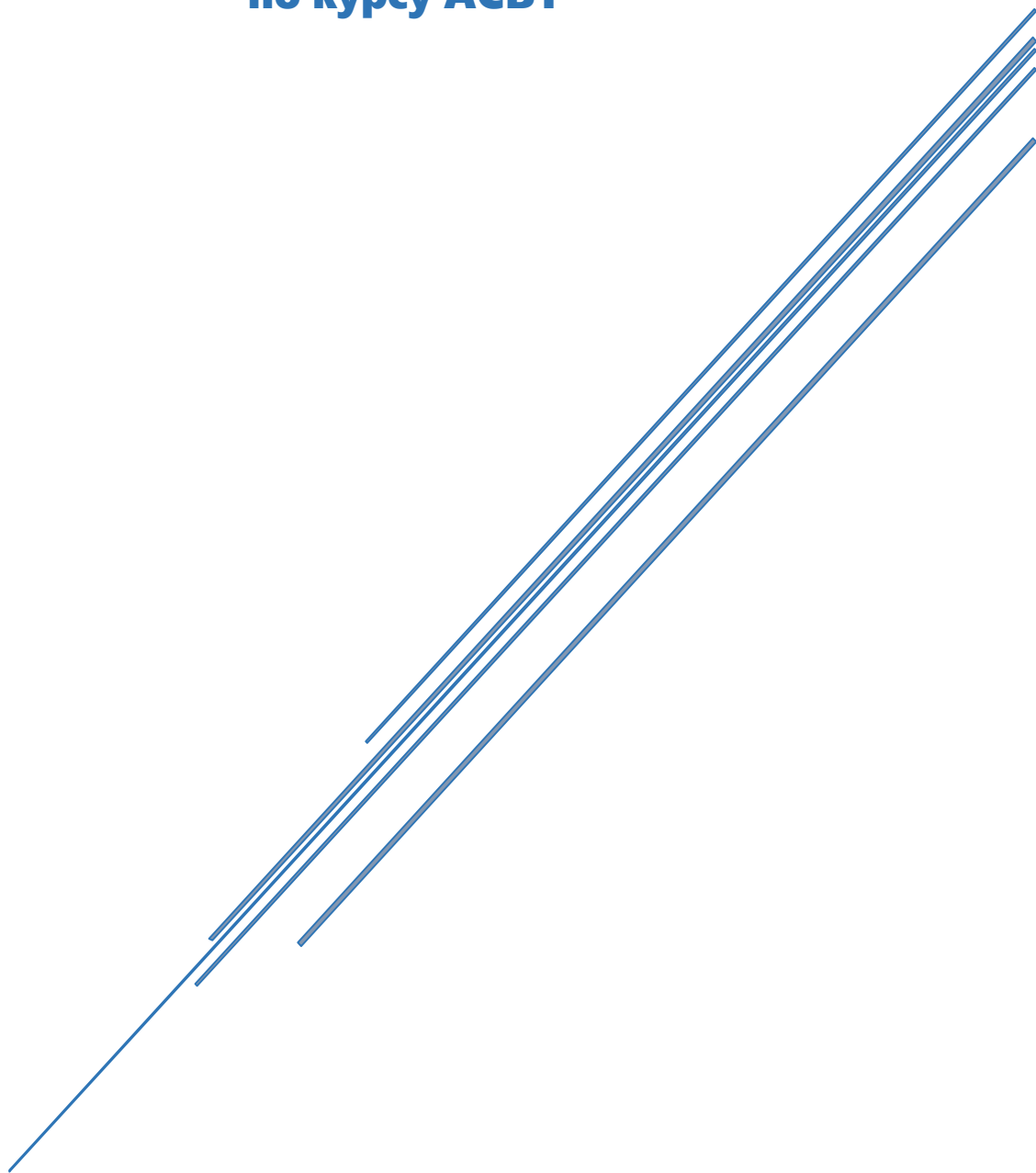


# **МИКРОПРОЦЕССОРЫ**

## **с микропрограммным управлением**

**Методические указания к Лабораторным работам  
по курсу АСВТ**



**МГТУ им. Н.Э. Баумана**  
**ИУ-8**  
**МОСКВА 2022**

Данный практикум предназначен для изучения секционных процессорных элементов, блоков микропрограммного управления, микропрограммирования операций в микропроцессорных системах с разрядно – модульной архитектурой.

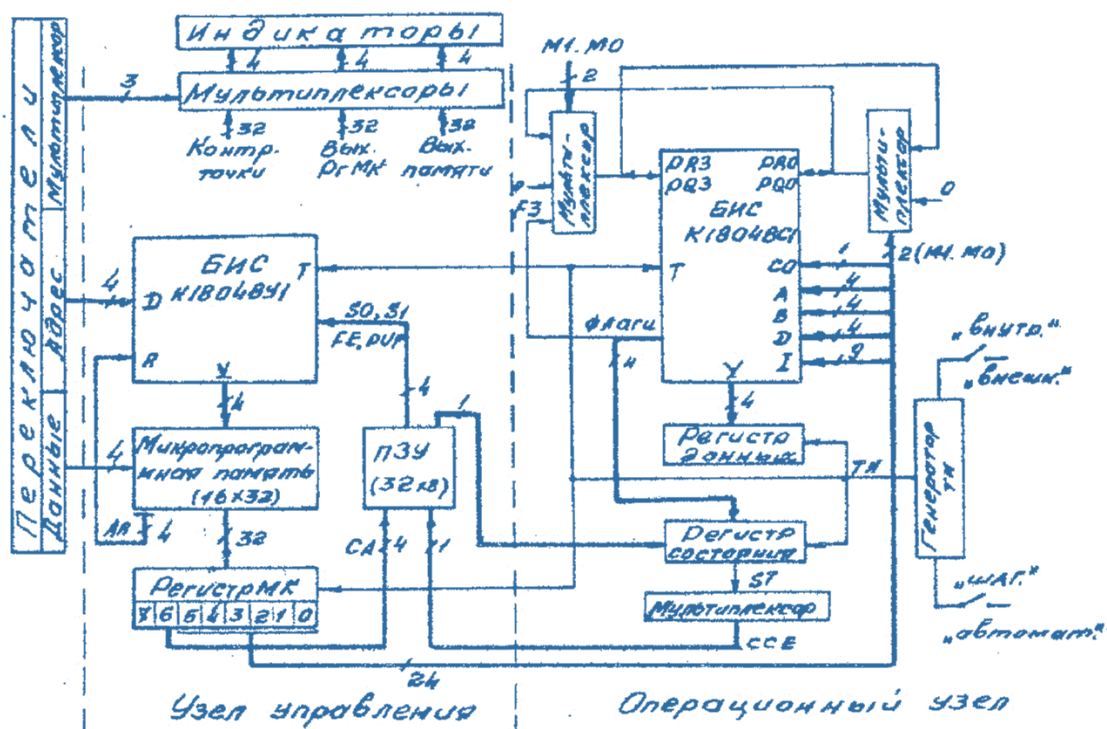
## КРАТКОЕ ОПИСАНИЕ УСТРОЙСТВА «МИКРОТРЕНАЖЕР МТ1804» (МТ1804)

### 1. Назначение и структура

Устройство **МТ1804** предназначено для изучения архитектуры, основ микропрограммирования и применения МПК серии К1804. Для работы устройства необходим источник питания постоянного тока напряжением  $+5\text{ В} \pm 5\%$  и максимальным током нагрузки не менее  $2,5\text{ А}$ .

Структурная схема МТ1804 приведена на рис.1. Функционально устройство делится на узел управления и операционный узел. Узел управления содержит: БИС управления адресом микрокоманды К1804ВУ1, микропрограммную память объемом шестнадцать 32-разрядных слов на микросхемах К155РУ2, 32-разрядный регистр микрокоманд (МК) на микросхемах К1804ИР1, ПЗУ для управления выборкой следующего адреса объемом тридцать два байта на микросхеме К155РЕ3, внутренний тактовый генератор с частотой 1 МГц. Операционный узел содержит: БИС центрального процессора К1804ВС1, мультиплексоры сдвига, регистр выходных данных, регистр состояния и мультиплексор флагов состояния.

Регистр МК, данных и регистр состояния имеют общую линию синхронизации от тактовых импульсов (ТИ). Запись информации в эти регистры происходит по положительному фронту (0/1) ТИ, т.е. в регистры записывается информация, представленная на входах этих регистров к моменту поступления ТИ. По этой причине установка признаков (флагов) в регистре состояния для последующего использования происходит на такт позже по сравнению с моментом их появления на выходах процессора. Выходные данные процессора  $Y$  записываются в выходной регистр данных также с задержкой на один такт.



микропрограммной памяти (АДРЕС), записываемого кода данных (ДАННЫЕ), для управления мультиплексорами (МУЛЬТИПЛЕКСОР); кнопки ПУСК и ЗАГРУЗКА.

Для контроля информации в **MT1804** установлено 12 светодиодных индикаторов (3 секции по 4 индикатора в каждой): правая секция - для индикации содержимого адресуемой ячейки микропрограммной памяти, средняя - регистра МК, левая - для индикации данных на основных шинах устройства. Подключение выходов памяти, регистра МК и контрольных точек (шин) к светодиодам осуществляется потетрадно через мультиплексоры. В зависимости от номера, устанавливаемого на переключателях МУЛЬТИПЛЕКСОР, к индикаторам подключается определенная группа разрядов памяти, регистра МК (табл. 1) или контрольных точек (табл. 2).

