

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Изучение архитектуры микроЭВМ

Методические указания

к лабораторным работам

Пенза, 1985

Даны методические указания к лабораторным работам по изучению структуры, форматов данных и команд, системы команд микроЭВМ "Электроника 60", а также по составлению и отладке программ для этой микроЭВМ.

Методические указания подготовлены на кафедре "Вычислительная техника" и предназначены для студентов специальностей 0608, 0606, 0640, 0642, 0646, 0647, 0648, 0701, изучающих основы микропроцессорной техники.

Ил.30, табл.17, библиогр.5 назв.

Составители: Н.Н. Коннов, канд. техн. наук, доц.;

В.Г. Пучков, канд. техн. наук, доц.;

Б.Д. Шашков, канд. техн. наук, доц.;

Под редакцией Н.П. Вашкевича, д-ра техн. наук, проф.

1. Цель лабораторных работ

Целью настоящих лабораторных работ является изучение архитектуры, принципов действия микроЭВМ "Электроника 60", а также получение навыков программирования и работы оператора микроЭВМ этого типа.

2. Описание лабораторной установки

2.1. Общая характеристика микроЭВМ

В качестве лабораторной установки используется микроЭВМ "Электроника 60М" или другие микроЭВМ, программно совместимые с указанной, с терминалом: электрической пишущей машинкой (ЭПМ) "Консул" или алфавитно-цифровым дисплеем.

МикроЭВМ "Электроника 60" в её различных модификациях (М, Т, В) и программно совместимые с ней микроЭВМ "Электроника НЦ-80", "Электроника С-41" и другие относятся к широко распространённому классу микроЭВМ, ориентированных на решение задач управления различными процессами и объектами. ЭВМ "Электроника 60" программно совместима с основным типом отечественных мини-ЭВМ (СМ-3, СМ-4, "Электроника 125"). Обобщённая структурная схема микроЭВМ представлена на рис.1. МикроЭВМ состоит из отдельных модулей: центрального процессора(ЦП), запоминающих устройств(ЗУ), устройств управления (контроллеров) периферийного оборудования, связь между которыми осуществляется через единый канал. Способ подключения (интерфейс) модулей к каналу и порядок обмена информацией между ними через канал стандартны, что позволяет обращаться ЦП к ЗУ и периферийным устройствам одинаковым образом.

2.2. Структура центрального процессора

Центральный процессор микроЭВМ извлекает из оперативного

запоминающего устройства (ОЗУ) команды программы, дешифрирует их и выполняет заданные ими действия над данными (операциями), которые могут находиться в самом ЦП или извлекаться им из ОЗУ или периферийных устройств.

Центральный процессор микроЭВМ "Электроника 60М" реализуется на нескольких больших интегральных схемах (БИС) серий 581, размещённых на одной печатной плате. В других типах аналогичных микроЭВМ ЦП строится на БИС серий 1801 или 588.

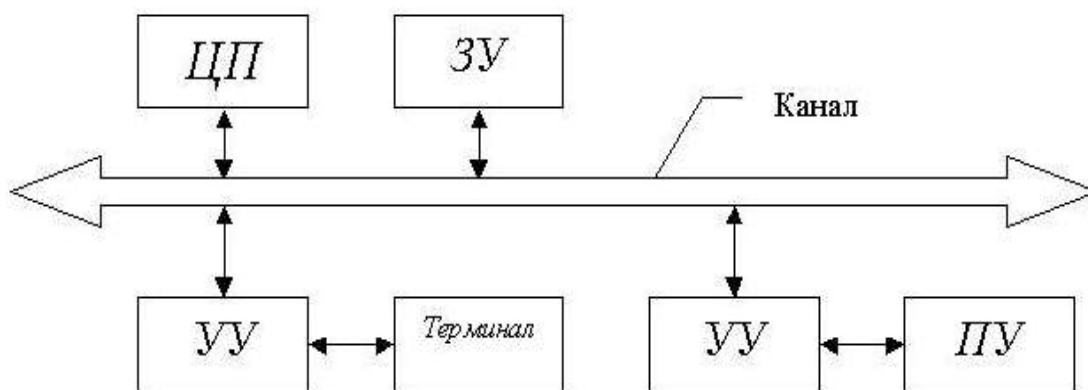


Рис.1

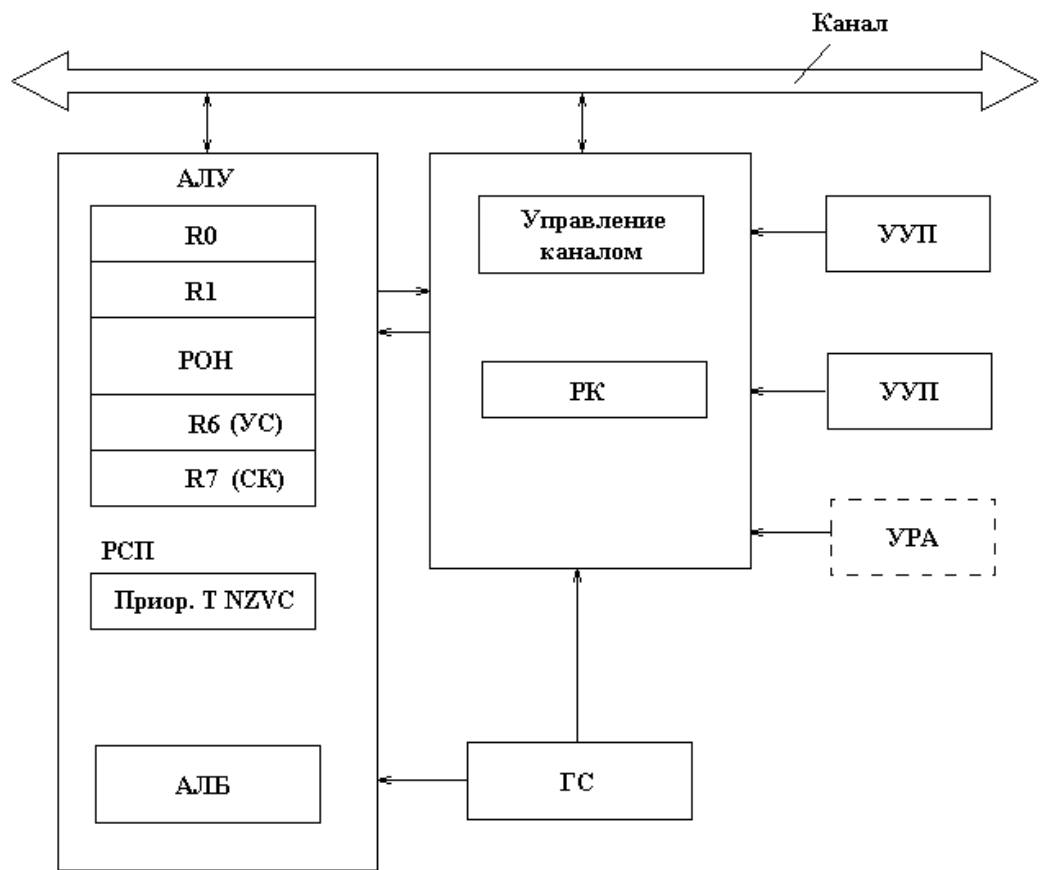


Рис.2

Укрупнённая структурная схема ЦП показана на рис.2. ЦП содержит:

Арифметико-логическое устройство (АЛУ), предназначенное для выполнения арифметических и логических операций над данными и вычисления адресов команд и данных. АЛУ включает:

1) блок регистров общего назначения (РОН) $R0 \div R7$. Шестнадцатиразрядные РОН являются программно доступными и могут использоваться для хранения операндов или формирования их адресов. Регистры $R6$ и $R7$ имеют, кроме того, специальное назначение: $R6$ используется как указатель стека (УС) (см. рис.2), а $R7$ – как счетчик команд(СК), формирующий адрес очередной команды;

2) регистр состояния процессора (РСП), на котором после выполнения каждой операции формируется слово состояния процессора (ССП). Разряды ССП кодируют признаки результата выполнения команды (перенос C , нуль Z , знак N , переполнение V , признак внутреннего прерывания T , а также приоритет процессора по отношению к другим устройствам, подключённым к каналу);

3) арифметико-логический блок (АЛБ), который непосредственно выполняет различные арифметико-логические операции над содержимым РОН или данными, поступающими в АЛУ из канала.

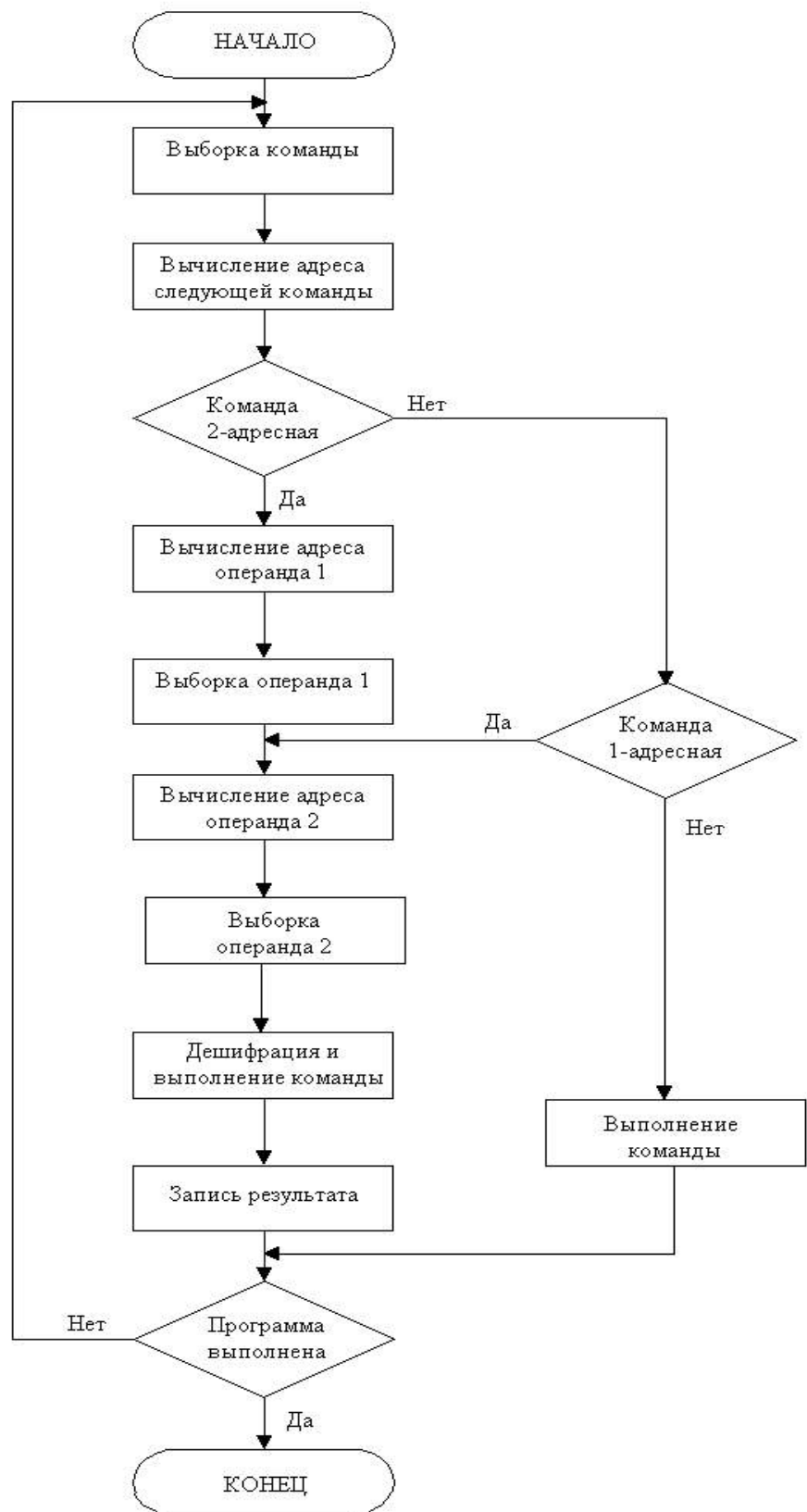
Устройство управления выполнения операций (УУ), предназначенное для управления работой АЛУ и канала. УУ содержит шестнадцатиразрядный регистр команд (РК), на который принимается очередная команда, схемы её дешифраций, анализа и выработки управляющих сигналов в АЛУ и канал, которые управляет непосредственно реализацией команды в АЛУ.

Устройства управляющей памяти (УУП), представляющие собой постоянные ЗУ, хранящие микропрограммы выполнения операций в ЦП и связи с пультом и терминалом.

Генератор синхронизации (ГС), осуществляющий тактирование остальных устройств ЦП.

Устройство расширенной арифметики(УРА), предназначенное для ускоренного аппаратного выполнения арифметических операций типа умножение-деление и действий над числами, представленными в формате с плавающей запятой. УРА входит в состав ЦП лишь некоторых типов микроЭВМ.

Обобщенный алгоритм выполнения операций, включающие следующие этапы (фазы): извлечение очередной команды и формирование адреса следующей, вычисление адресов и извлечение операндов, выполнение команды – показан на рис.3:



2.3 Форматы данных

МикроЭВМ оперируют над данными, представленными в двух основных форматах: шестнадцатиразрядное слово или восьмиразрядный байт (рис. 4). Данные рассматриваются как целые числа, старший разряд – знаковый (0 – плюс, 1- минус), отрицательные числа кодируются в дополнительном коде. Диапазоны представления чисел:

а) слово – от -32768 (100000_8) до $+32767$ (077777_8)

б) байт – от -128 (200_8) до $+127$ (177_8)



Рис.4

Следует отметить, что при выполнении некоторых команд микроЭВМ данные рассматриваются как целые положительные числа (т.е. без учёта знака).

