Leading University

Department of CSE

Course Title: Microprocessor, Assembly Language and Computer interfacing sessional

Course Code: EEE-3211

Title: Evalution of microprocessors

Submitted to:

Rezwana Afrin Ridhi

Adjunct Lecturer

Department of CSE

Submitted by:

Faiza Mubarok

ID: 2112020026

Date : December 10, 2023

Evolution of microprocessors:

Microprocessore is a silicon chip that compromises millions of transistors and other electronic components that process million of instructions per second. A microprocesson is a versatile chip, that is combined with memory and special-purpose chips and programmed by software. It accept digital data as i/p and processes it according to the instructions store in the memory.

The microprocessor has many functions like functions of data storage, interact with various other devices, and other time-related functions. But the main function is to send and receive data to make the function of the computer well.

Now Lets see, how the microprocessor come at the present state. i.e let the evaluation of microproc-essor.

The evaluation of microprocessor was divided into five generations such as first, second, third, fourth and fifth generation and the characteristics of these generations are dissussed as below —

First Generation Microprocessors:

The first generation microprocessors were introduced in the year 1971 - 1972. The instructions of these microprocessors were processed serially, they fetch

the instruction, decoded and then executed it. When an instruction of the microprocessor was finished, then the microprocessor updates the instruction pointer & fetched the following instruction, performing the consecutive operation for each instruction in turn.

It was named Intel 4004 since it was a 4 bit processor. It was a processor on a single chip that could perform simple arithmatic and logical operations such as addition, substruction, Boolean OR and AND.

Second Generation Microprocessor:

In the year of 1970, a small number of transistors were available on the integrated circuit in the second-generation microprocessors. Example of second generation microprocessors are 16-bit arithmetic & pipelined instruction processing, MC68000 Motorola microprocessor. which were introduced at 1979. and Intel 8080 processor also. The second generation of the microprocessor is defined by overlapped fetch, decode and execute the steps. When the first generation is processed in the execution unit then the second instruction is decoded and the third instruction is fetched.

The second generation microprocessors were introduced in 1973 again by Intel. It was a 8-bit microprocessor which could perform arithmetic and logic operations on 8-bits words. It was Intel 8008, and another improved version was Intel - 8088.

Third Generation Microprocessors:

The third generation microprocessors were introduced in the year 1978, as denoted by Intel's 8086 and the Zilog - Z8000. These were 16 bits processors with a performance like mini computers. These types of microprocessors were different from the previous generations in that all main workstation industrialists began to envolving their own IBC based microprocessor architectures.

Fourth Generation Microprocessors:

As many industries converted from commercial microprocessors to in house designs, the fourth gene-ration microprocessors are entered with outstanding design with millions of transistors. Leading-edge microproc-essors like Motorola's 88100 and Intel's 80960CA could issue and retire more than one instruction per clock cycle. It is 32-bit microprocessors.

Fifth Generation Microprocessors.

From 1995 to now we are in the fifth generation. After 808056, Intel came out with a new processor namely pentium processor followed by Pentium Pro CPU which allows CPU's in a single system to archieve multipro-cessing.

Leading University

Department of CSE

Course Title: Microprocessor, Assembly Language and Computer interfacing sessional

Course Code: EEE-3112

Submitted to:

Rezwana Afrin Ridhi

Adjunct Lecturer

Department of CSE

Submitted by:

Faiza Mubarok

ID: 2112020026

Date : December 10, 2023

Contents

Write an assembly code that will print A-Z using loop.

Implementation's Process:

For printing A-Z, there are 26 alphabets and we use a loop where we'll use a register to and first add 65 as it is the ASCII value of "A" and then use the loop to increase the ASCII value and print the alphabets.

