

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ»**

**Институт управления и информационных технологий**

**Кафедра «Вычислительные системы и сети»**

**А. Е. МАМЧЕНКО, М. И. ШАМРОВ**

**ОРГАНИЗАЦИЯ, СХЕМОТЕХНИКА  
И МИКРОПРОГРАММИРОВАНИЕ  
ПРОЦЕССОРОВ ЭВМ**

Рекомендовано редакционно-издательским советом университета  
в качестве методических указаний для студентов, обучающихся  
по направлению «Информатика и вычислительная техника»

**МОСКВА - 2012**

**УДК 681.3  
М-22**

Мамченко А. Е., Шамров М. И. Организация, схемотехника и микропрограммирование процессоров ЭВМ. Методические указания к курсовому проектированию по дисциплине «Организация ЭВМ и систем». -М.: МИИТ, 2012, - 66с.

В методических указаниях приведены основные сведения о проектируемых процессорах, задания на их проектирование, описание элементной базы, общие рекомендации по выполнению и требования по оформлению курсовых проектов

© МИИТ, 2012

## **Введение**

Курсовой проект по дисциплине «Организация ЭВМ и систем» является важным и одним из сложных учебных заданий, выполняемых студентами в рамках профессиональной подготовки по направлению «Информатика и вычислительная техника». В нем разрабатывается программируемое управляемое устройство на базе больших интегральных схем (БИС) многокристальных микропроцессоров, решаются нетривиальные и трудно формализуемые вопросы реального проектирования устройств вычислительной техники:

- выбор архитектуры,
- разработка общей структурной организации,
- разработка аппаратных и микропрограммных средств.

Оформление конструкторских документов проекта требует освоения системы автоматизированного проектирования цифровых устройств.

Указания содержат задания и рекомендации по их реализации для двух тем курсового проекта:

1. Программируемый логический контроллер на многокристальном микропроцессоре (раздел 1);
2. Сопроцессор с архитектурой RISC (раздел 2).

В разделах 3, 4 приведено краткое описание используемой в проекте элементной базы и даны варианты ее применения в проектируемых устройствах.

Раздел 5 определяет общие требования к оформлению курсового проекта решения и содержанию пояснительной записки. Перечень литературы приведен в разделе 6.

# 1. Программируемый логический контроллер на многокристальном микропроцессоре

## **1.1. Общие сведения о проектируемом устройстве.**

Программируемый логический контроллер (далее ПЛК) представляет собой простейшую специализированную ЭВМ для управления несложными объектами и технологическими процессами [1]. Входными сигналами ПЛК служат сигналы от  $m_1$  двоичных датчиков объекта и, возможно, сигналы прерываний. Выходные сигналы передаются контроллером к  $m_2$  исполнительным механизмам (рис.1).

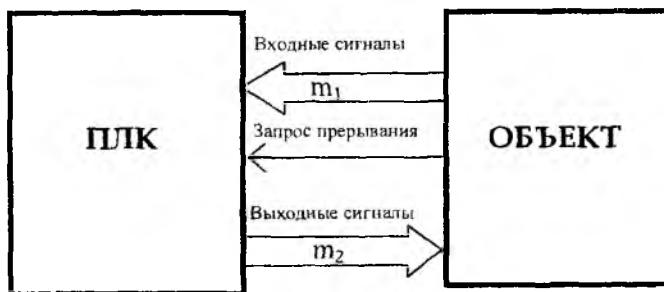


Рис. 1.1. Связи ПЛК с объектом управления

Система команд ПЛК ориентирована на выполнение следующих функций:

- вычисление любой (с учетом ограничений на объем программной памяти) из  $m_2$  булевых функций от  $m_1$  аргументов  
 $Y_j = f_j(x_0, x_1, \dots, x_b, \dots, x_{m_1-1})$ , где  $j = 0, 1, \dots, m_2 - 1$ ;
- реализация простейших вычислительных процедур, используемых, в основном, для реализации временных задержек;

- обработка запроса прерывания.

Управление объектом осуществляется по программам, фиксируемым пользователем в постоянной памяти программ (ПП). Для хранения промежуточных данных и состояний прерываемых программ используется память данных (ПД).

## **1.2. Задание на проектирование ПЛК**

Вариантом задания является строка из таблицы 1.1, номер которой совпадает с номером студента в журнале студенческой группы.

В строке указаны параметры устройств, входящих в состав ПЛК. Первые четыре параметра в строке определяют параметры памяти и блоков ввода-вывода:

$C_{пп}$  - количество ячеек памяти программ ПП (емкость ПП),

$C_{пд}$  - количество ячеек памяти данных ПД (емкость ПД),

$m_1$  - количество портов ввода для приема входных сигналов,

$m_2$  - количество портов вывода для выдачи выходных сигналов.

Параметр «ТИП АРХИТЕКТУРЫ И АДРЕСАЦИЯ ПДЭ» определяет организацию адресных пространств всех программно доступных элементов (ПДЭ) памяти и портов ввода-вывода:

1 - адресные пространства ПП и ПД организуются по принципу гарвардской архитектуры, а порты ввода и порты вывода адресуются в собственных адресных пространствах;

2 - адресные пространства ПП и ПД организуются по принципу принстонской архитектуры и образуют общее адресное пространство с портами ввода и вывода.

Параметр « НАЛИЧИЕ ПЕРЕКЛЮЧАТЕЛЯ S» задает наличие или отсутствие переключателя S, устанавливающего режим работы ПЛК:

«ОСТАНОВ» - режим ожидания,  
 «РАБОТА» - режим исполнения программ.

Таблица 1.1.

Варианты заданий

№ п/з	Емкость памяти программ (ячеек) Спр	Емкость памяти данных (ячеек) Сдд	Число портов		Тип архитектуры адресации ПДС	Панель переключателя S	Запрос прерывания		Элементная ба- за процессора
			ввода m <sub>1</sub>	вывода m <sub>2</sub>			маскиру- мый	немаски- руемый	
1	256	256	32	4	1	да	+		K1804
2	512	256	8	8	2	нет		+	K589
3	256	512	16	4	2	да	+		K589
4	2048	256	8	4	2	нет		+	K1804
5	1024	512	4	8	1	да	+		K589
6	1024	64	8	4	1	нет		+	K589
7	256	128	16	16	1	да	+		K1804
8	256	64	16	16	2	да		+	K589
9	1024	512	8	8	1	нет	+		K589
10	512	2048	4	4	2	да		+	K1804
11	2048	256	4	8	2	да	+		K1804
12	512	64	4	12	2	нет		+	K589
13	256	256	32	16	1	нет	+		K1804
14	512	256	8	4	2	нет		+	K589
15	2048	64	4	12	2	да	+		K1804
16	2048	128	12	4	2	да		+	K1804
17	1024	256	8	8	1	да	+		K589
18	4096	1024	4	8	1	нет		+	K1804
19	2048	512	8	4	2	да	+		K1804
20	1024	512	16	8	1	да		+	K589
21	512	256	8	8	2	да	+		K589
22	2048	256	12	4	2	нет		+	K1804
23	256	64	8	4	1	да	+		K1804
24	4096	1024	4	8	1	нет		+	K1804

Параметр «ЗАПРОС ПРЕРЫВАНИЯ» определяет наличие или отсутствие маскирования прерывания, формируемого объектом управления.

Параметр «ЭЛЕМЕНТНАЯ БАЗА ПРОЦЕССОРА» определяет серию интегральных схем, используемую для построения процессора ПЛК. Описание серий можно найти в разделе 3 настоящих указаний и в справочной литературе, перечисленной в разделе 6.

### **1.3. Рекомендации по разработке ПЛК.**

#### **Архитектура ПЛК**

Под архитектурой ПЛК, как и любой ЭВМ, чаще всего понимают набор сведений, необходимый программисту-пользователю для написания программ на языке машинных команд. Кроме структуры в архитектуру ПЛК включают:

- типы, формы представления и форматы обрабатываемых данных;
- программистскую модель;
- сведения о системе команд и используемых способах адресации;
- сведения об организации систем прерывания и ввода-вывода информации.

Разработка архитектуры ПЛК предшествует этапам системотехнического, схемотехнического и конструкторского проектирования, причем ПЛК с одной и той же архитектурой могут быть реализованы с использованием различных инженерных решений.

#### **Структурно-логическая организация и функционирование ПЛК**

Укрупненная структурная схема ПЛК показана на рис.1.2. Основу ПЛК составляет процессор, состоящий из двух блоков: операционного (ОБ) и управляющего (УБ),

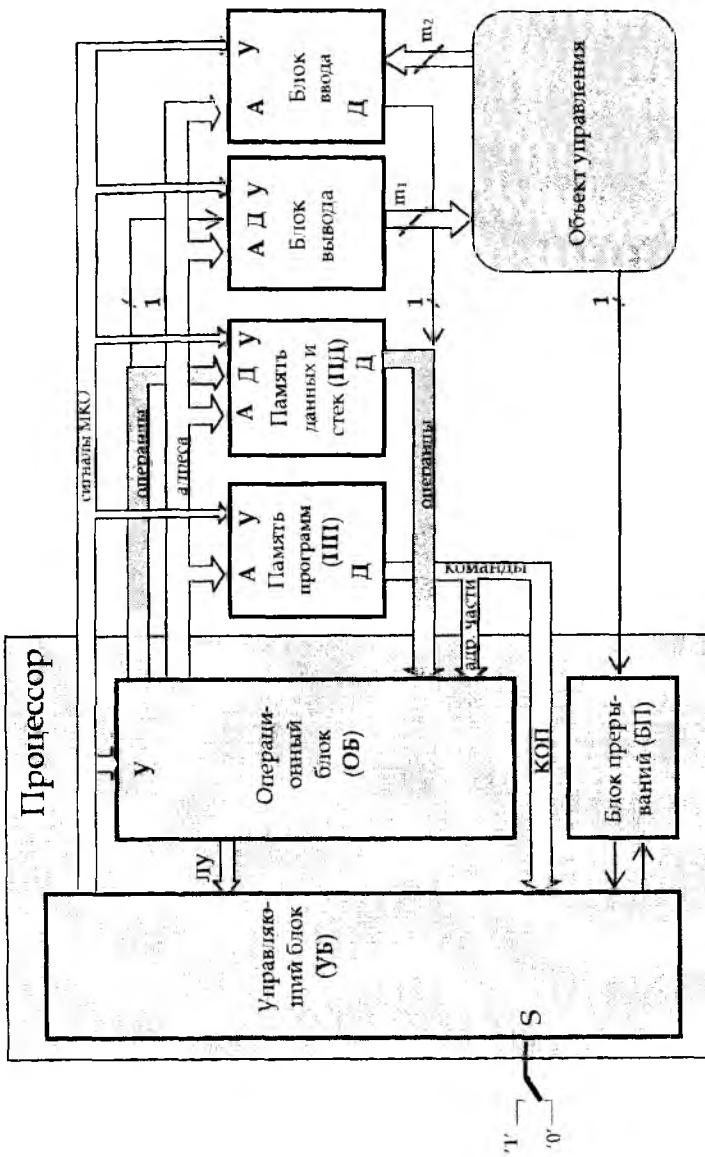


Рис. 1.2. Укрупненная структурная схема ПЛК

что соответствует принципу декомпозиции цифрового устройства на операционную и управляющую части. Кроме этого в состав процессора входит блок прерываний (БП). Процессор подключен посредством адресных (A), информационных (Д) и управляющих (У) шин к памяти программ (ПП), памяти данных (ПД) и блокам ввода-вывода (БВВ). При проектировании ПЛК данная схема уточняется и детализируется в соответствии с заданием.

Адресное взаимодействие процессора с устройствами ПП, ПД, БВВ осуществляется путем формирования в операционном блоке адресов обращения и их передачи в эти устройства с добавлением необходимых управляющих сигналов (выбор устройств, режим обращения и т.п.). Запись операндов может быть осуществлена из ОБ в память данных ПД или в блок вывода. Адресные части команд из ПП и операнды из ПД и блока ввода принимаются в ОБ, а коды операций команд (КОП) принимаются в управляющий блок УБ для последующей дешифрации.

Управляющий блок строится в проекте как автомат с программируемой (хранимой в памяти) логикой. Осведомительными сигналами или сигналами логических условий (ЛУ) для него являются флаги (признаки), вырабатываемые блоком ОБ, сигналы от БП и сигнал от переключателя S. При этом нулевому значению S соответствует режим “останов”, а единичному – “работа”. Укрупненная блок-схема микропрограммы, реализующей рабочий цикл процессора ПЛК, представлена на рис. 1.3.

Сложность реальной микропрограммы рабочего цикла ПЛК зависит от числа и сложности команд, организации их выборки и обработки, набора микроопераций в ОБ, типа запроса прерываний, наличия переключателя S.

Микропрограмма (МКП) начальной установки (адрес ее начала должен устанавливаться в УБ автоматически после включения ПЛК или по нажатию специальной кнопки) осуществляет обычно следующие действия:

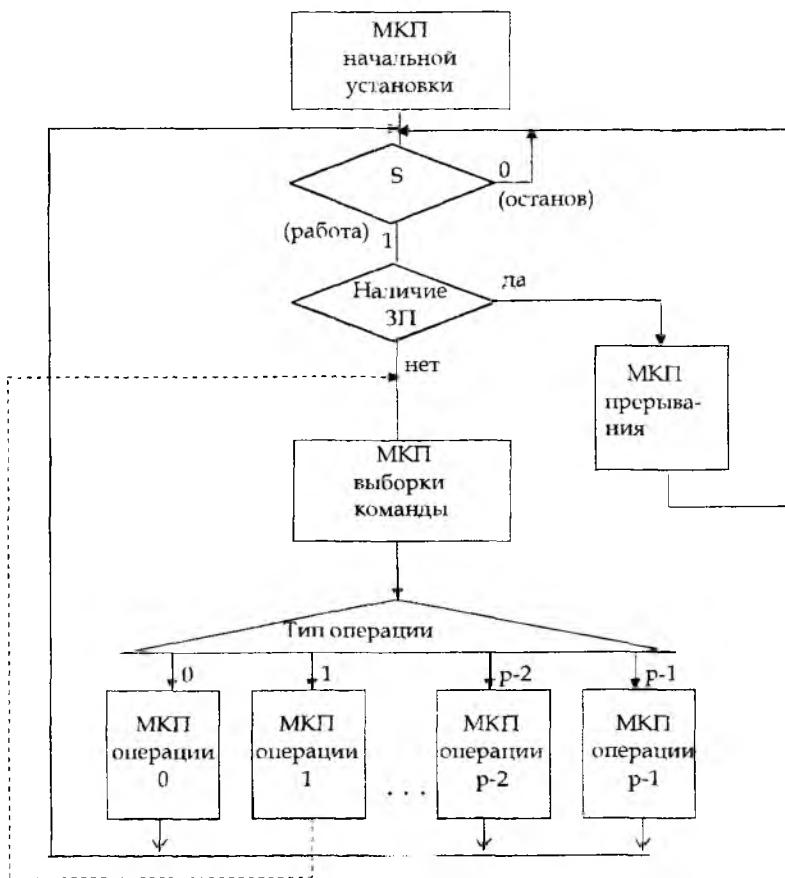


Рис.1.3. Укрупненная блок-схема микропрограммы рабочего цикла ПЛК

- обнуление программного счетчика (PC) в ОБ, что обеспечивает выборку первой команды из 0-ой ячейки ПП;

- установка в начальное состояние триггеров БП, триггеров фиксации выходных сигналов в блоке вывода и т.п.

