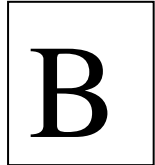


Reg.No:

--	--	--	--	--	--	--	--



SNS College of Technology, Coimbatore-35.
(Autonomous)
B.E/B.Tech- Internal Assessment -I
Academic Year 2022-2023(ODD)
Third Semester
Computer Science and Engineering



19ITT202 Computer Organization and Architecture
[Common to CSE & IT]

Time: 1.5 Hours

Maximum Marks: 50

Answer All Questions

PART - A (5x 2 = 10 Marks)

CO Blooms

1. List out the components of functional units of Computer. CO1 Rem

A computer in its simplest form comprises of five functional units namely

 - **input unit,**
 - **output unit,**
 - **memory unit,**
 - **arithmetic & logic unit &**
 - **control unit.**
2. Consider the $C \leftarrow [A] + [B]$ operation to be performed, write the sequence of instructions to be executed to perform the operation without destroying the former contents of location A and B, with respect to one, two & three address instruction. CO1 App

3 addr Add A,B,C
2 addr Move B,C
 Add A,C
1 addr Load A
 Add B
 Store C
3. Define Bus and label different types of buses used. CO1 Und

A group of lines, that serves as a connecting path for several devices is called as a bus.

Three types of bus are used

 - Address bus
 - Data bus
 - Control bus

4. If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds. How much faster is A than B. CO1 App

We know that A is n times as fast as B if

$$\frac{\text{Performance}_A}{\text{Performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = n$$

Thus the performance ratio is

$$\frac{15}{10} = 1.5$$

and A is therefore 1.5 times as fast as B.

5. Find 1's and 2's Complement of 1100 CO2 Und

To get 1's complement of a binary number, simply invert the given number. To get 2's complement of a binary number, simply invert the given number and add 1 to the least significant bit (LSB) of given result.

1's complement – 0011

2's complement - 0100

PART – B (13+13+14=40 Marks)

6. (a) Summarize the functional units of computer by extending the basic operational concepts. 13 CO1 Und

A computer in its simplest form comprises of five functional units namely

- **input unit,**
- **output unit,**
- **memory unit,**
- **arithmetic & logic unit &**
- **control unit.**

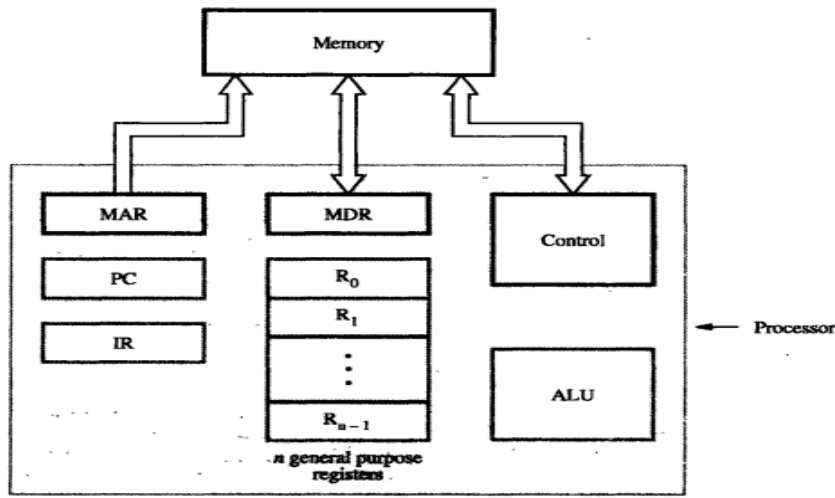


Figure 1.2 Connections between the processor and the memory.

(or)

- (b) Illustrate the execution of straight-line sequencing & branching instruction. Construct & compare the sequence of instruction to be performed for adding n numbers in both sequencing & branching instruction. 13 CO1 App

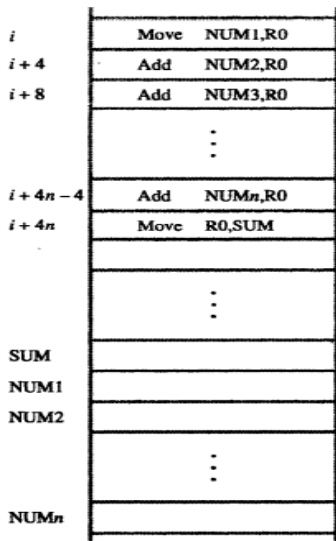


Figure 2.9 A straight-line program for adding n numbers.

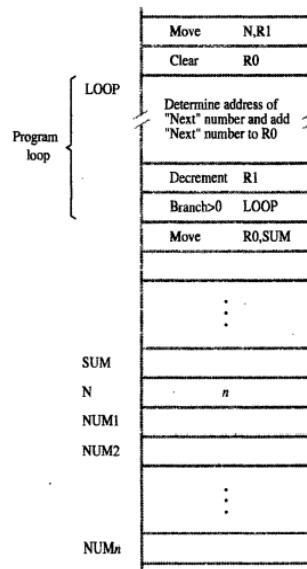


Figure 2.10 Using a loop to add n numbers.

