

## ОРГАНИЗАЦИЯ ПОДПРОГРАММ И ЦИКЛОВ

Цель работы : приобретение навыков практической работы со стеком при организации в микропрограмме подпрограмм и циклов.

### 3.1. Индивидуальное задание

1). Самостоятельно разработать четыре небольших фрагмента граф-схем микропрограмм, содержащих по 2-3 операторных и 2-3 условных вершины.

2). Оформить фрагменты граф-схем в виде основной программы и трех подпрограмм. Основная программа должна осуществлять вызов первой подпрограммы, та в свою очередь - второй, а вторая - третьей подпрограммы. Разработать граф-схему микропрограммы функционирования БМУ, разместить микрокоманды в управляющей памяти и обозначить содержимое стека при выполнении стековых операций.

3). Для одного-двух фрагментов организовать циклы с использованием стека.

### 3.2. Методические указания

Использование подпрограмм позволяет обращаться в различных местах основной программы к ее повторяющемуся фрагменту. При этом уменьшается количество ячеек памяти, занимаемой программой.

Во многих ЭВМ одна подпрограмма может вызвать другую, которая, в свою очередь, может вызвать третью подпрограмму и т. д. Для этого необходимо организовать "вложение" адресов возврата, служащих для связывания частей программы. Пример вложения подпрограмм показан на рис. 3.1. Организовать выборку адресов возврата в порядке, обратном их поступлению (дисциплина обслуживания типа LIFO), обеспечивает стековая память, имеющаяся в составе БИС K1804BY4 [1, 2].

Для обращения к подпрограмме можно использовать команду CJS (условный переход к подпрограмме), а для возврата из подпрограммы - команду CRTN (условный возврат из подпрограммы). Будем полагать, что сигнал на входе ССЕ БМУ имеет низкий уровень. Это приведет к отмене проверки условия  $\overline{CS}$  и принудительному выполнению названных операций.

