

**У С Т Р О Й С Т В О
СПЕЦИАЛИЗИРОВАННОЕ
УПРАВЛЯЮЩЕЕ ВЫЧИСЛИТЕЛЬНОЕ
«ЭЛЕКТРОНИКА ДЗ-28»**

Справочник программиста

И5М3.857.100 ДЗ



СОДЕРЖАНИЕ

	Стр.
1. Введение	2
2. Организация памяти	5
3. Ручные операции с пульта	9
3.1. Типы клавиш пульта и индикация	9
3.2. Режимы работы	9
3.3. Включение питания	10
3.4. Коды клавиатуры	10
3.5. Непрограммируемые операции	11
3.6. Кодирование и выполнение команд	12
3.7. Запуск программы	12
4. Размещение данных в ОЗУ	13
4.1. Регистры и ячейки	13
4.2. Организация десятичных ячеек	15
4.3. Восемьбайтовые ячейки	16
4.4. Двухбайтовые ячейки	17
4.5. Однобайтовые ячейки	17
5. Методы адресации	17
5.1. Методы адресации данных	17
5.2. Адресация программы	21
6. Команды обработки данных	21
6.1. Формирование десятичных чисел в регистрах	21
6.2. Команды пересылок и обмена для десятичных ячеек	22
6.3. Арифметические операции в десятичных ячейках	24
6.4. Математические функции для десятичных чисел	28
6.5. Обработка данных байтового формата	28
6.6. Обработка данных двухбайтового и восьмибайтового формата	30
6.7. Обработка символьных последовательностей	34
6.8. Контроль программы и данных	36
6.9. Специальные команды	37
7. Команды управления и отладки программы	38
7.1. Служебные команды	38
7.2. Ветвления, безусловные переходы и подпрограммы	39
8. Команды управления НМЛ	42
9. Управление периферией и пишущей машиной	46
10. Обслуживание прерываний	52
11. Справочные таблицы	54
12. Дополнения	55
12.1. Особенности выполнения некоторых команд	55
12.2. Дополнительные команды	55
12.3. Управление расширенной памятью	74
13. Лист регистрации изменений	76

1. ВВЕДЕНИЕ

1.1. В настоящем справочнике программиста приведены сведения, необходимые при разработке и отладке программ на машинном и машинно-ориентированных языках для устройства специализированного управляющего вычислительного «Электроника ДЗ-28» (ДЗ-28) всех исполнений.

1.2. Справочник распространяется на ДЗ-28, выпущенные после 1 октября 1979 года, от предыдущих модификаций они отличаются:

- возможностью использования блоков ОЗУ ёмкостью 16 или 32 Кбайт;
- возможностью работы (со сменными блоками) при длительности такта исполнения микрокоманды $1 \cdot 10^{-6}$ с или $2 \cdot 10^{-6}$ с;
- ускоренным исполнением «быстрых» команд;
- наличием дополнительных команд обслуживания периферии и НМЛ;
- возможностью дистанционного запуска при включении питания;
- принципиальной возможностью исключения случаев «зависания» магистрали ввода-вывода;

- отсутствием блокировок записи данных в служебную память микропрограмм.

Последнее обстоятельство требует повышенного внимания программиста.

1.3. Пользователя не должно смущать обилие команд, в основном обусловленное требованием обеспечения программной совместимости с устройством управляющим вычислительным 15ВСМ-5.

Для составления эффективных программ достаточно некоторого подмножества из системы команд, выбираемого исходя из времени выполнения команды и требований, предъявляемых к программе (рекурсивность, перемещаемость и т.д.).

Приведённые в справочнике рекомендации по использованию команд не являются обязательными для программиста и в конкретных программах могут не быть оптимальными по каким-либо критериям.

1.4. В связи с необходимостью вывода шестнадцатеричных чисел на цифровую индикацию вместо обычного обозначения шестнадцатеричных цифр **0, 1, 2, 3, ..., A, B, C, D, E, F** использованы группы из двух десятичных цифр, соответственно **00, 01, 02, 03, ..., 10, 11, 12, 13, 14, 15**.

Для облегчения чтения шестнадцатеричных цифр группы, как правило, разделены точками. Например: **00.12.05.10 = 0C5A**.

Для часто встречающегося шестнадцатеричного кода байта в соответствии со способом вывода его на индикацию принята запись двумя группами цифр, разделённых пробелом. Для шестнадцатеричных цифр (тетрад) байта в тексте принято общее обозначение: **B** – для старшей цифры и **A** – для младшей цифры. Биты байта в соответствии с их двоичными весами обозначаются через **b8, b4, b2, b1, a8, a4, a2, a1**.

1.5. Ниже приведена словесная расшифровка основных мнемокодов.

Расшифровка общепринятых обозначений математических функций, специальных, редко встречающихся в программе команд, мнемочкодов, расшифрованных в тексте или состоящих из сокращений нескольких мнемочкодов, не приводится.

ABGE (*addition one and BGE*) – прибавление единицы и ветвление, если больше или равно;

ADD (*addition*) – сложение;

AND – конъюнкция;

ANS (*analysis*) – анализ;

ATOI (*alteration to integer*) – преобразование в целочисленное;

BBIC (*branch if bit is clear*) – ветвление, если бит очищен;

BBIS (*branch if bit is set*) – ветвление, если бит установлен;

BEQ (*branch if equal*) – ветвление, если равны;

BEV (*branch if even*) – ветвление, если чётный;

BHIS (*branch if higher or same*) – ветвление, если больше или тождественно;

BGE (*branch if greater or equal*) – ветвление, если больше или равно;

BKEY (*branch if key*) – ветвление, если клавиша;

BLT (*branch if less than*) – ветвление, если меньше чем;

BMER (*branch if machine error*) – ветвление, если машинная ошибка;

BMI (*branch if minus*) – ветвление, если минус;

BNE (*branch if not equal*) – ветвление, если не равно;

BPER (*branch if programming error*) – ветвление, если программная ошибка;

BPL (*branch if plus*) – ветвление, если плюс;

BR (*branch*) – ветвление безусловное;

BSA (*branch if same*) – ветвление, если тождественно;

CAP (*cartesian in polar*) – декартовы в полярные;

CLR (*clear*) – очистка;

CMD (*command*) – команда;

COM (*complement*) – инвертирование;

DEG (*degree*) – градус;

DIG (*digital*) – цифра;

DIV (*divide*) – делить;

END – конец;

E (*exponent*) – показатель степени;

FOR (*forward*) – вперёд;

GO – пуск;

QRT (*quadrate*) – квадрат;

INP (*input*) – ввод;

INT (*integer*) – целое;

INV (*invert*) – обратная величина;

JMM (*jump to mark*) – передача управления метке;

JMP (*jump*) – передача управления;

JMT (*jump to table*) – передача управления по таблице;
LOAD – загружать;
MARK – метка;
MUL (*multiply*) – умножать;
MOV (*move*) – пересылать;
NEG (*negative*) – отрицание;
OUT (*output*) – вывод
OR – дизъюнкция;
POINT – точка;
PRINT – печать;
RES (*residue*) – остаток;
REW (*rewind*) – назад;
RTI (*return from interrupt*) – возврат из прерывания,
RTII (*RTI imaginary*) – псевдовозврат из прерывания;
RTS (*return from subroutine*) – возврат из подпрограммы;
RTSI (*RTS imaginary*) – псевдовозврат из подпрограммы;
SAVE – хранить;
SQR (*square*) – квадратный корень;
SOB (*subtract one and brunch*) – вычитание единицы и ветвление;
SUB (*subtract*) – вычитание;
SWA (*swap*) – обмен;
VER (*verify*) – контроль;
WAIT – ожидание;
XOR (*exclusive OR*) – исключающее или.

1.6. Условные обозначения, принятые в тексте.

(A) – ячейка с адресом **A**;
B1 A1 – код однобайтовой команды;
B1 A1 B2 A2 – код двухбайтовой команды;
BD – регистр базового адреса данных (индексный регистр данных);
BP – регистр базового адреса программы (индексный регистр программы);
ПМ – пишущая машина;
ВЫВ – регистр выходного байта «вывод»;
МЛ – магнитная лента;
НМЛ – накопитель на магнитной ленте;
ОЗУ – оперативное запоминающее устройство;
РС – программный счётчик;
((P)) – содержимое ячейки с адресом, равным **(P)**;
R_i, T_i – двухбайтовый регистр с номером $0 \leq i \leq 15$;
S_i – однобайтовый регистр с номером $0 \leq i \leq 15$;
SP – указатель стека;

УПР – регистр выходного байта «управление»;

X, Y, Z – десятичные регистры;

() – содержимое регистра или ячейки;

← – пересылка;

↔ – обмен.

1.7. Перед работой на ДЗ-28 и разработкой программ следует изучить следующие эксплуатационные документы:

– формуляр И5М3.857.100 ФО;

– инструкцию по эксплуатации И5М3.857.100 ИЭ;

– техническое описание М3.857.100 ТО.

Примечание. В период проводимой модернизации ДЗ-28 возможна поставка вместо указанных документов паспорта И5М3.857.100 ПС и паспорта И5М3.857.100-01.01 ПС.

1.8. В **разделе 11** звёздочкой * отмечены команды, отсутствующие в предыдущих реализациях микропрограммного обеспечения.

1.9. Последние модификации ДЗ-28 отличаются возможностью использования блоков ОЗУ объёмом 128 К и расширенной системой команд. Сведения о них приведены в **разделе 12**.

2. ОРГАНИЗАЦИЯ ПАМЯТИ

2.1. Адрес байта ОЗУ задаётся четырьмя шестнадцатеричными цифрами.

Младшая цифра адреса называется разрядом.

2.2. Условно ОЗУ разбито на две зоны: рабочую (область программы и данных) и служебную (псевдорегистры и данные микропрограмм).

2.3. Служебная зона ОЗУ расположена в последних 256 байтах ОЗУ.

Для ОЗУ объёмом 16 К эта зона имеет адреса от 3.15.00.00 до 3.15.15.15, для ОЗУ объёмом в 32 К от 7.15.00.00 до 7.15.15.15. Вся остальная зона ОЗУ считается рабочей.

В дальнейшем старшую цифру адресов служебной зоны будем обозначать знаком X.

Структура служебной зоны ОЗУ представлена на **рис. 1**.

Ячейки служебной зоны в отличие от ячеек рабочей зоны в дальнейшем именуется регистрами.

Доступная пользователю с помощью регистровой адресации часть служебной зоны выделена на **рис. 1** жирными линиями.

Адресация к данным микропрограмм, размещённых в выделенной штриховкой области, не допускается, так как может привести к непредсказуемым для пользователя последствиям.

Не рекомендуется также программирование записи в регистры R14 и R15, хранящие признаки состояния, маски прерывания и программный счётчик.

		00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
X.15.00	B A	R0		R1		R2		R3		R4		R5		RR	R6		R7
X.15.01	B A	R8		R9		R10		R11		R12		R13 (SP)		R14		(Прn)	R15 (PC)
X.15.02	B A	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
X.15.03	B A	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
X.15.04	B A	R0 ÷ R15															
X.15.05	B A	Уровень 1 (Пр4, Пр2, Пр1)															
X.15.06	B A	R0 ÷ R15															
X.15.07	B A	Уровень 2 (Пр8)															
X.15.08	B A	R0 ÷ R15															
X.15.09	B A	Уровень 4 (УП2, УП1)															
X.15.10	B A	BP		BD		BP		BD		BP		BD		BP		BD	
						Уровень 1				Уровень 2				Уровень 4			
X.15.11	B A	Wy															
		Wx															
X.15.12	B A	Z															
		Zx															
X.15.13	B A	Uy															
		Ux															
X.15.14	B A	Vy															
		Vx															
X.15.15	B A	Y															
		X															

Рис. 1. Структура служебной зоны

Некоторые сведения о содержимом регистров R14 и R15 для справки приведены ниже.

Изменение зоны уровней прерывания в прерывающих программах в принципе допустимы для квалифицированного программиста.

2.4. В адресах X.15.00.00...X.15.01.15 расположены регистры R00...R15, занимающие по 2 соседних байта каждый.

В регистрах R байт с меньшим адресом является старшим. Знак числа кодируется старшим битом b8 старшего байта.

Регистры R00...R12 используются для операций шестнадцатеричной арифметики.

Регистр R13 (SP) хранит указатель стека. При включении питания и при нажатии клавиши «C» в SP записывается адрес X.13.00.00.

2.5. Регистры R14 и R15, структура которых показана на рис. 2, используются для хранения служебных признаков, масок и программного счётчика.

2.6. Единичное состояние ПрЦ соответствует процессу формирования десятичного числа в регистре X и вызывает микропрограмму приведения числа к машинному виду перед выполнением любой команды, отличной от команд формирования числа, описанных в подразделе 6.1.

Единичное состояние Пр0 вызывает останов программы (сброс ПрП) после пересчёта (PC) на следующий адрес.

Нулевое состояние ПрП соответствует режиму останова или выполнения команд с внешней или встроенной клавиатуры при неизменном (PC), единичное – работе по программе.

Значение УПВ и масок приведены в разделе 10.

Признак ненулевой маски ПрМ равен дизъюнкции всех бит маски внешних прерываний.

Установка ПрЖ в единицу при ненулевой маске и ПрМ=1 вызывает закливание микропрограммы на ожидании сигналов прерывания (см. WAIT в разделе 10).

2.7. Каждый из двухбайтовых регистров R8...R15 разбит на два однобайтовых подрегистра S. Регистров S шестнадцать – от S0 до S15.

2.8. Шестнадцать регистров T расположены на адресах X.15.02.00...X.15.03.15.

Каждый из регистров T расположен в двух байтах, адреса которых отличаются на 16. Например, регистр T9 расположен в байтах X.15.02.09 и X.15.03.09.

В регистр T0 микропрограмма заносит адрес сбоя.

Восьмибайтовый регистр RR образован четырьмя регистрами R4÷R7.

2.9. Зона от X.15.04.00 до X.15.09.15 используется для запоминания R0...R15 при внешних прерываниях.

2.10. В байтах X.15.10.00 и X.15.10.01 хранится базовый адрес программы BP, в байтах X.15.10.02 и X.15.10.03 – базовый адрес данных BD.

При внешних прерываниях BD и BP запоминаются на адресах X.15.10.04...X.15.10.15.

		R14		R15	
		S12	S13	S14	S15
B	8	ПрЦ	Пр8	ПрП	A1
	4	ПрО	Пр4	A3	
	2	N	Пр2		
	1	ПрФ	Пр1		
A	8	ПрН	ПрЖ	A2	A0
	4	УПВ	ПрМ		
	2		УП1		
	1		УП2		

PC

ПрЦ – признак формирования десятичного числа;

ПрО – признак останова;

N – маска внутреннего прерывания;

ПрФ – признак подмикропрограмм;

ПрН – признак внутреннего прерывания;

УПВ – код уровня внешнего прерывания;

Пр8, Пр4, Пр2, Пр1 – биты маски соответствующих сигналов внешних прерываний;

УП2, УП1 – биты маски внешних прерываний от пишущей машины

ПрП – признак работы по программе;

ПрЖ – признак команды WAIT;

ПрМ – признак ненулевой маски внешних прерываний;

PC – программный счётчик, $(PC) = A3.A2.A1.A0$.

Рис. 2. Структура регистров R14, R15

2.11. Рабочими регистрами микропрограмм являются:

$W_y, W_x, Z_x, U_x, V_y, V_x$ (см. рис. 1).

2.12. Команды десятичного формата используют регистры:

Y – адресá X.15.15 (00...15)B;

X – адресá X.15.15 (00...15)A;

Z – адресá X.15.12 (00...15)B.

2.13. В регистре U_y хранятся рабочие признаки. Признак **ПрГ** является признаком передачи функций клавиатуры периферийному устройству и хранится в U_y 13.

2.14. Следует отметить, что в связи с унификацией микропрограмм исполнений ДЗ-28 с объёмом ОЗУ 16 и 32 Кбайт в первом из них физические ячейки памяти имеют два адреса, различающихся на константу $4.00.00.00_{16} = 16\ 384_{10}$.

Например, адресá 3.04.05.06 и 7.04.06.06 в ОЗУ объёмом 16 Кбайт указывают на одну и ту же ячейку.

3. РУЧНЫЕ ОПЕРАЦИИ С ПУЛЬТА

3.1. Типы клавиш пульта и индикация

3.1.1. Клавиши пульта ДЗ-28 функционально можно разделить на четыре типа:

- а) клавиш режима работы: **Р**, **В**, **ПВ**, **П**;
- б) клавиши, непосредственно устанавливающие начальный адрес микропрограммы: **С**, **КП**, **НШ**, **НМ**, **ПШ**, **ИШ**, **ШН**, **ЗЛ**;
- в) кодовые клавиши (клавиши прямого кодирования и символные клавиши);
- г) клавиша **Ш_х**.

3.1.2. Для индикации результатов вычислений на пульте имеется два регистра индикации **Y** и **X** по 16 позиций в каждом.

3.2. Режимы работы

3.2.1. В ДЗ-28 реализованы четыре режима работы аппаратуры. Соответствующий режим работы устанавливается при нажатии одной из клавиш **Р**, **В**, **ПВ**, **П**.

3.2.2. В режиме «**В**» производят ввод программы в ОЗУ с клавиатуры или с периферийного устройства, принявшего на себя функции клавиатуры.

Кроме клавиши **В**, этот режим можно установить с периферийного устройства подачей нулевого уровня сигнала **ВППП**.

3.2.3. В режиме «**ПВ**» производят ввод программы в ОЗУ с клавиатуры с одновременной пошаговой распечаткой программы.

3.2.4. Режим «**П**» позволяет распечатать записанную в ОЗУ программу.

В режимах «**В**», «**ПВ**» и «**П**» на индикацию в регистр **Y** выводится (**РС**) в шестнадцатеричной форме, в регистр **X** – номер шага, равный разности (**РС**) – (**ВР**), переведённой в десятичную форму, и код байта, записанного по адресу (**РС**).

3.2.5. Операции, не выделенные в специальный режим, производятся в режиме «Р». В режиме «Р» при остановленном ДЗ-28 на индикацию выводится содержимое регистров Y и X.

3.2.6. Запуск распечатки программы в режиме «П» осуществляется нажатием любой кодовой клавиши с адреса, равного (РС).

Распечатка останавливается перед номером шага, кратным 50, или по коду 0512. Остановить распечатку можно клавишей Ш.

3.2.7. Если ДЗ-28 находится на индикации, то в режимах «ПВ» и «П» нажатие клавиши Ш вызывает распечатку содержимого одного адреса ОЗУ.

3.2.8. В режиме «ПВ» ввод кода в ОЗУ осуществляется с выводом на ПМ.

3.3. Включение питания

3.3.1. При включении питания $BD \leftarrow 0.04.00.00$, $BP \leftarrow 0$, $SP \leftarrow X.13.00.00$, $PC \leftarrow (BP)$; $X \leftarrow 0$, $Y \leftarrow 0$, обнуляются все маски и признаки, отключается НМЛ, $УПР \leftarrow 00.04$, $ВВ \leftarrow 0$, $ВЫВ \leftarrow 00.15$.

Если в течение ~650 000 тактов после этого от ПУ поступит синхроимпульс СИП, то микропрограммно ДЗ-28 переходит к выполнению команды GR1 (адрес ПУ). При этом регистр ВЫВ устанавливается в соответствии с поступившим в ДЗ-28 по СИПу кодом. Если СИП не поступает, то устройство ДЗ-28 выходит на индикацию при $УПР = 0$.

3.3.2. При выполнении микропрограммы индикации $ВВ = 0$, $ПрП = 0$, $УПР = 0$.

3.3.3. При нажатии кодовой клавиши пульта ДЗ-28 переходит от микропрограммы индикации к выполнению соответствующей микропрограммы по обработке поступившего кода. При выполнении команды $ВВ \leftarrow 1$ и индикация погашена. После выполнения команды ДЗ-28 возвращается к выполнению микропрограммы индикации. Нажатие клавиши, снабжённой индикатором, не приводит к запуску какой-либо микропрограммы.

3.4. Коды клавиатуры

3.4.1. Восьмибитный код, вводимый в ДЗ-28 с клавиатуры, условно разбит на две части В и А, являющиеся шестнадцатеричными цифрами.

3.4.2. Любой из кодов от 0000 до 1515 может быть введён с помощью клавиш прямого кодирования (верхнего ряда). Составляющую кода В набирают на четырёх клавишах с индикаторами верхнего ряда в коде 8421. Если около клавиши включён индикатор, соответствующий бит установлен. Изменение состояния индикатора производится нажатием клавиши.

Ввод кода производится нажатием одной из клавиш 00...15, определяющей часть А кода.

3.4.3. Коды некоторых команд могут быть введены без использования клавиш прямого кодирования, с символьных клавиш.

В разделе 11 приведены коды, поступающие от кодовых клавиш пульта, имеющих символику.

3.5. Непрограммируемые операции

3.5.1. Клавиши, непосредственно устанавливающие начальный адрес микропрограммы, код в ДЗ-28 не вводят, но их нажатие прерывает выполнение любой команды и запускает соответствующую микропрограмму.

3.5.2. По нажатию клавиши **КП** (контроль программы) сумма кодов, записанных от адреса, указанного в **BP**, до адреса, на котором записана команда **END**, помещается в регистр **X**.

Программный счётчик **PC** при этом указывает на адрес команды **END**. Если команда **END** отсутствует, включается индикатор **ОП**.

При вычислении контрольной суммы складываются тетрады **B** и **A** колов байтов в десятичной системе. Максимальное приращение к контрольной сумме, получаемое суммированием кода 1515, равно 30.

3.5.3. Пять цифр, вводимых после нажатия клавиши **НШ** (номер шага), являются цифрами номера шага.

После того как ввод номера шага закончен, введённое число переводится в шестнадцатеричную систему, результат складывается с (**BP**) и записывается в программный счётчик **PC**.