2.4 Слово состояния процессора

Термин "слово состояния процессора" означает код, в котором отображается текущее его состояние путём формирования ряда признаков.

Формат ССП показан на рис.5. Признаки N, Z, V, C содержат информацию о результате последней выполненной команды, их значения могут программно анализироваться командами ветвления по условиям. Признак используется для прерывания программы при отладке. Разряд ССП задаёт приоритет ЦП по отношению к периферийным устройствам, если этот разряд равен 1, то прерывание текущей программы от периферийных устройств микроЭВМ невозможен.

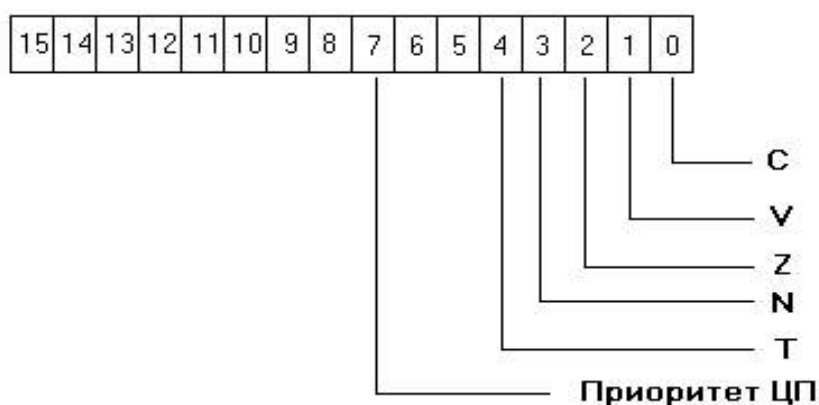


Рис.5

2.5 Организация памяти

За командами и данными ЦП обращается к ячейкам микроЭВМ, реализуемой в виде модулей оперативных или постоянных запоминающих устройств, подключённых к каналу.

Для обращения к запоминающим устройствам ЦП формирует и передаёт в канал шестнадцатиразрядный адрес искомой информации, т.е. адресное пространство ЦП состоит от 000000_8 до 177777_8 , т.е. равно 64 К ($K=10^{12}=1024$). Поскольку данные могут содержать 8 или 16 разрядов, то адрес указывает либо на байт, либо на слово, образованное двумя соседними байтами, в последнем случае адрес должен быть чётным (рис. 6). При

обращении к памяти необходимо указывать формат данных (слово или байт).

Обращение к периферийным устройствам выполняется так же, как к ячейкам ЗУ, для чего старшие адреса с 160000_8 до 177777_8 отводятся для нумерации регистров периферийных устройств, а не ячеек памяти. Максимальная емкость ЗУ в микроЭВМ составляет 28К слов или 56 Кбайт.



Рис.6

2.6 Способы адресации операндов

Для указания местоположения операндов в ЗУ в форматах команд выделяются специальные адресные поля. В зависимости от количества адресных полей в формате команды разделяются на одно- и двухадресные. Структура адресного поля, используемого для адресации операндов, приведена на рис. 7. Для адресации операндов используется содержимое одного из 8 РОН, номер которого указан в разрядах 0...2. Разряды с 3 по 5 задают способ вычисления адреса.

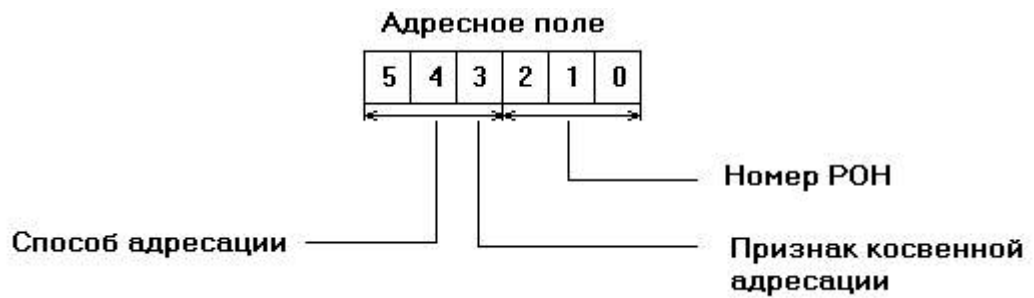


Рис.7

Все реализуемые в микроЭВМ способы адресации можно разделить на 3 группы: прямую, косвенную и адресацию с использованием СК.

Каждой группе соответствует четыре режима адресации.

2.6.1 Режимы прямой адресации

1. Прямая регистровая (режим 0, символическое обозначение А). В режиме 0 содержимое РОН номер R является операндом. Порядок извлечения операнда показан на рис. 8.

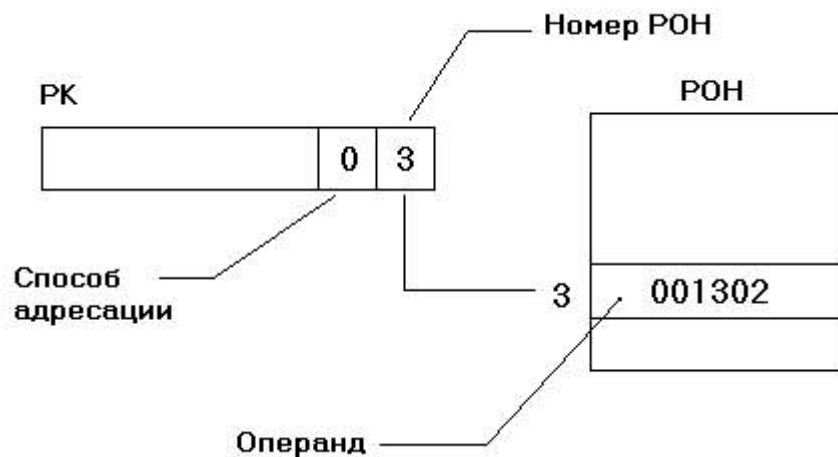


Рис.8

2. Прямая автоинкрементная адресация (режим 2, символическое обозначение (R)+). Регистр R содержит адрес операнда, после обработки

которого содержимое регистра R увеличивается на 1, если операнд – байт, и на 2, если операнд – слово. Порядок обращения к операнду показан на рис.9.

3. Прямая автодекрементная адресация (режим 4, символическое обозначение $-(R)$). После обращения к операнду содержимое регистра уменьшается на 1 для байтовых операций или на 2 для операции над числами, а затем используется в качестве адреса (рис. 10).

4. Индексная (режим 6, символическое обозначение $X(R)$). Адрес вычисляется как сумма содержимого регистра R ("базы") и слова X, следующего за командой ("смещения" или "индексного слова"). Команда, таким образом, будет состоять из 2 или 3 слов, размещённых в последовательных ячейках памяти. Порядок формирования адреса показан на рис.11.