Анализ состояния переключателя S для перехода в режим “останов” или “работа” исключается при его отсутствии. Наличие прерывания проверяется путем опроса состояния сигнала запроса прерывания. При наличии запроса осуществляется переход к микропрограмме прерывания (см. ниже).

МКП выборки команды реализует обращение в ПП по адресу из программного счетчика с его последующей инкрементацией для перехода к следующей команде. КОП считанной из ПП команды подается в УБ для формирования адреса перехода к МКП выполнения операции, заданной командой.

### **Программистская модель, способы адресации и система команд ПЛК.**

Все компоненты архитектуры проектируемого ПЛК являются взаимосвязанными и должны соответствовать перечню операций и форматам команд, входящих в реализуемую систему команд.

Данными, обрабатываемыми в ПЛК, являются:

- a) k-разрядные целые числа без знака, представляющие собой адреса;
- б) n-разрядные числа без знака;
- в) булевы переменные в виде бита в n-разрядном двоичном коде (возможно в виде n-разрядных кодов с одинаковыми значениями разрядов).

Программистскую модель ПЛК составляют все программно-доступные, т.е. идентифицируемые командами (кодами операций и адресными частями команд) элементы, в которых располагаются сами команды, входные, выходные и промежуточные данные. Функционирование ПЛК

можно организовать с использованием программистской модели, включающей:

- программный счетчик РС;
- аккумулятор А;
- один регистр общего назначения - R<sub>i</sub>;
- С<sub>пп</sub> ячеек памяти программ;
- С<sub>пд</sub> ячеек памяти данных;
- m<sub>1</sub> однобитовых портов ввода, через которые в ПЛК поступают сигналы от m<sub>1</sub> датчиков объектов;
- m<sub>2</sub> однобитовых портов вывода, с которыми соединяются входы m<sub>2</sub> исполнительных механизмов;
- триггер разрешения/запрещения маскируемых прерываний, управляемый командами типа EI и DI;

Что касается регистра флагов, по битам которого организуются переходы в программах процессоров ЭВМ, то без него можно обойтись, если осуществлять переход только по нулевому содержимому аккумулятора А.

В командах ПЛК достаточно использовать подразумеваемую, непосредственную и прямую адресацию данных и програмно-доступных элементов. При выборе системы команд следует стремиться к минимальному их числу, удовлетворяя, однако, требованию алгоритмической полноты системы команд, т. е. возможности представления в терминах команд алгоритмов управления объектом и обслуживания прерываний.

С учётом этого требования и выбранных способов адресации достаточной оказывается система из шестнадцати команд и для кодирования кода операции используется четыре разряда, т.е. n<sub>код</sub>=4. Адресная часть команд с прямой адресацией определяется следующими формулами (с округлением до большего целого):

- в случае 1 (гарвардская архитектура):  
 $k = \log_2\{\max [C_{пп}, C_{пд}, m_1, m_2]\};$

- в случае 2 (принстонская архитектура):  $k = \log_2\{C_{\text{пр}} + C_{\text{пд}} + \max[m_1, m_2]\}$ .

Для приведенных в таблице 1.1 значений  $C_{\text{пр}}$ ,  $C_{\text{пд}}$ ,  $m_1$  и  $m_2$  значение  $k = 8; 10; 12$ .

Разрядность коротких команд, таких как EI, DI, IRET и др. определяется только кодом операции.

Выбор разрядности ячеек памяти программ ( $n_{\text{пр}}$ ), памяти данных ( $n_{\text{пд}}$ ) и операционного блока ( $n_{\text{об}}$ ) зависит, прежде всего, от того располагаются ли длинные и короткие команды в ячейках ПП одного (длинного) формата или для хранения длинных команд используются несколько ячеек ПП.

Более понятной и более быстродействующей будет организация ПЛК, в которой принято:

$$n_{\text{пр}} = n_{\text{код}} + k = 4+k;$$

$$n_{\text{пд}} = n = k;$$

$n_{\text{об}} = n$ , т.е. и короткие и длинные команды располагаются в ячейках памяти программ одной и той же разрядности, а разрядность ячеек памяти данных и ширина обработки в ОБ одинакова и равна  $n$ .

Для выбора системы команд и их форматов можно воспользоваться примером, приведенным в таблице 1.2. Здесь A - аккумулятор, R- регистр общего назначения, PC-программный счётчик , addr - адреса ПП или ПД, port – адреса портов ввода-вывода, data- непосредственный операнд, “~” - произвольное значение бита. Пример приведён для случая принстонской архитектуры с адресацией внешних ПДЭ в общем адресном пространстве и  $C_{\text{пр}}=128$ ,  $C_{\text{пд}}=64$  и  $m_1=m_2=16$ . Для их адресации используется  $2^k = 2^8 = 256$  адресов, причём адреса от 0 до 127 (от 00h до 7Fh) адресуют ячейки памяти программ (в командах JMP addr и JA addr разряд  $A_7 = 0$ ). Для адресации ячеек памяти данных выбираются следующие 64 адреса от 128 до 191 (от 80h до 0BFh) и в командах обращения к ПД  $A_7 = 1$  и  $A_6 = 0$ .

Таблица 1.2

## Пример системы команд

Груп лы ко- манд	№ п/п	Мнемоника	Операция	Формат команды												Время выпол- нения (так- тов/мк с)		
				КОП				Адресная часть										
				11	10	9	8	7	6	5	4	3	2	1	0			
Команды пересылок и об- ращения к ПД и блокам ввода - вывода	1	MOV A, data	A←data	0	0	0	0											
	2	MOV R, data	R←data	0	0	0	1											
	3	MOV R, A	R←A	0	0	1	0									-		
	4	MOVA, R	A←R	0	0	1	1									-		
	5	MOV A, M	A←(addr)	0	1	0	0	1	0							адрес ПД		
			A←(port)	0	1	0	0	1	1	~	~					адрес порта ввода		
	6	MOV M, A	(addr)← A	0	1	0	1	1	0							адрес ПД		
			(port)← A	0	1	0	1	1	1	~	~					адрес порта вывода		
Логические операции	7	NOT A	A← $\bar{A}$	0	1	1	0									-		
	8	AND A, R	A← A&R	0	1	1	1									-		
	9	OR A, R	A← AvR	1	0	0	0									-		
Арифмети- ческие операции	10	DEC A	A←A-1	1	0	0	1									-		
	11	DEC R	R←R-1	1	0	1	0									-		
Команды переходов	12	JMP addr	PC← addr	1	0	1	1	0								адрес памяти программ		
	13	JA addr	PC← addr при A=0	1	1	0	0	0								адрес памяти программ		
Команды управ- ления прерыва- ниями	14	EI	Разрешить прерывание TMI:=1	1	1	0	1									-		
	15	DI	Запретить прерывание TMI:=0	1	1	1	0									-		
	16	IRET	Возврат из прерывания	1	1	1	1									-		

Для обращения к портам используются команды с теми же кодами операциями, но адресами от 0C0h до 0FFh

$(A_7 = A_6 = 1)$ . При этом каждый из 16 портов – как ввода, так и вывода – адресуется четырьмя младшими разрядами адреса, а разряды  $A_5$ ,  $A_4$  – любые (неполная адресация, см. [2]).

В случае адресации портов в собственных адресных пространствах в системе команд имеют место команды IN port и OUT port со своими кодами операции.

Из анализа примера также следует, что запрос прерывания – маскируемый (команды 14 и 15).

В таблицу 1.2 следует также поместить времена выполнения команд – в тактах и микросекундах, которые определяются после микропрограммирования в реальных форматах микрокоманд и выбора периода импульса синхронизации (такта).

Для проверки алгоритмической полноты выбранной системы команд студенту нужно составить программу вычисления пары булевых функций от трёх-четырёх переменных и передачи их значений к исполнительным механизмам объекта через неравные промежутки времени.

## **Разработка функциональной схемы ПЛК, формата микрокоманды и микропрограмм.**

Функциональная схема ПЛК образуется из функциональных схем образующих его блоков, построенных на заданной элементной базе - микропроцессорном комплекте и элементах серий ТТЛ, БИС ОЗУ и ПЗУ.

Операционный блок получается соединением  $N_{\text{ппэ}} = n/2$  центральных процессорных элементов серии К589 и из  $N_{\text{ппэ}} = n/4$  элементов серии К1804. При этом в случае серии К589 образуются две  $n$  – разрядные выходные шины – адреса и данных, являющиеся выходами соответствующих регистров с  $z$  – состоянием.

Как образуются  $n$  – разрядные обработчики из секций ЦПЭ, можно узнать из раздела 3. Здесь также не обсуждается вопрос построения памяти из выбранных БИС ОЗУ и ПЗУ.

Операционный блок из ЦПЭ серии К1804 имеет одну выходную шину. Для образования другой – адресной шины необходим  $n$  – разрядный регистр-защёлка адреса. Естественно, в формате микрокоманды предусматривается разряд для задания микрооперации «Приём в регистр адреса».

Блок ввода представляет собой мультиплексор с  $m_1$  информационными входами  $D_0 \div D_{m_1 - 1}$  и  $\log_2 m_1$  управляющих входов (рис. 1.4). Наличие входа выборки  $VZ$  и способ формирования сигнала «Разрешение ввода» зависит от типа архитектуры и адресации ПДЭ (см. таблицу 1.1), а также и от выбранного способа подключения мультиплексора к операционному блоку.

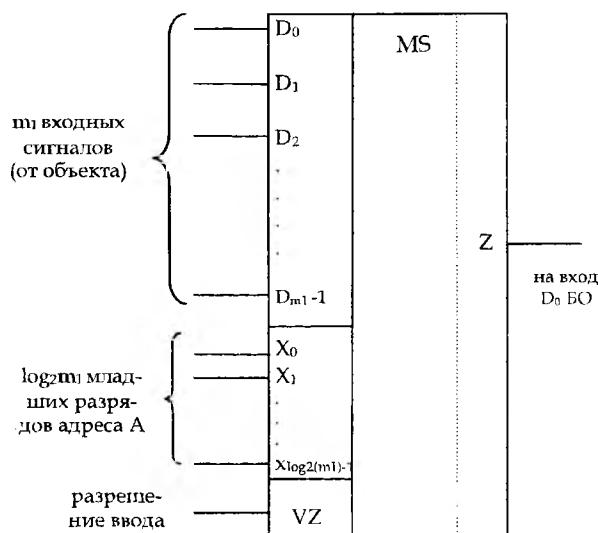


Рис. 1.4. Функциональная схема блока ввода

Блок вывода (рис. 1.5) состоит из  $m_2$  триггеров  $T_0 - T_{m_2-1}$ , являющихся собственно выходными портами ПЛК, и дешифратора с  $m_2$  выходами. Последний является, по сути дела, демультиплексором сигнала «вывод», формирование которого определяется заданной архитектурой ПЛК. В качестве сигнала «сброс» рациональнее всего использовать сигнал начальной установки адреса микрокоманды в УБ (или отдельную МКО).

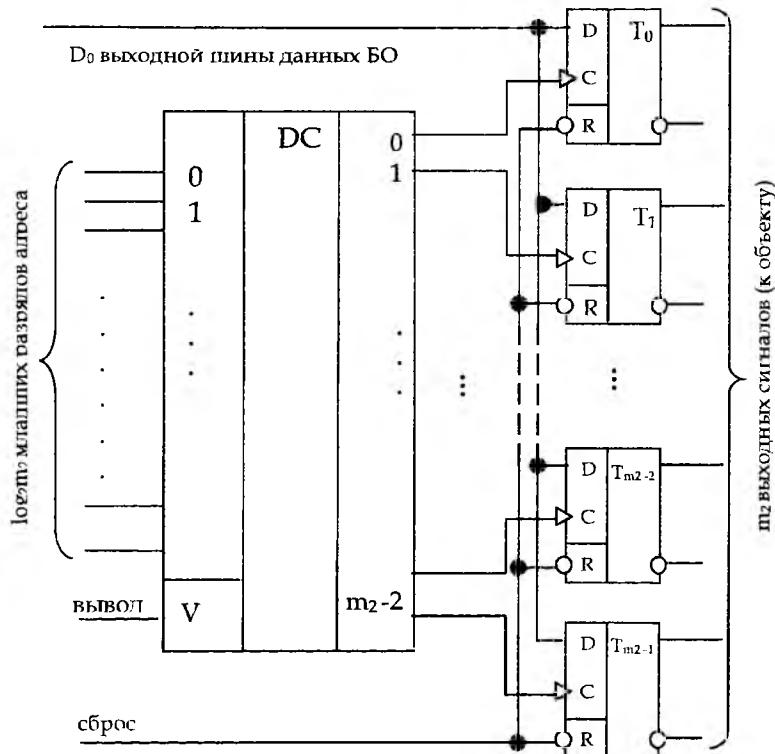


Рис. 1.5. Функциональная схема блока вывода

Блок прерываний (БП), не смотря на простоту своей схемы, требует более детального описания. Он состоит из двух триггеров (рис.1.6): триггера прерываний ТПР и триггера маски прерываний ТМП. Первый служит для запоминания запроса прерывания (INT или NMI), представляющего собой короткий импульс (рис 1.7). Триггер маски является программно доступным элементом (в командах EI и DI) в случае маскируемого запроса прерывания INT. В случае немаскируемого прерывания, он служит для исключения повторного прерывания в случае появления NMI во время выполнения прерывающей программы.

Обнаружение запроса прерывания осуществляется либо по осведомительным сигналам с выходов триггеров ТПР и ТМП (вариант 1, конъюнктор в схеме рис.1.6 отсутствует), либо по сигналу логического условия ТПР&ТМП при наличии конъюнктора (вариант 2). Выбор способа определяется разработчиком.

На рис.1.7 цифрой 1 обозначена команда основной программы (программы управления объектом), после которой наступает прерывание, а цифрой 2 – первая команда прерывающей команды. Сброс триггеров ТПР, ТМП осуществляется в процессе исполнения МКП прерывания. Завершающая часть прерывающей программы имеет следующие особенности:

- в случае маскируемого запроса восстановление состояния ТМП выполняет МКП команды EI, причём переход осуществляется к МКП выборки команды (см. пунктир на рис.1.3). Тем самым гарантируется выполнение последней команды прерывающей программы – команды IRET;
- в случае немаскируемого прерывания команда EI отсутствует, а установку в единицу триггера маски выполняет МКП команды IRET.