Не допускается вводить номер шага, больший 32767, так как микропрограммой перевод в шестнадцатеричную систему чисел, больших 32767, не предусмотрен.

3.5.4. Клавиша **НМ** (найти метку) служит для поиска меток **MARK** в ОЗУ. Ищется метка с кодом, введённым после нажатия клавиши **НМ**, от адреса, равного (**BP**), до адреса, на котором записана команда **END** или до адреса, равного (**SP**). **PC** устанавливается на адрес после найденной метки. Если в указанном диапазоне ОЗУ метки нет, включается индикатор **ОП**.

3.5.5. Нажатие клавиши **ПШ** (прибавить шаг) вызывает сдвиг на один шаг содержимого ОЗУ от адреса, равного текущему состоянию счётчика, до адреса, на котором записана команда **END**, в сторону увеличения адреса.

На текущем шаге записывается код 0514.

3.5.6. По нажатию клавиши **ИШ** (исключить шаг) на один шаг сдвигается содержимое ОЗУ от адреса, равного (**PC**) + 1, до адреса, на котором записана команда **END**, в сторону уменьшения адреса.

Содержимое текущего адреса при этом теряется. Следует иметь в виду возможность появления при редактировании двухшаговых команд ложных **END** и игнорирования истинного.

При нажатии клавиши **ШН** (шаг назад) содержимое **PC** уменьшается на единицу.

Если команда **END** в зоне от адреса, равного текущему содержимому **PC**, до адреса, равного содержимому **SP**, отсутствует, то по нажатию клавиш **ПШ** и **НШ** добавление и исключение шагов происходит с включением индикатора **ОП**. Если (**PC**) \geq (**SP**), то включается индикатор **ОП**, сдвига содержимого ОЗУ не происходит.

3.5.7. По нажатию клавиши «С» $SP \leftarrow X.13.00.00$, $PC \leftarrow (BP)$; $X \leftarrow 0$; $Y \leftarrow 0$, обнуляются все признаки и маски, отключается НМЛ. ДЗ-28 выходит на индикацию при $УПР = 0$.

3.5.8. Нажатие клавиши **ЗЛ** (запись на ленту) вызывает запись на МЛ содержимого ОЗУ с адреса, равного текущему (PC), по адрес, на котором записана команда **END**.

Если **END** от адреса, равного (PC), до конца рабочей зоны ОЗУ отсутствует, запись ведётся до конца рабочей зоны, после записи включается индикатор **ОП**. Начальный адрес в служебной зоне ОЗУ задавать нельзя, так как запись служебной зоны на МЛ не предусмотрена.

3.5.9. По нажатию всех клавиш, непосредственно устанавливающих начальный адрес микропрограмм, $УПР \leftarrow 0$, НМЛ отключается.

3.6. Кодирование и выполнение команд

3.6.1. Команда кодируется одним (**В1А1**) или двумя байтами (**В1А1** и **В2А2**). Запуск микропрограмм выполнения команд осуществляется в режиме «**Р**» одним из трёх способов: вводом кодов с клавиатуры или с ПУ, которому передано управление; по программе; по нажатию клавиши **Ш** (шаг).

3.6.2. Для выполнения команд по программе или по шагам (по нажатию клавиши **Ш**) коды команд предварительно должны быть введены в ОЗУ в режимах «**В**» или «**ПВ**» или в режиме «**Р**» с МЛ или ПУ.

При вводе в ОЗУ с клавиатуры в режимах «**В**» или «**ПВ**» код записывается по адресу, указанному в PC и индицируемому в регистре Y ; по окончании ввода кода содержимое PC увеличивается на единицу.

3.6.3. Не рекомендуется подавать с клавиатуры команды, осуществляющие безусловные и условные переходы и ветвления.

3.6.4. Если ДЗ-28 выполняет микропрограмму индикации, то нажатие клавиши **Ш** в режиме «**Р**» вызывает выполнение команды, адрес которой записан в PC .

Команда выполняется полностью за один шаг, независимо от того, однобайтовая она или двухбайтовая.

3.6.5. При нажатии клавиши **Ш** в режимах «**В**», «**ПВ**», «**П**» содержимое PC увеличивается на единицу.

3.7. Запуск программы

3.7.1. Запуск программы с текущего (PC) осуществляется командой **GO** (клавиша **S**) в режиме «**Р**». Другие виды запуска рассмотрены в последующих разделах.

3.7.2. Выполнение программы может быть остановлено нажатием клавиши **Ш**. После остановки PC указывает на адрес первой невыполненной команды.

При неудачной попытке остановки клавишей **Ш** (зацикливание на команде **BR . – 0**, незавершающееся обращение к НМЛ, ожидание ответа ПУ и т.п.) вывод на индикацию рекомендуется производить клавишей **ШН**.

4. РАЗМЕЩЕНИЕ ДАННЫХ В ОЗУ

4.1. Регистры и ячейки

4.1.1. Все доступные пользователю регистры реализованы микропрограммно в общем поле ОЗУ. Кроме регистров масок, в служебной зоне ОЗУ ДЗ-28 имеются:

- три десятичных регистра X , Y и Z ;
- шестнадцать двухбайтовых регистров $R0 \div R15$, из которых тринадцать ($R0 \div R12$) находятся в полном распоряжении пользователя;
- шестнадцать однобайтовых регистров $S0 \div S15$, являющихся подрегистрами $R8 \div R15$;
- шестнадцать двухбайтовых регистров $T0 \div T15$, из которых $T0$ используется микропрограммой для записи адреса сбоя, остальные находятся в полном распоряжении пользователя;
- восьмибайтовый регистр RR ;
- индексные регистры данных BD и программы BP .

4.1.2. При обработке данных содержимое десятичных регистров воспринимается как десятичное число со знаком в форме с плавающей запятой, двухбайтовых – как целое шестнадцатеричное число со знаком. Знак плюс кодируется нулём, минус – единицей.

На рис. 3 приведены форматы записи чисел в регистры и преобразование форматов при передачах $(X) \rightleftharpoons (RR)$, $(R) \rightleftharpoons (T)$. На этом рисунке в регистрах X и RR записано десятичное число

$$X = \underbrace{(3n\ m)}_{\substack{\text{знак} \\ \text{мантиссы}}} \underbrace{m_1 m_2 m_3 \dots m_{12}}_{\text{мантисса}} \cdot 10^{\underbrace{(3n\ P)}_{\text{знак}} \underbrace{P_2 P_1}_{\text{порядок}}}$$

Знаки десятичных чисел кодируются двоичными кодами:

0000 для знака *плюс* и

0001 для знака *минус*.

В регистрах R и T записано целое шестнадцатеричное число

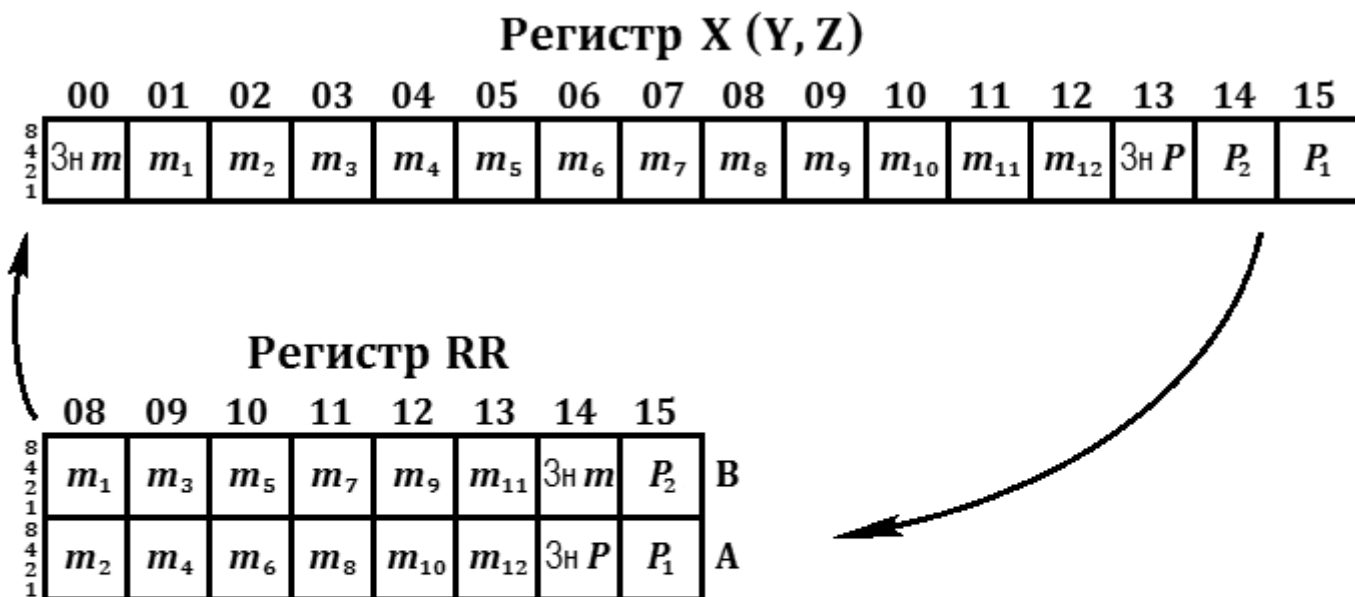
$$n = (3n\ n) n_3 n_2 n_1 n_0.$$

Знак кодируется в старшем бите старшего байта нулём (знак *плюс*) или единицей (знак *минус*).

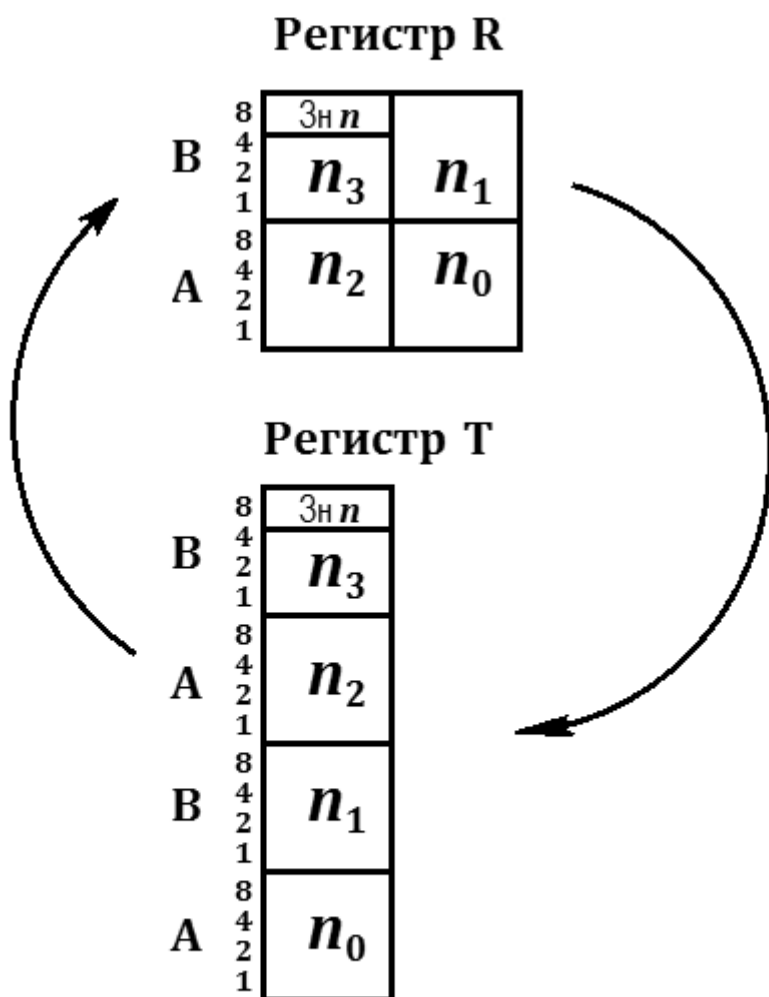
Следует отметить, что пересылки без изменения системы счисления могут применяться с любым содержимым регистров и ячеек.

4.1.3. Методы адресации, реализованные в ДЗ-28, позволяют обращаться к четырём типам ячеек ОЗУ:

- однобайтовым;
- двухбайтовым;
- восьмибайтовым;
- десятичным.



а) регистры для десятичных чисел



б) регистры для шестнадцатеричных чисел

Рис. 3. Форматы записи чисел в регистры

4.2.2. Адресация **CD** в командах задаётся десятичными номерами, начиная с нуля. При этом чётные номера соответствуют ячейке, размещённой в старших тетрадах (**B**) разрядов ОЗУ, нечётные – в младших (**A**).

Адрес ячейки $A(0)$ – адрес её нулевого разряда – индексируется тремя старшими разрядами содержимого регистра **BD**:

$$A(0) = (BD)' + (16 \text{ INT}(N/2))_{16}, \text{ где } N \text{ – десятичный номер } CD.$$

$(BD)' = 16 \text{ INT}(BD/16)$, то есть содержимое **BD** с нулевым младшим разрядом. В качестве $A(0)$ берутся пятнадцать младших бит суммы. **OP** при переполнении не включается.

На рис. 4 приведён фрагмент организации десятичных ячеек для ОЗУ в 32 Кбайт при $(BD) = 0$ с записанным в **CD 2** числом $-0,734\,291\,487\,652 \cdot 10^{-67}$, а в **CD 7** – содержимого **X** по команде **MOV X, CD 7**. В скобках для примера дана нумерация ячеек при $(BD) = 0.00.05.04$.

4.2.3. Номер ячейки **CD** при заданном (BD) и адресе её нулевого разряда $A(0)$ можно определить по формулам:

$$N_B = \left(\frac{A(0) - (BD)'}{8} \right)_{10} \quad \text{– для чётной ячейки}$$

$$N_A = \left(\frac{A(0) - (BD)'}{8} \right)_{10} + 1 \quad \text{– для нечётной ячейки}$$

4.3. Восьмибайтовые ячейки

4.3.1. Структура восьмибайтовой ячейки и способ записи числа в ней совпадает со структурой и способом хранения числа в восьмибайтовом регистре **RR**.

Ячейка адресуется заданием адреса её старшего байта $A(1)$. Адрес $A(1)$ может быть любым. На рис. 5 приведён пример восьмибайтовой ячейки с адресом $A(1) = 2.05.06.09$. В ячейке записано десятичное число $-\pi = -0.314159265359 \cdot 10^1$.

Адрес восьмибайтовой ячейки не индексируется.

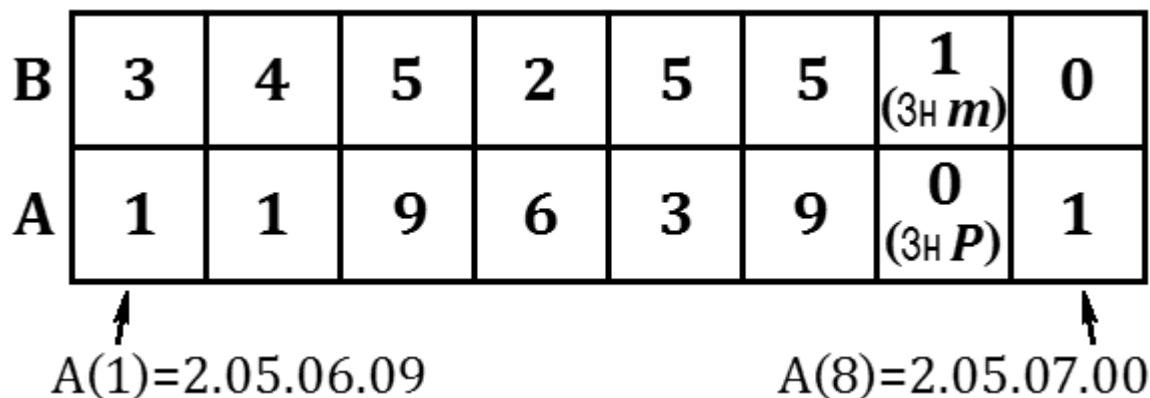


Рис. 5. Пример восьмибайтовой ячейки

4.3.2. С помощью адресации к восьмибайтовой ячейке обычно производится запись десятичных чисел в стек.

4.4. Двухбайтовые ячейки

4.4.1. Двухбайтовые ячейки используются для хранения шестнадцатеричных чисел и адресов (кодов). Структура и способ хранения числа в двухбайтовой ячейке совпадает со структурой и способом хранения чисел в двухбайтовом регистре **R**.

Под знак числа отведён бит **b8** старшего байта ячейки.

4.4.2. Ячейка адресуется заданием адреса **A(1)** её старшего байта.

В зависимости от метода адресации адрес индексируется или не индексируется (**BD**). Выбор адреса **A(1)** произволен.

На **рис. 6** приведён пример двухбайтовой ячейки с абсолютным адресом **A(1) = 3.10.11.12**, хранящей число $(-01.02.03.04)_{16}$ или код 09.02.03.04.

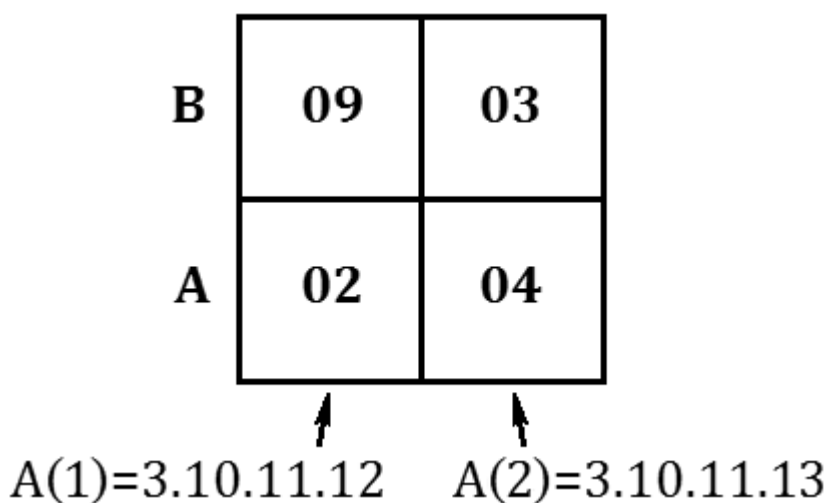


Рис. 6. Пример двухбайтовой ячейки

4.5. Однобайтовые ячейки

4.5.1. Однобайтовые ячейки подобны регистрам **S**.

4.5.2. Ячейка адресуется заданием адреса байта в ОЗУ.

В зависимости от применённого метода адресации адрес индексируется или не индексируется (**BD**).

5. МЕТОДЫ АДРЕСАЦИИ

5.1. Методы адресации данных

5.1.1. Методы адресации данных, используемые в ДЗ-28, приведены в **табл. 1**.

В **табл. 2** приведены примеры выполнения команд с различными методами адресации.

5.1.2. Операции над двумя операндами (двухместные) выполняются с помощью команд, в которых имеются два адреса.

Если в мнемокоде команды указаны два адреса, вторым указывается адрес операнда-приёмника, по которому будет занесён результат операции, первый адрес называется адресом операнда-источника.

Таблица 1

Метод	Обозначение	Содержание
Регистровый десятичный	X, Y или Z	Операнд в одном из регистров X, Y или Z соответственно.
Регистровый восьмибайтовый	RR	Операнд в регистре RR ($R4; R5, R6, R7$).
Регистровый по регистру R	R	Операнд в регистре R .
Регистровый по регистру T	T	Операнд в регистре T .
Регистровый по регистру S	S	Операнд в регистре S .
Косвенно-регистровый десятичный с индексированием $(BD)'$	$@Y$	Операнд в десятичной ячейке CD , номер которой равен (Y) относительно $(BD)'$.
Косвенно-регистровый	(R)	Операнд в восьмибайтовой ячейке, адрес которой равен (R) .
Косвенно-регистровый с индексированием (BD)	$@R$	Операнд в ячейке, для которой $(R) + (BD)$ равно адресу ячейки.
Автоинкрементный	$(R) +$	Операнд в двухбайтовой ячейке, адрес которой равен (R) ; после выполнения операции (R) увеличивается на 2.
Автодекрементный	$-(R)$	До операции содержимое регистра R уменьшается на 2, операнд в двухбайтовой ячейке, адрес которой равен новому (R) .
Непосредственный	$\#d$ или $\#l$	Операнд входит в код команды, d – второй байт команды, l – старшая тетрада второго байта команды.
Прямой с индексированием	C	Операнд в десятичной ячейке, номер которой равен $(10 \cdot B2 + A2)_{10}$ относительно $(BD)'$.

Команды, по которым обрабатывается один операнд, называются одноместными.

На рис. 7 показаны последовательности операций выполнении команды с различными методами адресации.

Непосредственный и прямой методы адресации, в которых операнд или его адрес входят в код команды, на рис. 7 не показаны.

5.1.3. При вычислении адреса ячейки по косвенно-регистровому методу с индексированием (BD) адрес ячейки является суммой кодов выбранного регистра и регистра BD . Знаковые разряды участвуют в сложении как старшие разряды чисел. Перенос из знакового разряда теряется, индикатор ОП при этом не включается.

Учитывая это, в команде можно указать на ячейку, адрес которой меньше (BD) . Например, при $(BD) = 02.01.00.00$ и содержимом регистра, указывающего адрес ячейки, равным 15.12.14.10 по косвенно-регистровому методу адресации с индексированием (BD) будет указана ячейка, адрес которой 01.13.14.10.

5.1.4. При косвенно-регистровом десятичном методе с индексированием $(BD)'$ программист не должен допускать указания номера ячейки, большего 4095, из-за непредсказуемости результата вычисления адреса.

