**ИТМО Кафедра Вычислительной техники**

Отчет по лабораторной работе №7

«Синтез команд БЭВМ»

**Выполнил: студент группы P3117**

**Плюхин Дмитрий**

**Преподаватель: Перминов И. В.**

1. **год**
2. **Задание**

Синтезировать цикл исполнения для выданной преподавателем команды. Разработать тестовые программы, которые проверяют синтезированную команду. Загрузить в микропрограммную память БЭВМ цикл исполнения синтезированной команды, загрузить в основную память БЭВМ тестовые программы. Проверить и отладить разработанные тестовые программы и микропрограммы.

Команда для синтеза: MAND M - побитовое И аккумулятора с ячейкой памяти с записью результата в ячейку памяти и установкой C/N/Z

1. **Текст синтезированной микропрограммы**

|  |  |  |
| --- | --- | --- |
| **Адрес МП** | **Микрокоманды** | **Действие : Комментарий** |
| B0 | 1120 | A & РД -> БР : Записываем в БР результат побитового  : И аккумулятора с регистром данных  : (где в данный момент находится  : содержимое выбранной ячейки памяти) |
| B1 | 4072 | БР -> C,N,Z,РД : Устанавливаем C/N/Z и пересылаем  : результат конъюнкции в РД |
| B2 | 0002 | РД -> ОП (РА) : Пересылаем результат конъюнкции из  : РД в ячейку памяти |
| B3 | 8390 | GOTO ПРЕ(90) : Переходим на цикл прерывания |

1. **Текст тестовых программ**

ORG 210

NUMT: WORD 0005 ;Количество тестов

ORG 008

RZS: WORD 0010 ;Адрес начала массива промежуточных результатов

ORG 015

RZULT: WORD 0000 ;Конечный результат

TWORD1: WORD FACE ;Первое тестовое слово

TWORD2: WORD FEED ;Второе тестовое слово

BUFF: WORD ? ;Буфер для временного хранения данных

TW2ADR: WORD 0017 ;Адрес второго тестового слова

COUNTR: WORD ? ;Счетчик

ORG 020 ;Основная программа

BEGIN: JSR CHK\_C0 ;Проверка сохранения нулевого состояния бита С

JSR CHK\_C1 ;Проверка сброса бита С

JSR VALNUL ;Проверка логического умножения на 16 нулей

JSR VALFUL ;Проверка логического умножения на 16 единиц

JSR VALCUR

JSR ALLOK ;Выдача конечного результата тестирования

HLT

SVTW2: WORD ? ;Сохранение второго тестового слова

CLA

ADD TWORD2

MOV BUFF

BR (SVTW2)

RETW2: WORD ? ;Восстановление второго тестового слова

CLA

ADD BUFF

MOV TWORD2

BR (RETW2)

;Проверка сохранения нулевого состояния бита С

CHK\_C0: WORD ? ;Адрес возврата

JSR SVTW2

CLA

CMA

AND TWORD1

WORD 7017 ;MAND TWORD2

BCS WRT1

CLA

WRT1: JSR WRTRZS

JSR RETW2

BR (CHK\_C0)

;Проверка сброса бита С

CHK\_C1: WORD ? ;Адрес возврата

JSR SVTW2

CLA

CMA

AND TWORD1

ADD TWORD2 ;Подбираем такие тестовые слова, чтобы при их

сумме происходил перенос

WORD 7017 ;MAND TWORD2

BCS WRT2 ;Если бит С оказался сброшен после команды MAND

то такой результат корректен

CLA

WRT2: JSR WRTRZS

JSR RETW2

BR (CHK\_C1)

;Проверка логического умножения на 16 нулей

VALNUL: WORD ? ;Адрес возврата

JSR SVTW2

CLA

WORD 7017 ;MAND TWORD2

CLA

ADD TWORD2 ;Должны были получить 0 в ячейке TWORD2

JSR WRTRZS

JSR RETW2

BR (VALNUL)

;Проверка логического умножения на 16 единиц

VALFUL: WORD ? ;Адрес возврата

JSR SVTW2

CLA

CMA

WORD 7017 ;MAND TWORD2

CLA

ADD TWORD2 ;Значение TWORD2 не должно было измениться

SUB BUFF

JSR WRTRZS

JSR RETW2

BR (VALFUL)

;Проверка логического умножения ячейки с

использованием косвенной адресации

VALCUR: WORD ? ;Адрес возврата

CLA

ADD TWORD1

AND (TW2ADR)

MOV BUFF ;Запишем результат побитового И

CLA

ADD TWORD1

WORD 7819 ;MAND (TW2ADR)

CLA

ADD (TW2ADR)

SUB BUFF ;Побитовое И и команда MAND должны были

одинаково преобразовать аргументы

JSR WRTRZS

BR (VALCUR)

WRTRZS: WORD ? ;Запись успешности прохождения теста

BEQ NOERR ;Если в аккумуляторе 0, то ошибки не возникло

CLA

MOV (RZS)

BR (WRTRZS)

NOERR: CLA

INC

MOV (RZS)

BR (WRTRZS)

ALLOK: WORD ? ;Запись успешности прохождения теста

CLA

SUB NUMT

MOV COUNTR

ADD RZS

MOV RZS

CLA

LOOP: ADD (RZS)

ISZ COUNTR

BR LOOP

SUB NUMT

JSR WRTRZS

BR (ALLOK)

1. **Таблица трассировки цикла исполнения команды (для 5-го теста)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Сч. МК до выборки МК** | **Содержимое памяти и регистров процессора после выборки и исполнения МК** | | | | | | | | | | | |
| **яч. 0000** | **РМК** | **СК** | **РА** | **РК** | **РД** | **А** | **С** | **БР** | **N** | **Z** | **СчМК** |
| 2A | 0000 | 83B0 | 065 | 017 | 7819 | FEED | FACE | 0 | 00004 | 1 | 0 | B0 |
| **Цикл исполнения** | | | | | | | | | | | | |
| B0 | 0000 | 1120 | 065 | 017 | 7819 | FEED | FACE | 0 | 0FACC | 1 | 0 | B1 |
| B1 | 0000 | 4072 | 065 | 017 | 7819 | FACC | FACE | 0 | 0FACC | 1 | 0 | B2 |
| B2 | 0000 | 0002 | 065 | 017 | 7819 | FACC | FACE | 0 | 00000 | 1 | 0 | B3 |
| B3 | 0000 | 8390 | 065 | 017 | 7819 | FACC | FACE | 0 | 00004 | 1 | 0 | 90 |

**Вывод**

Так, в результате лабораторной работы на практике были освоены принципы микропрограммирования и разработки адресных команд. Я узнал, как в базовой ЭВМ осуществляется разработка собственных команд, их загрузка в микропрограммную память и использование. Изученный материал можно использовать как основу для изучения принципов работы микропроцессорных наборов.