

МИКРОСХЕМЫ ИНТЕГРАЛЬНЫЕ Л1839ВМ1

Техническое описание

ШИЗ.480.319 ТО

Инв.№ подп.	Подп. и дата	Взам. инв.№	Инв. № дубл.	Подп. и дата
93353	Лисо/1904.91			

Получен
иск. 493/8-48
от 28.02.99
г. Москва
ОАО "Ангелес"

1991

СОДЕРЖАНИЕ

Лист

I.	СБИС ЦЕНТРАЛЬНОГО ПРОЦЕССОРА (ЦПР) Л1839ВМ1	3
I.I.	Общие характеристики	3
I.2.	Структурная схема и общие принципы функционирования	19
I.3.	Дешифрация команд	23
I.4.	Операционный блок	24
I.5.	Прерывания и исключительные состояния	29
I.6.	Состояние процессора	31
I.7.	Различия между прерываниями и исключительными состояниями	36
I.8.	Основные параметры прерываний и исключительных состояний	37
I.9.	Блок прерываний БПР	43
I.10.	Временные диаграммы процессора, идентификаторы обменов	48
I.II.	Функциональные и электрические схемы входных и выходных каскадов	76
I.I2.	Система микрокоманд	84
I.I3.	Описание ресурсов микропрограммного уровня процессора	103
I.I4.	Система команд	III

ШИЗ. 480.319 ТО

Микросхемы интегральные ИЛ839ВМТ

Техническое описание

Лит.	Лист	Листов8
21	2	152

I. СБИС ЦЕНТРАЛЬНОГО ПРОЦЕССОРА (ЦПР) Л1839ВМ1

I.I. Общие характеристики

I.I.I. ЦПР предназначен для выполнения в составе ЭВМ операций по чтению и обработке команд, вычислению и преобразованию адреса операндов, чтению из памяти, временному хранению на внутренних регистрах и записи операндов, а также для выполнения арифметических и логических операций над этими операндами.

ЦПР изготавливается по КМОП технологии с двойной металлизацией. Размер кристалла - 9,5x9,95 мм. Напряжение питания микросхем +5В. Частота тактового сигнала синхронизации до 10 МГц.

В качестве накопителя микрокоманд используется внешнее ПЗУ. Для обеспечения максимального быстродействия это ПЗУ должно иметь время чтения не более 60 нс при цикле обращения 200 нс.

Основные характеристики ЦПР следующие:

Разрядность операнда, бит	- 8,16,32,64;
Разрядность виртуального адреса, бит	- 32;
Разрядность физического адреса, бит	- 24;
Разрядность адреса микрокоманды, бит	- 14;
Разрядность слова микрокоманды, бит	- 32;
Объем накопителя микрокоманд, К двойных слов	- 16;
Количество регистров общего назначения	- 16;
Время цикла выполнения микрокоманды (микроцикла), нс	- 200;
Минимальное количество микропиков, необходимое для выполнения команды	- 1;
Быстродействие (операции типа "сложение" в формате "регистр-регистр"), млн. оп/с	- до 5.

Наб. № подп	Подп. и дата
93353	Чкалов/19.04.91

Условное графическое обозначение ШР представлено на рис. I.

Назначение выводов представлено в табл. I.

I.I.2. Электрические параметры, условия эксплуатации

I.I.2.1. Напряжение питания микросхем М1839ВМ1 $+5V \pm 10\%$.

I.I.2.2. Электрические параметры приведены в табл. 2.

I.I.2.3. Предельно допустимые режимы эксплуатации приведены в табл. 3.

I.I.2.4. Внешние воздействующие факторы:

Синусоидальная вибрация:

диапазон частот, Гц от 1 до 5000
амплитуда ускорения, $m \cdot c^{-2}$ (g) 400(40)

Механический удар:

одиночного действия:

пиковое ударное ускорение, $m \cdot c^{-2}$ (g) 1500(150)

длительность действия ударного ускорения, мс от 0,1 до 2,0
многократного действия:

пиковое ударное ускорение, $m \cdot c^{-2}$ (g) 1500(150)

длительность действия ударного ускорения, мс от 1 до 5

Линейное ускорение, $m \cdot c^{-2}$ (g) 5000(500)

Акустический шум:

диапазон частот, Гц 50-10000

уровень звукового давления (относительно
 $2 \cdot 10^{-5}$ Па), дБ 170

Атмосферное пониженное давление:

рабочее, Па (мм рт.ст.) $1,3 \cdot 10^{-4}$ (10^{-6})

пределальное, Па (мм рт.ст.) $1,3 \cdot 10^{-4}$ (10^{-6})

Атмосферное повышенное рабочее давление, ата 3

Повышенная температура среды:

рабочая, $^{\circ}\text{C}$ 85

пределенная, $^{\circ}\text{C}$ 125

№ подл.	Подл. и дата	Взам. инв. №	Инв. №	Подл. и дата
1	25.3.71	25.12.92	Одн.	23.3.53



Пониженная температура среды:

рабочая, $^{\circ}\text{C}$ минус 60
предельная, $^{\circ}\text{C}$ минус 60

Смена температур:

от пониженной предельной температуры среды, $^{\circ}\text{C}$ минус 60
до повышенной предельной температуры среды, $^{\circ}\text{C}$ 125

Повышенная относительная влажность при 35°C , % 98

Атмосферные конденсированные осадки (роса, иней)

Соляной (морской) туман

Плесневые грибы

Контрольные среды (среды заполнения):

объемная доля компонентов контрольной среды, %:

гелиево-воздушная 90
аргоно-воздушная 90
аргоно-азотная 90

I.I.2.5. Минимальная наработка микросхем 100000 часов, а в облегченном режиме при $U_{cc} = 5\text{V} \pm 3\%$ - 120000 часов.