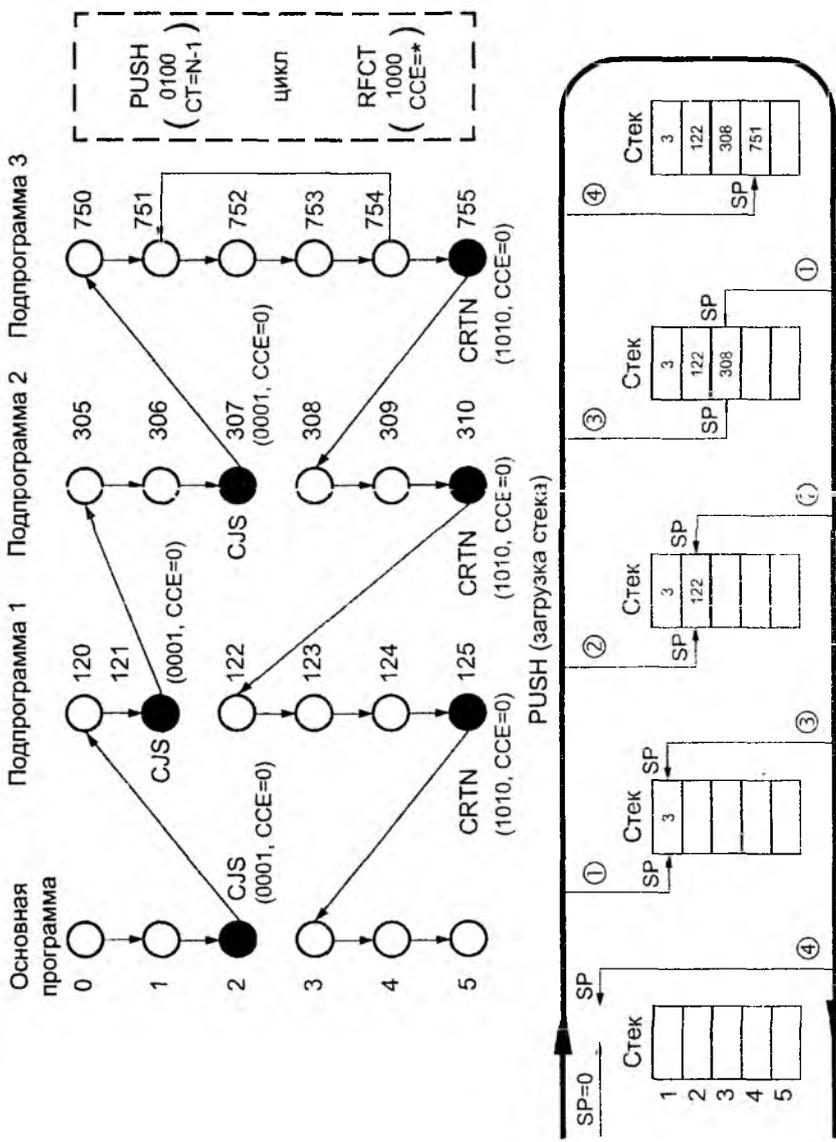


Рис. 3.1. Пример организации подпрограмм.

В основной программе перед обращением к подпрограмме по команде JZ производится начальная установка указателя стека ( $SP := 0$ ). При выполнении команды CJS в основной программе происходит передача управления по адресу 120 первой подпрограмме, а в первую ячейку стека записывается адрес 3 возврата в основную программу. При выполнении в подпрограмме команды CRTN адрес возврата извлекается из стека и управление передается основной программе.

Для организации в микропрограмме циклов обычно используются команды PUSH и RFCT. На рис. 3.1 цикл показан при выполнении третьей подпрограммы. По команде PUSH, находящейся по адресу 750, в стек загружается адрес 751, а в регистр-счетчик CT из регистра микрокоманд - величина, на единицу меньшая требуемого числа повторений (не более 4095). При выполнении команды RFCT проверяется, отлично ли от нуля содержимое счетчика CT. Если это так, то происходит отрицательное приращение содержимого счетчика, и адрес следующей микрокоманды извлекается из верхней части стека (это адрес 751). Если  $CT = 0$ , то это рассматривается как условие выхода из цикла. Управление в этом случае передается следующей по порядку микрокоманде с адресом 755. Кроме того, выполняется отрицательное приращение указателя стека SP, однако находящийся в вершине стека адрес не используется.

Загрузка стека в микросхеме K1804BY4 происходит следующим образом [1]. По положительному фронту тактового сигнала CLK происходит увеличение содержимого указателя стека SP на единицу по правилу  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$  и на следующем такте производится запись адреса возврата в регистр накопителя. После заполнения стека на выходе FL указателя стека появляется предупреждающий сигнал ( $\overline{FL} = 0$ ). Любая попытка загрузить заполненный стек стирает информацию в вершине стека, оставляя указатель стека неизменным.

Выталкивание из стека состоит в следующем. Происходит считывание информации из вершины стека, после чего на следующем такте производится уменьшение указателя стека на единицу по следующему правилу  $5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$ . Если стек пуст, происходит считывание неопределенной информации, но содержимое указателя стека остается равным нулю.

В микросхеме K1804BY1 глубина кольцевого стека равна 4. Стек управляется двумя сигналами: FE и PUP. Изменение значения указателя стека происходит по фронту сигнала CLK, а направление изменения (увеличение или уменьшение на единицу) зависит от значения сигнала PUP. При  $\overline{FE} = 0$  и  $PUP = 0$  по фронту сигнала CLK значение указателя стека SP изменяется по правилу:

$2 \rightarrow 1 \rightarrow 0 \rightarrow 3.$   


## Практическая работа N 4

### АЛГОРИТМЫ УМНОЖЕНИЯ ЦЕЛЫХ ЧИСЕЛ С ФИКСИРОВАННОЙ ЗАПЯТОЙ

Цель работы : приобретение практических навыков составления микропрограмм при выполнении операций умножения чисел с фиксированной запятой.

#### 4.1. Индивидуальное задание

1). Разработать граф-схему микропрограммы и таблицу кодирования при выполнении операции умножения 8 - разрядных двоичных целых чисел по алгоритму, заданному в табл. 4.1 - табл. 4.3. Первоначально операнды расположены в регистрах общего назначения центрального процессора (табл. 4.4).

2). Предусмотреть выполнение контрольного примера над операндами, заданными в табл. 4.5. При выполнении задания использовать структурную схему БМУ, приведенную на рис. 2.2. Разрядность процессора принять равной разрядности исходных операндов.