Таблица 1

# тетрады	7	6	5	4	3	2	1	0
# разряда	31-28	27-24	23-20	19-16	15-12	11-08	07-04	03-0

Код	Индикаторы данных				Функция
	3	2	1	0	
000	Y3	Y2	Y1	Y0	Выходы K1804BY1
001	Y3	Y2	Y1	Y0	Выходы KI804BCI
010	C4	OVR	F3	Z	Флаги KI804BCI
011	C4*	CCE	/P	/G	*)
100	ST3	ST2	ST1	ST0	Выходы регистра состояния
101	PQ3	PQ0	PR3	PR0	Входы/выходы сдвига KI804BCI
110	Y3'	Y2'	Y1'	Y0'	Выходы регистра выходных данных
III	Y3	Y2	Y1	Y0	То же

Таблица 2

## 2. Основные режимы работы MT1804

Устройство **MT1804** может работать в режиме ЗАГРУЗКА для программирования микропрограммной памяти и в режиме РАБОТА для выполнения микропрограммы.

2.1 Для выполнения режима ЗАГРУЗКА необходимо:

- переключатель режима установить в положение ЗАГРУЗКА,
- переключателями АДРЕС набрать код адреса ячейки,
- переключателями МУЛЬТИПЛЕКСОР установить двоичный номер тетрады,
- переключателями ДАННЫЕ набрать код, записываемый в указанную тетраду ячейки,
- нажав кнопку ЗАГРУЗКА, произвести запись,
- для загрузки 32-разрядной ячейки памяти действия в)-д) повторить 8 раз.

Контроль записываемой информации осуществляется по показаниям светодиодов памяти.

2.2. Для запуска программы на микропрограммном уровне необходимо произвести начальную установку регистра МК (инициализацию), т.е. ввести в регистр МК микрокоманду со стартовым адресом программы. Для этого после загрузки всей программы на переключателях АДРЕС нужно установить значение стартового адреса и нажать кнопку ПУСК. При этом из памяти по указанному

адресу будет выбрана микрокоманда, помещаемая в регистр МК, в чем можно убедиться с помощью индикаторов регистра МК.

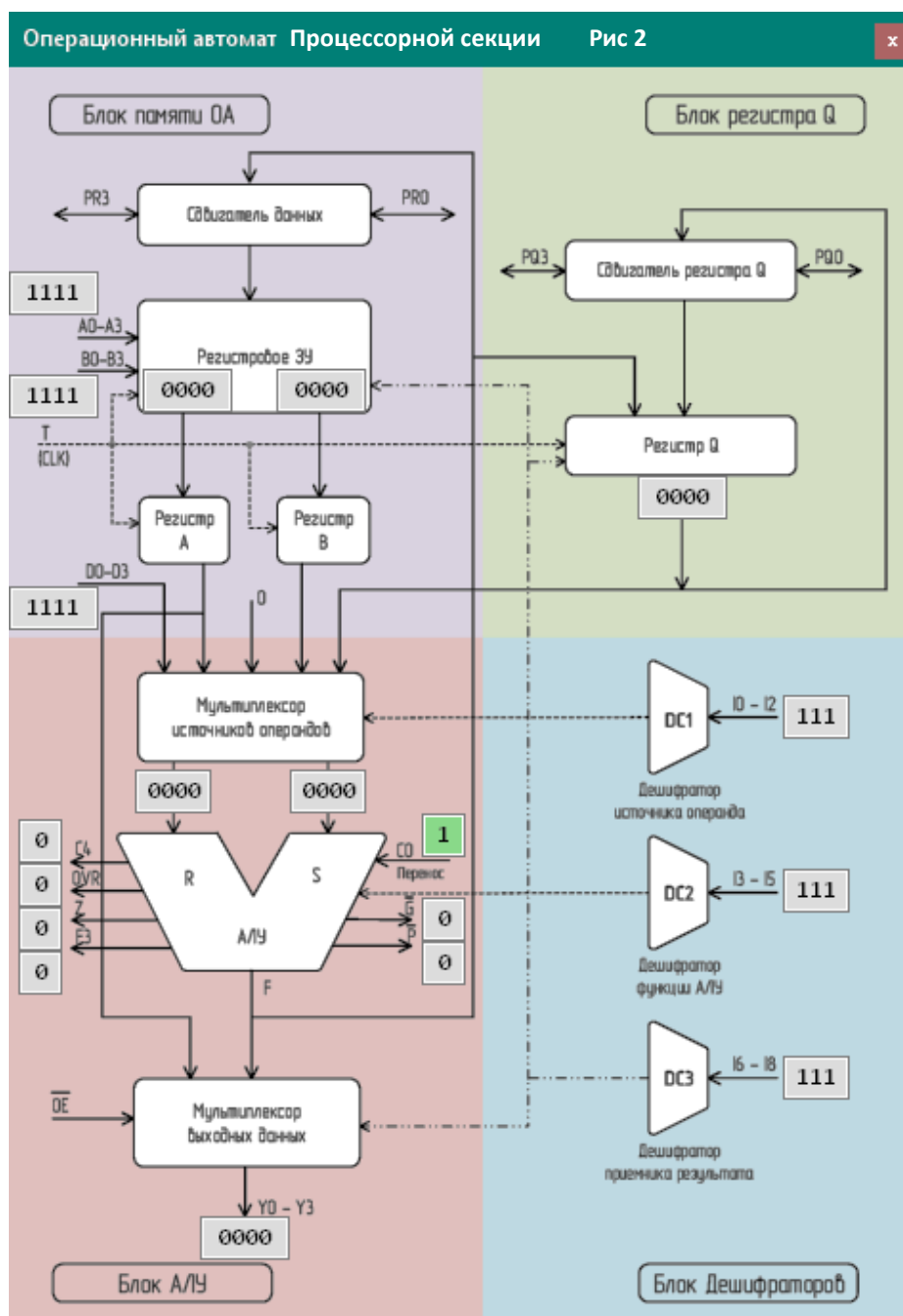
2.3. Для дальнейшего выполнения программы переключатель режима следует перевести в положение РАБОТА. В шаговом режиме при однократном нажатия кнопки ПУСК выполняется одна микрокоманда, в автоматическом режиме - вся программа.

## Работа № 1. ИССЛЕДОВАНИЕ ФУНКЦИЙ ПРОЦЕССОРА

**Цель работы** - изучение устройства МТ1804; изучение структуры, принципа действия и функций процессорного элемента (секции АЛУ операционного устройства) - К1804ВС1; программирование и выполнение линейных микропрограмм.

### Теоретическая часть

Функционирование ОА процессора рассмотрим по структурной схеме (рис. 2). На схеме можно выделить четыре блока: внутренней памяти, арифметико - логический, регистра Q, управления. Блок внутренней памяти содержит: регистровое запоминающее устройство (РЗУ), имеющее шестнадцать 4-разрядных регистров общего назначения (РОН); сдвигатель данных (СДА), позволяющий записывать в РЗУ информацию без сдвига и со сдвигом вправо или влево на один разряд; два регистра RгА и RгВ на выходе блока. Выбор регистров РЗУ как источников информации осуществляется по адресам на входах А и В. Информация из РОНов поступает на регистры RгА, RгВ по сигналу логической единицы (1) на тактовом входе Т. При сигнале логического нуля (0) эти регистры находятся в режиме хранения.



Запись информации в регистр РЗУ возможна только по адресу В и происходит при поступлении сигнала 0 на вход Т.

Данные (F) с выхода АЛУ загружаются в РОН без сдвига или со сдвигом влево или вправо по схеме (рис. 3) в зависимости от кода управления на входах I8-6.

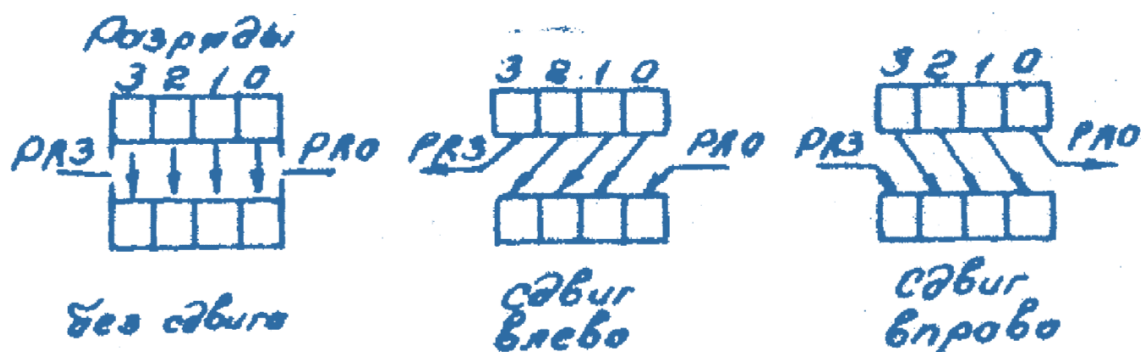


Рис. 3

Арифметико-логический блок содержит арифметико-логическое устройство (АЛУ), выполняющее 8 операций над операндами R и S в зависимости от кода управления на входах I5-3 (табл. 3).

Арифметические операции в АЛУ выполняются с учетом значения сигнала входного переноса CO в дополнительном коде. На выходах АЛУ формируются 4 флага (признака) результата: перенос из старшего разряда C4, переполнение  $OVR = C4 \oplus C3$ , знак (или содержимое старшего разряда АЛУ) F3 и признак нулевого результата  $z=1$ , если  $F=0$ . При выполнении логических операций C4 и OVR вычисляются иначе /см. [1]/.

Кроме того, формируются сигналы генерации G и распространения P переноса из АЛУ, необходимые для организация ускоренного переноса в многоразрядной схеме, построенной из нескольких секций процессора.