I.I.2.6. Интенсивность отказов в течение наработки не более $1 \cdot 10^{-6}$ 1/ч.

I.I.2.7. Гамма-процентный срок сохраняемости - 25 лет.

I.I.3. Архитектура ЦПР позволяет работать с широким диапазоном типов данных, с эффективным набором способов адресации и с большим пространством виртуальных адресов (4 Гбайт).

Система команд микропроцессора совместима с системой команд ЭВМ семейства VAX-II.

Инструкции, обрабатываемые ЦПР, могут быть сгруппированы на основании их функций по следующим классам:

1) арифметические и логические, для обработки целых чисел, чисел с плавающей запятой, упакованных десятичных строк символов и инструкций для работы с переменными битовыми полями;

2) управление путями выполнения программы;

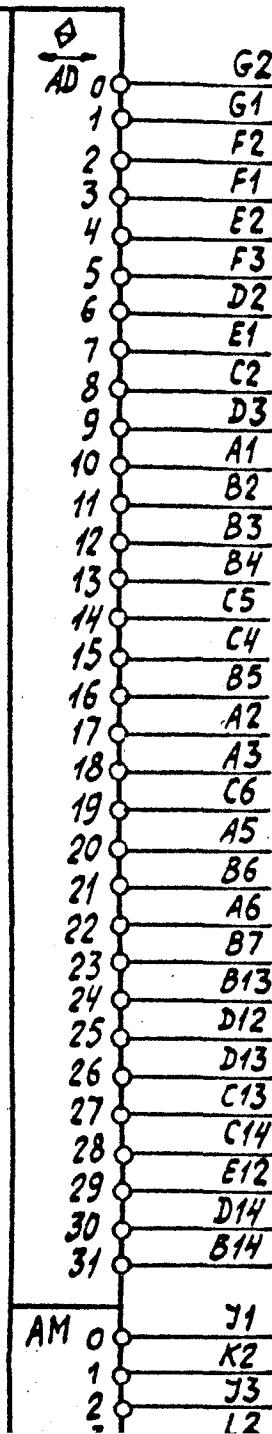
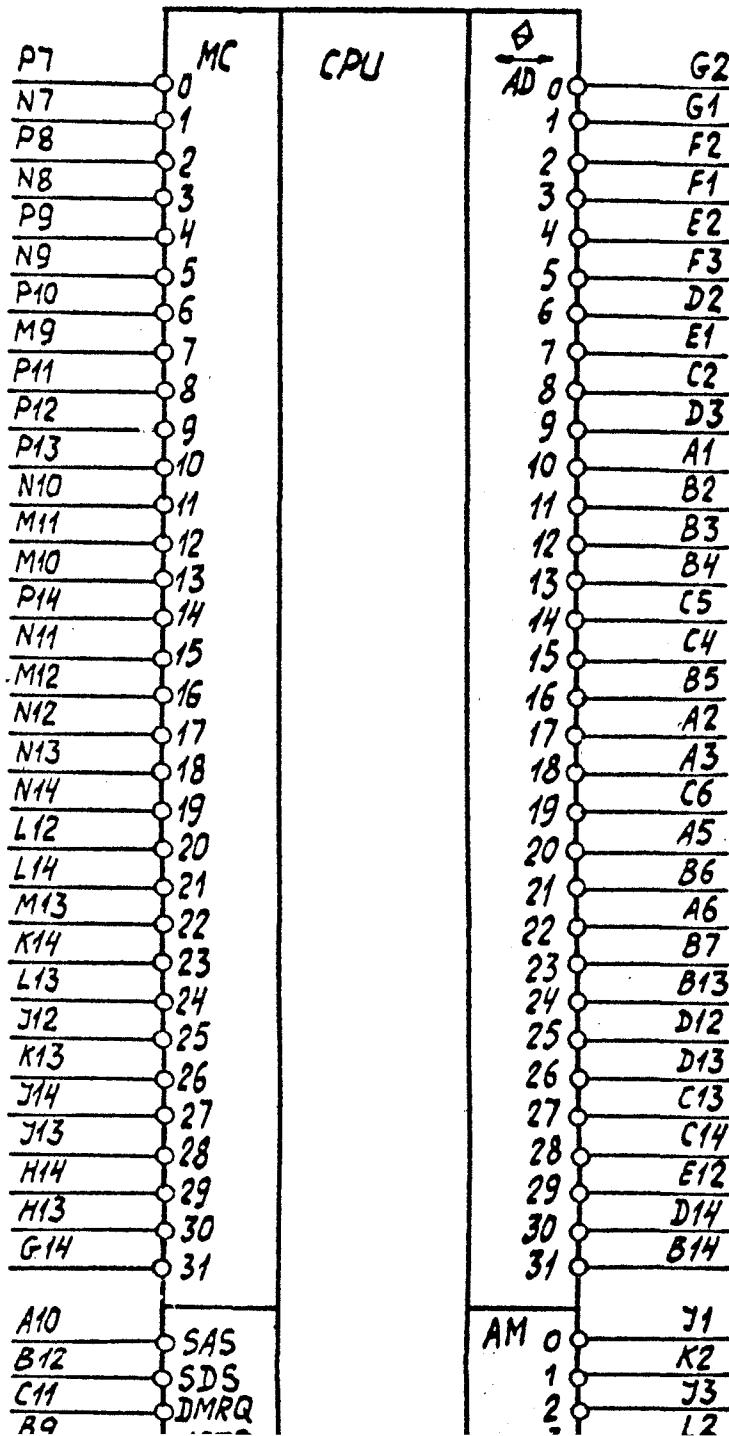
Инв. № подл.	Подл. и дата	Взам. инв. №	Инв. № подл.
93.353.1	25.12.92 (Зав.)	93.353	

