

Государственное образовательное учреждение
высшего профессионального образования
«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ»
Кафедра «Информационные вычислительные системы»

**Курсовой проект по дисциплине
«Организация ЭВМ и систем»
на тему:
«Микропрограммная интерпритация команд процессора»
Пояснительная записка**

Скачано с сайта <http://ivc.clan.su>

Разработал:
студент гр. ПВТ-711
Круглов В.А.

Руководитель:
профессор кафедры ИВС
Ефимова Р.С.

Санкт-Петербург
2010

PDF-document was made by [Vasena](#)

Аннотация

В курсовом проекте разрабатываются микрокоды для трёх команд 16-разрядного процессора, построенного на основе БИС К1804. Для этого сначала разрабатываются микропрограммы заданных команд в виде граф-схем алгоритмов. После чего выполняется кодирование микропрограмм в виде таблиц их кодов.

Для получения навыков микропрограммирования и изучения элементной базы процессора с микропрограммным управлением в ходе работы над проектом выполняется ряд лабораторных работ на микротренажере МТ 1804, результат которых оформляется в виде приложения к данной пояснительной записке.

Содержание

Аннотация	2
Содержание	3
Введение	4
1. Микропрограммная реализация вычислительных процедур на тренажере МТ1804	6
1.1. Лабораторная работа №2. Пересылки. Сдвиги. Арифметические и логические операции. Безусловные переходы	6
1.2. Лабораторная работа №3. Условные переходы. Микроподпрограммы. Организация циклов. Работа со стеком.	6
2. Разработка граф-схемы алгоритмов выполнения заданных команд	7
2.1. Граф-схема алгоритма команды FETCH	7
2.2. Граф-схема алгоритма команды OR.....	7
2.3. Граф-схема алгоритма команды ADD	8
2.4. Граф-схема алгоритма команды CALL.....	9
3. Разработка микропрограмм заданных команд в виде таблиц кодов	10
Таблицы микропрограмм	11
Заключение	13
Список использованных источников	14
Приложение	15

Введение

В любой вычислительной системе, в том числе и в процессоре, можно выделить две основные части: *исполнительную* – операционный автомат (ОА) и *управляющую* – управляющий автомат (УА). ОА предназначен для выполнения арифметических, логических и сдвиговых микроопераций. Основная функция УА – это выработка последовательности управляющих сигналов, необходимых для правильного функционирования исполнительской части вычислительной системы. Один и тот же УА может быть реализован в виде устройства с жесткой логикой или с программируемой логикой.

УА с жесткой логикой – это физическая реализация – некая электрическая схема, функционирование которой обеспечивает исполнение алгоритма работы системы.

В этом случае для каждого конкретного алгоритма необходимо конструировать уникальную схему. И соответственно изменение алгоритма функционирования системы невозможно.

УА с программируемой логикой (или блок микропрограммного управления – БМУ). В этом случае аппаратура вычислительной машины создается один раз, а для реализации конкретного алгоритма создается соответствующая система управляющих слов, которая может быть в дальнейшем изменена, что обеспечивает несравнимо большую гибкость по сравнению с УА с жесткой логикой.

Несмотря на то, что построение процессоров на основе жесткой логики, дает большую скорость выполнения операций, на практике гораздо чаще используют микропрограммную реализацию управления, как более простую, дешевую и гибкую.

Основные достоинства процессоров с микропрограммным управлением:

- возможность создания системы команд, наилучшим образом реализующей функции проектируемого устройства;
- возможность создания эмуляторов системы команд любых ЭВМ;
- возможность расширения списков команд;
- возможность записывать типовые процедуры, циклы и т.п.

К основным недостаткам можно отнести относительно низкое быстродействие.

В данной курсовой работе проектируется УА с программируемой логикой.

Рассматриваемый процессор является 16-разрядным и работает с 16-разрядными адресами основной памяти. Процессор ориентирован на выполнение множества операций, инициируемых командами, которые хранятся в основной памяти. Выполнение каждой команды предполагает запуск и выполнение соответствующей микропрограммы, хранящейся в микропрограммной памяти. Процессор состоит из двух основных блоков: блока обработки данных (БОД) и блока микропрограммного управления (БМУ).

Состав БОД:

- операционный блок (ОБ), который строится на основе 4-х процессорных секций K1804BC1 и включает схему ускоренного переноса (СУП) и схему управления сдвигами (СУС);
- регистр входных данных (РвхД);
- регистр выходных данных (РвыхД);
- регистр адреса (РА);
- дешифратор констант (DC const);
- мультиплексор сигнала состояния (МСС);
- мультиплексор сигнала входного переноса С0 (МС0).

Состав БМУ:

- регистр команд (РК);
- преобразователь начального адреса (ПНА);
- схема формирования адресов микрокоманд K1804BY4 (СФАМК);
- микропрограммная память (МПП);
- мультиплексор сигналов условий (МСУ);
- инвертор сигнала условий (ИСУ);
- регистр микрокоманд (РМК).

