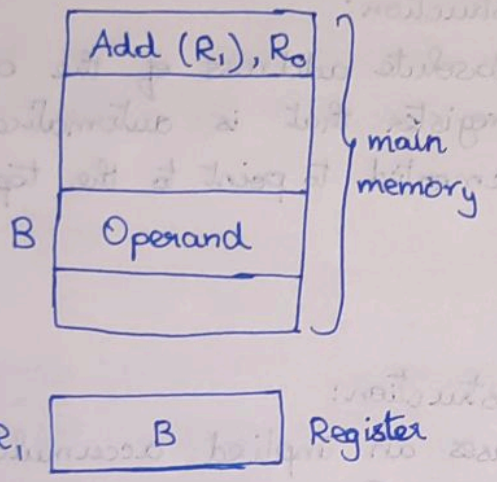
2. Explain Indirect Addressing Mode with example.

*. The effective address of the operand is the content of a register or memory location whose address appears in the instruction.

*. The register or memory location that contains the address of the operand is called a pointer.

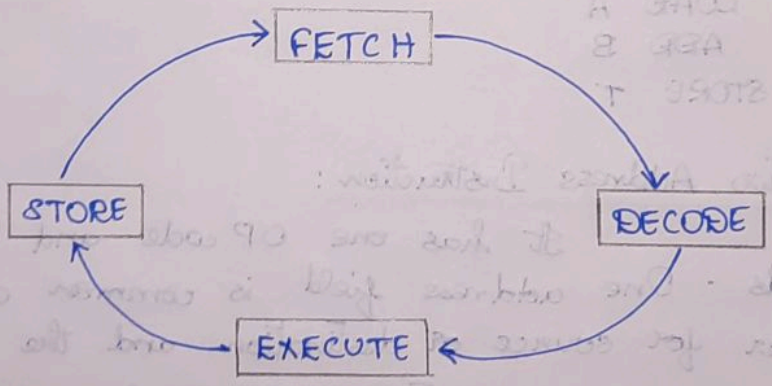
*. The instruction does not give the operand its address explicitly.

Example:



To execute the instruction, the processor uses the value B, which is in the register R_1, as the effective address of the operand.

3. Draw the flow of Instruction Cycle.



4. Define Register mode and absolute mode with example.

*. Register Mode:

The operand is the content of a processor register. The address of the register is given in the instruction. Eg: MOV LOC, R_2

*. Absolute mode:

The operand is in the memory location and the address of this is given in the instruction

Eg: MOV LOC, R_2 → uses two modes. Processor registers are used as temporary storage location where the data in a register are accessed using the register mode.

5. What are the various types of operation required for instruction?

(i) Data Transfer between the main memory and CPU registers.

(ii) Arithmetic and logical operations on data.

(iii) Program sequencing and control.

(iv) Input - Output Transfers.

10 marks: (any 2)

1. Describe in detail about different instruction types and instruction sequencing.

*. Instruction Types:

(i) Zero Address Instruction

(ii) One Address Instruction

(iii) Two Address Instruction

(iv) Three Address Instruction

(i) Zero Address Instruction:

The absolute address of the operand is held in a special register that is automatically incremented or decremented to point the location of the top of the stack. Eg:

PUSH T-A
PUSH T-B
ADD T(A+B)

(ii) One Address Instruction:

This uses an implied accumulator register for data manipulation. One operand is in accumulator and the other is in the register or memory location.

Eg:
```
LOAD  A
ADD   B
STORE T
```

(iii) Two Address Instruction:

It has one operand code and two address fields. One address field is common and can be used either for source or destination and the other address field is used for source. Eg:
```
MOV R1, A
ADD R1, B
```

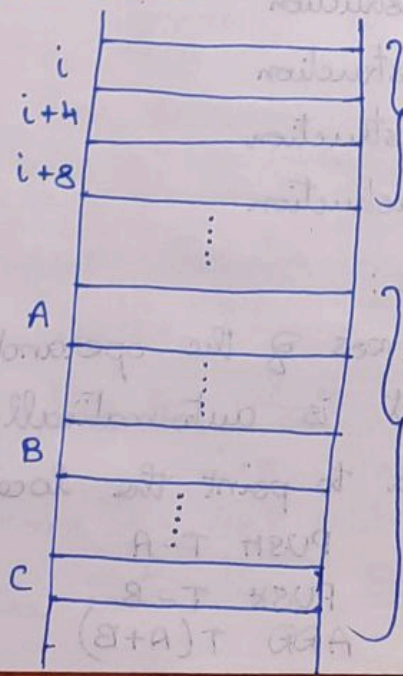
(iv) Three Address Instruction:

It has one operand code and three address fields. One address field is used for destination and two address fields for source. Eg: ADD R1, E, F

*. Instruction Sequencing:

Straight Line Sequencing:

Consider C ← [A] + [B]. We assume that the word length is 32 bits and the memory is addressable.



