

Computer organization

chandras
Date: 7/7/20

Module-9

- With a neat diagram Explain different processor Registers?
List the steps needed to execute the machine instruction.
 ADD, LOC, A, R_0 ?

Ans

A Processor Registers is a local storage space on a processor that holds the data that is being processed by CPU.

Processor Registers generally occupy the top most position in memory hierarchy providing high speed storage space & fast access data.

Classification

- General-Purpose: - Temporarily stored data i.e. processed by CPU
- Special-Purpose: - These may store instruction Counter which contains address of next sequential instruction to be processed.

These can be classified into several types

- Conditional
- Address
- Vector
- Data
- Control & Status
- model Specific

The Steps needed to Execute machine instruction

1. Fetch Instruction
2. Decode Instruction.
3. Perform ALU operation
4. Access memory
5. update Register file.
6. update Program Counter [PC].

2. What is Bus? Explain Single & Multiple Bus Structure used to interconnect functional units in Computer System.

Ans A Bus is a group of lines that serve as Connecting path for several devices.

- * A Bus may be Lines (or) Wires
- * The line carry Data (or) address (or) Control Signal.

There are 2 types of Bus Structure.

1. Single Bus Structure.

2. Multiple Bus Structure.

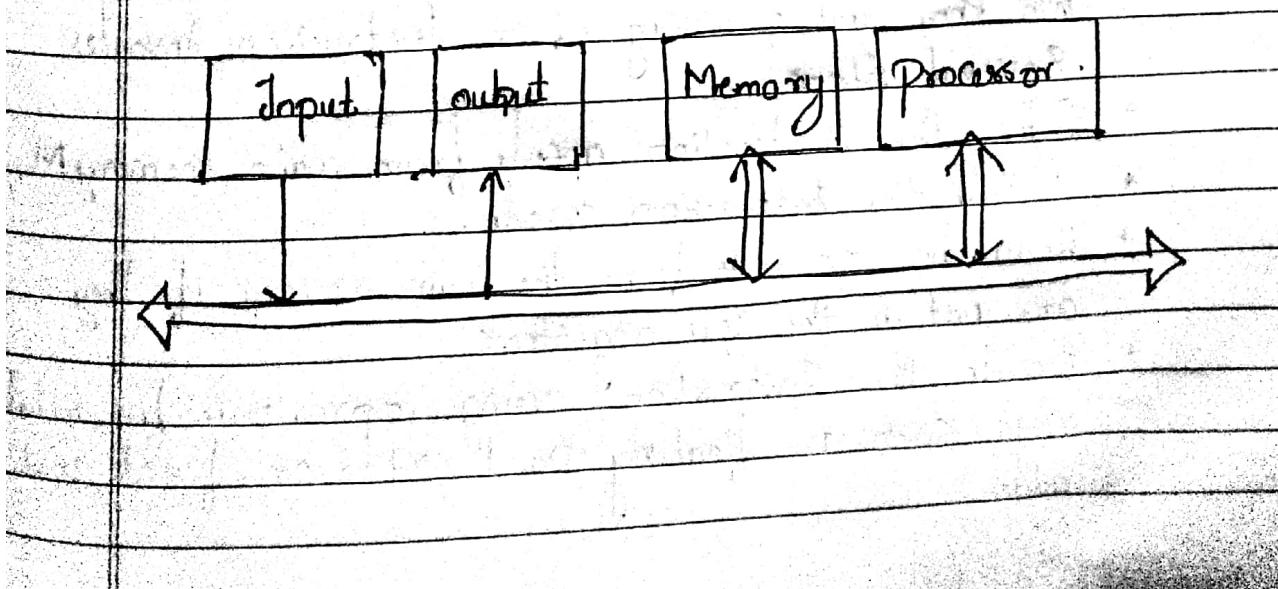
1. Single Bus Structure

The Bus used for only one transfer at a time. Only 2 units can actively use the bus at any given time.

- * Bus Control lines are used to attribute multiple requests for use of the bus.

Advantages :- Low cost.

flexibility for attaching multiple devices.



Multiple Bus Structures

The System that Contain multiple buses, achieve more Concurrency in operations.

- * Two or more operations Can be Carried at the Same time.

Advantages:- Better Performance.

Disadvantages:- Increased Cost.

