МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по лабораторной работе №3

«Организация параллельного обмена микро-ЭВМ и ПУ по прерываниям

с применением контроллера К580ВН59»

по дисциплине

Интерфейсы периферийных устройств

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_ Киселёв Ю.Н.

СТУДЕНТЫ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Игнаков К.М.

\_\_\_\_\_\_\_\_\_\_\_\_ Тихонов В.А.

19-В-2

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

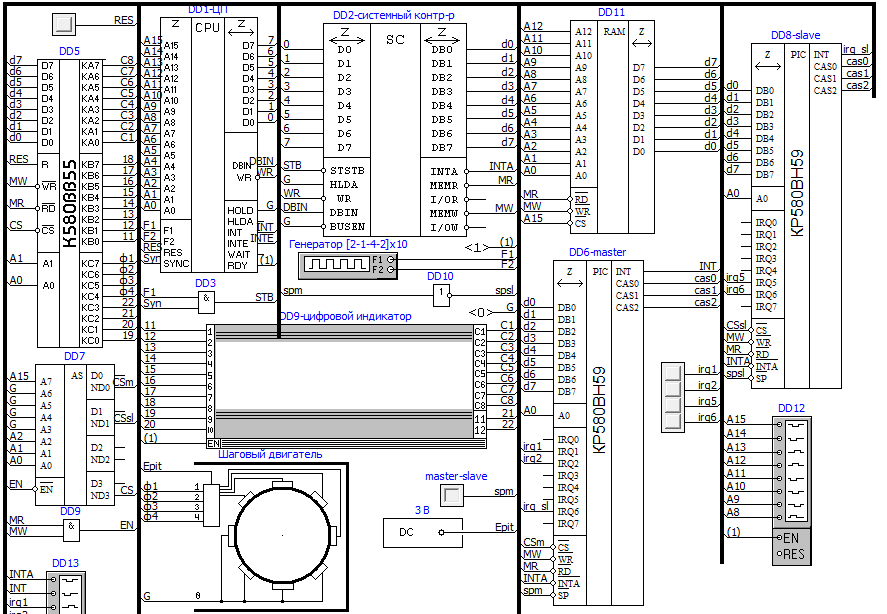
**Цель работы:**

Используя заготовку, где фоновой программой является вывод слова (Вася), реализовать:

Прием и обработку n-запросов прерываний (ЗПР) по входам IRQ (1,3,5,6) в режиме с фиксированными приоритетами, где 1 и 3 – номера входов мастера, 5 и 6 – номера входов ведомого. Запросы мастера более приоритетны. В общем с памятью адресном пространстве.

Подпрограммы обслуживания запросов должны выводить фамилию исполнителя и время (число месяц год) выполнения работы на индикатор - два любых запроса в режиме бегущей строки, остальные два – управлять шаговым двигателем, заставляя ротор вращаться по часовой или против часовой стрелки.

**Схема микро-ЭВМ:**

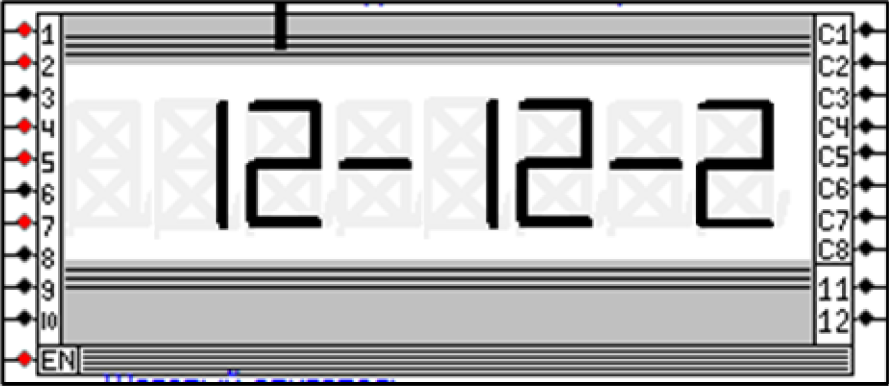


**Контрольный пример:**

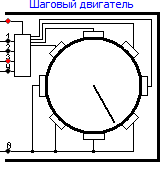
Вывод фамилии на индикатор:



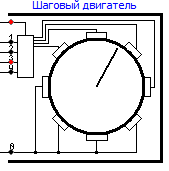
Вывод времени выполнения работы на индикатор:



Вращение шагового двигателя по часовой стрелке:

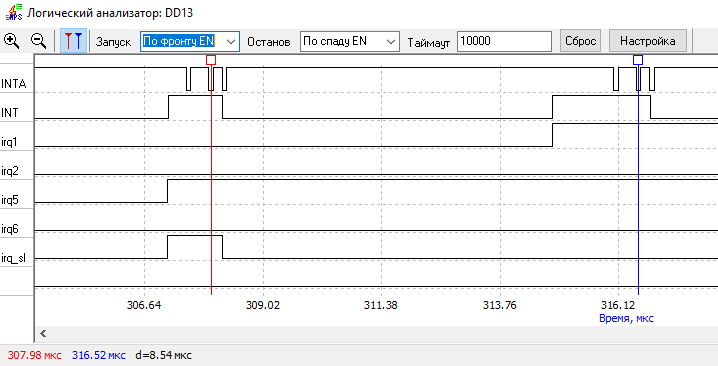


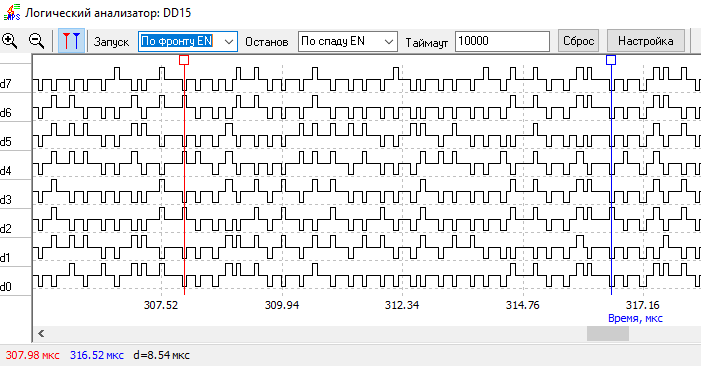
Вращение шагового двигателя против часовой стрелке:



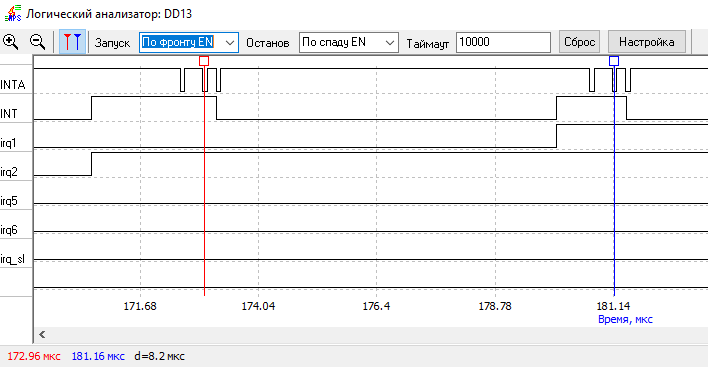
**Временные диаграммы:**

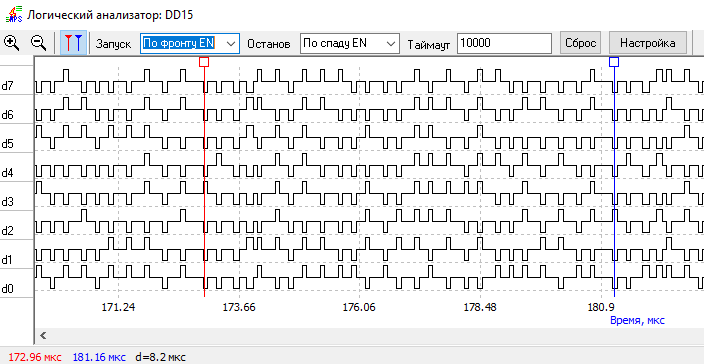
Проверка приоритетности шагового двигателя по запросу irq5, при запуске приоритетного irq1 c мастера, появляется сигнал INT.



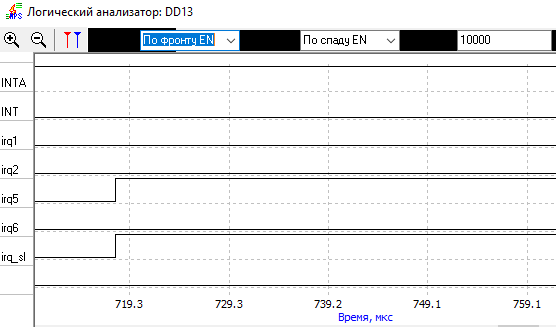


Проверка приоритетности irq1. Запрос irq3 прерывается при запуске irq1 и появляется сигнал INT.





Запрет обработки прерывания на линии irq5.



**Вывод:**

В ходе работы мы ознакомились с программной моделью микропроцессора КР580ВН59, пакетом «Моделирование цифровых систем» и программированием простейших задач в кодах процессора. Мы собрали схему, написали подпрограмму обработки прерываний согласно нашему заданию и протестировали.

**Код программы:**

Org 0

; В этом ПО предполагается что в схеме примера запросы

;на \*master\* поступают на входы IRQ1 and IRQ2 от клавиатуры

;и на IRQ6 от \*slave\*. Обьем памяти 8кб. Для задач данной л.р.