Таблица 4.1

Вариант= $(N)_{\text{mod}4}$	Алгоритм умножения
0	A [ R1 ( $\Sigma$ ) , млад. Мт ]
1	Б [ L1 ( Мн ) , млад. Мт ]
2	В [ L1 ( $\Sigma$ ) , стар. Мт ]
3	Г [ R1 ( Мн ) , стар. Мт ]

Таблица 4.2

Вариант= $(N)_{\text{mod}3}$	Представление операндов
0	Прямой код
1	Дополнительный код
2	Обратный код

Таблица 4.3

Вариант= $(N)_{\text{mod}4}$	Операционная и управляющая части
0	K1804BC1, ВУ1
1	K1804BC2, ВУ4
2	K1804BC1, ВУ4
3	K1804BC2, ВУ1

Таблица 4.4

Вариант= $(N)_{\text{mod}4}$	Мн	Мт
0	R1	RQ
1	R10	R6
2	RQ	R7
3	R8	R13

Таблица 4.5

Вариант= $(N)_{\text{mod}5}$	Мн	Мт
0	- 5	+ 7
1	+ 6	- 4
2	- 10	- 23
3	+ 4	- 7
4	- 5	+ 7

#### 4.2. Методические указания

В качестве примера рассмотрим реализацию алгоритма умножения по А в дополнительном коде на процессорных элементах K1804BC2 [ 1 ].

Умножение по алгоритму А выполняется с младших разрядов множителя со сдвигом частичных сумм вправо на один разряд. Множимое в этом алгоритме неподвижно. В процессе перемножения чисел в дополнительном коде в результате получается одновременно знаковая и цифровая части произведения. Результат произведения дополнительных кодов сомножителей равен дополнительному коду результата в случае положительного множителя. Если множитель отрицательный, то произведение чисел в дополнительном коде получается прибавлением поправки (-Мн) к произведению дополнительных кодов сомножителей. Пример выполнения операции умножения приведен на рис. 4.1.

Начальные условия выполнения алгоритма умножения следующие: регистр R0 установлен в нулевое состояние, множимое находится в регистре R1, множитель - в R2. В результате выполнения умножения двух n разрядных сомножителей получается 2n - разрядное произведение, n старших разрядов которого находятся в регистре R0, а n младших разрядов - в регистре RQ.

Соединение шин при выполнении умножения в дополнительном коде показано на рис. 4.2. На входе C0 должен быть "0" до последнего такта, когда он должен принимать значение сигнала на шине Z. Поэтому перед входом C0 ставится вентиль, на который подается сигнал Z и разрешающий сигнал OEZ, принимающий единичное значение только на последнем такте.

**R1=Mn=A= +5, R2=Mt=B= -3**

	3	R1 0	3	RQ 0	
	0	101	,		A
	1	101	,		Вдк
+	0	000		0000	Σ0
	0	101			A
	0	101		0000	Σ1
+	0	010		1000	R1(Σ1)
	0	000			0
	0	010		1000	Σ2
+	0	001		0100	R1(Σ2)
	0	101			A
	0	110		0100	Σ3
+	0	011		0011	R1(Σ3)
	1	011			Адк (кор)
	1	110		001,0	Σ4,
	1	111		000 1	, Пдк=Σ4/2

**Примечание:**

Пдк - форматное произведение в ДК

Рис. 4.1. Алгоритм умножения

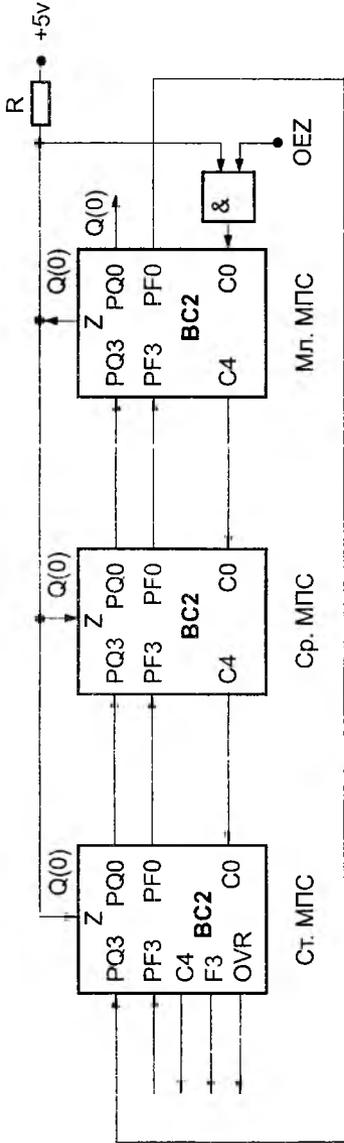


Рис.4.2. Структурная схема алгоритма умножения

Таблица 4.6.