из микрокоманды

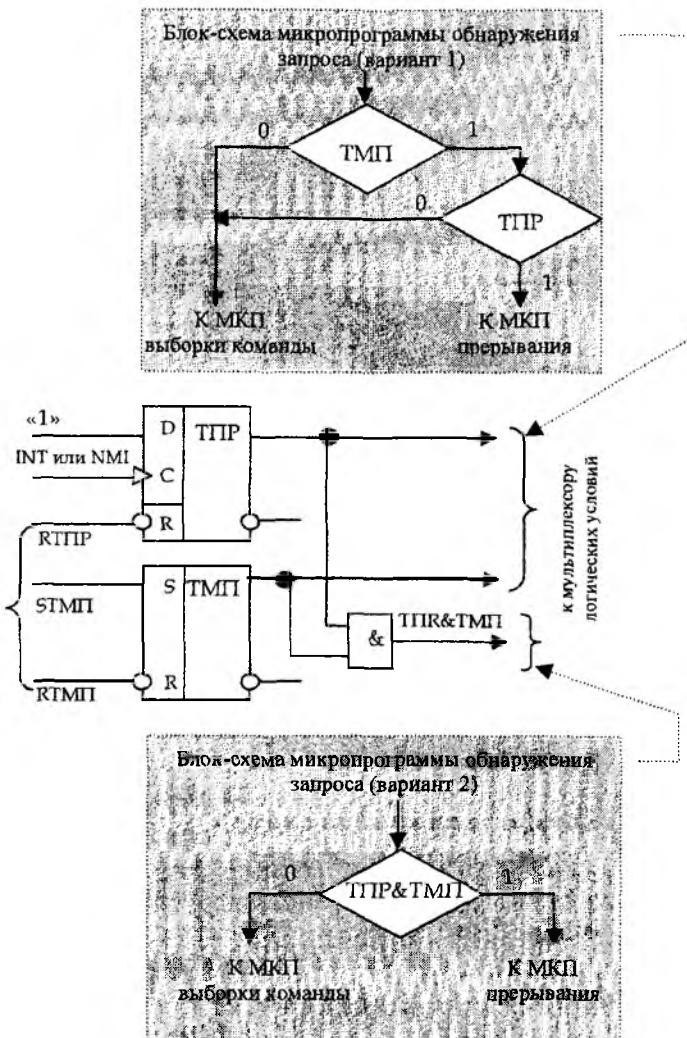


Рис. 1.6. Функциональная схема блока прерывания

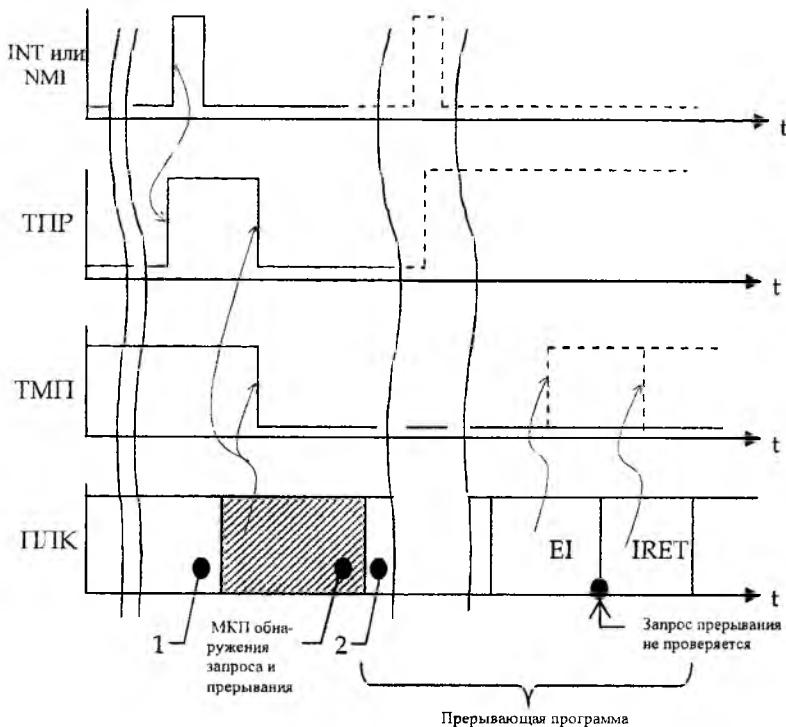


Рис. 1.7. Временная диаграмма процесса прерывания.

В таблице 1.3 показано, в каких точках микропрограммы рабочего цикла ПЛК происходит сброс и восстановление состояний триггеров ТПР и ТМП.

Микропрограмма прерывания требует формирования адресов ячеек ПД для сохранения состояния РС, А и R и адреса начала прерывающей программы. Чтобы не использовать микроконстанту и не усложнять устройство, рационально сохранять регистры в последних ячейках ПД. Адресом начала прерывающей программы удобно сделать адрес  $C_{пп} - 1$ , расположив в ней команду перехода на фактическое начало этой программы.

Таблица 1.3  
Манипуляции с ТПР и ТМП

Запрос прерывания	Триггер прерывания ТПР		Триггер маски прерывания ТМП	
	Установка в «0»	Установка в «1»	Установка в «0»	Установка в «1»
INT (маскируемый запрос)	1) в МКП начальной установки 2) в МКП прерывания	сигналом INT	1) в МКП начальной установки 2) в МКП команды DI 3) в МКП прерывания	в МКП команды EI
NMI (немаскируемый запрос)	1) в МКП начальной установки 2) в МКП прерывания	сигналом NMI	в МКП прерывания	1) в МКП начальной установки 2) МКП команды IRET

При разработке функциональной схемы ПЛК определяется полное множество входных сигналов УБ (включая логические условия, внешние сигналы и разряды кода операции команды) и набор сигналов микроопераций и кодов, требуемых для управления блоками процессора, (то есть формат микрокоманды). Здесь могут быть полезны сведения из [3].

Операционная часть микрокоманды включает поле управляющего кода и бит входного переноса для операционного блока, поле маски (в случае K589), биты микроопераций для памяти и блоков ввода, вывода и прерываний.

Адресная часть микрокоманды содержит поля задания способа перехода, адреса следующей микрокоманды и управления мультиплексором логических условий в УБ. Суммарное число разрядов операционной и адресной частей микрокоманды определяет разрядность ячеек памяти микрокоманд  $n_{\text{имк}}$

Емкость памяти микрокоманд  $C_{\text{пмк}}$  в УБ определяется после разработки микропрограмм ПЛК. Последние представляются двумя способами:

- в виде содержательных граф – схем алгоритмов (ГСА) для каждого блока в микропрограмме рабочего цикла ПЛК;
- в виде таблиц «адрес – содержимое ячейки», т.е. в виде кодированных микропрограмм. Общее число строк в этих таблицах, дополненной до ближайшего числа, кратного целой степени 2, есть ёмкость ПМК. После этого возможны корректировка адресной части микрокоманды в сторону уменьшения числа её разрядов и, следовательно, упрощение схемы УБ.

Управляющий блок разрабатывается с использование раздела 3 настоящих методических указаний или [4, 5, 6]. Он состоит из:

- памяти микрокоманд (ПМК), объёмом  $C_{\text{пмк}} \times n_{\text{пмк}}$  бит;
- БИС блоков микропрограммного управления (БМУ);
- регистра адреса РАМК или информационного регистра РИМК (могут отсутствовать);
- мультиплексора логических условий;
- узла синхронизации и начальной установки (УСНУ).

Блок микропрограммного управления в серии К589 формирует девятиразрядный адрес микрокоманды и адресует память ёмкостью 512 ячеек. Этого количества достаточно для размещения микропрограмм ПЛК. В серии К1804 один блок (секция) БМУ формирует четырехразрядный адрес, поэтому необходимо соединение двух БМУ для адресации памяти микрокоманд ёмкостью до 256 ячеек.

Мультиплексор логических условий служит коммутатором осведомительных сигналов, формируемых блоком обработки, блоком прерываний и переключателем режимов S.

Функциональная схема УСНУ – общая для двух тем проекта - приведена в разделе 4 данных методических указаний.

### **Разработка принципиальной схемы устройства и выбор параметров импульса синхронизации.**

Рекомендации по использованию заданной элементной базы при разработке принципиальных схем приведены в разделе 3 данных методических указаний. Особенностью принципиальной схемы ПЛК является наличие двух электрических соединителей (разъемов): одного - для связи с объектом управления и другого - для соединения с блоком питания.

## 2. Сопроцессор с архитектурой RISC

### 2.1. Общие сведения о проектируемом устройстве.

Структурная схема подключения сопроцессора (СП) приведена на рис. 2.1.

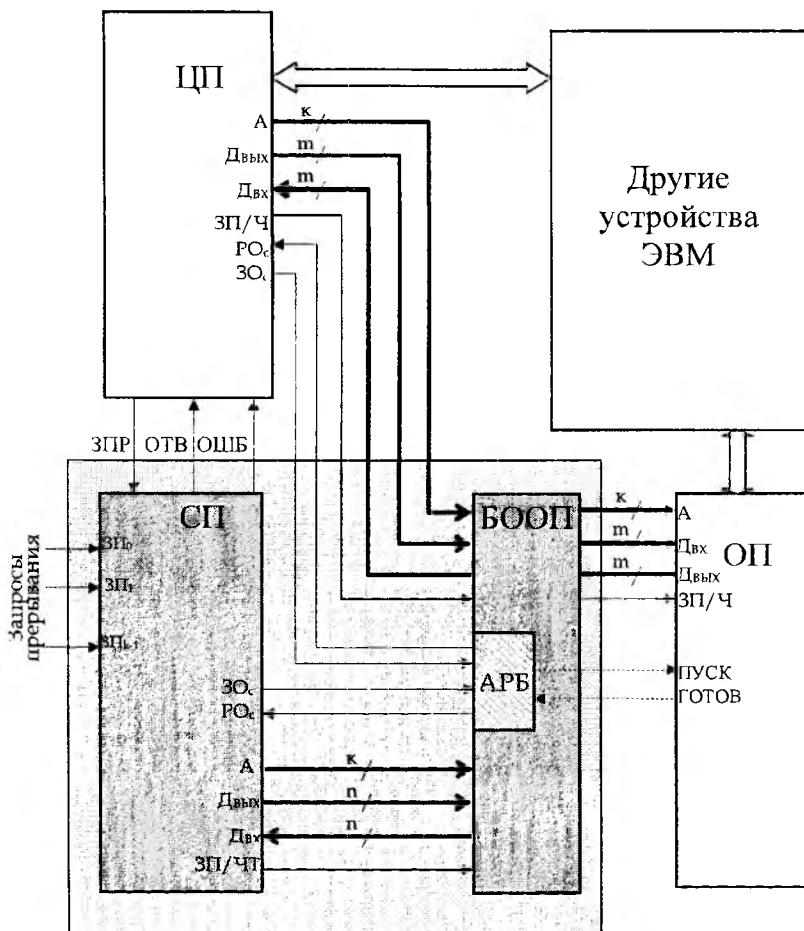


Рис.2.1. Структурная схема подключения СП

Сопроцессор (СП) функционирует параллельно с центральным процессором ЦП и использует общую с ЦП оперативную память (ОП). Для организации обращений двух процессоров в общую ОП используется блок обращений в ОП (БООП), в состав которого входит арбитр (АРБ) для разрешения конфликтов при одновременных обращениях процессоров.

Для взаимодействия устройств СП используются следующие сигналы и шины:

ЗПР	- сигнал запуска СП в работу,
ОТВ	- сигнал подтверждения запуска от СП,
ОШБ	- сигнал от СП об обнаружении ошибки,
$ЗП_0 \div ЗП_{k-1}$	- запросы прерывания,
А	- адресная шина,
Двых	- выходная шина данных,
Двх	- входная шина данных,
ЗП/ЧТ	- сигналы управления режимами записи/чтения ОП,
Пуск,	- сигналы управления ОП.
Готов	

Каждый из процессоров реализует собственную систему команд и работает по своей программе, находящейся в ОП. Для обращении в ОП процессоры формируют k-разрядный адрес, сигналы управления режимами записи/чтения (ЗП/ЧТ) и выставляют запрос на обращение, передаваемый в арбитр (ЗО<sub>С</sub> от СП и ЗО<sub>Ц</sub> от ЦП). Обращение в ОП разрешается при получении от арбитра сигнала разрешения (РО<sub>С</sub> для СП и РО<sub>Ц</sub> для ЦП). Обмен данными между СП и БООП выполняется по n-разрядным шинам, а между БООП и ОП по m-разрядным шинам.