Таблица 3

I5-3	Функция АЛУ (F)
000	$R+S+CO$
001	$S-R-1+CO$
010	$R-S-1+CO$
011	$R \vee S$
100	$R \cdot S$
101	$\overline{R} \cdot S$
110	$R \oplus S$
111	$\overline{R} \oplus \overline{S}$

Источниками операндов R и S могут быть регистры РЗУ, внешняя шина данных D, выделенный регистр Q и шина "0". Выбор источников по входам R и S проводится с помощью селектора источника данных (СИД), управляемого кодом 12-0 (табл. 4). Результат операции (F) из АЛУ поступает на селектор выходных данных (СВД), на сдвигатель СДА и регистр Q. Приемник результата (адресуемый по адресу В регистр общего назначения в РЗУ, регистр Q или выходная шина Y) зависит от кода управления на входах I8-6 (табл. 5).

Таблица 4

<b>I2-0</b>	<b>Источники операндов</b>	
	<b>R</b>	<b>S</b>
<b>000</b>	РОН(A)	<b>PQ</b>
<b>001</b>	РОН(A)	РОН(B)
<b>010</b>	<b>0</b>	<b>PQ</b>
<b>011</b>	<b>0</b>	РОН(B)
<b>100</b>	<b>0</b>	РОН(A)
<b>101</b>	<b>D</b>	РОН(A)
<b>110</b>	<b>D</b>	<b>PQ</b>
<b>111</b>	<b>D</b>	<b>0</b>

Таблица 5

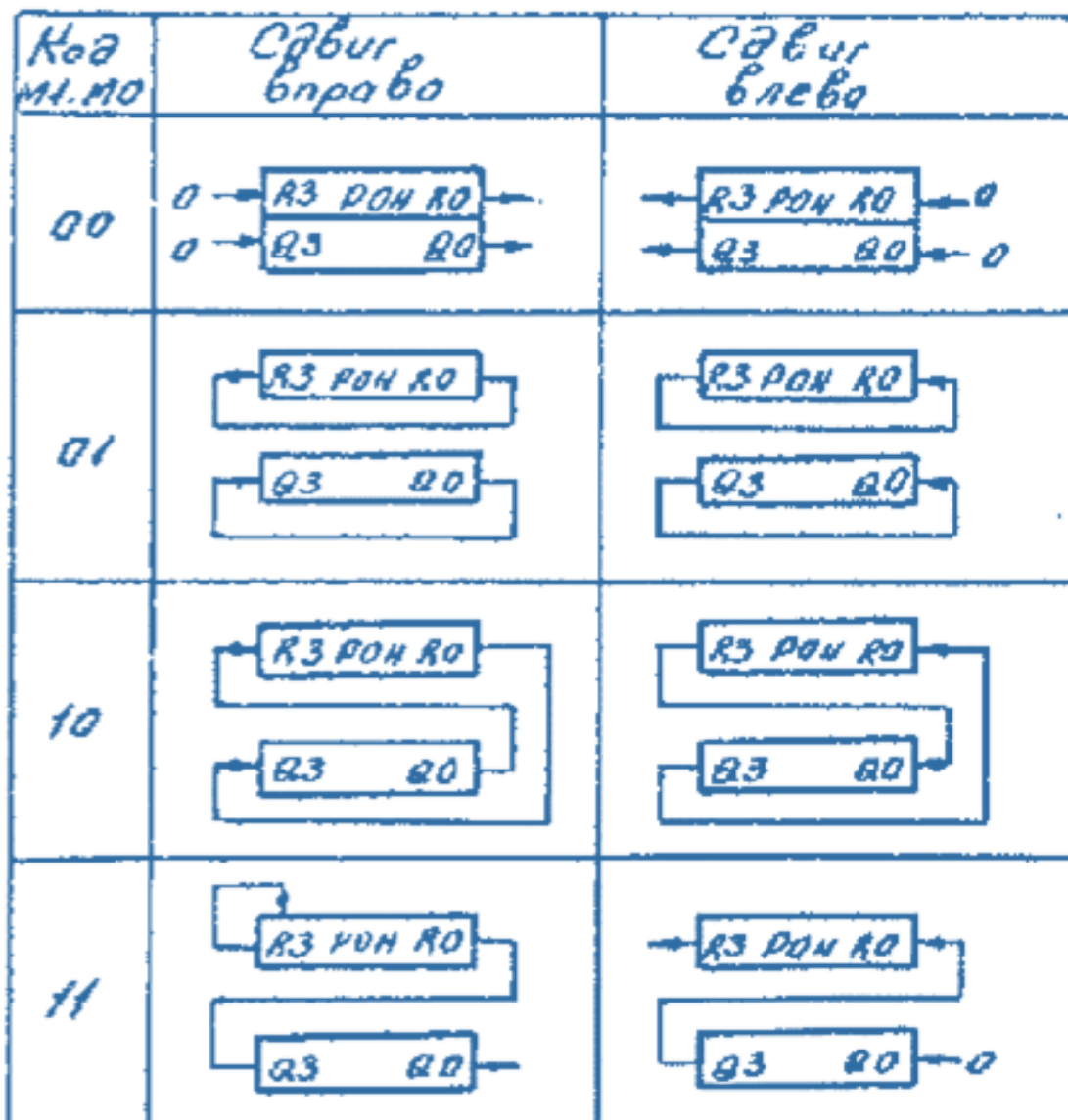
<b>I8-6</b>	<b>Приемник результата</b>	
	<b>Тип загрузки</b>	<b>Выход Y</b>
<b>000</b>	<b>PQ</b>	<b>F</b>
<b>001</b>	Нет загрузки	<b>F</b>
<b>010</b>	РОН(B)	<b>A</b>
<b>011</b>	РОН(B)	<b>F</b>
<b>100</b>	<b>F/2→РОН(B) Q/2→PQ</b>	<b>F</b>
<b>101</b>	<b>F/2→РОН(B)</b>	<b>F</b>
<b>110</b>	<b>2F→РОН(B) 2Q→PQ</b>	<b>F</b>
<b>111</b>	<b>2F→РОН(B)</b>	<b>F'</b>

Раздельное управление операциями в АЛУ и приемником результата позволяет реализовать в одной микрокоманде комбинированные операции: сложения-вычитания со сдвигом влево или вправо, логические операции со сдвигом влево или вправо и др. Для организации логического, циклического или арифметического сдвигов влево или вправо при одинарной или двойной длине операндов (рис. 4) необходимо управлять входами/выходами PR0, PR3, PQ0, PQ3 процессора. Это осуществляется путем соответствующей коммутации входов/выходов и линий сигналов логического 0 и F3. Для этой цели в схеме (см. рис. 1) используются мультиплексоры, установленные в цепи сдвигов. Управление мультиплексорами осуществляется 2-разрядным кодом (M1, M0) микрокоманды, определяющим тип сдвига, и кодом 18-6, определяющим направление сдвига.

Блок регистра Q состоит из Q - регистра (PQ) и сдвигателя регистра (СДР), который осуществляет сдвиг содержимого PQ вправо или влево на один разряд.

Блок управления вырабатывает управляющие сигналы на остальные блоки процессора в зависимости от кода I8-0 на входах.

В устройстве МТ1801 применяются 32-разрядные микрокоманды. Формат микрокоманды показан в табл. 6. В табл. 7 приведены примеры программирования некоторых операций процессора К1804ВС1. Незаполненные поля в микрокомандах обозначают несущественное значение их для данной операции. XXXX - исходные данные на шине D процессора, устанавливаемые через тетраду 0 регистра МК.



00 - логический сдвиг, 01 - циклический сдвиг,  
10 - циклический сдвиг двойной длины  
11 - арифметический сдвиг двойной длины

Рис. 4



Таблица 6

Номер тетрады	Номер бита	Назначение бита	Функция
0	0	<i>D0</i>	Данные для D-шины
	1	<i>D1</i>	
	2	<i>D2</i>	
	3	<i>D3</i>	
1	4	<i>B0</i>	Адрес РОН на входах В
	5	<i>B1</i>	
	6	<i>B2</i>	
	7	<i>B3</i>	
2	8	<i>A0</i>	Адрес РОН на входах А
	9	<i>A1</i>	
	10	<i>A2</i>	
	11	<i>A3</i>	
3	12	<i>I3</i>	Функция АЛУ
	13	<i>I4</i>	
	14	<i>I5</i>	
	15	<i>CO</i>	Значение входного переноса в АЛУ
4	16	<i>I0</i>	Указатель операндов в АЛУ
	17	<i>I1</i>	
	18	<i>I2</i>	
	19	<i>M0</i>	Нулевой бит управления мультиплексорами сдвига
5	20	<i>I6</i>	Определение приемника результата операции
	21	<i>I7</i>	
	22	<i>I8</i>	
	23	<i>M1</i>	Первый бит управления мультиплексорами сдвига
6	24	<i>CA0</i>	Управление выборкой адреса следующей микрокоманды
	25	<i>CA1</i>	
	26	<i>CA2</i>	
	27	<i>CA3</i>	
7	28	<i>AR0</i>	Адрес перехода
	29	<i>AR1</i>	
	30	<i>AR2</i>	
	31	<i>AR3</i>	

Порядок прохождения микропрограммы в устройстве рассмотрим на примере выполнения последовательности микрокоманд МК0, МК5, МК5 размещенных в памяти по адресам 0, 5 (табл. 7 и табл. 8).

В табл. 8 представлен процесс обработки микрокоманд, учитывающий действия оператора и свойства процессора.

В таблице использованы следующие обозначения: значения тактового импульса ТИ, принимаемые в течение одного такта работы (0/1 - переход из 0 в 1, 1 - уровень логической единицы, 0 - уровень логического нуля); → - операция пересылки; (РОНО) - содержимое регистра общего назначения.

В табл. 9 приведены наблюдаемые состояния шин после выполнения каждой микрокоманды.