3. Explain How the Performance of a Computer Can be measured?

Ans When we Compare the performance of different Computers say A, B and C, we may observe that Some programs run faster by Computer A, Some by Computer B and Some by Computer C. In this Situation they present a Confusing picture and we cannot have a clear idea of Which Computer is faster. This happens because Each Computer has an ability to Execute particular instruction or Step in the Instruction Execution faster than other.

We know that, processing of an instruction involves Several steps.

- * Fetch the Instruction ~~opcode~~ from main memory M.
- * Decode the Instruction opcode.
- * Load the Operands from the main memory if they are not in the CPU registers.
- * Execute the Instruction using appropriate functional unit, Such as floating point adder or fixed point adder.

- Q. * Store the results in the main memory unless they can be retained in CPU registers.

All the instructions do not require to perform all steps listed above. When the instruction has all its operands in CPU registers, it will run faster whereas the instruction which requires multiple memory access takes more time to execute.

Measures of instruction execution performance are based on average figures, which are usually determined experimentally by measuring the run times of representative called benchmark programs.

The selected benchmark programs are compiled for the computer under test, and the running time on a real computer is measured.

- * A non-profit organization called System Performance Evaluation Corporation (SPEC) specified the benchmark programs.

SPEC rating = $\frac{\text{Running time on the Reference Computer}}{\text{Running time on the computer under Test.}}$

$$\text{SPEC rating} = \left(\prod_{i=1}^n \text{SPEC}_i \right)^{\frac{1}{n}}$$

n: no of benchmark programs.

4. **Q** What is an addressing mode? Explain different generic addressing modes with an example for each?

Ans

The different ways in which the location of the operand is specified in an instruction are referred to as Addressing Mode.

1) Register Mode

- * The operand is in the Contents of a Register.
- * The Name of the Register is given in the Instruction.
- * Registers are used as temporary storage locations where the data in Registers are accessed.

Ex:- MOVE, R, R₂.

2) Absolute Mode

- * The operand is in Memory Locations.
- * The address of memory location is given Explicitly as the Instruction.
- * It Can Represent a global Variable.

Ex:- MOVE Loc, R₂.

3) Immediate Mode

The operand is given Explicitly in Instruction.

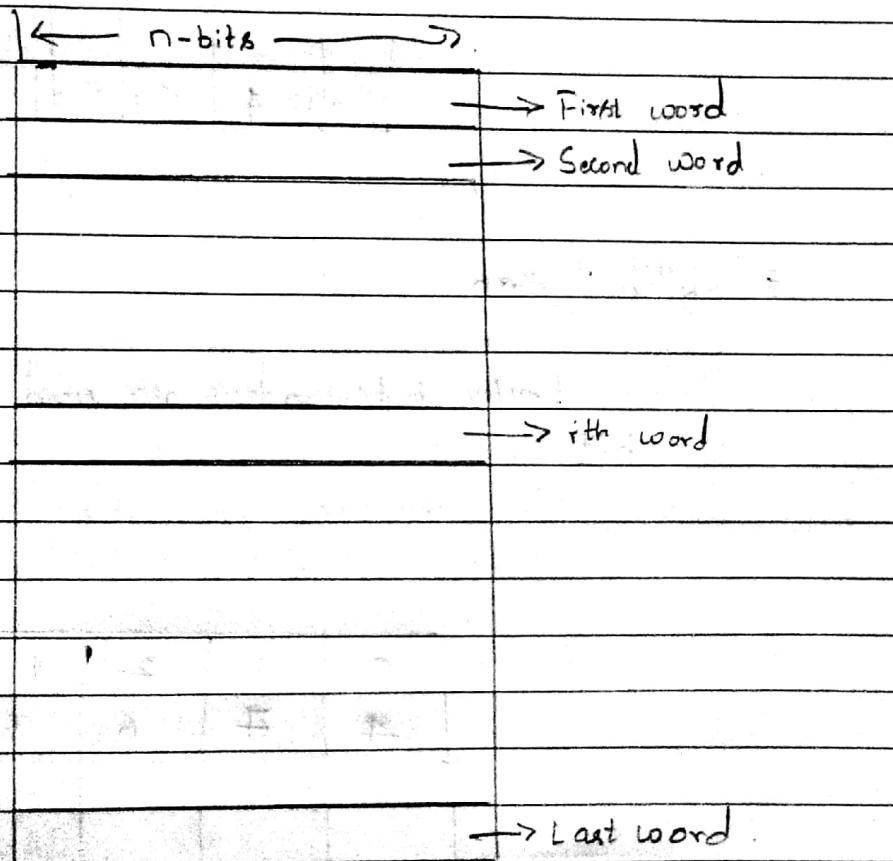
Ex:- MOVE #200, R₀.