Лист
5

Изм.	Зам.	ЦЧН633-92	Выс. Сост.	220792
Изм.	Лист	№ докум.	Подл.	Дата

ЩИЗ.480.319 ТО

Условное графическое обозначение ИМР



ФОРМАТ А3

Уим	Номен	Ном.	Лам
-----	-------	------	-----

НЧ3.480.319 ТО

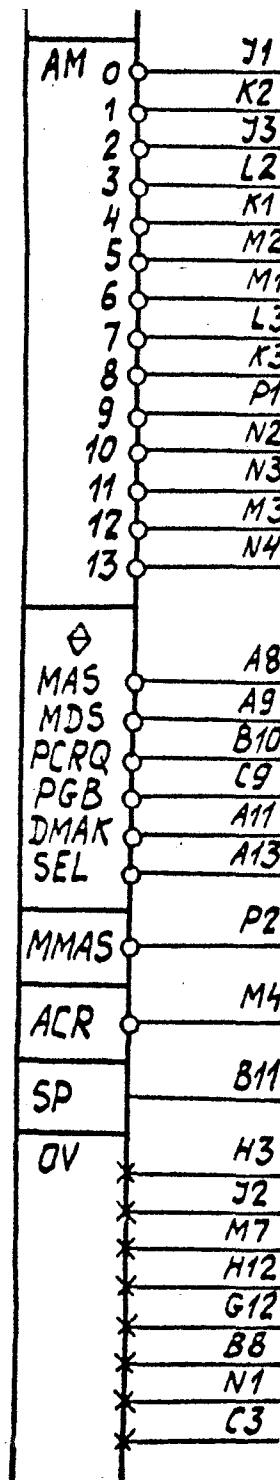
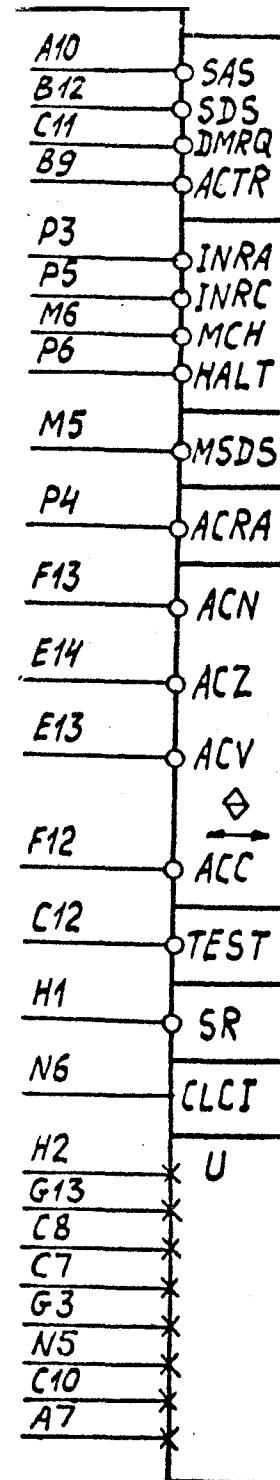


Рис.1.

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата
93353	Фев 19.04.91			

ИЗМ.	Лист	№ документа

ИП3.480.319 ТО

ТАБЛИЦА НАЗНАЧЕНИЯ ВЫВОДОВ

Таблица 1

Номер вывода	Обозначение	Тип вывода	Наименование
H3	OV		Общий вывод
J2	OV		Общий вывод
M7	OV		Общий вывод
H12	OV		Общий вывод
GI2	OV		Общий вывод
B8	OV		Общий вывод
H2	U		Выход питания от источника напряжения
GI3	U		Выход питания от источника напряжения
C8	U		Выход питания от источника напряжения
C7	U		Выход питания от источника напряжения
G3	U		Выход питания от источника напряжения
P7	MC0	Вход	Нулевой разряд микрокоманды
N7	MC1	Вход	Первый разряд микрокоманды
P8	MC2	Вход	Второй разряд микрокоманды
N8	MC3	Вход	Третий разряд микрокоманды
P9	MC4	Вход	Четвертый разряд микрокоманды