Таблица кодирования при умножении по алгоритму А

Адр. МК	МИКРОКОМАНДА												Примечание
	Управление К1804BC2						Управление К1804БУ4						
	I	0	OEY	AY	AB	CFE	OFZ	RFD	I <sup>0</sup>	D <sup>0</sup>			
8765	4321	0	OEY	AY	AB	CFE	OFZ	RFD	I <sup>0</sup>	D <sup>0</sup>			
112	0110	1	X	R2	X	1	0	0	4	14			Операция ВС2
113	0010	0	0	R1	R0	X	0	1	8	X			Умножение в ДК
114	0110	0	0	R1	R0	X	1	1	14	X			Коррекция

На рис. 4.3 представлена граф-схема микропрограммы, а в табл. 4.6 - таблица кодирования при умножении чисел по алгоритму А в дополнительных кодах. После начальной установки множитель из регистра R2 засылается в регистр RQ. В счетчик тактов СТ БМУ устанавливается код 14, а в стек загружается адрес 113 для организации цикла умножения.

Цикл умножения чисел в дополнительном коде реализуется второй микрокомандой. Здесь имеется в виду, что признак Z совпадает с разрядом RQ[0] младшей процессорной секции. По каждому положительному фронту сигнала CLK выходы АЛУ сдвигаются в сторону младших разрядов и полученное частичное произведение записывается в регистр R0. При этом в старший разряд PF3 при реализации спецфункции умножения ДК поступает сигнал  $F3 \oplus OVR$ , чем обеспечивается передача "F<sub>3</sub>" при возникновении переполнения при сложении в АЛУ BC2. Младший разряд сдвинутого частичного произведения передается с выхода PF0 на вход PQ3. Одновременно выполняется сдвиг множителя в регистре RQ на один разряд вправо.

На  $(n + 1)$ -м такте знаковый разряд множителя находится на шине Z младшей процессорной секции. В это время необходимо подать микрокоманду последнего такта умножения в дополнительном коде (коррекция результата).

Если сомножители представлены в прямом коде, то знак произведения формируется отдельно от цифровой части произведения. Знак произведения получают в виде суммы по модулю два знаковых разрядов множимого и множителя. В конце операции умножения его следует переслать на место знакового разряда регистра результата.

При представлении сомножителей в целом формате в процессе операции умножения должны быть получены все  $2n$  разрядов произведения. Числовой пример умножения чисел должен быть выполнен с целью проверки результата в десятичной и в двоичной системах счисления.

При нестандартном подключении процессорных секций должна быть приведена функциональная схема их подключения.

Начальные условия:  $(MH)_{ДК} \rightarrow R1$ ,  $(MT)_{ДК} \rightarrow R2$ ,  $0 \rightarrow R0(\Sigma)$   
 112, 113, 114 - адреса микрокоманд ВС2