7. (a) Interpret different addressing modes and experiment all modes by assuming the addition operation of N numbers to be performed and saved in SUM. 13 CO1 App

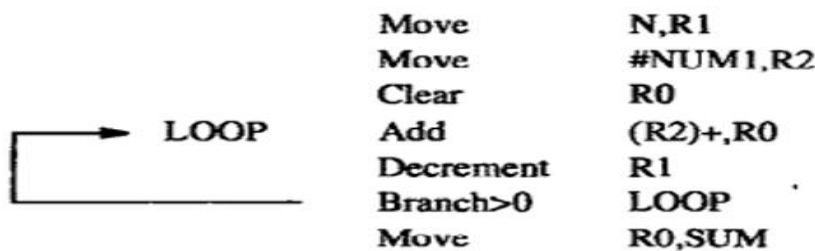


Table 2.1 Generic addressing modes

Name	Assembler syntax	Addressing function
Immediate	#Value	Operand = Value
Register	R _i	EA = R _i
Absolute (Direct)	LOC	EA = LOC
Indirect	(R _i)	EA = [R _i]
	(LOC)	EA = [LOC]
Index	X(R _i)	EA = [R _i] + X
Base with index	(R _i ,R _j)	EA = [R _i] + [R _j]
Base with index and offset	X(R _i ,R _j)	EA = [R _i] + [R _j] + X
Relative	X(PC)	EA = [PC] + X
Autoincrement	(R _i)+	EA = [R _i]; Increment R _i
Autodecrement	-(R _i)	Decrement R _i ; EA = [R _i]

EA = effective address
Value = a signed number

(or)

- (b) Identify the concept of addition and subtraction of signed numbers and examine the usage of each level in a problem. 13 CO2 Ana

Addition (subtraction) Algorithm

- When the sign of A and B are identical (**different**), add the magnitudes and attach the sign of A to the result.
- When the signs of A and B are different (**identical**), compare the magnitudes and subtract the smaller number from the larger.
 - Choose the sign of result to be same as A if A>B
 - or the complement of sign of A if A<B
 - if A=B subtract B from A and make the sign of result positive

Operation	Add Magnitudes	Subtract Magnitudes		
		A>B	A<B	A=B
(+A) + (+B)	+(A+B)			
(+A) + (-B)		+(A-B)	-(B-A)	+(A-B)
(-A) + (+B)		-(A-B)	+(B-A)	+(A-B)
(-A) + (-B)	-(A+B)			
(+A) - (+B)		+(A-B)	-(B-A)	+(A-B)
(+A) - (-B)	+(A+B)			
(-A) - (+B)	-(A+B)			
(-A) - (-B)		-(A-B)	+(B-A)	+(A-B)

- a) Load 20(R1),R5
- b) Move #3000,R5
- c) Store R5,30(R1,R2)
- d) Add -(R2),R5
- e) Subtract (R1)+,R5

Registers R1 and R2 of a computer contain the decimal values 1200 and 4600, we have to find effective address of associated memory operand in each instruction:

Load 20(R1),R5 : This means load 20+R1 into R5 . R1= 1200 , R1 + 20 = 1220 , so R5 have 1220 , Effective address of R5 is 1220.

Move #3000,R5 : This means move value 3000 into R5 , so effective address is part of the instruction whose value is 3000.

Now R5 = 3000

Store R5,30(R1,R2) : This means 30+R1+R2 and store the result into R5 .

so R5 = 30+1200+4600 = 5830 , so now R5 value is 5830 , the effective address is 5830.

Add -(R2),R5 : This means -1 from R2 value and store the result into R5 . So R5= 4600 - 1 = 4599 , effective address of R5 is 4599 . It is pre decrement addressing.

Subtract (R1)+,R5 : This means effective address is contents of R1 so EA = 1200 .

It is post increment addressing .

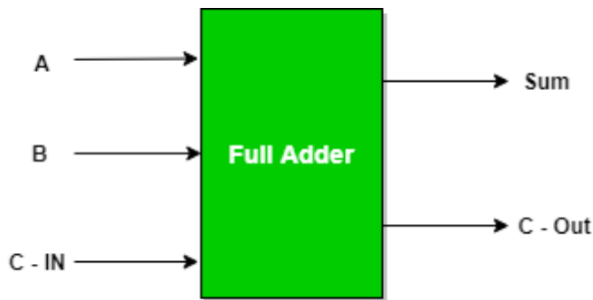
Effective addresses are

1. 1220
2. 3000 [it is not the effective address , it is the address of the instruction part where 3000 is stored]
3. 5830
4. 4599
5. 1200

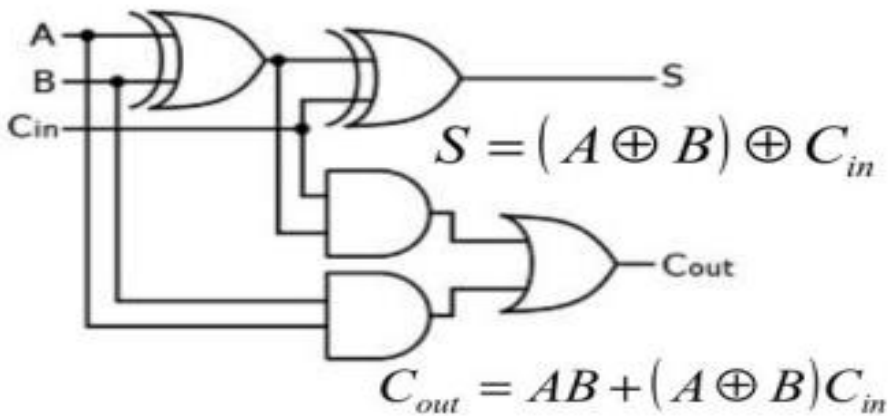
(or)

(b) Design and inspect the operation of Full Adder.

14 CO2 Und



Inputs			Outputs	
A	B	C - IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



(Note: Und-Understand Rem-Remember Ana-Analyze App-Apply Cre- Create)

Prepared By

Verified By

HoD