Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №3

«Программирование системного таймера»

Проверил: Выполнил:

к. т. н., доцент студент гр. 150501

Одинец Д. Н. Кардаш С. П.

Минск, 2023

**1 ПОСТАНОВКА ЗАДАЧИ**

Запрограммировать второй канал таймера таким образом, чтобы динамик компьютера издавал звуки.

Для всех каналов таймера считать слово состояния и вывести его на экран в двоичной форме.

Для всех каналов таймера рассчитать коэффициент деления (значение счетчика CE) и вывести его на экран в шестнадцатеричной форме.

2 АЛГОРИТМ

2.1 Структура программы

Программа состоит из двух частей. В первой части в регистр делителя частоты заносятся значения и открывается канал OUT, из которого данные переходят к динамику устройства. Во второй части программы происходит вывод состояния каналов в двоичной форме, а также вывод делителя частоты для каждого канала в шестнадцатеричной форме записи числа

**2.2 Описание переменных**

buf – массив, содержащий значения делителя частоты одного регистра.

**2.3Описание алгоритма программы**

Для разрешения подачи сигнала на динамик устройства в порт 61h записывается байт с не выставленными в «1» битами. После этого в регистр делителя частоты записываться значения, которые обеспечивают, частоту указанную в задании. Также устанавливается время звучания определенной частоты. В конце программы выводиться содержимое регистров состояния каналов таймера, а также значения регистра СЕ. **3 КОД ПРОГРАММЫ**

.model tiny

.code

org 100h

start:

in al, 61h ;set gate and lock at 1

or al, 00000011b ;write in port 61h

out 61h, al

mov al, 10110110b ;initialization word

out 43h, al ;write in port 43h

;;;first friquenci

mov al, 11100011b ;write lower byte of frequency divider

out 42h, al ;set lower byte

mov al, 00001011b ;write higher byte of frequency divider

out 42h, al ;set higher byte

mov ah, 86h ;bios function(wait)