The Immediate mode is only used to Specify the Value of a Source-operand.

5. Explain Byte Addressability. Mention the two ways that byte addresses can be assigned across the word with proper example?

Ans. In byte-addressable memory, successive addresses refer to successive byte locations in the memory.

- * Byte locations have addresses 0, 1, 2, ...
- * If the word-length is 32 bits, successive words are located at addresses 0, 4, 8 with each word having 4 bytes.



↑ Sign bit $b_3 = 0$ for +ve num
 $b_3 = 1$ for -ve num

8-bits	8-bits	8-bits	8-bits
--------	--------	--------	--------

ASCI

Character

ASCII
Character

There are two ways that byte addresses can be assigned.

1. Big Endian

Lower byte addresses are used for more significant bit.

Ex:-	0	0	1	2	3
	4	4	5	6	7
	$2^k - 4$	$(2^k - 4)$	$2^k - 3$	$2^k - 2$	$2^k - 1$

2. Little Endian

Lower byte address are used for less-significant bit

Ex:-	0	3	2	1	0
	4	4	6	5	4
	$2^k - 4$	$(2^k - 1)$	$2^k - 2$	$2^k - 3$	$2^k - 4$

6. Explain Assembler directives? Explain any two directives.

Ans * Directives are the assembly Commands to the assembler concerning the program being assembled.

* These Commands are not translated into machine opcode in the Object-program.

1. EQU :- informs the Assembler about the Value of an Identifier.

Ex: SUM EQU 900;

2. ORIGIN :- Tells the Assembler about the Starting-address of memory-area to place the datablock.

Ex: ORIGIN 204;

3. DATAWORD :- Directive tells the Assembler to load a Value into the location.

Ex:- N DATAWORD 100;

4. END RESERVE :- Directive tells the Assembler that is the End of the Source-program ber.

5. RESERVE :- directive is used to reserve a block of memory.

Ex:- NUM1, RESERVE 400,

7.

With a neat diagram Explain Simple I/O operations involving a Keyboard and a display device?

Ans

Consider the problem of moving a Character-Code from the Keyboard to the processor.

For this transfer, buffer-Registers DATAIN & a Status Control flag (SIN) are used.

- * When a Key is Pressed, the Corresponding ASCII Code is stored in a DATAIN Register associated with Keyboard.

$SIN = 1 \rightarrow$ when character is typed in keyboard. This informs the processor that a valid character is in DATAIN.

$SIN = 0 \rightarrow$ when character is transferred to the processor.

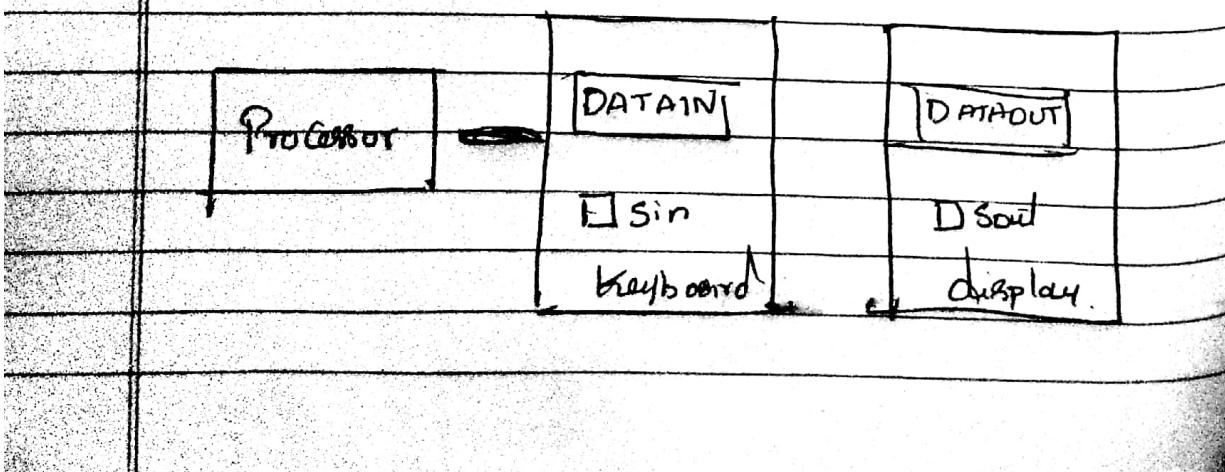
- * An analogous process takes place when characters are transferred from the processor to the display.

