**Лабораторная работа №4.**

**Работа со звуком**

**Теоретические сведения.**

**Извлечение звука в IBM PC**

В простейшем случае в IBM PC для генерации звука используется микросхема интегрального таймера 8253 или 8254. Эта микросхема имеет три независимых канала, каждый из которых может программироваться для работы в режиме делителя частоты или генератора одиночных импульсов. Каждый канал содержит 16-разрядный счетчик, в который записывается значение делителя частоты или коэффициента пересчета (в зависимости от режима работы). Каждый канал имеет вход частоты (clk) и вход разрешения (gate). На вход частоты всех каналов подается импульсный сигнал частотой 1,19 МГц. Канал 0 микросхемы таймера используется для выработки сигнала прерывания по таймеру (частотой 18,2 Гц). Канал 1 работает в режима генерации одиночных импульсов через каждые 15 мкс. Этот сигнал используется для регенерации динамической памяти ЭВМ.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Порт | |  | |  |  | Таймер | |  |  |  |  | |
| 8255 | |  | |  |  | 8253 | |  | Прерывание от | | | |
|  |  |  | |  |  |  |  |  | таймера | | | |
|  |  |  | |  |  | clk0 | out0 |  | 18,2 kHz | | | IRQ0 |
|  |  |  | |  |  | gate0 |  |  | Регенератор памяти | | | |
|  |  |  | |  |  | clk1 | out1 |  | 15 мкс | | | DRQ0 |
|  |  |  | 1,19 |  |  | gate1 |  |  |  |  |  | |
|  |  |  | MHz |  |  | clk2 | out2 |  |  |  |  | |
|  |  |  | |  |  | gate2 |  |  |  |  |  | |
| pb: | 61h |  | |  |  |  |  |  | & | |  | |
|  | pb1 |  | |  |  |  |  |  |  |  | Speaker | |
|  | pb0 |  | |  |  |  |  |  |  |  |  | |
|  |  |  | |  |  |  |  |  |  |  |  | |
|  |  |  | |  |  |  |  |  |  |  |  | |
| Схема ггенерации звука в IBM PC | | | | | | | | | | | | |

Канал 2 микросхемы исходно программируется для работы в режиме делителя частоты. Выход канала используется для генерации звука через встроенный динамик. Для управления звуком используются биты 0 и 1 системного порта В (микросхема 8255). Бит 0 используется для разрешения прохождения сигнала на выход канала 2 таймера. Сигнал с выхода канала 2 подается на схему "И", на второй вход который подается сигнал бита 1 системного порта В. Этот сигнал может разрешать или запрещать прохождения сигнала с выхода канала 2 таймера, а при закрытом канале 2 (битом 0 порта В) сигнал бита 1 порта В может использоваться для непосредственной генерации звука в динамике.

Адрес системного порта В - 61h, адреса каналов таймера - 40h, 41h, 42h, 43h - для каналов 0, 1, 2 и управляющего регистра соответственно. Ниже рассмотрены примеры генерации звука с помощью сигнала бита 1 системного порта В, а также с помощью таймера. Рассмотрен случай извлечения звука с использованием прерывания.

**Примеры генерации звука**

*Задача 2.4.2.1.* Написать программу, издающую различные звуки при нажатии на клавиши '1' и '2'. Для генерации звука следует использовать сигнал бита 1 системного порта В. Выход из программы должен осуществляться по нажатию клавиши 'q'.

Assume CS: Code, DS: Code

Code SEGMENT

org 100h

frequency1 equ 300 ; Задержка переключения 1

frequency2 equ 500 ; Задержка переключения 2

number\_cycles1 equ 1000 ; Количество циклов (длит.)

number\_cycles2 equ 600 ; Количество циклов (длит.)

port\_b equ 61h ; Адрес системного порта В

.286

Start proc near

mov ax,cs

mov ds,ax ; DS = CS

beg1: call kbin ; Опрос клавиатуры

cmp al,'1' ; = '1' ?

jnz beg2 ; Нет

call ton1 ; Звук высоты 1

jmp beg1 ; Переход на начало цикла

beg2: cmp al,'2' ; = '2' ?

jnz beg3 ; Нет

call ton2 ; Звук высоты 2

jmp beg1 ; Переход на начало цикла

beg3: cmp al,'q' ; = 'q' ?