mov dx, 3500h ;write in dx lower two bytes of number 800ms

mov cx, 0ch ;write in cx higer two bytes

int 15h ;start waiting

;;;second friquenci

mov al, 11100011b

out 42h, al

mov al, 00001011b

out 42h, al

mov ah, 86h

mov dx, 1A80h

mov cx, 06h

int 15h

;;;third friquenci

mov al, 11101000b

out 42h, al

mov al, 00001111b

out 42h, al

mov ah, 86h

mov dx, 1A80h

mov cx, 06h

int 15h

;;;fourth friquenci

mov al, 11000111b

out 42h, al

mov al, 00010111b

out 42h, al

mov ah, 86h

mov dx, 1A80h

mov cx, 06h

int 15h

;;;fifth friquenci

mov al, 11000111b

out 42h, al

mov al, 00010111b

out 42h, al

mov ah, 86h

mov dx, 1A80h

mov cx, 06h

int 15h

;;;sixth friquenci

mov al, 11100011b

out 42h, al

mov al, 00001011b

out 42h, al

mov ah, 86h

mov dx, 1A80h

mov cx, 06h

int 15h

;;;seventh friquenci

mov al, 11100011b

out 42h, al

mov al, 00001011b

out 42h, al

mov ah, 86h

mov dx, 1A80h

mov cx, 06h

int 15h

;;;eigth friquenci

mov al, 11101000b

out 42h, al

mov al, 00001111b

out 42h, al

mov ah, 86h

mov dx, 1A80h

mov cx, 06h

int 15h

;;;ninth friquenci

mov al, 11000111b

out 42h, al

mov al, 00010111b

out 42h, al

mov ah, 86h

mov dx, 1A80h

mov cx, 06h

int 15h

in al, 61h ;set gate and lock at 0

and al, 11111100b ;write in port 61h

out 61h, al

;output first channel

mov ah, 09h

mov dx, offset first

int 21h

mov al, 11000010b ;controll word for get registr of 0 channel

out 43h, al

in al, 40h

call output\_bytes

mov ah, 09h ;friquenci devider

mov dx, offset ce

int 21h

mov al, 00000000b ;set controll word to get ce

out 43h, al ;send controll word

in al, 40h ;get higher byte

mov di, offset buf+2 ;get string with hex code of divider

call byte\_to\_hex\_str

in al, 40h

mov di, offset buf

call byte\_to\_hex\_str

mov ah, 09h

mov dx, offset buf

int 21h

mov ah, 09h

mov dx, offset next

int 21h

;output second channel

mov ah, 09h

mov dx, offset second

int 21h

mov al, 11000100b

out 43h, al

in al, 41h

call output\_bytes

mov ah, 09h ;friquenci devider

mov dx, offset ce

int 21h

mov al, 01000000b ;set controll word to get ce

out 43h, al ;send controll word

in al, 41h ;get higher byte

mov di, offset buf+2 ;get string with hex code of divider

call byte\_to\_hex\_str

in al, 41h

mov di, offset buf

call byte\_to\_hex\_str

mov ah, 09h

mov dx, offset buf

int 21h

mov ah, 09h

mov dx, offset next

int 21h

;output third channel

mov ah, 09h

mov dx, offset third

int 21h

mov al, 11001000b

out 43h, al

in al, 40h

call output\_bytes

mov ah, 09h ;friquenci devider

mov dx, offset ce

int 21h

mov al, 10000000b ;set controll word to get ce

out 43h, al ;send controll word

in al, 42h ;get higher byte

mov di, offset buf+2 ;get string with hex code of divider

call byte\_to\_hex\_str

in al, 42h

mov di, offset buf

call byte\_to\_hex\_str

mov ah, 09h

mov dx, offset buf

int 21h

mov ah, 09h

mov dx, offset next

int 21h

mov ax,4C00h ;retutn to operation system

int 21h

output\_bytes proc

mov cx, 08

loop1:

rol al, 1

jc write1

push ax

mov ah, 06h

mov dl, 30h

int 21h

jmp finish

write1:

push ax

mov ah, 06h

mov dl, 31h

int 21h

finish:

pop ax

loop loop1

;mov ah, 09h

;mov dx, offset next

;int 21h

ret

output\_bytes endp

byte\_to\_hex\_str proc

push ax

mov ah,al

shr al,4

call to\_hex\_digit

mov [di],al

inc di

mov al,ah

and al,0Fh

call to\_hex\_digit

mov [di],al

inc di

pop ax

ret

byte\_to\_hex\_str endp

to\_hex\_digit proc

add al,'0'

cmp al,'9'

jle thd\_end

add al,7

thd\_end:

ret

to\_hex\_digit endp

ret

first db "0\_channel:", 20h, "$"

second db "1\_channel:", 20h, "$"

third db "2\_channel:", 20h, "$"

ce db 20h, 20h, "ce:", 20h, "$"

next db 0dh, 0ah, "$"

buf db 4 dup(?), "$"

end start

**4 ТЕСТИРОВАНИЕ**

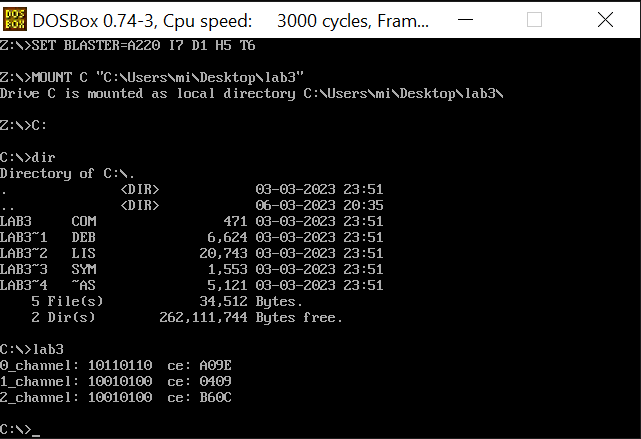


Рисунок 4.1 – Результат выполнения программы

**4 ЗАКЛЮЧЕНИЕ**

В ходе лабораторной работы был запрограммирован третий канал таймера, который подключен к динамику устройства, а также выведено состояние трех каналов таймера.

Для запуска программы использовался DOS-BOX установленный на операционную систему Windows 10.