for this transfer, buffer-register DATAOUT & a Status Control flag SOUT are used

$SOUT = 1 \rightarrow$ when the display is ready to receive character

$SOUT = 0 \rightarrow$ when the character is being transferred

DATAOUT



8. Define Subroutine Explain Sub-routine Linkage using a Link Register & Stack frame? changes

Ans

A Subtask Consisting of a Set of instruction which is Executed many times is Called a Subroutine.

→ Sub-Routine Linkage :-

The way in which a Computer makes it possible to Call & Return from Subroutines is Referred to as Sub-routine Linkage.

Sub-routine Linkage using Link register

The Simplest Sub-routine Linkage method is to Save the return address in a Specific location, which may be register dedicated to this function Such a Register is Called a Link Register.

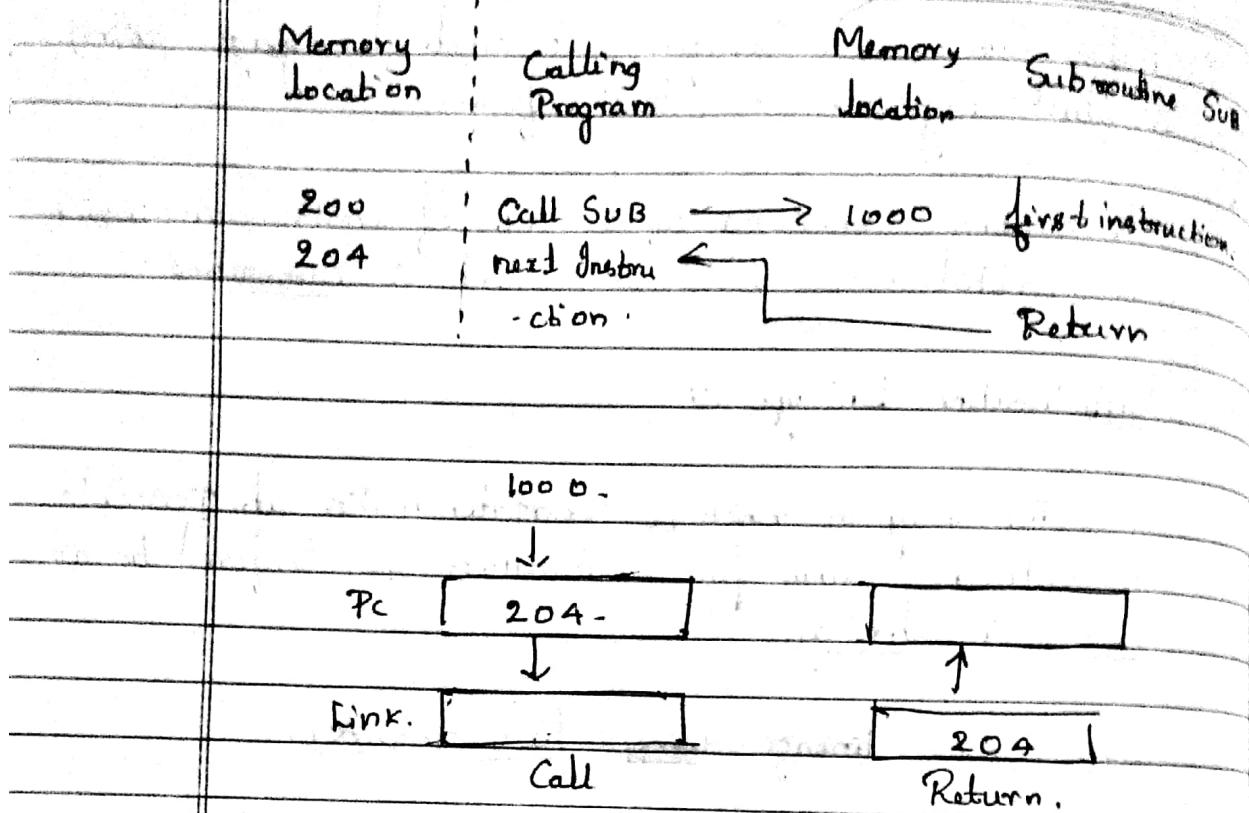
When the Sub-routine Completes its task, the Return instruction returns to the Calling program by branching directly through Link Register.