jnz beg1 ; Нет

int 20h ; Выход из программы

start endp

ton2 proc near ; Процедура генерации звука 2

mov dx,number\_cycles2 ; Длительность 2

mov di,frequency2 ; Задержка 2

jmp ton0 ; Переход на универсальную процедуру генерации звука

ton1 proc near ; Процедура генерации звука 1

mov dx,number\_cycles1 ; Длительность 1

mov di,frequency1 ; Задержка 1

; Универсальная процедура генерации звука

; DX - количество циклов, DI - задержка

ton0 proc near

cli ; Запрещение прерываний

in al,port\_b ; Чтение сост. системн. порта В

and al,11111110b ; Отк. динамика от таймера

ton01: or al,00000010b ; Включение динамика

out port\_b,al ; Запись в системный порт В

mov cx,di ; Счетчик цикла задержки

loop $ ; Задержка

; Выключение звука

and al,11111101b ; Выключение динамика

out port\_b,al ; Запись в системный порт В

mov cx,di ; Счетчик цикла задержки

loop $ ; Задержка

dec dx ; Декремент счетчика колич. циклов

jnz ton01 ; Переход на начало нового периода

sti ; Разрешение прерываний

ret ; Выход из процедуры

ton0 endp ; Конец универсальной процедуры

ton1 endp ; Конец процедуры генерации звука 1

ton2 endp ; Конец процедуры генерации звука 2

kbin proc near ; Ввод с клавиатуры с ожиданием

mov ah,0 ; Функция 0

int 16h ; клавиатурного прерывания

ret ; Выход из процедуры

kbin endp ; Конец процедуры ввода с клавиатуры

code ends ; Конец сегмента (кодового)

END Start ; Указание на точку входа

*Задача 2.4.2.2.* Написать программу, издающую различные звуки при нажатии на клавиши '1' и '2'. Для генерации звука следует выход канала 2 таймера. Выход из программы должен осуществляться по нажатию клавиши 'q'.

Assume CS: Code, DS: Code

Code SEGMENT

org 100h

frequency1 equ 1000 ; Коэффиц. деления 1

frequency2 equ 3000 ; Коэффиц. деления 2

duration equ 50000 ; Длительность

port\_b equ 61h ; Адрес системного порта В

.286

Start proc near ; Основная процедура

mov ax,cs

mov ds,ax ; DS = CS

beg1: call kbin ; Опрос клавиатуры

cmp al,'1' ; = '1' ?

jnz beg2 ; Нет

call ton1 ; Звук высоты 1

jmp beg1 ; Переход на начало цикла

beg2: cmp al,'2' ; = '2' ?

jnz beg3 ; Нет

call ton2 ; Звук высоты 2

jmp beg1 ; Переход на начало цикла

beg3: cmp al,'q' ; = 'q' ?

jnz beg1 ; Нет

int 20h ; Выход из программы

start endp ; Конец основной процедуры

ton2 proc near ; Процедура генерации звука 2

mov dx,duration ; Длительность

mov di,frequency2 ; Коэффициент деления 2

jmp ton0 ; Переход на универсальную процедуру

ton1 proc near ; Процедура генерации звука 1

mov dx,duration ; Длительность

mov di,frequency1 ; Коэффициент деления 2

; Универсальная процедура генерации звука

; DX - длительность, DI - коэффиц. деления

ton0 proc near

cli ; Запрещение прерываний

; Включение динамика и таймера

in al,61h ; Чтение состояния системного порта В

or al,3 ; Разрешение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

; Программирование делителя частоты 2 канала

mov ax,di ; Делитель частоты

out 42h,al ; Мл.байт частоты  канал 2 таймера

xchg al,ah ; AH  AL

out 42h,al ; Ст.байт частоты  канал 2 таймера

; Формирование задержки

mov cx,dx ; Счетчик цикла задержки

ton01: push cx ; Команды, используемые только для

pop cx ; увеличения длит. цикла задержки

loop ton01 ; Задержка

; Выключение звука

in al,61h ; Чтение состояния системного порта В

and al,0fch ; Запрещение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

sti ; Разрешение прерываний

ret ; Выход из процедуры

ton0 endp ; Конец универсальной процедуры

ton1 endp ; Конец процедуры генерации звука 1

ton2 endp ; Конец процедуры генерации звука 2

kbin proc near ; Ввод с клавиатуры и проверка на выбор игры

; Процедура совпадает с одноименной в задаче 2.4.2.1.

kbin endp

code ends ; Конец сегмента (кодового)

END Start ; Указание на точку входа

*Задача 2.4.2.3.* Написать программу, издающую различные звуки при нажатии на клавиши '1' и '2'. Для генерации звука следует выход канала 2 таймера. Выход из программы должен осуществляться по нажатию клавиши 'q'. Использовать прерывание от таймера.