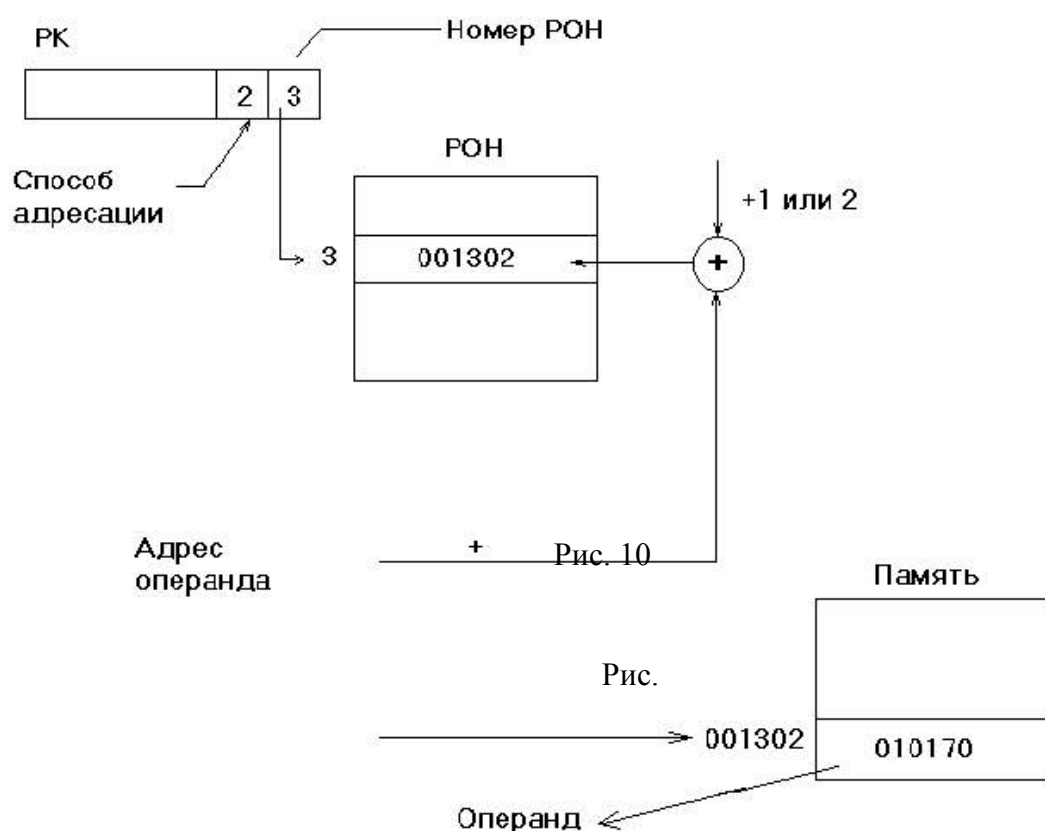


Рис.9

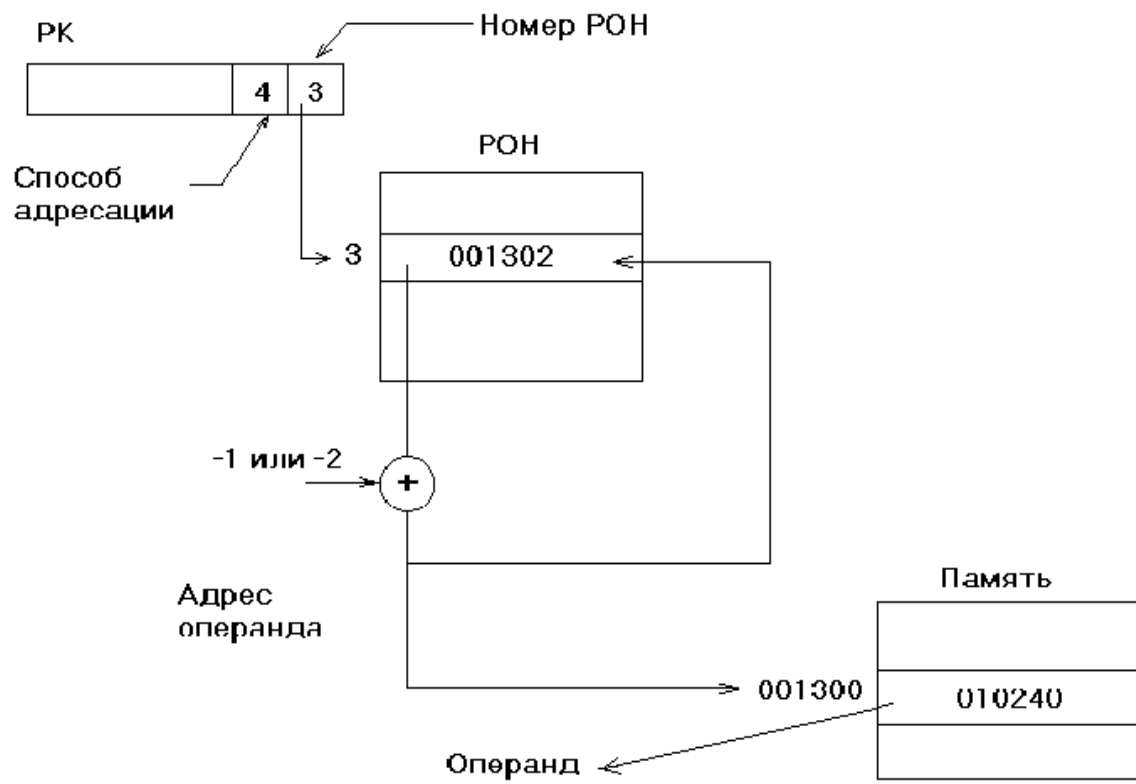


Рис.10

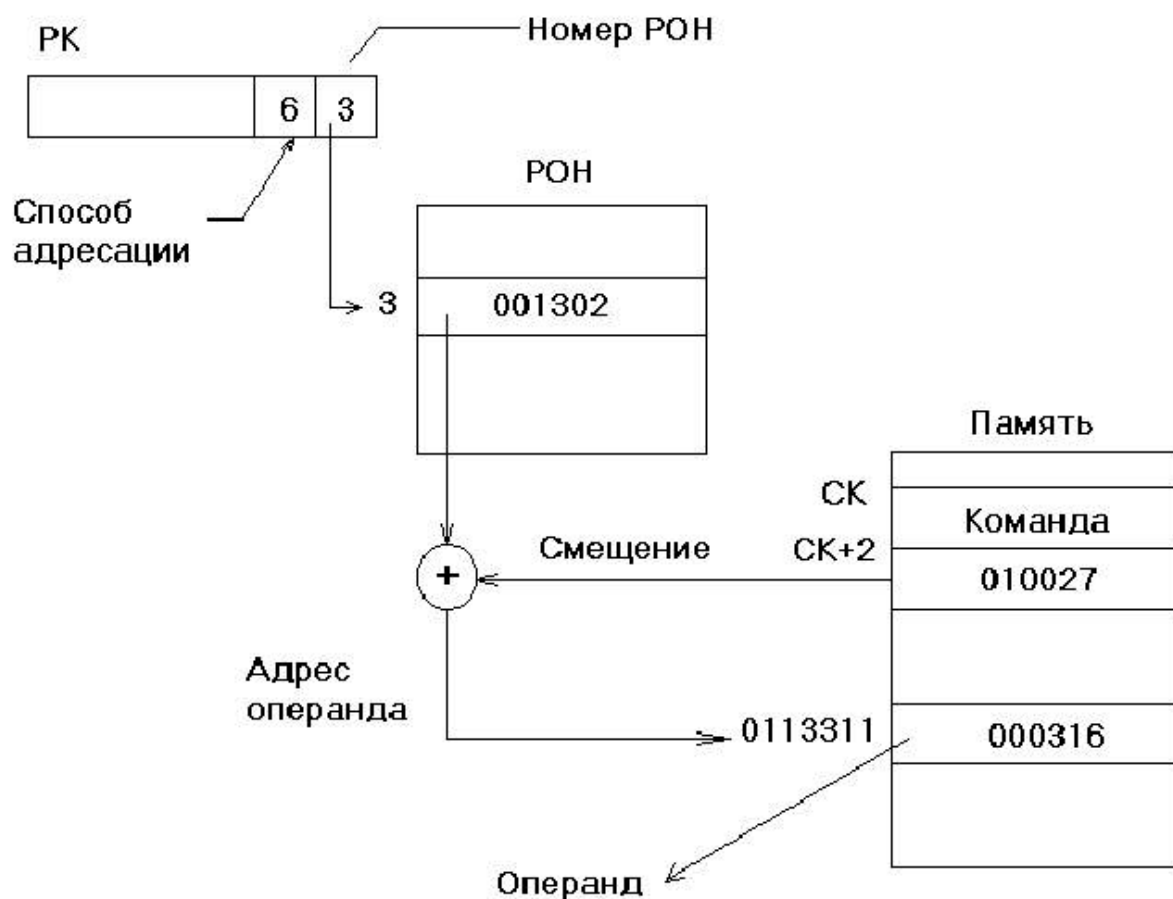


Рис.11

2.6.2 Режимы косвенной адресации

1. Косвенная регистровая адресация (режим 1, символическое обозначение (R) или @R). Содержимое регистра R является адресом операнда (рис. 12).

2. Косвенная автоинкрементная адресация (режим 3, символическое обозначение @ (R)+). Содержимое регистра R используется как адрес адреса операнда, после обращения к которому содержимое регистра увеличивается на 2(рис. 13).

3. Косвенная автодекрементная адресация (режим 5, символическое обозначение @(R)). Содержимое регистра R уменьшается на 2, а затем

используется как адрес адреса операнда (рис. 14).

4. Косвенная индексация (режим 7, символическое обозначение $@X(R)$).
Вычисленное как сумма смещения и базы число используется как адрес
адреса операнда (рис. 15) .