Алгоритм работы СП, реализуемый на микропрограммном уровне, представлен на рис 2.2. Запуск СП в режим исполнения программы осуществляется ЦП выдачей сигнала ЗПР. После получения ЗПР=1 СП выдает сигнал

$OTB=1$  и осуществляет загрузку своих регистров начальными состояниями (предварительно оговоренными или загружаемыми из фиксированной области ОП значениями). Далее СП ожидает от ЦП сброса сигнала ЗПР=0. Для ис-

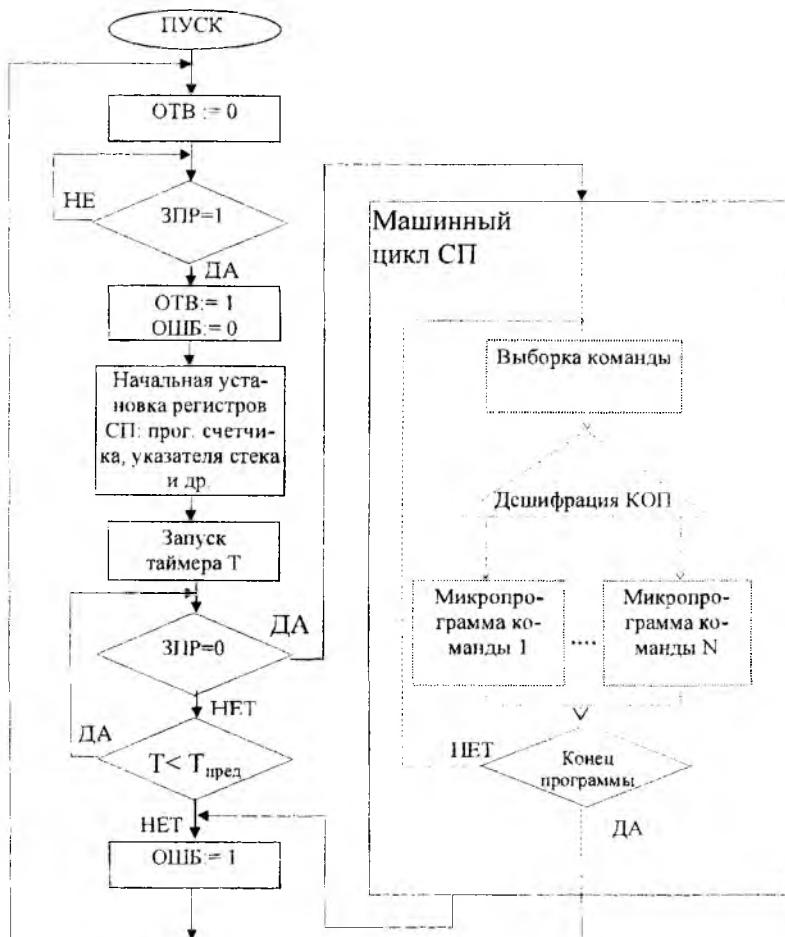


Рис.2.2. Укрупненный алгоритм работы СП

ключения “зависания” в этом состоянии в СП на микропрограммном уровне запускается таймер для ограничения времени ожидания. При отсутствии ЗПР=0 более  $T_{\text{пред}}$  СП выдает сигналы ОШБ=1, ОТВ=0 и возвращается в начальное состояние ожидания запуска. В противном случае после получения ЗПР=0 СП переходит к реализации программы, размещенной в ОП путем повторения машинного цикла исполнения команд. Выход из программы осуществляется при обнаружении признака «конец программы» (может формироваться специальной командой СП или сравнением программного счетчика с заданным конечным значением), после чего СП формирует сигнал ОТВ =0 и переходит в начальное состояние ожидания запуска. При возникновении ошибок в процессе исполнения программы СП выходит из машинного цикла и сообщает об этом в ЦП сигналаами ОШБ=1, ОТВ =0.

## 2.2. Программистская модель СП

Система команд разрабатываемого СП является подмножеством систем команд микропроцессоров серии K580 (i8080) или K1810 (i8086). Такая система является функционально полной для СП, однако включает только основные команды из базовых систем команд с упрощенными форматами и способами адресации. ЭВМ, реализующие подобные системы команд, получили название RISC (Reduced Instruction Set Computers). В СП с архитектурой K580 (i8080) все команды имеют формат длиной один байт (трехбайтные команды JC, CALL заменяются на однобайтные заданием адреса перехода в регистровой паре HL), а в СП с архитектурой K1810 (i8086) используются только двухбайтные команды (однобайтная команда RET преобразуется в двухбайтную добавлением нулевого байта).

Программистская модель СП должна включать все программно-доступные регистры базовых моделей микропроцессоров K580 (i8080) или K1810 (i8086). В таблице 2.1 приведены два варианта систем команд СП с использованием мнемоники команд, принятой в описаниях соответствующих микропроцессоров. Описания систем команд можно найти в литературе [7,8].

Таблица 2.1  
Варианты системы команд

K580 (i8080)	K1810 (i8086)
MOV R1,R2; MOV R,M; MOV M,R; PUSH RP; POP RP	MOV reg,r/m; MOV r/m, reg; MOV sr,r/m; MOV r/m, sr (sr≠CS); PUSH r/m; PUSH F ; POP r/m; POP F;
ADC R; SBB R; INC R; DCR R; INX RP; CMP R;	ADC reg,r/m; SBB reg,r/m; INC r/m; DCR r/m; CMP reg,r/m
ANA R; XRA R; ORA R; CMA;	AND reg r/m; XOR reg,r/m; OR reg,r/m; NOT r/m;
RAL; RAR;	RCL r/m,CL; RCR r/m,CL; SHL r/m,CL; SHR r/m,CL; SAR r/m,CL
PCHL; JC (по адресу в HL) (C= Z, NZ, C, NC); CALL(по адресу в HL); RST n; RET;	JMP WORD r/m; JMP DWORD r/m JC (C= Z, NZ, C, NC, S, NS, O, NO); CALL WORD r/m; CALL DWORD r/m; RET;
STC; CMC	CLC; STC; CMC
-	md=11; md=00 при r/m≠110

## **2.3. Структурная организация СП и БООП**

### ***2.3.1 Основные блоки СП, БООП и их подключение***

Обобщенная структурная схема СП и БООП представлена на рис.2.3. В состав СП входят блоки, перечисленные табл. 2.2. Организация каждого из перечисленных на схеме блоков определяется вариантом задания.

### ***2.3.2. Организация системы прерывания СП***

На рис. 2.4 приведена обобщенная структурная организация БП с микропрограммным опросом запросов на прерывание  $Z\Pi_0, \dots, Z\Pi_{k-1}$  через коммутатор КППР, подключаемый к коммутатору логических условий в составе МУУ. Основные функции системы прерываний реализуются при этом на микропрограммном уровне.

Прерывания могут быть вызваны либо уровнем (статическая фиксация), либо по фронту (динамическая фиксация) сигнала запроса. Для динамической фиксации может быть использован вариант схемы, приведенный на рис 2.4.

Система приоритетов прерывающих программ реализуется с использованием маскирования.

В проекте реализуются следующие варианты прерывающих процедур:

1. Задержка в работе СП на заданное время с возвратом к прерванной программе.
2. Загрузка в фиксированную область ОП значений заданных программно-доступных регистров СП.
3. Задержка в работе СП на заданное время с переходом в исходное состояние.

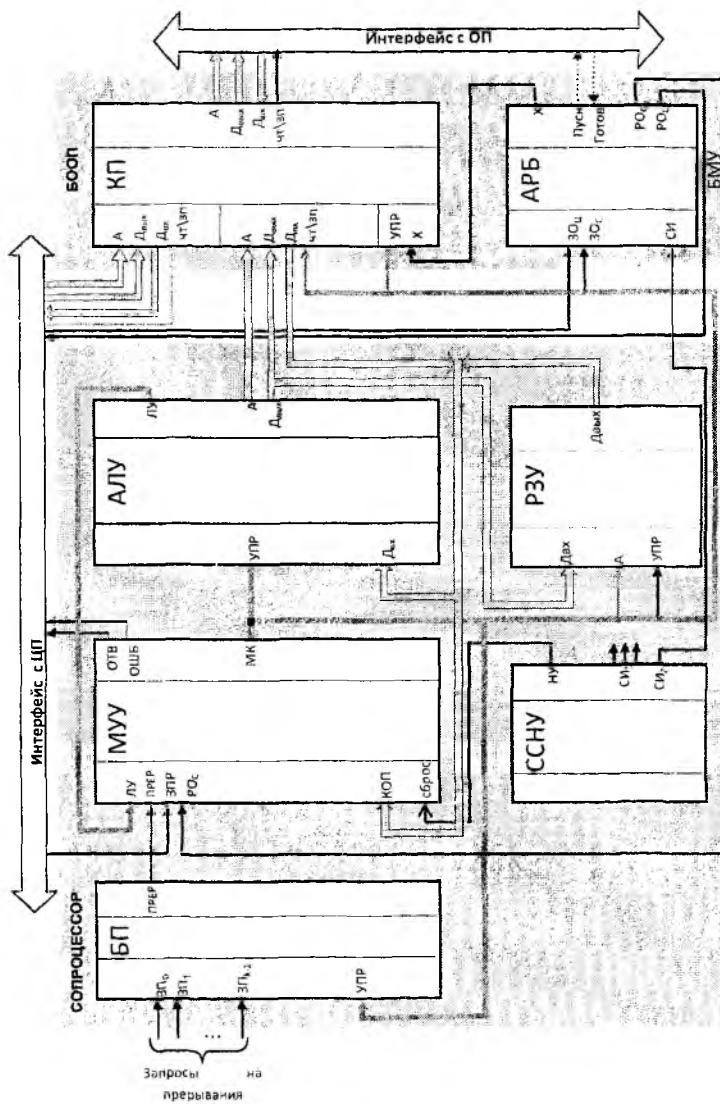


Рис. 2.3 Обобщенная структурная схема СП и БОП

## Блоки в составе СП

Таблица 2.2

<b>МУУ</b>	<b>Микропрограммное устройство управления</b>	ЛУ - логические условия (признаки) из АЛУ КОП - код операции команды, принимаемых из ОП ЭИР, СТВ, ОШБ - сигналы взаимодействия с ЦП ПРЕР - сигнал разрешения прерывания от блока прерываний Сброс - сигнал начальной установки процессора МК - операционная часть микрокомпьютера РО <sub>1</sub> - сигнал разрешения обращения в ОП
<b>АЛУ</b>	<b>Арифметико-логическое устройство</b>	Дж.Дых - входные и выходные шины данных А - адресная шина в ОП УПР-шина выбора функции АЛУ ЛУ-логические условия (признаки)
<b>РЗУ</b>	<b>Регистровое ЗУ</b>	Дж.Дых - входные и выходные шины данных. А - адресная шина. УПР - управление памятью памяти запись/чтение/запись отключения и др.). ЗП <sub>0</sub> -ЗП <sub>k-1</sub> - запросы на прерывания. УПР - шина управления БП (маска, сброс триггеров запросов, номер запроса и др.).
<b>БП</b>	<b>Блок прерываний</b>	СЧ <sub>1</sub> , СЧ <sub>2</sub> - сигналы синхронизации работы блоков СП и БОП. НУ - сигнал сброса СП в начальное состояние.
<b>СЧНУ</b>	<b>Схема синхронизации и начальной установки</b>	ЭО <sub>1</sub> , ЭО <sub>2</sub> - запросы на обращение в ОП от ЦП и СП. Х - сигнал управления КП. РО <sub>11</sub> , РО <sub>12</sub> - разрешение обращения в ОП для ЦП и СП. Пуск, Готов - квантующие сигналы организаций асинхронной работы ОП
<b>АРБ</b>	<b>Арбитр, управляющий доступом в ОП</b>	А - адресная, Д <sub>вход</sub> /Д <sub>выход</sub> - выходная/входная, ЧТ/ЗП - управляющая шины
<b>КП</b>	<b>Коммутатор шин</b>	

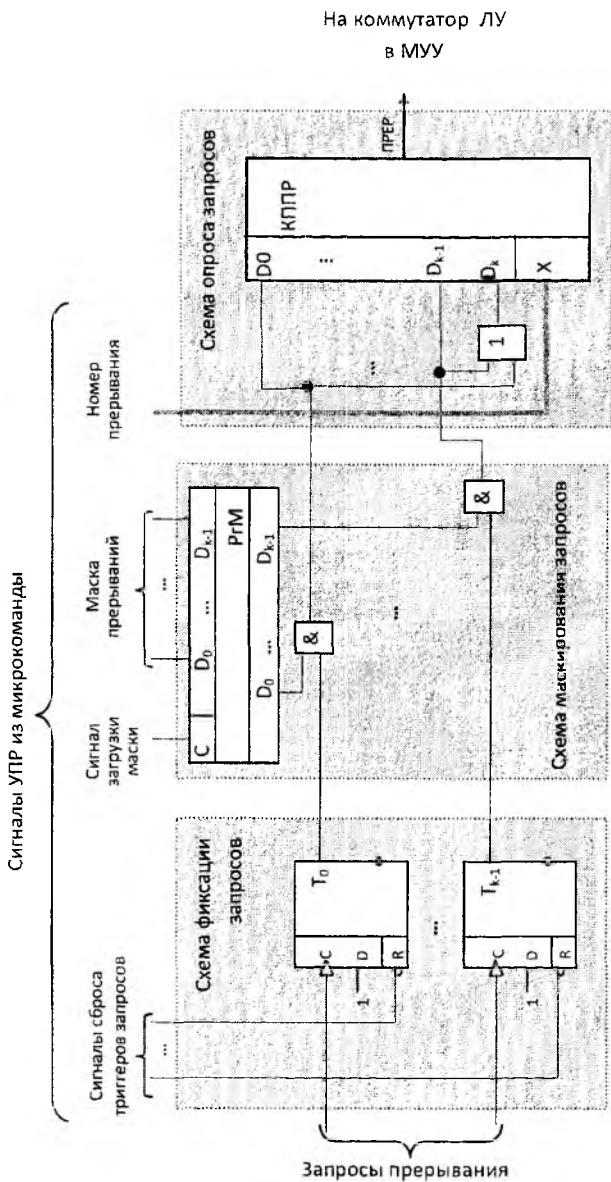


Рис.2.4 Структурная схема блока прерываний

Реализация прерывающих процедур выполняется на программном уровне в виде прерывающей программы.

Ниже приведена таблица 2.3 с вариантами организации системы прерываний, используемыми в проекте.

Таблица 2.3

Варианты организации системы прерываний

№ варианта	статическая\ динамическая фиксация запросов	наличие маскирования	прерывающая процедура
1	стат	нет	1 (по ЗП0)
2	стат	нет	2 (по ЗП1)
3	дин	нет	3 (по ЗП3)
4	дин	нет	1 (по ЗП2)
5	стат	да	1 (по ЗП0)
6	стат	да	2 (по ЗП1)
7	дин	да	3 (по ЗП3)
8	дин	да	1 (по ЗП2)

### 2.3.3. Организация арбитража ОП

СП и ЦП осуществляют обращения к ОП через общий коммутатор адресных, информационных и управляющих шин, входящий в состав БООП. Для разрешения конфликтов при их одновременных обращениях в состав БООП включен арбитр, на входы которого подаются запросы на обращения в ОП от двух процессоров. Арбитр управляет работой коммутатора и формирует сигналы разрешения обращений для СП и ЦП. Схема арбитра определяется способом взаимодействия процессоров с ОП. Варианты организации арбитра приведены в виде графа смены состояний на рис 2.5. Во всех вариантах большим приоритетом обладает ЦП.

1. Отсчет времени обращения с учетом заданной максимальной длительности цикла обращения в ОП осу-

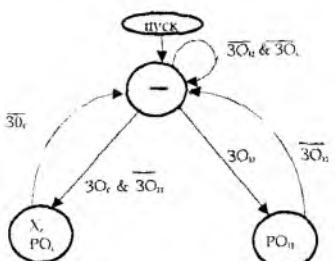
ществляется средствами процессора (рис. 2.5а). После получения от процессоров запросов арбитр в соответствии с присвоенными приоритетами выдает сигнал разрешения (РО) одному из процессоров и сигнал X для его подключения к ОП. Подключенный процессор занимает ОП и отсчитывает на микропрограммном уровне время готовности ОП, после чего завершает процедуру обращения и сбрасывает сигнал запроса.