Цель работы: изучение элементной базы процессора с микропрограммным управлением (комплекта микропрограммируемых БИС К1804) и разработка микропрограмм в виде граф-схем алгоритмов (ГСА) и таблиц кодов микрокоманд для рассматриваемого процессора.

1. Микропрограммная реализация вычислительных процедур на тренажере МТ1804

В данном разделе приводятся лабораторные работы № 2 и № 3 с учётом номера варианта выбранного перед их выполнением. Используемый вариант – шестой (0110).

1.1. Лабораторная работа №2

Пересылки. Сдвиги. Арифметические и логические операции. Безусловные переходы.

Задание:

- 1) Занести код номера бригады в любой регистр CPU. Удвоить код, используя: операцию сложения; операцию сдвига. Возможное переполнение и выходной перенос игнорируется.
- 2) Занести в регистр CPU код номера бригады, в регистр Q – число, на 1 большее. Поменять содержимое регистров местами, используя: операции пересылок с использованием свободного регистра CPU; операцию сдвига двойной длины.
- 3) Занести код номера бригады в любой регистр CPU. Выделить, используя операцию конъюнкции, каждый из четырех разрядов кода и разместить выделенные разряды в четырех свободных регистрах.
- 4) Занести код номера бригады в любой регистр. Получить обратный код в свободном регистре.
- 5) Обнулить регистр Q, используя: операцию занесения нуля; операцию «исключающее ИЛИ»; операцию конъюнкции; операцию вычитания. Перед каждым вариантом обнуления предусмотреть занесение в регистр ненулевого кода.
- 6) Занести код номера бригады в любой регистр. Организовать: декрементирование содержимого регистра (операция вычитания); инкрементировать его же (операция сложения) до обнуления регистра (использовать операцию безусловного перехода на операцию вычитания - сложения).

Таблицы кодов микрокоманд данных микропрограмм приводятся в приложении в таблице 2.

1.2. Лабораторная работа №3

Условные переходы. Микроподпрограммы. Организация циклов. Работа со стеком.

Задание:

Подготовить микропрограмму, реализующую алгоритм «Д».

В качестве константы цикла взять результат выражения

$E[N_{60}/3]+1$, где $E[...]$ – ближайшее большее целое.

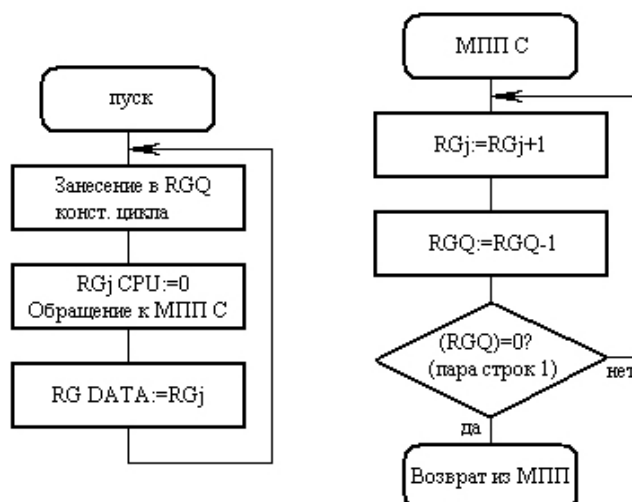


Рис. 1 Алгоритм «Д» лабораторной работы №3

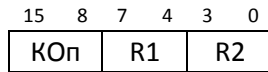
Таблицы кодов микрокоманд данных микропрограмм приводятся в приложении в таблице 3(Д).

2. Разработка граф-схемы алгоритмов выполнения заданных команд

В данном разделе приводятся граф-схемы алгоритмов (ГСА) заданных команд и команда выборки первого слова команды из памяти.

2.1. Граф-схема алгоритма команды FETCH

Формат команды



КОп – код операции

R1 – номер регистра первого операнда и/или приемник результата

R2 – номер регистра второго операнда

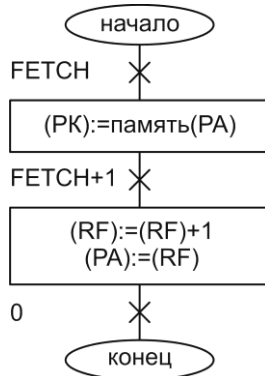


Рис. 2 Граф-схема выборки первого слова команды из основной памяти

FETCH – обеспечивает чтение первого слова команды по адресу, который зафиксирован в РА (предполагается, что РА к моменту передачи управления микропрограмме содержит адрес команды). Запись в РК осуществляется по положительному фронту тактового сигнала в конце микроцикла, управляемого данной микрокомандой.