Продолжение табл. 1

Номер вывода	Обозначение	Тип вывода	Наименование
N9	MC5	Вход	Пятый разряд микрокоманды
PI0	MC6	Вход	Шестой разряд микрокоманды
M9	MC7	Вход	Седьмой разряд микрокоманды
PII	MC8	Вход	Восьмой разряд микрокоманды
PI2	MC9	Вход	Девятый разряд микрокоманды
PI3	MC10	Вход	Десятый разряд микрокоманды
NIO	MC11	Вход	Одиннадцатый разряд микрокоманды
MII	MC12	Вход	Двенадцатый разряд микрокоманды
MIO	MC13	Вход	Тринадцатый разряд микрокоманды
PI4	MC14	Вход	Четырнадцатый разряд микрокоманды
NII	MC15	Вход	Пятнадцатый разряд микрокоманды
MI2	MC16	Вход	Шестнадцатый разряд микрокоманды
NI2	MC17	Вход	Семнадцатый разряд микрокоманды
NI3	MC18	Вход	Восемнадцатый разряд микрокоманды
NI4	MC19	Вход	Девятнадцатый разряд микрокоманды
LI2	MC20	Вход	Двадцатый разряд микрокоманды
LI4	MC21	Вход	Двадцать первый разряд микрокоманды

Продолжение табл. 1

Номер вывода	Обозначение	Тип вывода	Наименование
M13	MC22	Вход	Двадцать второй разряд микрокоманды
K14	MC23	Вход	Двадцать третий разряд микрокоманды
L13	MC24	Вход	Двадцать четвертый разряд микрокоманды
J12	MC25	Вход	Двадцать пятый разряд микрокоманды
K13	MC26	Вход	Двадцать шестой разряд микрокоманды
J14	MC27	Вход	Двадцать седьмой разряд микрокоманды
J13	MC28	Вход	Двадцать восьмой разряд микрокоманды
H14	MC29	Вход	Двадцать девятый разряд микрокоманды
H13	MC30	Вход	Тридцатый разряд микрокоманды
G14	MC31	Вход	Тридцать первый разряд микрокоманды
A10	SAS	Вход	Сигнал подтверждения приема адреса подчиненным устройством
B12	SDS	Вход	Сигнал обработки данных подчиненным устройством
CII	DMRQ	Вход	Сигнал запроса на прямой доступ к памяти
B5	ACTR	Вход	Сигнал захвата магистрали акселератором
P3	INRA	Вход	Сигнал "Прерывание от адаптера магистрали"
P5	INMC	Вход	Сигнал "Прерывание от контроллера памяти"
M6	MCH	Вход	Сигнал "Машинный сбой"

Инв. № подл.	подп. и дата	взам. инв. №	инв. № будл.	подп. и дата
93353	дек 19.04.91			

Продолжение табл. 1

Номер вывода	Обозначение	Тип вывода	Наименование
P6	HALT	Вход	Сигнал "Переход в пультовый режим"
M5	MSDS	Вход	Сигнал сопровождения микрокоманды
P4	ACRA	Вход	Сигнал готовности акселератора
F13	ACN	Вход	Признак "Результат отрицателен" от акселератора
E14	ACZ	Вход	Признак "Результат равен нулю" от акселератора
E13	ACV	Вход	Признак "Переполнение" от акселератора
F12	ACC	Вход-выход	Признак "Расширение" от акселератора
C12	TEST	Вход	Признак тестирования
H1	SR	Вход	Сигнал установки в исходное состояние
N6	CLCI	Вход	Сигнал тактовой частоты
G2	ADO	Вход-выход	Нулевой разряд адреса-данных 32-разрядной шины
GI	ADI	Вход-выход	Первый разряд адреса-данных 32-разрядной шины
F2	AD2	Вход-выход	Второй разряд адреса-данных 32-разрядной шины
F1	AD3	Вход-выход	Третий разряд адреса-данных 32-разрядной шины
E2	AD4	Вход-выход	Четвертый разряд адреса-данных 32-разрядной шины
F3	AD5	Вход-выход	Пятый разряд адреса-данных 32-разрядной шины
D2	AD6	Вход-выход	Шестой разряд адреса-данных 32-разрядной шины