Таблица 7

Адрес памяти	Номер тетрады							Операции
	7-6	5	4	3	2	I	0	
	AR-CA	M1.I8-6	M0.I2-0	CO.I5-3	A	B	D	
0		011	111	011		0000	XXXX	Загрузка $POH\ 0$
I		001	011	011		0000		Чтение $POH\ 0$
2		???	???	???				Загрузка PQ
3		???	???	???				Чтение PQ
4		???	???	???				Установка 0 в PQ
5		0 101	0 011	011		0000		Сдвиг $POH\ 0$ вправо
6		? ???	? ???	???		????		Сдвиг $POH\ 0$ влево
7		1 110	0 011	011		0000		Двойной сдвиг влево
8		? ???	? ???	???		????		Двойной сдвиг вправо
9		011	011	I 000		0000		$POH\ 0+1 \rightarrow POH\ 0$
10		???	???	? ???		????		$POH\ 0-I \rightarrow POH\ 0$
11		0 101	0 011	1 000		0000		Сложение со сдвигом вправо
12		? ???	? ???	? ???		????		Сложение со сдвигом влево

Таблица 8

Пуск по адресу	ТИ	Выполняемая операция
АО	0/1 1 0	$МК0 \rightarrow P_2МК$ (занесение МК0 в регистр МК) $F = D \rightarrow Y$ (на выходе МП значение D) $F = D \rightarrow POH\ 0$ (загрузка D в $POH\ 0$ )
A5	0/1 1 0	$МК5 \rightarrow P_2МК$ $F = (POH\ 0) \rightarrow Y$ $F/2 = D/2 \rightarrow POH\ 0$
A5	0/1 1 0	$МК5 \rightarrow P_2МК$ $F = (POH\ 0) \rightarrow Y$ $F/2 = D/4 \rightarrow POH\ 0$

Таблица 9

Пуск по адресу	Вых. Y K1804BCI	Флаги	Вых. $P_2$ данных	Вых. $P_2$ сост.
		C4 OVR F3 Z		
A0	0110	?? 00	XXXX	-
A5	0110	?? 00	0110	-
A5	0011	?? 00	0110	-
A5	0001	?? 00	0011	-

### **Задание для самостоятельной подготовки**

1. Ознакомьтесь с описанием и основными режимами работы устройства MT1804, способами контроля информации в MT1804. Изучите структуру микропроцессора K1804BC1 и выполняемые им функции.
2. Запрограммируйте неопределенные поля микрокоманд в табл.7 для выполнения всех указанных операций. Подготовьте в форме табл.8,9 описание заданной преподавателем последовательности операций.
3. Составьте микропрограммы для решения заданий п. 5 (см. порядок выполнения работы).

### **Порядок выполнения работы**

1. Изучить конструкцию устройства. Найти основные блоки узла управления и операционного узла.
2. Подключить к клеммам ХР1 источник питания напряжением +5 В и нагрузочной способностью не менее 2,5 А. Установить переключатель генератора тактовых импульсов в положение ВНУТРЕННИЙ, переключатели режима в положение ЗАГРУЗКА и ШАГ.

Внимание! При выключении источника питания содержимое микропрограммной памяти не сохраняется.

3. Загрузить в память микропрограмму из табл. 7. Выполнить заданную последовательность МК, адресуя их с переключателей адреса в режиме ЗАГРУЗКА. После выполнения каждой МК наблюдать на индикаторах состояния выходов Y и флаги процессора, выходов регистров данных и состояния. Результаты записать в виде табл. 9, сравнивая их с ожидаемыми. Объяснить полученные результаты.

4. Проверить при многократном исполнении действие остальных микрокоманд из табл. 7, предварительно очистив регистр Q.

5. Разработать и выполнить микропрограммы следующих операций:

- а) очистка регистра  $POH_i$ ;
- б) обмен данными регистров  $POH_i$  и  $POH_j(PQ)$ ;
- в) алгебраическое сложение/вычитание в дополнительном коде. Изменяя операнды, получить наибольшее количество возможных комбинаций флагов Z, C4, OVR, F3. Запротоколировать результаты наблюдений;
- г) изменение знака числа;
- д) алгебраическое сложение/вычитание чисел в обратном коде;

Содержание отчета

В отчет должны входить: структурная схема и таблицы функций управления процессорного элемента; описание выполнения заданной последовательности микрокоманд; микропрограммы и результаты наблюдений в виде таблиц.

### **Содержание отчета**

В отчет должны входить:

- Структурная схема и таблицы функций управления процессорного элемента;
- Описание выполнения заданной последовательности микрокоманд;
- Микропрограммы и результаты наблюдений в виде таблиц;
- Скриншоты экрана с последовательностью выполнения микропрограмм.
- Пошаговые комментарии к микропрограммам

### **Работа выполняется на эмуляторе – mtemu\_1.0.3 (1.0.4)**

**Справочная информация на странице -** [https://ru.bmstu.wiki/Микропренажёр\\_MT1804](https://ru.bmstu.wiki/Микропренажёр_MT1804)

### Микротренажер МТ1804

Предназначен для изучения Архитектуры секционных Процессоров, Алгоритмов микропрограмм и получения навыков разработки системы команд Процессора на базе Секционного Процессорного Комплекта К1804

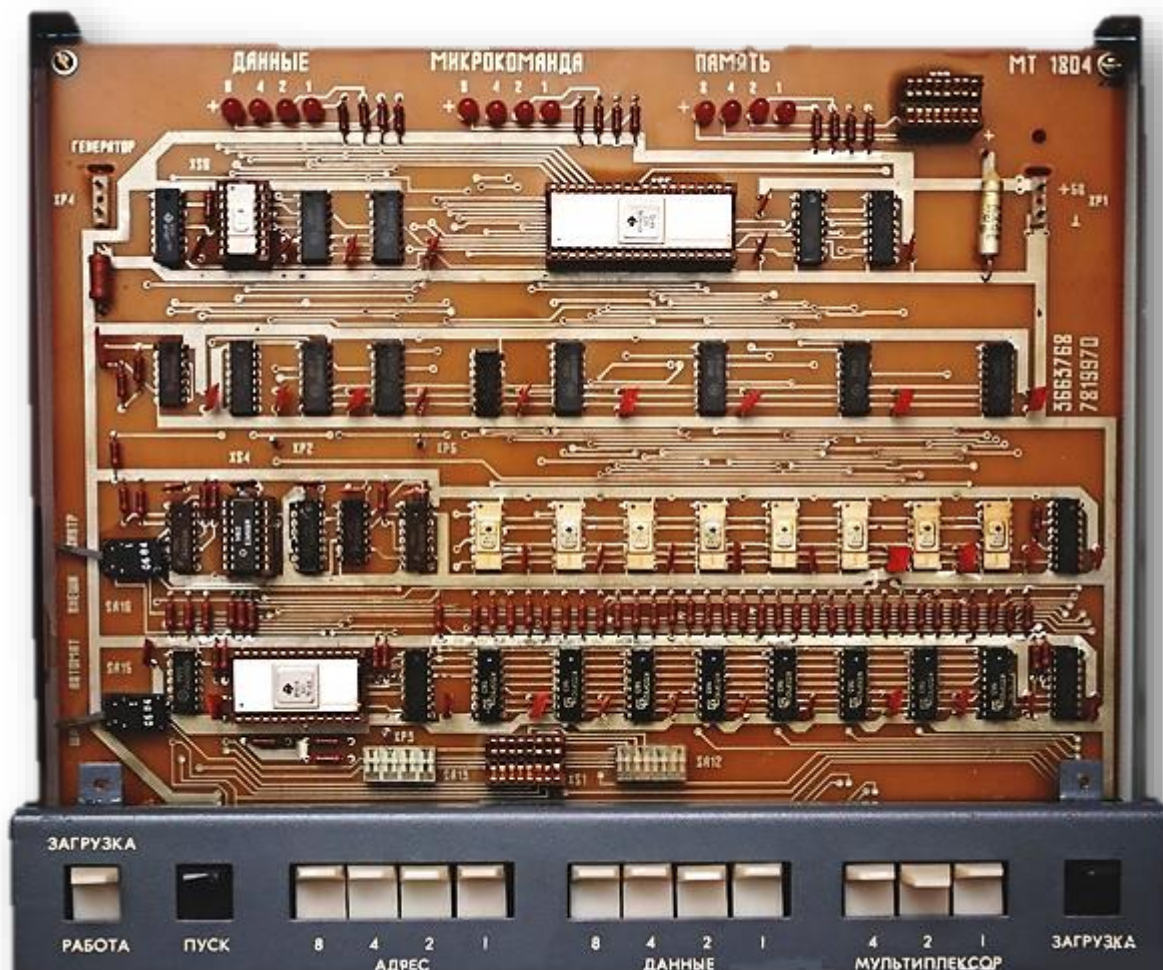


Фото 1 Микротренажер МТ1804

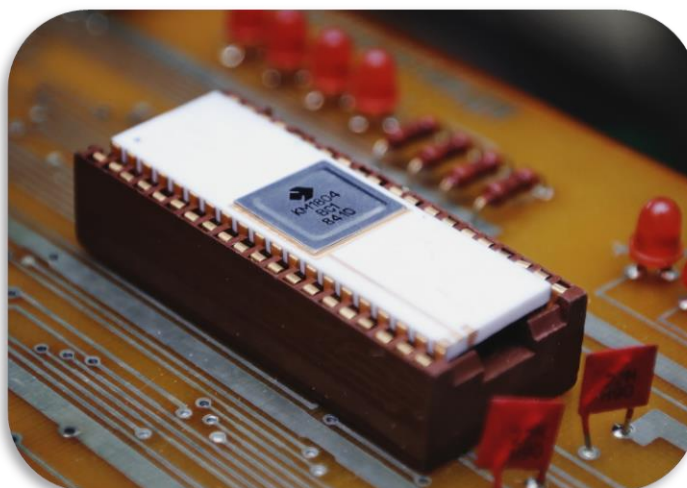


Фото 2 KM1804BS1

## Работа № 2. БЛОК МИКРОПРОГРАММНОГО УПРАВЛЕНИЯ (БМУ). ПЕРЕХОДЫ В МИКРОПРОГРАММАХ С ИСПОЛЬЗОВАНИЕМ СТЕКА

**Цель работы** - изучение структуры и функций БМУ K1804ВУ1, способа управления узлами БМУ с помощью микрокоманды; исследование функций перехода с использованием стека.