2. Отсчет времени обращения с учетом заданной максимальной длительности цикла обращения в ОП осуществляется средствами арбитра (рис. 2.5б). Для отсчета времени обращения в состав арбитра включен таймер (счетчик тактов синхронизации арбитра). После получения от процессоров запросов арбитр в соответствии с присвоенными приоритетами подключает шины одного из процессоров к ОП (по сигналу X) и одновременно запускает таймер. После срабатывания таймера (по сигналу  $\alpha$ ) выдается разрешение РО подключенному процессору, который завершает цикл обращения и сбрасывает запрос.

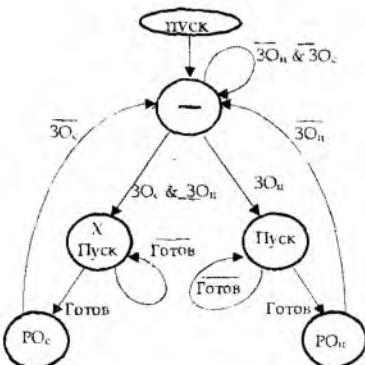
3. Арбитр реализует асинхронный способ обращения в ОП, при котором отсчет фактической длительности обращения осуществляется в ОП. (рис. 2.5в). С этой целью введен сигнал ПУСК, выдаваемый арбитром в ОП после подключения к ОП одного из процессоров. После завершения цикла ОП отвечает арбитру сигналом ГОТОВ, по которому арбитр формирует сигнал разрешения процессору. Процессор завершает цикл обращения и сбрасывает запрос.

## 2.4 Варианты заданий на проектирование СП.

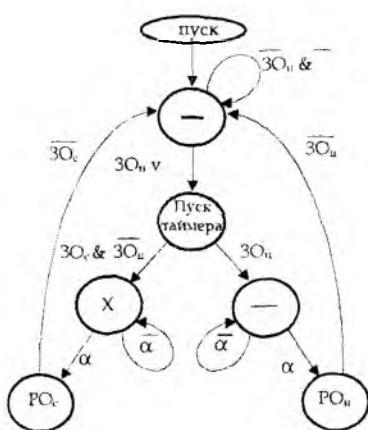
Варианты заданий на курсовой проект приведены в таблице 2.4.



а) синхронный обмен с отсчетом максимального времени обращения в СП



в) асинхронный обмен с отсчетом фактического времени обращения в ОП.



б) синхронный обмен с отсчетом максимального времени обращения в АРБ

Рис. 2.5. Варианты организации арбитра

Таблица 2.3.

## Варианты заданий

№ вар	Систе- ма команд 1-K580 2-K1810	АЛУ		МУУ		
		Элементная база	Разряд- ность	Элементная база БМУ	Элемен- тая база ПМК	Команда для микропро- граммирования
1.	1	K1804BC1	8	K1804BV1	K541	MOV R1,R2
2.	1	K1804BC2	8	K1804BV2	K573	MOV R,M
3.	1	K1804BC1	8	K1804BV2	K555	MOV M,R
4.	1	K1804BC2	8	K1804BV1	K541	PUSH RP
5.	1	K589ИK02	8	K589ИK01	K573	POP RP
6.	1	K1804BC1	8	K589ИK01	K555	ADC R
7.	1	K1804BC2	8	K589ИK01	K541	SBB R
8.	1	K589ИK02	8	K1804BV1	K573	INC R
9.	1	K589ИK02	8	K1804BV2	K555	DCR R
10.	2	K1804BC1	8	K1804BV1	K541	MOV reg,r/m
11.	2	K1804BC2	8	K1804BV2	K573	MOV r/m, reg
12.	2	K1804BC1	8	K1804BV2	K555	MOV sr,r/m
13.	2	K1804BC2	8	K1804BV1	K541	MOV r/m, sr
14.	2	K589ИK02	8	K589ИK01	K573	PUSH r/m
15.	2	K1804BC1	8	K589ИK01	K555	PUSH F
16.	2	K1804BC2	8	K589ИK01	K541	POP r/m
17.	2	K589ИK02	8	K1804BV1	K573	POP F
18.	2	K589ИK02	8	K1804BV2	K555	ADC reg,r/m
19.	1	K1804BC1	16	K1804BV1	K541	JNX RP
20.	1	K1804BC2	16	K1804BV2	K573	CMP R
21.	1	K1804BC1	16	K1804BV2	K555	ANAR
22.	1	K1804BC2	16	K1804BV1	K541	XRA R
23.	1	K589ИK02	16	K589ИK01	K573	ORA R
24.	1	K1804BC1	16	K589ИK01	K555	CMA
25.	1	K1804BC2	16	K589ИK01	K541	RAL
26.	1	K589ИK02	16	K1804BV1	K573	RAR
27.	1	K589ИK02	16	K1804BV2	K555	CALL
28.	2	K1804BC1	16	K1804BV1	K541	SBB reg,r/m
29.	2	K1804BC2	16	K1804BV2	K573	INC r/m
30.	2	K1804BC1	16	K1804BV2	K555	DCR r/m
31.	2	K1804BC2	16	K1804BV1	K541	CMP reg,r/m
32.	2	K589ИK02	16	K589ИK01	K573	JMP WORD r/m
33.	2	K1804BC1	16	K589ИK01	K555	JMP DWORD r/m
34.	2	K1804BC2	16	K589ИK01	K541	CALL WORD r/m
35.	2	K589ИK02	16	K1804BV1	K573	CALL DWORD r/m
36.	2	K589ИK02	16	K1804BV2	K555	JC

Таблица 2.3

## Варианты заданий (продолжение)

№ вар	БП	БОП					
		Вари- ант	Тип арбит- ра	Разрядность информационных шин	Совмещен- ность информа- ционных шин	Разряд- ность адресной шины	
1.	1	2		16	нет	16	0,5
2.	2	1		16	да	16	1,0
3.	3	2		16	да	16	0,7
4.	4	3		16	нет	16	1,0
5.	5	1		8	нет	16	0,4
6.	6	2		8	да	16	0,3
7.	7	3		8	да	16	0,8
8.	8	1		8	нет	16	0,5
9.	1	2		8	нет	16	0,6
10.	2	3		16	да	20	0,8
11.	3	1		16	да	20	0,6
12.	4	2		16	нет	20	0,4
13.	5	2		16	нет	20	0,5
14.	6	1		8	да	20	1,0
15.	7	2		8	да	20	0,7
16.	8	3		8	нет	20	1,0
17.	1	1		8	нет	20	0,4
18.	2	2		8	нет	20	0,3
19.	3	3		16	нет	16	0,8
20.	4	1		16	да	16	0,5
21.	5	2		16	да	16	0,6
22.	6	3		16	нет	16	0,8
23.	7	1		8	нет	16	0,6
24.	8	2		8	да	16	0,4
25.	1	3		8	да	16	0,4
26.	2	1		8	нет	16	0,5
27.	3	2		8	нет	16	1,0
28.	4	3		16	да	20	0,7
29.	5	1		16	да	20	1,0
30.	6	2		16	нет	20	0,4
31.	7	3		16	нет	20	0,3
32.	8	1		16	да	20	0,8
33.	1	2		16	да	20	0,5
34.	2	3		16	нет	20	0,6
35.	3	1		16	нет	20	0,5
36.	4	2		16	нет	20	0,6

### **3. Элементная база для построения процессоров**

Процессор может быть реализован в виде одной интегральной схемы (однокристальный микропроцессор) либо с использованием стандартных комплектов интегральных схем, отличающихся быстродействием, энергопотреблением, степенью интеграции и другими параметрами (многокристальный микропроцессор). В данном проекте выбран второй вариант.

В состав комплектов многокристальных микропроцессоров входят два основных типа БИС:

- центральный процессорный элемент (ЦПЭ) для построения секционированных АЛУ,
- блок микропрограммного управления (БМУ) для построения устройств управления процессоров.

Ниже приводятся краткие описания БИС ЦПЭ и БМУ используемых в проекте серий.

#### **3.1. Серия БИС на основе двухразрядных ЦПЭ (серия К589)**

##### **3.1.1. БИС ЦПЭ для построения АЛУ**

ЦПЭ представляет собой двухразрядную процессорную секцию, выполняющую функции арифметической и логической обработки двухразрядных операндов, их временного хранения и формирования различных признаков результатов. Структурная схема ЦПЭ приведена на рис.3.1. В состав ЦПЭ включены:

- двухразрядное комбинационное АЛУ,
- мультиплексоры входных operandов: МА на входе А и МВ на входе В АЛУ,
- блок регистров (РОН), включающий 10 регистров общего назначения R0 – R9 и регистр Т,

- вспомогательные регистры для временного хранения: регистр-аккумулятор данных (AC), регистр адреса (PA),
- буферы данных (БД) и адреса (БА) с тремя состояниями,
- дешифратор кода микроинструкций (ДШ МК) .

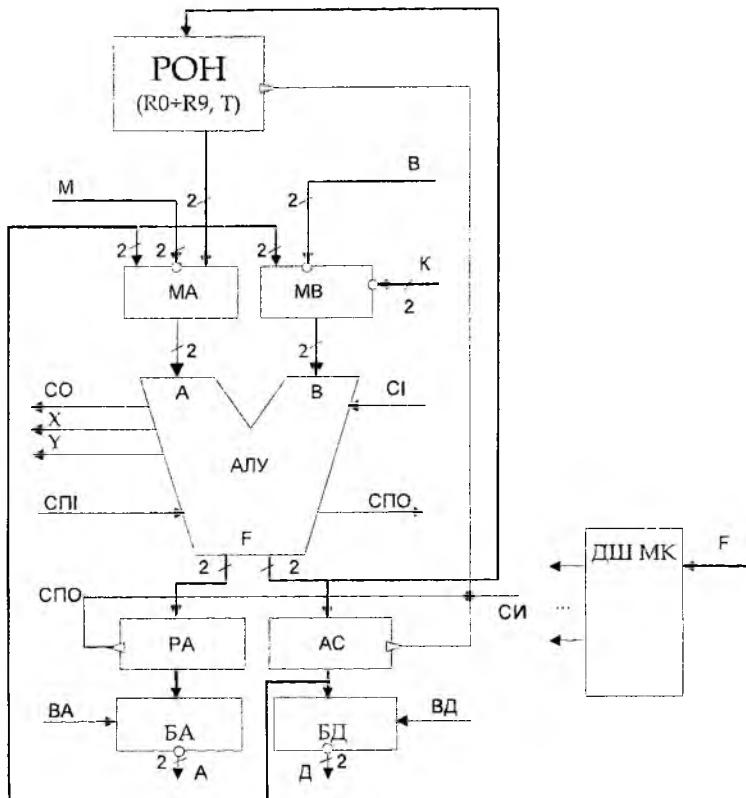


Рис. 3.1. Двухразрядная процессорная секция

Прием данных в ЦПЭ осуществляется по двум входным шинам: шине В, обычно используемой для приема информации из устройств ввода-вывода и ОЗУ данных, и шине М для приема адресной части команды из ОЗУ программ. Шина К предназначена для приема кода маски.

Выдача информации из ЦПЭ осуществляется по шине данных Д и шине адреса А. Выдача информации разрешается нулевым сигналом на управляющих входах ВА (выдача адреса) и ВД (выдача данных). Длястыковки с аналогичными модулями с целью наращивания разрядности обрабатываемых слов ЦПЭ имеет входы для переноса в 0-ой разряд (С1) и для приема информации в 1-ый разряд при сдвиге вправо (СП1), а также выходы переноса из 1-го разряда (СО) и выдачи информации из 0-го разряда при сдвиге вправо (СП0). Выходы СО и СП0 являются трехстабильными и взаимно блокированы, вследствие чего допускается их монтажное объединение с целью формирования общего сигнала. Синхронизация работы всех узлов ЦПЭ осуществляется подачей синхроимпульсов с периодом не менее 100 нс на вход СИ.

Выбор операции, реализуемой ЦДЭ, осуществляется семиразрядным кодом F [0÷6]. Старшие три бита кода F4, F5, F6 (код F) задают одну из восьми групп операций, выполняемых в АЛУ, а младшие биты F0 - F3 (код R ) - одну из трех операций в каждой группе F . Кроме этого код R определяет источники и приемники operandов, используемых в операциях ЦПЭ.

Выбор источника operandов по входу А АЛУ осуществляется мультиплексором МА, управляемым кодом R . В качестве таких источников могут служить: один из РОНов, входная шина М, регистр-аккумулятор АС. Выбор operandов по входу В АЛУ осуществляется мультиплексором МВ, причем источниками operandов могут быть регистр-аккумулятор АС или входная шина В. Код маски

с шины K позволяет осуществлять поразрядное маскирование операндов по входу В АЛУ.

Результат операции из комбинационного АЛУ заносится в один из следующих приемников: регистр адреса РА, регистр-аккумулятор АС или в один из РОНов. Выбор РОНа для записи результата осуществляется кодом R.

В зависимости от кода R в каждой группе операций, задаваемых кодом F, выделяются три вида операций, отличающихся операндами и типом операций. В таблице 3.1 представлены коды F для групп операций АЛУ. В таблице 3.2 представлены коды R и соответствующие им операнды, используемые в операциях ЦПЭ. В таблице 3.3 приведен полный перечень операций, выполняемых в ЦПЭ для произвольной маски K, для K=00 и для K=11.