ИМЗ-480-319 TO

Продолжение табл. 1

Номер вывода	Обозначение	Тип вывода	Наименование
E1	AD7	Вход-выход	Седьмой разряд адреса-данных 32-разрядной шины
C2	AD8	Вход-выход	Восьмой разряд адреса-данных 32-разрядной шины
D3	AD9	Вход-выход	Девятый разряд адреса-данных 32-разрядной шины
A1	AD10	Вход-выход	Десятый разряд адреса-данных 32-разрядной шины
B2	AD11	Вход-выход	Одиннадцатый разряд адреса-данных 32-разрядной шины
B3	AD12	Вход-выход	Двенадцатый разряд адреса-данных 32-разрядной шины
B4	AD13	Вход-выход	Тринадцатый разряд адреса-данных 32-разрядной шины
C5	AD14	Вход-выход	Четырнадцатый разряд адреса-данных 32-разрядной шины
C4	AD15	Вход-выход	Пятнадцатый разряд адреса-данных 32-разрядной шины
B5	AD16	Вход-выход	Шестнадцатый разряд адреса-данных 32-разрядной шины
A2	AD17	Вход-выход	Семнадцатый разряд адреса-данных 32-разрядной шины
A3	AD18	Вход-выход	Восемнадцатый разряд адреса-данных 32-разрядной шины
C6	AD19	Вход-выход	Девятнадцатый разряд адреса-данных 32-разрядной шины
A5	AD20	Вход-выход	Двадцатый разряд адреса-данных 32-разрядной шины
B6	AD21	Вход-выход	Двадцать первый разряд адреса-данных 32-разрядной шины
A6	AD22	Вход-выход	Двадцать второй разряд адреса-данных 32-разрядной шины
B7	AD23	Вход-выход	Двадцать третий разряд адреса-данных 32-разрядной шин

Продолжение табл. 1

Номер вывода	Обозначение	Тип вывода	Наименование
В13	AD24	Вход-выход	Двадцать четвертый разряд адреса-данных 32-разрядной шины
DI2	AD25	Вход-выход	Двадцать пятый разряд адреса-данных 32-разрядной шины
DI3	AD26	Вход-выход	Двадцать шестой разряд адреса-данных 32-разрядной шины
CI3	AD27	Вход-выход	Двадцать седьмой разряд адреса-данных 32-разрядной шины
CI4	AD28	Вход-выход	Двадцать восьмой разряд адреса-данных 32-разрядной шины
EI2	AD29	Вход-выход	Двадцать девятый разряд адреса-данных 32-разрядной шины
DI4	AD30	Вход-выход	Тридцатый разряд адреса-данных 32-разрядной шины
В14	AD31	Вход-выход	Тридцать первый разряд адреса-данных 32-разрядной шины
J1	AM0	Выход	Нулевой разряд адреса микрокоманды
K2	AM1	Выход	Первый разряд адреса микрокоманды
J3	AM2	Выход	Второй разряд адреса микрокоманды
L2	AM3	Выход	Третий разряд адреса микрокоманды
K1	AM4	Выход	Четвертый разряд адреса микрокоманды
M2	AM5	Выход	Пятый разряд адреса микрокоманды
M1	AM6	Выход	Шестой разряд адреса микрокоманды
L3	AM7	Выход	Седьмой разряд адреса микрокоманды
K3	AM8	Выход	Восьмой разряд адреса микрокоманды

93353 / Авт 19.04.91

Продолжение табл. 1

Номер вывода	Обозначение	Тип вывода	Наименование
P1	AM9	Выход	Девятый разряд адреса микрокоманды
N2	AM10	Выход	Десятый разряд адреса микрокоманды
N3	AM11	Выход	Одиннадцатый разряд адреса микрокоманды
M3	AM12	Выход	Двенадцатый разряд адреса микрокоманды
N4	AM13	Выход	Тринадцатый разряд адреса микрокоманды
A8	MAS	Выход	Сигнал сопровождения адреса от главного устройства
A9	MDS	Выход	Сигнал сопровождения данных от главного устройства
B10	PCRQ	Выход	Сигнал запроса команды по счетчику команд
C9	PGB	Выход	Сигнал "Граница страницы памяти"
A11	DMAK	Выход	Сигнал предоставления прямого доступа к памяти
A13	SEL	Выход	Сигнал чтения регистра внешних условий
P2	MMAS	Выход	Сигнал сопровождения адреса для накопителя микрокоманд
M4	ACR	Выход	Сигнал сброса акселератора
B11	SP	Выход	Резервный
LI	-		Не используется
N5	U		Вывод питания от источника напряжения
NI	OV		Общий вывод

ШИЗ.480.319 ТО

Продолжение табл. 1

Номер вывода	Обозначение	Тип вывода	Наименование
M8	-		Не используется
MI4	-		Не используется
KI2	-		Не используется
FI4	-		Не используется
AI4	-		Не используется
AI2	-		Не используется
C10	U		Вывод питания от источника напряжения
A4	-		Не используется
A7	U		Вывод питания от источника напряжения
BI	-		Не используется
C3	0V		Общий вывод
CI	-		Не используется
E3	-		Не используется
DI	-		Не используется

Таблица 2

Наименование параметра, единица измерения, режим измерения	Буквенное обозначе- ние	Норма	Темпера- тура, $^{\circ}\text{C}$
		не менее	не более
Выходное напряжение низкого уровня, В при $I_{OL} = 2,5 \text{ мА}$	U_{OL}	-	0,45
		-	0,5
Выходное напряжение высокого уровня, В при $I_{OH} = -0,2 \text{ мА}$	U_{OH}	4,05	25 ± 10
		4,0	-60 85
Ток потребления, мА	I_{CC}	-	8,0
		-	10,0
Ток потребления динамический, мА	I_{CCD}	-	170
		-	200
Ток утечки на входе, мкА	I_{LI}	-	3,0
		-	10,0
Ток утечки на выходе, мкА	I_{LO}	-	10,0
		-	50,0
Частота следования импульсов тактовых сигналов, МГц	f_C	10,0	25 ± 10 -60 85
Входная емкость, пФ	C_I	-	12,0
Выходная емкость, пФ	C_O	-	12,0
Емкость входа/выхода, пФ	$C_{I/O}$	-	14,0
Время выполнения операции "сложение" при регистровом методе адресации, нс	t_{ADD}	-	200
			25 ± 10 -60 85