;более чем достаточный. Запросы на slave поступают на

;IRQ5,IRQ6.

;Обычно таблицы переходов по прерываниям размещают на

;самом верху памяти, ниже стек и

;все остальное. Обьем таблиц зависит от числа входов системы

; прерываний и величины сегментов. Базовый адрес по млашему байту

;задаетя в icw1 для интервала 4 это разряды A7A6A5, остальные

;будут нулями - обнуляет сам контроллер. Для интервала 8 только A7A6.

;По этому для 000(интервал 4) мл.байт адреса будет 00h(ICW1 - 14h),

;для 001 - 20h(ICW1- 34h), 010 - 40h\*(ICW1 - 54h)

;011- 60h, 100 - 80h, 101 - A0h, 110 - C0h,111- E0h

;В данном примере для простоты положим базовый адрес \*mastera\* - адрес сегмента по IRQ0

;равным 1000h, величину сегмента 4 байта. Адрес сегмента по

; IRQ1 - 1004h, IRQ2-1008h,

; базовый адрес \*slave\*-1040h, адрес сегмента по IRQ5sl-1054h,

;по IRQ6sl-1058h

lxi sp,0ff0h

mvi a,80h

sta 8003h ;порты ВВ55 на вывод в режиме 0

; программирование мастера на базовый адрес 1000h

mvi a, 14h ; ICW1, интервал таблицы 4 байта, А7-А0 = 00h

sta 8004h

mvi a, 10h ;ICW2

sta 8005h

mvi a, 40h ;ICW3,запрос от ведомого подан на IRQ6

sta 8005h

;программирование ведомого на базовый адрес 1040h

mvi a,54h ;ICW1

sta 8006h

mvi a, 10h ;ICW2

sta 8007h

mvi a, 06h ;ICW3 ,6h это 00000110 - код входа мастера куда подан запрос от ведомого

sta 8007h

; Фоновая прога \*Привет\*

;Состав схемы - индикатор и ШД. Адреса регистров индикатора, ШД

;в общем с памятью пространстве,

jmp mmm ;переход на фоновую (основную)прогу

; Фоновая прога

:mmm

;PP2- выдержка длительности отбражения символа и гашения индикатора,

; lab -адрес 1-го символа строки, n - количество символов в строке

;d -номер текущей позции индикатора

const N 0b0h ; N=nx8, n - длина строки, \*e\* - счетчик отображенных сиволов

:m6 ; повтор вывода на индикатор всей фразы

mvi e,0h

mvi a,00h

sta 8000h ; гасим индикатор

mvi d,01h ; задаем 1-ю позицию в индикаторе

lxi h,lab ; читаем адрес мл.байта 1-го символа в строке

:m2 ;начало отображения 8-и символов на индикаторе

push h ; адрес мл.байта 1-го символа в строке сохраняем в стеке

:m

mov a,m

sta 8001h ;занесли мл.байт в регистр

inx h ;сформировали адрес ст.байта символа

mov a,m

sta 8002h ;занесли ст.байт в регистр

mov a,d ;задаем позицию отображения символа

sta 8000h ;установка позиции - зажгли букву

mov d,a ; сохраняем текущую позицию в регистре D, т.к. аккумулятором еще будем пользоваться

call pp2 ; выдержка времени отображения и гашение индикатора

inr e

mov a,d ;восстанавливаем значение текущей позиции отображенного символа для сравнения с мах-80h

cpi 80h ;сравнение дв.кода номера текущей позиции с макс

jz m5 ;если 0-отображено 8 символов и надо проверить все ли символы строки выведены,а если не 0

;надо сдвинуть указатель адреса на следующий символ в строке и продолжить их вывод

;на индикатор в следующие позиции

rlc ;сдвиг - новая позиция

mov d,a ; сохраняем новое значение позиции в D

inx h ;сформировали адрес мл.байта сдующего символа

jmp m ;крутимся для отображения всех 8-ми символов

:m5 ; вывелено очередные 8 символов

inx h ;готовом следующий адрес

mvi d,01h ;готовим 1-ю позицию индикатора

mov a,e ; номер следующего символа

cpi N ;проверка вывода всей фразы

jz m6 ; если N меньше Е,фпаза отобпажена и повтор вывода на индикатор всей фразы

;если выведены не все символы строки надо увеличить указатель на 2 и продолжить

pop h ;восстанавливаем адрес мл.байта 1-го символа и увеличиваем

inx h

inx h ; 2

ei ; разрешаем прерывания

nop

jmp m2 ;фраза не выведена

; Строка символов для отображения

:lab

dw 0h,0h,0h,0h,0h,0h,0h,0037h,0073h,0436h,0939h,0079h,0007h,0031h,0000h,0031h,0073h,0080h,0006h,006fh,0040h,0939h,0130h,0206h,0h,0h,0h,0h,0h,0h,0h ; Привет гр. 19-В

