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:**

Имеется СРВ, включающая в себя некоторую аппаратную часть периферийных устройств (ПУ) и ядро в виде УВМ (рис. 4.2), которая осуществляет обмен с периферией через один 16-разрядный порт ввода с заданным адресом АIN и один 16-разрядный порт вывода с заданным адресом АOUT. 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 порта ввода. Начальное значение старшего байта УСРР (УСРР[1]) – заданная комбинация U0. При каждом следующем запросе данных значение УСРР[1] меняется на заданное приращение ΔU. Вследствие аппаратных временных затрат периферийной части актуализация данных происходит через заданный интервал времени τ.

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рисунок 1 – Структурная схема СРВ

Младший байт УСРР (УСРР[0]) подвергается побитовой модификации следующим образом:

* на каждой нечётной итерации заданные биты SR устанавливаются, а биты RS сбрасываются;
* на каждой чётной итерации заданные биты SR сбрасываются, а биты RS устанавливаются.

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 заданной длины N. Реализацию задержки времени τ осуществлять при помощи подпрограммы DELAY.

**Ход работы:**

В качестве заданных параметров работы СРВ в варианте 29 предлагаются следующие: A\_IN = 089h, A\_OUT = 08Ah, U\_0 = D0h, *Δ*U = 04h, SR = 2/6, RS = 0, τ = 222 (мс), R\_C = 15, N = 10.

1. Для предоставления процедуры расчета времени задержки при составлении подпрограммы DELAY использовалась следующая информация: при тактовой частоте процессора 3,4 ГГц длительность одного такта процессора приблизительно составляет 0,294 нс; формула поиска числа тактов, за которое выполняется подпрограмма: Т= TCALL+ТMOV+It\_vn\*(ТMOV+250⋅6\*ТNOP+249⋅TLOOP\_with\_jump+TLOOP\_without\_jump+

+TDEC+TJNZ\_with\_jump) - TJNZ\_with\_jump+TJNZ\_without\_jump+TRET.

В таком случае, для осуществления задержки в 222 мс (222 000 000 нс) необходимо, чтобы процессор сделал . Подставим в формулу след значения: TCALL = 29 , ТMOV = 4 , ТNOP = 3 , TLOOP\_with\_jump = 17, TLOOP\_without\_jump = 5, TDEC = 3, TJNZ\_with\_jump = 16,TJNZ\_without\_jump= 4,TRET = 20, T= 755 102 041, x – количество итераций внутреннего цикла, It\_vn – внешнего цикла. Получим:

29 + 4 + It\_vn \* (4\*3 + 6\*3\*x + (x - 1)\*17 + 5 + 3 +16) – 16 + 4 + 20 = 755 102 041

Определив методом подбора It\_vn = 510, получим x 42302. Подставим найденные значения:

41 + 510 \* (4\*3 + 6\*3\*42302 + (42302 - 1)\*17 + 5 + 3 +16) = 755102471.

Переведем полученное значение в мс: = 222,000126474 мс. Ниже представлен код подпрограммы задержки на 222 мс.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DELAY: | | MOV BX, 510 | | | ; задаётся 19 533 200 итераций двойным циклом | | |
| L1: | MOV CX, 42302 | | ; 4 такта | | |
|  | NOP | |  | | |
|  | NOP | |  | | |
|  | NOP | |  | | |
|  | NOP | |  | | |
| L2: | NOP ; 3 такта | |  | | |
|  | NOP | |  | | |
|  | NOP | |  | | |
|  | NOP | |  | | |
|  | NOP | |  | | |
|  | NOP | |  | | |
|  | LOOP L2 | | ; 5 тактов без перехода, 17 тактов с переходом | | |
|  | DEC BX | | ; 3 такта | | |
|  | JNZ L1 | | ; 4 такта без перехода, 16 тактов с переходом | | |
|  | RET | | | ; 20 тактов | | |

1. Была составлена программа 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 заданной длины *N*. Реализация задержки времени τ осуществлялась при помощи подпрограммы DELAY.

Код был построен на основе следующей схемы:

Изображение выглядит как текст, диаграмма, снимок экрана, План

Автоматически созданное описание

Код программы:

.data

A dw 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; массивы из 10 элементов

B db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

.code

; Инициализация на начальном этапе

start:

MOV EBX, offset A

MOV ESP, offset B

MOV DH, 11010000b ; Cтарший U\_0 = D0h

MOV ECX, 10 ; N=10

MAIN:

MOV AH, 178 ; 178 - 10110010 00000000, Rc=15. Вместо нее - IN AH, 089h

TEST AH, 10000000b ; Проверка 15 разряда

JZ MAIN

; Проверка четности, этап модификации

TEST CX, 00000001b

JZ CHET\_FUNK

MOV DL, 01000100b ; S\_R = 2,6

JMP OUT\_DELAY

CHET\_FUNK:MOV DL, 00000001b ; R\_S = 0

; Сохранение значений в регистр BP,AX и вызов DELAY. Этап вывода + DELAY

OUT\_DELAY:

;OUT 08Ah, DX

MOV BP, DX

MOV AX, CX

JMP DELAY

; Этапы ввода состояний, сохранения в массивы и приращения

MAIN\_MASS:

MOV CX, AX

MOV DX, BP

MOV AX, 1 ; Вместо IN AX, 089h

; Работа с массивами

MOV [EBX], DX

MOV [ESP], AL

ADD EBX, 2

INC ESP

ADD DH, 04h ; приращение

LOOP MAIN

JMP EXIT

DELAY:

MOV DX, 510

L1: MOV CX,42302

MOV DX, DX

NOP

NOP

NOP

NOP

L2:

NOP

NOP

NOP

NOP

NOP

NOP

LOOP L2

DEC DX

JNZ L1

JMP MAIN\_MASS

EXIT:

end start

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 2 – Дамп памяти, отображающий результаты массивов

Результаты массива А: 01 D0 44 D4 01 D8 44 DC 01 E0 44 E4 01 E8 44 EC 01 F0 44 F4

Результаты массива B: 01 01 01 01 01 01 01 01 01 01

**Вывод:** изучили принципы организации ввода информации извне в УВМ и вывода информации из УВМ вовне, организации временной задержки при обработке данных на языке ассемблера, а также приобрели практические навыки программирования указанных операций.