### Теоретическая часть

Управление выборкой следующей микрокоманды из микропрограммной памяти осуществляется с помощью блока микропрограммного управления, который на схеме (см. рис. 1) представлен узлами:

БИС управления адресом микрокоманды K1804ВУ1,  
Микропрограммная память,  
Регистр микрокоманды,  
ПЗУ для управления выборкой следующего адреса.

В составе микрокоманды, выбираемой из памяти в регистр МК, имеются два поля AR (тетрада 7) и CA (тетрада 6), используемые для управления БИС K1804ВУ1. 4-разрядный код функции перехода CA и сигнал с выхода мультиплексора флагов состояния поступают на адресные входы ПЗУ. На выходах ПЗУ вырабатываются сигналы управления БИС K1804ВУ1, необходимые для реализации заданной функции перехода. Код адреса перехода AR из памяти поступает непосредственно на R -входы БИС K1804ВУ1. Чтобы уяснить значения сигналов на выходах ПЗУ, необходимых для управления БИС K1804ВУ1, обратимся к структурной схеме БИС (рис. 5), Схема содержит:

регистр адреса (PrA), на входы R которого поступает адрес с выходов памяти;  
счетчик микрокоманд (СМК), состоящий из схемы приращения - инкрементатора - и регистра (Pr СМК);  
стек, состоящий из 4 регистров накопителя, указателя стека и схемы записи/считывания;  
мультиплексор, с помощью которого выбирается источник адреса следующей микрокоманды.

Источником адреса могут быть

регистр адреса,  
счетчик микрокоманд,  
стек

D -шина адреса, подключенная в устройстве МТ1804 к переключателям адреса (см. рис.1).

Выбор источника адреса осуществляется в зависимости от значений управляющих сигналов на входах мультиплексора SO, S1 (табл. 10). Выходы мультиплексора соединены с четырьмя двухвходовыми элементами ИЛИ, позволяющими модифицировать выбранный адрес с помощью маски, подаваемой на входы OR. При наличии 1 в i-м разряде маски адрес может быть изменен на  $2^i$  (в устройстве на входы OR из ПЗУ поступает 0). С выходов ИЛИ код адреса поступает на элементы И, управляемые сигналом  $\overline{Z}A$ . Адрес выводится на выходы Y через трехстабильные буферы, управляемые сигналом  $\overline{OE}$  (в устройстве МТ1804  $\overline{Z}A=1$ ,  $\overline{OE}=0$ ), что обеспечивает постоянную выдачу адреса на шину Y). Инкрементор счетчика микрокоманд увеличивает текущий адрес Y на 1, если сигнал CO = 1 (в МТ1804 на входе CO постоянно действует сигнал 1). Запись нового адреса в PrСМК происходит по положительному фронту тактового импульса на входе T. По положительному фронту T-импульса и сигналу разрешения  $\overline{RE}=0$  происходит запись адреса в PrA (в МТ1804 на входе  $\overline{RE}=0$  постоянно действует 0).



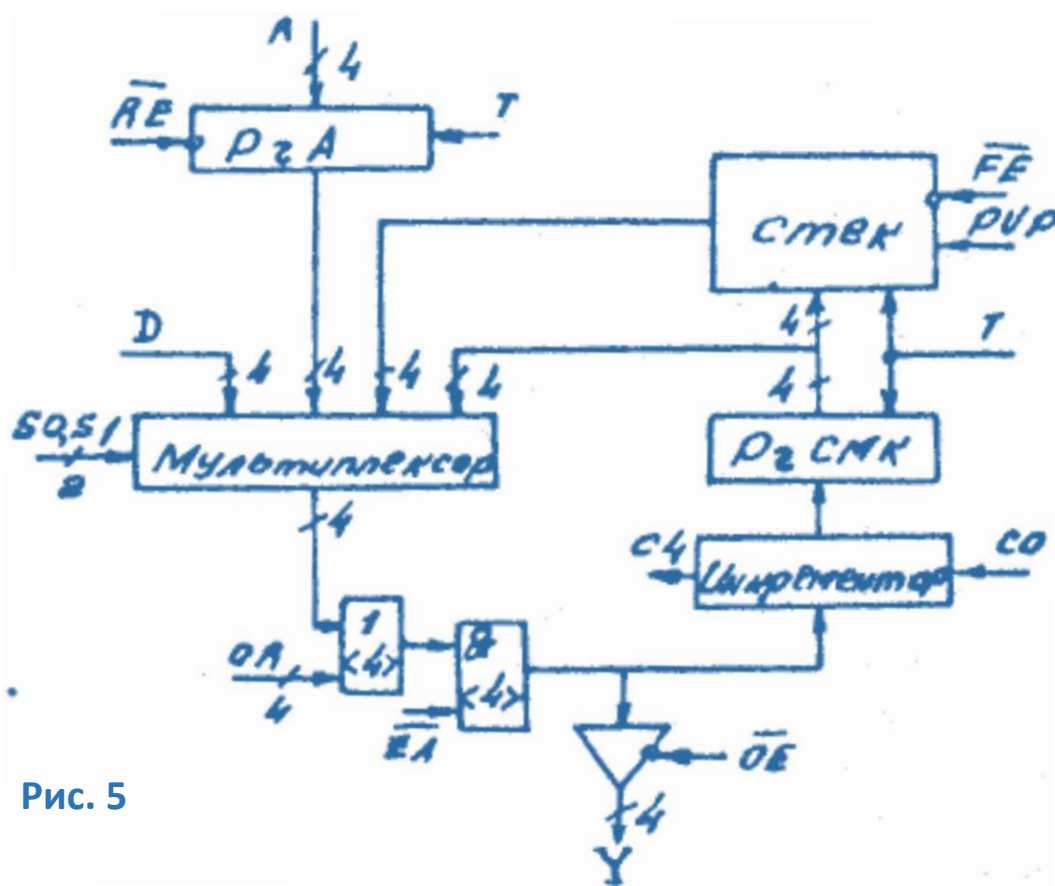


Рис. 5

Таблица 10

SI	SO	Y
0	0	CMK
0	1	PzA
1	0	Стек
1	1	D

Таблица 11

$\overline{FE}$	PUP	Операция
1	X	Стек отключен
0	1	PUSH: содержимое CMK загружается в стек
0	0	POP: циклический сдвиг содержимого стека

Стек БИС KI804BU1 используется для временного хранения адресов со счетчика микрокоманд. Адрес, записанный в стек последним, извлекается из него первым. Операции со стеком определяются сигналами управления, действующими на входах EF, PUP (табл. 11). Обращение к ячейкам накопителя стека осуществляется через указатель стека (УС), выполненный на основе 2-разрядного реверсивного счетчика. При выполнении операции PUSH (загрузка в стек) по фронту T-импульса происходит увеличение на 1 содержимого УС и запись содержимого CMK в стек при T = 0. При выполнении операции POP (выборка из стека) источником адреса, поступающим на выход стека, является верхняя ячейка стека, определяемая содержимым указателя стека. После выборки содержимое УС уменьшается на 1, что приводит к изменению статуса верхней ячейки стека. Внешне это воспринимается как циклический сдвиг содержимого стека. Состояния указателя стека до и после выполнения операций PUSH и POP.

УС до/после операции <b>PUSH</b>	0/1	1/2	2/3	3/0
УС до/после операции <b>POP</b>	0/3	1/0	2/1	3/2

Здесь 0, 1, 2, 3 - номера ячеек накопителя стека. При сигнале  $\overline{FE}=1$  изменения содержимого УС не происходит, считывание информации осуществляется из верхней ячейки стека.

Основное назначение стека - сохранение адреса основной программы при переходе к подпрограмме. Автоматическое запоминание и восстановление адреса основной программы происходит благодаря программированию в ПЗУ операций PUSH и POP соответственно при переходе и возврате из подпрограммы, а также программированию сигналов на управляющих входах S0, S1. В табл. 12 приведены коды, программируемые в ПЗУ в зависимости от функций перехода. Кроме кодов управления, подаваемых на входы БИС K1804ВУ1, на выходе ПЗУ вырабатывается бит управления регистром состояния (флагов), запрещающий запись флагов в регистр при значении 1.

Автоматическое сохранение и восстановление адреса при обращении и выхода из подпрограммы позволяют сделать подпрограммы вложенными, т.е. осуществить вызов одной подпрограммы из другой. Уровень вложенности микропрограмм определяется размером стека (для БИС K1804ВУ1 не превышает 4). Функция "условный переход к подпрограмме" (в табл. 12 код CA = 0100) позволяет перейти к подпрограмме по значению разряда Z регистра состояния (по F=0).

В устройстве МТ1804 стек может быть использован для организации циклического выполнения участка программы. Для этого перед началом цикла в один из внутренних регистров процессора загружается число повторений k. После выполнения циклической части программы это число уменьшается на единицу и затем осуществляется проверка конца цикла с помощью микрокоманды "окончить цикл и вытолкнуть из стека, если F=0".