* Call Instruction :- Special Branch Instruction that Perform the following operations.

- * Store the Contents of PC into Link-Register.
- * Branch to the Target address Specified by the Instruction.

* Return Instruction :- Special Branch Instruction that Perform the operation.

- * Branch to the address Contained link Registers.

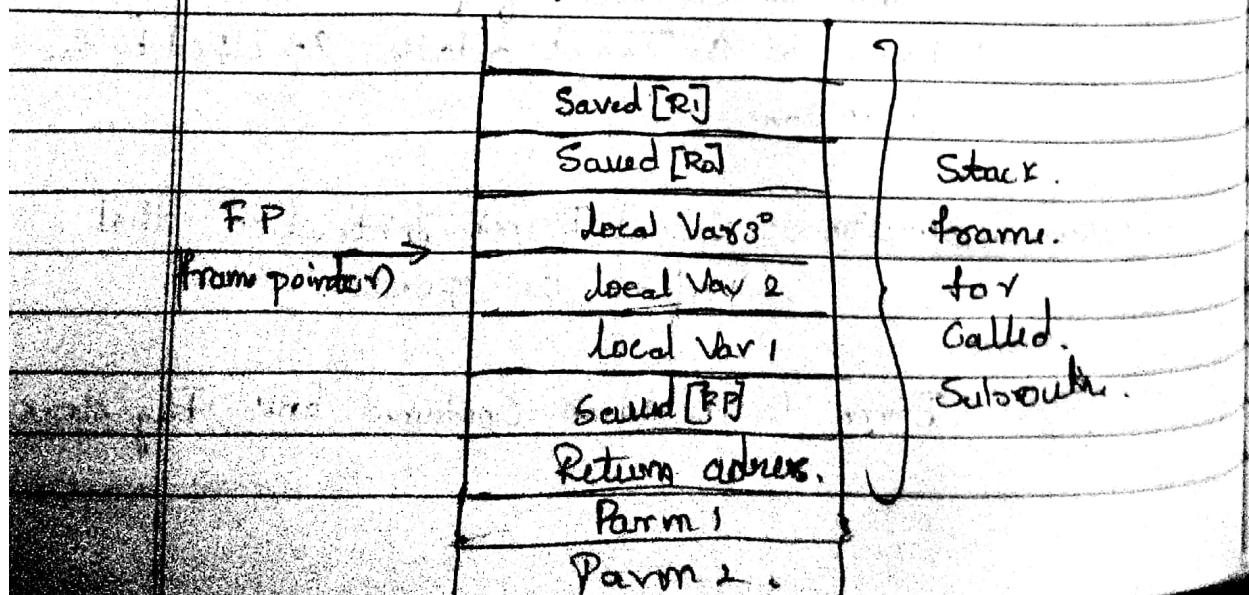


Sub-routine linkage using Stack - Frame.

Stack-frame refers to locations that constitute a private work-space for the Sub-routine.

The Work Space is.

- * Created at the time the Sub-routine is Entered.
- * Freed up when the Subroutine Returns Control to the Calling program.



Frame Pointer

It is used to access the parameters passed.

- * to the ~~Subroutine~~ & Subroutine

- * to the local memory - Variables

~~* Operation on Stack Frame~~

- * Initially Sp is pointing to the address of old TOS.
- * The Calling Program saves 4 parameters on the Stack.
- * The Call instruction is now Executed, pushing the Return address onto the stack.
- * The FP is initialized to the Value of Sp i.e both Fp and Sp point to the saved Fp address.

9. Explain different parameter passing techniques with usage of stacks in a nested Subroutine Calls.

Ans

Subroutine Nesting

It means one Subroutine Calls another Sub-routine.

- * In this Case, the return-address of the second call is also stored in the Link-Register, destroying its previous contents.
- * Sp is used to point the Processor.
- * Subroutine nesting can be carried out at any depth.

Parameter Passing

- * The Exchange of Information b/w a Calling program & a Sub-routine is referred as Parameter Passing.
- * The parameters may be placed in registers or in memory location where they can be accessed the Subroutine.