1	Зарн	10006-122	Б.Ч.1	3277
2	Индикаторы	10006	Б.Ч.1	6210

МК3.490.519 Т0

Лист

Таблица 3

Наименование параметра, единица измерения	Буквенное обозначение	Норма	
		не менее	не более
Напряжение питания, В	U_{CC}	4,5	5,5
Входное напряжение высокого уровня, В	U_{IH}	$(0,85 \cdot U_{CC})^*$	U_{CC}^{**}
Входное напряжение низкого уровня, В	U_{IL}	0	0,8*
Напряжение на любом входе, В	U_I	0	U_{CC}^{**}
Выходной ток низкого уровня, мА	I_{OL}	-	2,5
Выходной ток высокого уровня, мА	I_{OH}	-	0,2
Емкость нагрузки, пФ	C_L	-	100
Время фронта нарастания и спада входных сигналов, нс	t_{LH}, t_{HL}	-	$0,1 \cdot T_{CLK}$

* - с учетом всех видов помех

** - но не более конкретного значения напряжения питания, приложенного к микросхеме

3) обработка данных специального вида:

инструкции манипуляции очередью;

инструкции обработки адресов;

программируемые пользователем инструкции загрузки и сохранения универсальных регистров;

4) для быстрого выполнения особых функций операционной системы, позволяющие: загружать и сохранять переменные контексты, найти операционной системе выполняемый процесс с наивысшим приоритетом;

5) предназначенные специально для языка высокого уровня.

ЦПР оперирует со следующими типами данных:

Целое: байт (8 бит), слово (16 бит), длинное слово (32 бита), четверное слово (64 бита).

Плавающая запятая: 4 байта (F-формат), 8 байтов (D-формат), 8 байтов (G-формат).

Упакованное десятичное: строка байтов (до 31 десятичной цифры, по две цифры в байте).

Строка символов: строка байтов, интерпретируемая как коды символов (до 64 Кбайт). Числовая строка является строкой символьных кодов для десятичных чисел (до 31 цифры).

Биты и поля битов: длина поля переменна от 0 до 32 бит (определяется пользователем).

Так как ЦПР полностью реализует архитектуру центральной части ЭВМ VAX-II/750, включая эмуляцию системы команд, то на его основе могут создаваться вычислительные средства, совместимые по математическому обеспечению на уровне операционных систем с ЭВМ ряда VAX.

I.I.4. Микросхемы ЦПР Л1839ВМ1 соответствуют техническим условиям АЕЯР.431200.005-01 ТУ.

I.I.5. В микропроцессоре реализован конвейерный метод обработки команд на макро- и микро-программном уровнях. Осуществляется опережающий прием команд в количестве трех 32-разрядных слов команды. В пределах страницы (512 байт) прием команд безадресный с цик-

лом 200 нс. Совмещение операций при выполнении команд регистр-ре-
гистр показано на рис.2.

Совмещение операций при выполнении
команд регистр-регистр

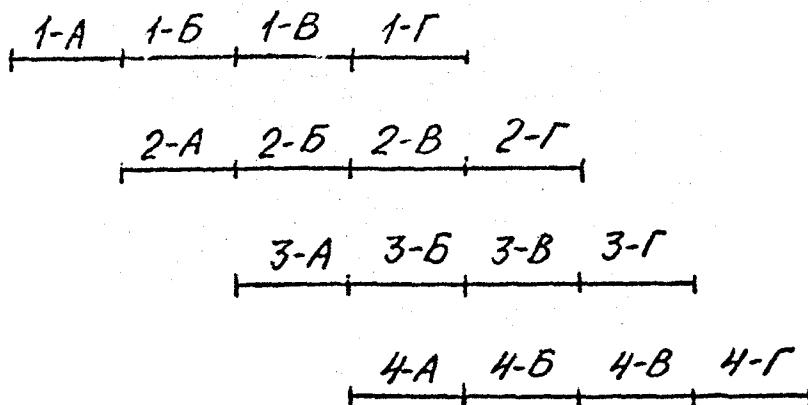


Рис. 2.

Здесь цифрами (I-4) обозначены номера последовательно
выполняемых команд, а буквами – операции:

- А – прием команды;
- Б – дешифрование команды;
- В – выборка микрокоманды;
- Г – исполнение операции.

Данные принимаются в процессор на буфер, который организо-
ван в виде стека "первый вошел – первый вышел". Этот стек со-
стоит из четырех 32-разрядных ячеек и его применение позволяет
производить накопление данных при выполнении операции с памятью.
Если в команде много операндов, это позволяет сократить время
выполнения команды.