Таблица 12

Функция перехода	Входы ПЗУ		Выходной код ПЗУ					Операции в БМУ
	Код CA	Флаг	SI	.SO	$\overline{FE}$	PUP	P <sub>2</sub> ST	
Переход на следующий адрес (продолжить)	0010	X	0	0	1	X	0	CMK→Y
Безусловный переход на адрес	0001	X	0	1	1	X	0	P <sub>2</sub> A→Y
Переход на адрес из PгМК, если F=0*)	1100	0	0	0	1	X	1	CMK→Y
	1100	1	0	1	1	X	1	PгA→Y
Переход на адрес из PгМК, если F≠0	0000	0	0	1	1	X	1	PгA→Y
	0000	1	0	0	1	X	1	CMK→Y
Загрузить в стек (и продолжить)	1001	X	0	0	0	1	0	CMK→Y, PUSH
Вытолкнуть в стек (и продолжить)	1010	X	0	0	0	0	0	CMK→Y, POP
Переход по стеку	0111	X	1	0	1	X	0	СТЕК→Y
Окончить цикл и вытолкнуть из стека, если F=0 **)	1000 1000	0	1	0	1	X	1	СТЕК→Y
		1	0	0	0	0	1	CMK→Y, POP
Переход по адресу вектора (на переключателях адреса)	0011	X	1	1	1	X	0	D→Y
Переход к подпрограмме	0101	X	0	1	0	1	0	P <sub>2</sub> A→Y, PUSH
Переход к подпрограмме, если F≠0	0100	0	0	1	0	1	1	P <sub>2</sub> A→Y, PUSH
	0100	1	0	0	1	X	1	CMK→Y
Возврат из подпрограммы	0110	X	1	0	0	0	0	СТЕК→Y, POP

\*) Аналогичные операции перехода по адресу из Pг.МК, если F3=1 (CA=1101), OVR=1 (CA=1110), C4=1 (CA=1111)

\*\*) Аналогичная операция по условию C4=1 (CA = 1011).

Если F≠0, через стек происходит возврат к началу циклического участка программы, в противном случае выбирается следующий адрес из счетчика микрокоманд.

Наряду с описанными функциями перехода в табл. 12 приведены коды функций перехода по условиям и др. В табл. 13 приведены две программы переходов с использованием стека: программа 1 реализует цикл с бесконечным числом повторений, программа 2 - работу с обращением к подпрограмме, размещенной в ячейках 12, 13. Рассмотрим подробнее работу БМУ при выполнении программы 2. В табл. 14 приведена диаграмма состояний регистров БМУ и сигналов управления на входах и выходах БИС K1804ВУ1. В такте Т0 выполняется микрокоманда МК0 с адресом 0: S0 = S1 = 0, поэтому на выход Y БИС из счетчика микрокоманд поступает адрес 1 и из памяти извлекается микрокоманда "переход к подпрограмме" (JSR12), адресная часть которой AR = 12 поступает на входы R.. В такте Т1 эта команда принимается в РгМК, а адрес 12 - в РгА. На входах БИС устанавливаются сигналы для выполнения перехода и сохранения адреса возврата в стеке. Адрес 12 передается из РгА на выход Y и по нему выбирается первая микрокоманда подпрограммы МК12. В такте Т2 МК12 загружается в РгМК, адрес возврата 2 - в стек, а адрес (12+1) = 13 - в СМК. Из СМК под действием сигналов S0=S1=0 адрес 13 поступает по шине Y на вход памяти, из которой считывается микрокоманда "возврат из подпрограммы" (RTS). В такте Т3 на входах БИС устанавливаются сигналы для выполнения возврата из подпрограммы. Из стека на выход Y поступает адрес возврата 2, и из памяти выбирается; МК2 основной программы. В такте Т4 микрокоманда МК2 заносится в РгМК, происходит циклический сдвиг содержимого стека, в результате чего адрес 2 выталкивается, а адрес 3 записывается в СМК. Содержимое СМК поступает на выход Y, и из памяти выбирается микрокоманда безусловного перехода по адресу 0 (JP0). Адрес 0 поступает на R -входы. В такте Т5 "JP0" записывается в РгМК, адрес 0-в РгА и под действием сигналов S0 = 1, S1 = 0 содержимое РгА передается на выход Y для выборки микрокоманды МК0. Далее цикл работы БМУ повторяется.

Таблица 13

Адрес памяти	Тетрада 7 (AR)	Тетрада 6 (CA)	Выход Y	Примечания
0		0010	0001	<u>Программа 1</u>
1		1001	?	Продолжить
2		0010	?	Загрузить стек
3		0010	?	Продолжить
4		0111	?	"
				Переход по стеку
0		0010	?	<u>Программа 2</u>
1	1100	0101	?	Продолжить
2		0010	?	Переход к подпрограмме I2
3	0000	0001	?	Продолжить
⋮				Переход на 0
⋮				
I2		0010	?	Продолжить
I3		0110	?	Возврат



Таблица I4

Выполняемый такт	T0	T1	T2	T3	T4	T5
Входы К1804ВУ1:						
$\overline{SO}, SI$	00	I0	00	01	00	I0
$\overline{FF}, PUP$	IX*)	01	IX	00	IX	IX
R	I2	X	X	X	0	X
Содержимое регистров:						
СМК	I	2	I3	I4	3	4
Стек 0	-	-	2	2	-	-
Стек I,2,3	-	-	-	-	-	-
РГА	X	I2	X	X	X	0
Выход К1804ВУ1	I	I2	I3	2	3	0
Выход памяти	XSR12	MR12	RTS	MR2	XPO	MR0
Содержимое РГМК	MR0	XSR12	MR12	RTS	MR2	XPO

\*) X — состояние безразлично.

#### Задание для самостоятельной подготовки

1. Изучите структуру БМУ, назначение его узлов и сигналов управления.
2. По табл. 12 изучите функции переходов в БМУ МТ1804.
3. Разработайте программу с обращением к подпрограмме из подпрограммы. Постройте диаграмму состояний БМУ в форме табл. 14, приняв уровень вложенности подпрограмм, равным 2.
4. Изучите программу, представленную в символической записи функций перехода на рис. 6, где числами обозначены адреса ячеек. Составьте диаграмму состояний стека при работе программы. Подготовьте программу для выполнения в МТ1804.

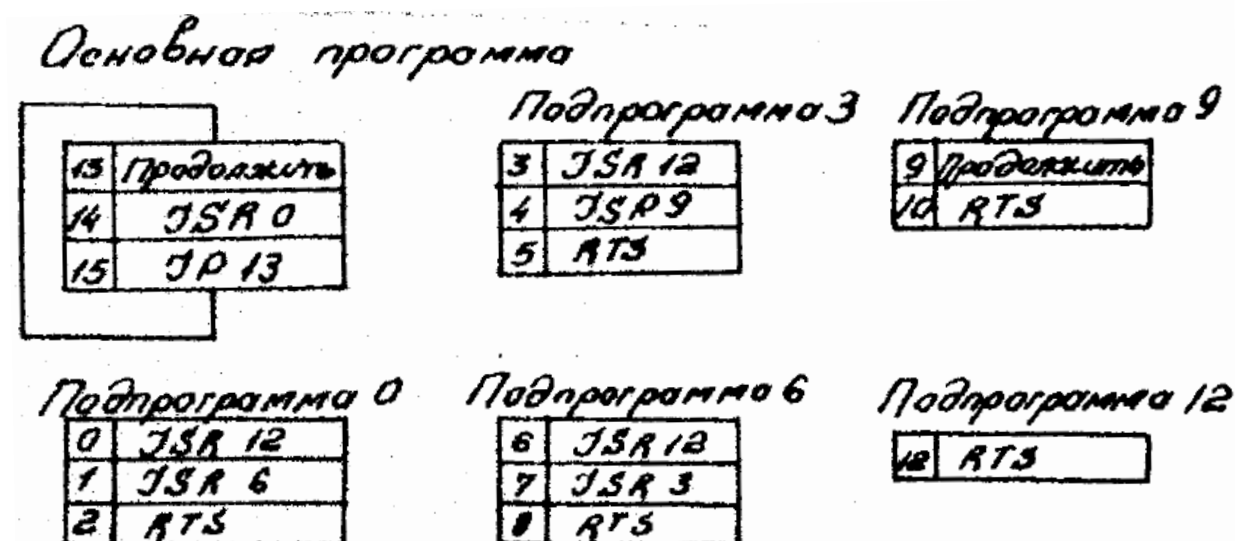


Рис.6

## Порядок выполнения работы

1. Загрузить в память (в тетрады 6, 7) программу, обеспечивающую выполнение 3 групп микрокоманд по адресам  $A_0, A_1$ ,  $(A_4, A_5)$ ,  $(A_{14}, A_{15})$  с остановом по адресу  $A_{15}$ . Выполнить начальную установку регистра МК. Перевести переключатель режима в положение **РАБОТА**. Установить переключатели **МУЛЬТИПЛЕКСОР** в положение 0 для индикации выходной шины адреса  $Y$  БИС К1804ВУ1. Нажимая кнопку **ПУСК**, проверить последовательность адресов, выводимых на шину  $Y$ .
2. Проверить работу программы 1 из табл. 13, фиксируя последовательность адресов на шине  $Y$ . Изменить программу, обеспечив выход из цикла по условию.
3. Загрузить и выполнить программу условного перехода по адресу  $A_j$ , в которой проверяются 2 заданных признака, вырабатываемых в результате выполнения арифметической операции (из совокупности признаков  $F3, C4, OVR, F=0, F\neq 0$ ). Обратить внимание на значение второго признака на выходе процессорного элемента после выполнения команды условного перехода по первому признаку. Объяснить, каким значением второго признака (первоначальным или изменившимся) обусловлен второй переход, и почему.
4. Выполнить программы, разработанные по пп. 3,4 задания для самостоятельной подготовки, сравнивая наблюдаемые результаты с ожидаемыми.