Following is a program for adding a list of numbers using Sub-routine with the parameters passed through Registers.

Calling program

Move	N,R1	R. Serves as a Counter
MUL	#NUM1,R2	R2 Points to the list
Call	LIST ADD	Call Subroutine.
Move	R0,SUM	Save Result.

Subroutine

LISTADD	Clear R ₀	Initialize Sum to 0
Loop	Add (R ₂) R ₀	Add Entry from list
Decrement R ₁		Return to Calling Program.
Branch > 0	Loop	
Return		

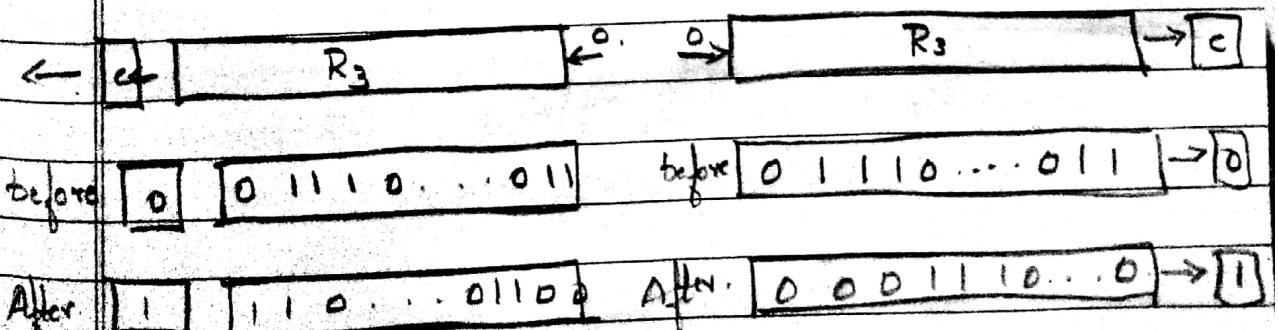
10. Explain different Shift & Rotate instruction with proper Examples?

Ans There are many Applications that require the bits of an operand to be Shifted Right or left some Specified no of bit-positions.

Logical Shifts

- Two Logical Shift Instruction are
 - Shift Left
 - Shift Right.

These Instructions Shift an operand over a number of bit positions specified in a Count Operand Contained in the Instruction



Logical Shift Left

Logical Shift Right.

