Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра ЭВМ

Отчёт по лабораторной работе №3

“Программирование системного таймера”

Проверил: Выполнил:

Минск 2023

#### **Постановка задачи**

#### Задание состоит из двух частей. Первая часть общая для всех. Вторая часть по вариантам

#### Первая часть:

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

#### Вторая часть:

#### 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.

#### **Алгоритм решения задачи**

Системный таймер - устройство, подключенное к линии запроса IRQ0 и вырабатывающее прерывание int8h. Таймер реализуется на микросхеме Intel 8253 или 8254 (Отечественные аналоги: К1810ВИ53 и К1810ВИ54).

Таймер состоит из трёх независимых каналов: 0 - отсчитывает текущее время после включения компьютера, 1- для работы с контроллером прямого доступа к памяти, 2 - связан с системным динамиком.

Каждый канал содержит регистры:

 состояния канала RS (8 разрядов);

 управляющего слова RSW (8 разрядов);

 буферный регистр OL (16 разрядов);

 регистр счетчика CE (16 разрядов);

 регистр констант пересчета CR (16 разрядов)

Каналы таймера подключаются к внешним устройствам при помощи линий:

 GATE - управляющий вход;

 CLOCK - вход тактовой частоты;

 OUT - выход таймера.

Регистр счетчика CE работает в режиме вычитания. Его содержимое уменьшается по спаду сигнала CLOCK при условии, что на вход GATE установлен уровень логической 1.

В зависимости от режима работы таймера при достижении счетчиком CE нуля тем или иным образом изменяется выходной сигнал OUT.

Буферный регистр OL предназначен для запоминания текущего содержимого регистра счетчика CE без остановки процесса счета. После запоминания буферный регистр доступен программе для чтения.

Регистр констант пересчета CR может загружаться в регистр счетчика, если это требуется в текущем режиме работы таймера.

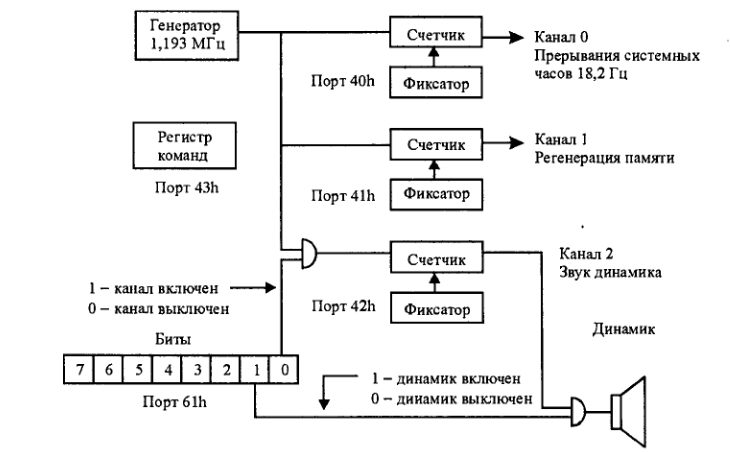
Возможны шесть режимов работы таймера, которые делятся на три типа:

Режимы 0, 4 - однократное выполнение функций.

Режимы 1, 5 - работа с перезапуском.

Режимы 2, 3 - работа с автозагрузкой.

Для подключения динамика используется порт 61h.



1. **Листинг программы**

#include <stdio.h> // ввод/вывод

#include <conio.h> // работа с консолью

#include <dos.h> // функции DOS

#define COUNT 28 // количество элементов в массиве

#define COUNTER 8 // -//-

#define DELAY 10 // задержка в мс

#define GO 65536 // констатнта для цикла

void sound(); // генерация звука на динамике

void div(); // расчет делителя частоты

void statusw(); // чтение состояния регистра

void main()

{

clrscr(); // очистка экрана

char command;

do

{

printf("\n----------------------------\n");

printf("1. Play another sound\n");

printf("2. Division factor\n");

printf("3. Status word\n");

printf("0. Exit the program\n");

printf("----------------------------\n");

printf("\nSelect command: ");

fflush(stdin);

scanf("%s", &command);

switch (command)