## Содержание отчета

*В отчет должны входить:*

- Структурная схема БМУ;
- Таблица функций перехода;
- Диаграмма состояний БМУ при выполнении вложенных подпрограмм
- Микропрограммы и результаты наблюдений в виде таблиц;
- Описание выполнения заданной последовательности микрокоманд;
- Скриншоты экрана с последовательностью выполнения микропрограмм.
- Пошаговые комментарии к микропрограммам

## Работа № 3. МИКРОПРОГРАММИРОВАНИЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

**Цель работы** - изучение способов представления чисел в микро-ЭВМ и алгоритмов арифметических операций; микропрограммирование операций в системе микрофункций процессора K1804.

### Теоретическая часть

#### 1. Представление чисел в микро-ЭВМ.

Отрицательные числа обычно представляются в виде дополнений до основания системы счисления. При операциях над числами в микро-ЭВМ обычно полагают, что числа имеют следующий вид:

$$D = d_{n-1} d_{n-2} \dots d_1 d_0$$

то есть точка находится справа и числа являются целыми.

В общем случае дополнение любого  $n$  – разрядного числа  $D$  до основания  $b$  системы счисления можно получить путем вычитания  $D$  из  $b^n$ . Если  $D$  находится в пределах от 1 до  $b^n - 1$ , то при вычитании получается другое число в тех же пределах.

Если  $D=0$ , то результат вычитания равен  $b^n$  и имеет вид  $100\dots 0$  при общем числе разрядов, равном  $(n+1)$ . Отбросив цифру старшего разряда, получим 0. Следовательно, в системе представления чисел дополнением до основания системы счисления существует только одно представление 0.

В десятичной системе счисления дополнение до основания есть дополнение до десяти, которое можно получить путем вычитания  $n$  – разрядного числа из  $10^n$ .

*Пример.*

Десятичное число  $A = 1849$ . Дополнение до десяти  $[A]_{\text{доп}} = 10000 - A = 8151$ .

Для двоичных чисел дополнение до основания системы счисления называется дополнением до двух. В системе представления дополнением до двух, или в дополнительном коде, число является положительным, если значение старшего разряда  $d_{n-1} = 0$ , и отрицательным, если  $d_{n-1} = 1$ . Десятичный эквивалент двоичного числа, представленного дополнением до двух, вычисляется так же, как и для числа без знака, за исключением того, что вес старшего разряда равен  $-2^{(n-1)}$ , а не  $+2^{(n-1)}$ . Представляемые числа находятся в диапазоне от  $-2^{(n-1)}$  до  $+2^{(n-1)} - 1$ .

В системе представления чисел неполным дополнением до основания дополнение  $n$  – разрядного числа  $D$  получается путем его вычитания из  $b^n - 1$ . Для двоичных чисел неполное дополнение называется дополнением до единицы или обратным кодом. При вычислении десятичного эквивалента числа, записанного как дополнение до единицы, старшему разряду приписывается вес  $-(2^{(n-1)} - 1)$ , а не  $-2^{(n-1)}$ .

Представляемые числа находятся в диапазоне от  $-(2^{(n-1)} - 1)$  до  $+(2^{(n-1)} - 1)$ . Ноль имеет два представления - положительный ноль (00..00) и отрицательный ноль (11..11). Представления положительных чисел в системах с дополнением до единицы и до двух совпадают, тогда как представления отрицательных чисел отличаются на 1.

#### 2. Сложение и вычитание чисел в дополнительном коде.

Графическое представление 4–разрядных двоичных чисел в дополнительном коде приведено на рис.1. Сложение с положительными числами легко интерпретировать, перемещая указатель по часовой стрелке на  $+n$  позиций; вычитание  $(-n)$ , перемещая указатель против часовой стрелки, или перемещая по часовой стрелке на  $(16-n)$  позиций, что равносильно замене вычитания сложением с дополнением числа до двух. Если при сложении получают результат, который выходит за пределы диапазона представляемых чисел, то имеет место переполнение.

Правило выявления переполнения. При сложении переполнение происходит только в том случае, если слагаемые имеют одинаковые знаки, а знак суммы отличается от знака слагаемых. Правило переполнения можно сформулировать иначе, используя понятие переносов, возникающих при сложении. Переполнение возникает, если значения переносов в знаковый разряд и из

знакового разряда различны. Из анализа рис.1 следует, что переполнение возникает при сложении в случае, если указатель перейдет границу между позициями +7 и –8.

Числа в дополнительном коде складываются и вычитаются так же, как числа без знака той же длины. Поэтому для выполнения операций над числами обоих типов необходим всего один тип команды сложения или вычитания. Различие заключается лишь в том, что результаты операций интерпретируются по-разному в зависимости от того, какими числами оперирует ЭВМ: числами со знаком (то есть от –8 до +7) или без знака (от 0 до 15).

На рис.2 приведено графическое представление 4–разрядных двоичных чисел без знака. Из него видно, что двоичные кодовые комбинации занимают те же позиции, что и на рис.1, а сложение и вычитание можно осуществить, перемещая указатель на  $n$  позиций в том или ином направлении. При сложении чисел без знака результат выходит за пределы диапазона представления при переходе границы между 15 и 0. В этом случае говорят о возникновении переноса из старшего разряда. При вычитании чисел без знака результат выходит за пределы диапазона при переходе границы между 0 и 15. В этом случае возникает заем. Но так как вычитание  $n$  можно заменить сложением с дополнительным кодом числа  $n$ , равным  $(16 - n)$ , то заем возникает при отсутствии переноса.

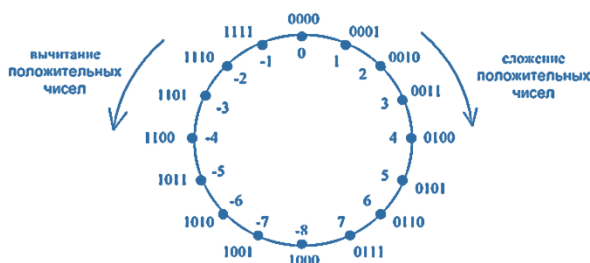


Рис.1

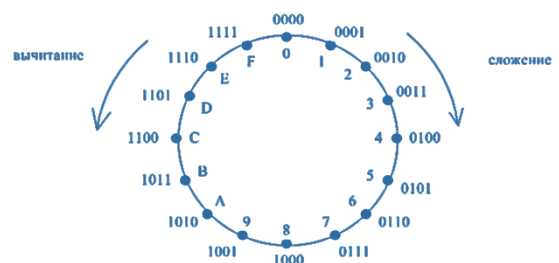


Рис.2

### 3. Двоично – десятичное сложение – вычитание.

При сложении двух двоично – десятичных чисел  $A = a_{n-1} a_{n-2} \dots a_1 a_0$  и  $B = b_{n-1} b_{n-2} \dots b_1 b_0$  поступают следующим образом. Если оба операнда имеют одинаковые знаки, то выполняют сложение модулей этих чисел ( $|A| + |B|$ ), а знаковый разряд сумм определяют по знаку одного из слагаемых. Если операнды имеют разные знаки, то предварительно знак суммы устанавливают по знаку первого операнда  $A$ . Затем производят вычитание модулей чисел ( $|A| - |B|$ ). Если полученная разность больше 0, знак суммы сохраняется без изменений. Если разность меньше 0, следует найти дополнительный код разности и изменить знак суммы на противоположный.

**а.** Операция сложения модулей ( $|A| + |B|$ ) выполняется по алгоритму, схема которого приведена на рис. 3:

1) двоично – десятичный код первого операнда  $a_{n-1} a_{n-2} \dots a_1 a_0$  складывается с кодом 66...66, образуя первую промежуточную сумму  $S'_{n-1} S'_{n-2} \dots S'_1 S'_0$ ;

2) к полученной сумме прибавляется двоично – десятичный код 2-го операнда  $b_{n-1} b_{n-2} \dots b_1 b_0$ , образуя вторую промежуточную сумму  $S''_{n-1} S''_{n-2} \dots S''_1 S''_0$ ;

3) выполняется потетрадно коррекция результата. Правило коррекции формулируется следующим образом: если в результате второго сложения перенос из  $i$  – ой тетрады отсутствует ( $c_{i+1} = 0$ ), то к  $S''_i$  прибавляется код 1010<sub>2</sub> или A16, что соответствует вычитанию 6. При возникновении переноса из  $i$  – ой тетрады коррекция не выполняется (или прибавляется код 0000), а полученный результат  $S''_i$  является истинным. При выполнении коррекции потетрадные переносы в двоичном сумматоре блокируются.

**б.** Вычитание модулей ( $|A| - |B|$ ) выполняют по алгоритму, схема которого представлена на рис. 4.

1) операнд  $B$  представляют двоично - десятичным дополнением до десяти;

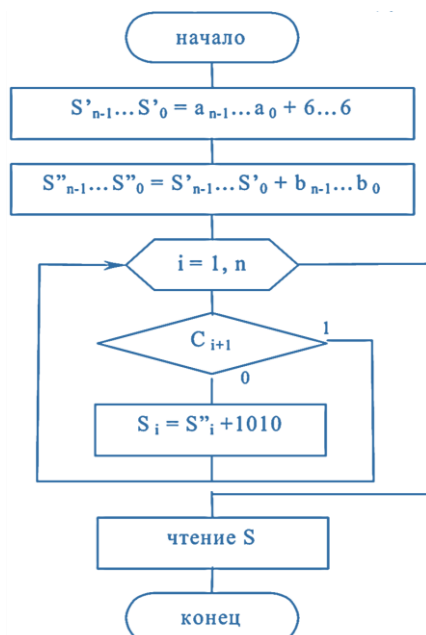
2) двоично – десятичный код  $A$  складывают с дополнительным кодом  $B$ . Если в результате сложения образуется перенос из старшей тетрады ( $c_n = 1$ ), результат является положительным. При

отсутствии переноса результат является отрицательным и его следует перевести в дополнительный код.