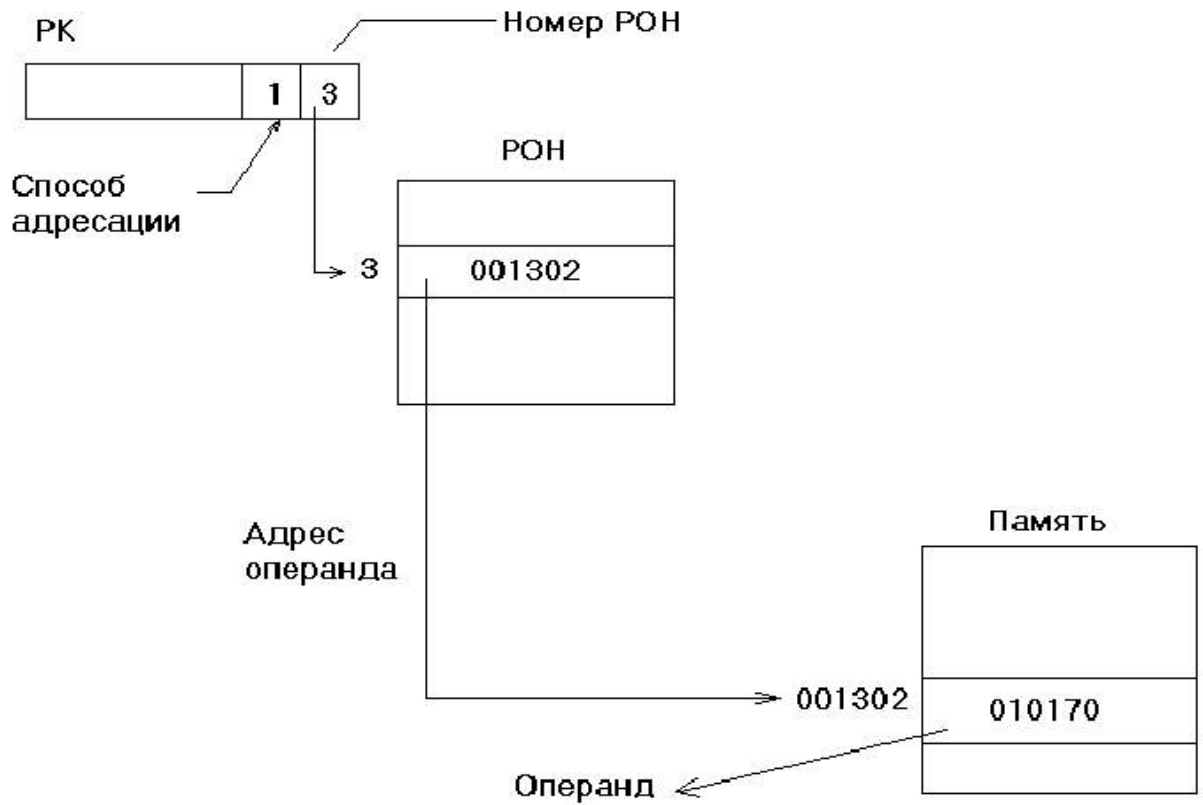


Рис.12

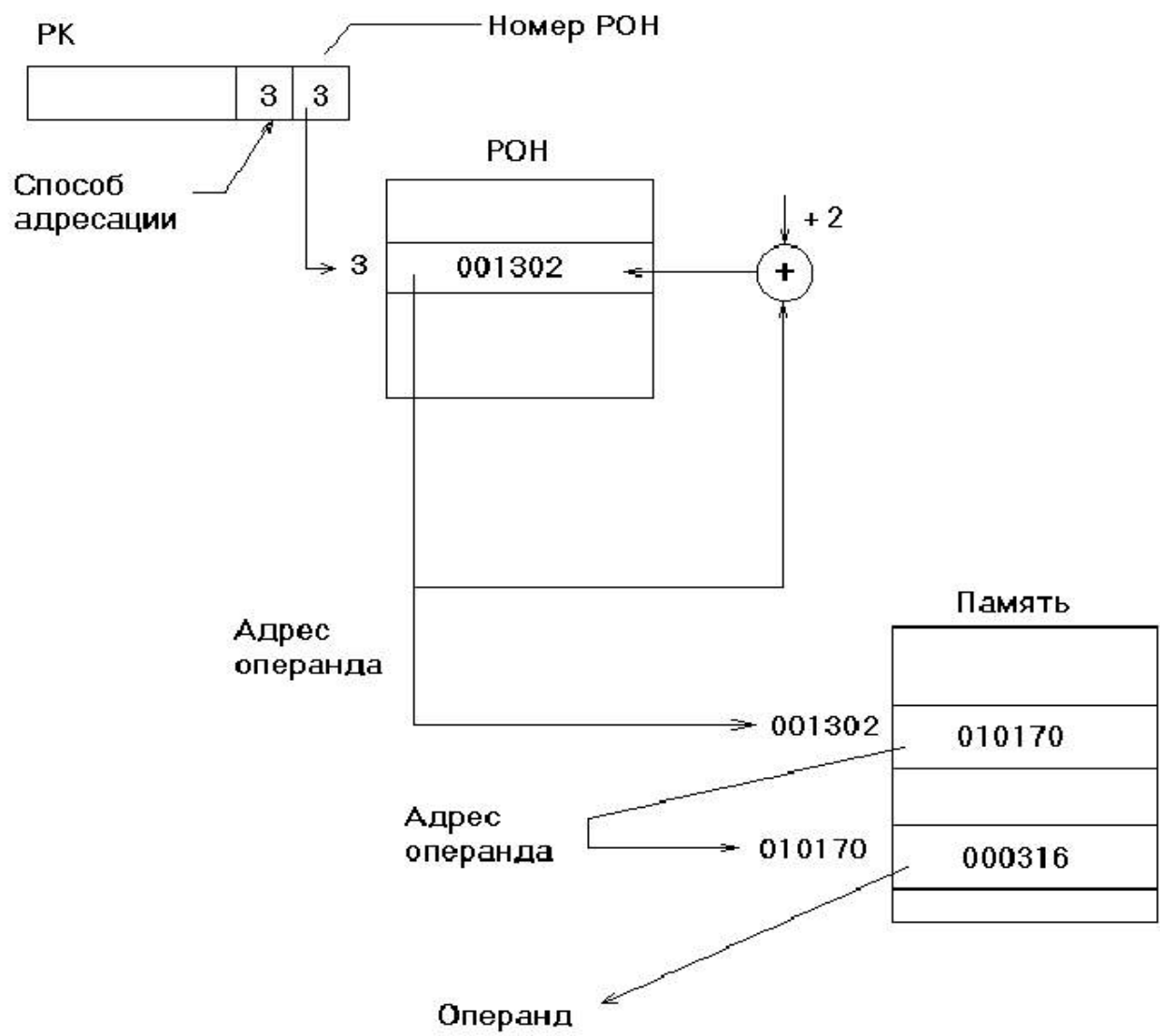


Рис.13

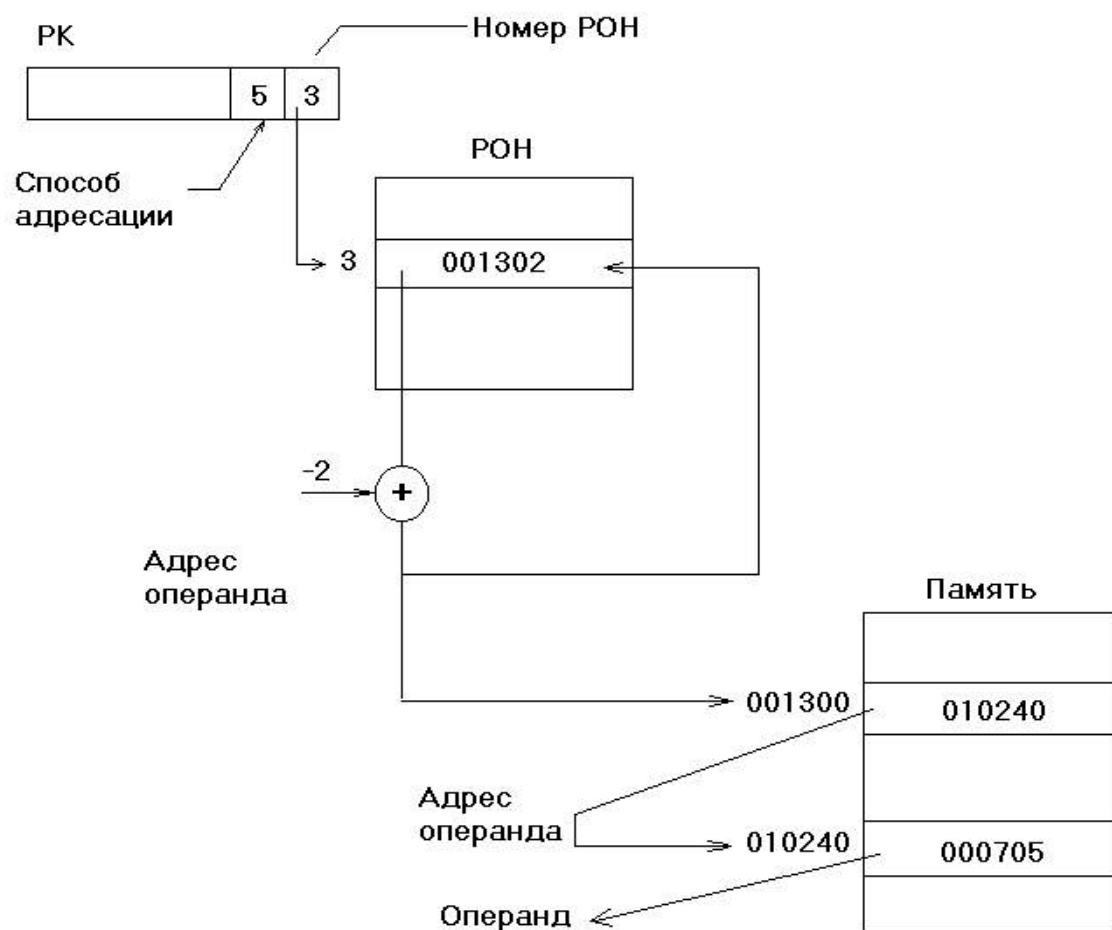


Рис.14

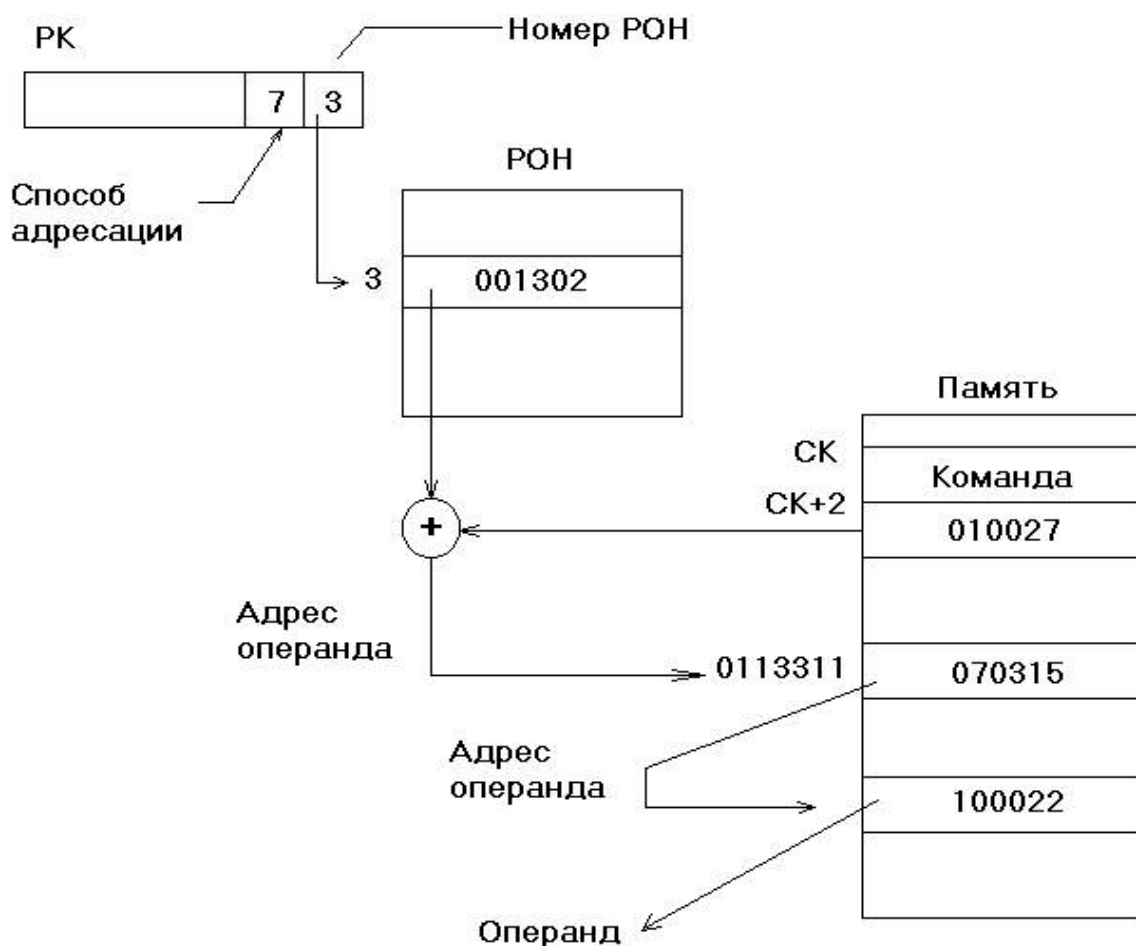


Рис.15

2.6.3 Режим адресации с использованием счётчика команд

Использование для адресации операнда регистра РОН с номером 7 – счётчика команд(СК) позволяет расширить способы адресации к операнду.

1. Непосредственная (режим 2, регистр 7, символическое обозначение #E). Операндом является E – второе (или третье) слово команды. Процессор извлекает по адресу, задаваемому СК, команду, дешифрует её, затем увеличивает СК на 2 и использует новое значение СК как адрес операнда (рис. 16).

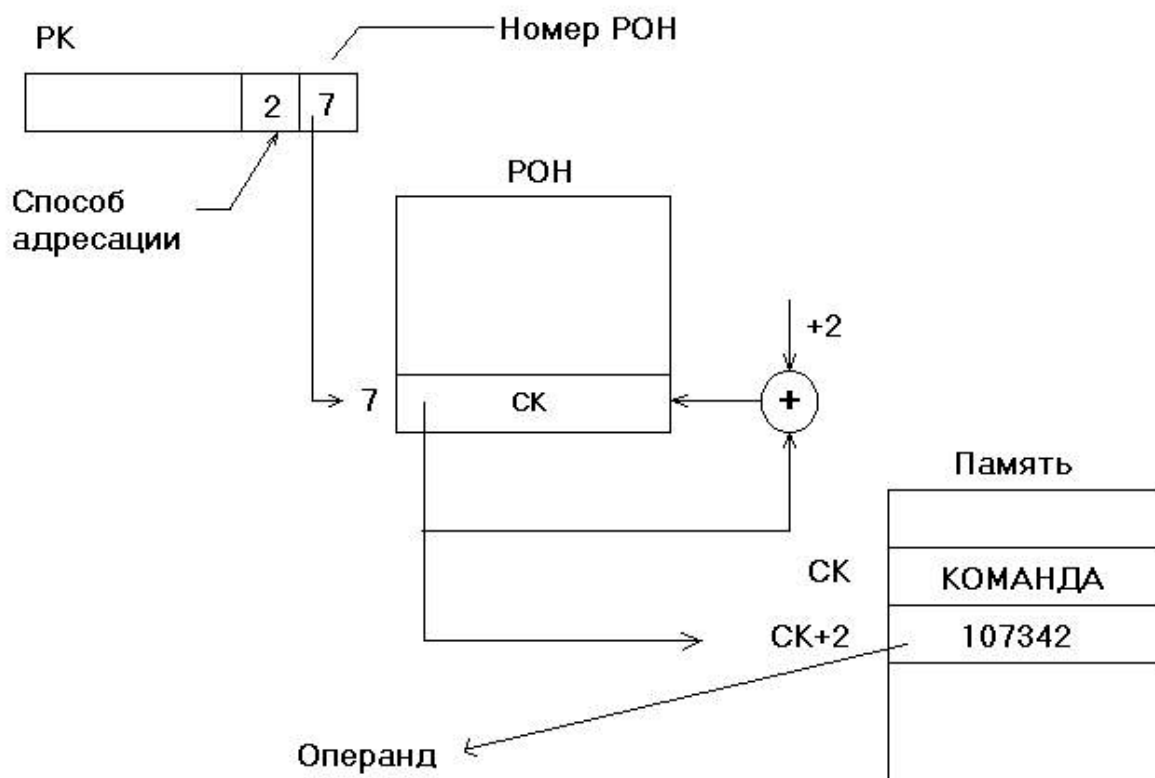


Рис.16

2. Абсолютная (режим 3, регистр 7, символическое обозначение @#E). Адресом операнда является содержимое второго (третьего) слова команды. Процессор после извлечения и дешифрации первого слова команды увеличивает содержимое СК на 2 и использует содержимое ячейки, адресуемой по новому значению СК, как адрес операнда (рис.17).

3. Относительная (режим 6, регистр 7, символическое обозначение E). Адрес вычисляется как сумма смещения E (второго или третьего слова команды) и содержимое СК. Обращение к операнду, порядок которого показан на рис. 18, включает увеличение счётчика команд на 2, извлечение второго слова команды и сложения его с новым содержимым СК, результат используется как адрес.

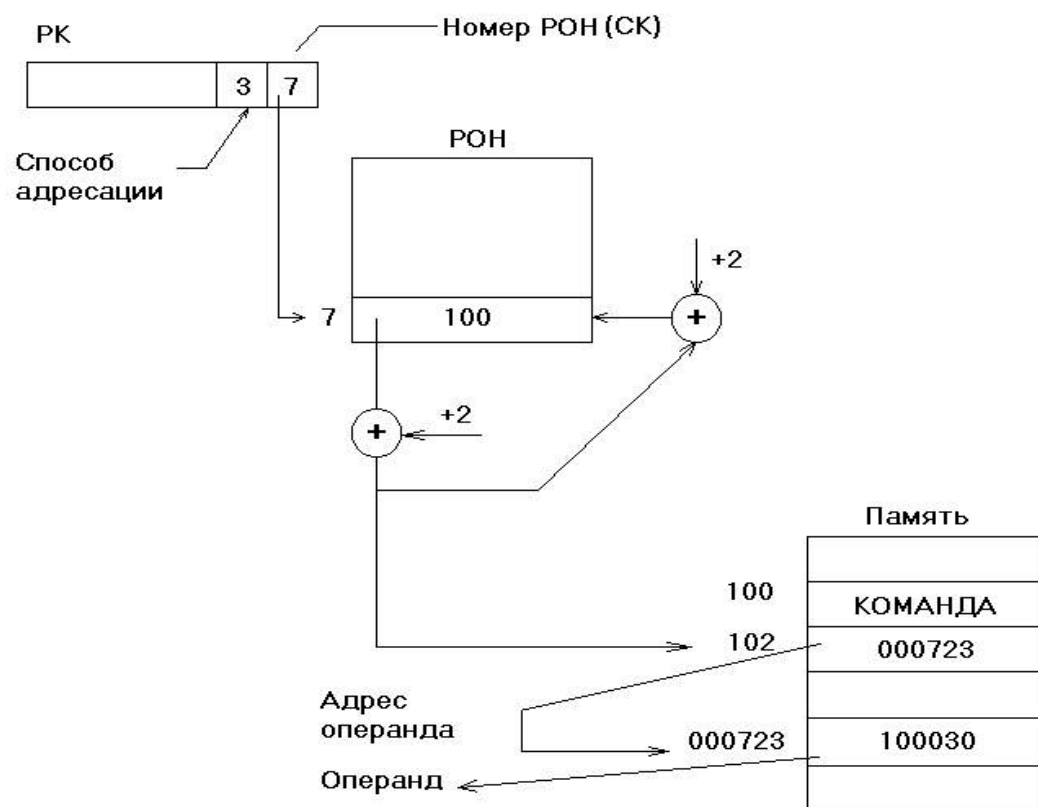


Рис.17

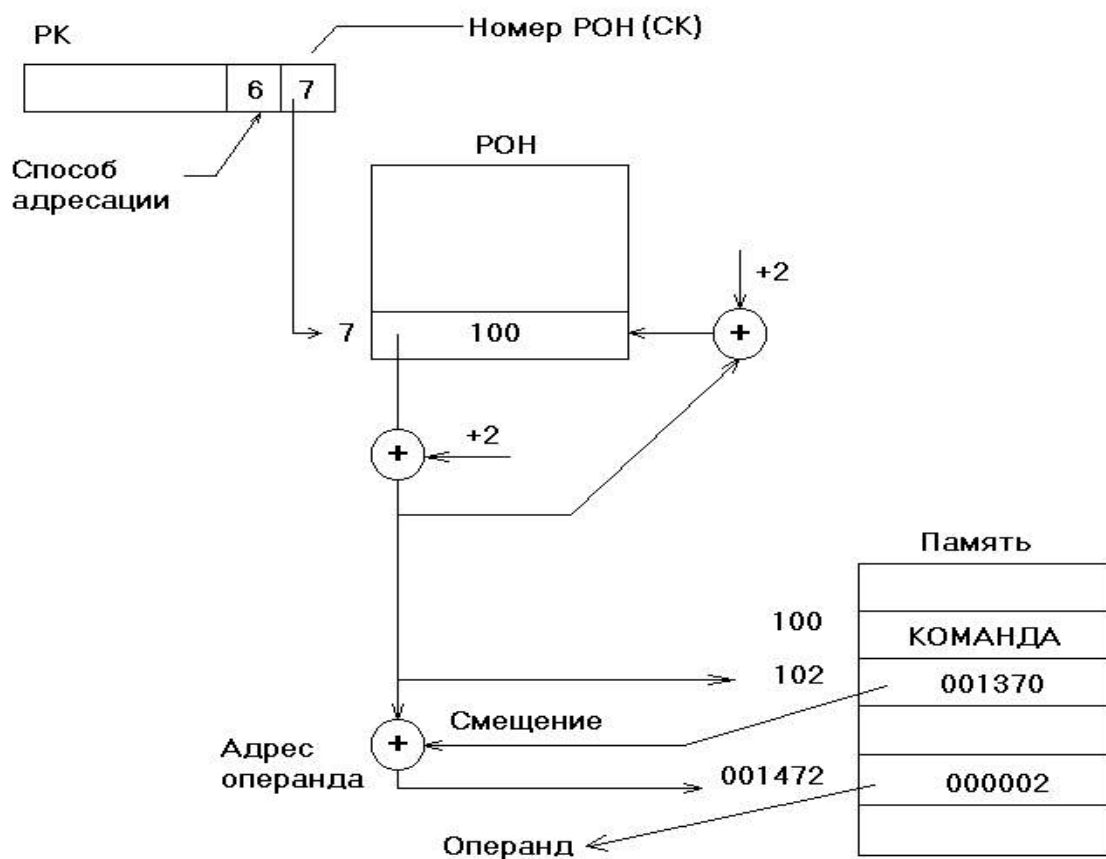


Рис.18

4. Относительно-косвенная (режима 7, регистр 7, символическое обозначение @E). Сумма смещения E и содержимого СК задаёт в этом случае адрес адреса операнда. Порядок обращения к операнду приведён на рис. 19.