Преобразование виртуального адреса в физический в блоке
диспетчера памяти выполняется полностью аппаратно, параллельно

1	ЗЭМ	ЦИ633-9.1	Б.С.С.,	17/52
Изм	Лист	№ ЗОКУМ	Подп.	Дата

с другими операциями процессора. Прием и модификация описателей страниц (PTE) автоматически вставляется в общий поток обменов по внешней магистрали. В ДП имеется ассоциативный буфер PTE на 16 ячеек. Как показали исследования фирмы DEC, такого объема буфера достаточно для эффективного выполнения типовых программ.

В микропроцессоре имеются аппаратные и микропрограммные средства для организации эффективной связи с дополнительной микросхемой, на которой аппаратно выполняются операции умножения и деления целых чисел и операции над числами в формате с плавающей запятой.

I.2. Структурная схема и общие принципы функционирования ЦПР состоит из следующих функциональных блоков (рис.3):

- дешифратор команд (ДК);
- блок приема и формирования микрокоманд (БПМК);
- операционный блок (ОБ);
- интерфейсный блок (ИБ);
- диспетчер памяти (ДП);
- блок анализа приоритетов прерываний (БПР).

Блоки ДК, ОБ и ДП связаны с внутренней 32-разрядной магистралью процессора, по которой они могут обмениваться информацией как между собой, так и с внешней магистралью. Внутренняя магистраль связана с внешней элементами входа-выхода, которые управляются интерфейсным блоком (ИБ). По этой магистрали передаются адреса, данные и команды.

Передача различного типа информации мультиплексирована по времени. Для приема микрокоманд в микропроцессоре имеются раздельные каналы выдачи адреса микрокоманды (14 разрядов) и приема микрокоманды (32 разряда). Этими каналами управляет БПМК.

TEST

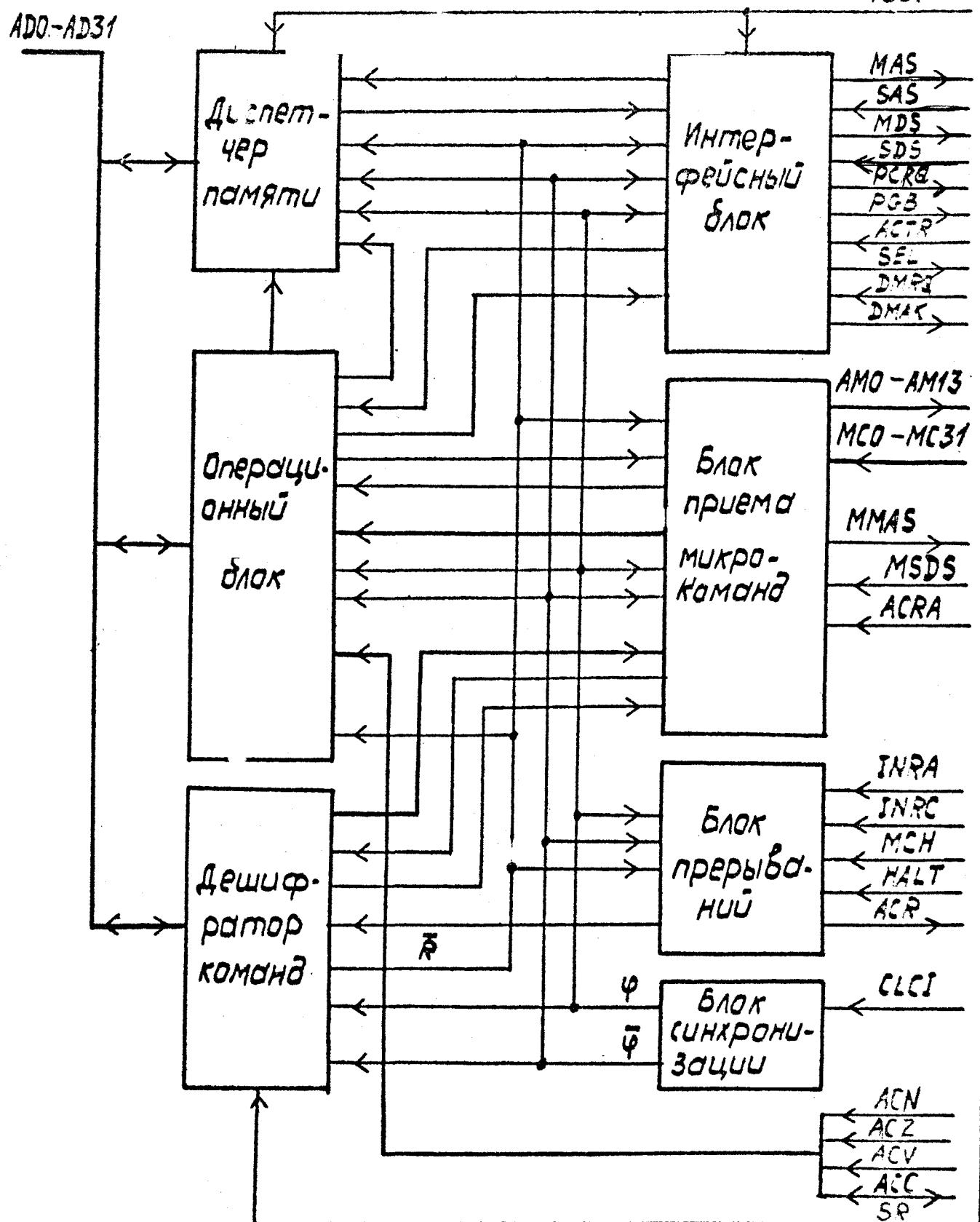


Рис. 3,

Начальный пуск микропроцессора выполняется микропрограммно. По переходу сигнала на выводе SR из низкого в высокий уровень в ДМК вырабатывается первый адрес микропрограммы начального пуска и процессор переходит к выполнению этой микропрограммы.