Assume CS: Code, DS: Code

Code SEGMENT

org 100h

.286

Start proc near ; Основная процедура

mov ax,cs

mov ds,ax ; DS = CS

jmp beg ; "Перескок" через переменные

frequency equ 0500h ; Коэффициент деления

iniflag db 0 ; Флаг звучания

old\_int1c\_off dw 0 ; Смещение старого вектора

old\_int1c\_seg dw 0 ; Сегмент старого вектора

beg: ; Сохранение старого вектора прерывания 1Сh

mov ax,35h ; Функция взятия вектора

mov al,1ch ; Вектор 1Сh

int 21h ; Вызов функции DOS

mov cs:old\_int1c\_off,bx ; Запись смещения

mov cs:old\_int1c\_seg,es ; Запись сегмента

; Установка в вектор прерывания адреса новой

; программы обработки прерывания

lea dx,new\_int1c ; Запись нового вектора 1c

mov ah,25h ; Функция установки вектора прерыв.

mov al,1ch ; Номер вектора прерывания

int 21h ; DS:DX - адрес новой программы обр.

beg1: call kbin ; Опрос клавиатуры

cmp al,'1' ; = '1' ?

jnz beg2 ; Нет

mov byte ptr iniflag,1 ; Взведение флага звуч.

jmp beg1 ; Переход на начало цикла

beg2: cmp al,'2' ; = '2' ?

jnz beg3 ; Нет

mov byte ptr iniflag,0 ; Сброс флага звуч.

jmp beg1 ; Переход на начало цикла

beg3: cmp al,'q' ; = 'q' ?

jnz beg1 ; Нет

; Восстановление старого вектора 1с и выход

mov dx,old\_int1c\_off ; Смещение старого вектора

mov ax,old\_int1c\_seg ; Сегмент старого вектора

mov ds,ax ; DS:DX - адрес устанавл. вектора

mov ax,251ch ; Установка старого вектора 1ch

int 21h ; Вызов функции DOS

int 20h ; Выход из программы

start endp ; Конец основной процедуры

; Новый обработчик прерывания 1ch

new\_int1c proc far ; Дальняя процедура

pusha ; Сохранение всех регистров (для Intel286)

call muz ; Вызов процедура извлечения звука

popa ; Восстановление всех регистров

iret ; Возврат из программы обработки прерывания

new\_int1c endp ; Конец нового обработчика прерывания 1сh

muz proc near ; Процедура генерации звука

test byte ptr cs:iniflag,0ffh ; Проверка флага

jnz muz1 ; Продолжение

in al,61h ; Чтение системного порта В

and al,0fch ; Запрещение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

ret ; Выход, если флаг не взведен

muz1: ; Программирование делителя частоты 2 канала

mov ax,frequency ; Делитель частоты

out 42h,al ; Мл.байт частоты  канал 2 таймера

xchg al,ah ; AH  AL

out 42h,al ; Ст.байт частоты  канал 2 таймера

; Разрешение звучания

in al,61h ; Чтение системного порта В

or al,3 ; Разрешение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

ret ; Нормальный выход

muz endp ; Конец процедуры генерации звука

kbin proc near ; Ввод с клавиатуры и проверка на выбор игры

; Процедура совпадает с одноименной в задаче 2.4.2.1.

kbin endp

code ends ; Конец сегмента (кодового)

END Start ; Указание на точку входа

*Задача 2.4.2.4.* Написать программу, исполняющую три различные мелодии при нажатии на клавиши '1', '2'и '3'. Для генерации звука следует использовать выход канала 2 таймера. Выход из программы должен осуществляться по нажатию клавиши 'q'. Использовать прерывание от таймера.