Таблица 2

Мнемокод	Код B1A1 B2A2	Метод адресации источника	Тип ячейки (регистра) источника	Метод адресации приёмника	Тип ячейки (регистра) приёмника	Содержание
MOV X, Y	0604	X	Десятичная	Y	Десятичная	$Y \leftarrow (X)$
MOV Y, Z	1214	Y	— " —	Z	— " —	$Z \leftarrow (Y)$
MOV X, RR	1206	Y	— " —	RR	Восьмибайтовая	$RR \leftarrow (X)$
MOV X, R8	0413 1108	Y	— " —	R	Двухбайтовая	$R8 \leftarrow INT(X)$
MOV R3, T11	1412 0311	R	Двухбайтовая	T	Формат T-регистра	$T11 \leftarrow (R3)$
MOV S9, S0	1112 0900	S	Однобайтовая	S	Однобайтовая	$S0 \leftarrow (S9)$
MOV @Y, X	0505	@Y	Десятичная	X	Десятичная	$X \leftarrow (CD(Y))$
MOV X, (R9)	0412 0209	X	Десятичная	(R)	Восьмибайтовая	$(R9) \leftarrow (X)$
MOV S1, @R7	0912 0107	S	Однобайтовая	@R	Однобайтовая	$(R7) + (BD) \leftarrow (S1)$
MOV (R11)+, R8	1015 0811	(R) +	Двухбайтовая	R	Двухбайтовая	$R8 \leftarrow ((R11)); ((R11)) + 2$
MOV R10, -(R7)	1012 1007	R	— " —	-(R)	— " —	$(R7) - 2; (R7) \leftarrow (R10)$
MOV #9, S3	1303 0009	#d	—	S	Однобайтовая	$S3 \leftarrow 0009$
MOV X, 74	0404 0704	X	Десятичная	C	Десятичная	$CD74 \leftarrow (X)$

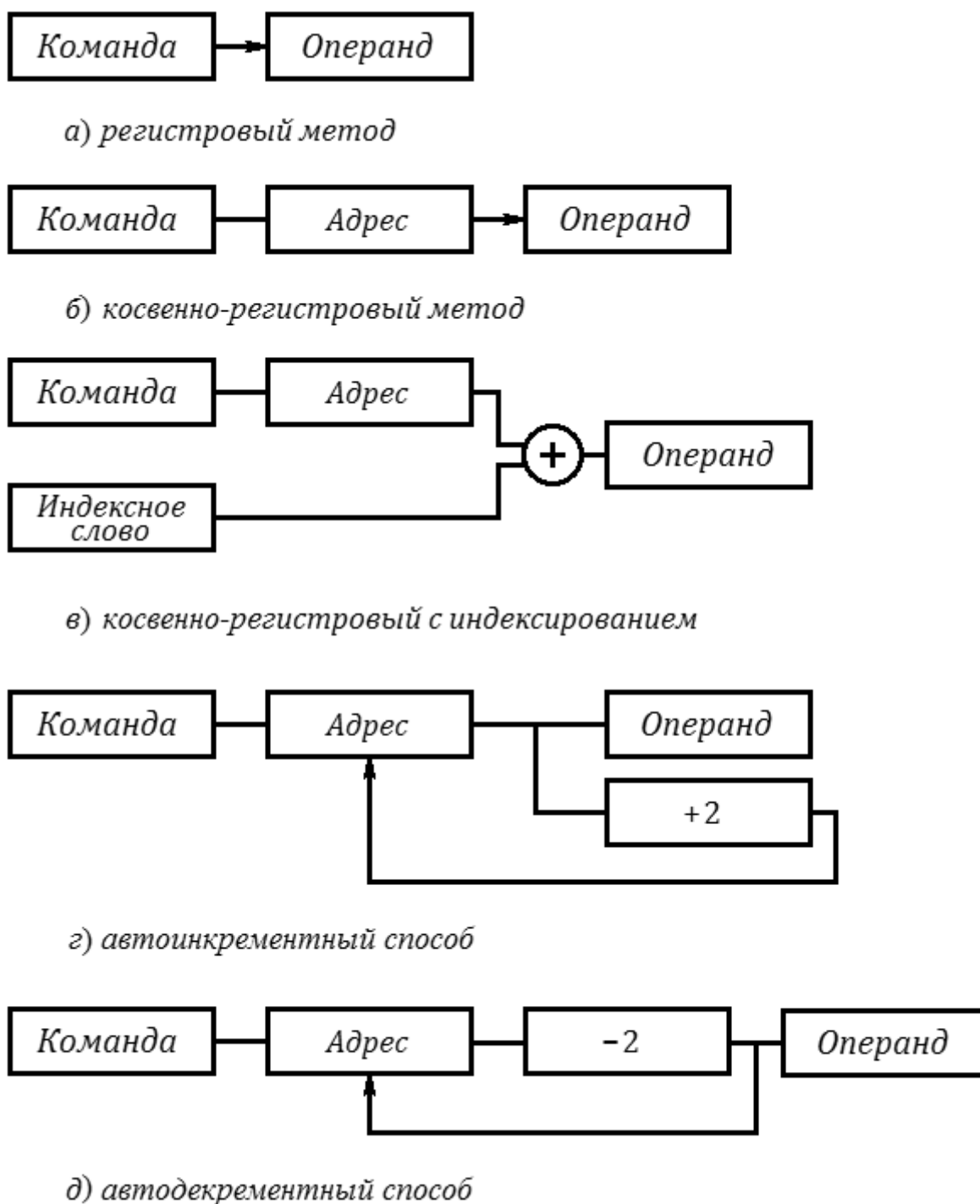


Рис. 7. Методы адресации

Так же, как и при косвенно-регистровом методе с индексированием (BD), может быть указана ячейка, адрес которой меньше (BD).

Например, при (BD) = 01.00.00.12, номере ячейки, равном 4092, адрес ячейки равен

$$16 \text{ INT} \frac{01.00.00.12}{16} + \left(16 \text{ INT} \frac{4092}{2} \right)_{16} = 01.00.00.00 + 07.15.14.00 = 08.15.14.00 .$$

Так как старший бит четырёхразрядного шестнадцатеричного числа в образовании адреса не участвует, то адрес ячейки равен 00.15.14.00.

5.1.5. В обозначении метода адресации и во всех таблицах, кроме табл. 2, в графе «Содержание» отсутствует информация об индексировании для простоты записи.

Но все методы адресации имеют отличные друг от друга обозначения и, кроме того, в таблицах определения команд введена специальная графа.

Адреса десятичных ячеек **CD** в табл. 2 могут быть найдены в соответствии с подразделом 4.2.

5.2. Адресация программы

Адресация программы в ОЗУ осуществляется относительно базового адреса программы (**BP**).

Относительно (**BP**) ищутся метки, вычисляется контрольная сумма, производится загрузка программы с МЛ и запись её на МЛ, осуществляются безусловные переходы.

Например, команда **JMP @X** осуществляет безусловный переход по адресу, равному $(BP) + (X)_{16}$.

Некоторые команды (типа **JMT**) адресуются к абсолютным адресам.

Сведения об индексации при переходах в программе приведены в таблицах раздела 7.

6. КОМАНДЫ ОБРАБОТКИ ДАННЫХ

6.1. Формирование десятичных чисел в регистрах

6.1.1. В табл. 3 приведены команды формирования числа в регистре **X**.

Таблица 3

Мнемокод	Код		Содержание
	B1A1	B2A2	
DIG l	07	l	Ввод десятичной цифры l ($0 \leq l \leq 9$) в регистр X
E	07	10	Сброс порядка (X), начало формирования порядка (X)
NEG X	07	11	Изменение знака мантиссы (X) или порядка (X)
POINT	07	12	Установка запятой (десятичной точки)
CLR X	07	15	Гашение (X)
ADD #10,E	04	12 0700	Прибавление 10 к порядку (X)
ADD #l,E	04	12 07 l	Прибавление l ($0 \leq l \leq 9$) к порядку (X)
SUB #10,E	04	12 0400	Вычитание 10 из порядка (X)
SUB #l,E	04	12 04 l	Вычитание l ($0 \leq l \leq 9$) из порядка (X)

6.1.2. По командам, следующим за последовательностью команд цифрового ввода **DIG A1**, **E**, **NEG X**, **POINT**, **CLR X** содержимое **X** приводится к машинному виду: погашенные разряды мантиссы заполняются нулями, в разряды порядка, если они погашены, записывается порядок введённого числа.

6.1.3. Разряды регистра **X**, как и любой десятичной ячейки, используются следующим образом:

- нулевой – для знака числа;
- разряды с первого по двенадцатый – для мантииссы;
- разряд тринадцатый для знака порядка;
- разряд четырнадцатый и пятнадцатый для порядка числа.

Ввод числа должен начинаться с цифры мантииссы или запятой.

Команда **POINT** (запятая) присваивает вводимому числу порядок, равный количеству значащих цифр до запятой. Незначащие нули до запятой игнорируются.

Командой **E** числу присваивается нулевой порядок. Если после **E** вводятся цифры, то двумя последними из них определяется величина порядка.

Если команда **NEG X** вводится до команды **E**, она меняет знак числа, если после, то знак порядка.

6.1.4. Возможны различные варианты ввода одного и того же числа. Например, число 100 можно ввести следующими способами:

DIG 1		DIG 1
DIG 0	или	E
DIG 0		DIG 3
или даже		
DIG 1		
DIG 0		
DIG 0		
E		
DIG 0		
DIG 3		

Как правило, самым быстрым является второй способ.

6.2. Команды пересылок и обмена для десятичных ячеек

6.2.1. В табл. 4 приведены команды пересылок и обмена для десятичных ячеек.

6.2.2. В командах обмена и пересылок используются все методы адресации десятичных ячеек: регистровый десятичный **X**, **Y** или **Z**, косвенно-регистровый десятичный с индексированием $(BD)' @Y$, прямой с индексированием $(BD)' C$.

6.2.3. Примеры пересылок и обмена для десятичных ячеек:

а) **MOV X,Y** (0604)

До выполнения операции:

(X) = 789; **(Y)** = 456

После выполнения операции:

(X) = 789; **(Y)** = 789

б) **MOV X,120** (0404 1200) при **BD = 0**

До выполнения операции

(X) = 432; **(CD 120)** = 0

Таблица 4

Мнемокод	Код		Содержание	Индексация
	B1A1	B2A2		
MOV X,C	0404	C	$CDC \leftarrow (X)$	$(BD)'$
MOV X,Y	0604		$Y \leftarrow (X)$	
MOV X,@Y	0504		$(Y) \leftarrow (X)$	$(BD)'$
MOV Y,C	0414	C	$CDC \leftarrow (Y)$	$(BD)'$
MOV Y,X	0605		$X \leftarrow (Y)$	
MOV Y,Z	1214		$Z \leftarrow (Y)$	
MOV C,X	0405	C	$X \leftarrow (CDC)$	$(BD)'$
MOV @Y,X	0505		$X \leftarrow ((Y))$	$(BD)'$
MOV C,Y	0415	C	$Y \leftarrow (CDC)$	$(BD)'$
MOV Z,Y	1215		$Y \leftarrow (Z)$	
SWA X,C	0406	C	$(CDC) \rightleftharpoons (X)$	$(BD)'$
SWA X,Y	0606		$(Y) \rightleftharpoons (X)$	
SWA X,@Y	0506		$(X) \rightleftharpoons ((Y))$	$(BD)'$

После выполнения операции:

$$(X) = 432; \quad (CD\ 120) = 432$$

Начальный адрес ячейки в ОЗУ равен

$$01.00 \cdot INT\left(\frac{120}{2}\right)_{16} - 0.03.12.00$$

в) MOV Y,15 (04.14.00.15) при $(BD) = 01.02.03.04$

До выполнения операции:

$$(Y) = 31; \quad (CD\ 15) = 7$$

После операции:

$$(Y) = 31; \quad (CD\ 15) = 31$$

Начальный адрес ячейки в ОЗУ равен

$$01.02.03.00 + 01.00 \cdot INT\frac{15}{2} = 01.02.10.00, \text{ ячейка расположена в младших тетрадах байтов.}$$

г) SWA X,@Y (0506) при $(BD) = 01.07.06.05$

До операции:

$$(X) = 314; \quad (Y) = 517; \quad (CD\ 517) = 4$$

После операции

$$(X) = 4; \quad (Y) = 517; \quad (CD\ 517) = 314$$

Начальный адрес ячейки, заданный методом @Y, равен

$$01.07.06.00 + 01.00 \cdot \left(INT\frac{517}{2} \right)_{16} =$$

$$= 01.07.06.00 + 01.00.02.00 = 02.07.08.00$$

Ячейка расположена в младших тетрадах байтов.

6.3. Арифметические операции в десятичных ячейках

6.3.1. В десятичных ячейках можно совершать четыре арифметических действия: слежение, вычитание, умножение, деление.

6.3.2. Для операций десятичной арифметики регистр **Z** не является адресуемым. Все остальные методы адресации десятичных ячеек могут быть использованы. Команды арифметических операций и вызова остатка приведены в табл. 5.

6.3.3. Пример сложения $2 + 3$ по команде **ADD X,Y** с указанием последовательности действий:

DIG 2; MOV X,Y; DIG 3; ADD X,Y

В результате этих действий $(Y) = 5$.

6.3.4. Если номер ячейки по косвенно-регистравному методу адресации с индексированием указан больше 9999 или меньше нуля, то выполнение команды прекратится, будет включён индикатор **ОП**.

Если при алгебраическом сложении первые разряды обоих операндов равны нулю, то происходит выход из микропрограммы сложения, операнд приёмника остаётся без изменения.

Например: $-0 + 0 = -0$

При умножении и делении перед выполнением операции анализ операндов на равенство нулю, как в сложении и вычитании, не производится.

Таблица 5

Мнемокод	Код		Содержание	Индексация
	B1A1	B2A2		
ADD X,C	0400	C	$CDC \leftarrow (CDC) + (X)$	(BD)'
ADD X,Y	0600		$Y \leftarrow (Y) + (X)$	
ADD X,@Y	0500		$(Y) \leftarrow ((Y)) + (X)$	(BD)'
SUB X,E	0401	C	$CDC \leftarrow (CDC) - (X)$	(BD)'
SUB X,Y	0601		$Y \leftarrow (Y) - (X)$	
SUB X,@Y	0501		$(Y) \leftarrow ((Y)) - (X)$	(BD)'
MUL X,C	0402	C	$CDC \leftarrow (CDC) \cdot (X)$	(BD)'
MUL X,Y	0602		$Y \leftarrow (Y) \cdot (X)$	
MUL X,@Y	0502		$(Y) \leftarrow ((Y)) \cdot (X)$	(BD)'
DIV X,C	0403	C	$CDC \leftarrow (CDC)/(X)$	(BD)'
DIV X,Y	0603		$Y \leftarrow (Y)/(X)$	
DIV X,@Y	0503		$(Y) \leftarrow ((Y))/(X)$	(BD)'
RES	0714		$X \leftarrow \text{остаток}$	

6.3.5. Результат арифметической операции может получиться с порядком $E = -0$, например:

а) POINT; DIG 0; DIG 9; MOV X,Y; DIG 2; MUL X,Y

После операции $(Y) = 0,18 \cdot 10^{-0}$

б) POINT; DIG 0; DIG 8; MOV X,Y; ADD X,Y

В результате $(Y) = 0,16 \cdot 10^{-0}$

в) DIG 4; E; NEG X; DIG 2; MOV X,Y; POINT; DIG 0; DIG 2; DIV X,Y

После операции $(Y) = 0,2 \cdot 10^{-0}$

6.3.6. Если в процессе выполнения операции произойдёт переполнение разрядной сетки, выполнение команды прекращается, включается индикатор ОП. При этом по адресу операнда-источника записан промежуточный результат операции, который нельзя использовать в дальнейших вычислениях как истинный. Возможность получения в арифметических операциях равных результатов, имеющих разные коды, типа 0.18E-0 и 0.18E+0 ограничивают использование команды BSA X,Y для сравнения на равенство.

6.3.7. После выполнения арифметической операции в регистре остатка V_y остаётся информация, пользуясь которой можно уточнить результат операции.

Содержимое регистра остатка можно поместить в X, пользуясь командой RES. Команда RES должна следовать за интересующей пользователя командой арифметики, так как регистр остатка используется в качестве рабочего регистра микропрограмм в ряде команд.

При алгебраическом сложении остаток имеет знак меньшего по абсолютной величине операнда. Остаток суммируется с результатом алгебраического сложения для увеличения разрядности результата и для уменьшения погрешности, которая в ряде случаев может оказаться значительной.

6.3.8. Примеры использования остатка при вычитании и сложении:

а) $123\ 456\ 769\ 123 + 32\ 198\ 765,4321 = 123\ 488\ 987\ 888;$

остаток 0,4321;

окончательный результат $123\ 488\ 987\ 888 + 0,4321 = 123\ 488\ 987\ 888,4321.$

б) $648\ 375\ 916\ 029 - 921\ 385\ 274,628 = 647\ 454\ 530\ 755;$

остаток $-0,628;$

окончательный результат $647\ 454\ 530\ 755 - 0,628 = 647\ 454\ 530\ 754,372.$

в) $100 - 99,9999999999 = 0,1 \cdot 10^{-8};$

остаток $-0,9 \cdot 10^{-9};$

окончательный результат $0,1 \cdot 10^{-8} - 0,9 \cdot 10^{-9} = 0,1 \cdot 10^{-9}$, то есть окончательный результат в десять раз меньше первоначального.

6.3.9. При умножении в регистре остатка записываются младшие разряды произведения со знаком.

Пример: $349\ 264\ 571\ 932 \cdot 513\ 914\ 216\ 056 = 0,181\ 238\ 353\ 096 \cdot 10^{24}$

Остаток $0,970\ 849\ 508\ 192 \cdot 10^{12}$

Окончательный результат: $181\ 238\ 353\ 096\ 970\ 849\ 508\ 192.$

Таблица 6

Мнемокод	Код B1A1 B2A2	Содержание	Допустимые значения аргумента	Область значений функции
ABS X	0607	$X \leftarrow (X) $	Весь диапазон чисел	$0 \leq (X)$
ACS	0806	$X \leftarrow \arccos(X)$	$-1 \leq (X) \leq 1$	$0 \leq (X) \leq \pi$
ANC	0814	$X \leftarrow \text{arch}(X)$	$1 \leq (X) \leq 0,999\,999\,999\,999 \cdot 10^{49}$	$0 \leq (X) \leq 113,519\,816\,737$
AHS	0813	$X \leftarrow \text{arsh}(X)$	$-629\,357 \leq (X) \leq 0,999\,999\,999\,999 \cdot 10^{49}$	$-13,815\,510\,557\,9 \leq (X) \leq 113,519\,816\,737$
ANT	0815	$X \leftarrow \text{arth}(X)$	$-1 < (X) < 1$	$-13,010\,791\,601\,6 \leq (X) \leq 13,010\,791\,601\,7$
ASN	0805	$X \leftarrow \arcsin(X)$	$-1 \leq (X) \leq 1$	$-\frac{\pi}{2} \leq (X) \leq \frac{\pi}{2}$
ATN	0807	$X \leftarrow \text{arctg}(X)$	Весь диапазон чисел	$-\frac{\pi}{2} \leq (X) \leq \frac{\pi}{2}$
CAP	0808	$\begin{cases} Y \leftarrow \sqrt{(X)^2 + (Y)^2} \\ X \leftarrow \text{arctg} \frac{(Y)}{(X)} \end{cases}$	$\begin{cases} (X) < 0,1 \cdot 10^{50} \\ (Y) < 0,1 \cdot 10^{50} \end{cases}$	$0 \leq (X) \leq 2\pi$
COS	0803	$X \leftarrow \cos(X)$	Весь диапазон чисел	$-1 \leq (X) \leq 1$
DEG	0801	$X \leftarrow (X) \cdot \frac{180}{\pi}$	$ (X) \leq 0,999\,999\,999\,999 \cdot 10^{47}$	$ (X) \leq 0,572\,957\,795\,13 \cdot 10^{99}$
EXP	0614	$X \leftarrow e^{(X)}$	$(X) \leq 227,955\,924\,206$	$0 \leq (X) \leq 0,999\,999\,999\,309 \cdot 10^{99}$
EXT	0613	$X \leftarrow 10^{(X)}$	$(X) \leq 98,999\,999\,999\,9$	$0 \leq (X) \leq 0,999\,999\,999\,769 \cdot 10^{99}$
QRT	0713	$X \leftarrow (X)^2$	$ (X) \leq 0,999\,999\,999\,999 \cdot 10^{49}$	$0 \leq (X) \leq 0,999\,999\,999\,998 \cdot 10^{98}$
HCS	0811	$X \leftarrow \text{ch}(X)$	$ (X) \leq 227,955\,924\,206$	$0 \leq (X) \leq 0,499\,999\,999\,654 \cdot 10^{99}$
HSN	0810	$X \leftarrow \text{sh}(X)$	$ (X) \leq 227,955\,924\,206$	$0 \leq (X) \leq 0,499\,999\,999\,654 \cdot 10^{99}$
HTN	0812	$X \leftarrow \text{th}(X)$	$ (X) \leq 227,955\,924\,206$	$0 \leq (X) \leq 1$
INT	0608	Сброс дробной части (X)	Весь диапазон чисел	Все целые числа
INV	0615	$X \leftarrow \frac{1}{(X)}$	$ (X) > 0,1 \cdot 10^{-98}$	$0,1 \cdot 10^{-98} \leq (X) \leq 0,1 \cdot 10^{98}$
LGT	0610	$X \leftarrow \lg(X)$	$(X) > 0$	$-100 \leq (X) \leq 99$
LOG	0611	$X \leftarrow \ln(X)$	$(X) > 0$	$-230,258\,509\,299 \leq (X) \leq 227,955\,924\,206$
NORM	1209	Нормализация (X)	В разрядах мантииссы и порядка десятичные цифры, в знаковых разрядах мантииссы и порядка нули или единицы	Весь диапазон чисел

Продолжение таблицы 6

Мнемокод	Код B1A1 B2A2	Содержание	Допустимые значения аргумента	Область значений функции
PI	0609	$X \leftarrow \pi$	—	—
POC	0809	$\begin{cases} Y = (Y) \sin(X) \\ X = (Y) \cos(X) \end{cases}$	(X) – любое $ Y < 0,1 \cdot 10^{99}$	$ X < 0,1 \cdot 10^{99}$ $ Y < 0,1 \cdot 10^{99}$
RAD	0800	$X \leftarrow (X) \cdot \frac{\pi}{180}$	Весь диапазон чисел	$ X \leq 0,174\,532\,925\,198 \cdot 10^{98}$
SQR	0612	$X \leftarrow \sqrt{(X)}$	$(X) \geq 0$	$0 \leq (X) \leq 0,316\,227\,660\,17 \cdot 10^{50}$
SIN	0802	$X \leftarrow \sin(X)$	Весь диапазон чисел	$-1 \leq (X) \leq 1$
SPCMD 0514	0412 0514	$X \leftarrow \frac{180}{\pi}$	—	—
SPCMD 0515	0412 0515	$X \leftarrow \frac{\pi}{180}$	—	—
TAN	08 04	$X \leftarrow \operatorname{tg}(X)$	Весь диапазон чисел	$ X \leq 0,204\,028\,391\,725 \cdot 10^{12}$

6.3.10. При делении (v_y) равно остатку от деления. Для определения последующих цифр частного необходимо разделить остаток на делитель, предварительно присвоив делителю нулевой порядок.