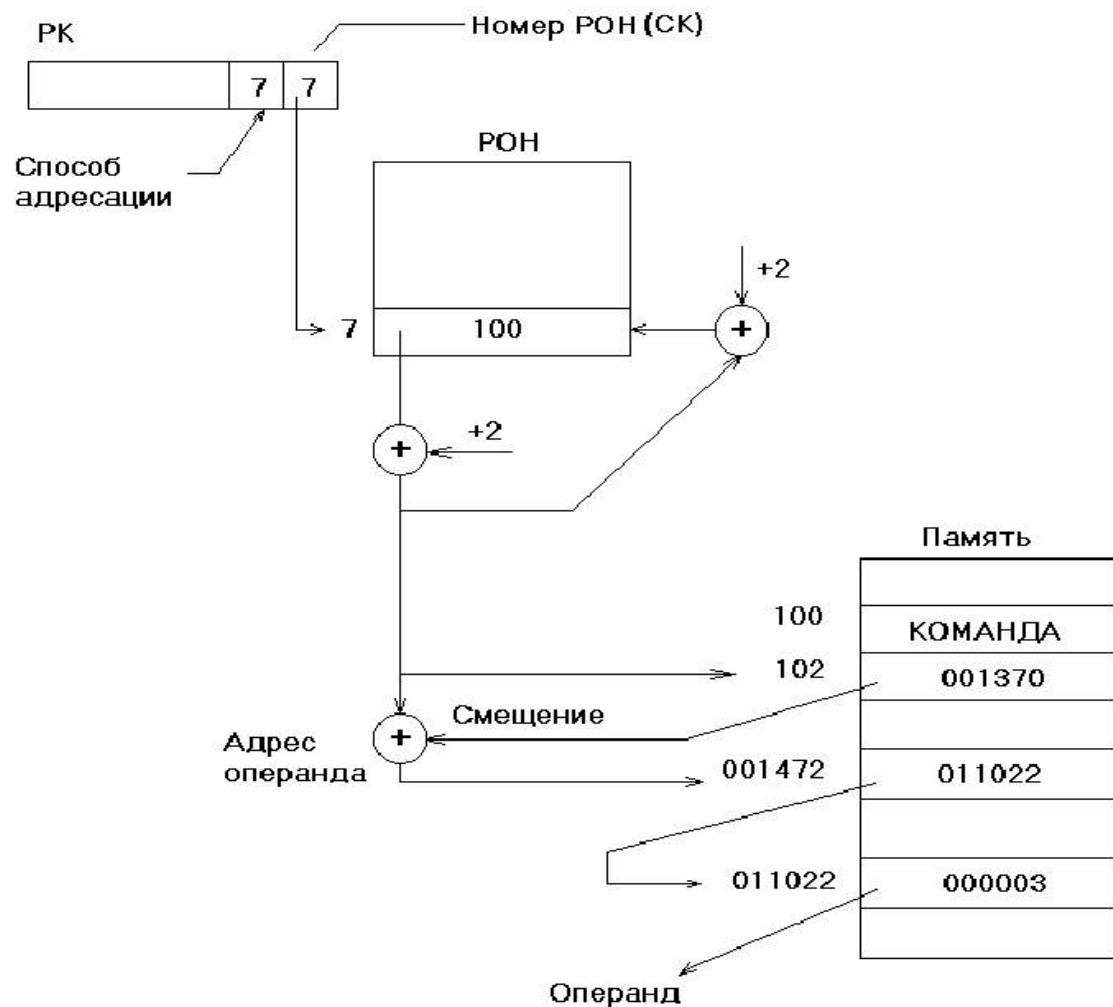


Рис.19

2.7. Система команд

Система команд микроЭВМ включает 64 основные команды и 8 дополнительных команд: расширенной арифметики и операции с плавающей запятой. Все команды можно условно разделить на ряд групп:

- 1) одно- и двухадресные команды, задающие арифметические, логические и посылочные операции над одним или двумя операндами соответственно;

- 2) команды передачи управления;
- 3) команды работы с подпрограммами;
- 4) команды установки кодов условий, которые позволяют программно управлять признаками ССП;
- 5) команды прерывания, используемые при программировании прерываний работы ЦП;
- 6) команды управления состоянием ЦП.

2.7.1. Одноадресные команды

Одноадресные команды задают операции, выполняемые над одним операндом, называемым приёмником. Форматы одноадресных команд приведены на рис. 20. В зависимости от способа адресации приемника команда может состоять из одного или двух слов, в последнем случае второе слово – смещение (режим адресации 6,7 и режимы 2, 3 с номером регистра 7). Список и содержание одноадресных команд приведены в табл.1, при этом приняты следующие условные обозначения, которые будут использоваться в дальнейшем:

1. () – содержимое ячейки или регистра; *src* – ячейка – источник; *loc* – ячейка памяти; *R* – регистр общего назначения; *SS*, *DD* – адреса ячеек; *XXX*, *NN* - смещение (8 разрядов); \wedge - логическое И; \vee -логическое ИЛИ; \forall - исключающее ИЛИ; \bar{A} – инверсия *A*; \leftarrow пересылка; $\uparrow\downarrow$ - занесение в стек и извлечение из стека; *B* – байтовая команда;

Таблица 1

Команда		Наименование	Содержание	Признаки			
мнемоника	код			N	Z	V	C
SWAB	0003DD	Перестановка байт	Переставляются местами чётный и нечётный байты в (dst)	+	+	0	0
CLR (B)	.050DD	Очистка	$\text{dst} \leftarrow 0$	0	1	0	0
COM (B)	.051DD	Инвертирование	$(\text{dst}) \leftarrow \overline{(\text{dst})}$	+	1	0	1
INC (B)	.052DD	Прибавление единицы	$(\text{dst}) \leftarrow (\text{dst}) + 1$	+	+	+	-
DEC (B)	.053DD	Вычитание единицы	$(\text{dst}) \leftarrow (\text{dst}) - 1$	+	+	+	-
NEG (B)	.054DD	Инвертирование и прибавление единицы	$(\text{dst}) \leftarrow \overline{(\text{dst})} + 1$	+	+	+	+
ADC (B)	.055DD	Прибавление переноса	$(\text{dst}) \leftarrow (\text{dst}) + c$	+	+	+	+
SBC(B)	.056DD	Вычитание переноса	$(\text{dst}) \leftarrow (\text{dst}) - c$	+	+	+	+
TST(B)	.057DD	Проверка	$(\text{dst}) \leftarrow (\text{dst})$	+	+	0	0

ROR (B)	.060DD	Вращение вправо	\rightarrow $\text{dst} \leftarrow \text{c}, \text{dst}$	+	+	+	+
ROL(B)	.061DD	Вращение влево	\leftarrow $\text{dst} \leftarrow \text{dst}, \text{c}$	+	+	+	+
ASR(B)	.062DD	Арифметический сдвиг вправо	$\text{dst} \leftarrow (\text{dst})/2$	+	+	+	+
ASL(B)	.063DD	Арифметический сдвиг влево	$\text{dst} \leftarrow (\text{dst}) * 2$	+	+	+	+
SXT(B)	.067DD	Расширение знака	$\text{dst} \leftarrow 0$, если $N=0$ $\text{dst} \leftarrow$, иначе	-	+	0	-
MTPS	1064DD	Запись ССП	$\text{ССП} \leftarrow (\text{src})$	+	+	+	+
MEPS	106735	Чтение ССП	$\text{dst} \leftarrow \text{ССП}$	+	+	0	-

2. точка перед кодом операции означает, что разряд 15 команды устанавливается в 1, если операция байтовая, и в 0, если операция выполняется над словом. Мнемоническое обозначение байтовых операций содержит символ В;



Рис.20

3. обозначение кодов признаков;

0,1 – устанавливается в 0 или 1 принудительно;

+ - - признак устанавливается по результату операции по правилам;

$Z=1$, если результат операции равен 0;

$N=1$, если результат операции - 0;

$C=1$, если при выполнении операции имел место перенос их старшего разряда результата;

$V=1$, если при выполнении операции имело место арифметическое переполнение;

- - - признак не меняется;

Команды SWAB, CLR, COM, INC, DEC, NEG, ADC, SBC выполняют арифметические и логические операции над содержимым ячейки dst. Команда TST, не изменяя содержимого dst, позволяет проверить его на равенство нулю и определить его знак. Правила выполнения команд ASR и ASL (арифметические сдвиги) для случая операций над словами и

байтами приведены на рис. 21 и рис. 22. Соответственно следует обратить внимание, что в операциях сдвига принимает участие признак C. По команде SXT в зависимости от значения признака N все разряды приёмника выполняются нулями или единицами.

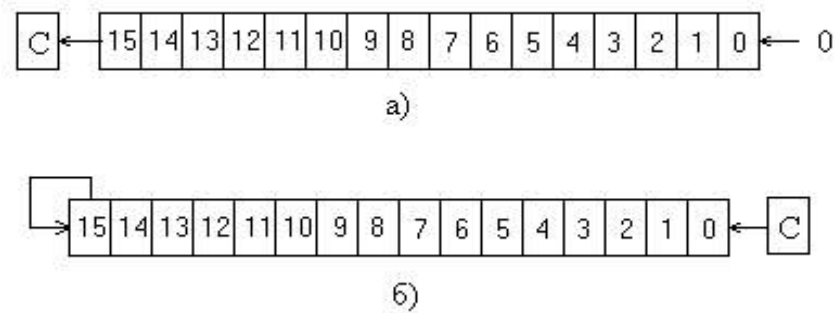


Рис.21

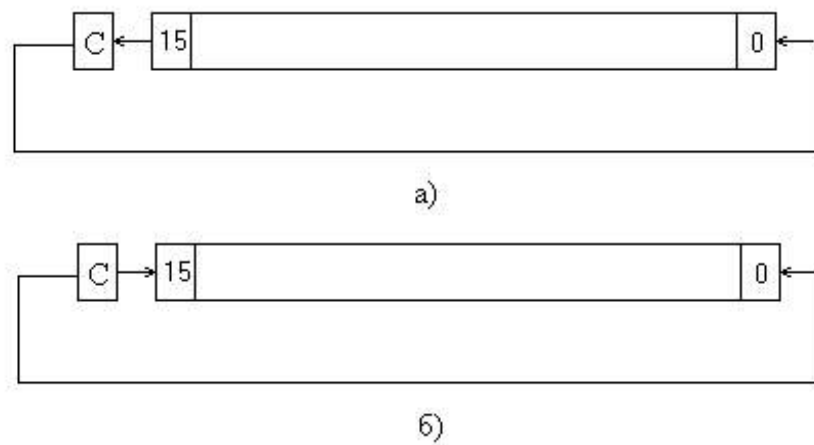


Рис.22

Команды MTRS и MEPS используются для запоминания и восстановления значения РСП только к словам.

Примеры выполнения команд:

Символическое	Адрес, регистр	Содержимое

обозначение и код		до операции	после операции
CLR R4	R4	010312	000000
005004	NZVC	0001	0100
COMB (R4)+	R4	010312	010313
105124	NZVC	0001	1000
	01312	077026	077351
INC200(R4)	R4	010312	010312
005264	NZVC	0001	1010
000200	010512	077777	100000
DEC @-(R4)	R4	010312	010310
005354	NZVC	0001	0000
	010310	000200	000200
	010312	177000	177000
	000200	007777	007776

ASR#741	CK	001000	001004
006227	NZVC	0000	0001
000741	001000	006227	006227
	001002	000741	000360

2.7.2 Двухадресные команды

Двухадресные команды задают двухместные операции, т.е. операции, в которых участвуют два операнда, называемые источником и приёмником. Форматы двухадресных команд приведены на рис. 23. В зависимости от режима адресации каждого из операндов команды могут состоять из одного, двух или трёх слов, второе и третье слова задают смещения. Команды могут выполняться как над словами, так и над байтами, в последнем случае в разряде 15 кода команды записывается 1. Список и содержание двухадресных команд приведены в табл.2.

Таблица 2

Команда		Наименование	Содержание	Признаки			
мнемоника	код			N	Z	V	C
MOV(B)	.ISSDD	(dst)←(src)	Пересылка	+	+	0	-
CMP(B)	.2SSDD	(src)←(dst)	Сравнение	+	+	+	+

BIT(B)	.3SSDD	$(src) \wedge (dst)$	Проверка разрядов	+	+	0	-
BIC(B)	.4SSDD	$(dst) \leftarrow \overline{(src)} \wedge (dst)$	Очистка разрядов	+	+	0	-
BIS(B)	.5SSDD	$(dst) \leftarrow (src) \vee (dst)$	Логическое "ИЛИ" Установка разрядов	+	+	0	-
ADD(B)	06SSDD	$(dst) \leftarrow (dst) + (src)$	Сложение	+	+	+	+
SUB(B)	16SSDD	$(dst) \leftarrow (dst) - (src)$	Вычитание	+	+	+	+



Рис.23

Примеры использования команд:

Символическое обозначение и код	Адрес, регистр	Содержимое	
		до операции	после операции

MOV R1, R3	R1	077777	077777
010103	R3	100002	077777
	NZVC	0000	0000
BIT R2, R4	R2	077077	077077
130214	R4	001003	001003
	NZVC	0101	0001
	001002	010412	010412
ADD @10(R1), @#200	CK	00100	001006
067137	NZVC	0000	1000
000010	R1	010312	010312
000200	010322	100744	100744
	000200	000001	100745

2.7.3. Команды передачи управления

Команды передачи управления выполняют условные и безусловные переходы (ветвление) по адресу, содержащемуся в команде. Формат команд показан на рис. 24. Особенностью формата команд ветвления является то, что в их адресной части указывается байт смещения, задающий адрес перехода

относительно СК. Смещение задается в словах, а не байтах, причём разряд 7 указывает на его знак. Действительный адрес перехода А вычисляется по формуле:

$$A = T + 2XX + 2 \quad (1)$$

где Т – адрес команды перехода; ХХ – значение смещения с учётом знака.

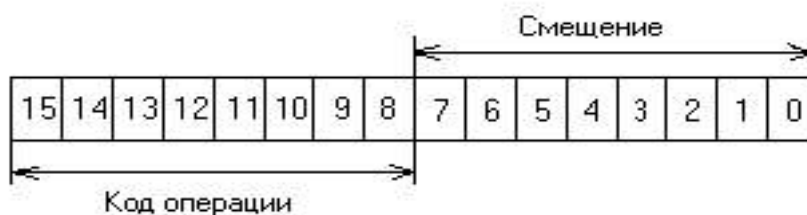


Рис.24

Обратная задача, т.е. определение величины смещения, решается по следующей формуле:

$$XX = \frac{A - T - 2}{2} \quad (2)$$

Пример. Необходимо передать управление из ячейки 001000 по адресу 000716. Вычисляем по формуле (2) в восьмеричной системе счисления:

$$XX = \frac{716 - 1000 - 2}{2} = -30$$

В дополнительном коде смещение -30 представляется кодом 350. Содержание команд перехода приведено в табл. 3. Восьмиразрядное смещение позволяет передать управление в сторону старших адресов ("назад") на 200_8 слов и в сторону старших адресов ("вперед") на 177_8 слов.