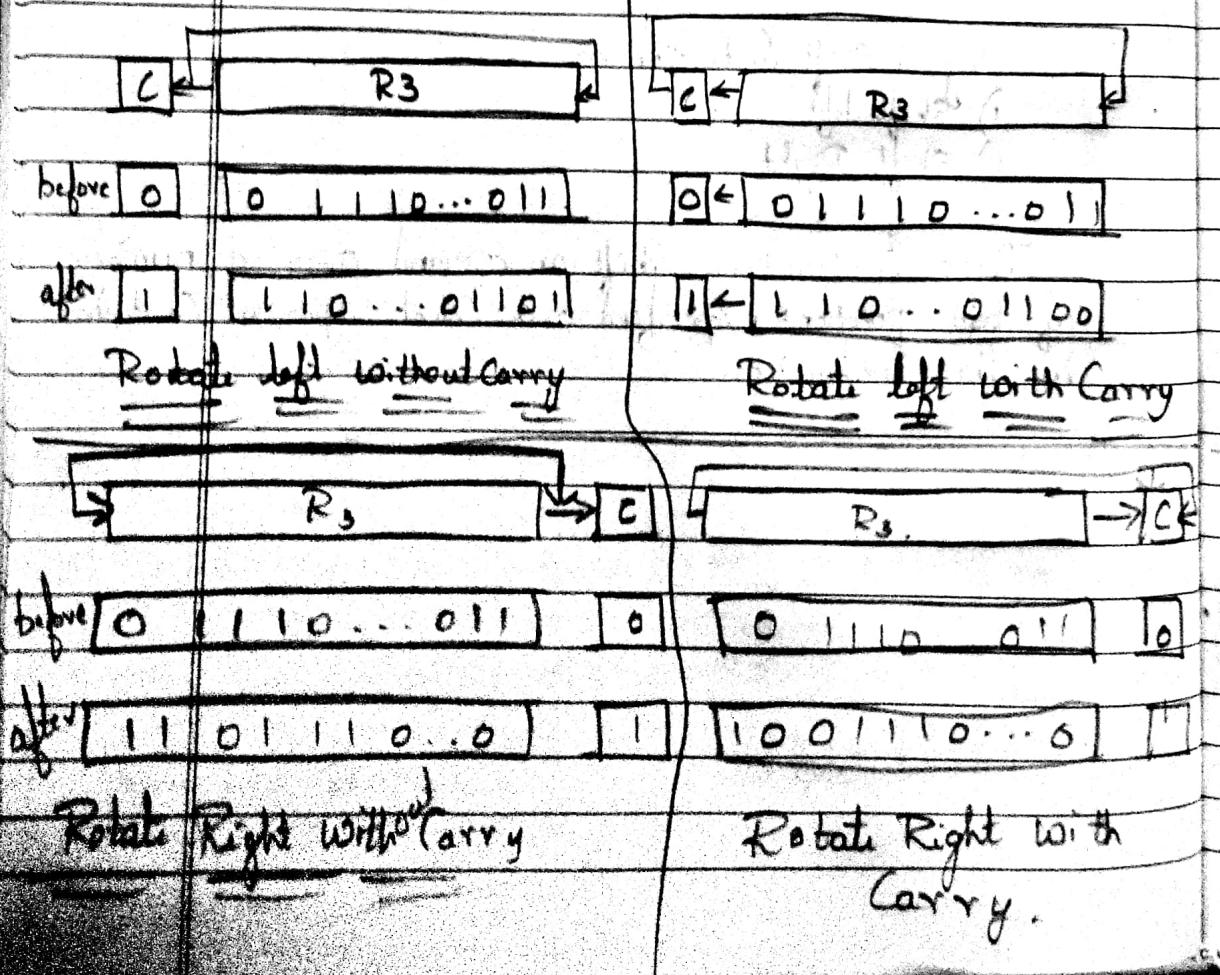
Rotate operations

In Shift operations, the bits shifted out the operand are lost. Except for the last bit shifted out which is retained in the Carry-flag C.

- * To preserve all bits, a set of rotate instructions can be used.
- * They move the bits that are shifted out of one end of the operand back to the other end.
- * Two versions of both the left and Right rotate instructions are usually provided.

In one version, the bits of the operand is simply rotated.

In other version, the rotation includes the C flag.



11. Explain Encoding of Machine Instruction into 32-bit words?

Ans. An Instruction must be Encoded in a binary pattern Such Encoded Instruction is Called as Machine Instructions.

- * It is to be Executed in processor, the Instruction that use the Symbolic names & acronyms are Called Assembly language Instructions.
- * We have Some Instructions that performs operations Such as Add, Subtract, move, Shift, Rotate and branch.
- * These Instructions may use operands of different sizes, Such as 32-bit & 8-bit numbers.
- * The op Code for given Instructions Can be Encoded. Refers to type of operation that is to be performed.
- * Source & the destination fields Refers to Source & destination operand Successfully.
- * CISC approach the result in instruction of Variable length, dependent on the number of operands and the type of addressing mode used.
- * In RISC (reduced Instruction Set Computers), any Instruction occupies only one word.
- * In RISC type machine, the memory references are limited to only Load/Store operations.

opcode	source	dest	other info.
--------	--------	------	-------------

(a) one word instruction.

opcode	source	dest	other info.
--------	--------	------	-------------

Memory address / Immediate operand.

(b) Two-word Instruction.

OP Code	Ri	Rj	Rk	other info.
---------	----	----	----	-------------

(c) Three-operand Instruction.

Example

The Instruction:

Move 24(R0), R5.

Requires 16 bits to denote the opcode and the two Registers and some bits to express that the source operand uses the Index addressing Mode and that index value is 24.

In the above Examples, the Instructions can be encoded in a 32-bit word

12. Explain the operations of Stack & Queue.

Ans Stack

It is a ordered Collection of Elements that has one End i.e. Front end. The Insertion & Deletion of the Element takes place only at one end.

Operations on Stack

1. To Create a Stack.
2. To Insert an Element on to Stack.
3. To delete an Element from Stack.
4. To check which Element from is on top of Stack.
5. To check weather Stack is Empty or not.

Queues

It is a ordered Collection of Elements that has two End i.e. Front End & Rear End. The Insertion takes place at Front End & Deletion takes place at Rear End.