6.3.11. Пример получения 24 знаков результата деления с указанием последовательности действий (в качестве рабочей используется ячейка CD 0):

DIG 5; DIG 2; DIG 4; DIG 6; DIG 8; MOV X,Y; DIG 7; DIG 5;
DIG 9; DIV X,Y; E, MOV X,0; RES; SWA X,0; DIV X,0; MOV 0,X

В результате (Y) = 69,127 799 736 4; (X) = 0,953 886 693 017 · 10⁻¹⁰.

Окончательный результат 69,127 799 736 495 388 669 301 7.

6.4. Математические функции для десятичных чисел

6.4.1. В ДЗ-28 одной командой может быть выполнено вычисление ряда математических функций, перечисленных в табл. 6.

6.4.2. Все функции, за исключение CAP и POC, в качестве аргумента используют содержимое регистра X, результат вычислений или вызываемая константа также помещается в регистр X.

Для команд CAP и POC, осуществляющих преобразование координат, аргументы занимают как регистр X, так и регистр Y.

В тригонометрических функциях угол измеряется в радианах.

6.4.3. По команде NORM число, записанное в регистре X, приводится к машинному виду: мантисса сдвигается влево на столько разрядов, сколько нулей имеется до первого значащего разряда мантиссы, из порядка вычитается число, равное количеству произведённых сдвигов.

6.4.4. По команде INT обнуляются разряды числа, вес которых меньше 1.

Если до выполнения команды INT (X) = 542,7845, например, то после команды INT (X) = 542.

Тригонометрические функции имеют точность не ниже половины младшего разряда мантиссы, при значениях аргумента $0 \leq |X| \leq 2\pi$. За этим диапазоном результат может оказаться менее точным.

6.5. Обработка данных байтового формата

6.5.1. Ввод информации в байтовые ячейки обычно осуществляется через регистры S0...S15.

В байтовых ячейках могут быть произведены следующие операции: конъюнкция (\wedge), дизъюнкция (\vee), исключающее или (∇), дополнение до кода 1515 ($\bar{\quad}$).

6.5.2. Команды пересылок и обмена для однобайтовых ячеек приведены в табл. 7, команды логических операций в табл. 8.

В одноместных командах COM S_i и COM @R_i старшая цифра кода B2 может быть любой.

6.5.3. Как и для всех данных, адресация с индексированием для байтовых ячеек ведётся относительно (BD).

Таблица 7

Мнемокод	Код		Содержание	Индексация
	B1A1	B2A2		
MOV @R _i , S _j	0913	j i	$S_j \leftarrow ((R_i))$	(BD)
MOV S _i , S _j	1112	i j	$S_j \leftarrow (S_i)$	
MOV S _i , @R _j	0912	i j	$(R_i) \leftarrow (S_i)$	(BD)
MOV #d, S _i	13	i d	$S_i \leftarrow d, d$ – код второго байта команды	
SWA S _i , S _j	1114	i j	$(S_i) \rightleftharpoons (S_j)$	
SWA S _i , @R _j	0914	i j	$(S_i) \rightleftharpoons (R_j)$	(BD)
SWA S _i	0412	09 i	Обмен тетрад в S _i	

Таблица 8

Мнемокод	Код		Содержание	Индексация
	B1A1	B2A2		
OR S _i , S _j	1111	i j	$S_j \leftarrow (S_i) \vee (S_j)$	
OR S _i , @R _j	0911	i j	$(R_j) \leftarrow ((R_j)) \vee (S_i)$	(BD)
AND S _i , S _j	1108	i j	$S_j \leftarrow (S_i) \wedge (S_j)$	
AND S _i , @R _j	0908	i j	$(R_j) \leftarrow ((R_j)) \wedge (S_i)$	(BD)
XOR S _i , S _j	1109	i j	$S_j \leftarrow (S_i) \veebar (S_j)$	
XOR S _i , @R _j	0909	i j	$(R_j) \leftarrow ((R_j)) \veebar (S_i)$	(BD)
COM S _j	1110	B2 j	$S_j \leftarrow (\bar{S}_j)$	
COM @R _j	0910	B2 j	$(R_j) \leftarrow ((\bar{R}_j))$	(BD)

6.5.4. Примеры обработки данных байтового формата

- а) MOV @R2, S5 (0913 0502); (BD) = 2.10.15.06
 До операции: (R2) = 0.02.01.07
 Адрес ячейки = 2.10.15.06 + 0.02.01.07 = 2.13.00.13
 (S5) = 0315; (2.13.00.13) = 0402
 После операции: (R2) = 0.02.01.07; (S5) = 0402; (2.13.00.13) = 0402
- б) MOV S0, S7 (1112 0007)
 До операции: (S0) = 1513; (S7) = 0006
 После операции: (S0) = 1513; (S7) = 1513
- в) MOV #1513, S10; (1310 1513)
 До операции: (S10) = 0801
 После операции: (S10) = 1513
- г) SWA S1, @R3 (0914 0103); (BD) = 0.04.00.00
 До операции: (R3) = 01.07.02.04; (01.11.02.04) = 0702; (S1) = 1500

- После операции: (R3) = 01.07.02.04; (01.11.02.04) = 1500; (S1) = 0702
- д) **OR S10,S2** (1111 1002)
 До операции: (S10) = 1308; (S2) = 0301
 После операции: (S10) = 1308; (S2) = 1509
- е) **XOR S0,@R7** (0909 0007); (BD) = 0.00.00.00
 До операции: (R7) = 3.07.15.11; (3.07.15.11) = 0407; (S0) = 0512
 После операции: (R7) = 3.07.15.11; (3.07.15.11) = 0111; (S0) = 0512
- ж) **COM S4** (1110 0004)
 До операции: (S4) = 0208
 После операции: (S4) = 1307

6.6. Обработка данных двухбайтового и восьмибайтового форматов

6.6.1. Данные, хранимые в двухбайтовых ячейках и в регистрах R0...R15, в операциях участвуют как целые шестнадцатеричные числа со знаком. Числа в ячейки вводятся обычно через регистры R0 – R12. Первоначальная загрузка информации в регистры R0 – R12 может производиться через регистр X, однако более быстрым способом является использование команд **MOV #d, Si**.

Ввод в регистры R8...R15 следует производить, учитывая, что каждый из этих регистров состоит из двух однобайтовых регистров S.

Разряды числа в шестнадцатеричной форме записываются в регистры S по команде **MOV #d, Si**.

Например, для ввода в R9 числа $-16\ 125_{10}$ целесообразно проделать следующие действия:

```
MOV #11 14,S2
MOV #15 13,S3
```

Ввод чисел в регистры R0...R7 можно производить пересылкой чисел из других регистров.

6.6.2. В табл. 9 приведены команды пересылок между регистром X и двухбайтовыми регистрами R, а также между регистрами X и Y и восьмибайтовыми ячейками.

В методах адресации команд табл. 9 индексация не используется.

Пересылки по этим командам осуществляются с преобразованием формата, а по некоторым командам и с преобразованием системы счисления.

При пересылке (X) в регистр R дробная часть отбрасывается.

6.6.3. Команды пересылок и обмена для двухбайтовых ячеек приведены в табл. 10.

6.6.4. В операциях пересылок могут быть адресованы регистры BD (базового адреса данных) и BP (базового адреса программы).

6.6.5. Описание команд арифметических операций и простейших функций для двухбайтовых ячеек приведено в табл. 11.

Команд, обрабатывающих содержимое восьмибайтовых ячеек, в ДЗ-28 нет.

Таблица 9

Мнемокод	Код B1A1 B2A2	Содержание	Тип ячейки источника	Тип ячейки приёмника
MOVD R_i, X	0413 04 i	$X \leftarrow (R_i)_{10}$ (перевод в десятичную систему)	Двухбайтовая	Десятичная
MOVH X, R_i	0413 12 i	$R_i \leftarrow \text{INT}(X)_{16}$ (перевод в шестнадцатеричную систему)	Десятичная	Двухбайтовая
MOV R_i, X	0413 03 i	$X \leftarrow (R_i)$	Двухбайтовая	Десятичная
MOV X, R_i	0413 11 i	$R_i \leftarrow \text{INT}(X)$	Десятичная	Двухбайтовая
MOV RR, X	1207	$X \leftarrow (RR)$	Восьмибайтовая	Десятичная
MOV X, RR	1206	$RR \leftarrow (X)$	Десятичная	Восьмибайтовая
MOV $(R_i), X$	0412 03 i	$X \leftarrow ((R_i))$	Восьмибайтовая	Десятичная
MOV $X, (R_i)$	0412 02 i	$(R_i) \leftarrow X$	Десятичная	Восьмибайтовая
MOV $(R_i), Y$	0412 01 i	$Y \leftarrow ((R_i))$	Восьмибайтовая	Десятичная
MOV Y, R_i	0412 00 i	$(R_i) \leftarrow Y$	Десятичная	Восьмибайтовая

Таблица 10

Мнемокод	Код B1A1 B2A2	Содержание	Индексация
MOV BD, R_i	0413 13 i	$R_i \leftarrow (BD)$	
MOV BP, R_i	0413 14 i	$R_i \leftarrow (BP)$	
MOV R_i, BD	0413 05 i	$BD \leftarrow (R_i)$	
MOV R_i, BP	0413 06 i	$BP \leftarrow (R_i)$	
MOV R_i, R_j	1104 ij	$R_j \leftarrow (R_i)$	
MOV R_i, T_j	1412 ij	$T_j \leftarrow (R_i)$	
MOV $R_i, -(R_j)$	1012 ij	$R_j \leftarrow (R_i) - 2; (R_j) \leftarrow (R_i)$ (запись в стек)	
MOV $R_i, @R_j$	0904 ij	$(R_j) \leftarrow (R_i)$	(BD)
MOV $@R_i, R_j$	0905 ji	$(R_j) \leftarrow ((R_i))$	(BD)
MOV $(R_i)+, R_j$	1015 ji	$R_j \leftarrow ((R_i)); R_i \leftarrow (R_i) + 2$ (вызов из стека)	
MOV T_i, R_j	1413 ij	$R_j \leftarrow (T_i)$	
SWA R_i, R_j	1106 ij	$R_i \rightleftharpoons (R_j)$	
SWA $R_i, @R_j$	0906 ij	$R_i \rightleftharpoons ((R_j))$	(BD)
SWA R_i	0412 08 i	Обмен байтов R_i	

6.6.6. В результате арифметических действий над содержимым двухбайтовых ячеек и при пересылках из X в R включается индикатор ОП, если в регистр R записывается число, большее по модулю, чем 7.15.15.15.

6.6.7. В качестве индексного регистра в косвенных методах адресации для двухбайтовых ячеек, как и для всех данных, используется регистр BD .

Таблица 11

Мнемокод	Код		Содержание	Индексация
	B1A1	B2A2		
ABS R_i	0413	08 i	$R_i \leftarrow (R_i) $	
CLR R_i	0413	10 i	$R_i \leftarrow 0$	
NEG R_i	0413	09 i	$R_i \leftarrow -(R_i)$	
ADD R_i, R_j	1100	ij	$R_j \leftarrow (R_j) + (R_i)$	
ADD $R_i, @R_j$	0900	ij	$R_j \leftarrow ((R_j)) + (R_i)$	(BD)
ADD $\#e, R_i$	1000	ei	$R_i \leftarrow (R_i) + e$	
MUL R_i, R_j	1102	ij	$R_j \leftarrow (R_j) \cdot (R_i)$	
MUL $R_i, @R_j$	0902	ij	$(R_j) \leftarrow ((R_j)) \cdot (R_i)$	(BD)
SUB R_i, R_j	1101	ij	$R_j \leftarrow (R_j) - (R_i)$	
SUB $R_i, @R_j$	0901	ij	$(R_j) \leftarrow ((R_j)) - (R_i)$	(BD)
SUB $\#e, R_i$	1001	ei	$R_i \leftarrow (R_i) - e$	

Примечание. В командах ADD $\#e, R_i$ и SUB $\#e, R_i$ R_i рассматривается как код, переполнение (заём) игнорируется.

6.6.8. Пользуясь командами пересылок MOV $R_i, -(R_j)$ и MOV $(R_i)+, R_j$, программист может организовать стек.

Содержимое X и Y может быть также запомнено в стеке и вызвано из стека командами MOV X, (R_i) , MOV $(R_i), X$, MOV Y, (R_i) и MOV $(R_i), Y$. Но при этом (R_i) не пересчитывается микропрограммно. Пересчёт можно организовать, пользуясь командами ADD $\#e, (R_i)$ и SUB $\#e, (R_i)$, учитывая, что (X) запоминается в восьми байтах.

В качестве указателя стека может быть использован любой регистр. Если использовать регистр R13, то необходимо помнить, что в стеке с этим указателем микропрограммно запоминается адрес при обращении к подпрограмме, а при возврате из подпрограммы из этого стека восстанавливается (PC).

6.6.9. Пример обработки данных двухбайтового формата:

а) MOV X, R0 (0413 1100)

До операции: (X) = -1234; (R0) = 02.07.15.11

После операции: (X) = -1234; (R0) = 09.02.03.04

б) MOVN X, R1 (0413 1201)

До операции: (X) = 1234; (R1) = 00.00.00.00

После операции: (X) = 1234; (R1) = 00.04.13.02

$$1234_{10} = 0.04.13.02_{16}$$

в) MOV X, RR (1206)

До операции: (X) = $0,448\,571\,619\,328 \cdot 10^{-54}$;

(R4) = 0; (R5) = 0; (R6) = 0; (R7) = 0

После операции: $(X) = 0,448\,571\,619\,328 \cdot 10^{-54}$;
 $(R4) = 04.04.08.05$; $(R5) = 07.01.06.01$;
 $(R6) = 09.03.02.08$; $(R7) = 00.01.05.04$

г) **MOV X, (R1)** (0412 0201)

До операции: $(X) = -123\,456\,789\,123 \cdot 10^{45}$; $(R1) = 03.09.08.07$
 $(03.09.08.07) = 1412$ $(03.09.08.11) = 0215$
 $(03.09.08.08) = 0000$ $(03.09.08.12) = 1513$
 $(03.09.08.09) = 0706$ $(03.09.08.13) = 0001$
 $(03.09.08.10) = 0503$ $(03.09.08.14) = 1515$

После операции: $(X) = -123\,456\,789\,123 \cdot 10^{45}$; $(R1) = 03.09.08.07$
 $(03.09.08.07) = 0102$ $(03.09.08.11) = 0901$
 $(03.09.08.08) = 0304$ $(03.09.08.12) = 0203$
 $(03.09.08.09) = 0506$ $(03.09.08.13) = 0100$
 $(03.09.08.10) = 0708$ $(03.09.08.14) = 0405$

д) **MOVD R2, X** (0413 0402)

До операции: $(R2) = 00.01.06.03$; $(X) = 0$

После операции: $(R2) = 00.01.06.03$; $(X) = 355$; $00.01.06.03_{16} = 355_{10}$

е) **MOV R6, X** (0413 0306)

До операции: $(R6) = 00.01.06.03$; $(X) = 0$

После операции: $(R6) = 00.01.06.03$; $(X) = 163$

ж) **MOV BP, R8** (0413 1408)

До операции: $(BP) = 00.07.08.03$; $(R8) = 15.00.01.02$

После операции: $(BP) = 00.07.08.03$; $(R8) = 00.07.08.03$

з) **MOV R10, BD** (0413 0510)

До операции: $(R10) = 02.00.10.03$; $(BD) = 0$

После операции: $(R10) = 02.00.10.03$; $(BD) = 02.00.10.03$

и) **MOV R6, R7** (1104 0607)

До операции: $(R6) = 11.09.01.15$; $(R7) = 15.00.00.11$

После операции: $(R6) = 11.09.01.15$; $(R7) = 11.09.01.15$

к) **MOV R11, T3** (1412 1103)

До операции: $(R11) = 15.13.14.12$; $(T3) = 0$

После операции: $(R11) = 15.13.14.12$; $(T3) = 5.13.14.12$

л) **MOV R2, -(R13)** (1012 0213)

По команде **MOV R2, -(R13)** осуществляется запись в стек, указатель которого хранится в **R13**.

До операции: $(R2) = 07.15.02.01$; $(R13) = 03.13.00.00$;
 $(03.12.15.14) = 0802$; $(03.12.15.15) = 0007$

После операции: $(R2) = 07.15.02.01$; $(R13) = 03.12.15.14$;
 $(03.12.15.14) = 0715$; $(03.12.15.15) = 0201$

м) **MOV (R13)+, R2** (1015 0213)

По команде **MOV (R13)+, R2** осуществляется вызов из стека. Указатель стека в **R13**.

До операции: **(R13)** = 03.12.15.14; **(R2)** = 15.15.15.15;
(03.12.15.14) = 0715; **(03.12.15.15)** = 0201

После операции: **(R13)** = 03.13.00.00; **(R2)** = 07.15.02.01;
(03.12.15.14) = 0715; **(03.12.15.15)** = 0201

н) **SWA R8, @R1** (0906 0301) при **(BD)** = 0

До операции: **(R8)** = 10.05.12.06; **(R1)** = 07.04.10.12;
(07.04.10.12) = 0409; **(07.04.10.13)** = 0800

После операции: **(R8)** = 04.09.08.00; **(R1)** = 07.04.10.12;
(07.04.10.12) = 1005; **(07.04.10.13)** = 1206

о) **CLR R9** (0413 1009)

До операции: **(R9)** = 08.00.04.05

После операции: **(R9)** = 0

п) **ADD R8, R9** (1100 0809)

До операции: **(R8)** = 04.01.10.03; **(R9)** = 00.01.06.02

После операции: **(R8)** = 04.01.10.03; **(R9)** = 04.03.00.05

р) **ADD #11, R1** (1000 1101)

До операции: **(R1)** = 04.08.02.01

После операции: **(R1)** = 04.08.02.12

с) **MUL R9, @R1** (0902 0901) при **(BD)** = 00.04.00.00

До операции: **(R9)** = 00.00.07.02; **(R1)** = 00.00.00.00;
(00.04.00.00) = 0000; **(00.04.00.01)** = 1503

После операции: **(R9)** = 00.00.07.02; **(R1)** = 00.00.00.00;
(00.04.00.00) = 0612; **(00.04.00.01)** = 0306

6.7. Обработка символьных последовательностей

6.7.1. В системе команд ДЗ-28 предусмотрены пять специальных команд обработки символьных последовательностей. Символьные последовательности в коде ГОСТ 13052-74, в частности, могут вводиться в ДЗ-28 с ПМ «Консул 260.1». Адреса в этих командах не индексируются.

6.7.2. Команда **ATOI d** (1003 **B2A2**) предназначена для преобразования числа в кодах ГОСТ 13052-74 с нулевым дополнительным разрядом, записанного в ОЗУ, в целое десятичное число в машинном виде с записью его в регистр **X**.

Чтение числа из ОЗУ начинается с адреса, записанного в **R1**. Цифры числа заносятся в регистр **X**, символы с кодом **B2A2** пропускаются. Первый отличный от **B2A2** и от кода цифр код или код тринадцатой цифры заносится в **S3**. Указатель **(R1)** устанавливается на адрес, следующий за адресом кода, занесённого в **S3**.

Пример чтения числа:

АТОI 0200

До операции: (R1) = 03.00.00.00; (S3) = 0213; (X) = 42

(03.00.00.00) = 0301

(03.00.00.01) = 0305

(03.00.00.02) = 0200

(03.00.00.03) = 0304

(03.00.00.04) = 0306

(03.00.00.05) = 0211

После операции: (R1) = 03.00.00.06; (S3) = 0211; (X) = 1546

(03.00.00.00) = 0301

(03.00.00.01) = 3305

(03.00.00.02) = 0200

(03.00.00.03) = 0304

(03.00.00.04) = 0306

(03.00.00.05) = 0211

6.7.3. По команде NSN *d* (1004 B2A2) с адреса, равного (R1), осуществляется поиск первого, не равного B2A2, кода. Найденный код записывается в S3, в R1 – адрес кода, следующего за найденным.

Например:

NSN 0200

До операции: (R1) = 03.04.15.02; (S3) = 0211

(03.04.15.02) = 0200

(03.04.15.03) = 0200

(03.04.15.04) = 0200

(03.04.15.05) = 0307

После операции: (R1) = 03.04.15.06; (S3) = 0307

(03.04.15.02) = 0200

(03.04.15.03) = 0200

(03.04.15.04) = 0200

(03.04.15.05) = 0307

6.7.4. Команда NSS *d* (1005 B2A2) аналогична NSN *d*, но ищется код, равный заданному.