Таблица 3

Команда		Наименование	Содержание	Признаки			
мнемоника	код			N	Z	V	C
BR	0004XXX	Ветвление безусловное	$(CK) \leftarrow (CK) + 2XX$ $(CK) \leftarrow (CK) + 2XX$, если:	-	-	-	-
BNE	0010XXX	УП, если $\neq 0$	$Z=0$	-	-	-	-
BEQ	0014XXX	УП, если $=0$	$Z=1$	-	-	-	-
BGE	0020XXX	УП, если ≥ 0	$N \vee V = 0$	-	-	-	-
BLT	0024XXX	УП, если < 0	$N \vee V = 1$	-	-	-	-
BGT	0030XXX	УП, если > 0	$Z \vee (N \vee V) = 0$	-	-	-	-
BLE	0034XXX	УП, если ≤ 0	$Z \vee (N \vee V) = 1$	-	-	-	-
BPL	1000XXX	УП, если "+"	$N=0$	-	-	-	-

BMI	1004XXX	УП, если "-"	N=1	-	-	-	-
BHI	1010XXX	УП, если больше	$Z \vee C=0$	-	-	-	-
BLOS	1014XXX	УП, если меньше	$Z \vee C=1$	-	-	-	-
BVC	1020XXX	УП, если нет переполнения	V=0	-	-	-	-
BYS	1024XXX	УП, если переполнение	V=1	-	-	-	-
BHCS	1030XXX	УП, если нет переноса	C=0	-	-	-	-
BLO	1034XXX	УП, если перенос	C=1	-	-	-	-

Одна из команд ветвления – BR – обеспечивает безусловную передачу управления, все остальные – передачу управления при выполнении определенных условий, формируемых значениями признаков ССП. Все команды перехода не изменяют значения этих признаков.

Пример использования команд:

Символическое обозначение и код	Адрес, регистр	Содержимое	
		до операции	после операции

BR 20	CK	001016	001060
000420	NZVC	0000	0000
BPL 20	CK	001016	001020
100020	NZVC	1000	1000
BGE 210	CK	001016	000470
002210	NZVC	0000	0000

2.7.4. Команды работы с подпрограммами

При программировании больших задач целесообразно всю программу разделить на отдельные части – подпрограммы. Каждая подпрограмма может многократно использоваться, при этом после её выполнения управление автоматически передаётся на продолжение основной программы, вызвавшей подпрограмму (рис.25). Применение подпрограмм позволяет более эффективно использовать память ЭВМ, избежав многократного кодирования в программе повторяющихся участков, и облегчить отладку и написание больших программ. Обращение к подпрограммам включает операции передачи управления к началу подпрограммы и запоминание адреса возврата (адрес команды, следующей за командой обращения к подпрограмме). По окончании подпрограммы адрес возврата используется для передачи управления назад, программе, вызвавшей подпрограмму (см.рис.25).

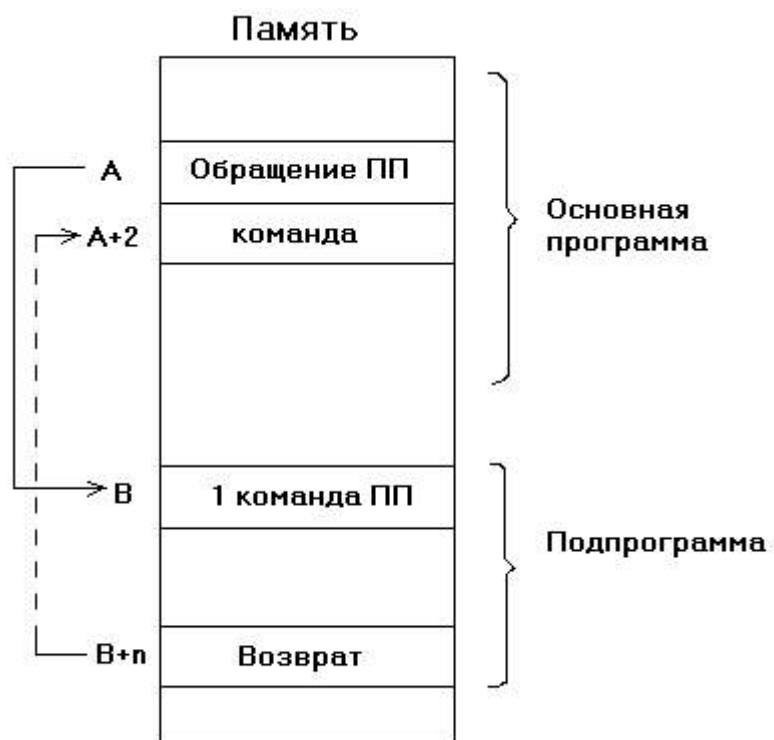


Рис.25

В микроЭВМ “Электроника 60” для работы с подпрограммами применяются стеки. Стек – область оперативной памяти, используемая для временного хранения данных при обращении к подпрограммам и прерываниям. Обращение к стеку выполняется по правилу “первый вошёл, последний вышел”, т.е. данные считываются в порядке, обратном их записи.

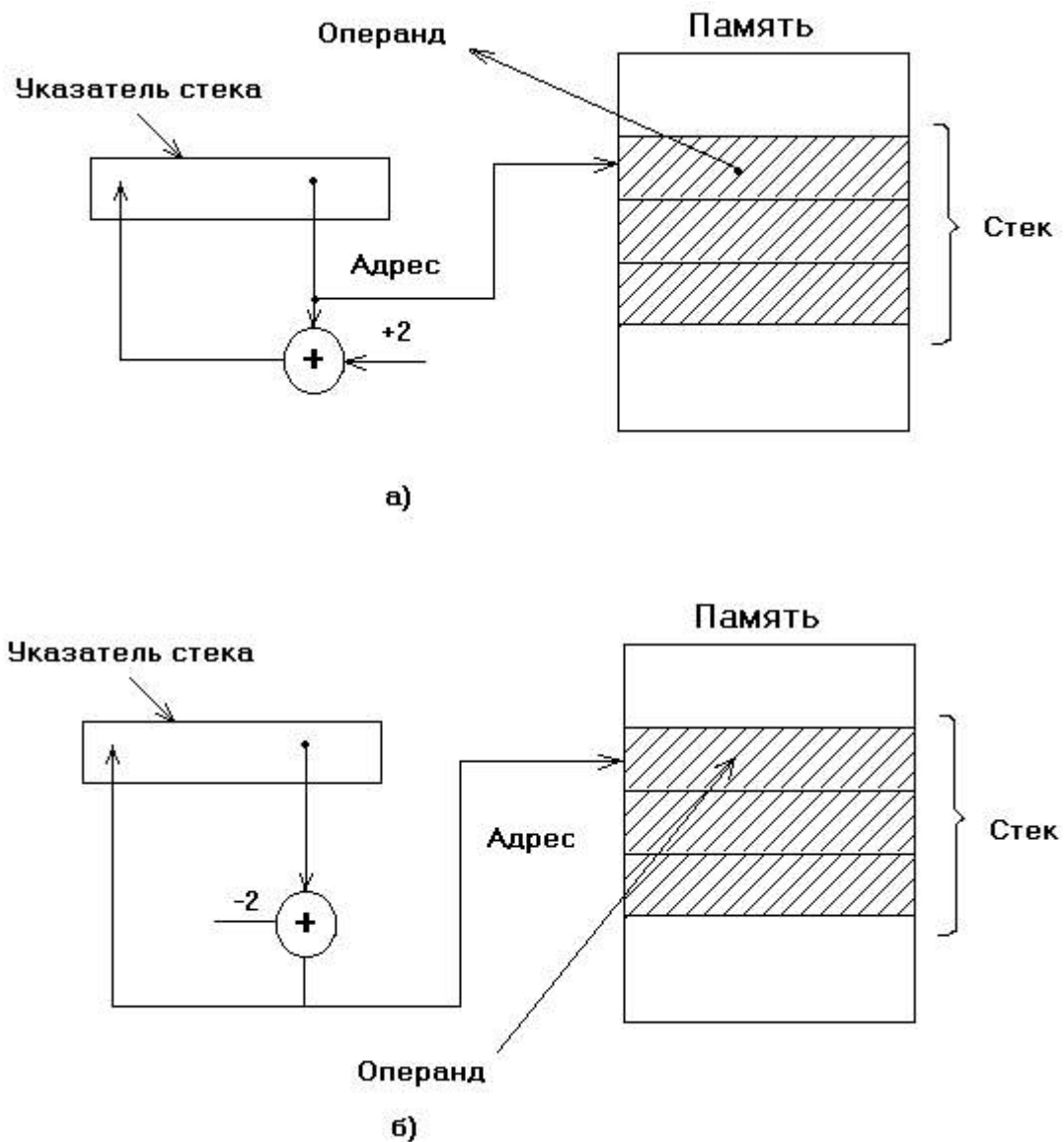


Рис.26

Адресация к стеку реализуется с помощью регистра-указателя УС. Для чтения из стека данных содержимое УС используется в качестве адреса и после чтения увеличивается на шаг (рис.26,а). При записи данных в стек содержимое УС сначала уменьшается на 2, а затем используется в качестве адреса записываемой информации (рис.26,б). Такая процедура обращения к памяти может быть реализована программно с помощью команд пересылки с автоинкрементной и автодекрементной адресацией через регистры. Однако при работе с подпрограммами и прерываниях обращение к стеку реализуется

аппаратно в процессоре, в качестве УС используется **R6**. Форматы команд работы с подпрограммами даны на рис.27, содержание команд приведено в табл.4.

Таблица 4

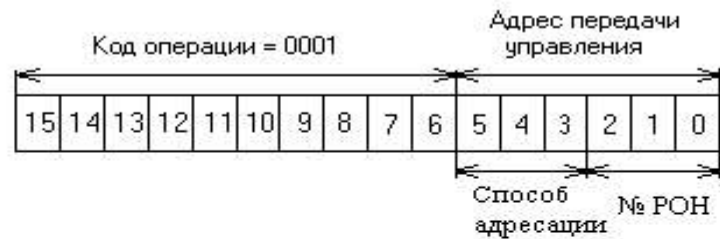
Команда		Наименование	Содержание	Признаки			
мнемоника	код			N	Z	V	C
JSR	004RDD	Переход к программе	$\downarrow(УС) \leftarrow R,$ $R \leftarrow CK,$ $CK \leftarrow (dst)$	-	-	-	-
RTS	00020R	Возврат из программы	$CK \leftarrow (R)$ $(R) \leftarrow (УС) \uparrow$	-	-	-	-
JMP	00	Безусловный переход	$(CK) \leftarrow (dst)$	-	-	-	-
MARK	0064NN	Восстановление УС	$(УС) \leftarrow (CK) + 2N$ $(CK) \leftarrow (R5)$ $(R5) \leftarrow (УС) \uparrow$	-	-	-	-
SOB	077RNN	Вычитание единицы и ветвление	$(R) \leftarrow (R) - 1$ если $\neq 0$, то $(CK) \leftarrow (CK) - 2NN,$ если $= 0$, то $(CK) \leftarrow (CK)$	-	-	-	-

Команда **RTS** передаёт управление из подпрограммы по адресу возврата, записанному в регистре, номер которого указан в формате, затем в регистр записывается содержимое “верхней” ячейки стека.

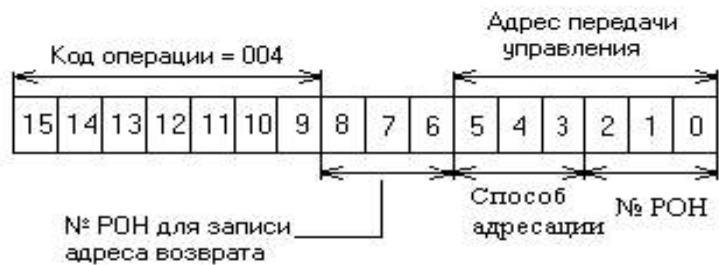
Пример использования команды:

Символическое обозначение и код	Адрес, регистр	Содержимое	
		до операции	после операции

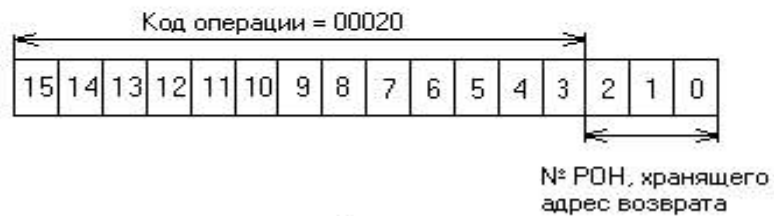
RTS R5	R5	001002	017722
000205	CK	000276	001002
	YC	020000	020002
	020000	017722	017722
	020002	000000	000000



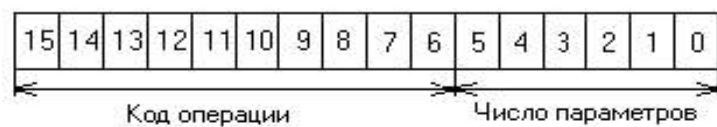
а)



б)



в)



г)



д)

Рис.27

Команда ***JMP*** передаёт управление по адресу, определённомu в соответствии с заданным режимом адресации, причём режим 0 использовать нельзя. Адрес возврата не запоминается.

Пример использования команды:

Символическое обозначение и код	Адрес, регистр	Содержимое	
		до операции	после операции
JMP R1+ 000121	R1	0001000	001002
	СК	000700	001000

Команда **JSR** передаёт управление по заданному адресу, одновременно записывая в указанный в ней регистр адрес возврата (адрес команды, следующей за **JSR**), причём содержимое заполняемого регистра предварительно записывается в стек.