Таблица 3.1

Код F	F6	F5	F4
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Таблица 3.2

F3	F2	F1	F0	Код R	Регистр
0	0	0	0		R0
0	0	0	1		R1
0	0	1	0		R2
0	0	1	1		R3
0	1	0	0		R4
0	1	0	1		R5
0	1	1	0	1	R6
0	1	1	1	1	R7
1	0	0	0		R8
1	0	0	1		R9
1	1	0	0		T
1	1	0	1		AC
1	0	1	0	2	T
1	0	1	1	2	AC
1	1	1	0	3	T
1	1	1	1	3	AC

Таблица 3.3  
Операции, выполняемые в ЦПЭ

Код F	Код R	Операция (маска K – произвольная)	Операция (маска K = 11)	Операция (маска K = 00)
0	1	$R_n + (AC \& K) + CI \rightarrow R_n, AC$	$R_n + AC + CI \rightarrow R_n, AC$	$R_n + CI \rightarrow R_n, AC$
	2	$M + (AC \& K) + CI \rightarrow AT$	$M + AC + CI \rightarrow AT$	$M + CI \rightarrow AT$
	3	-	-	сдвиг AT вправо
1	1	$K \vee R_n \rightarrow PA;$ $R_n + K + CI \rightarrow R_n$	$11 \rightarrow PA$ $R_n - 1 + CI \rightarrow R_n$	$R_n \rightarrow PA$ $R_n + CI \rightarrow R_n$
	2	$K \vee M \rightarrow PA;$ $M + K + CI \rightarrow AT$	$11 \rightarrow PA$ $M - 1 + CI \rightarrow AT$	$M \rightarrow PA$ $M + CI \rightarrow AT$
	3	$(\bar{A}T \vee K) + (AT \vee K) + CI \rightarrow AT$	$AT - 1 + CI \rightarrow AT$	$\bar{AT} + CI \rightarrow AT$
2	1	$(AC \& K) - 1 + CI \rightarrow R_n$	$AC - 1 + CI \rightarrow R_n$	$CI - 1 \rightarrow R_n$
	2	$(AC \& K) - 1 + CI \rightarrow AT$	$AC - 1 + CI \rightarrow AT$	$CI - 1 \rightarrow AT$
	3	$(B \& K) - 1 + CI \rightarrow AT$	$B - 1 + CI \rightarrow AT$	$CI - 1 \rightarrow AT$
3	1	$R_n + (AC \& K) + CI \rightarrow R_n$	$R_n - AC + CI \rightarrow R_n$	$R_n + CI \rightarrow R_n$
	2	$M + (AC \& K) + CI \rightarrow AT$	$M + AC + CI \rightarrow AT$	$M + CI \rightarrow AT$
	3	$AT + (B \& K) + CI \rightarrow AT$	$B + AT + CI \rightarrow AT$	$AT + CI \rightarrow AT$
4	1	$R_n \& (AC \& K) \rightarrow R_n;$ $CI \vee (R_n \& AC \& K) \rightarrow C0$	$R_n \& AC \rightarrow R_n;$ $CI \vee (R_n \& AC) \rightarrow C0$	$0 \rightarrow R_n;$ $CI \rightarrow C0$
	2	$M \& (AC \& K) \rightarrow AT;$ $CI \vee (M \& AC \& K) \rightarrow C0$	$M \& AC \rightarrow AT;$ $CI \vee (M \& AC) \rightarrow C0$	$0 \rightarrow AT;$ $CI \rightarrow C0$
	3	$AT \& (B \& K) \rightarrow AT;$ $CI \vee (AT \& B \& K) \rightarrow C0$	$AT \& B \rightarrow AT;$ $CI \vee (AT \& B) \rightarrow C0$	$0 \rightarrow AT;$ $CI \rightarrow C0$
5	1	$K \& R_n \rightarrow R_n;$ $CI \vee (R_n \& K) \rightarrow C0$	$R_n \rightarrow R_n;$ $CI \vee R_n \rightarrow C0$	$0 \rightarrow R_n;$ $CI \rightarrow C0$
	2	$K \& M \rightarrow AT;$ $CI \vee (M \& K) \rightarrow C0$	$M \rightarrow AT;$ $CI \vee M \rightarrow C0$	$0 \rightarrow AT;$ $CI \rightarrow C0$
	3	$K \& AT \rightarrow AT,$ $CI \vee (AT \& K) \rightarrow C0$	$AT \rightarrow AT;$ $CI \vee AT \rightarrow C0$	$0 \rightarrow AT;$ $CI \rightarrow C0$
6	1	$R_n \vee (AC \& K) \rightarrow R_n;$ $CI \vee (AC \& K) \rightarrow C0$	$R_n \vee AC \rightarrow R_n;$ $CI \vee AC \rightarrow C0$	$R_n \rightarrow R_n;$ $CI \rightarrow C0$
	2	$M \vee (AC \& K) \rightarrow AT;$ $CI \vee (AC \& K) \rightarrow C0$	$M \vee AC \rightarrow AT;$ $CI \vee AC \rightarrow C0$	$M \rightarrow AT;$ $CI \rightarrow C0$
	3	$AT \vee (B \& K) \rightarrow AT;$ $CI \vee (B \& K) \rightarrow C0$	$AT \vee B \rightarrow AC;$ $CI \vee B \rightarrow C0$	$R_n \rightarrow R_n;$ $CI \rightarrow C0$
7	1	$R_n \oplus (AC \& K) \rightarrow R_n;$ $CI \vee (R_n \& AC \& K) \rightarrow C0$	$R_n \oplus AC \rightarrow R_n;$ $CI \vee (R_n \& AC) \rightarrow C0$	$\bar{R}_n \rightarrow R_n$ $CI \rightarrow C0$
	2	$M \oplus (AC \& K) \rightarrow AT;$ $CI \vee (M \& AC \& K) \rightarrow C0$	$M \oplus AC \rightarrow AT;$ $CI \vee (M \& AC) \rightarrow C0$	$\bar{M} \rightarrow AT$ $CI \rightarrow C0$
	3	$AT \oplus (B \& K) \rightarrow AT;$ $CI \vee (AT \& B \& K) \rightarrow C0$	$AT \oplus B \rightarrow AT;$ $CI \vee (AT \& B) \rightarrow C0$	$\bar{AT} \rightarrow AT$ $CI \rightarrow C0$

В таблице 3.3 во всех операциях при  $R=1$  в качестве операндов используется один из задаваемых кодом R регистров РОНО – РОН9, T, AC. При  $R = 2$ ,  $R=3$  вместо AT подставляется T или AC, также задаваемые кодом R. При F=0 и R=3 выполняется сдвиг AT вправо: AT[0] → СПО, AT[1] → AT[0], СПИ → AT[1]. Необходимо также помнить, что в таблице 3.3 все операции указаны для прямых значений В, М, К, СI, СПI. Поскольку на этих входах ЦПЭ стоят инверторы, коды В, М, К и переменные СI, СПI необходимо подавать на вход ЦПЭ инвертированными.

### **3.1.2. БИС БМУ для построения микропрограммных устройств управления**

БМУ предназначен для построения микропрограммных устройств управления с принудительной адресацией микрокоманд. Структурная схема БМУ представлена на рис.3.2. В состав БМУ включены следующие основные узлы:

- Триггеры ТФ, ТС, TZ для хранения логических условий, подаваемых на вход Ф и используемых для организации ветвлений в микропрограммах. Прием инверсий условий в ТФ осуществляется по единичному уровню СИ. Прием условия из ТФ в TZ (ТС) осуществляется по переднему фронту сигнала СИ и при наличии нулевого сигнала на управляющем входе УФ0, (УФ1) (см. табл. 3.4). Логическое условие, хранящееся в ТС (TZ), может быть выдано на трехстабильный выход Ф<sub>в</sub> при наличии разрешающего сигнала ОС и сигнала на входе УФ2 (УФ3) (см. табл. 3.4).
- Регистр команд РК, служащий для приема и хранения младших четырех бит кода операции команды процессора, подаваемого на входы K0 – K7. Прием в РК осуществляется по переднему фронту сигнала СИ и при наличии разрешающего сигнала от СФАМК. Три младших бита

РК могут быть выданы на выход РК через буферную схему БВК.

УА ОС РС УФ1  $\Phi$  УФ0 УФ2 УФ3 ЗМ К СИ

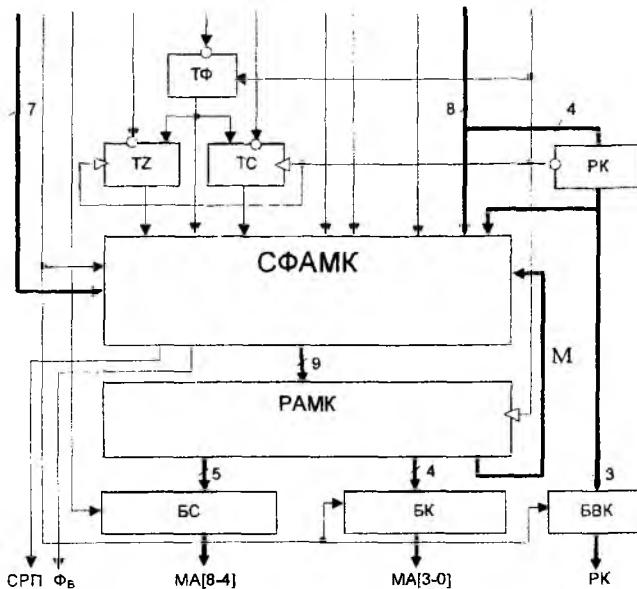


Рис. 3.2. Структурная схема БМУ

Таблица 3.4

Операции приема и выдачи логических условий

Тип операции	УФ3	УФ2	УФ1	УФ0	Операция
Прием логических условий	~	~	0	0	$TZ = T\Phi$ , $TC = T\Phi$
	~	~	0	1	$TZ = T\Phi$
	~	~	1	0	$TC = T\Phi$
	~	~	1	1	Хранение
Выдача логических условий	0	0	~	~	$\Phi_B = 0$
	0	1	~	~	$\Phi_B = TC$
	1	0	~	~	$\Phi_B = TZ$
	1	1	~	~	$\Phi_B = 1$

- Комбинационная схема формирования адреса очередной микрокоманды (СФАМК), работа которой описывается таблицей 3.5. На входы схемы подаются следующие коды:
  - код УА, являющийся адресной частью микрокоманды, задающей способ формирования адреса следующей микрокоманды, а также часть кода этого адреса;
  - код адреса текущей микрокоманды, хранящийся в регистре РАМК;
  - логические условия из триггеров ТФ, ТС, TZ , используемые для условных ветвлений в микропрограммах;
  - код команды К0- К7, задающий, как правило, начальный адрес (или его часть) микропрограммы, реализующей данную команду;
  - код из регистра РК, используемый для ветвлений по младшей части кода команды.

- Регистр адреса микрокоманд (РАМК) для хранения кода адреса, получаемого на выходах СФАМК. Адрес записывается в РАМК по переднему фронту сигнала синхронизации СИ и хранится в течение одного такта синхронизации.

Адрес микрокоманд, формируемый СФАМК, условно разбивается на две части и адресуемое ПЗУ МК представляется двухкоординатным (см. рис.3.3). Каждая ячейка ПЗУ МК адресуется адресом строки (старшие пять разрядов адреса МА8-МА4) и адресом колонки (младшие четыре разряда адреса МА3-МА0). На пересечении выбранных строк и колонки находится адресуемая микрокоманда.

При формировании девятиразрядного адреса очередной микрокоманды используется семиразрядный код адресной части микрокоманда УА6-УА0 и код текущего адреса микрокоманды из РАМК М8-М0. От текущего адреса

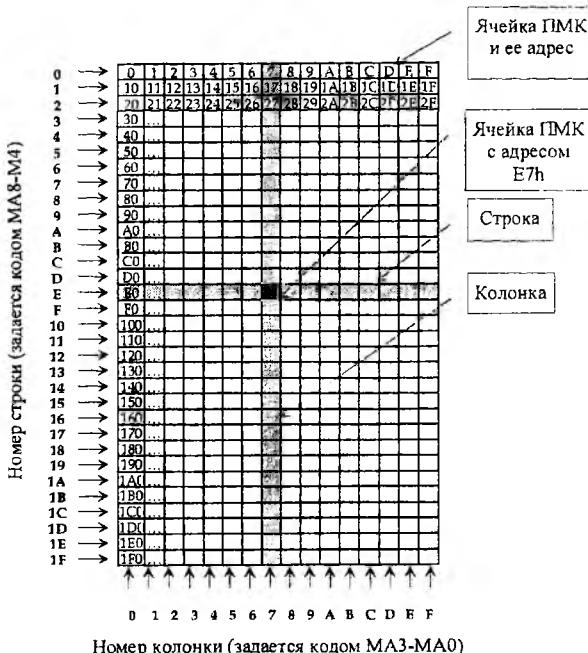


Рис. 3.3. Адресация микрокоманд

микрокоманды возможны переходы не к любому адресу, а к одному из некоторой группы адресов, задаваемых таблицей 3.5. Все возможные переходы разбиты на 3 группы: безусловные переходы, условные переходы по логическим условиям и переходы с ветвлением по коду операции команды.

При выполнении ветвления по содержимому регистра РК (см. табл. 3.5) автоматически вырабатывается сигнал приема в РК младшей тетрады кода команды К3-К0 (при СИ=0). Младшие три бита кода команды К2-К0 могут быть выданы через буфер БВК при выполнении перехода в текущей колонке в группе строк и при ОС=1. При выполнении перехода в нулевую строку в колонку 15

Таблица 3.5

## Способы вычисления адреса микрокоманды

Тип перехода	Название способа	Код управления адресом УА в текущей микрокоманде								Адрес строки следующей микрокоманды								Прос колонки следующей микрокоманды
		УA6	УA5	УA4	УA3	УA2	УA1	УA0	МA8	МA7	МA6	МA5	МA4	МA3	МA2	МA1	МA0	
Безуслов- ный переход	Переход в текущий колонке	0	0	УA4	УA3	УA2	УA1	УA0	УA4	УA3	УA2	УA1	УA0	М3	М2	М1	М0	
	Переход в нулевую строку	0	1	0	УA3	УA2	УA1	УA0	0	0	0	0	0	0	УA3	УA2	УA1	УA0
	Переход в текущей строке	0	1	1	УA3	УA2	УA1	УA0	M8	M7	M6	M5	M4	УA3	УA2	УA1	УA0	
	Переход в текущий колонке в группе строк выхода из РК	1	1	1	0	УA2	УA1	УA0	M8	M7	УA2	УA1	УA0	М3	М2	М1	М0	
Ветвление по признаку	Ветвление по ГФ	1	0	0	УA3	УA2	УA1	УA0	M8	УA3	УA2	УA1	УA0	М3	0	1	ГФ	
	Ветвление по ТС	1	0	1	0	УA2	УA1	УA0	M8	M7	УA2	УA1	УA0	М3	0	1	ТС	
	Ветвление по ГZ	1	0	1	1	УA2	УA1	УA0	M8	M7	УA2	УA1	УA0	М3	0	1	TZ	
Ветвление по коду операции	Ветвление по содержимому регистра РК	1	1	0	0	УA2	УA1	УA0	M8	M7	УA2	УA1	УA0	РК3	РК2	РК1	РК0	
	Ветвление по старшим битам регистра РК	1	1	0	1	УA2	УA1	УA0	M8	M7	УA2	УA1	УA0	0	1	РК3	РК2	
	Ветвление по младшим битам регистра РК	1	1	1	1	1	УA1	УA0	M8	M7	1	УA1	УA0	1	1	РК1	РК0	
	Ветвление по старшим битам кода команды K7-K4 и загрузка РК	1	1	1	1	0	УA1	УA0	M8	M7	M6	УA1	УA0	K7	K6	K5	K4	

в течение одного такта формируется сигнал СРП=1, который может быть использован для организации прерываний.

Помимо рассмотренных в таблице 3.5 способов формирования адреса в РАМК, возможна непосредственная загрузка кода К7-К0 в РАМК по сигналу ЗМ (может быть использована для начальной установки РАМК при запуске МУУ).

Выходные буферы БК (буфер колонки) и БС (буфер строк), имеют трехстабильные выходы. БК открыт при ОС=1, БС открыт при ОС=1, РС=1. Во всех остальных случаях выходы БС, БК находятся в отключенном состоянии (Z - состояние).