Для исполнения мелодии сначала формируется массив делителей частоты, соответствующих различным нотам звукоряда. Для нот используются номера от 1 до 48. Условное соответствие номеров нот и их значений показано на рисунке. Массив делителей частоты для нот называется NOTY. Значение 0 используется в качестве признака окончания мелодии. Значение 255 используется для обозначения паузы.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **2** | |  | **4** | |  |  | **7** | |  | **9** | |  | **11** | |  |  | **14** | |  | **16** | |  |  | **19** | |  | **21** | |  | **23** | |  |  | **26** | |  | **28** | |  |
| **1** | | **3** | | | **5** | | **6** | | **8** | | | **10** | | | **12** | | **13** | | **15** | | | **17** | | **18** | | **20** | | | **22** | | | **24** | | **25** | | **27** | | | **29** | |
| Номера нот и их соответствие | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

В программе сформированы массивы для трех мелодий - "Чижик Пыжик" (mel1), "Подмосковные вечера" и " Кан-кан" (mel3). Длительность элементарного звука берется равной одному "тику" таймера (примерно 1/18 сек.). Для получения большей длительности в массиве мелодии записываются подряд несколько одинаковых нот. Для получения четко выраженных соседних одинаковых нот используется пауза.

Assume CS: Code, DS: Code

Code SEGMENT

org 100h

.286

Start proc near ; Основная процедура

mov ax,cs

mov ds,ax ; DS = CS

jmp beg ; "Перескок" через переменные

iniflag db 0 ; Флаг звучания

old\_int1c\_off dw 0 ; Смещение старого вектора

old\_int1c\_seg dw 0 ; Сегмент старого вектора

tek\_mel dw ? ; Адрес текущей ноты выбранной мелодии

beg: mov ax,351ch ; Сохранение старого вектора 1c

int 21h

mov cs:old\_int1c\_off,bx ; Запись смещения

mov cs:old\_int1c\_seg,es ; Запись сегмента

lea dx,new\_int1c ; Запись нового вектора 1c

mov ah,25h ; Функция установки вектора прерыв.

mov al,1ch ; Номер вектора прерывания

int 21h ; DS:DX - адрес новой программы обр.

beg1: call kbin ; Опрос клавиатуры

cmp al,'1' ; = '1' ?

jnz beg2 ; Нет

mov byte ptr iniflag,1 ; Взведение флага звуч.

lea ax,mel1

mov tek\_mel,ax

jmp beg1 ; Переход на начало цикла

beg2: cmp al,'2' ; = '2' ?

jnz beg3 ; Нет

mov byte ptr iniflag,1 ; Сброс флага звуч.

lea ax,mel2

mov tek\_mel,ax

jmp beg1 ; Переход на начало цикла

beg3: cmp al,'3' ; = '3' ?

jnz beg4 ; Нет

mov byte ptr iniflag,1 ; Сброс флага звуч.

lea ax,mel3

mov tek\_mel,ax

jmp beg1 ; Переход на начало цикла

beg4: cmp al,'q' ; = 'q' ?

jnz beg1 ; Нет

; Восстановление старого вектора 1с и выход

mov dx,old\_int1c\_off ; Смещение старого вектора

mov ax,old\_int1c\_seg ; Сегмент старого вектора

mov ds,ax ; DS:DX - адрес устанавл. вектора

mov ax,251ch ; Установка старого вектора 1ch

int 21h

int 20h

start endp

; Новый обработчик прерывания 1ch

new\_int1c proc far ; Дальняя процедура

pusha ; Сохранение всех регистров

call muz ; Вызов процедура извлечения звука

popa ; Восстановление всех регистров

iret ; Возврат из программы обработки прерывания

new\_int1c endp

; Процедура извлечения очередного звука

; tek\_mel - адрес текущей ноты выбранной мелодии

muz proc near

test byte ptr cs:iniflag,0ffh ; Проверка флага

jnz muz1 ; Продолжение

muze: in al,61h ; Чтение состояния системного порта В

and al,0fch ; Запрещение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

ret ; Выход, если флаг не взведен

muz1: mov si,cs:tek\_mel ; Адрес текущей ноты

mov bl,cs:[si] ; Текущая нота

cmp bl,255 ; Пауза ?

jnz muz2

; Выключение звука

in al,61h ; Чтение состояния системного порта В

and al,0fch ; Запрещение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

inc cs:tek\_mel ; Переход к адресу след. ноты

ret

muz2: or bl,bl ; = 0 ?

jnz muz3 ; Продолжение

jmp muze ; Выход, если признак конца

muz3: shl bl,1 ; Умножение bl на 2

xor bh,bh ; bh = 0

mov ax,cs:noty[bx] ; В DI частота ноты

; Программирование делителя частоты 2 канала

inc cs:tek\_mel ; Переход к адресу след. ноты

out 42h,al ; Мл.байт частоты  канал 2 таймера

xchg al,ah ; AH  AL

out 42h,al ; Ст.байт частоты канал 2 таймера

; Разрешение звучания

in al,61h ; Чтение состояния системного порта В

or al,3 ; Разрешение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

ret ; Нормальный выход

muz endp

kbin proc near ; Ввод с клавиатуры и проверка на выбор игры

; Процедура совпадает с одноименной в задаче 2.4.2.1.