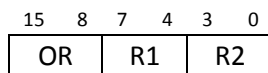
FETCH+1 – инкрементирует счетчик команд, организованный в одном из РОН операционного блока и обеспечивает запись инкрементированного значения счетчика в РА. Фиксация нового содержимого в РА осуществляется по положительному фронту тактового сигнала, т.е. в конце текущего микроцикла.

При выполнении второй микрокоманды в СФАМК на адресные входы МПП выдается адрес первой микрокоманды микропрограммы, соответствующей КОп в РК.

Микропрограмма в виде таблицы кодов приведена в таблице 1.

2.2. Граф-схема алгоритма команды OR

Формат команды



OR – команда ИЛИ

R1 – номер регистра первого операнда и приемник результата

R2 – номер регистра второго операнда

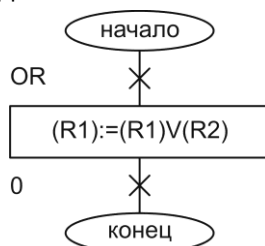


Рис. 3 Граф-схема микропрограммы ИЛИ

Мнемоника команды: OR R1 R2

Описание команды: (R2)OR(R1)→(R1)

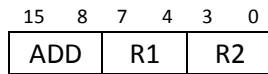
Выполняемое действие: выполнение логической операции ИЛИ над содержимым регистра R1 и содержимым регистра R2 и сохранение результата в регистре R1. Далее

следует переход по нулевому адресу (0000) – код инструкции, реализуемой СФАМК, по которому расположена микропрограмма выборки первого слова команды.

Микропрограмма в виде таблицы кодов микрокоманд приведена в таблице 1.

2.3. Граф-схема алгоритма команды ADD

Формат команды



ADD – команда сложения

R1 – номер регистра первого операнда и приемник результата

R2 – номер регистра второго операнда

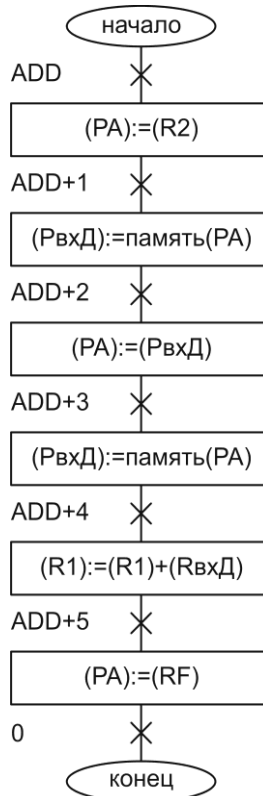


Рис. 4 Граф-схема микропрограммы сложения

Мнемоника команды: ADD R1 ZR2, где ZR2 – регистр косвенного указателя

Описание команды: $\{[(R2)]\}+(R1) \rightarrow (R1)$

Выполняемое действие: сложение содержимого регистра R1 и содержимого регистра R2 и сохранение результата в регистре R1. Далее следует переход по нулевому адресу (0000) – код инструкции, реализуемой СФАМК, по которому расположена микропрограмма выборки первого слова команды.

Описание микрокоманд микропрограммы:

- ADD – занесение в PA адреса указателя на второй операнд
- ADD +1 – получение на регистре входных данных указателя – адреса второго операнда
- ADD +2 – занесение в PA адреса второго операнда
- ADD +3 – получение на регистре входных данных значения второго операнда
- ADD +4 – выполняется логическая микрооперация сложения R1 и R2. Результат записывается на регистр по адресу R1
- ADD +5 – осуществление перехода к адресу следующей микрокоманды

Микропрограмма в виде таблицы кодов микрокоманд приведена в таблице 1.

2.4. Граф-схема алгоритма команды CALL

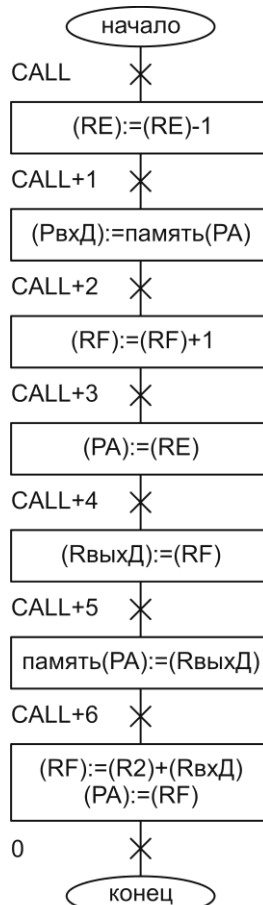
Формат команды



ADD – команда сложения

R2 – номер регистра второго операнда

ADDR – адрес перехода



Граф-схема микропрограммы сложения

Мнемоника команды: CALL IR2 ADDR

Описание команды: $(RE)-1 \rightarrow (RE)$; $(RF) \rightarrow [(RE)]$; $(R2)+ADDR \rightarrow (RF)$

Выполняемое действие: Сохранение в стеке счетчика команд. Занесение в счетчик команд адрес перехода.