Code:

.model small model defines the main memory spaces. Small defines when code<= 64 Kb and data <= 64 Kb

.data used to declare and initialize data variable

.code this section contains the actual instructions or code that will be executed

main proc the main procedure works as the entry point for the program

mov cl,26 moving 26 in cl register. As there are 26 alphabets

mov bl,0 we use this register for tracking the alphabets order

L1: assigning the loop

mov dl,bl moving the bl vaule to dl value, as for printing we use dl register

add dl,65 we add 65 as it's the ASCII value of "A"

mov ah,2 2 is kind of function used for outputs. Here to set ah for displaying a character

int 21h call interrupt 21h to read the character

add bl,1 adding 1 in bl for moving to next alphabet

loop L1 to implement the loop

Exit:

mov ah,4ch 4ch is function which is used to terminate program. So it's to set up ah for program termination

int 21h call interrupt 21h to terminate the program

main endp used to end the main procedure

end main used to end program

Write an assembly code that will take three inputs from user and find the smallest number among these three inputs.

Implementation's Process:

We take 3 inputs and store them into 3 different registers. Then we compare two register values at a time and find the smallest among the two registers. We jump into different blocks according to the smallest value and then print the value.

Code:

.model small model defines the main memory spaces. Small defines when code<= 64 Kb and data <= 64 Kb

.data used to declare and initialize data variable

.code this section contains the actual instructions or code that will be executed

main proc the main procedure works as the entry point for the program

mov ah,1

int 21h

mov bl,al taking input and move it to bl registers

mov ah,1

int 21h

mov cl,al taking input and move it to cl registers

mov ah,1

int 21h

mov bh,al taking input and move it to bl registers

mov ah,13

mov ah,2

int 21h printing ASCII 13 i.e cret for starting the line from the start

mov ah,10

mov ah,2

int 21h printing ASCII i.e newl for newline

cmp bl,cl compare the value of bl with the value of cl

jg L1 jg means jumps greater. i.e if bl's value is greater than cl's value then jump to L1 block

L1: L1 block

mov bl,cl moving the cl's value to bl for making the bl's value smaller between cl and bl value

cmp bl,bh comparing bl's value with bh value.

jg L2 if bl is greater than bh, then will jump L2

mov dl,bl moving the value bl register to dl register, for printing the smallest value

mov ah,2

int 21h

jmp Exit jump to Exit block

L2:

mov dl,bh moving bh value to dl as bh holds the smallest value

mov ah,2

int 21h for printing

jmp Exit

Exit:

mov ah,4ch

int 21h

main endp

end main

Write an assembly code that will print “Welcome” if you press 1, will print “Today is your lab exam” If you press 2 and will print “Best of luck!” if you press 3.

Implementation’s Process:

We first stores the messages Into data. Then according to the input we get the offset address of the msg and then print the messages.

Code:

.model small

.data

msg1 db "Welcome$" msg1 is a label used to reference the data type. Db stands for data bytes.

Welcome is the string value for msg1. And $ refers the termination of the string.

msg2 db "Today is your lab final exam$" msg2 refers data type with the string message

msg3 db "Best of luck!$" msg3 refers data type

.code

main proc

mov ax,@data ax is 16 bits registers. As we are using string we need to use ax to store the address.

@data refers to the start address of dat segment

mov ds,ax

mov ah,1 for taking input

int 21h

mov bl,al

cmp bl,49 compare if the input is 1. The ASCII value of 1 is 49

je L1

cmp bl,50 compare if the input is 2. The ASCII value of 1 is 50

je L2

cmp bl,51 compare if the input is 3. The ASCII value of 1 is 51

je L3

jmp Exit

L1:

mov dl,offset msg1 stores in dl the offset address of msg1

mov ah,9 for printing a string , we use function 9

int 21h

jmp Exit

L2:

mov dl,offset msg2 stores in dl the offset address of msg1

mov ah,9

int 21h

jmp Exit

L3:

mov dl,offset msg3

mov ah,9

```
    int 21h



    Exit:


    mov ah,4ch
    int 21h
    main endp
end main
```