kbin endp

; Мелодия "Чижик Пыжик"

mel1 db 17,17,255,13,13,255,17,17,255,13,13,255,18,18,255

db 17,17,255,15,15,15,15,255,255

db 8,8,255,8,8,255,8,8,255,10,255,12,255

db 13,13,255,13,13,255,13,13,13,13

db 0

; Мелодия "Подмосковные вечера"

mel2 db 1,1,1,4,4,4,8,8,8,4,4,4,6,6,6,6,6,6,4,4,4,3,3,3

db 8,8,8,8,8,8,6,6,6,6,6,6,1,1,1,1,1,1,1,1,1,1,1,1

db 0

; Мелодия "Кан-кан"

mel3 db 18,6,25,13,22,18,25,13,20,1,23,8,22,5,20,1

db 25,6,13,13,25,10,13,13,25,10,27,13,22,10,23,13,20,1

db 11,11,20,5,11,11,20,1,23,11,22,5,20,11,18,6,30,18

db 29,17,27,15,25,13,23,11,22,10,20,8,18,6,13,13,18,10

db 13,13,20,1,23,8,22,5,20,8,25,6,13,13,25,10,13,13

db 25,6,27,13,22,10,23,13,20,1,8,8,20,5,8,8,20,1,23,8

db 22,5,20,8,18,6,25,10,20,13,22,10,18,6,6,6,6,6,6,6

db 34,8,24,12,24,15,34,12,32,1,25,5,25,8,29,5,30,6,34,13

db 37,10,34,13,34,1,32,8,32,5,8,8,34,8,24,12,24,15,34,12

db 32,1,25,5,25,8,29,5,29,3,27,7,29,10,27,13,34,12,32,8

db 34,6,32,3,34,8,24,15,24,12,34,15,32,1,29,8,25,5,29,8

db 30,6,34,13,37,10,34,13,34,1,32,5,32,8,5,5,34,8,24,15

db 24,12,34,15,32,1,25,6,25,5,29,8,29,3,27,7,29,10,27,7

db 32,8,30,6,29,5,27,3,25,1,8,8,25,5,8,8,27,12,30,15

db 29,8,27,12,32,1,8,8,32,5,8,8,32,1,34,8,29,5,30,8,27,8

db 15,15,27,12,15,15,27,8,30,12,29,15,27,12,25,1,37,1

db 36,5,34,6,32,8,30,8,29,10,27,12,25,1,8,8,25,5,8,8

db 27,8,30,15,29,12,27,15,32,1,8,8,32,5,8,8,32,1,34,8

db 29,5,30,8,27,8,15,15,27,12,15,15,27,8,30,15,29,12

db 27,15,25,1,32,8,27,5,29,8,25,1,32,32,37,37,0

; Коэффициенты деления для нот

noty dw 0eeeh,0e18h,0d49h,0c8eh,0bdfh,0b2fh,0abeh

dw 9f7h,968h,8e0h,861h,7e8h,777h,70ch,6a5h,647h

dw 5edh,597h,547h,4fbh,4b4h,470h,430h,3f4h

dw 3bbh,386h,352h,323h,2f6h,2cbh,2a3h,27dh,25ah,238h,218h,1fah

dw 1ddh,1c3h,1a9h,192h,17bh,166h,152h,13fh,12dh,11ch,10ch,0fdh

dw 0

code ends

END Start

*Задача 2.4.2.5.* Модифицировать программу из предыдущей задачи так, чтобы длительность каждого звука задавалась в массиве мелодии. Для каждого звука должна записываться пара значений: номер ноты и длительность, выраженная в элементарных "тиках" таймера. Для простоты в программе будут реализованы только первые две короткие мелодии.