- CALL – изменение указателя стека
- CALL +1 – получение адреса перехода на регистре входных данных
- CALL +2 – инкрементирование счетчика команд
- CALL +3 – занесение в РА адрес стековой ячейки памяти
- CALL +4 – сохранение на регистре выходных данных счетчика команд
- CALL +4 – сохранение в стеке счетчика команд
- CALL +5 – установка счетчика команд на новый адрес

Микропрограмма в виде таблицы кодов микрокоманд приведена в таблице 1.

3. Разработка микропрограмм заданных команд в виде таблиц кодов

В данном разделе содержатся микропрограммы в виде таблиц кодов команд, представленных граф–схем.

Формат микрокоманды

Адресная часть МК:

- Addr – поле для указания адреса перехода в микропрограмме (12 бит);
 I – поле для указания кода операции, реализуемой СФАМК К1804ВУ4 (4 бита);
 X – поле управления выбором условия (2 бита);
 Inv – поле управления полярностью сигнала условия (1 бит)
 0 – сигнал инвертируется;
 1 – сигнал не инвертируется;
 CCE – поле разрешения проверки условия (1 бит);
 0 – проверка разрешена;
 1 – проверка запрещена;

Операционная часть МК:

- I(8:0) – микрокод управления процессорными секциями К1804ВС1 (9 бит);
 A/V – поле выбора адреса А либо В регистра ОБ (4 бита);
 A/D – поле может использоваться для указания адреса А либо для задания кода константы (4 бита);
 SA – поле управления выбором источника адреса А (2 бита);
 00 - РК(3:0)(R2);
 01 - РМК(A/D);
 10 - РК (7:4)(R1);
 11 - РМК(A/V);
 SB – поле выбора источника адреса В (1 бит);
 0 - РК (7:4)(R1);
 1 - РМК(A/V);
 CO – значение входного сигнала переноса для младшей процессорной секции (1 бит)
 SCO – поле управления выбором сигнала CO (1 бит)
 0 – разряд МК;
 1 – из РС;
 SS – поле управления выбором состояния (1 бит)
 0 – признаки в РС сохраняются;
 1 – признаки устанавливаются;
 DC-OE – поле управления работой дешифратора констант (1 бит)
 0 – разрешение выдачи константы
 1 – запрет выдачи
 SHFT – поле управления сдвигами (не менее 2 бита)

Сигналы управления регистрами:

- | | |
|---|--|
| $\left. \begin{array}{l} PK - \overline{E} \\ PB_{\text{ВыхD}} - \overline{E} \\ PB_{\text{xD}} - \overline{E} \\ PA - \overline{E} \end{array} \right\}$ | <p>0 – разрешение загрузки в регистр по положительному фронту тактового сигнала
 1 – запрет загрузки</p> |
| $\left. \begin{array}{l} PB_{\text{ВыхD}} - \overline{OE} \\ PB_{\text{xD}} - \overline{OE} \\ PA - \overline{OE} \end{array} \right\}$ | <p>0 – разрешение выдачи из соответствующего регистра
 1 – запрет выдачи</p> |

Выходные сигналы процессора:

- MRD – чтение из основной памяти;
 MWR – запись в основную память.

Таблицы микропрограмм

Таблица 1

Поля МК	Addr	I	X	Inv	CCE	I(8:6)	I(5:3)	I(2:0)	A/B	A/D	SA	SB	CO
Номера разрядов	54:43	42:39	38:37	36	35	34:32	31:29	28:26	25:22	21:18	17:16	15	14
FETCH	-	1110	-	-	-	-	-	-	-	-	-	-	-
FETCH+1	-	0000	-	-	-	011	000	011	1111	-	-	1	1
OR	-	0000	-	-	-	011	011	001	-	-	00	0	-
ADD	-	1110	-	-	-	001	011	100	-	-	00	-	-
ADD+1	-	1110	-	-	-	-	-	-	-	-	-	-	-
ADD+2	-	1110	-	-	-	001	011	111	-	-	-	-	-
ADD+3	-	1110	-	-	-	-	-	-	-	-	-	-	-
ADD+4	-	1110	-	-	-	011	000	101	-	-	10	0	0
ADD+5	-	0000	-	-	-	001	011	100	-	1111	01	-	-
CALL	-	1110	-	-	-	011	001	011	1110	-	-	1	-
CALL+1	-	1110	-	-	-	-	-	-	-	-	-	-	-
CALL+2	-	1110	-	-	-	011	000	011	1111	-	-	1	1
CALL+3	-	1110	-	-	-	001	011	100	-	1110	01	-	-
CALL+4	-	1110	-	-	-	001	011	100	-	1111	01	-	-
CALL+5	-	1110	-	-	-	-	-	-	-	-	-	-	-
CALL+6	-	0000	-	-	-	011	000	101	1111	-	00	1	0