Пример команды:

Символическое обозначение и код	Адрес, регистр	Содержимое	
		до операции	после операции
JSR R5,(R1) 004511	R1	000230	000230
	СК	001000	000230
	R5	017722	001002
	УС	020002	020000
	020000	000000	017722
	020002	000000	000000

Команда **MARK** обеспечивает выход из подпрограммы с одновременным освобождением стека от данных, записанных в него при обращении к подпрограмме.

Команда **SOB** используется для организации циклов. Эта команда передаёт управление на $2N$ адресов “назад”, если содержимое регистра, указанного в ней, не равно 0, и одновременно уменьшает на 1 содержимое

регистра. Если содержимое счётчика равно 0, то управление передаётся команде, следующей за SOB. Правила и примеры использования команд MARK и SOB подробнее изложены в [2,4].

2.7.5. Команды изменения признаков

Команды этой группы позволяют программно изменять признаки, установленные в ССП. Форматы команд показаны на рис.28, а кодирование – в табл.5. Разряды с 0 по 3 управляют установкой соответствующих признаков, а разряд 4 указывает на значение устанавливаемых признаков (0 – установка в 0, 1 – установка в 1).

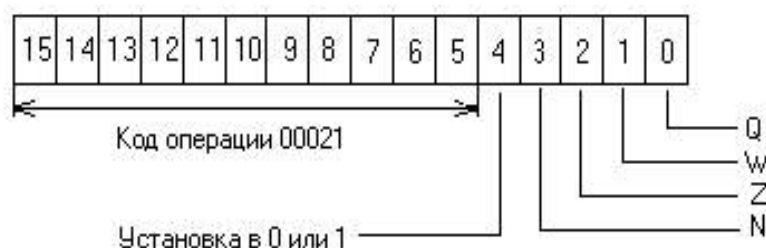


Рис.28

Таблица 5

Команда		Наименование	Содержание	Признаки			
мнемоника	код			N	Z	V	C
CLC	000241	Очистка C	Очистка разряда переноса "C"	-	-	-	0
CLV	000242	Очистка V	Очистка разряда переполнения	-	-	0	-
CLZ	000244	Очистка Z	Очистка разряда нуля "Z"	-	0	-	-
CLN	000250	Очистка N	Очистка разряда знака "N"	0	-	-	-
SEC	000261	Установка C	Установка разряда переноса "C"	-	-	-	1
SEV	000262	Установка V	Установка разряда переполнения	-	-	1	-
SEZ	000264	Установка Z	Установка разряда нуля "Z"	-	1	-	-

CCC	000257	Очистка разрядов РСП	Очистка всех разрядов кодов условий	0	0	0	0
SEN	000270	Установка N	Установка разряда знака "N"	1	-	-	-
SCC	000277	Установка разрядов РСП	Установка всех разрядов кодов условий	1	1	1	1

Примеры использования команд:

Символическое обозначение и код	Адрес, регистр	Содержимое	
		до операции	после операции
SEV 000262	NZVC	1000	1010
CLC 000241	NZVC	0001	0000
SCC	NZVC	0100	1111

2.7.6. Команды обслуживания прерываний

При работе микроЭВМ возможны появления программно независимых событий, т.е. событий, не предусмотренных в выполненной в данный момент в процессоре программе. Процессор в этом случае должен прерывать выполнение текущей программы и передать управление программе, обслуживающей прерывание, при этом должна сохраняться возможность возврата к прерванной программе.

В микроЭВМ "Электроника 60" предусмотрен специальный аппарат обработки прерываний, источниками которых могут быть внешние устройства, посылающие запросы на связь с ЦП, средства диагностики и специальные команды процессора.

Каждому из источников прерываний отводится две ячейки в области оперативной памяти (адреса с 000000 по 000777) для хранения вектора прерываний, состоящего из двух слов. Первое слово – адрес первой команды программы, обслуживающей соответствующее прерывание, а второе слово – ССП, установленное для этой программы. Независимо от источника прерываний его процедура включает следующие операции, выполняемые ЦП автоматически (рис.29):

1. одновременно с сигналом прерывания источник передаёт адрес своего вектора;
2. если прерывание ЦП в данный момент разрешено (7 разряд ССП=0), то процессор записывает содержимое СК и ССП прерываемой программы в стек;
3. в счётчик команд и регистр состояния ЦП записываются соответственно значения первого и второго слова вектора, т.е. новые значения СК и ССП;
4. запускается программа обслуживания прерывания с адреса, заданного новым содержимым СК;
5. по окончании выполнения программы обслуживания прерываний её последней командой ВОЗВРАТ ИЗ ПРЕРЫВАНИЯ **RTI** осуществляется извлечение из стека и запись на счётчик команд и регистр состояния ранее записанных в него значений СК и ССП на момент прерывания, т.е. продолжение выполнения прерванной программы (рис.30).

Состав команд обслуживания прерываний приведён в табл.6, форматы даны на рис.29.

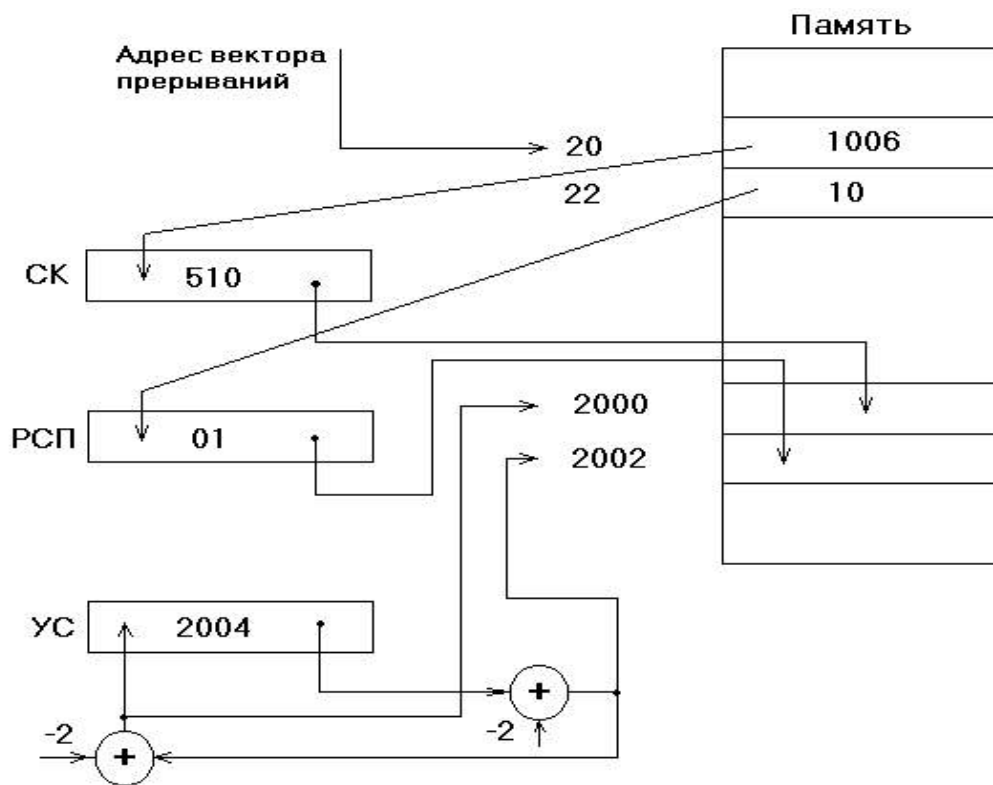


Рис.29

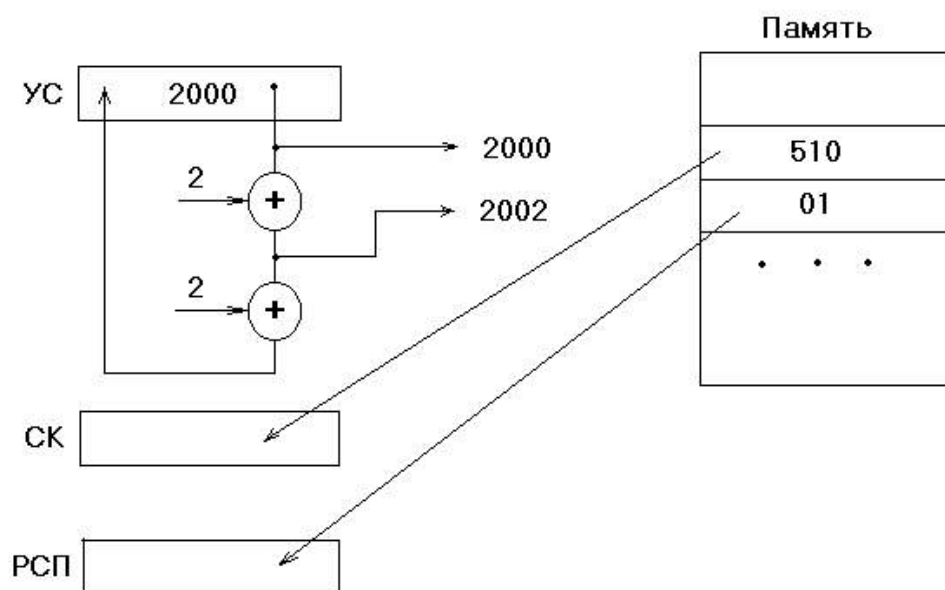


Рис.30

Таблица 6

Команда		Наименование	Содержание	Признаки			
мнемоника	код			N	Z	V	C

RTI	000002	Возврат из прерывания	$CK \leftarrow (YC) \uparrow$, $RSP \leftarrow (YC) \uparrow$, $(N, Z, V, C) \leftarrow (YC) \uparrow$	-	-	-	-
BPT	000003	Прерывание для отладки	$\downarrow(YC) \leftarrow RSP$ $\downarrow(YC) \leftarrow CK$ $CK \leftarrow (14); RSP \leftarrow (16)$	-	-	-	-
IOT	000004	Прерывание для ввода-вывода	$\downarrow(YC) \leftarrow RSP$ $\downarrow(YC) \leftarrow CK$ $CK \leftarrow (22); RSP \leftarrow (24)$	-	-	-	-
RTT	000006	Возврат из прерывания	$CK \leftarrow (YC) \uparrow$ $RSP \leftarrow (YC) \uparrow$ $(N, Z, V, C) \leftarrow (YC) \uparrow$	-	-	-	-
EMT	104000 104377	Командное прерывание	$\downarrow(YC) \leftarrow RSP$ $\downarrow(YC) \leftarrow CK$ $CK \leftarrow (30); RSP \leftarrow (32)$	-	-	-	-
TRAP	104400 104777	Командное прерывание	$\downarrow(YC) \leftarrow RSP$ $\downarrow(YC) \leftarrow CK$ $CK \leftarrow (34); RSP \leftarrow (36)$	-	-	-	-

Пример выполнения команды **RTI**:

Символическое обозначение и код	Адрес, регистр	Содержимое	
		до операции	после операции
RTI 000002	CK	000770	001000
	NZVC	0000	0011
	YC	020000	020004
	020002	000003	000003
	020000	001000	001000

Действие команд **EMT**, **TRAP**, **BPT** и **IOT** аналогично и заключается в вызове программных прерываний по закреплённому за каждой командой вектору. Однако пользователю в своих программах рекомендуется применять лишь команду **TRAP**, так как команды **EMT**, **BPT** и **IOT** используются в операционных системах.

Пример выполнения команды **TRAP**:

Символическое обозначение и код	Адрес, регистр	Содержимое	
		до операции	после операции
TRAP 104400	СК	001000	000670
	NZVC	0011	0001
	000034	000670	000670
	000036	000001	000001
	УС	020004	020000
	020002	000000	000003
	020000	000000	001002

Команды **RTI** и **RTT** обеспечивают возврат из подпрограммы, и ими должна оканчиваться каждая программа обработки прерываний. Отличие заключается в том, что при установке разряда **T** в единицу при восстановлении ССП из стека по команде **RTI**, команда, следующая за **RTI**, выполняться не будет, и произойдёт прерывание с вектором 14, а такая же ситуация при выполнении команды **RTT** вызовет прерывание после выполнения первой команды следующей команды **RTT**.

2.7.7. Команды управления состоянием процессора

Форматы команд этой группы состоят из одного слова, содержащего код операции. Коды и содержание команд приведены в табл.7.

Таблица 7

Команда		Наименование	Содержание	Признаки			
мнемоника	код			N	Z	V	C
HALT	000000	Останов	Процессор прекращает выполнение всех функций	-	-	-	-

WAIT	000001	Ожидание	Процессор ждёт внешних прерываний	-	-	-	-
RESET	000005	Сброс	Сброс внешних устройств	-	-	-	-
NOP	000240	Нет операции		-	-	-	-

Команда ***HALT*** вызывает останов работы ЦП. Работа процессора может быть продолжена заданием с терминала команды ПРОДОЛЖЕНИЕ.

Команда ***WAIT*** вызывает временное прекращение выполнения программы до поступления прерывания от внешних устройств.

Команда ***RESET*** вызывает выработку сигнала сброса внешних устройств длительностью 10 мкс, передаваемого в канал. ЦП возобновляет работу через 90 с после выработки сигнала сброса.

3. Эмулятор МП машин класса PDP-11. ver 18.02

Руководство пользователя

Рабочий стол состоит из полосы меню, строки статуса, окна ОЗУ(1), окна ЦП(2) и трех дополнительных окон ОЗУ(3,4,5).

МЕНЮ

Файл/Новый <***Alt+N***> - обнуляет память и регистры.

Файл/Открыть <***F3***> - загружает файл в память с нулевого адреса.

Файл/Сохранить <***F2***> - сохраняет память в файл с нулевого адреса до

заданного.