:pp2 ; выдержка и гашение

mvi a,02h

:m11

dcr a

cpi 00h

jnz m11

mvi a,0h

sta 8000h ; гашение индикатора

ret

:irq1 ;моя фамилия - прерывание от Мастера

ei

jmp my\_family

:irq1\_back

di

mvi a, 20h

sta 8004h

sta 8006h

mvi a,0h

ei

ret

:irq2 ; вывод даты

;всё сохраняем

ei

jmp second\_family

:irq3\_back

di

mvi a, 20h

sta 8004h

sta 8006h

mvi a,2h

ei

ret

:irq5sl ; прога управления ШД

;всё сохраняем

mvi c,30h

ei ;

mvi a, 20h

sta 8006h ;сбросил запрос в РОЗП

:povtor

mvi a, 10h ;

sta 8002h ;подали сигнал в обмотку 4(фаза 4)

call temp ;длительность сигнала

mvi a, 20h

sta 8002h

call temp

mvi a, 40h

sta 8002h

call temp

mvi a, 80h

sta 8002h

call temp

dcr c

jnz povtor

;всё восстанавливаем

ei

ret

:temp

nop

nop

nop

nop

nop

nop

nop

nop

nop

ret

:irq6sl

;всё сохраняем

mvi c,30h

ei ;

mvi a, 20h

sta 8006h ;сбросил запрос в РОЗП

:povtor2

mvi a, 80h ;

sta 8002h ;подали сигнал в обмотку 4(фаза 4)

call temp ;длительность сигнала

mvi a, 40h

sta 8002h

call temp

mvi a, 20h

sta 8002h

call temp

mvi a, 10h

sta 8002h

call temp

dcr c

jnz povtor2

;всё восстанавливаем

ei

ret

:my\_family

:m1

mvi l,1h

mvi b,0FEh

:p

mov a,l

sbi 8

jm q

mvi e,8

mov b,l

jmp q1

:q

mov e,l

mov b,e

:q1

mvi d,1

dcr b

mov a,b

rlc

mov b,a

mvi a,80h

:r

push h

push b

sub a

sta 8000h

mvi h,0

mov l,b

lxi b,str

dad b

mov b,h

mov c,l

ldax b

mov l,a

inx b

ldax b

mov h,a

shld 8001h

mov a,d

rrc

mov d,a

sta 8000h

pop b

pop h

dcr e

jz r2

dcr b

dcr b

jmp r

:r2

inr l

mov a,l

cpi 15

jnz p

jmp irq1\_back

;end my\_family

:second\_family

:m12

mvi l,1h

mvi b,0FEh

:p2

mov a,l

sbi 8

jm qq

mvi e,8

mov b,l

jmp q12

:qq

mov e,l

mov b,e

:q12

mvi d,1

dcr b

mov a,b

rlc

mov b,a

mvi a,80h

:rr

push h

push b

sub a

sta 8000h

mvi h,0

mov l,b

lxi b,str2

dad b

mov b,h

mov c,l

ldax b

mov l,a

inx b

ldax b

mov h,a

shld 8001h

mov a,d

rrc

mov d,a

sta 8000h

pop b

pop h

dcr e

jz r22

dcr b

dcr b

jmp rr

:r22

inr l

mov a,l

cpi 16

jnz p2

jmp irq3\_back

;end 2nd\_family

str dw 0007h,0031h,0436h,0600h,0900h,003Fh,0076h,003Fh,0571h,000h,0571h,0436h,0007h,0031h,0467h, 0h,0h,0h,0h,0h,0h,0h,0h ; Тихонов

str2 dw 0006h,005Bh,0040h,0006h,005Bh,0040h,005Bh,003Fh,005Bh,005Bh, 0h,0h,0h,0h,0h,0h,0h,0h ; 12-12-2022

; оформление таблицы переходов по прерываниям (рекомндуется таблицу размещать на

;самом верху памяти чтобы не создавать пробдем с нехваткой ячеек ЗУ под основное ПО)

;Таблицы создаем оператором Ассемблера - Skip

skip 1004h ;скачёк к адресу 1004h оператором ассемблера

jmp irq1 ;запись команды перехода в ячейки ОЗУ с адресами 1004h - 1006h

skip 1008h

jmp irq2 ;запись команды перехода в ячейки ОЗУ с адресами 1008h - 100Ah

skip 1054h

jmp irq5sl ;запись команды перехода в ячейки ОЗУ с адресами 1054h - 1047h

skip 1058h

jmp irq6sl ;запись команды перехода в ячейки ОЗУ с адресами 1058h - 103Eh