SC0	SS	DC-OE	SHFT	PK-E	PiD-E	PoD-E	PA-E	PiD-OE	PoD-OE	PA-OE	MRD	MWR	Комментарий
13	12	11	10:9	8	7	6	5	4	3	2	1	0	
-	0	1	-	0	1	1	1	1	1	0	1	0	(RK):=m(PA)
0	0	1	-	1	1	1	0	1	1	1	0	0	(RF):=(RF)+1; (PA):=(RF)
-	1	1	-	1	1	1	1	1	1	1	0	0	(R1):=(R1)+(R2)
-	0	1	-	1	1	1	0	1	1	1	0	0	(PA):=(R2)
-	0	1	-	1	0	1	1	1	1	0	1	0	(PiD):=m(PA)
-	0	1	-	1	1	1	0	0	1	1	0	0	(PA):=(PiD)
-	0	1	-	1	0	1	1	1	1	0	1	0	(PiD):=m(PA)
0	1	1	-	1	1	1	1	0	1	1	0	0	(R1):=(R1)+(PiD)
-	0	1	-	1	1	1	0	1	1	1	0	0	(PA):=(RF)
-	0	1	-	1	1	1	1	1	1	1	0	0	(RE):=(RE)-1
-	0	1	-	1	0	1	1	1	1	0	1	0	(PiD):=m(PA)
0	0	1	-	1	1	1	1	1	1	1	0	0	(RF):=(RF)+1
-	0	1	-	1	1	1	0	1	1	1	0	0	(PA):=(RE)
-	0	1	-	1	1	0	1	1	1	1	0	0	(PoD):=(RF)
-	0	1	-	1	1	1	1	1	0	0	0	1	m(PA):=(PoD)
0	0	1	-	1	1	1	0	0	1	1	0	0	(RF):=(R2)+(PiD); (PA):=(RF)

Заключение

В результате выполнения курсового проекта были изучены структура и элементная база процессора, разработаны микропрограммы заданных команд в виде граф-схем алгоритмов и выполнено кодирование микропрограмм заданных команд в виде таблиц кодов микропрограмм; приобретены навыки микропрограммирования на электронном микротренажере МТ1804.

Список использованных источников

- 1) Р.С. Ефимова, Г. И Кожомбердиева. Учебное пособие «Микропрограммная интерпретация команд процессора» СПб 2006 г.
- 2) Р.С. Ефимова, Конспект лекций по курсу «Организация ЭВМ», 2009г.

Приложение

Таблица 2

Тетрады МК	7	111			6	110			5	101			4	100			3	011		
Разряды МК	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12
Адрес МК	R3	R2	R1	R0	P3	P2	P1	P0	MS2	I8	I7	I6	MS1	I2	I1	I0	C0	I5	I4	I3

1.1 Удвоение кода номера бригады сложением регистра самим с собой

0000	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0001	0	0	0	0	0	0	0	1	-	0	1	1	-	0	1	1	0	0	0	0

1.2 Удвоение кода номера бригады сдвигом влево на 1 бит

0000	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0001	-	-	-	-	0	0	1	0	-	1	1	1	-	0	1	1	-	0	1	1
0010	0	0	0	0	0	0	0	1	-	0	1	1	-	0	1	1	-	0	1	1

2.1 Занесение в регистр Q числа, на 1 большего. Обмен содержимого регистров — пересылка

0000	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0001	-	-	-	-	0	0	1	0	-	0	0	0	-	0	0	0	1	0	0	0
0010	-	-	-	-	0	0	1	0	-	0	1	1	-	1	0	0	-	0	1	1
0011	-	-	-	-	0	0	1	0	-	0	1	1	-	0	1	0	-	0	1	1
0100	0	0	0	0	0	0	0	1	-	0	0	0	-	0	1	1	-	0	1	1

2.2 Занесение в регистр Q числа, на 1 большего. Обмен содержимого регистров — сдвиг

0000	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0001	-	-	-	-	0	0	1	0	-	0	0	0	-	0	0	0	1	0	0	0
0010	-	-	-	-	0	0	1	0	1	1	0	0	0	0	1	1	-	0	1	1
0011	-	-	-	-	0	0	1	0	1	1	0	0	0	0	1	1	-	0	1	1
0100	-	-	-	-	0	0	1	0	1	1	0	0	0	0	1	1	-	0	1	1
0101	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	-	0	1	1

3 Выделение разрядов из кода номера бригады и размещения их в регистрах

0000	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0001	-	-	-	-	0	0	1	0	-	0	1	1	-	1	0	1	-	1	0	0
0010	-	-	-	-	0	0	1	0	-	0	1	1	-	1	0	1	-	1	0	0
0011	-	-	-	-	0	0	1	0	-	0	1	1	-	1	0	1	-	1	0	0
0100	0	0	0	0	0	0	0	1	-	0	1	1	-	1	0	1	-	1	0	0