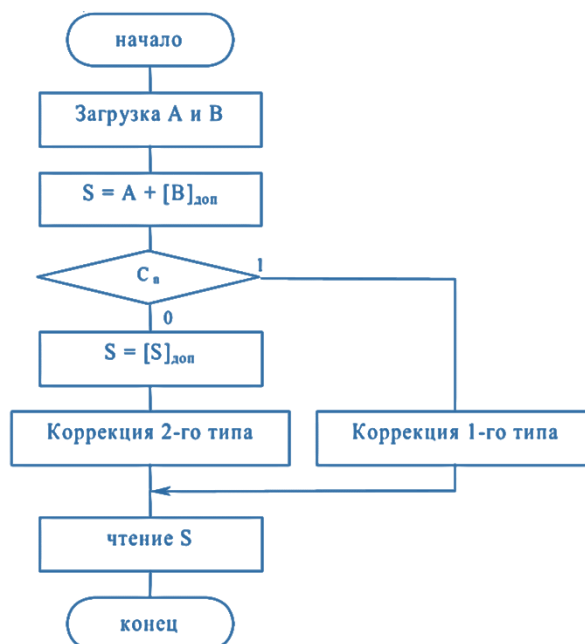
3) коррекция положительного результата осуществляется по правилу, сформулированному для сложения модулей чисел.

Коррекция отрицательного результата выполняется иначе:

Если имел место перенос из  $i$  – ой тетрады при сложении  $A + [B]_{\text{доп}}$ , то к  $i$  – ой тетраде прибавляется код 1010; если перенос отсутствует – прибавляется 0000 (см. приложение).



а. Рис.3



б. Рис.4

Пример.

Найти разность  $A - B$ .

$A = 237, B = 623$ .

Двоично – десятичное представление чисел A и B:

$A = 0010\ 0011\ 0111$        $B = 0110\ 0010\ 0011$

Дополнение числа B:  $[B]_{\text{доп.}} = 1001\ 1101\ 1101$

Складываем числа  $A + [B]_{\text{доп.}}$ , получаем:

```

0010 0011 0111
+ 1001 1101 1101
1100 0001 0100
  
```

Дополнение суммы складываем с кодами коррекции:

```

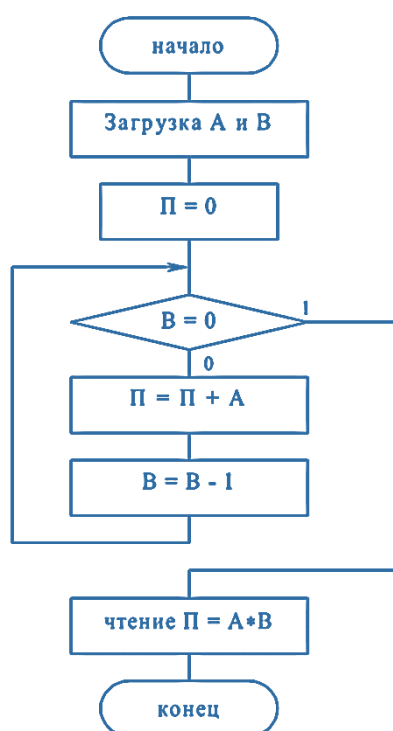
0011 1110 1100
+ 0000 1010 1010
0011 1000 0110 = 386
  
```

#### 4. Умножение чисел без знака.

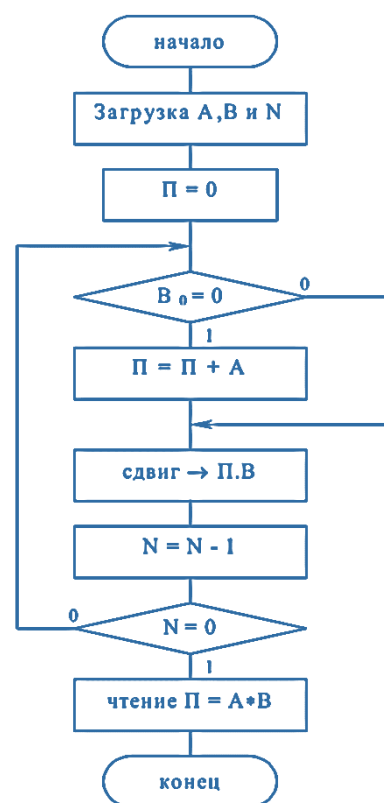
- а. Наиболее просто умножение можно выполнить по итерационной схеме алгоритма, изображенной на рис. 5. После загрузки множимого A и множителя B в регистры общего назначения и обнуления регистра произведения П производится анализ содержимого регистра множителя. Если  $B \neq 0$ , то к сумме частичных произведений П прибавляется множимое A. Затем содержимое регистра множителя уменьшается на 1 и цикл умножения повторяется до тех пор, пока содержимое регистра множителя не окажется равным 0. При умножении  $n$  – разрядных сомножителей  $2n$  – разрядное произведение размещают в двух регистрах. Данный метод

умножения находит ограниченное применение в сравнительно несложных микропроцессорных системах.

- b. На практике большое распространение имеют методы умножения путем сложения ряда сдвинутых относительно друг друга множимых, с учетом цифр множителя. Один из алгоритмов умножения, начиная с младших разрядов множителя, приведен на рис. 6. Этот алгоритм может быть использован для получения произведения двух двоичных чисел без знака. Количество итераций умножения  $N$  определяется числом разрядов множителя. Поскольку в процессе умножения на каждой итерации осуществляется сдвиг множителя  $B$  на 1 разряд вправо, на место освобождаемых разрядов можно записать выталкиваемые при сдвиге вправо разряды произведения  $P$ . При использовании  $n$  – разрядного сумматора или АЛУ исходные двоичные числа без знака не должны выходить за пределы диапазона от 1 до  $2^{(n-1)} - 1$ .



a. Рис.5



b. Рис.6

## 5. Деление чисел без знака.

Для типичного алгоритма деления делимым является двойное слово, а делителем – одинарное; частное и остаток получаются в виде одинарных слов. Если при выполнении такого деления окажется, что делитель равен 0, или для представления частного потребуется более одного слова, то происходит переполнение. Последнее имеет место в том случае, если делитель больше или равен старшего слова делимого.

В качестве примера рассмотрим метод деления  $A/B$  без восстановления остатка. В этом случае алгоритм деления представляет итерационную процедуру, на каждой итерации которой производится либо вычитание делителя  $B$ , представленного в дополнительном коде, либо прибавление  $B$ , в зависимости от знака остатка, полученного на предыдущей итерации деления. Если полученный остаток был больше 0, при очередной итерации деления производится вычитание  $B$ ; если остаток был меньше 0, производится прибавление  $B$ . Перед каждым вычитанием (или сложением) производят удвоение остатка путем сдвига влево. На начальной итерации деления делимое сдвигается на 1 разряд влево.

Пример.

Деление 8 – разрядного числа **A** на 4–разрядное число **B** ( $C = c_4c_3c_2c_1$  – частное, **P** – перенос).

**A** = 19 = 0001 0011<sub>2</sub>

**B** = 4 = 0100<sub>2</sub> [**B**]<sub>доп</sub> = 1100<sub>2</sub>

'P'	0001 0011	; делимое <b>A</b>
↓	+ 0010 0110	; сдвиг <b>A</b> ← влево
↓	+ 1100	; вычитание <b>B</b>
0	1110	; <b>s</b> <sub>1</sub> – 1-ый остаток
└───────────>	0	; <b>c</b> <sub>4</sub> = 0 Определение разряда частного
↓	+ 1100 1100	; <b>s</b> <sub>1</sub> * <b>c</b>
↓	+ 0100	; прибавление <b>B</b>
1	0000	; <b>s</b> <sub>2</sub> – 2-ой остаток
└───────────>	1	; <b>c</b> <sub>3</sub> = 1
↓	+ 0001 1001	; <b>s</b> <sub>2</sub> * <b>c</b>
↓	+ 1100	; вычитание <b>B</b>
0	1101	; <b>s</b> <sub>3</sub> – 3-й остаток
└───────────>	0	; <b>c</b> <sub>2</sub> = 0
↓	+ 1011 0010	; <b>s</b> <sub>3</sub> * <b>c</b>
↓	+ 0100	; прибавление <b>B</b>
0	1111	; <b>s</b> <sub>4</sub> – 4-й остаток
└───────────>	0	; <b>c</b> <sub>1</sub> = 0
	↓	
	0100	; частное
	+ 0100	; прибавление <b>B</b> для получения остатка
	0011	; остаток

### Задание для самостоятельной подготовки

Составить схемы алгоритмов и подготовить микропрограммы по всем пунктам работы. Написать оригинальные примеры для проверки работы микропрограмм.

### Порядок выполнения работы

1. Выполнить операции сложения и вычитания двух 4 – разрядных чисел со знаком и без знака. Привести **примеры** образования признаков **переноса, заема, переполнения и нуля**.
2. Разработать и выполнить микропрограммы сложения и вычитания 8 – разрядных чисел без знака.
3. Разработать и выполнить микропрограммы сложения и вычитания 8 – разрядных чисел со знаком. Отрицательные числа должны быть представлены в дополнительном коде.
4. Разработать и выполнить микропрограмму сложения модулей 2–разрядных двоично – десятичных чисел.
5. Разработать и выполнить в автоматическом режиме микропрограмму умножения 4-разрядных сомножителей без знака по схеме алгоритма на рис. 6.
6. Разработать и выполнить микропрограмму деления чисел без знака, полагая известным, что делитель всегда больше 0 и переполнение невозможно для заданных операндов.
7. Составить отчет.

### Содержание отчета

В отчет должны входить:

- Блок-схемы алгоритмов;
- Примеры арифметических операций;
- Микропрограммы и результаты наблюдений в виде таблиц;
- Описание выполнения заданной последовательности микрокоманд;
- Скриншоты экрана с последовательностью выполнения микропрограмм.
- Пошаговые комментарии к микропрограммам