Например:

NSS 0010

До операции: (R1) = 04.00.00.00; (S3) = 0305

(04.00.00.00) = 0200

(04.00.00.01) = 0200

(04.00.00.02) = 0010

После операции: (R1) = 04.00.00.03; (S3) = 0010

(04.00.00.00) = 0200

(04.00.00.01) = 0200

(04.00.00.02) = 0010

При пользовании командами **NSN** d и **NSS** d следует иметь в виду, что для них граница поиска кода не определена, и при отсутствии нужного кода в ОЗУ микропрограмма может заиклиться. Выход из цикла можно осуществить только одной из клавиш, непосредственно устанавливающих начальный адрес микропрограммы, например, клавишей **C**.

6.7.5. Команда **ANS** S_i, S_j (1006 ij) осуществляет анализ содержимого. Результат анализа заносится в S_j . Функция $S_j \leftarrow f(S_i)$ задана в табл. 12.

Таблица 12

(S_i)	(S_j)	Группа кодов ГОСТ 13052-74
0300 ÷ 0309	0001	Цифры
0401 ÷ 0510	0003	Латинские буквы
0600 ÷ 0715	0005	Русские буквы и забой
0000 ÷ 0200	0009	Служебные знаки
0800 ÷ 1115	0013	—
1200 ÷ 1515	0011	—
Прочие	0007	—

Таблица 13

(S_i)	(R_j)	Символ ГОСТ 13052-74
0208	0002 0001	(
0209	0001 0003)
0210	0004 0005	*
0211	0003 0007	+
0213	0003 0011	—
0215	0004 0015	/
0514	0005 0013	¬
Прочие	0000 0009	

6.7.6. Команда **PRIOR** S_i, R_j (1007 ij) аналогично **ANS** S_i, S_j , анализирует содержимое S_i . Результат анализа в R_j . Функция $R_j \leftarrow f(S_i)$ задаётся в табл. 13.

6.8. Контроль программы и данных

6.8.1. В системе команд ДЗ-28 предусмотрена возможность контроля сохранности хранимой в ОЗУ информации путём вычисления контрольных сумм.

6.8.2. По команде **VERX** (1201) сумма частей «В» и частей «А» кодов ОЗУ, записанных от адреса $(X)_{16} + (BP)$ до адреса, на котором записана команда **END**, заносится в регистр **X**. Код команды **END** в контрольную сумму не входит.

6.8.3. Контрольная сумма кодов ОЗУ от адреса $(R_i) + (BD)$ до адреса $(R_j) + (BD)$ включительно вычисляется командой **VERR B2 A2** (1404 **B2A2**) с занесением шестнадцатеричной контрольной суммы в регистры **R0** и **R1**.

При этом регистры **R0** и **R1** используются как один четырёхбайтовый регистр, старший разряд которого расположен в байте с меньшим адресом.

6.8.4. По команде **VEX** ij (1405 ij) в **S0** заносится контрольный байт, являющийся суммой по модулю 2 (исключающее или) всех байт от $(R_i) + (BD)$ до $(R_j) + (BD)$ включительно.

6.8.5. Из команд контроля команда **VEX** самая быстродействующая, по ней осуществляется операция «исключающее или» над всеми заданными байтами.

Самая медленная контролирующая команда – **VERX**, по которой осуществляется десятичное суммирование всех байт в регистре **X**.

Если при вычислении контрольной суммы по команде **VERX** в ОЗУ отсутствует команда **END**, начиная от адреса $(X)_{16} + (BP)$ до адреса 7.15.00.00, то включается индикатор **ОП**.

6.8.6. Примеры вычисления контрольных сумм:

- а) **VERX** (1201) при $(BP) = 01.00.00.00$
 До операции: $(X) = 256$
 $(01.01.00.00) = 0701$
 $(01.01.00.01) = 0611$
 $(01.01.00.02) = 0512$
 После операции: $(X) = 7 + 1 + 6 + 11 = 25$
 $(01.01.00.00) = 0701$
 $(01.01.00.01) = 0611$
 $(01.01.00.02) = 0512$
- б) **VERR 08 09** (1404 0809) при $(BD) = 02.00.00.00$
 До операции: $(R8) = 0; (R9) = 0.00.00.09;$
 $(R0) = 0; (R1) = 0$
 $(02.00.00.00) = 1515$
 $(02.00.00.01) = 1515$
 \vdots
 $(02.00.00.09) = 1515$
 После операции: $(R8) = 0; (R9) = 0.00.00.09$
 $(R0) = 0; (R1) = 0.01.02.12$
 $(02.00.00.00) = 1515$
 $(02.00.00.01) = 1515$
 \vdots
 $(02.00.00.09) = 1515$
- в) **VEX 10 11** (1505 1011) при $(BD) = 01.00.00.00$
 До операции: $(R10) = 0.00.00.05; (R11) = 0.00.00.07; (S0) = 0$
 $(01.00.00.05) = 1102$
 $(01.00.00.06) = 0007$
 $(01.00.00.07) = 0601$
 После операции: $(R10) = 0.00.00.05; (R11) = 0.00.00.07; (S0) = 1304$
 $(01.00.00.05) = 1102$
 $(01.00.00.06) = 0007$
 $(01.00.00.07) = 0601$

6.9. Специальные команды

В ДЗ-28 для обеспечения программной совместимости с 15VCM-5 имеются специальные команды:

SPCMD 0512	(0412 0512)
SPCMD 0513	(0412 0513)
SPCMD 0612	(0412 0612)
SPCMD 0613	(0412 0613)
SPCMD 0614	(0412 0614)
SPCMD 0713	(0412 0713)
SPCMD 0715	(0412 0715)

Вышеприведённые команды предназначены для программы вычисления тригонометрических функций и не рекомендованы для применения в новых программах ДЗ-28, в которой тригонометрические функции реализованы микропрограммно.

7. КОМАНДЫ УПРАВЛЕНИЯ И ОТЛАДКИ ПРОГРАММЫ

7.1. Служебные команды

7.1.1. К группе служебных команд управления программой относятся команды:

GO	(0514)
STOP	(0515)
MARK	(0408 <i>d</i>)
END	(0512)
CMD R _{<i>i</i>}	(0413 02 <i>i</i>)
PAUSE	(0412 0615)
PAUSER	(0412 1402)

7.1.2. Командой GO осуществляется запуск программы с текущего состояния программного счётчика PC (установкой признака работы по программе ПрП в единичное состояние).

7.1.3. Заранее предусмотренный останов программы (сброс признака ПрП) может быть осуществлён командой STOP. Программа останавливается с (PC), указывающим на адрес, следующий за командой STOP.

По командам GO и STOP регистр УПР и признак передачи управления периферийным устройством ПрГ очищаются.

Команда GO является «пустой» и может стоять в любом месте программы, в котором состояние регистра УПР может быть нулевым.

7.1.4. Любое место в программе может быть помечено командой MARK *d* (0408 B2A2). Код *d* служит для идентификации MARK, адресуемой командами поиска метки.

7.1.5. Командой END (0512) осуществляется останов программы, при этом PC ← ← (BD).

7.1.6. Для визуального контроля за ходом выполнения программы имеется возможность останова программы на определённое время.

Командой PAUSE (0412 0615) на индикацию в течение ~470 000 тактов выводится (Y) и (X).

По команде PAUSER (0412 1402) длительность паузы зависит от (R10) и примерно равна (R10) · 10 тактов, индикация погашена.

7.1.7. По команде CMD R_i может быть выполнена команда, код которой записан в регистре R_i . Если команда одношговая, её код должен быть записан в старшем байте R_i . Нельзя выполнять ввод числа, используя команду CMD R_i , если во вводе должно участвовать больше одной команды.

7.2. Ветвления, безусловные переходы и подпрограммы

7.2.1. Нормальная последовательность выполнения команд изменяется условными и безусловными переходами.

7.2.2. Ветвление осуществляется относительно программного счётчика PC.

В мнемосодах команд ветвлений присутствует буква B (от слова *branch* – ветвление); содержимое PC, соответствующее первому байту команды, обозначено точкой «.».

Если в мнемосоде команды не предусмотрено указание количества байтов, при выполнении условия осуществляется переход к команде, адрес которой для однобайтовых команд $.+3$, а для двухбайтовых $.+4$, то есть пропускаются два байта программы.

При невыполнении условия управление передаётся следующей команде, то есть команде с адресом $.+1$ для однобайтовых или $.+2$ для двухбайтовых команд. Если в командах ветвления сравнение производится с нулём, то в мнемосоде имеется буква Z (*zero* – ноль).

7.2.3. Команда BR $.-d$ (1402 B2A2) обеспечивает безусловную передачу управления по адресу $(PC) - d$, где $d = 16 \cdot B2 + A2 - 1$.

По команде BR $.+d$ (1403 B2A2) происходит безусловная передача управления по адресу $(PC) + d$, где $d = 16 \cdot B2 + A2 + 1$.

7.2.4. Команды ветвлений сгруппированы в следующих таблицах:

- табл. 14 – ветвление по анализу десятичных регистров;
- табл. 15 – ветвление по анализу операндов двухбайтового формата;
- табл. 16 – ветвление по анализу операндов однобайтового формата;
- табл. 17 – ветвление по битам и сигналам.

Все величины переходов команд ветвления указаны для команд, выполняемых по программе или по шагам.

Если команда ветвления по условию подана с клавиатуры, то, как правило, изменение (PC) не происходит при любом исходе проверки условия.

При выполнении безусловных переходов относительно (PC) BR $.+d$ и BR $.-d$ с клавиатуры текущее (PC) меняется на $16 \cdot B2 + A2$.

Команда BSA Y, X и BEQ Y, X отличаются друг от друга тем, что по BSA Y, X проверяется поразрядное совпадение кодов всех шестнадцати разрядов регистров Y и X,

а по **BEQ Y,X** производится анализ на нуль первого разряда разности $(Y) - (X)$. Команду **BSA Y,X** следует использовать при сравнении кодов. Для этой команды числа -0 и $+0$ не равны.

В командах **BEQZ** и **BNEZ** анализ на нуль (Y) или (X) осуществляется по первому разряду мантиссы.

Ветвление типа **BMI** и **BPL** осуществляется по знаку (Y) и (X) .

Таблица 14

Мнемокод	Код		Условие ветвления	(PC) при выполнении условия
	B1A1	B2A2		
BEQ Y,X	0509		$(Y) = (X)$ (проверка условия $(Y) - (X) = 0$)	.+3
BEQZ X	0412	0611	$(X) = 0$ (по первому разряду мантиссы)	.+4
BEQZ Y	0412	0411	$(Y) = 0$ (по первому разряду мантиссы)	.+4
BGE Y,X	0507		$(Y) \geq (X)$ (проверка условия $(Y) - (X)$ положительно)	.+3
BLT Y,X	0508		$(Y) < (X)$ (проверка условия $(Y) - (X)$ отрицательно)	.+3
BMI X	0412	0710	(X) отрицательно (по ненулевому знаковому разряду)	.+4
BMI Y	0412	0510	(Y) отрицательно (по ненулевому знаковому разряду)	.+4
BNEZ X	0412	0711	$(X) \neq 0$ (по первому разряду мантиссы)	.+4
BNEZ Y	0412	0511	$(Y) \neq 0$ (по первому разряду мантиссы)	.+4
BPL X	0412	0610	(X) положительно (по знаковому разряду)	.+4
BPL Y	0412	0410	(Y) положительно (по знаковому разряду)	.+4
BSA Y,X	1204		Код (Y) равен коду (X) (последовательное совпадение всех разрядов)	.+3

Таблица 15

Мнемокод	Код		Условие ветвления	(PC) при выполнении условия	Индексация
	B1A1	B2A2			
ABGE R_i, R_j	1409	ij	$R_j \leftarrow (R_j) + 1;$ $(R_j) \geq (R_j)$.+4	
BSAZ $R_i, +e$	1103	$e-1 i$	$(R_i) = 0$.+e, при $e = 1$ PC \leftarrow .+2	
BSAZ $@R_i, +e$	0903	$e-1 i$	$((R_i)) = 0$.+e, при $e = 1$ PC \leftarrow .+2	(BD)
BSA R_i, R_j	1107	ij	$(R_i) = (R_j)$.+4	
BSA $R_i, @R_j$	0907	ij	$(R_i) = ((R_j))$.+4	(BD)
BGE R_i, R_j	1410	ij	$(R_j) \geq (R_j)$.+4	
SOBZ $R_i, +e$	1408	$e-1 i$	$R_i \leftarrow (R_i) - 1;$ $(R_i) = 0$.+e, при $e = 1$ PC \leftarrow .+2	

Таблица 16

Мнемокод	Код		Условие ветвления	(PC) при выполнении условия	Индексация
	B1A1	B2A2			
BSA S_i, S_j	1115	ij	$(S_i) = (S_j)$.+4	
BSA $S_i, @R_j$	0915	ij	$(S_i) = (R_j)$.+4	(BD)
BEV $@R_i, +e$	1407	$e-1 i$	Количество единичных бит в байте по адресу (R_i) – чётное	.+e, при $e = 1$ PC \leftarrow .+2	(BD)
BNS $\#d, S1$	1008	d	$(S1) \neq d$.+4	
BNS $\#d, S3$	1009	d	$(S3) \neq d$.+4	
BHS S_i, S_j	1002	ij	$(S_i) \geq (S_j)$.+4	

По командам **SOBZ** $R_i, +e$ и **ABGE** R_i, R_j включается индикатор ОП, если в исходном состоянии содержимое анализируемых регистров отрицательное. Для команды **SOBZ** $R_i, +e$ также некорректна ситуация получения отрицательного (R_i) .

Вычитание единицы (в команде **SOBZ** $R_i, +e$) или прибавление единицы (в команде **ABGE** R_i, R_j) производится до анализа содержимого регистров.

В командах **BBIC** $i, (R_j)$, **BBIS** $i, (R_j)$ биты нумеруются от 0 до 7 соответственно от старшего к младшему.

Сигнал $\overline{P4}$, анализируемый командой **BKEY** $.+d$, поступает в ДЗ-28, если нажата третья слева клавиша на передней нижней части ПМ «Консул 260.1».

7.2.5. Команды безусловных переходов и организации подпрограмм имеют в своих мнемокодах одну или несколько букв из слова JUMP – передача управления.

Команды типа JUMP отличаются от команд типа BRANCH тем, что переходы по ним осуществляются не относительно счётчика команд.

7.2.6. Переходы типа JUMP бывают трёх типов:

– к метке (третья буква в мнемокоде – M), в PC заносится адрес, следующий за меткой **MARK** с заданным кодом команды, поиск производится от адреса (BP) до адреса, на котором записана команда **END** или до адреса, равного (SP);

– по содержимому регистра (третья буква в мнемокоде P или R), управление передаётся команде с адресом, равным $(BP) + (X)_{16}$ или $(BP) + (R_i)$ в соответствии с мнемокодом команды;

– по содержимому таблицы (третья буква в мнемокоде – T), управление передаётся команде, адрес которой записан в ячейке 7.13.d или 7.14.d, где d – код младшего байта команды.

При задании перехода по содержимому регистра X или (R_i) сумма $(BP) + (X)_{16}$ или $(BP) + (R_i)$ не должна превышать 32 767.

Задание отрицательного содержимого X для перехода не допускается.

В переходах по (R_i) знак (R_i) игнорируется.

Таблица 17

Мнемокод	Код B1A1 B2A2	Условие ветвления	(PC) при выполнении условия	Индек- сация
BBIC $i, @R_j$	1415 $i < 8 j$	Бит, номер которого i , в байте по адресу (R_j) равен нулю ($B2 < 8$)	. +4	(BD)
BBIS $i, @R_j$	1415 $i \geq 8 j$	Бит, номер которого i , в байте по адресу (R_j) равен единице ($B2 \geq 8$)	. +4	(BD)
BKEY . +d	1411 B2A2	$P4 = 1$ (спец. клавиша ПМ отката)	. +16B2 + A2 + 1	
BMER . +d	1414 B2A2	$OM = 1$, после анализа $OM = 0$. +16B2 + A2 + 1	
BPER	0510	$OP = 1$, после анализа $OP = 0$. +3	

7.2.7. Вторая буква **M** в мнемокоде команд типа *JUMP* означает переход без запоминания адреса обращения.

Если в мнемокоде команд типа *JUMP* вторая буква **S**, это означает обращение к подпрограмме с запоминанием адреса обращения.

7.2.8. При обращении к подпрограмме (**SP**) уменьшается на 2, затем по адресу, равному новому содержимому **SP**, записывается (**R15**), то есть (**PC**) и признак **ПрП** в момент обращения.

Для двухбайтовых команд программы в стек заносится адрес второго байта команды.

При запоминании (**PC**) и **ПрП** производятся те же действия, что по команде **MOV R15, -(R13)**.

7.2.9. Возврат из подпрограммы обычно осуществляется командой **RTS**.

В **R15** записывается содержимое ячейки, на которую указывает указатель стека, затем (**SP**) увеличивается на 2, то есть признак программы **ПрП** и (**PC**) восстанавливаются из стека, **SP** указывает на следующую ячейку стека

При возврате совершаются те же действия, что и по команде **MOV (R13)+, R15**.

Если **ПрП** установится в единицу, то управление передаётся программе с адреса (**PC**) + 1. В противном случае программа останавливается с (**PC**), восстановленным из стека.

7.2.10. Команды безусловных переходов и организации подпрограмм описаны в табл. 18.

8. КОМАНДЫ УПРАВЛЕНИЯ НМЛ

8.1. Информация может быть записана и считана с МЛ командами **SAVEX**, **LOADX**, **LOADP**.

Блок, записываемый на МЛ, должен завершаться командой **END**, в противном случае запись ведётся до конца рабочей зоны ОЗУ и включается индикатор **ОП**.

Таблица 18

Мнемокод	Код B1A1 B2A2	Содержание
JMM <i>d</i>	0407 <i>d</i>	$PC \leftarrow$ адрес команды, следующей за MARK <i>d</i> ; $ПрП \leftarrow 1$
JMP @X	1213	$PC \leftarrow (BP) + (X)_{16}$
JMP @R _{<i>i</i>}	0413 00 <i>i</i>	$PC \leftarrow (BP) + (R_i)$
JMTT <i>d</i>	1010 <i>d</i>	$PC \leftarrow (7.13.d)$; $ПрП \leftarrow 1$
JMTF <i>d</i>	1011 <i>d</i>	$PC \leftarrow (7.14.d)$; $ПрП \leftarrow 1$
JSM B1A1	00A1 или 01B1	$R13 \leftarrow (R13) - 2$; $(R13) \leftarrow$ ПрП, (PC); $Z \leftarrow (Y)$; $PC \leftarrow$ адрес команды, следующей за MARK B2A2, где код B2A2 команды MARK совпадает с кодом B1A1 команды JSM; $ПрП \leftarrow 1$
JSM B1A1	02A1 или 03B1	$R13 \leftarrow (R13) - 2$; $(R13) \leftarrow$ ПрП, (PC); $PC \leftarrow$ адрес команды, следующей за MARK B2A2, где код B2A2 команды MARK совпадает с кодом B1A1 команды JSM; $ПрП \leftarrow 1$
JSR @R _{<i>i</i>}	0413 01 <i>i</i>	$R13 \leftarrow (R13) - 2$; $(R13) \leftarrow$ ПрП, (PC); $PC \leftarrow (BP) + (R_i)$; $ПрП \leftarrow 1$
JSTT <i>d</i>	1013 <i>d</i>	$R13 \leftarrow (R13) - 2$; $(R13) \leftarrow$ ПрП, (PC); $PC \leftarrow (7.13.d)$; $ПрП \leftarrow 1$
JSTF <i>d</i>	1014 <i>d</i>	$R13 \leftarrow (R13) - 2$; $(R13) \leftarrow$ ПрП, (PC); $PC \leftarrow (7.14.d)$; $ПрП \leftarrow 1$
RTS	0511	$ПрП, PC \leftarrow ((R13))$; $PC \leftarrow (PC) + ПрП$; $R13 \leftarrow (R13) + 2$
RTSI	1212	$R13 \leftarrow (R13) + 2$
RTSGO	0412 0712	$ПрП, PC \leftarrow ((R13))$; $PC \leftarrow (PC) + ПрП$; $R13 \leftarrow (R13) + 2$; $ПрП \leftarrow 1$; $ПрГ \leftarrow 0$; $УПР \leftarrow 0$
TRAP	1205	Эквивалентно JSTF 0000

Начальный адрес зоны ОЗУ нельзя указывать в служебной зоне, так как это может привести к неправильному выполнению команды.

Если записывается на МЛ блок данных, то в самом блоке не должен встречаться код 0512, который может быть воспринят ДЗ-28 как команда END. При записи по команде SAVEX или клавише ЗЛ микропрограммно каждый байт дополняется контрольным битом, по которому при считывании осуществляется контроль байта на чётность.

После блока данных код команды END рекомендуется ставить два раза, так как в зависимости от кода, заверяющего блок данных, код 0512 может быть воспринят как второй байт двухбайтовой команды.

Если считываемый блок не помещается в рабочую зону ОЗУ, включается индикатор **ОП**, блок считывается до конца без записи не помещившихся байт в ОЗУ. Не следует задавать начальный адрес в служебной зоне ОЗУ, так как это может привести к непредсказуемым результатам.

Индикатор **ОМ** включается при считывании, если импульсы с МЛ не приходят в заданные промежутки времени, или подсчитанный контрольный бит не совпадает со считанным.

Перемотка магнитной ленты осуществляется командами **FORW** и **REW**. Выключение перемотки происходит по сигналу фотодатчика о ракорде. Применяемая магнитная лента должна иметь ракорд.

Выключение двигателей НМЛ может осуществляться также по команде **STTAP** или по сбросу **C**.

8.2. В табл. 19 приведены команды управления НМЛ.

Двухбайтовые команды управления с кодом первого байта 0412 могут быть употреблены пользователем для организации программной записи и считывания с МЛ.