The 3 address instruction in the program are in successive word location starting at i. Since each instruction is 4 byte, the second and third instruction starts at i+4 and i+8

program for C ← [A] + [B]

The processor contains a register called program counter (PC) which holds the address of the instruction to be executed next.

To begin executing a program the address of the instruction must be placed in the program counter (PC).

Thus, the program control circuit uses the information in the PC to fetch and execute instruction one at a time in order of increasing address. This is called straight line sequencing.

During the execution of each instruction the PC is incremented by 4 to point the next instruction. Executing a given program is a two phase procedure: (i) Instruction fetch, (ii) Instruction Execute.

(i) Instruction Fetch:

The instruction is fetched from the memory location whose address is in the PC. The instruction is placed in the instruction register (IR).

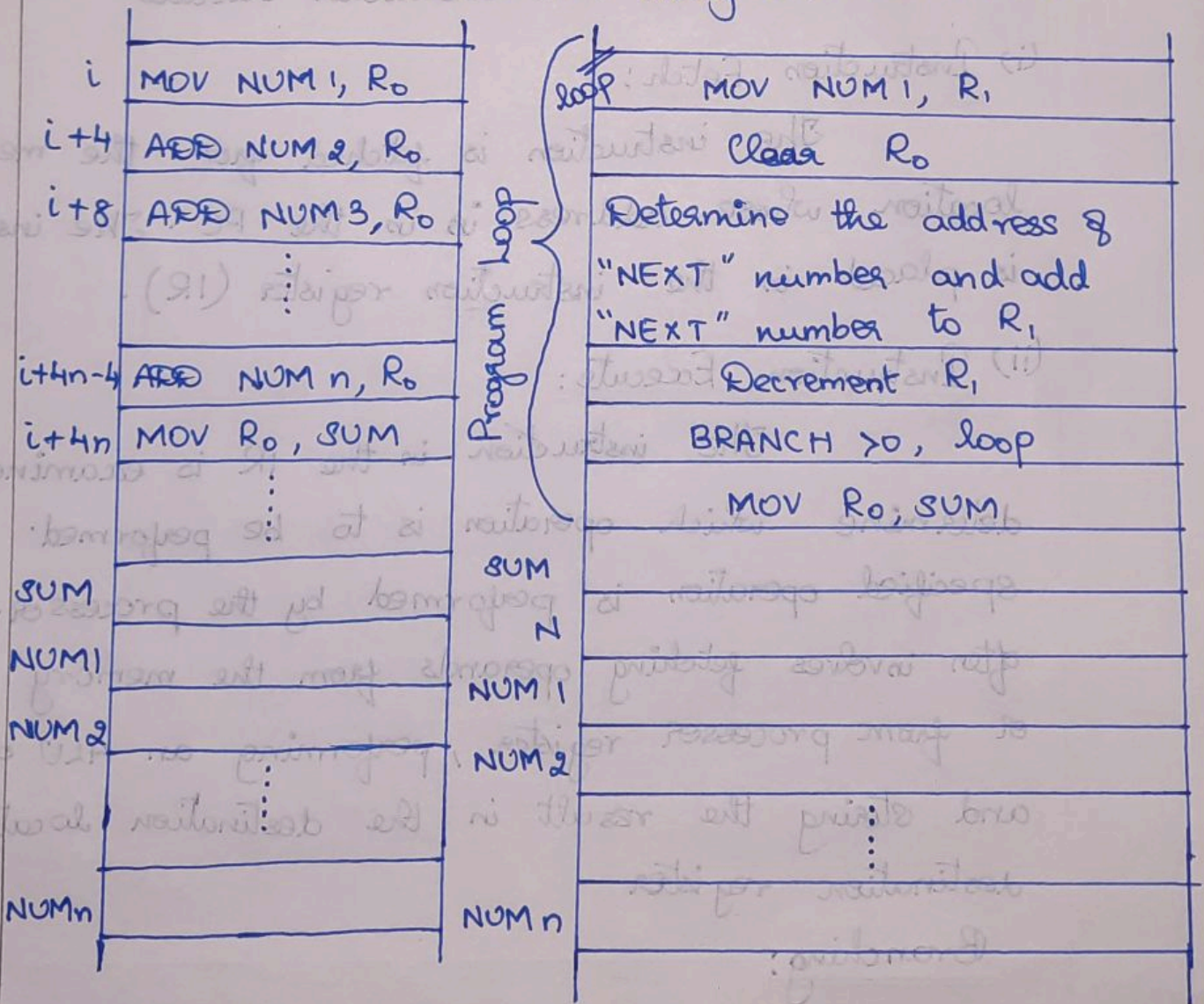
(ii) Instruction Execute:

The instruction in the IR is examined to determine which operation is to be performed. The specified operation is performed by the processor. This often involves fetching operands from the memory location or from processor register, performing an ALU operation and storing the result in the destination location or destination register.

Branching:

Consider the task of adding a list of 'n' numbers. The address of the memory location

containing 'n' numbers, are symbolically given as NUM1, NUM2, ..., NUMn and separate address instruction is used to add each number to the content of the register R_0. After all the numbers are added the result is placed in the memory location SUM. Instead of using a long list of address instruction, it is possible to place the single instruction in the program. The loop is a straight line sequence of instruction and executes as many times as needed.



It starts at location loop and ends at instruction BRANCH > 0 loop. Assume that the number of entities in the list n is stored in memory location N.

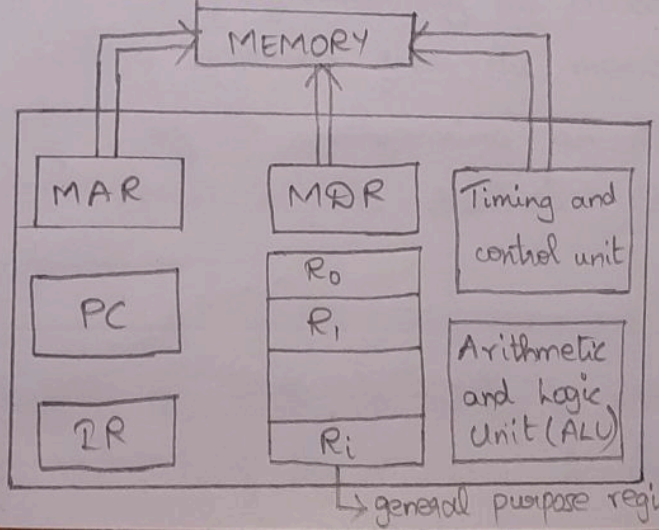
Register R_1 is used as the counter to determine the number of times the loop is executed. Hence, the content of the location N is loaded into register R_1 at the beginning of the program.

Within the body of the loop, the instruction Decrement R_1 reduces the content of R_1 by each time to the loop. Execution of the loop is repeated as long as the result of the decrement operation is greater than zero.

Branch instruction loads a new value into the PC. As a result, the processor fetches and executes instructions at the new address called BRANCH TARGET.

A conditional branch instruction causes a branch only if the specific condition is satisfied. If the condition is not satisfied, the PC is incremented in normal way and the next instruction in sequential address is fetched and executed.

2. Explain about memory location, memory address and memory operations in detail.



The program or task is performed with a group of instructions. The instructions are stored in the memory.

1 byte ⇒ 8 bits

A collection of n bits is referred to as a word, where 'n' is the word length.

The MAR holds the address of the data, whereas, MDR holds the Data.

Accessing the information or data has to be done using the bit or byte address.

Since we cannot operate or access the bits separately, we use continuous memory locations. The address is usually numbers from 0 to $2^k - 1$.

(i) Byte Addressibility:

If the word contains 32 bits, the allocations are 0, 4, 8, ... so on. For a register bit data 0, 1, 2....

(ii) Big-Endian and Little-Endian:

*. Big-Endian ⇒ Lower byte address is stored at Most Significant Bit (MSB)

*. Little-Endian ⇒ Lower byte address is stored at Least Significant Bit (LSB)

(iii) Word Allignment:

Address is alligned in the memory if they begin at a byte address (i.e) multiple number of bytes in a word. There can be unaligned words also. Number of bytes in a word is in powers of 2.

(iv) Accessing numbers, characters and character strings:

A number usually occupies one word. All are accessed by the address. End of a string is used to convey the last character of the string.

Memory Operations:

To perform operations, the instructions and data are stored in the memory location. Processor circuits transfer the operands and results from memory to processor and vice versa.

Two basic operations - Load and Store.

Load operation is used to read the instruction and data from the memory to processor.

Store operation destroyes the previous content from processor to memory. It sends the location along with the data that has to be transferred.