### **3.1.3. Совместное использование блоков БМУ и ЦПЭ для построения процессора**

На рис. 3.4 представлен один из возможных вариантов структурной организации процессора, состоящего из БМУ, n модулей ЦПЭ и памяти микрокоманд (ПМК), связанных между собой следующими линиями и шинами:

- для организации цепей переноса и сдвига в АЛУ выход СО и вход СПI каждого ЦПЭ соединяется со входом СI и выходом СПО соседнего старшего ЦПЭ.
- входы F, ВА, ВД, СИ всех ЦПЭ соединяются для подачи общих сигналов и кодов, а остальные одноименные выводы ЦПЭ объединяются в соответствующие 2n - разрядные шины.
- вход Ф БМУ соединяется с объединенными выходами СО старшего ЦПЭ, СПО младшего ЦПЭ для подачи в БМУ логических условий, формируемых выполняемыми в ЦПЭ операциями;

- выход ФВ БМУ соединен со входами С1 младшего ЦПЭ, СП1 старшего ЦПЭ для подачи в ЦПЭ фиксированных значений сигналов переноса или сдвига, а также значений этих сигналов, полученных в предыдущих тактах;

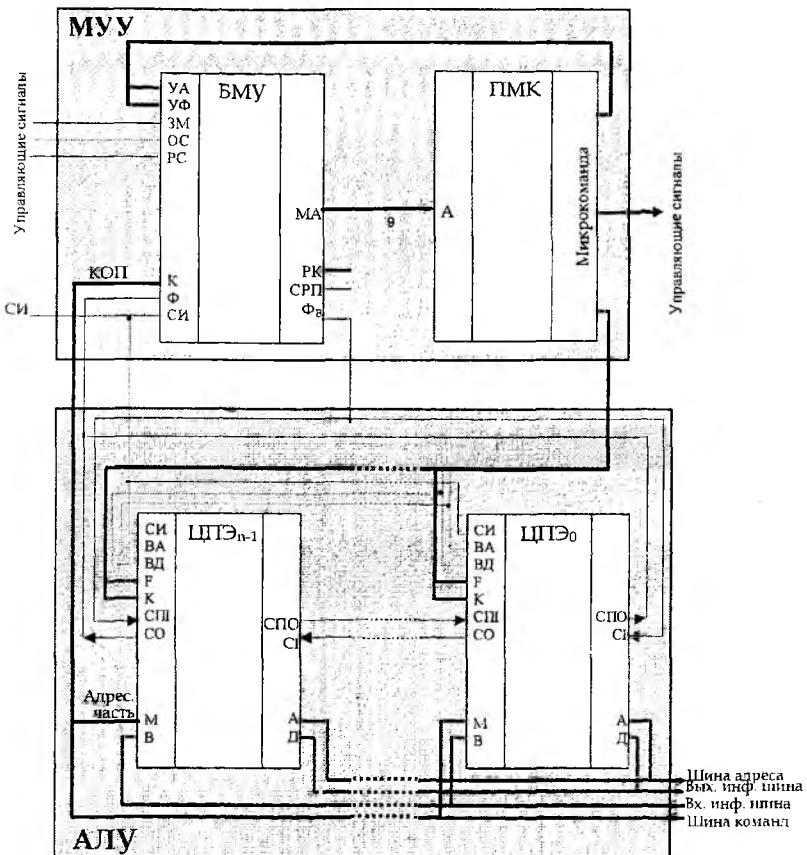


Рис. 3.4. Объединение БМУ и n ЦПЭ для построения 2n-разрядного процессора

- на входы F, ВД, ВА, К ЦПЭ подаются коды из полей микрокоманды или константы.

Информация, обрабатываемая в АЛУ по микропрограммам МУУ, подается в АЛУ по шинам В, М. Как правило, шина М используется для приема команд из памяти программ: адресная часть команды подается в АЛУ, а код команды - на входы К0-К7 БМУ. Для обмена операндами с памятью и другими устройствами используются шина адреса А, выходная Д и входная В информационные шины.

### **3.2. Серия БИС на основе четырехразрядных ЦПЭ (серия K1804)**

В состав данной серии входят несколько типов БИС ЦПЭ И БМУ. В методических указаниях [4] подробно описаны используемые в проекте ЦПЭ К1804ВС1 и БМУ К1804ВУ1. Ниже приводятся краткие описания ЦПЭ К1804ВС2 и БМУ К1804ВУ2, также используемых в курсовом проекте.

#### **3.2.1. БИС ЦПЭ К1804ВС2 для построения АЛУ**

ЦПЭ представляет собой четырехразрядную процессорную секцию, выполняющую функции арифметической и логической обработки четырехразрядных операндов, их временного хранения и формирования различных признаков результатов. Структурная схема ЦПЭ приведена на рис.3.5. В состав ЦПЭ включены:

- четырехразрядное комбинационное арифметико-логическое устройство ALU,
- мультиплексоры входных operandов: MR на входе R и MS на входе S ALU,
- два регистра-защелки RA, RB для временного хранения входных operandов ALU,

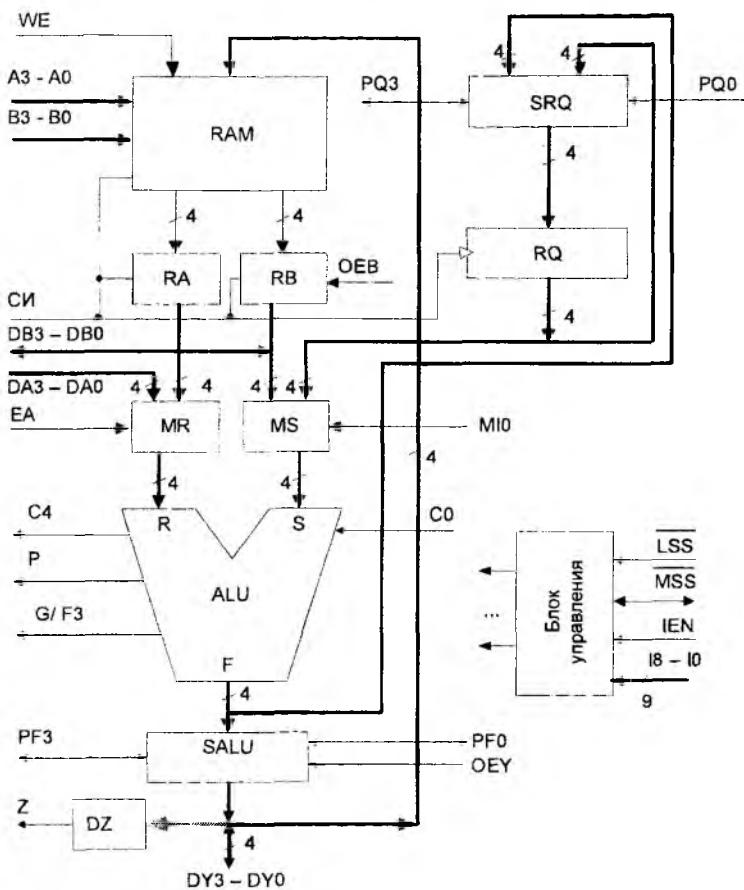


Рис. 3.5. Четырехразрядная процессорная секция

- блок регистров (RAM), включающий 16 четырехразрядных регистров общего назначения R0 – R15, адресуемых кодами A3-A0, B3-B0,
- рабочий регистр RQ,
- два сдвигателя: SALU - сдвигатель результата операции на выходе ALU и SRQ - сдвигатель регистра RQ,
- схема DZ проверки операндов на 0,
- блок управления для формирования микроинструкций, реализующих заданные операции.

Для обмена операндами с другими устройствами используются три четырехразрядные шины: двунаправленные шины DB, DY и входная шина DA.

При объединении ЦПЭ с целью увеличения разрядности обработки в данном типе ЦПЭ в отличие от ЦПЭ K1804BC1 имеются специальные входы задания позиции ЦПЭ: старшая, средняя, младшая. С этой целью используются управляющие сигналы LSS, MSS (табл. 3.6). При LSS = 0 вывод MSS является выходом для выдачи признака записи в RAM.

Таблица 3.6  
Задание позиции ЦПЭ

<u>MSS</u>	<u>LSS</u>	Позиция ЦПЭ
Выход признака записи в RAM	0	младшая
1	1	средняя
0	1	старшая

Управляющие сигналы EA, MIO, OEB определяют выбор источников operandов для ALU (табл. 3.7) Код микроинструкции I8-I0 определяет выполняемую в ЦПЭ операцию. При I0=1 код I4 - I1 определяет набор базовых арифметических и логических операций, приведенных в табл. 3.8. Результаты могут быть выданы на выходные

Таблица 3.7

## Источники operandов ALU

Управляющий код			Источники operandов	
EA	MIO	OEB	на входе R	на входе S
0	0	0	RA	RB
0	0	1	RA	DB
0	1	~	RA	RQ
1	0	0	DA	RB
1	0	1	DA	DB
1	1	~	DA	RQ

Таблица 3.8

## Функции, выполняемые в АЛУ

Управляющий код				Функция при I0 = 1
I4	I3	I2	I1	
0	0	0	0	1111
0	0	0	1	S - R - I + C0
0	0	1	0	R - S - I + C0
0	0	1	1	R + S + C0
0	1	0	0	S + C0
0	1	0	1	$\bar{S} + C0$
0	1	1	0	R + C0
0	1	1	1	$\bar{R} + C0$
1	0	0	0	0000
1	0	0	1	$\bar{R} \& S$
1	0	1	0	$\bar{R} \oplus S$
1	0	1	1	$R \oplus S$
1	1	0	0	R & S
1	1	0	1	$\bar{R} \& S$
1	1	1	0	$\bar{R} \vee S$
1	1	1	1	$R \vee S$

Таблица 3.9

## Приемники результатов из АЛУ

Управляющий код				Сдвиг результата на выходе Y АЛУ							Сдвиг регистра RQ			
18	17	16	15	Тип сдвига	PF3 ст/ост	DY3 ст/ост	DY2 ст/ост	DY1	DY0	PF0	MSS	Тип сдвига	PQ3	PQ0
0	0	0	0	САЛ	bx/bx	F3/ PF3	PF3/F3	F2	F1	F0	0	-	z	z
0	0	0	1	СЛП	bx/bx	PF3/PF3	F3	F2	F1	F0	0	-	z	z
0	0	1	0	САЛ	bx/bx	F3/ PF3	PF3/F3	F2	F1	F0	0	СЛП	bx	RQ0
0	0	1	1	СЛП	bx/bx	PF3/PF3	F3	F2	F1	F0	0	СЛП	bx	RQ0
0	1	0	0	DY = F	bx/bx	F3	F2	F1	F0	P	0	-	z	z
0	1	0	1	DY = F	bx/bx	F3	F2	F1	F0	P	1	СЛП	z	RQ0
0	1	1	0	DY = F	bx/bx	F3	F2	F1	F0	P	1	RQ=F	z	z
0	1	1	1	DY = F	bx/bx	F3	F2	F1	F0	P	0	RQ=F	z	z
1	0	0	0	САЛ	F2/F3	F3/F2	F1	F0	PF0	bx	0	-	z	z
1	0	0	1	СЛЛ	F3/F3	F3/F2	F1	F0	PF0	bx	0	-	z	z
1	0	1	0	САЛ	F2/F3	F3/F2	F1	F0	PF0	bx	0	СЛЛ	RQ3	bx
1	0	1	1	СЛЛ	F3/F3	F3/F2	F1	F0	PF0	bx	0	СЛЛ	RQ3	bx
1	1	0	0	DY = F	F3	F3	F2	F1	F0	P	1	-	z	z
1	1	0	1	DY = F	F3	F3	F2	F1	F0	P	1	СЛЛ	RQ3	bx
1	1	1	0	PF0→DY	PF0	PF0	PF0	PF0	PF0	PF0	0	-	z	z
1	1	1	1	DY = F	F3	F3	F2	F1	F0	P	0	-	z	z

Пояснения:

- ст – для ЦПЭ в старшей позиции
- ост – для ЦПЭ в средней и младшей позициях
- bx – вывод является входом для приема произвольного значения
- $P = F0 \oplus F1 \oplus F2 \oplus F3 \oplus PF3$

шины DY, DB и сохранены в RAM или RQ со сдвигом или без него. Предусмотрено два типа сдвига: арифметический влево (САЛ) и вправо (САР), логический влево (СЛЛ) и вправо (СЛР). В табл. 3.9 приведены способы выбора приемников результатов, определяемые кодом I8-I5. При этом на входах I4-I1 должен быть задан код I4 -I1 ≠ 0000 и IEN=0

При I0=0, I4 - I1= 0000, IEN=0 в ЦПЭ выполняются специальные функции, упрощающие реализацию слож-

ных арифметических операций: умножения, деления, нормализации и др. (табл. 3.10). Выбор специальных функций определяется кодом I8 – I5. Более подробное описание специальных функций и их использования можно найти в [9].

Таблица 3.10

Специальные функции ALU

Управляющий код				Операции (при I0 = 0, I4-I1= 0000)
I8	I7	I6	I5	
0	0	0	0	Умножение без знака
0	0	1	0	Умножение в дополнительном коде
0	1	0	0	Инкремент
0	1	0	1	Преобразование числа со знаком в дополнительный код
0	1	1	0	Последний цикл умножения в дополнительном коде
0	1	1	1	Нормализации обычной длины
1	0	1	0	Нормализации двойной длины
1	1	0	0	Деление в дополнительном коде
1	1	1	0	Деление в дополнительном коде (коррекция)
Остальные коды I8-I5 запрещены				

### 3.2.2. БИС БМУ К1804ВУ2 для построения микропрограммных устройств управления

Четырехразрядная секция БМУ К1804ВУ2 предназначена для построения микропрограммных управляемых автоматов. Основной функцией блока является формирование адресов микрокоманд с учетом признаков ветвления, кода выполняемой операции (КОП) и других сигналов. Структурная схема секции БМУ приведена на рис. 3.6.