{

case '1': sound(); break;

case '2': div(); break;

case '3': statusw(); break;

}

} while (command != '0');

}

void sound()

{

int countHZ; // перебор элементов массива

int byte; // хранение частоты делителя

int HZ[COUNTER] = { 659, 622, 659, 622, 659, 493, 622, 523 }; // массив частот

int MS[COUNTER] = { 200, 200, 200, 200, 200, 200, 200, 200 }; // массив времени в мс

int countDelay[COUNTER] = { 100, 100, 100, 100, 100, 100, 100, 100 }; // массив задержек в мс

long unsigned base = 1193180; // IRQ 18,2 раза в секунду

for (countHZ = 0; countHZ < COUNTER; countHZ++)

{

// настройка 2-го канала:

// порт 42h (системный таймер, канал 2, звук динамика)

// порт 43h (командный регистр)

outp(0x43, 0xB6); // 0xB6 - настройка 2 канала по порту 43h

byte = base / HZ[countHZ]; // base - константа для расчета делителя частоты

outp(0x42, byte % 256); // //(младший)наименьший байт делителя частоты

outp(0x42, byte /= 256); // (старший) старший байт делителя частоты

outp(0x61, inp(0x61) | 3); // вкл

delay(MS[countHZ]); // ожидание

outp(0x61, inp(0x61) & 0xFC); // выкл; inp() - чтение значения из порта

delay(countDelay[countHZ]); // ожидание/задержка

}

}

void div()

{

// порт 40h (канал 0, прерывание системных часов)

// порт 41h (канал 1, регенерация памяти)

// порт 42h (канал 2, звук динамика)

// порт 40h (канал 0, прерывании)

int channel; // для переборов каналов

int ports[] = { 0x40, 0x41, 0x42 }; // массив портов для всех каналов

int controlWord[] = { 0x0, 0x40, 0x80 }; // массив команд для чтения состояния регистра счетчика канала

unsigned byte, lowByte, highByte, maxByte; // переменные для чтения регистров счетчика

printf("\nDivision factor: \n");

for (channel = 0; channel < 3; channel++)

{

byte = 0;

maxByte = 0;

for (unsigned long i = 0; i < GO; i++)

{

outp(0x43, controlWord[channel]); // выбор канала

lowByte = inp(ports[channel]); // чтение мл

highByte = inp(ports[channel]); // чтение ст

byte = highByte \* 256 + lowByte; // генерирование

if (byte > maxByte)

maxByte = byte;

}

printf("\nChannel %d: %4X\n", channel, maxByte);

}

}

void statusw()

{

// порт 40h (канал 0, прерывание системных часов)

// порт 41h (канал 1, регенерация памяти)

// порт 42h (канал 2, звук динамика)

int channel, state; // channel - перебор каналов; state - хранение состояния регистра

char stateWord[8]; // хранение двоичного преставления состояния регистра

int ports[] = { 0x40, 0x41, 0x42 }; // массив портов для каждого канала

int controlWord[] = { 226, 228, 232 }; // массив команд для чтения состояния регистра

printf("\nStatus word: \n");

for (channel = 0; channel < 3; channel++)

{

outp(0x43, controlWord[channel]); // выбор канала/запись значения в порт

state = inp(ports[channel]); // чтения значения

for (int i = 7; i >= 0; i--) // создание двоичного значения

{

stateWord[i] = state % 2 + '0';

state /= 2;

}

printf("\nChannel %d: ", channel);

for (i = 0; i < 8; i++)

