МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ**

о выполнении лабораторной работы № 4

по дисциплине «Системы реального времени»

Выполнил: ст. гр. 45/2

Посевин Р. Р.

Проверил: доц. каф. ИТ

Полетайкин А.Н.

Краснодар

2021

**Лабораторная работа №4**

**Тема:** Программирование ввода-вывода в режиме реального времени.

**Цель:** Изучение принципов организации ввода информации извне в УВМ и вывода информации из УВМ вовне, организации временной задержки при обработке данных на языке ассемблера, а также приобретение практических навыков программирования указанных операций.

**Задание**

Вариант 27.

Имеется СРВ, включающая в себя некоторую аппаратную часть периферийных устройств (ПУ) и ядро в виде УВМ (рис. 1), которая осуществляет обмен с периферией через один 16-разрядный порт ввода с адресом АIN=8B5h и один 8-разрядный порт вывода с адресом АOUT=8B6h. Входные 8-разрядные данные поступают на младший байт порта ввода. Для актуализации входных данных необходимо предварительно выводить через порт вывода заданное управляющее слово режима работы (УСРР). Отправление УСРР тактируется высоким уровнем сигнала через заданный разряд RC=13 порта ввода. Начальное значение УСРР – заданная комбинация U0=0Ah. При каждом следующем запросе данных УСРР меняется на приращение U=0Dh. Вследствие аппаратных временных затрат периферийной части актуализация данных происходит через заданный интервал времени =103 мс.



Рис. 1. Структурная схема СРВ

Составить программу на языке ассемблера, которая осуществляет опрос внешних устройств через порт ввода и записывает отправленные УСРР и соответствующие им данные о состоянии периферийных устройств в одномерные массивы длины N=18. Реализацию задержки времени =103 мс осуществлять при помощи подпрограммы DELAY. Блок-схема алгоритма управляющей программы показана на рис. 6. Заданные параметры СРВ представлены в Таблице 1.

Таблица 1 — Заданные параметры работы СРВ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер варианта | Адреса портов | | Параметры УСРР | | Параметры синхронизации | | |
| АIN | АOUT | U0 | U | , мс | *RC* | N |
| 27 | 085h | 086h | B0h | 08h | 201 | 13 | 8 |

Ниже представлен код программы на языке Ассемблер.

.686

include \masm32\include\io.asm

.data

U DB 0B0h

A DB 8 DUP(0)

.code

DELAY:

MOV BX, 4h

L1:

MOV CX, 0F2h

NOP

NOP

NOP

NOP

L2:

NOP

NOP

NOP

NOP

NOP

NOP

LOOP L2

DEC BX

JNZ L1

RET

ONE\_MSEC:

MOV DX, 100

L3:

CALL DELAY

DEC DX

JNZ L3

RET

DELAY\_201:

MOV DX, 201

L4:

PUSH DX

CALL ONE\_MSEC

POP DX

DEC DX

JNZ L4

RET

LStart:

MOV CX, 8

XOR BX, BX

cycle:

read\_label:

IN AX, 085h

AND AX, 0001000000000000b

JZ read\_label

MOV AL, U

OUT 086h, AL

PUSH CX

CALL DELAY\_201

POP CX

MOV A[BX], AL

INC BX

IN AX, 085h

MOV A[BX], AL

INC BX

ADD U, 08h

LOOP cycle

MOV AX, 4C00h

INT 21h

end LStart

Создаем пустой массив для данных нужной размерности (в данном варианте N=8, так что размер массива 8). Задача состоит в том, чтобы заполнить созданный массив данными УСРР.

В начале получаем данные с порта ввода с заданным адресом АIN =085h, в этих полученный данных смотрим, есть ли 1 на 13 бите, если 0, то вводим пока не получим 1. Далее с помощью порта порт вывода с заданным адресом АOUT  =086h выводим УСРР. Дальше используем ожидание актуализации данных о состоянии ПУ СРВ с помощью написанной функции DELAY, где ожидаем 201мс. После ожидания выводим состояние ПУ СРВ. Записываем в массив A U0, затем делаем приращение Ud к U0 и идём опять по циклу, пока массив не будет заполнен полностью.

Опишем функцию задержки времени.

DELAY:

MOV BX, 4h

L1:

MOV CX, 0F2h

NOP

NOP

NOP

NOP

L2:

NOP

NOP

NOP

NOP

NOP

NOP

LOOP L2

DEC BX

JNZ L1

RET

ONE\_MSEC:

MOV DX, 100

L3:

CALL DELAY

DEC DX

JNZ L3

RET

DELAY\_201:

MOV DX, 201

L4:

PUSH DX

CALL ONE\_MSEC

POP DX

DEC DX

JNZ L4

RET

При тактовой частоте процессора 3,4 ГГц длительность одного такта процессора приблизительно составляет 0,294 нс. При этом число тактов подпрограммы DELAY составит:

Т=29+4+4\* (4\*3+4+242\*6\*3+241\*17+5+3+16)–16+4+20=34 013 тактов

а длительность выполнения подпрограммы DELAY:

Т=34013\*0,294=9999,82 нс ~= 10 мкс.

Для реализации задержки в 1 мс используется подпрограмма ONE\_MSE. Она вызывает подпрограмму DELAY 100 раз. Подпрограмма DELAY\_201 вызывает подпрограмму ONE\_MSE 201 раз, создавая задержку в 201 мс.

На рисунке 1 изображен дамп памяти в конце работы программы.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 — Дамп памяти в конце работы программы

## Вывод

В данной работе были изучены принципы организации ввода информации извне в УВМ и вывода информации из УВМ вовне, принципы организации временной задержки при обработке данных на языке ассемблера, а также приобретены практические навыки программирования указанных операций.