Примечание. Работоспособность команд управления НМЛ с кодом первого байта 0412 зависит от модификации блока управления НМЛ, применённого в ДЗ-28.

8.3. Команды **SAVE R_i** и **LOAD R_i** предназначены, соответственно, для записи и чтения массивов данных при непрерывном движении МЛ.

Команда **SAVE R_i** записывает блок в 256 байт с начальным адресом, заданным в **R_i**. Каждый байт записывается восьмью битами (без контроля на чётность). Начальный адрес произволен. В начале и конце блока **SAVE R_i** записывает синхронизирующие паузы $\sim 2 \cdot 10^{-3}$ с.

8.4. Запись массива производится программой в следующей рекомендуемой последовательности:

- а) в цикле опрашивается завершение перемотки командой **CRFS**;
- б) включается лампа (**LAMP**) и электромагнит (**ELMG**) с последующей выдержкой $\sim 100 \cdot 10^{-3}$ с;
- в) включаются двигатели протяжки командой **TRTAP**;
- г) включается стирание ленты командами **SAVS** или **SAVC**;
- д) опрашивается ракорд (**CLDRS**), при его наличии ракорд «выбирается», то есть опрос производится до конца ракорда; задаётся пауза $\sim 300 \cdot 10^{-3}$ с;
- е) задаётся пауза записи межблочного промежутка и разгона двигателя $\sim 150 \cdot 10^{-3}$ с;
- ж) формируются в ОЗУ и записываются командами **SAVE R_i** последовательно все блоки массива;
- з) последовательностью команд **STTAP** и **ELMG** отключаются двигатели протяжки, с задержкой $\sim 100 \cdot 10^{-3}$ с. Командами **TRTAP** и **STTAP** НМЛ приводится в исходное состояние.

Таблица 19

Мнемокод	Код		Содержание
	B1A1	B2A2	
CLDRS	0412	1205	Опрос ракорда: $S0 \leftarrow \begin{cases} 0, & \text{если есть ракорд} \\ 1, & \text{если нет ракорда} \end{cases}$
CRFS	0412	1213	Опрос перемотки: $S0 \leftarrow \begin{cases} 0, & \text{если есть перемотка} \\ 1, & \text{если нет перемотки} \end{cases}$
ELMG	0412	1211	Включение электромагнита
INFS	0412	1207	Опрос ИИ: $S0 \leftarrow ИИ$
FORW	0412	1209	Перемотка вперёд
LAMP	0412	1202	Включение лампы
LOADP	0513		Загрузка в ОЗУ с МЛ; начальный адрес загрузки (PC) + ПрП
LOADR <i>i</i>	0412	10 <i>i</i>	Загрузка в ОЗУ с МЛ блока в 256 байт, начальный адрес загрузки (R_i)
LOADX	1202		Загрузка в ОЗУ с МЛ, начальный адрес загрузки (BP) + (X) ₁₆ .
REW	1200		Перемотка назад
SAVC	0412	1204	Запись нулевого бита
SAVER <i>i</i>	0412	11 <i>i</i>	Запись из ОЗУ на МЛ блока в 256 байт с адреса (R_i)
SAVEX	1203		Запись из ОЗУ на МЛ от адреса (BP) + (X) ₁₆ до адреса, на котором записан END
SAVS	0412	1212	Запись единичного бита
SMER	0412	1210	$OM \leftarrow 1$
SNCS	0412	1206	Опрос СИ: $S0 \leftarrow СИ$
STTAP	0412	1200	Останов двигателей, отключение лампы фотодатчика и электромагнита
TRTAP	0412	1203	Включение двигателей, отключение записи
WTRT	0412	1403	Поиск паузы, превышающей $(R10) \cdot 10$ тактов

- Примечания:*
1. При записи последующих массивов операции подпунктов 8.4-а и 8.4-д могут быть исключены.
 2. Лампу после анализа и «выборки» ракорда рекомендуется отключать для увеличения её срока службы командой STTAP. Команда STTAP, кроме лампы, отключает двигатели и электромагнит, поэтому после неё необходимо повторить команды ELMG, TRTAP, SAVS.
 3. При записи массива, которая следует за чтением без отключения протяжки ленты, исключаются операции подпунктов 8.4-а – 8.4-б и 8.4-д.
 4. Для увеличения надёжности блоки массива продублировать.

5. Блок обычно начинается идентификатором и заканчивается контрольной суммой.

8.5. Команда **LOAD R_i** загружает с адреса (R_i) 256-байтовый блок, записанный командой **SAVE R_i** . Начальный адрес загрузки – произвольный. Существует принципиальная возможность загрузки в регистровый файл, в том числе установка (**PC**), **ПрП** и других признаков.

Команда загружает любой блок, удовлетворяющий условиям:

– длина блока 256×8 бит;

– наличие синхронизирующих пауз в начале и конце блока, превышающих 1 мс.

При отсутствии записи на МЛ дольше 256 мс команда завершается включением **ОП**.

8.6. Рекомендуются следующая последовательность загрузки массивов, сформированных в соответствии с п. 8.4:

а) в цикле опрашивается завершение перемотки командой **CRFS**;

б) включается лампа командой **LAMP** и электромагнит командой **ELMG** с последующей выдержкой $\sim 100 \cdot 10^{-3}$ с;

в) включаются двигатели протяжки командой **TRTAP**;

г) опрашивается ракорд командой **CLDRS**, при его наличии продолжается в цикле опрос до конца ракорда с последующей выдержкой времени $\sim 300 \cdot 10^{-3}$ с;

д) рекомендуется отключить лампу командой **STTAP**, включив затем электромагнит и двигатели (**ELMG**, **TRTAP**);

е) задаётся пауза на разгон двигателя $\sim 100 \cdot 10^{-3}$ с;

ж) читается по адресу (R_i) командой **LOADR i** очередной блок; проверяется соответствие идентификатора и контрольной суммы;

з) подпункт 8.6-ж повторяется до загрузки завершающего блока массива;

и) после чтения последнего блока НМЛ приводится в исходное состояние операциями **STTAP**, **ELMG**, задержка $\sim 50 \cdot 10^{-3}$ с, **TRTAP**, **STTAP**.

9. УПРАВЛЕНИЕ ПЕРИФЕРИЕЙ И ПИЩУЩЕЙ МАШИНОЙ

9.1. По командам ввода-вывода может быть осуществлён обмен информацией ДЗ-28 с периферийными устройствами (ПУ), совместимыми с ним по интерфейсу ввода-вывода, в том числе и с ПУ 15ВСМ-5.

Адресация ПУ может производиться по одному из 256 состояний регистра **УПР**.

9.2. Выбор ПУ 15ВСМ-5 производится командами **GR1 <адрес>** (0409 **B2A2**) или **GR2 <адрес>** (0410 **B2A2**). Этими командами осуществляется выбор ПУ по состоянию регистра **УПР** (4 или 5 соответственно) и регистра **ВЫВ**, равного **B2A2** команд **GR**. Функции клавиатуры передаются ПУ, **ПрГ** ← 1, **ПрП** ← 0. В этом случае все коды, посланные с ПУ, воспринимаются ДЗ-28 как команды.

Команды GR, за исключением случаев обеспечения ПУ 15ВСМ-5, следует применять только для передачи функций клавиатуры выносному пульту.

Таблица 20

Мнемокод	Код команды B1A1 B2A2	Кол-во байт по каналу	Содержание	Примечание
INPS <i>d</i>	1500 <i>d</i>	(S9)	УПР ← <i>d</i> ; приём в ОЗУ с начального адреса (R10) + (BD) (S9) байтов	Время ожидания ответа ПУ (СИП) не ограничено; по окончаний приёма УПР ← 0; РС ← . + 2
INPR <i>d</i>	1504 <i>d</i>	(R12)	УПР ← <i>d</i> ; приём в ОЗУ с начального адреса (R10) + (BD) (R12) байтов	
INPAS <i>d</i>	1508 <i>d</i>	4 + (S9)	УПР ← 0002; ВЫВ ← 0000; (S2); (S3); 0000; УПР ← <i>d</i> ; приём в ОЗУ с начального адреса (R10) + (BD) (S9) байтов	
INPAR <i>d</i>	1512 <i>d</i>	4 + (R12)	УПР ← 0002; ВЫВ ← 0000; (S2); (S3); 0000; УПР ← <i>d</i> ; приём в ОЗУ с начального адреса (R10) + (BD) (R12) байтов	
INPSV <i>d</i>	1502 <i>d</i>	(S9) + 1	УПР ← <i>d</i> ; приём в ОЗУ с начального адреса (R10) + (BD) (S9) байтов; приём контрольного байта.	Время ожидания ответа ПУ (СИП) не ограничено; по окончании приёма УПР ← 0; РС ← . + 2; если побитная сумма по модулю 2 всех принятых байтов не равна 1515, ОП ← 1; контрольный байт в ОЗУ не записывается
INPRV <i>d</i>	1506 <i>d</i>	(R12) + 1	УПР ← <i>d</i> ; приём в ОЗУ с начального адреса (R10) + (BD) (R12) байтов; приём контрольного байта.	
INPASV <i>d</i>	1510 <i>d</i>	4 + (S9) + 1	УПР ← 0002; ВЫВ ← 0000; (S2); (S3); 0000; УПР ← <i>d</i> ; приём в ОЗУ с начального адреса (R10) + (BD) (S9) байтов; приём контрольного байта	
INPARV <i>d</i>	1514 <i>d</i>	4 + (R12) + 1	УПР ← 0002; ВЫВ ← 0000; (S2); (S3); 0000; УПР ← <i>d</i> ; приём в ОЗУ с начального адреса (R10) + (BD) (R12) байтов; приём контрольного байта	

Таблица 20 (продолжение)

Мнемокод	Код команды B1A1 B2A2	Кол-во байт по каналу	Содержание	Примечание
INPO <i>d</i>	1400 <i>d</i>	1	УПР ← <i>d</i> ; приём в ОЗУ по адресу (R10) + (BD) одного байта	Время ожидания ответа ПУ (СИП) не ограничено; по окончании приёма УПР ← 0; РС ← +2
INPOWC	0412 1404	1 (при поступлении ответа ПУ)	Приём в S3 одного байта при неизменном коде (УПР)	Время ожидания ответа ПУ (СИП) ≤ (R10) · 10 тактов; РС ← +2, если ответ ПУ в указанное время не поступил; РС ← +4, если ответ ПУ в указанное время поступил
INPOWS	0412 1406	1 (при поступлении ответа ПУ)	УПР ← (S2), приём в S3 одного байта; по окончании приёма УПР ← 0	Время ожидания ответа ПУ (СИП) не ограничено; по окончании передачи УПР ← 0; РС ← +2
OUTS <i>d</i>	1501 <i>d</i>	(S9)	УПР ← <i>d</i> ; передача из ОЗУ с начального адреса (R10) + (BD) (S9) байтов	Время ожидания ответа ПУ (СИП) не ограничено; по окончании передачи УПР ← 0; РС ← +2
OUTR <i>d</i>	1505 <i>d</i>	(R12)	УПР ← <i>d</i> ; передача из ОЗУ с начального адреса (R10) + (BD) (R12) байтов	
OUTAS <i>d</i>	1509 <i>d</i>	4 + (S9)	УПР ← 0002; ВЫВ ← 0000; (S2); (S3); 0800; УПР ← <i>d</i> ; передача из ОЗУ с начального адреса (R10) + (BD) (S9) байтов	
OUTAR <i>d</i>	1513 <i>d</i>	4 + (R12)	УПР ← 0002; ВЫВ ← 0000; (S2); (S3); 0800; УПР ← <i>d</i> ; передача из ОЗУ с начального адреса (R10) + (BD) (R12) байтов	

Использование команд GR для выборки ПУ и приёма информации в новых разработках не рекомендуется из-за низкого быстродействия.

9.3. Команда LNCN (0412 1400) используется для установки регистра УПР, по ней УПР ← (S2).

9.4. Команды ввода информации, приведённые в табл. 20, имеют в своём мнемокоде буквы INP (от слова *input* – ввод).

Вывод информации осуществляется командами OUT (*output* – вывод).

9.5. Если после названия команды в мнемокоде присутствует буква A, это означает наличие адресных передач до начала ввода-вывода, аналогично командам ввода-вывода 15BCM-5.

Таблица 20 (продолжение)

Мнемокод	Код команды B1A1 B2A2	Кол-во байт по каналу	Содержание	Примечание
OUTSV <i>d</i>	1503 <i>d</i>	(S9) + 1	УПР ← <i>d</i> ; передача из ОЗУ с начального адреса (R10) + (BD) (S9) байтов; передача контрольного байта	Время ожидания ответа ПУ (СИП) не ограничено; по окончании передачи УПР ← 0; РС ← . + 2; контрольный байт дополняет побитную сумму информационных байтов до кода 1515
OUTRV <i>d</i>	1507 <i>d</i>	(R12) + 1	УПР ← <i>d</i> ; передача из ОЗУ с начального адреса (R10) + (BD) (R12) байтов; передача контрольного байта	
OUTASV <i>d</i>	1511 <i>d</i>	4 + (S9) + 1	УПР ← 0002; ВЫВ ← 0000; (S2); (S3); 0800; УПР ← <i>d</i> ; передача из ОЗУ с начального адреса (R10) + (BD) (S9) байтов; передача контрольного байта	
OUTARV <i>d</i>	1515 <i>d</i>	4 + (R12) + 1	УПР ← 0002; ВЫВ ← 0000; (S2); (S3); 0800; УПР ← <i>d</i> ; передача из ОЗУ с начального адреса (R10) + (BD) (R12) байтов; передача контрольного байта	
OUTO <i>d</i>	1401 <i>d</i>	1	УПР ← <i>d</i> ; передача из ОЗУ одного байта с адреса (R10) + (BD)	Время ожидания ответа ПУ (СИП) не ограничено; по окончании передачи УПР ← 0; РС ← . + 2
OUTOWC	0412 1405	1 (при поступлении ответа ПУ)	Передача (S3) при неизменном коде (УПР)	Время ожидания ответа ПУ (СИП) (R10) · 10 тактов; РС ← . + 2, если ответ ПУ в указанное время не поступил; РС ← . + 4, если ответ ПУ в указанное время поступил
OUTOWS	0412 1407	1 (при поступлении ответа ПУ)	УПР ← (S2); передача (S3); по окончании передачи УПР ← 0	

Примечания. 1. В графе «Кол-во байт по каналу» приведено число синхроимпульсов СИП, необходимых для завершения команды.

2. (РС) при подаче команды с клавиатуры не изменяется.

Для адресных передач начальный адрес зоны ПУ должен быть подготовлен в R9 (в шестнадцатеричной системе).

Адресных передач четыре, идут они при $(УПР) = 2$. По первой передаче передаётся код 0000, по второй и третьей – содержимое регистра $R9$, по четвёртой – 0000 для команд ввода или 0800 для команд вывода.

9.6. Далее в мнемокоде следует одна из букв R , S или O . Наличие буквы R означает, что количество байт, подлежащих вводу-выводу, указано в $R12$ ($R12$ рассматривается как код). Если присутствует буква S , количество байт указано в $S9$. Буква O (от слова *one* – один) соответствует вводу-выводу одного байта.

9.7. После буквы, определяющей задание количества байт, может следовать буква V , означающая наличие контроля при вводе-выводе.

При выводе после заданного числа байт передаётся контрольный байт, дополняющий сумму переданных байт по модулю 2 до кода 1515. При вводе проверяется совпадение байта, принятого после заданного числа байт, с подсчитанным контрольным байтом. При несовпадении включается индикатор $ОП$. Контрольный байт в ОЗУ не записывается.

9.8. Буква W в мнемокоде означает, что время ожидания ответа ПУ ограничено и не превышает $(R10) \cdot 10$ тактов.

Если ответ от ПУ поступает, то после завершения команды пропускается два байта программы при выполнении команды по программе. В этих командах при выводе код выдаётся из $S3$, при вводе – поступает в $S3$.

Если буква W отсутствует, обмен ведётся через ОЗУ, время ожидания ответа ПУ не ограничено, начальный адрес массива ввода-вывода равен $(R10) + (BD)$.

9.9. Завершает мнемокод информация о состоянии регистра управления при вводе-выводе информации.

Если записывается код второго байта команды, значит, ввод-вывод ведётся при $(УПР) = B2A2$. Наличие S в мнемокоде означает, что обмен ведётся при $(УПР) = (S2)$.

При наличии буквы C состояние регистра $УПР$ во время выполнения команды и после её окончания остаётся таким же, какое было до команды.

Если в мнемокоде команды нет буквы C , то после выполнения команды ввода-вывода регистр $УПР$ очищается.

Примеры использования команд ввода-вывода:

а) командой $OUTR 1400$ при

$$BD = 00.01.00.00; (R10) = 00.00.05.00; (R12) = 00.01.00.00$$

на ПК выводится 256 байт, начиная с адреса 00.01.05.00, при $УПР = 1400$ и бесконечно долгим временем ожидания ответа ПУ;

б) по команде $INP 1300$ при

$$(BD) = 02.00.00.00; (R10) = 0.00.00.00; (S9) = 1515$$

с ПМ принимается 255 байт с адреса 02.00.00.00 при $УПР = 1300$ и бесконечно долгим временем ожидания ответа ПУ;

в) по команде $OUTAV 0003$ при

(R9) = 00.02.03.04; (R10) = 01.00.00.00; (BD) = 00.04.00.00; (S9) = 0003
 (01.04.00.00) = 0701; (01.04.00.01) = 0502; (01.04.00.02) = 0008

на ПУ передаётся четыре байта адресных: 0000, 0002, 0304, 0800 при УПР = 0002; три байта информационных: 0701, 0502, 0008; и один байт контрольный: 1304; время ожидания ответа не ограничено;

г) по команде **OUTOWS** при (УПР) = 1308 и при

(R10) = 00.00.00.05; (S3) = 0314; (S2) = 1308

на ПУ поступает код 0314; если ответ от ПУ не поступает в течение 50 тактов, то после завершения команды $PC \leftarrow . + 2$, если ответ поступает, $PC \leftarrow . + 2$, то есть пропускается два байта программы.

9.10. Для адресации ПУ, подключаемых пользователем, не рекомендуется применять значения регистра УПР, используемые для ПУ 15ВСМ-5 и разработанных ПУ ДЗ-28:

0002 – адресные передачи;

0004 и 0005 – команды **GR1** и **GR2**;

0800, 0900, 1000 – графопостроитель;

1200 – ввод с фотосчитывающего устройства;

1300 – ввод с ПМ;

1400 – вывод на ПМ;

1500 – вывод на перфоратор.

9.11. Пишущей машиной можно управлять, используя команды ввода-вывода при соответствующем состоянии регистра УПР и две специальные команды, в которых микропрограммным путём предусмотрено значение УПР, равное 1400:

PRINT #d (1406 **B2A2**), осуществляющая вывод кода **B2A2** на ПМ;

PRINT d (0411 **B2A2**), выводящая на ПМ (**X**) в соответствующем формате:

- При $B2 < 10$, $A2 < 10$, порядке $(X) \leq B2$ вывод (**X**) на ПМ осуществляется с фиксированным положением запятой.
- Если $B2 \geq 10$, то выводится **A2** пробелов. Во всех остальных случаях число печатается с плавающей запятой.

Примеры распечатки чисел в зависимости от кода формата печати и значения числа представлены в **табл. 21**.

Число, печатаемое с фиксированным положением запятой, при $A2 \neq 0$ занимает $B2 + A2 + 2$ знакомест (**B2** знаков до запятой, **A2** знаков после запятой, знак числа и точка «.»), при $A2 = 0$ точка не печатается. Незначащие нули в целой части числа выводятся на печать как пробелы.

Число, печатаемое с плавающим положением запятой, занимает 19 знакомест (знак числа, точка «.»), 12 разрядов мантииссы, пробел, буква «E», знак порядка, 2 разряда порядка).

Таблица 21

Число	Код формата печати	Распечатка чисел																				
132,151419182	03 09		1	3	2	.	1	5	1	4	1	9	1	8	2							
132,151419182	05 09				1	3	2	.	1	5	1	4	1	9	1	8	2					
132,151419182	03 11		.	1	3	2	1	5	1	4	1	9	1	8	2		E	0	3			
132,151419182	02 09		.	1	3	2	1	5	1	4	1	9	1	8	2		E	0	3			
-132,151419182	03 09	-	1	3	2	.	1	5	1	4	1	9	1	8	2							
-132,151419182	09 11	-	.	1	3	2	1	5	1	4	1	9	1	8	2		E	0	3			
0,013215	00 09		.	0	1	3	2	1	5	0	0	0										
0,013215	04 09						.	0	1	3	2	1	5	0	0	0						
0,013215	00 11		.	1	3	2	1	5	0	0	0	0	0	0	0		E	-	0	1		
0,013215	08 11		.	1	3	2	1	5	0	0	0	0	0	0	0		E	-	0	1		
-0,013215	00 09	-	.	0	1	3	2	1	5	0	0	0										
-0,013215	00 11	-	.	1	3	2	1	5	0	0	0	0	0	0	0		E	-	0	1		
1,32	05 00						1															
-13,2151419182	09 09									-	1	3	.	2	1	5	1	4	1	9	1	8

10. ОБСЛУЖИВАНИЕ ПРЕРЫВАНИЙ

10.1. Прерывания могут быть внутренними и внешними.

10.2. Внутреннее прерывание происходит после выполнения команды, вызвавшей установку сигнала ОП в единичное состояние, и разрешённой маске внутреннего прерывания N, если не выполняется обработка внутреннего прерывания. При обработке внутреннего прерывания все виды прерываний запрещены.

Установка N осуществляется командой MOV X, N (0412 0714). При этом N становится равным младшему биту первого разряда регистра X. При N = 1 внутреннее прерывание разрешено.