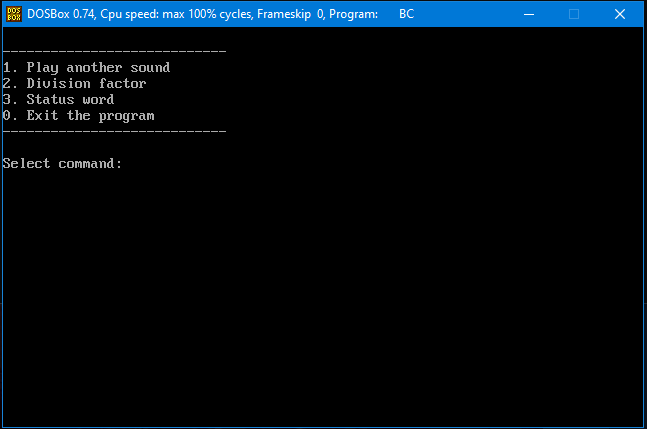
printf("%c", stateWord[i]);

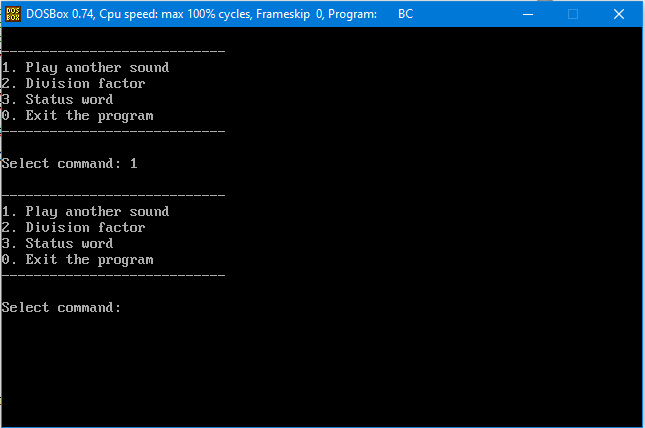
printf("\n");

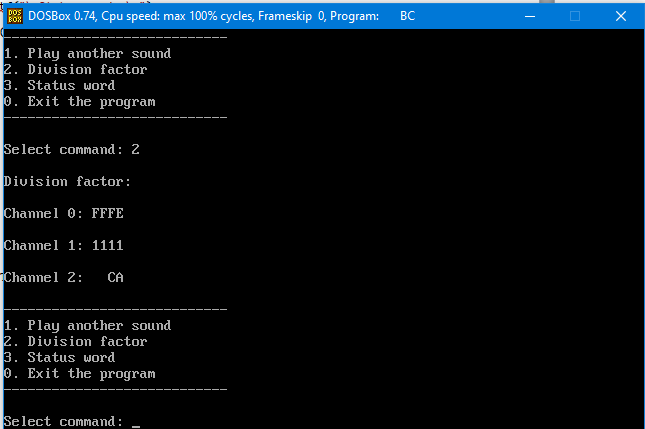
}

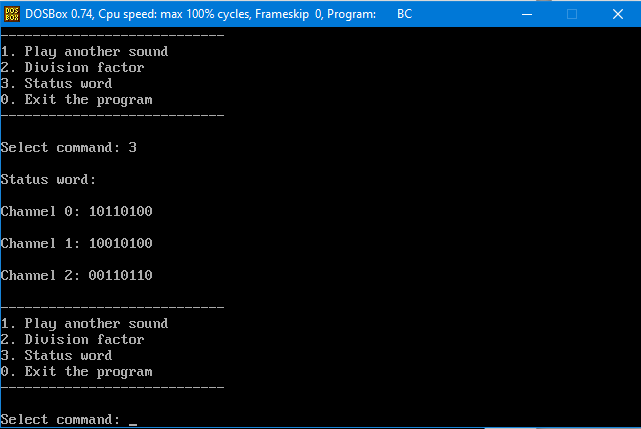
}

**Тест**



****

****

****

#### **Вывод**

Запрограммировал второй канал таймера таким образом, чтобы динамик компьютера издавал звуки с соответственной частотой: 659(200), 622(200), 659(200), 622(200), 659(200), 493(200), 622(200), 523(200), 440(200). Считал для всех каналов таймера слово состояния и вывел его на экран в двоичной форме, а также для всех каналов таймера рассчитал коэффициент деления (значение счетчика CE) и вывел его на экран в шестнадцатеричной форме.