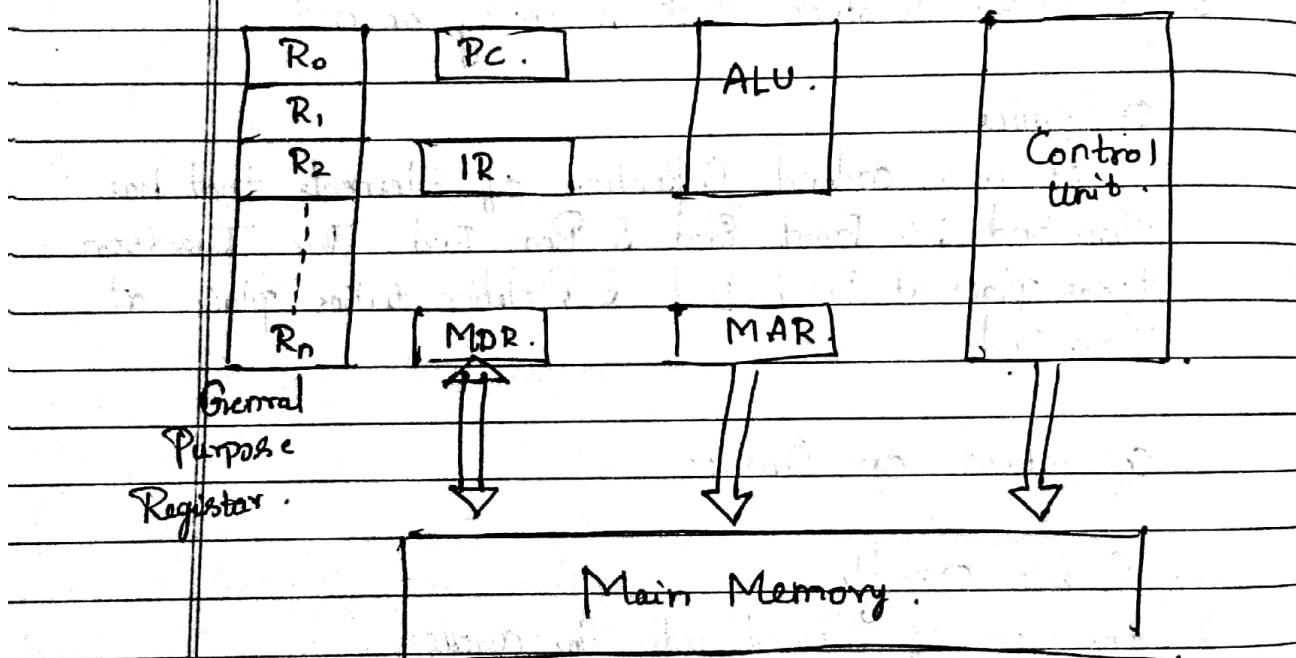
Operations on Queues

1. Queue overflow.
2. Insertion of Element into the queue.
3. Queue underflow.
4. Deletion of Element from the queue.
5. Display of the queue.

13. Explain the Basic operational Concepts of Computer?

Ans

- 1) The Basic function of Computer is to Execute a Program, Sequence of Instructions.
- 2) Instructions are Stored in the Computer memory.
- 3) Instructions are Executed to process data which is loaded into the Computer memory through Input unit.
- 4) After processing the data, the result is Either stored back into the Computer memory for further Reference.
- 5) All functional units of the Computer Contribute to Execute a program.



1) Program Counter [Pc] :-

A program is a Series of Instructions Stored in the memory. These instructions tell the CPU Exactly how to get the desired Result. It is important that those Instructions must be Executed in a proper order to get the Correct Result.

2) Information Register (IR)

It is used to hold the Instruction that is currently being executed. The Contents of IR are available to the Control Unit, which generate the timing signals that control the various processing elements involved in executing the instruction.

3) Memory Address Registers & Memory Data Registers

These Registers are used to handle the data transfers b/w main memory & the processor.

The MAR holds the address of main memory to or from which data is to be transferred.

4) General Purpose Registers

These are used to hold the operands for arithmetic & logic operation and used to store the result of operations. The access time of these Registers is lower, these are used to store frequently used data.

5) Execution of Program

- * PC is set to point to first instruction of program.
- * The contents of PC are transferred to MAR, whose output is connected to the address line of memory.
- * After access time of memory, the addressed word is recalled from memory & stored in the MDR.