После выполнения очередной команды программы при ОП = 1 и N = 1 в регистре T0 запоминается текущее состояние PC, управление передаётся команде, адрес которой равен 0.00 (0.00.00.08), то есть старший байт адреса нулевой младший равен содержимому ячейки 0.00.00.08: ОП ← 0.

При N = 0 в T0 запоминается адрес первой (после включения или нажатия клавиши С) команды программы, вызвавшей включение ОП. Выполнение программы продолжается, ОП остаётся включённым.

10.3. Внешнее прерывание вызывается сигналами УП2, УП1, Пр8, Пр4, Пр2, Пр1.

Разрешение внешних прерываний задаётся установкой соответствующих разрядов регистра M.

При включении ДЗ-28 и при нажатии клавиши «С» M ← 0 и N ← 0.

Расположение бит в шестибитном регистре **M** показано на рис. 2. При пересылках по командам **MOV S_i, M** (0413 07 *i*) и **MOV M, S_i** (0413 15 *i*) шести младшим битам регистра **S** соответствуют биты **УП2**, **УП1**, **Пр8**, **Пр4**, **Пр2**, **Пр1** регистра **M** (см. табл. 25).

По команде **MOV S_i, M** в **M** засылается (**S_i**), засылка (**M**) в **S_i** осуществляется командой **MOV M, S_i**.

При нулевом состоянии регистра **M**, точнее, нулевом признаке **ПрМ** (см. п. 2.6) сигналы прерывания не опрашиваются вообще.

Если **ПрМ = 1** и хотя бы один из битов **M** разрешает прерывание соответствующему сигналу, после выполнения в программном режиме каждой команды, за исключением команд переходов типа **BR** и команд формирования числа в регистре **X**, опрашиваются сигналы прерывания, что удлиняет время выполнения команды на 7 тактов.

10.4. Сигналы внешних прерываний разделены на три уровня:

УП2, **УП1** – уровень 4

Пр8 – уровень 2

Пр4, **Пр2**, **Пр1** – уровень 1

Сигналы **УП2** и **УП1** на внешний разъём не выведены.

УП1 сигнализирует о готовности ПМ принять информацию.

При нажатии клавиши ПМ в ДЗ-28 поступает сигнал **УП2**, но символ будет напечатан ПМ только после того, как ДЗ-28 по одной из команд **INP** произведёт ввод символа.

10.5. Если происходит обработка сигнала прерывания, в **УПВ** записывается код (4, 2 или 1) соответствующего уровня. В этом случае прерывание может произойти только по сигналу более высокого уровня или по внутреннему прерыванию.

При наличии прерывающего сигнала, уровень которого больше (**УПВ**), и единичном состоянии соответствующего ему бита маски происходит прерывание программы.

10.6. Начальный адрес прерывающей программы может находиться только в первых 256 байтах ОЗУ. Старший байт адреса нулевой, младший равен содержимому ячейки от 0.00.00.02 до 0.00.00.07 соответственно для сигналов **УП2**, **УП1**, **Пр8**, **Пр4**, **Пр2**, **Пр1**.

Если одновременно приходит больше одного сигнала одинакового уровня, первым опрашивается сигнал, начальный адрес прерывающей программы для которого записан в ячейке с меньшим адресом.

УПВ устанавливается в нуль нажатием клавиши **C** и восстанавливает своё значение при возврате и псевдовозврате из прерывания.

10.7. Ожидание поступления сигналов внешнего прерывания может быть осуществлено командой **WAIT** (1208), устанавливающей **ПрЖ** в единицу. Предварительно должны быть подготовлены соответствующие биты **M**.

С длительностью цикла в 8 тактов по команде **WAIT** опрашиваются сигналы прерывания.

В табл. 22 для команды **WAIT** указано время от начала команды **WAIT** до начала программы прерывания, если сигнал прерывания приходит в первом цикле опроса.

При уходе на внешнее прерывание состояние регистров **R0...R15**, **BP**, **BD** запоминается на соответствующих адресах ОЗУ (см. рис. 1).

10.8. Возврат из прерывания, как внутреннего, так и внешнего, может быть осуществлён командой **RTI** (1211).

Если возврат осуществляется из внутреннего прерывания, $PC \leftarrow (T0)$, разрешается внутреннее прерывание.

При возврате из внешнего прерывания в **R0...R15**, **BP** и **BD** переписывается содержимое соответствующих данному уровню адресов ОЗУ.

Если прерывание произошло с команды **WAIT**, то по команде **RTI** управление передаётся следующей за **WAIT** команде.

10.9. Выход из прерывания может быть осуществлён псевдовозвратом **RTII** (1210), действие которого заключается для внутреннего прерывания в восстановлении разрешения внутреннего прерывания, для внешнего – в восстановлении (**УПВ**) из той ячейки, в которой он запоминался.

10.10. Анализ сигналов **УП1** и **УП2** при работе с ПМ не является обязательным.

Ввод-вывод информации на ПМ может осуществляться по командам **INPUT-OUTPUT**, как и на любое ПУ, без использования системы прерываний. Для уменьшения времени ожидания сигнала от ПМ о готовности к работе используется система прерываний по сигналам **УП1** и **УП2**.

В этом случае ДЗ-28 обращается к ПМ только при готовности ПМ принять или передать информацию.

11. СПРАВОЧНЫЕ ТАБЛИЦЫ

Приведённые в справочнике времена выполнения команд являются оценками верхней границы времён, выраженными в тактах выполнения микрокоманд. Действительное время выполнения команды для конкретных операндов меньше указанного, время выполнения такта микрокоманды для исполнения указано в формуляре М3.857.100 ФО и может быть равно $1 \cdot 10^{-6}$ с или $2 \cdot 10^{-6}$ с.

11.1. В табл. 22, являющейся таблицей кодирования для всех команд, приведены их коды, указаны пункты справочника, в которых команды описаны, и времена выполнения.

Если команда выполняется после команд формирования числа в регистре **X** (см. п. 6.1), то время её выполнения увеличивается на 54 такта, идущие на приведение числа к машинному виду.

Если работа по программе идёт при ненулевой маске внешних прерываний **M** и $PrM = 1$, то время выполнения команды увеличивается на 7 тактов.

11.2. Табл. 23 является таблицей декодирования. Если в табл. 23 границы изменения В (А) не указаны, то В (А) может быть в диапазоне от 00 до 15.

Мнемокод SETPER, не пояснённый в предыдущих разделах, означает включение сигнала программой ошибки ОП.

Если мнемокод не указывается, значит, данный код не должен применяться в программе в качестве кода команды.

11.3. В табл. 24 приведены коды ПМ «Консул 260.1», находящиеся в соответствии с ГОСТ 13052-74.

11.4. В табл. 25 собраны справочные данные для обслуживания внешнего прерывания.

11.5. Коды клавиш пульта приведены в табл. 26.

12. ДОПОЛНЕНИЯ

В настоящем разделе приводятся дополнительные сведения об особенностях команд, описываются новые команды и их использование.

12.1. Особенности выполнения некоторых команд

Микропрограммы реализации команды SAVEX и непрограммируемой операции ЗЛ используют в качестве переключателя сигнал ОП. Поэтому наличие ОП = 1 на входе микропрограммы может повлиять на результат выполнения этих команд, а ОП после выполнения команды сбросится.

12.2. Дополнительные команды

Ниже приведено описание расширения системы команд, реализованное в устройствах, выпускаемых с 1983 года.

Команда подключения сегментов ONSEGM описана в следующем подразделе.

12.2.1. Шестнадцатеричное целое деление

Мнемоника команды: DIVR

Кодирование: 0412 1503

Исходные данные: (R9) – делимое
(R10) – делитель

Результат: (R9) – остаток от деления
(R10) – целое частное

При попытке деления на нуль устанавливается признак программной ошибки ОП.

12.2.2. Пересылка блоков

12.2.2.1. Левая пересылка

Мнемоника команды: MOVBL

Кодирование: 0412 1504

Таблица 22

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
ABGE R_i, R_j	1409 ij		7.2	55	
ABS R_i	0413 08 i		6.6	28	
ABS X	0607		6.4	23	
ACS	0806		6.4	55 500	Функция в радианах
ADD R_i, R_j	1100 ij		6.6	62	
ADD $R_i, @R_j$	0900 ij	(BD)	6.6	73	
ADD X, C	0400 B2A2	(BD)'	6.3	750	$C = 10 \cdot B2 + A2$
ADD X, Y	0600		6.3	600	
ADD X, @Y	0500	(BD)'	6.3	770	
ADD #10, E	0412 0700		6.1	54	
ADD #e, E	0412 07 e		6.1	54	$1 \leq e \leq 9$
ADD #e, R_i	1000 ei		6.6	33	
AHC	0814		6.4	49 000	
AHS	0813		6.4	49 000	
AHT	0815		6.4	28 500	
AND S_i, S_j	1108 ij		6.5	37	
AND $S_i, @R_j$	0908 ij	(BD)	6.5	48	
ANS S_i, S_j	1006 ij		6.7	31	
ASN	0805		6.4	50 300	Функция в радианах
ATN	0807		6.4	15 500	То же
ATOI d	1003 d		6.7	200	
BBIC $i, @R_j$	1415 ij	(BD)	7.2	49	

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
BBIS $i, @R_j$	1415 $i+8j$	(BD)	7.2	49	
BEQ Y, X	0509		7.2	680	
BEQZ X	0412 0611		7.2	32	
BEQZ Y	0412 0411		7.2	32	
BEV $@R_j, +e$	1407 $e-1i$	(BD)	7.2	54	
BGE R_i, R_j	1410 ij		7.2	50	
BGE Y, X	0507		7.2	680	
*BHIS S_i, S_j	1002 ij		7.2	36	
BKEY $.+d$	1411 B2A2		7.2	22	$d = 16 \cdot B2 + A2$
BLT Y, X	0508		7.2	680	
BMI X	0412 0710		7.2	32	
BMI Y	0412 0510		7.2	32	
BNS $\#d, S1$	1008 d		7.2	28	
BNS $\#d, S3$	1009 d		7.2	28	
BNEZ X	0412 0711		7.2	32	
BNEZ Y	0412 0511		7.2	33	
BMER $.+d$	1414 B2A2		7.2	22	$d = 16 \cdot B2 + A2 + 1$
BPER	0510		7.2	24	
BPL X	0412 0610		7.2	32	
BPL Y	0412 0410		7.2	32	
BR $.+d$	1403 B2A2		7.2	20	$d = 16 \cdot B2 + A2 + 1$
BR $.-d$	1402 B2A2		7.2	22	$d = 16 \cdot B2 + A2 - 1$

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
*BSA Y, X	1204		7.2	51	
BSA S_i, S_j	1115 ij		7.2	37	
BSA $S_i, @R_j$	0915 ij	(BD)	7.2	48	
BSAZ $@R_i, +e$	0903 $e-1 i$	(BD)	7.2	58	
BSAZ $R_i, +e$	1103 $e-1 i$		7.2	47	
BSA R_i, R_j	1107		7.2	49	
BSA $R_i, @R_j$	0907 ij	(BD)	7.2	60	
CAP	0808		6.4	64 600	Угол в радианах
*CLDRS	0412 1205		8	31	
CLR R_i	0413 10 i		6.6	33	
CLR X	0715		6.1	66	
CMD R_i	0413 02 i		7.1	$16 + t_{\text{ком}}$	
COM S_i	1110 B2 i		6.5	35	B2 – любое
COM $@R_i$	0910 B2 i	(BD)	6.5	46	То же
COS	0803		6.4	62 500	Аргумент в радианах
*CRFS	0412 1213		8	31	
DEG	0801		6.4	4 700	
DIG e	07 e		6.1	123 для мантиссы 40 для порядка	$0 \leq e \leq 9$
DIV X, C	0403 C	(BD)'	6.3	4 650	$C = 10 \cdot B2 + A2$
DIV X, Y	0603		6.3	4 500	
DIV X, @Y	0503	(BD)'	6.3	4 670	

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
*ELMG	0412 1211		8	28	
END	0512		7.1	22	
E	0710		6.1	36	
EXP	0614		6.4	23 000	
EXT	0613		6.4	18 500	
*FORW	0412 1209		8	23	t до включения перемотки
GO	0514		7.1	31	
GR1 d	0409 d		9	47 (до ВВ ← 0)	УПР = 0004 d = <адрес ПУ>
GR2 d	0410 d		9	47 (до ВВ ← 0)	УПР = 0005 d = <адрес ПУ>
HCS	0811		6.4	32 800	
HSN	0810		6.4	32 700	
HTN	0812		6.4	32 800	
*INFS	0412 1207		8	31	
INPS d	1500 d	(BD)	9	$65 + K(15 + \tau)$	K – количество вводимых в ОЗУ байтов; τ – время ожидания ответа ПУ (не ограничено)
INPAS d	1506 d	(BD)	9	$81 + 4\tau + K(15 + \tau)$	
INPAR d	1512 d	(BD)	9	$92 + 4\tau + K(21 + \tau)$	
INPARV d	1514 d	(BD)	9	$100 + 5\tau + K(27 + \tau)$	
INPASV d	1510 d	(BD)	9	$89 + 5\tau + K(21 + \tau)$	
INPR d	1504 d	(BD)	9	$76 + K(21 + \tau)$	
INPRV d	1506 d	(BD)	9	$84 + \tau + K(27 + \tau)$	
INPSV d	1502 d	(BD)	9	$73 + \tau + K(21 + \tau)$	
INPO d	1400 d	(BD)	9	$54 + \tau$	

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание	
*INPOWC	0412 1404		9	39 + τ	$\tau \leq (R10) \cdot 10$	
*INPOWS	0412 1406		9	42 + τ	$\tau \leq (R10) \cdot 10$	
INT	0608		6.4	65		
INV	0615		6.4	4600		
JMM <i>d</i>	0407 <i>d</i>	(BP)	7.2	26 + 4K	K – количество просмотренных байтов	
JMP @R _i	0413 00 <i>i</i>	(BP)	7.2	47		
JMP @X	1213	(BP)	7.2	166		
JSM B1A1	B1A1	(BP)	7.2	125 + 4K	B1 ≤ 1	K – количество просмотренных байтов
JSM B1A1	B1A1	(BP)	7.2	58 + 4K	2 ≤ B1 ≤ 3	
*JMTT <i>d</i>	1010 <i>d</i>		7.2	39		
*JMTF <i>d</i>	1011 <i>d</i>		7.2	39		
JSR @R _i	0413 01 <i>i</i>	(BP)	7.2	79		
JSTT <i>d</i>	1013 <i>d</i>		7.2	63		
JSTF <i>d</i>	1014 <i>d</i>		7.2	63		
*LAMP	0412 1202		8	29		
LGT	0610		6.4	13 550		
*LNCN	0412 1400		9	35		
LOADP	0513		8	260 000 + 2 250K + $\tau_{\text{рак}}$	K – количество считываемых байтов; $\tau_{\text{рак}}$ – время протяжки до конца ракорда	
*LOADR <i>i</i>	0412 10 <i>i</i>		8	516 000 min		
LOADX	1202	(BP)	8	260 000 + 2 250K + $\tau_{\text{рак}}$		
LOG	0611		6.4	18 000		
MARK <i>d</i>	0408 <i>d</i>		7.1	23		

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
MOV C, X	0405 C	(BD)'	6.2	225	$C = 10 \cdot B2 + A2$
MOV C, Y	0415 C	(BD)'	6.2	225	$C = 10 \cdot B2 + A2$
MOV BD, R _i	0413 13 <i>i</i>		6.6	33	
MOV BP, R _i	0413 14 <i>i</i>		6.6	33	
MOV M, S _i	0413 15 <i>i</i>		10	32	
MOV R _i , BD	0413 05 <i>i</i>		6.6	37	
MOV R _i , BP	0413 06 <i>i</i>		6.6	37	
MOVD R _i , X	0413 04 <i>i</i>		6.6	400	
MOV R _i , X	0413 03 <i>i</i>		6.6	71	
MOV R _i , R _j	1104 <i>ij</i>		6.6	51	
MOV R _i , T _j	1412 <i>ij</i>		6.6	33	
*MOV R _i , -(R _j)	1012 <i>ij</i>		6.6	51	
MOV X, N	0412 0714		10	40	
MOV X, R _i	0413 1100		6.6	60	
MOV X, RR	1206		6.6	115	
MOV X, Y	0604		6.2	91	
MOV X, (R _i)	0412 02 <i>i</i>		6.6	134	
*MOV Y, (R _i)	0412 00 <i>i</i>		6.6	134	
MOV X, @Y	0504	(BD)'	6.2	243	
MOV Y, C	0414 C	(BD)'	6.2	225	$C = 10 \cdot B2 + A2$
MOV Y, X	0605		6.2	91	
MOV Y, Z	1214		6.2	68	

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
MOV @Y,X	0505	(BD)'	6.2	243	
MOV #d,S _i	13 i d		6.5	26	
MOV R _i ,@R _j	0904 ij	(BD)	6.6	62	
MOV RR,X	1207		6.6	89	
MOV @R _i ,R _j	0905 ji	(BD)	6.6	59	
*MOV (R _i)+,R _j	1015 ji		6.6	51	
MOV @R _i ,S _j	0913 ji	(BD)	6.5	47	
MOV (R _i),X	0412 03 i		6.6	111	
*MOV (R _i),Y	0412 01 i		6.6	111	
MOV S _i ,M	0413 07 i		10	32	
MOV S _i ,S _j	1112 ij		6.5	36	
MOV S _i ,@R _j	0912 ij	(BD)	6.5	47	
MOV T _i ,R _j	1413 ij		6.6	33	
MOVH X,R _i	0413 12 i		6.6	170	
MOV X,C	0404 C	(BD)'	6.2	225	$C = 10 \cdot B2 + A2$
MUL R _i ,R _j	1102 ij		6.6	420	
MUL R _i ,@R _j	0902 ij	(BD)	6.6	431	
MUL X,C	0402 C	(BD)'	6.3	4 650	$C = 10 \cdot B2 + A2$
MUL X,Y	0602		6.3	4 500	
MUL X,@Y	0502	(BD)'	6.3	4 670	
NEG R _i	0413 09 i		6.6	28	
NEG X	0711		6.1	30	

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание	
NORM	1209		6.4	220		
NSS <i>d</i>	1005 <i>d</i>		6.7	44 + 15K	K – количество просмотренных байтов	
NSN <i>d</i>	1004 <i>d</i>		6.7	44 + 15K		
OR <i>S_i, S_j</i>	1111 <i>ij</i>		6.5	38		
OR <i>S_i, @R_j</i>	0911 <i>ij</i>	(BD)	6.5	49		
OUTS <i>d</i>	1501 <i>d</i>	(BD)	9	65 + K(14 + τ)	K – количество выводимых из ОЗУ байтов; τ – время ожидания ответа ПУ (СИП)	
OUTAS <i>d</i>	1509 <i>d</i>	(BD)	9	81 + 4τ + K(14 + τ)		
OUTAR <i>d</i>	1513 <i>d</i>	(BD)	9	92 + 4τ + K(20 + τ)		
OUTARV <i>d</i>	1515 <i>d</i>	(BD)	9	100 + 5τ + K(26 + τ)		
OUTASV <i>d</i>	1511 <i>d</i>	(BD)	9	89 + 5τ + K(20 + τ)		
OUTR <i>d</i>	1505 <i>d</i>	(BD)	9	76 + K(20 + τ)		
OUTRV <i>d</i>	1507 <i>d</i>	(BD)	9	84 + τ + K(26 + τ)		
OUTSV <i>d</i>	1503 <i>d</i>	(BD)	9	73 + τ + K(20 + τ)		
OUTO <i>d</i>	1401 <i>d</i>	(BD)	9	56 + τ		
*OUTOWC	0412 1405		9	39 + τ		τ ≤ (R10) · 10
*OUTOWS	0412 1407		9	42 + τ		τ ≤ (R10) · 10
PAUSE	0412 0615		7.1	479 292		
*PAUSER	0412 1402		7.1	35 + R10 · 10		
PI	0609		6.4	53		
POC	0809		6.4	72 000	(X) в радианах	
POINT	0712		6.1	77		
PRINT <i>d</i>	0411 <i>d</i>		9	200 + 19τ	τ – время ожидания ответа ПМ	

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
*PRINT #d	1406 d		9	37 + τ	τ – время ожидания ответа ПМ
PRIOR S_i, R_j	1007 ij		6.7	38	
QRT	0713		6.4	4 660	
RAD	0800		6.4	9 200	
RES	0714		6.3	95	
REW	1200		8	20	t до включения перемотки
RTI	1211		10	170	
RTII	1210		10	30	
RTS	0511		7.2	50	
RTSGO	0412 0712		7.2	65	
RTSI	1212		7.2	28	
*SAVER i	0412 11 i		8	516 000 min	
SAVEX	1203	(BP)	8	320 000 + 2 250K + $\tau_{\text{рак}}$	K – количество записываемых байтов; $\tau_{\text{рак}}$ – время протяжки до конца ракорда
*SAVC	0412 1204		8	28	
*SAVS	0412 1212		8	28	
*SMER	0412 1210		8	29	
SQR	0612		6.4	25 000	
SIN	0802		6.4	67 000	Аргумент в радианах
*SNCS	0412 1206		8	31	
SOBZ $R_i, +e$	1408 e-1 i		7.2	39	
SPCMD 0512	0412 0512		6.9	36	

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
SPCMD 0513	0412 0513		6.9	36	
SPCMD 0514	0412 0514		6.9	65	
SPCMD 0515	0412 0515		6.9	4 500	
SPCMD 0612	0412 0612		6.9	439	
SPCMD 0613	0412 0613		6.9	453	
SPCMD 0614	0412 0614		6.9	36	
SPCMD 0713	0412 0713		6.9	34	
SPCMD 0715	0412 0715		6.9	4 548	
STOP	0515		7.1		
*STTAP	0412 1200		8	28	
SUB R_i, R_j	1101 ij		6.6	65	
SUB $R_i, @R_j$	0901 ij	(BD)	6.6	76	
SUB X, Y	06 01		6.3	600	
SUB X, C	0401 C	(BD)'	6.3	750	$C = 10 \cdot B2 + A2$
SUB X, @Y	0501	(BD)'	6.3	770	
SUB #10, E	0412 0400		6.1	54	
SUB #e, E	0412 04 e		6.1	54	$1 \leq e \leq 9$
SUB #e, R_i	1001 ei		6.6	33	
*SWA R_i	0412 08 i		6.6	39	
SWA R_i, R_j	1106 ij		6.6	52	
SWA $R_i, @R_j$	0906 ij	(BD)	6.6	63	
*SWA S_i	0412 09 i		6.5	33	

Таблица 22 (продолжение)

Мнемокод	Код B1A1 B2A2	Индек- сация	Пункт	Время выполнения	Примечание
SWA S_i, S_j	1114 ij		6.5	39	
SWA $S_i, @R_j$	0914 ij	(BD)	6.5	50	
SWA X, C	0406 C	(BD)'	6.2	274	$C = 10 \cdot B2 + A2$
SWA X, Y	0606		6.2	140	
SWA X, @Y	0506	(BD)	6.2	292	
TAN	0804		6.4	32 300	Аргумент в радианах
*TRAP	1205		7.2	38	
*TRTAP	0412 1203		8	28	
VERR B2A2	1404 B2A2	(BD)	6.8	99 + 19K	K – количество контролируемых байтов
VERX	1201	(BP)	6.8	157 + 24K	
VEX B2A2	1405 B2A2	(BD)	6.8	93 + 10K	
WAIT	1208		10	230	
*WTRT	0412 1403		8	$\leq 35 + K(R10) \cdot 10$	K – количество считываемых бит
XOR S_i, S_j	1109 ij		6.5	37	
XOR $S_i, @R_j$	0909 ij	(BD)	6.5	48	

Примечание. Дополнительные команды см. в разделе 12.