Адрес микрокоманды подается на внешние выходы Y3 – Y0 через мультиплексор от одного из четырех источников. Управление мультиплексором осуществляется двухразрядным кодом S1 – S0 в соответствии с табл. 3.11. В качестве источника адреса могут быть использованы:

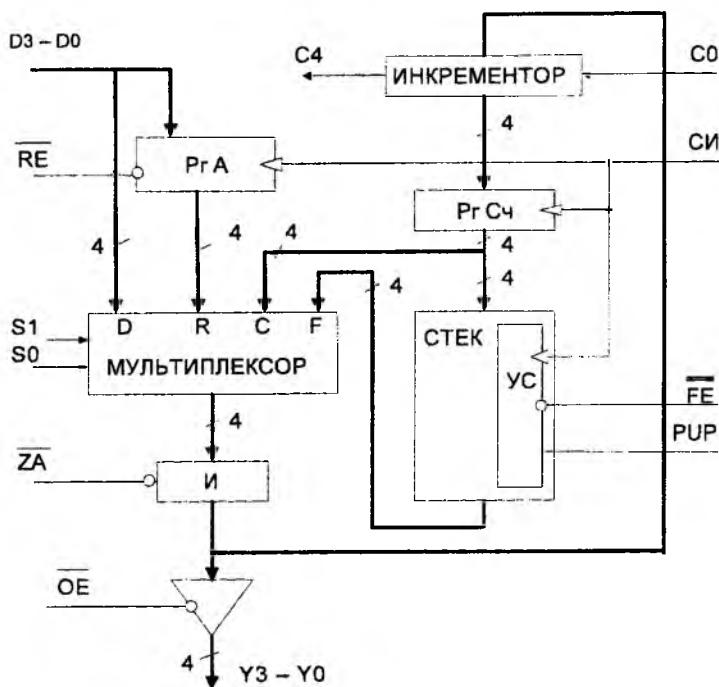


Рис.3.6. Структурная схема БМУ K1804BY2

- Входная шина D3 – D0, предназначенная для приема принудительного адреса.
- Регистр адреса РГА, который может быть использован для ветвления по коду операции (КОП). Регистр адреса может принимать информацию с шины D3 – D0 по переднему фронту СИ при  $\overline{RE} = 0$ .
- Счетчик микрокоманд, предназначенный для организации естественной адресации микрокоманд. Состоит из регистра РгСч и инкрементора. Текущий адрес микрокоманды подается в регистр РгСч через инкрементор, суммирующий его значение с входным переносом С0. Значение переноса из инкрементора выдается на внешний выход С4. Запись в РгСч осуществляется по переднему фронту СИ.
- Стек, предназначенный для организации переходов с возвратом (микроподпрограмм). Стек состоит из четырех ячеек, адресуемых указателем стека УС. Операции записи и считывания возможны только для той ячейки, на которую указывает указатель стека.

Все операции со стеком реализуются через схему записи/считывания, находящуюся внутри стека. Управление стеком приведено в табл. 3.12. При записи в стек ( $\overline{FE} = 0$  и  $PUP = 1$ ) вначале содержимое указателя стека увеличивается на единицу, затем происходит запись в стек из счетчика микрокоманд.

При чтении из стека с выталкиванием ( $\overline{FE} = 0$  и  $PUP = 0$ ) происходит сначала чтение из стека, затем уменьшение содержимого указателя стека на единицу. Чтение из стека без выталкивания ( $\overline{FE} = 1$ ) не изменяет указатель стека и допускает повторные циклы чтения.

Таблица 3.11

## Управление мультиплексором БМУ

S1	S0	Источник адреса
0	0	счетчик микрокоманд РгСч
0	1	регистр адреса РгА
1	0	стек
1	1	входная шина D

Таблица 3.12

## Управление стеком БМУ

FE	PUP	Операция в стеке
1	~	нет операции
0	0	чтение из стека, УС = УС – 1
0	1	УС = УС + 1, запись в стек

Таблица 3.13

## Управление выдачей адреса из БМУ

OE	ZA	Значение на выходе $Y_i$
1	~	Z-состояние
0	0	константа «0»
0	1	с выхода мультиплексора

Указатель стека представляет собой двухразрядный реверсивный счетчик, операции увеличения и уменьшения в котором осуществляются по модулю четыре. Таким образом, отпадает необходимость в его начальной установке.

Адрес, полученный от одного из четырех источников, с выхода мультиплексора поступает на схему «И», которая обнуляет адрес (при  $\overline{ZA} = 0$ ) или передает его без изменений (при  $\overline{ZA} = 1$ ). С выхода схемы «И» адрес поступает в счетчик микрокоманд и через буфер с тремя состоя-

ниями на внешние выходы Y3 – Y0. Буфер переводится в третье состояние при  $\overline{OE} = 1$ . Управление выдачей адреса из БМУ поясняет табл. 3.13.

Наращивание разрядности БМУ осуществляется с целью увеличения разрядности адреса и, следовательно, объема памяти микропрограмм. Выполняется аналогично наращиванию разрядности ЦПЭ путем объединения  $n$  секций БМУ в единый блок. При этом выходы переноса C4 каждой секции БМУ (кроме самой старшей) соединяются с входами C0 соседней старшей секции БМУ. На вход переноса C0 в младшую секцию БМУ для правильной организации естественной адресации подается константа «1». Входы D3 – D0 и выходы Y3 – Y0 всех секций БМУ становятся  $4n$ -разрядными шинами. Остальные одноименные входы всех секций БМУ соединяются.

#### **4. Схема синхронизации и начальной установки**

Основное назначение схемы синхронизации и начальной установки (ССНУ) – формирование сигнала начальной установки (НУ) и синхросигнала (СИ) для тактирования работы всех устройств процессора. ССНУ включает генератор импульсов (ГИ) и схему начальной установки (СНУ) (рис. 3.7).

Генератор построен по стандартной схеме на двух инверторах с использованием кварцевого резонатора G1 для стабилизации частоты ( $R1=R2 = 560\text{ом}$ ,  $C1=1300\text{пФ}$ ).

Схема начальной установки формирует сигнал НУ по нажатию кнопки S1 или при включении питания. В состав схемы входят два триггера: T1 – для устранения “дребезга” контактов кнопки S1, T2 – для привязки сигнала НУ к синхросерии СИ. Временная работа ССНУ приведена на рис. 3.8.

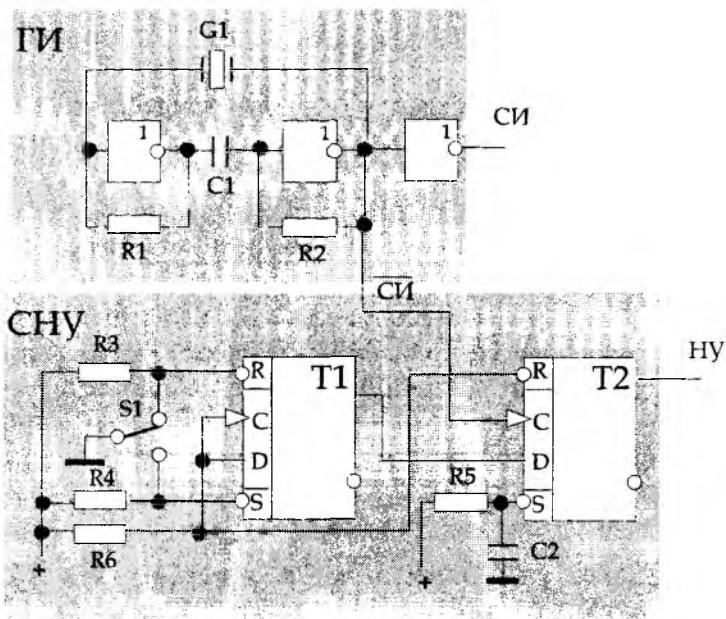


Рис.3.7. Схема синхронизации и начальной установки

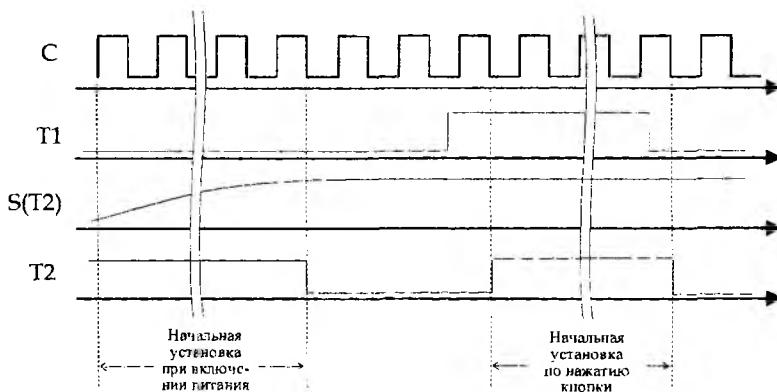


Рис.3.8. Временная диаграмма работы ССНУ

## **5. Требования к оформлению курсового проекта**

### **5.1. Состав документации**

Комплект документации на курсовой проект состоит из следующих обязательных документов:

- титульный лист,
- ведомость курсового проекта,
- задание на курсовой проект,
- пояснительная записка,
- схема электрическая функциональная (по заданию преподавателя),
- схема электрическая принципиальная,
- перечень элементов.

Оформления перечисленных документов должно соответствовать требованиям Единой Системы Конструкторской Документации (ЕСКД) [10]. Схема электрическая принципиальная выполняется в обязательном порядке в системе P-CAD.

### **5.2. Содержание пояснительной записи**

Пояснительная записка должна состоять из следующих разделов.

1. Программистская модель процессора и описание системы команд.
  - *регистровая модель процессора (форматы и адреса всех программно-доступных регистров).*
  - *форматы используемых операндов.*
  - *форматы и описания всех команд процессора, алгоритмы исполнения сложных команд,*

Примечание: в том случае, если система команд не задана, должен быть приведен ее выбор, включающий:

- выбор множества операций (команд), определение их форматов и способа кодирования на основе анализа заданных алгоритмов, исполняемых процессором,
  - определение форматов данных,
  - выбор множества программно доступных элементов,
  - разработка алгоритмов исполнения выбранных команд.
- 2. Разработка структурной и функциональной схем.
  - детальная проработка структурной или функциональной схем базовых блоков, входящих в разрабатываемое устройство, и их описание:
    - АЛУ,
    - устройство управления,
    - блок прерывания,
    - устройство сопряжения с памятью (памятью программ и памятью данных, если они раздельные) и другими устройствами (если они имеются), включая схемы адресации и арбитража,
    - схема синхронизации и начальной установки;
  - описание общей структурной или функциональной схемы включающей все разработанные блоки и их соединения.
- 3. Разработка электрической принципиальной схемы и перечня элементов.
  - реализация разработанной структурной или функциональной схемы на заданной элементной базе.
- 4. Временные диаграммы работы.

- разработка временных диаграмм работы устройства в основных режимах (начальная установка, чтение из оперативной памяти и др.),
  - расчет временных параметров устройства.
- 5. Разработка блок-схем и кодирование микрограмм машинного цикла.
  - разработка и описание подробных блок-схем микропрограммы машинного цикла с учетом выбранных схемных решений (начальная установка, чтение и дешифрация команд, исполнение команд),
  - микропрограммно-реализуемые процедуры системы прерывания.
  - описание формата микрокоманд и кодирование микропрограмм машинного цикла.
  - оценка длительности выполнения операций (команд) устройства.
- 6. Разработка программного обеспечения.
  - разработка блок-схем и программ, реализующих процедуры прерывания, управления или другие процедуры, заданные в проекте.

## **6. Список литературы**

1. Ж. Мишель. Программируемые контроллеры: архитектура и применение. Пер. с франц., М.: Машиностроение, 1992.
2. Мамченко А. Е. Адресация программно-доступных элементов микропроцессорных систем и распределение адресов при проектировании микросистем. Метод. ук. М.: МИИТ 1996.
3. Мамченко А.Е. Организация микропрограммных управляемых устройств с программируемой (хранимой в памяти) логикой. Метод. ук. М.: МИИТ, 1984.
4. Шамров М.И., Тельнов Г.Г. Микроархитектура процессоров для информационных систем на железнодорожном транспорте. Метод. пособие. М.: МИИТ, 2004.
5. Микропроцессоры и микроЭВМ в системах автоматического управления; Справочник. Под общ. ред. С.Т. Хвоща. - Л.: Машиностроение. Ленингр. отд., 1987.
6. Бесков Б. А., Шамров М. И. Микропрограммный микропроцессор. В 3-х частях, Метод. ук. М.: МИИТ, 1988 .
7. Шамров М.И., Тельнов Г.Г. Организация устройств на базе процессоров малой разрядности для информационных систем на железнодорожном транспорте. Метод. ук. М.: МИИТ, 2007.
8. Шамров М.И., Тельнов Г.Г. Учебная микроЭВМ УМПК-86. Часть 1 и 2. Метод. ук. М.: МИИТ, 1993-1994.
9. Проектирование цифровых систем на комплектах микропрограммируемых БИС. Под ред. В.Г. Колесникова. М.: Радио и связь, 1984.
- 10.Разработка и оформление конструкторской документации радиоэлектронной аппаратуры. Справочник, под редакцией Э.Т. Романычевой, М.: Радио и связь, 1989.

## СОДЕРЖАНИЕ

<b>Введение .....</b>	<b>3</b>
<b>1. Программируемый логический контроллер на многокристальном микропроцессоре .....</b>	<b>4</b>
1.1. Общие сведения о проектируемом устройстве .....	4
1.2. Задание на проектирование ПЛК .....	5
1.3. Рекомендации по разработке ПЛК. ....	7
<b>2. Сопроцессор с архитектурой RISC .....</b>	<b>24</b>
2.1. Общие сведения о проектируемом устройстве .....	24
2.2. Программистская модель СП .....	27
2.3. Структурная организация СП и БООП 2.3.1. Основные блоки СП и БООП и их подключение .....	29
2.3.2. Организация системы прерываний СП .....	29
2.3.3. Организация арбитража ОП .....	33
2.4. Варианты заданий на проектирование СП .....	34
<b>3. Элементная база для построения процессоров .....</b>	<b>38</b>
3.1. Серия БИС на основе двухразрядных ЦПЭ (серия К589) .....	38
3.1.1. БИС ЦПЭ для построения АЛУ .....	38
3.1.2. БИС БМУ для построения микропрограммных устройств управления.....	43
3.1.3. Совместное использование блоков БМУ и ЦЦЭ для построения процессора .....	48
3.2. Серия БИС на основе четырехразрядных ЦПЭ (серия К1804) .....	50
3.2.1. БИС ЦПЭ К1804ВС2 для построения АЛУ ....	50
3.2.2. БИС БМУ К1804ВУ2 для построения микропрограммных устройств управления	55
<b>4. Схема синхронизации и начальной установки ...</b>	<b>59</b>
<b>5. Требования к оформлению курсового проекта ...</b>	<b>61</b>
5.1. Состав документации .....	61
5.2. Содержание пояснительной записи .....	61
<b>6. Список литературы .....</b>	<b>64</b>

Учебно-методическое издание

Мамченко Александр Евгеньевич  
Шамров Михаил Иванович

Организация, схемотехника и микропрограммирование  
процессоров ЭВМ

Методические указания к курсовому проектированию  
по дисциплине «Организация ЭВМ и систем»

---

Сдано в набор

Подписано к печати

Формат 60×84 / 16

Усл. л. л.

Заказ

Изд № 273-12

Тираж 100экз.

---

150048, г. Ярославль, Московский пр-т, д. 151  
Типография Ярославского филиала МИИТ