Файл/Помощь **<F1>** - просмотр данного текста.

Файл/Выход **<Alt+X>** - выход.

Отчет/Открыть - открывает файл отчета.

Отчет/Заккрыть - сохраняет файл отчета.

Отчет/Очистить - очищает файл отчета.

Отчет/Сохранить содержимое окна **<F5>** - помещает информацию из текущего окна в файл отчета.

После завершения работы с файлом отчета, необходимо выполнить операцию закрытия, иначе информация будет потеряна.

ОКНО ОЗУ(1)

Окно состоит из трех полей: адрес ячейки, содержимое, мнемоника команды. Переход между ними **<Tab>**. При нажатии на **<Enter>** на соответствующем поле можно перейти по адресу, ввести код команды (данных) или мнемонику. Передвижение по памяти также можно осуществить клавишами управления курсором и **PgUp, PgDn**.

Комбинация клавиш **<Alt+P>** позволяет быстро установить счетчик команд по текущему положению курсора.

ОКНО ЦП(2)

Отображает содержимое регистров, флагов и фазы команд. При нажатии

на **<Enter>** можно изменить содержимое регистров.

ОКНА ОЗУ(3,4,5)

Окна для просмотра и изменения содержимого памяти. См. ОЗУ(1).

ОТЛАДКА ПРОГРАММЫ

1.Обычный режим (**F9**). Выполняет программу до команды или точки останова (**F4**).

2.Пошаговый режим (**F7**). Выполняет одну команду.

3.Микрошаговый режим (**F8**). Выполняет одну команду. Дает возможность просмотра всех фаз команды (**F8**).

4. Задания к лабораторным работам

4.1. Изучение основных команд работы с пультовым терминалом

4.1.1. Открыть ячейку памяти с заданным адресом, прочитать записанную в ней информацию, записать в ячейку новые данные. Закрыть ячейку. Повторно открыть ячейку, убедиться в правильности записанной информации.

4.1.2. Открыть ячейку памяти с заданным адресом (табл.8),закрыть ячейку, одновременно открыв сначала следующую, а затем предыдущую ячейки.

Таблица 8

4.1.3. Открыть ячейку с заданным адресом (табл.8), записать в неё

Номер варианта	1	2	3	4	5	6
1 адрес	1000	1100	1200	1300	1400	1500
2 адрес	500	510	520	530	540	550

константу, которая будет использоваться как исполнительный адрес. Закрывать ячейку. Повторно открыть ячейку и использовать её содержимое при косвенной адресации.

4.1.4. Используя данные табл.8, открыть ячейку с относительным адресом.

4.2. Изучение способов адресации

4.2.1. Различные способы адресации изучаются на примере выполнения команды пересылки (**MOV**).

Необходимо записать в память, начиная с адреса 1000, восьмеричный код команды с заданными адресами и признаками методов адресации. В ячейку, следующую за командой пересылки, записывается код команды **HALT** (3). В ячейках для хранения пересылаемого операнда, записывается константа, равная номеру варианта.

Запуск команды выполняется директивой **1000G** в режиме ПРОГРАММА. В режиме ПУЛЬТ необходима дополнительная команда **P**.

После завершения выполнения команды просматривается операнд по адресу приёмника и регистры (ячейки), содержащие адрес при косвенной адресации.

4.2.2. Прямая адресация регистров (табл.9). Команда ***MOV R_i,R_j***.

4.2.3. Автоинкрементная адресация (табл.10). Команда ***MOV R0,(R_i)+***.

4.2.4. Автодекрементная адресация (табл.10). Команда ***MOV R0,-(R_i)***.

4.2.5. Индексация (табл.10). Команда ***MOV R0,X(R_i)***.

Таблица 9

Номер варианта	1	2	3	4	5	6
1 адрес	R0	R1	R2	R3	R4	R5
2 адрес	R1	R2	R3	R4	R5	R0

4.2.6. Косвенные
адресации через ячейку
(табл.11):

1) команда ***MOV R0,@(R_i)***.

2) команда ***MOV R0,@(R_i)+***.

3) команда ***MOV R0,@-(R_i)***.

4.2.7. Косвенная индексация (табл.11). Команда ***MOV R0,@X(R_i)***.

4.2.8. Непосредственная адресация (табл.12). Команда ***MOV #X, R_i***.

4.2.9. Абсолютная адресация (табл.12). Команда ***MOV R0,@#X***.

4.2.10. Относительная адресация (табл.12). Команда ***MOV R0,X***.
Содержимое ***R7***=1000.

4.2.11. Относительно-косвенная адресация (табл.12). Команда ***MOV R0,@X***.
Содержимое ***R7***=1000.

Таблица 10

Номер варианта	1	2	3	4	5	6
Номер R_i	1	2	3	4	5	6
Содержимое R_i	1100	1102	1104	1106	1110	1112
Содержимое R_0	1	2	3	4	5	6
Значение X	2	4	6	10	12	14

Таблица 11

Номер варианта	1	2	3	4	5	6
Номер R_i	1	2	3	4	5	6
Содержимое R_i	500	502	504	506	510	512
Содержимое ячейки памяти	1100	1102	1104	1106	1110	1112
Значение X	2	4	6	10	12	14
Содержимое R_0	1	2	3	4	5	6

Таблица 12

Номер варианта	1	2	3	4	5	6
Содержимое R_0	100	200	300	400	500	600
Значение X	10	20	30	40	50	60
Содержимое адресуемой ячейки памяти	1010	1020	1030	1040	1050	1060

4.3. Выполнение команд процессором

4.3.1. Записать восьмеричный код команды в оперативную память, начиная с ячейки с адресом 1000. Операнды размещать в регистрах ***R0*** и ***R1***.

В ячейке, следующей за командой, поместить код команды ***HALT (0)***.

Перед исполнением команды распечатать содержимое ячейки с командой, содержимое общих регистров ***R0*** и ***R1***, содержимое регистра состояний.

Команду исполнять в режиме ПРОГРАММА по инструкции пультавого терминала 1000G.

После выполнения команды распечатать содержимое тех же регистров,

что и перед исполнением команды.

4.3.2. Одноадресные команды (табл.13):

Таблица 13

Номер варианта	1	2	3	4	5	6
Признак байта	0	1	0	1	0	1
Операнд	-6	6	-4	4	-5	5
Номера команд	1,7	2,8	3,9	4,10	5,11	6,12

- 1) очистка,
- 2) инвертирование и увеличение на 1,
- 3) прибавление 1,
- 4) вычитание 1,
- 5) инвертирование,
- 6) проверка,
- 7) сдвиг вправо,
- 8) сдвиг влево,
- 9) вращение вправо,
- 10) вращение влево,
- 11) перестановка байт,

12) расширение знака.

4.3.3. Двухадресные команды (табл.14):

1) сравнение,

2) очистка бит,

3) установка бит (логическое “ИЛИ”),

Таблица 14

4) сложение,

5) вычитание,

Номер варианта	1	2	3	4	5	6
Операнд 1	1035	5	123	1111	77	55
Операнд 2	1030	-3	120	1010	1	-5

6) проверка бит.

4.3.4. Команды для работы с операндами увеличенной длины. Операнды представлены 32-разрядными словами, хранящимися в парах общих регистров. Первый операнд находится в регистрах **R0** и **R1**, второй – в регистрах **R4** и **R5**. Выполнить команды сложения операндов (табл.15):

ADD<мл. часть опер. 1>, <мл. часть опер. 2>

ADD<мл. часть опер. 2>, **C**

ADD<ст. часть опер. 1>, <ст. часть опер. 2>

Таблица 15

Номер варианта		1	2	3	4	5	6
Операнд1	ст. часть	011017	000003	000004	000005	000006	000000
	мл. часть	176543	140003	177777	111117	144333	177774
Операнд2	ст. часть	001133	000011	000004	000005	000006	000000
	мл. часть	143000	140011	000001	170001	170001	135321

4.4. Организация переходов

4.4.1. Подготовить программу в соответствии с лабораторного задания (табл.16). Операнды размещать в общих регистрах. Записать восьмеричные коды команд в ячейки памяти, начиная с адреса 1000.

Таблица 16

Номер варианта	1	2	3	4	5	6
Тип команды для сравнения	CMP	SUB	CMP	SUB	CMP	SUB
Задача	А		Б		В	

Задачи:

А. Сложить операнд, хранящийся в регистре R3, со значением наименьшего из 2 чисел, хранящихся в регистрах R0 и R1.

Б. Вычислить абсолютную величину числа, хранящегося в общем регистре R0.

В. В регистрах R0 и R1 записаны два произвольных числа. Переписать их в регистры 2 и 3, упорядочив по возрастанию.

4.4.2. Очистить регистр состояний. Перевести переключатель режима в положение ПУЛЬТ. Командой пульта терминала 1000 установить счётчик команд на начало программы.

4.4.3. Выполнить программу покомандно, индицируя после каждой команды содержимое общих регистров и регистра состояний.

При использовании для сравнения команды “Вычитание”, предусмотреть сохранение второго операнда.

4.5. Использование стека при выполнении арифметических операций

4.5.1. Вычислить значение арифметического выражения

$$Y=K*X$$

для различных значений X (см. табл. 10). Параметр K и результат вычислений должны находиться в общем регистре $R0$.

Вычисления производить при разложении X на множители, являющиеся степенями 2. Промежуточные результаты хранить в стеке.

Пример: $Y=5*K=(4+1)*K=2^2*K+K$

Умножение на 2^n выполняется последовательным сдвигом на n разрядов влево. Перед началом выполнения установить указатель стека на адрес стека ($R6=2000$).

4.6. Переход на подпрограмму

4.6.1. Выполнить переход на подпрограмму без передачи параметров. Для этого необходимо подготовить последовательность команд, соответствующую вызывающей программе. Программу расположить, начиная с ячейки с адресом 1000.

Команды подпрограммы расположить в ячейках, начиная с адреса 1000.

Вызов подпрограммы:

CLR R0

JSR R5, SVB

CLR R0

HALT

Подпрограмма

SVB: INC R0

RTS R5

После того, как основная программа и подпрограмма будут записаны в оперативную память, установить указатель стека на начало стека (ячейка 1200).

Переключатель режима должен находиться в положении ПУЛЬТ. Командой пульта терминала 1000G установить счётчик команд на начало программы. Распечатать содержимое регистров **R0**, **R5**, **R6** и **R7**, а также содержимое вершины стека. Выполнить программу в потактном режиме. После выполнения каждой команды фиксировать содержимое указанных выше регистров и содержимое стека.

Пояснить полученные результаты.

4.6.2. Команда программного прерывания. Выполнить переход на подпрограмму с помощью команды **TRAP** (табл. 17).

В вектор прерывания по команде **TRAP** в ячейку 34 необходимо записать адрес диспетчера команды **TRAP** – 1100, в ячейку 36 – слово состояния прерывающей программы – 1.

Указатель стека (**R6**) должен содержать адрес вершины стека (2000).

Таблица 17

Номер варианта	1	2	3	4	5	6
Номер регистра R _i	1	2	3	4	5	1
Код в команде TRAP	400	404	410	414	420	424

Основную программу расположить, начиная с адреса 1000:

CLR R_i

TRAP <код_i>

CLR

HALT

Расположить **TRAP**-диспетчер, начиная с адреса 1100:

MOV (SP), R0

BIC #177400, R0

ASL R0

JMP @1140(R0)

Таблицу переходов команды **TRAP** расположить, начиная с адреса 1140:

1140 – 2000

1142 – 2004

1144 – 2010

1146 – 2014

1150 – 2020

1152 – 2024

Команды подпрограммы расположить, начиная с ячейки 2000:

2000 – INC R1

2002 – RTI

2004 – INC R2

2006 – RTI

2010 – INC R3

2012 – RTI

2014 – INC R4

2016 – RTI

2020 – INC R5

2022 – RTI

2024 – INC R1

Выполнить программу покомандно, аналогично пп. 5.1. Пояснить вычисление команд и принятие содержимого общих регистров и стека.

Литература

1. Малые ЭВМ и их применение / Дедов Ю.А., Островский М.А., Песелев К.В. и др.; Под ред. Б.М. Наумова. – М.: Статистика, 1980. – 231 с.
2. Основы программирования на Ассемблере для СМ ЭВМ / Вигдорчик Г.В., Воробьёв А.Ю., Праченко В.Д. – М.: Финансы и статистика, 1983. – 256с.
3. Соучек Б. Микропроцессоры и микроЭВМ: Пер. с англ. / Под ред. А.И.Петренко. – М.: Сов. радио, 1980. – 520 с.
4. Экхауз Р., Моррис Л. Мини-ЭВМ: Организация и программирование: Пер. с англ. / Под ред. Г.П. Васильева. – М.: Финансы и статистика, 1983. – 359 с.
5. Уокерли Дж. Архитектура и программирование микроЭВМ: В 2-х кн. Пер. с англ. – М.: Мир, 1984. – Кн. 2, 341 с.

СОДЕРЖАНИЕ

1. Цель лабораторных работ	3
2. Описание лабораторной установки.....	3
2.1. Общая характеристика микроЭВМ.....	3
2.2. Структура центрального процессора	3
2.3 Форматы данных	9
2.4 Слово состояния процессора	9
2.5 Организация памяти	10
2.6 Способы адресации операндов.....	11
2.7. Система команд.....	23
3. Эмулятор МП машин класса PDP-11. ver 18.02.....	50
Руководство пользователя.....	50
4. Задания к лабораторным работам	52
4.1. Изучение основных команд работы с пультовым терминалом	52
4.2. Изучение способов адресации	53
4.3. Выполнение команд процессором.....	55

4.4. Организация переходов.....	58
4.5. Использование стека при выполнении арифметических операций	59
4.6. Переход на подпрограмму	59
Литература	64