Таблица 23

Код		Мнемокод		Код		Мнемокод	
B1A1	B2A2			B1A1	B2A2		
0000		JSM	0000	:		:	
:		:		0412	0609		---
0115		JSM	0115	0412	0610	BPL	X
0200		JSM	0200	0412	0611	BEQZ	X
:		:		0412	0612	SPCMD	0612
0315		JSM	0315	0412	0613	SPCMD	0613
0400	B2A2	ADD	X, C	0412	0614	SPCMD	0614
0401	B2A2	SUB	X, C	0412	0615	PAUSE	
0402	B2A2	MUL	X, C	0412	0700	ADD	#10, E
0403	B2A2	DIV	X, C	0412	0701	ADD	#01, E
0404	B2A2	MOV	X, C	:		:	
0405	B2A2	MOV	C, X	0412	0709	ADD	#09, E
0406	B2A2	SWA	X, C	0412	0710	BMI	X
0407	B2A2	JMM	B2A2	0412	0711	BNEZ	X
0408	B2A2	MARK	B2A2	0412	0712	RTSGO	
0409	B2A2	GR	B2A2	0412	0713	SPCMD	0713
0410	B2A2	GR	B2A2	0412	0714	MOV	X, N
0411	B2A2	PRINT	B2A2	0412	0715	SPCMD	0715
*0412	00A2	MOV	Y, (RA2)	*0412	08A2	SWA	RA2
*0412	01A2	MOV	(RA2), Y	*0412	09A2	SWA	SA2
0412	02A2	MOV	X, (RA2)	*0412	10A2	LOADR	A2
0412	03A2	MOV	(RA2), X	*0412	11A2	SAVER	A2
0412	0400	SUB	#10, E	*0412	1200	STTAP	
0412	0401	SUB	#01, E	0412	1201		---
:		:		*0412	1202	LAMP	
0412	0409	SUB	#09, E	*0412	1203	TRTAP	
0412	0410	BPL	Y	*0412	1204	SAVC	
0412	0411	BEQZ	Y	*0412	1205	CLDRS	
0412	0412		---	*0412	1206	SNCS	
:		:		*0412	1207	INFS	
0412	0415		---	0412	1208		---
0412	0500		---	*0412	1209	FORW	
:		:		*0412	1210	SMER	
0412	0509		---	*0412	1211	ELMG	
0412	0510	BMI	Y	*0412	1212	SAVS	
0412	0511	BNEZ	Y	*0412	1213	CRFS	
0412	0512	SPCMD	0512	0412	1214		---
0412	0513	SPCMD	0513	0412	1215		---
0412	0514	SPCMD	0514	0412	1300		---
0412	0515	SPCMD	0515	:		:	
0412	0600		---	0412	1315		---

Таблица 23 (продолжение)

Код		Мнемокод	Код		Мнемокод
B1A1	B2A2		B1A1	B2A2	
*0412	1400	LNCN	0509	BEQ	Y, X
0412	1401	---	0510	BPER	
*0412	1402	PAUSER	0511	RTS	
*0412	1403	WTRT	0512	END	
*0412	1404	INPOWC	0513	LOADP	
*0412	1405	OUTOWC	0514	GO	
*0412	1406	INPOWS	0515	STOP	
*0412	1407	OUTOWS	0600	ADD	X, Y
*0412	1408	SETPER	0601	SUB	X, Y
:	:		0602	MUL	X, Y
*0412	1415	SETPER	0603	DIV	X, Y
0412	1500	} См. раздел 12	0604	MOV	X, Y
:	:		0605	MOV	Y, X
0412	1515		0606	SWA	X, Y
0413	00A2	JMP @RA2	0607	ABS	X
0413	01A2	JSR @RA2	0608	INT	
0413	02A2	CMD RA2	0609	PI	
0413	03A2	MOV RA2, X	0610	LGT	
0413	04A2	MOVD RA2, X	0611	LOG	
0413	05A2	MOV RA2, BD	0612	SQR	
0413	06A2	MOV RA2, BP	0613	EXT	
0413	07A2	MOV SA2, M	0614	EXP	
0413	08A2	ABS RA2	0615	INV	
0413	09A2	NEG RA2	0700	DIG	0
0413	10A2	CLR RA2	:	:	
0413	11A2	MOV X, RA2	0709	DIG	9
0413	12A2	MOVH X, RA2	0710	E	
0413	13A2	MOV BD, RA2	0711	NEG	X
0413	14A2	MOV BP, RA2	0712	POINT	
0413	15A2	MOV M, SA2	0713	QRT	
0414	B2A2	MOV Y, C	0714	RES	
0415	B2A2	MOV C, Y	0715	CLR	X
0500		ADD X, @Y	0800	RAD	
0501		SUB X, @Y	0801	DEG	
0502		MUL X, @Y	0802	SIN	
0503		DIV X, @Y	0803	COS	
0504		MOV X, @Y	0804	TAN	
0505		MOV @Y, X	0805	ASN	
0506		SWA X, @Y	0806	ACS	
0507		BGE Y, X	0807	ATN	
0508		BLT Y, X	0808	CAP	

Таблица 23 (продолжение)

Код		Мнемокод	Код		Мнемокод
B1A1	B2A2		B1A1	B2A2	
0809		POC			
0810		HSN			
0811		HCS			
0812		HTN			
0813		AHS			
0814		AHC			
0815		AHT			
0900	B2A2	ADD	RB2, @RA2		
0901	B2A2	SUB	RB2, @RA2		
0902	B2A2	MUL	RB2, @RA2		
0903	B2A2	BSAZ	@RA2, +B2 + 1		
0904	B2A2	MOV	RB2, @RA2		
0905	B2A2	MOV	@RA2, RB2		
0906	B2A2	SWA	RB2, @RA2		
0907	B2A2	BSA	RB2, @RA2		
0908	B2A2	AND	SB2, @RA2		
0909	B2A2	XOR	SB2, @RA2		
0910	B2A2	COM	@RA2		
0911	B2A2	OR	SB2, @RA2		
0912	B2A2	MOV	SB2, @RA2		
0913	B2A2	MOV	@RA2, SB2		
0914	B2A2	SWA	SB2, @RA2		
0915	B2A2	BSA	SB2, @RA2		
1000	B2A2	ADD	#B2, RA2		
1001	B2A2	SUB	#B2, RA2		
*1002	B2A2	BHIS	SB2, SA2		
1003	B2A2	ATOI	B2A2		
1004	B2A2	NSN	B2A2		
1005	B2A2	NSS	B2A2		
1006	B2A2	ANS	SB2, SA2		
1007	B2A2	PRIOR	SB2, RA2		
1008	B2A2	BNS	#B2A2, S1		
1009	B2A2	BNS	#B2A2, S3		
*1010	B2A2	JMTT	B2A2		
*1011	B2A2	JMTF	B2A2		
*1012	B2A2	MOV	RB2, -(RA2)		
1013	B2A2	JSTT	B2A2		
1014	B2A2	JSTF	B2A2		
*1015	B2A2	MOV	(RA2)+, RB2		
1100	B2A2	ADD	RB2, RA2		
1101	B2A2	SUB	RB2, RA2		
1102	B2A2	MUL	RB2, RA2		
1103	B2A2	BSAZ	RA2, +B2 + 1		
1104	B2A2	MOV	RB2, RA2		
1105	B2A2	MOV	RA2, RB2		
1106	B2A2	SWA	RB2, RA2		
1107	B2A2	BSA	RB2, RA2		
1108	B2A2	AND	SB2, SA2		
1109	B2A2	XOR	SB2, SA2		
1110	B2A2	COM	SA2		
1111	B2A2	OR	SB2, SA2		
1112	B2A2	MOV	SB2, SA2		
1113	B2A2	MOV	SA2, SB2		
1114	B2A2	SWA	SB2, SA2		
1115	B2A2	BSA	SB2, SA2		
1200		FEW			
1201		VERX			
1202		LOADX			
1203		SAVEX			
*1204		BSA	Y, X		
*1205		TRAP			
1206		MOV	X, RR		
1207		MOV	RR, X		
1208		WAIT			
1209		NORM			
1210		RTII			
1211		RTI			
1212		RTSI			
1213		JMP	@X		
1214		MOV	Y, Z		
1215		MOV	Z, Y		
1300	B2A2	MOV	#B2A2, S0		
:		:			
1315	B2A2	MOV	#B2A2, S15		
1400	B2A2	INPO	B2A2		
1401	B2A2	OUTO	B2A2		
1402	B2A2	BR	.-16B2 - A2 + 1		
1403	B2A2	BR	.-16B2 + A2 + 1		
1404	B2A2	VERR	B2A2		
1405	B2A2	VEX	B2A2		
*1406	B2A2	PRINT	#B2A2		
1407	B2A2	BEV	@RA2, +B2 + 1		

Таблица 23 (продолжение)

Код		Мнемокод		Код		Мнемокод	
B1A1	B2A2			B1A1	B2A2		
1408	B2A2	SOBZ	RA2, +B2 + 1	1504	B2A2	INPR	B2A2
1409	B2A2	ABGE	RB2, RA2	1505	B2A2	OUTR	B2A2
1410	B2A2	BGE	RB2, RA2	1506	B2A2	INPRV	B2A2
1411	B2A2	BKEY	. +16B2 + A2 + 1	1507	B2A2	OUTRV	B2A2
1412	B2A2	MOV	RB2, TA2	1508	B2A2	INPAS	B2A2
1413	B2A2	MOV	TB2, RA2	1509	B2A2	OUTAS	B2A2
1414	B2A2	BMER	. +16B2 + A2 + 1	1510	B2A2	INPASV	B2A2
1415	B2A2	BBIC	B2, @RA2	1511	B2A2	OUTASV	B2A2
1415	B2A2	BBIS	B2 – 8, @RA2	1512	B2A2	INPAR	B2A2
1500	B2A2	INPS	B2A2	1513	B2A2	OUTAR	B2A2
1501	B2A2	OUTS	B2A2	1514	B2A2	INPARV	B2A2
1502	B2A2	INPSV	B2A2	1515	B2A2	OUTARV	B2A2
1503	B2A2	OUTSV	B2A2				

Исходные данные:

(R12) – длина блока в байтах;

(R10) – абсолютный адрес последнего байта пересылаемого блока;

(R9) – абсолютный адрес последнего байта блока-приёмника.

Описание:

Блоки байтов задаются длиной и старшими адресами. Блок-источник последовательно переписывается в блок-приёмник, начиная от старших адресов.

При адресации наложенных блоков команда реализует операцию раздвижки.

При (R12) = 0 пересылка не выполняется.

По завершению команды (R12) = 15.15.15.15.

12.2.2.2. Правая пересылка

Мнемоника команды: **MOVBR**

Кодирование: 0412 1512

Исходные данные:

(R12) – длина блока в байтах;

(R10) – абсолютный адрес первого байта пересылаемого блока;

(R9) – абсолютный адрес первого байта блока-приёмника.

Описание:

Блоки байтов задаются длиной и младшими адресами. Блок-источник последовательно переписывается в блок-приёмник, начиная от младших адресов.

При адресации наложенных блоков команда реализует операцию сдвижки.

При (R12) = 0 пересылка не выполняется.

По завершению команды (R12) = 15.15.15.15.

Таблица 24

Символ	Код	Символ	Код	Символ	Код
А	0601	К	0411	&	0206
Б	0602	Л	0412	@	0400
В	0707	М	0413	Ы	0709
Г	0607	Н	0414	Ь	0708
Д	0604	О	0415	Э	0712
Е	0605	Р	0500	Ю	0600
Ж	0706	Q	0501	Я	0701
З	0710	R	0502	[0511
И	0609	S	0503]	0513
Й	0610	Т	0504	(0208
К	0611	U	0505)	0209
Л	0612	V	0506	0	0300
М	0613	W	0507	1	0301
Н	0614	X	0508	2	0302
О	0615	Y	0509	3	0303
П	0700	Z	0510	4	0304
Р	0702	:	0310	5	0305
С	0703	;	0311	6	0306
Т	0704	,	0212	7	0307
У	0705	.	0214	8	0308
Ф	0606	?	0315	9	0309
Х	0608	!	0201	Пробел	0200
Ц	0603	'	0207	КЛ	0101
Ч	0714	+	0211	ЧЛ	0102
Ш	0711	—	0213		
Щ	0713	*	0210	ВШ	0008
А	0401	/	0215		
В	0402	└	0514	ГТ	0009
С	0403	<	0312		
Д	0404	=	0313	ПС	0010
Е	0405	>	0314	ВК	0013
Ф	0406	\	0512	ВР	0014
Г	0407	—	0515	НР	0015
Н	0406	#	0203		
І	0409	¤	0204	Забой	0715
Ј	0410	%	0205		

Таблица 25

Причина прерывания	Бит регистра S, соответствующий биту M	Начальный адрес прерывающей программы	Уровень прерывания	Адреса запоминания (R0)...(R15)	Адреса запоминания (BP)	Адреса запоминания (BD)
УП2	b2	0.00. (0.00.00.02)	4	7.15.00.00...7.15.09.15	и 7.15.10.14	и 7.15.10.12
УП1	b1	0.00. (0.00.00.03)			и 7.15.10.15	и 7.15.10.13
Пр8	a8	0.00. (0.00.00.04)	2	7.15.06.00...7.15.07.15	и 7.15.10.10	и 7.15.10.08
Пр4	a4	0.00. (0.00.00.05)	1	7.15.04.00...7.15.05.15	и 7.15.10.11	и 7.15.10.09
Пр2	a2	0.00. (0.00.00.06)			и 7.15.10.06	и 7.15.10.04
Пр1	a1	0.00. (0.00.00.07)			и 7.15.10.07	и 7.15.10.05

Примечание. Время реакции на внешнее прерывание равно 230 тактов плюс время выполнения последовательности команд, между которыми анализ прерываний не производится (см. п. 10.3).

Таблица 26

Клавиша	Код	Клавиша	Код
M	0408	4	0704
СЛ	0513	5	0705
▷	0407	6	0706
S	0514	7	0707
ОП	0510	8	0708
\sqrt{x}	0612	9	0709
10^x	0613	◇	0411
e^x	0614	↕	0606
$\frac{1}{x}$	0615	↓	0605
x^2	0713	↑	0604
lg x	0610	÷	0603
ln x	0611	×	0602
СК	0715	–	0601
ЗН	0711	+	0600
E	0710	÷П	0403
π	0609	×П	0402
,	0712	П–	0401
0	0700	П+	0400
1	0701	↕П	0406
2	0702	ВП	0405
3	0703	ЗП	0404

12.2.3. Сложение блоков

По аналогии с командами пересылок реализованы команды сложения блоков. В каждый байт блока-приёмника записывается шестнадцатеричная сумма соответствующих байтов приёмника и источника с учётом предшествующего межбайтового переноса. При входе в команду ненулевое значение (S1) является признаком прибавления единицы к сумме младших байтов. Перенос из старшего байта суммы заносится в регистр S1.

При задании нулевой длины блока в R12 операнды не изменяются.

По завершению операции регистр R12 содержит 15.15.15.15.

12.2.3.1. Левое сложение блоков

Мнемоника команды: **ADDBL**

Кодирование: 0412 1500

Исходные данные:

(R12) – длина блока в байтах;

(R10) – абсолютный адрес последнего (младшего) байта первого слагаемого;

(R9) – абсолютный адрес последнего (младшего) байта второго слагаемого и суммы;

(S1) – признак добавления единицы в младший байт.

12.2.3.2. Правое сложение блоков

Мнемоника команды: **ADDBR**

Кодирование: 0412 1508

Исходные данные:

(R12) – длина блока в байтах;

(R10) – абсолютный адрес первого (младшего) байта первого слагаемого;

(R9) – абсолютный адрес первого (младшего) байта второго слагаемого и суммы;

(S1) – признак добавления единицы в младший байт.

12.3. Управление расширенной памятью

12.3.1. Настоящий подраздел относится к модификациям устройства с объёмом ОЗУ 128 Кб.

С момента включения питания до применения команды управления расширенной памятью эти модификации с точки зрения пользователя эквивалентны устройствам с объёмом ОЗУ 32 Кб и могут использоваться без каких-либо изменений их программное обеспечение.

12.3.2. Расширенная память представляет собой набор из шестнадцати 8-килобайтовых сегментов, пронумерованных от 00 до 15.

Адресное пространство ДЗ-28 разделено на четыре страницы объёмом 8 Кб;

00.00.00.00 – 01.15.15.15 – страница 0;

02.00.00.00 – 03.15.15.15 – страница 1;

04.00.00.00 – 05.15.15.15 – страница 2;

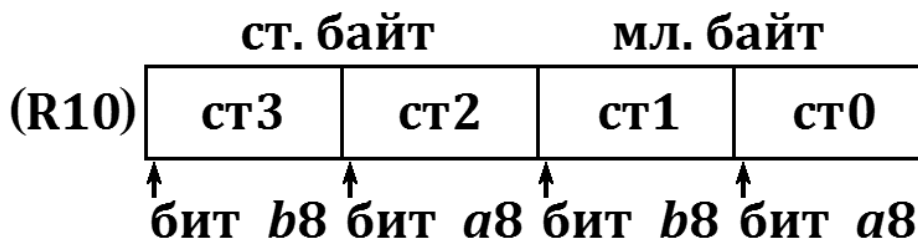
06.00.00.00 – 07.15.15.15 – страница 3.

12.3.3. В любой момент времени в качестве каждой страницы используется один из сегментов памяти. Подключением сегментов управляет аппаратный 16-битовый регистр, каждая тетрада которого хранит номер сегмента, используемого для соответствующей страницы.

По включению питания в этом регистре устанавливается код 15.14.13.12, соответствующий подключению сегментов 12, 13, 14, 15 соответственно к 0, 1, 2, 3 страницам.

12.3.4. Содержимое регистра подключения сегментов и, следовательно, набор сегментов в адресном пространстве могут быть изменены командой **ONSEGM** (0412 1506). По этой команде устанавливается набор сегментов, заданный содержимым регистра **R10**. **R10** обнуляется.

Каждая тетрада регистра **R10** интерпретируется как номер сегмента для соответствующей страницы:



Контроль корректности задания набора сегментов в команде отсутствует. В связи с этим обстоятельством использование команды требует предельного внимания программиста, особенно при переключении сегментов третьей и нулевой страницы.

Напомним, что третья страница содержит все регистры, в том числе счётчик команд, а нулевая – векторá прерываний.

Поэтому в программах пользователя рекомендуется, по возможности, использовать для подключения дополнительных сегментов памяти только страницы 1 и 2, размещая в них данные программы.

12.3.5. Приведём простейший пример использования расширенной памяти.

Напишем подпрограмму вызова в регистр *X* содержимого восьмибайтовой ячейки, заданной номером сегмента в регистре *S1* и относительным адресом внутри сегмента в регистре *R9*.

Вызов сегментов будем производить в страницу 2, подключение сегментов к остальным страницам оставим в исходном состоянии. Текст подпрограммы:




```

MOV    #1500,S4    ; подготовка для
MOV    #1312,S5    ; страниц 0, 1, 3
XOR    S1,S4       ; запись заданного номера сегмента
ONSEGM                ; вызов сегментов в стр. 2
SWA    S2          ; вычисление абсолютного
ADD    #4,S2       ; адреса
SWA    S2
MOV    (R9),X      ; вызов значения
RTS

```

Аналогично строится подпрограмма чтения.

13. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводительного документа и дата	Подпись	Дата
	изменённых	заменённых	новых	аннулированных					
1	3,6,7,13,14, 15,16,17,18,19, 20,23,31,33, 36,37,38,42, 43,44,46,48,49, 51,52,62,67, 70,73,75,76, 79,80,82,83,84, 86,87,88,89, 90,91,93,94,95, 98,100,101,103, 104,109,110,54, 81,97,111,89					И52270-79			9.10.79
2		112				И52239-79			9.10.79
3	37,40,47,65,77, 78,81,93,100,101, 102,107					И52412-79			1.12.79