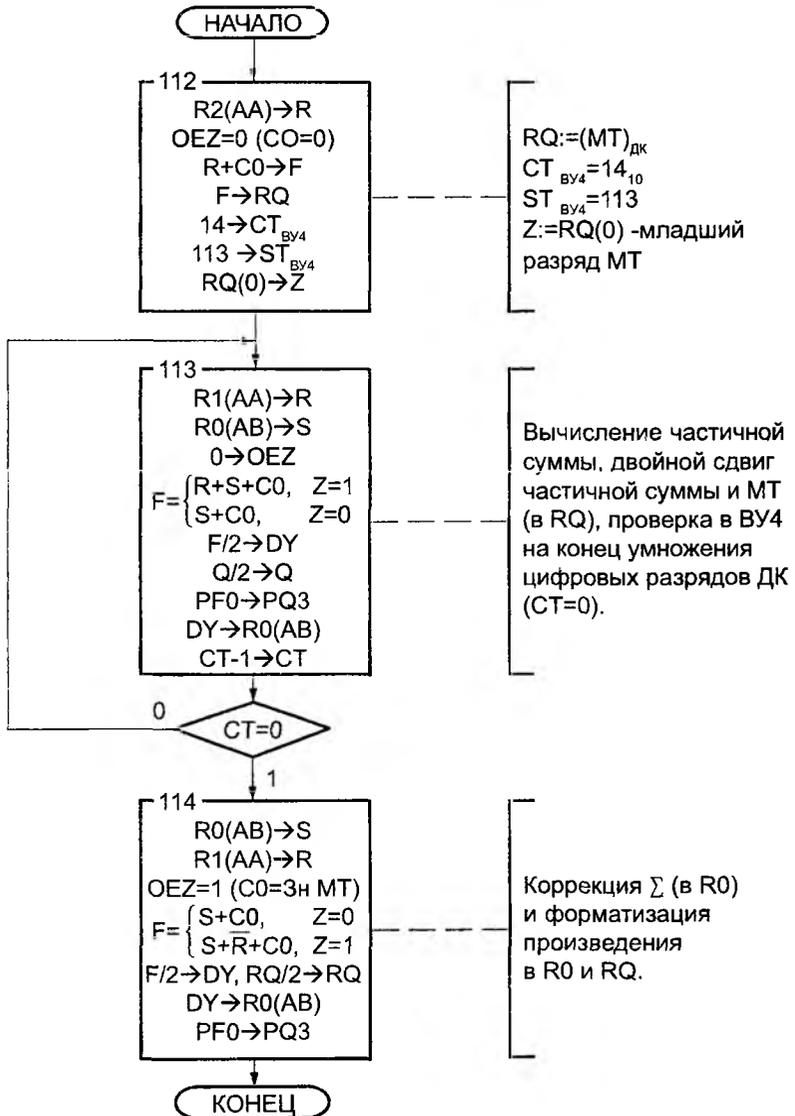


Рис. 4.3. Граф-схема микропрограммы умножения дополнительных кодов (ДК) в ВС2

## АЛГОРИТМЫ ДЕЛЕНИЯ ЦЕЛЫХ ЧИСЕЛ ОДИНАРНОГО И ДВОЙНОГО ФОРМАТОВ

Цель работы : приобретение практических навыков составления микропрограмм при выполнении операции деления целых чисел с фиксированной запятой в прямом и дополнительном кодах.

### 5.1. Индивидуальное задание

1). Разработать граф-схему микропрограммы и таблицу кодирования при выполнении операции деления 8 - разрядных целых чисел по алгоритму, заданному в табл. 5.1. Представление операндов и элементы операционной части взять из предыдущей работы. Выполнить контрольный пример над операндами, представленными в табл. 5.2. Разрядность процессора принять равной разрядности исходных операндов. Блок микропрограммного управления выполнить на базе БИС К1804ВУ4 (с использованием регистра состояния для хранения признаков).

2). Оформить граф-схему микропрограммы в виде подпрограммы. Обеспечить задание исходных данных из РМК и вызов подпрограммы из основной программы.

Таблица 5.1

Вариант= $(N)_{\text{mod}4}$	Алгоритм деления
0	а (ДК)
1	б (ДК)
2	а (ПК)
3	б (ПК)

Таблица 5.2

Вариант= $(N)_{\text{mod}16}$	Дм	Дт	Дм	Дт
0	R11	R15	$\pm 76$	$\pm 7$
1	R2	RQ	$\pm 58$	$\pm 6$
2	R3	R4	$\pm 64$	$\pm 5$
3	RQ	R5	$\pm 83$	$\pm 4$
4	R1	R2	$\pm 102$	$\pm 7$
5	R3	RQ	$\pm 96$	$\pm 6$
6	RQ	R4	$\pm 72$	$\pm 5$
7	R6	R10	$\pm 63$	$\pm 4$
8	R0	R15	$\pm 110$	$\pm 10$
9	R2	RQ	$\pm 120$	$\pm 11$
10	RQ	R4	$\pm 115$	$\pm 12$
11	R4	R5	$\pm 96$	$\pm 5$
12	R6	R7	$\pm 99$	$\pm 7$
13	R8	RQ	$\pm 68$	$\pm 5$
14	R9	R11	$\pm 56$	$\pm 4$
15	R10	R14	$\pm 77$	$\pm 6$