COAL MIDTERM SOLUTION

(PART I) marks = 20

Q1…………………………………………………………………………………………………………………………(Total marks=8)

1. What is Von-Neuman Architecture main concept……………………………………………………..… (4)

Von-Neuman gave the concept of stored program concept. According to it the computer has a main memory which contains both data and program to manipulate it.

1. Describe Fetch, decode, execute cycle. Explain using Load x instruction. (in MARIE ), write micro instructions using RTL……………………………………………………………………………………………….... (4)

When a computer executes a program it starts with the first instruction of the program.

The program counter register contains the address of the instruction to be executed.

So PC contains the address of the instruction to be executed.

The CPU tries to get the instruction stored at this address, for this purpose it First copies the contents of PC to MAR , because MAR is directly connected to Address lines of the Memory. in RTL notation first micro operation can be written as

1. PC 🡪 MAR

The aim is to get the memory word out on data bus and on to the instruction register, IR

This transfer of contents of PC to MAR is the first micro-operation.

To get the contents of memory pointed to by MAR memory read signal is generated.

So we get the contents of memory location pointed to by MAR.

2. M[MAR] 🡪 IR

The IR now contains the binary of the instruction to be executed. The binary instruction has

Both op-code and operands.

So we need to decode the op-code part to proceed with the execution of the instruction.

According to the Format the op-code is 4 higher order bits of the instruction. The remaining 12 bits is the operand which is address of the operand.

For example in case op load x instruction the x is the address of the operand.

This address is again loaded in MAR to get the operand from memory.

3. IR[11..0] 🡪 MAR

To get the value of operand we use address in MAR.

4. M[MAR] 🡪 MBR

And in the final step we transfer the contents of MBR to accumulator AC

5. MBR 🡪 AC

Q2……………………………………………………………………………………………………………………….………….……(4,4,4)

c) Describe the “indexed” mode instruction of the MARIE processor …………………..……………….(4)

d) if MARIE had 64 instructions in its instruction set and main memory 4kx16 then what would be the

FORMAT of Load x instruction. ………………………………………………………………………………………….. (4)

If instruction set contains 64 instructions we have to use 6 bit op-code. So we are left with 6 bits in the first word to accommodate operands which is not sufficient to contain 12 bit operand address. So we move the operand to the next word of the memory. So the instruction becomes two word instruction.

|  |  |  |
| --- | --- | --- |
| Op code | Unused 6 bits | |
| First 12 bits contain operand address | | Unused 4 bits |

e) Write RTLs for Load x instructions for the in part d above……………………………………………….. (4)

RTL with new format.

PC 🡪 MAR ; MAR has the address of the first word

M[MAR]🡪 IR ; IR contains the first word which is the first Word.

PC + 1 🡪 PC ; PC points to second word.

PC 🡪 MAR ; MAR has the address of second word

M[MAR] 🡪 MBR ;as IR Contains the op-code and Now MBR contains operand address.

MBR[15.. 4] 🡪 MAR ; MAR now contains the operand address.

M[MAR] 🡪 MBR ; MBR now has operand.

=====================================================================================

(PART II) Marks =20

Q1………………………………………………………………………………………………………………………………….(4,4)

Write a procedure “read-char” that inputs a character using DOS function.

read-char proc

Mov ah,01h

int 21h

ret

read-char endp

1. Write a procedure that inputs a string from keyboard and stores it into a local buffer. Use “read-char” procedure in this procedure.

buffer DB 80 DUP(0)

get-string proc

lea si,buffer

call read-char

mov [si], al

inc si

cmp al,13 ; check for enter

je l1

l1: ret

get-string endp

Q2………………………………………………………………………………………………………………………………….(4,4,4)

1. writ a program that copies unsigned numbers from one array to another but it copies numbers

that are in range of 50 to 180.

Array1 DW 1,2,3,4,5

Array2 DW 5 DUP(0)

Copy-unsigned proc

Mov si, offset Array1

Mov di, offset Array2

Mov cl,5

L1: Mov ax,[si]

Inc si

Call check

Jnc l1

Mov [di],ax

dec cl

jnz L1

ret

Copy-unsigned endp

Check proc

Cmp ax,50

Jl no

Cmp ax,180

Jge no

Stc

Ret

no: clc

ret

Check endp

1. write a procedure to ask user to type your name and store in a buffer. Then it uses the following

procedure to check each character and convert it to capital letter if it is a small letter.

Msg1 db “/nPiease enter your name$”

Msg2 db “/nplease enter alphabets only$”

Name db 80 dup(0)

Main proc

Lea si,Name

Lea DX, Msg1

Mov ah,09h

Int 21h ; message

Mov cl,80 ; maximum string length

L1: Mov ah,01h ; input a character

int 21h ; al contains input char

call chkalpha ; call to check if it is alphabet

jc save ; yes alphabet then go to save

jmp L1

save: [si],al

inc si

Main endp

End main

1. The procedure is used in part b) to check whether the character is a small letter or not and uses Z flag , it returns character in AL register and sets Z = 0 if it is not a small leter and Z= 1 if it is.

Chkalpha proc

Cmp al,’a’

Jl no

Cmp al,’z’

Jle yes1

Cmp al,’A’

Jl no

Cmp al,’Z’

Jle yes1

no: clc

ret

yes1: stc

ret

Chkalpha endp