4 Получение обратного кода в свободном регистре

0000	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0001	0	0	0	0	1	1	0	0	-	0	0	1	-	1	0	1	-	1	0	0
0010	0	0	0	0	0	0	0	1	-	0	1	1	-	1	0	1	-	1	1	0

5.1 Обнуление регистра Q. Занесение 0

0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	0	0	0	0	0	0	0	1	-	0	0	0	-	1	1	1	-	0	1	1

5.2 Обнуление регистра Q. Исключающее ИЛИ

0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	-	-	-	-	0	0	1	0	-	0	1	1	-	0	1	0	-	0	1	1
0010	0	0	0	0	0	0	0	1	-	0	0	0	-	0	0	0	-	1	1	0

5.3 Обнуление регистра Q. Конъюнкция

0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	0	0	0	0	0	0	0	1	-	0	0	0	-	0	1	0	-	1	0	0

5.4 Обнуление регистра Q. Вычитание

0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	-	-	-	-	0	0	1	0	-	0	1	1	-	0	1	0	-	0	1	1
0010	0	0	0	0	0	0	0	1	-	0	0	0	-	0	0	0	1	0	0	1

6.1 Декрементирование регистра до обнуления

0000	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0001	0	0	0	1	0	0	0	0	-	0	1	1	-	0	1	1	0	0	0	1
0010	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	1	-	0	1	1

6.2 Инкрементирование регистра до обнуления

0000	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0001	0	0	0	1	0	0	0	0	-	0	1	1	-	0	1	1	1	0	0	0
0010	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	1	-	0	1	1

2 010				1 001				0 000				Комментарии
11	10	9	8	7	6	5	4	3	2	1	0	
A3	A2	A1	A0	B3	B2	B1	B0	D3	D2	D1	D0	
1.1 Удвоение кода номера бригады сложением регистра самим с собой												
-	-	-	-	0	0	0	0	0	1	1	0	RG0:=6; Y:=6
0	0	0	0	0	0	0	0	-	-	-	-	RG0:=RG0+RG0; Y:=RG0
1.2 Удвоение кода номера бригады сдвигом влево на 1 бит												
-	-	-	-	0	0	0	0	0	1	1	0	RG0:=6; Y:=6
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0; RG0:=2RG0
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0
2.1 Занесение в регистр Q числа, на 1 большего. Обмен содержимого регистров — пересылка												
-	-	-	-	0	0	0	0	0	1	1	0	RG0:=6; Y:=6
-	-	-	-	0	0	0	0	-	-	-	-	RGQ:=RG0+1; Y:=RGQ
0	0	0	0	0	0	0	1	-	-	-	-	RG1:=RG0; Y:=RG1
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RGQ; Y:=RG0
-	-	-	-	0	0	0	1	-	-	-	-	RGQ:=RG1; Y:=RGQ
2.2 Занесение в регистр Q числа, на 1 большего. Обмен содержимого регистров — сдвиг												
-	-	-	-	0	0	0	0	0	1	1	0	RG0:=6; Y:=6
-	-	-	-	0	0	0	0	-	-	-	-	RGQ:=RG0+1; Y:=RGQ
-	-	-	-	0	0	0	0	-	-	-	-	Циклический сдвиг влево
-	-	-	-	0	0	0	0	-	-	-	-	Циклический сдвиг влево
-	-	-	-	0	0	0	0	-	-	-	-	Циклический сдвиг влево
-	-	-	-	0	0	0	0	-	-	-	-	Циклический сдвиг влево
3 Выделение разрядов из кода номера бригады и размещения их в регистрах												
-	-	-	-	1	1	1	1	0	1	1	0	RGF:=6; Y:=6
1	1	1	1	0	0	0	0	0	0	0	1	RG0:=RGF&1; Y:=RG0
1	1	1	1	0	0	0	1	0	0	1	0	RG1:=RGF&2; Y:=RG1
1	1	1	1	0	0	1	0	0	1	0	0	RG2:=RGF&4; Y:=RG2
1	1	1	1	0	0	1	1	1	0	0	0	RG3:=RGF&8; Y:=RG3
4 Получение обратного кода в свободном регистре												
-	-	-	-	0	0	0	0	0	1	1	0	RG0:=6; Y:=6
0	0	0	0	-	-	-	-	1	0	0	0	Y:=RG0&8; Переход 0000 если Y=0
0	0	0	0	0	0	0	0	0	1	1	1	RG0:=RG0(+); Y:=RG0
5.1 Обнуление регистра Q. Занесение 0												
-	-	-	-	-	-	-	-	0	1	1	0	RGQ:=6; Y:=6
-	-	-	-	-	-	-	-	0	0	0	0	RGQ:=0; Y:=0
5.2 Обнуление регистра Q. Исключающее ИЛИ												
-	-	-	-	-	-	-	-	0	1	1	0	RGQ:=6; Y:=6
-	-	-	-	1	1	1	1	-	-	-	-	RGF:=RGQ; Y:=RGF
1	1	1	1	-	-	-	-	-	-	-	-	RGQ:=RGQ(+)&RGF; Y:=RGQ
5.3 Обнуление регистра Q. Конъюнкция												
-	-	-	-	-	-	-	-	0	1	1	0	RGQ:=6; Y:=6
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ&0; Y:=RGQ
5.4 Обнуление регистра Q. Вычитание												
-	-	-	-	-	-	-	-	0	1	1	0	RGQ:=6; Y:=6
-	-	-	-	1	1	1	1	-	-	-	-	RGF:=RGQ; Y:=RGF
1	1	1	1	-	-	-	-	-	-	-	-	RGQ:=RGQ-RGF; Y:=RGQ
6.1 Декрементирование регистра до обнуления												
0	-	-	-	0	0	0	0	0	1	1	0	RG0:=6; Y:=6
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RG0-1; Y:=RG0; Переход 0001 если Y<>0
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0
6.2 Инкрементирование регистра до обнуления												
-	-	-	-	0	0	0	0	0	1	1	0	RG0:=6; Y:=6
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RG0+1; Y:=RG0; Переход 0001 если Y<>0
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0

