**COMPUTER ORGANIZATION AND ARCHITECTURE**

**MINI PROJECT**

**PROJECT TITLE: Piano Sound Generation**

Project ID: 11

Group members involved in the project:

|  |  |
| --- | --- |
| **NAME** | **ROLL NUMBER** |
| Zuleen Khan | AP20110010670 |
| Rajesh Reddy Anumula | AP20110010641 |
| Sri Harsha Kolusu | AP20110010651 |
| N. Santhi Lokesh | AP20110010691 |
| Joel Kalakoti | AP20110010663 |

**CONTENTS:**

* Objective.
* Assembly language Code for piano sound generation
* Functionality of the Piano.
* Conclusion.
* References.
* Objective

The prime objective of the assigned project is to make use of the computer keyboard as the piano keyboard to produce simple notes and rhythm. We can even observe the whole procedure through the icon image space which produces varying notes on different inputs from the user.

* Assembly language Code for piano sound generation.

.model tiny

.code

org 100h

main: jmp start

stor dw 0

start:

call curs\_off

get\_key:

mov ah,0

int 16h

cmp ah,1

je exit

cmp ah,02h

je tone\_1

cmp ah,03h

je tone\_2

cmp ah,04h

je tone\_3

cmp ah,05h

je tone\_4

cmp ah,06h

je tone\_5

cmp ah,07h

je tone\_6

cmp ah,08h

je tone\_7

cmp ah,09h

je tone\_8

cmp ah,0ah

je tone\_9

cmp ah,0bh

je tone\_0

;

jmp get\_key

;

exit:

call curs\_on

int 20h

;

tone\_1:

mov ax, 272

mov stor, ax

call sounder

jmp get\_key

tone\_2:

mov ax, 294

mov stor, ax

call sounder

jmp get\_key

tone\_3:

mov ax, 314

mov stor, ax

call sounder

jmp get\_key

tone\_4:

mov ax, 330

mov stor, ax

call sounder

jmp get\_key

tone\_5:

mov ax, 350

mov stor, ax

call sounder

jmp get\_key

tone\_6:

mov ax, 370

mov stor, ax

call sounder

jmp get\_key

tone\_7:

mov ax, 392

mov stor, ax

call sounder

jmp get\_key

tone\_8:

mov ax, 419

mov stor, ax

call sounder

jmp get\_key

tone\_9:

mov ax, 440

mov stor, ax

call sounder

jmp get\_key

tone\_0:

mov ax, 475

mov stor, ax

call sounder

jmp get\_key

curs\_off:

mov ch,10h

mov ah,1

int 10h

ret

curs\_on:

mov cx,0506h

mov ah,1

int 10h

ret

sounder:

mov al,10110110b

out 43h,al

mov ax,stor

out 42h,al

mov al,ah

out 42h,al

in al,61h

or al,00000011b

out 61h,al

call delay

and al,11111100b

out 61h,al

call clr\_keyb

ret

delay:

mov ah,00h

int 01Ah

add dx,4

mov bx,dx

pozz:

int 01Ah

cmp dx,bx

jl pozz

ret

clr\_keyb:

push es

push di

mov ax,40h

mov es,ax

mov ax,1Ah

mov di,ax

mov ax,1Eh

mov es: word ptr [di],ax

inc di

inc di

mov es: word ptr [di],ax

pop di

pop es

ret

;

end main

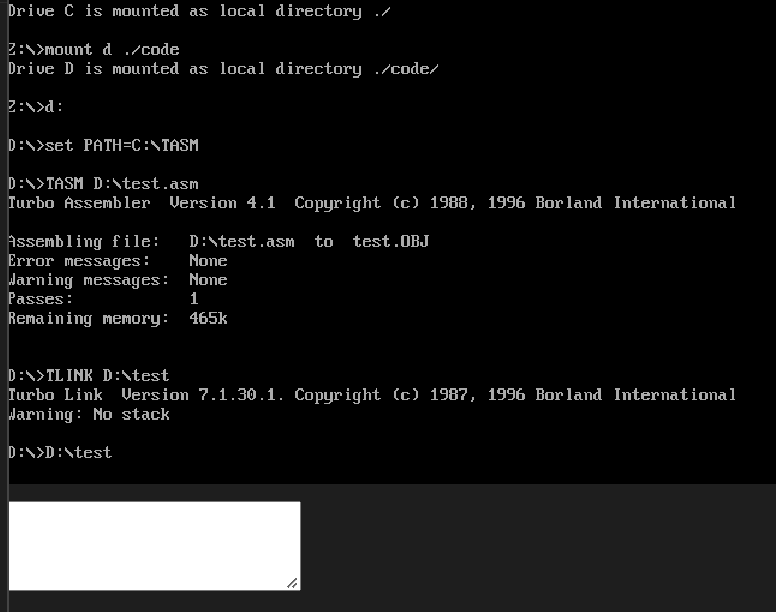
end

* The Functionality of the Piano.

We try to make the specific sounds through the computer speaker implemented by Assembly Language.

We use the DOSBox to implement the whole program with assembly, including sound and image aspects. We want to touch BIOS which is protected by Windows indirectly. So, we use the emulator---DOSBox to simulate the behavior. And we implement this project in two parts---sound and image.

The following steps may describe a detailed state to let us know more about the project.



1. The above code output snippet clearly defines the basic functionality of the piano. The expandable white rectangular serves as an input from the user.
2. It is, however, to be noted that the input must only be numeric keys to produce the desired sound or rhythm.
3. Combination of different numeric keys can help the end-user to produce sounds or notes of their respective choice.

* CONCLUSION

Using this project program can play simple songs with single notes. This is already a nice performance and also an interesting experience for us. There may be some improvements in this project. Through this project, we learn the skill to simulate it and do all the things in this basic language.

We firmly believe it can be improved furthermore. Advice is the power motivating us to advance it and so the advice from our teacher was very essential in making this project a success.

* REFERENCES

<https://codereview.stackexchange.com/questions/87322/a-virtual-piano>

<https://stackoverflow.com/questions/48507031/assembly-playing-audio-file-tasm-16-bit>

<https://en.wikipedia.org/wiki/DOSBox>