Отчет

**по лабораторной работе №7**

**«Синтез команд БЭВМ»**

по дисциплине «Основы профессиональной деятельности» вариант 1067

Выполнил: Караганов П.Э., группа P3110

Преподаватель: Блохина Е.Н.

Оглавление

[Текст задания 3](#_Toc199541711)

[Исходный код синтезируемой команды 3](#_Toc199541712)

[Текст тестовой программы на языке Ассемблера БЭВМ 3](#_Toc199541713)

[Таблица трассировки микропрограммы 3](#_Toc199541714)

[Методика проверки программы 3](#_Toc199541715)

[Таблицы результатов работы комплекса 3](#_Toc199541716)

[Вывод 3](#_Toc199541717)

# Текст задания

Синтезировать цикл исполнения для выданных преподавателем команд. Разработать тестовые программы, которые проверяют каждую из синтезированных команд. Загрузить в микропрограммную память БЭВМ циклы исполнения синтезированных команд, загрузить в основную память БЭВМ тестовые программы. Проверить и отладить разработанные тестовые программы и микропрограммы.

1. SHR X - сдвиг аккумулятора вправо на X разрядов, 15 разряд заполняется значением 0, количество сдвигов содержится в коде команды. Признаки N/Z/V/C не устанавливать
2. Код операции - 0F2X
3. Тестовая программа должна начинаться с адреса 00CF16

# Исходный код синтезируемой команды

E0 0000200000 RESERVED 0 → C

E1 81ED801002 if CR(7) = 1 then GOTO ED

E2 81ED401002 if CR(6) = 1 then GOTO ED

E3 80ED201002 if CR(5) = 0 then GOTO ED

E4 81ED101002 if CR(4) = 1 then GOTO ED

E5 81E9081002 if CR(3) = 1 then GOTO E9

E6 81E9041002 if CR(2) = 1 then GOTO E9

E7 81E9021002 if CR(1) = 1 then GOTO E9

E8 80ED011002 if CR(0) = 0 then GOTO ED

E9 0002009202 ~0 + CR → CR

EA 0010180010 ROR(AC) → AC

EB 0000200000 0 → C

EC 80E4109040 GOTO E4

ED 80C4101040 GOTO INT @ C4

# Текст тестовой программы на языке Ассемблера БЭВМ

        ORG 0x00CF

pnt:        WORD 0x0001

X1:         WORD 0xFFFE

chec1:      WORD 0x7FFF

X2:         WORD 0x70FF

chec2:      WORD 0x0007

X3:         WORD 0x0010

chec3:      WORD 0x0010

START:  CLA

        CALL TEST1

        CALL TEST2

        CALL TEST3

        HLT

TEST1:

        LD X1

        NOP

        WORD 0x0F21 ; сдвиг на 1 бит вправо

        NOP

        CMP chec1

        BEQ equal1

        RET

equal1: LD #0x01

        ST (pnt)+

        RET

TEST2:

        LD X2

        NOP

        WORD 0x0F2C ; сдвиг на 12 битов вправо

        NOP

        CMP chec2

        BEQ equal2

        RET

equal2: LD #0x01

        ST (pnt)+

        RET

TEST3:

        LD X3

        NOP

        WORD 0x0F3F ; неверный код программы, не должно работать

        NOP

        CMP chec3

        BEQ equal3

        RET

equal3: LD #0x01

        ST (pnt)+

        RET

# Таблица трассировки микропрограммы

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **МР до выборки МК** | **Содержимое памяти и регистров процессора после выборки микрокоманды** | | | | | | | | | |
| **MR** | **IP** | **AR** | **DR** | **DR** | **SP** | **BR** | **AC** | **NZVC** | **СчМК** |
| 01 | 00A0009004 | 013 | 0100 | 013 | 0100 | 000 | 0013 | 0002 | 0000 | 02 |
| 02 | 0104009420 | 014 | 0100 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | 03 |
| 03 | 0002009001 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | 04 |
| 04 | 8109804002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | 05 |
| 05 | 810C404002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | 06 |
| 06 | 810C204002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | 07 |
| 07 | 8078104002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | 78 |
| 78 | 81A4084002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | A4 |
| A4 | 81B5044002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | B5 |
| B5 | 81BB024002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | BB |
| BB | 81E0014002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E0 |
| E0 | 0000200000 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E1 |
| E1 | 81ED801002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E2 |
| E2 | 81ED401002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E3 |
| E3 | 80ED201002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E4 |
| E4 | 81ED101002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E5 |
| E5 | 81E9081002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E6 |
| E6 | 81E9041002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E7 |
| E7 | 81E9021002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E8 |
| E8 | 80ED011002 | 014 | 0F21 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | E9 |
| E9 | 0002009202 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0002 | 0000 | EA |
| EA | 0010180010 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | EB |
| EB | 0000200000 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | EC |
| EC | 80E5109040 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | E5 |
| E5 | 81E9081002 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | E6 |
| E6 | 81E9041002 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | E7 |
| E7 | 81E9021002 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | E8 |
| E8 | 80ED011002 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | ED |
| ED | 80C4101040 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | C4 |
| C4 | 80DE801040 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | C5 |
| C5 | 8001401040 | 014 | 0F20 | 013 | 0F21 | 000 | 0013 | 0001 | 0000 | 01 |

# Методика проверки программы

1. Загрузить комплекс разработанных микропрограмм в микропрограммную память БЭВМ
2. Загрузить тестовую программу в память базовой ЭВМ.
3. Запустить основную программу с адреса 0x00CF в режиме работа.
4. Дождаться останова.
5. Проверить значение ячеек памяти с адресами 0x0001,0x0002 и 0x0003 если все значения равны 0x0001 – все тесты выполнены успешно.

Комментарии к методике

* Для проверки используется три теста, где в первом проверяется сдвиг 0xFFFE на 1 бит вправо, во втором – сдвиг 0x70FF на 12 бит и в третьем – проверка на некорректную программу
* Данные значения показывают правильную работу программы, а также работу при неправильном задании нашей команды.
* Результат каждого теста записывается в соответствующую ячейку в памяти (результат первого теста лежит в 0x0001, второго в 0x0002 и так далее), значение ячеек равное 0x0001 означает успешное выполнение. 0x0000 – ошибку при выполнении.

# Таблицы результатов работы комплекса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **AC** | **Код команды** | **Теоретический результат** | **Фактический результат** |
| Тест №1 | 0xFFFE | 0x0F21 | 0x7FFF | 0x7FFF |
| Тест №2 | 0x70FF | 0x0F2C | 0x0007 | 0x0007 |
| Тест №3 | 0x0010 | 0x0F3F | 0x0010 | 0x0010 |

# Вывод

В ходе лабораторной работы по синтезу команд БЭВМ изучены принципы формирования микрокоманд и их выполнение. Практически освоены операции с регистрами (CR, AC, DR), синтез арифметических и логических команд, условные переходы. Выявлены типичные ошибки: некорректные маски переходов, неправильная инициализация регистров. Полученные навыки микропрограммирования и отладки полезны для понимания архитектуры процессоров. Работа показала важность точного соблюдения форматов команд. ***Цикл прерывания:***

1. if PS(W) = 0, то останавливаемся
2. if PS(IRQ) = 0, то просто идём дальше в цикл выборки следующей команды
3. SP – 1 -> SP, AR; IP -> DR; DR -> Mem(AR) ⇔ IP -> -(SP) (тут мы просто сохранили IP для того чтобы было куда возвращаться)
4. SP – 1 -> SP, AR; PS -> DR; DR -> Mem(AR) ⇔ PS -> -(SP) (тут мы просто сохранили PS для того чтобы при возвращении мы не сломались)
5. LTOL(CR) -> BR (сохранили младшие 8 бит в BR)
6. SHL(BR) -> BR, AR (умножаем номер вектора прерывания на 2)
7. Mem(AR) -> DR
8. DR -> IP
9. LTOL(BR+1) -> AR
10. MEM(AR) -> DR
11. DR -> PS
12. Идём в цикл выборки команды по-новому IP (мы в обработке прерывания)