Тетрады МК	7 111				6 110				5 101				4 100				3 011			
Разряды МК	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12
Адрес МК	R3	R2	R1	R0	P3	P2	P1	P0	MS2	I8	I7	I6	MS1	I2	I1	I0	C0	I5	I4	I3
А																				
0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	-	-	-	-	1	0	0	1	-	0	1	1	-	1	1	1	-	1	1	1
0010	-	-	-	-	0	0	1	0	-	0	1	1	-	0	1	1	1	0	0	0
0011	-	-	-	-	1	0	0	0	-	0	0	0	-	0	1	0	0	0	0	1
0100	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	1	-	0	1	1
Б																				
0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	-	-	-	-	1	0	0	1	-	0	1	1	-	1	1	1	-	1	1	1
0010	-	-	-	-	0	0	1	0	-	0	1	1	-	0	1	1	1	0	0	0
0011	-	-	-	-	1	0	0	0	-	0	0	0	-	0	1	0	1	0	0	0
0100	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	1	-	0	1	1
В																				
0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0010	-	-	-	-	0	0	1	0	-	0	1	1	-	0	1	1	1	0	0	0
0011	0	0	1	0	0	0	0	0	-	0	0	0	-	0	1	0	0	0	0	1
0100	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	1	-	0	1	1
Г																				
0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	-	-	-	-	0	0	1	0	-	0	1	1	-	1	1	1	-	0	1	1
0010	-	-	-	-	0	0	1	0	-	0	1	1	-	0	1	1	1	0	0	0
0011	0	1	0	1	1	1	1	1	-	0	0	0	-	0	1	0	1	0	0	0
0100	0	0	1	0	0	0	0	1	-	0	0	1	-	-	-	-	-	-	-	-
0101	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	1	-	0	1	1
Д																				
0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	0	0	1	1	0	1	0	1	-	0	1	1	-	1	1	1	-	0	1	1
0010	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	1	-	0	1	1
0011	-	-	-	-	0	0	1	0	-	0	1	1	-	0	1	1	1	0	0	0
0100	0	0	1	1	0	0	0	0	-	0	0	0	-	0	1	0	0	0	0	1
0101	0	0	0	1	0	1	1	0	-	0	0	1	-	-	-	-	-	-	-	-
Е																				
0000	-	-	-	-	0	0	1	0	-	0	0	0	-	1	1	1	-	0	1	1
0001	-	-	-	-	1	0	0	1	-	0	1	1	-	1	1	1	-	0	1	1
0010	0	1	0	0	0	0	0	0	-	0	0	0	-	0	1	0	0	0	0	1
0011	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	1	-	0	1	1
0100	-	-	-	-	0	1	1	1	-	0	1	1	-	0	1	1	1	0	0	0
Ж																				
0000	-	-	-	-	0	0	1	1	-	0	0	1	-	-	-	-	-	-	-	-
0001	0	0	1	1	0	1	0	1	-	0	0	0	-	1	1	1	-	0	1	1
0010	0	0	0	0	0	0	0	1	-	0	0	1	-	0	1	0	-	0	1	1
0011	0	1	1	0	0	1	0	1	-	0	0	0	-	0	1	0	1	0	0	0
0100	0	1	1	1	0	1	0	1	-	0	0	0	-	1	1	0	0	0	0	0
0101	-	-	-	-	0	1	1	0	-	0	0	1	-	-	-	-	-	-	-	-
0110	-	-	-	-	0	1	1	0	-	0	0	0	-	0	1	0	0	0	0	1
0111	0	1	1	0	0	1	0	1	-	0	0	0	-	1	1	0	0	0	0	0
1000	1	0	1	0	0	1	0	1	-	0	0	0	-	1	1	0	0	0	0	0
1001	-	-	-	-	0	1	1	0	-	0	0	1	-	-	-	-	-	-	-	-
1010	1	1	0	0	0	1	0	1	-	0	0	0	-	0	1	0	0	0	0	1
1011	-	-	-	-	0	1	1	0	-	0	0	1	-	-	-	-	-	-	-	-
1100	0	1	1	0	0	1	0	1	-	0	0	0	-	1	1	0	0	0	0	1
1101	-	-	-	-	0	1	1	0	-	1	1	0	-	-	-	-	-	-	-	-