Assume CS: Code, DS: Code

Code SEGMENT

org 100h

.286

Start proc near

mov ax,cs

mov ds,ax

jmp beg

pausa equ 255

frequency dw 1000h

iniflag db 0 ; Флаг звучания

old\_int1c\_off dw 0 ; Смещение старого вектора

old\_int1c\_seg dw 0 ; Сегмент старого вектора

tek\_mel dw ?

duration db 0 ; Длительность текущей ноты

beg: mov ax,351ch ; Сохранение старого вектора 1c

int 21h

mov cs:old\_int1c\_off,bx ; Запись смещения

mov cs:old\_int1c\_seg,es ; Запись сегмента

lea dx,new\_int1c ; Запись нового вектора 1c

mov ah,25h ; Функция установки вектора прерыв.

mov al,1ch ; Номер вектора прерывания

int 21h ; DS:DX - адрес новой программы обр.

beg1: call kbin ; Опрос клавиатуры

mov byte ptr duration,1

cmp al,'1' ; = '1' ?

jnz beg2 ; Нет

mov byte ptr iniflag,1 ; Взведение флага звуч.

lea ax,mel1

mov tek\_mel,ax

jmp beg1 ; Переход на начало цикла

beg2: cmp al,'2' ; = '2' ?

jnz beg3 ; Нет

mov byte ptr iniflag,1 ; Сброс флага звуч.

lea ax,mel2

mov tek\_mel,ax

jmp beg1 ; Переход на начало цикла

beg3: cmp al,'q' ; = 'q' ?

jnz beg1 ; Нет

; Восстановление старого вектора 1с и выход

mov dx,old\_int1c\_off ; Смещение старого вектора

mov ax,old\_int1c\_seg ; Сегмент старого вектора

mov ds,ax ; DS:DX - адрес устанавл. вектора

mov ax,251ch ; Установка старого вектора 1ch

int 21h

int 20h

start endp

; Новый обработчик прерывания 1ch

new\_int1c proc far

pusha

dec byte ptr cs:duration

jnz ex

call muz ; Вызов процедуры извлечения звука

ex: popa

iret

new\_int1c endp

muz proc near

test byte ptr cs:iniflag,0ffh ; Проверка флага

jnz muz1 ; Продолжение

muze: in al,61h ; Чтение состояния системного порта В

and al,0fch ; Запрещение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

ret ; Выход, если флаг не взведен

muz1: mov si,cs:tek\_mel ; Адрес текущей ноты

mov bx,word ptr cs:[si] ; BL -текущая нота, BH - длительность

mov cs:duration,bh ; Длит. в системную переменную

cmp bl,255 ; Пауза ?

jnz muz2

; Выключение звука

in al,61h ; Чтение состояния системного порта В

and al,0fch ; Запрещение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

inc cs:tek\_mel ; Переход к адресу след. ноты

inc cs:tek\_mel ; след. длительности

ret

muz2: or bl,bl ; = 0 ?

jnz muz3 ; Продолжение

jmp muze ; Выход, если признак конца

muz3: shl bl,1 ; Умножение bl на 2

xor bh,bh ; bh = 0

mov ax,cs:noty[bx] ; В DI частота ноты

; Программирование делителя частоты 2 канала

inc cs:tek\_mel ; Переход к адресу след. ноты

inc cs:tek\_mel ; и след. длительности

out 42h,al ; Мл.байт частоты канал 2 таймера

xchg al,ah ; AH AL

out 42h,al ; Ст.байт частоты  канал 2 таймера

; Разрешение звучания

in al,61h ; Чтение состояния системного порта В

or al,3 ; Разрешение звучания (биты 0 и 1)

out 61h,al ; Запись в системный порт В

ret ; Нормальный выход

muz endp

kbin proc near ; Ввод с клавиатуры

; Процедура совпадает с одноименной в задаче 2.4.2.1.

kbin endp

mel1 db 17,2,255,1,13,2,255,1,17,2,255,1,13,2,255,1,18,2,255,1, 17,2

db 255,1,15,4,255,2, 8,2,255,1,8,2,255,1,8,2,255,1,10,1,255,1

db 12,1,255,1,13,2,255,1,13,2,255,1,13,4,0

mel2 db 1,4,4,4,8,4,4,4,6,8,4,4,3,4,8,8,6,8,1,12,0

noty dw 0eeeh,0e18h,0d49h,0c8eh,0bdfh,0b2fh,0abeh, 9f7h,968h

dw 8e0h,861h,7e8h,777h,70ch,6a5h,647h, 5edh,597h,547h

dw 4fbh,4b4h,470h,430h,3f4h, 3bbh,386h,352h,323h,2f6h,2cbh

dw 2a3h,27dh,25ah,238h,218h,1fah, 1ddh,1c3h,1a9h,192h,17bh

dw 166h,152h,13fh,12dh,11ch,10ch,0fdh,0

code ends

END Start