2 010				1 001				0 000				Комментарии
11	10	9	8	7	6	5	4	3	2	1	0	
A3	A2	A1	A0	B3	B2	B1	B0	D3	D2	D1	D0	
А												
-	-	-	-	-	-	-	-	0	0	1	1	Q:=3; Y:=3
-	-	-	-	0	0	0	0	0	0	0	0	RG0:=0; Y:=0; Загрузка RGC в стек
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RG0+1; Y:=RG0
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ-1; Y:=RGQ; Переход по стеку если Y<>0
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0; Переход на 0000
Б												
-	-	-	-	-	-	-	-	0	0	1	1	Q:=3; Y:=3
-	-	-	-	0	0	0	0	0	0	0	0	RG0:=0; Y:=0; Загрузка RGC в стек
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RG0+1; Y:=RG0
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ+1; Y:=RGQ; Переход по стеку если C<>1
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0; Переход на 0000
В												
-	-	-	-	-	-	-	-	0	0	1	1	Q:=3; Y:=3
-	-	-	-	0	0	0	0	0	0	0	0	RG0:=0; Y:=0
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RG0+1; Y:=RG0
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ-1; Y:=RGQ; Переход 0010 если RGQ<>0
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0; Переход на 0000
Г												
-	-	-	-	-	-	-	-	0	0	1	1	Q:=3; Y:=3
-	-	-	-	0	0	0	0	0	0	0	0	RG0:=0; Y:=0
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RG0+1; Y:=RG0
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ+1; Y:=RGQ; Переход 0101 если C=1
-	-	-	-	-	-	-	-	-	-	-	-	Переход на 0010
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0; Переход на 0000
Д												
-	-	-	-	-	-	-	-	0	0	1	1	Q:=3; Y:=3
-	-	-	-	0	0	0	0	0	0	0	0	RG0:=0; Y:=0; Переход к МПП
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0; Переход на 0000
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RG0+1; Y:=RG0
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ-1; Y:=RGQ; Переход 0011 если RGQ<>0
-	-	-	-	-	-	-	-	-	-	-	-	Возврат из подпрограммы
Е												
-	-	-	-	-	-	-	-	0	0	1	1	Q:=3; Y:=3
-	-	-	-	0	0	0	0	0	0	0	0	RG0:=0; Y:=0; Загрузка RGC в стек
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ-1; Y:=RGQ; Переход 0100 если RGQ<>0
-	-	-	-	0	0	0	0	-	-	-	-	Y:=RG0; Переход на 0000
-	-	-	-	0	0	0	0	-	-	-	-	RG0:=RG0+1; Y:=RG0; Переход по вершине стека
Ж												
-	-	-	-	-	-	-	-	-	-	-	-	Безусловный переход по векторному адресу
-	-	-	-	-	-	-	-	0	0	0	0	RGQ:=0; Y:=0; Переход к ММП Е
-	-	-	-	-	-	-	-	-	-	-	-	Y:=RGQ; Переход на 0000
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ+1; Y:=RGQ; Переход к ММП F
-	-	-	-	-	-	-	-	0	0	1	1	RGQ:=RGQ+3; Y:=RGQ; Переход к ММП G
-	-	-	-	-	-	-	-	-	-	-	-	Возврат из ММП Е
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ-1; Y:=RGQ; Возврат из ММП F
-	-	-	-	-	-	-	-	0	1	0	1	RGQ:=RGQ+5; Y:=RGQ; Переход к ММП F
-	-	-	-	-	-	-	-	0	0	1	0	RGQ:=RGQ+2; Y:=RGQ; Переход к ММП K
-	-	-	-	-	-	-	-	-	-	-	-	Возврат из ММП G
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=RGQ-1; Y:=RGQ; Переход к ММП L
-	-	-	-	-	-	-	-	-	-	-	-	Возврат из ММП K
-	-	-	-	-	-	-	-	0	0	0	1	RGQ:=RGQ-2; Y:=RGQ; Переход к ММП F
-	-	-	-	-	-	-	-	-	-	-	-	RGQ:=2RGQ; Y:=?; Возврат из ММП L