Прием первой команды выполняется всегда в процедуре "адресное чтение". Физический адрес этой команды запоминается контроллером памяти в счетчике физического адреса команды. Дальнейший прием команд идет по этому счетчику, который автоматически инкрементируется на четыре после чтения каждой команды. Переход адреса команды на новую страницу отслеживается в процессоре и при каждом таком переходе прием команды выполняется в процедуре адресного чтения с проверкой доступности страницы в диспетчере памяти.

При выполнении команд, если количество операндов не больше четырех, сначала выполняется адресация всех операндов и прием их на буфер данных. Операнды могут находиться в памяти по адресам данных, в регистрах общего назначения (РОН) процессора или в теле команды. Данные из памяти всегда принимаются в буферный регистр данных (БРД). В формате "регистр-регистр" и "регистр-память" данные из РОН не пересыпаются в БРД, действия над ними выполняются непосредственно в РОН. Во всех остальных случаях данные пересыпаются в БРД. Если операнд находится в теле команды, то при дешифрации он выделяется из команды на специальный регистр и затем пересыпается в БРД. Если количество операндов больше четырех, то выполняется прием четырех операндов, затем процесс адресации прерывается для освобождения БРД, после чего прием продолжается. Максимальное количество операндов - 6.

При выполнении косвенной адресации адрес из памяти принимается не в БРД, а в специальный регистр косвенной адресации (РКА), из которого он затем пересыпается в регистр виртуального адреса (РВА). Это сделано потому, что принятый адрес нужно сразу же после приема использовать для продолжения адресации, а из БРД данные доступны по чтению только последовательно в порядке их приема, поскольку БРД организован по принципу стека.

Команды принимаются на буфер команд, который также организован как стек из трех 32-разрядных ячеек. Команды принимаются в буфер без выравнивания по началу слова, т.е. команда может начинаться с любого из четырех байт принимаемого 32-разрядного слова. За один цикл ДШК анализирует от 1 до 4 байт слова команды. Для начала обработки команды в буфере должны быть загружены две ячейки, поскольку анализируемые байты могут находиться в первой и второй ячейках, третья нужна для опережающего приема команды.

Счетчик команд (СК) в процессоре состоит из трех регистров. В первом регистре хранится виртуальный адрес начала команды, во втором – адрес анализируемого кода операции или моды адресации, и в третьем – адрес анализируемой в текущем цикле ДШК группы байтов команды. При завершении обработки кода операции или моды адресации содержимое третьего регистра переписывается во второй, а при завершении обработки всей команды содержимое второго регистра переписывается в первый. При начальном приеме команды все три регистра загружаются одним и тем же значением адреса команды. Для адресации команд СК используется только в командах ветвления. Для адресации переходов в командах ветвления и для адресации операндов в соответствующих модах адресации используется второй регистр СК. Для отслеживания границы страницы и адресации команды на этой границе используется специальный

Лист №005м.	1/одн.	Дата	

регистр-счетчик виртуального адреса команды (РВАК). Этот регистр загружается одновременно с загрузкой СК и инкрементируется на четыре после каждого цикла приема команды. Если адрес команды находится внутри страницы, прием команд выполняется в безадресном цикле обмена. Когда адрес команды совпадает с адресом последнего слова в странице, в РВАК вырабатывается специальный сигнал, по которому адрес, соответствующий адресу первого слова в следующей странице, переписывается из РВАК в РВА и инициируется адресный обмен с преобразованием адреса в ДП.

Все блоки процессора синхронизируются единым сигналом тактовой частоты и связаны между собой сигналами управления, обеспечивающими конвейерную микропрограммно-аппаратную обработку команд.

I.3. Дешифрация команд

I.3.1. Функции дешифратора команд

I) формирование начального адреса микропрограммы содержательной обработки команды;

2) формирование начального адреса микропрограммы адресации очередного операнда команды;

3) управление по выделению из области программы смещений адресации и констант;

4) управление мультиплексором буфера команд для загрузки полей кода операции и кода моды адресации очередного операнда.

I.3.2. Типы команд по дешифрации

I) команды "без operandов"

(HALT; NOP; REI; BPT; RET; LDPCtx; SVPCTx; XFC; RSB);

2) простые команды - команды с количеством operandов от

I до 3;

3) длинные команды - команды с количеством operandов боль-

шё трёх;

- 4) команды перехода по смещению и без операндов (*BRRB; BRW; BEQL ...*);
- 5) команды перехода по смещению и с операндами (*ACBB, SOBGEQ ...*).

На рис. 4 показан общий алгоритм дешифрации команд.

На рис. 4 приняты следующие обозначения:

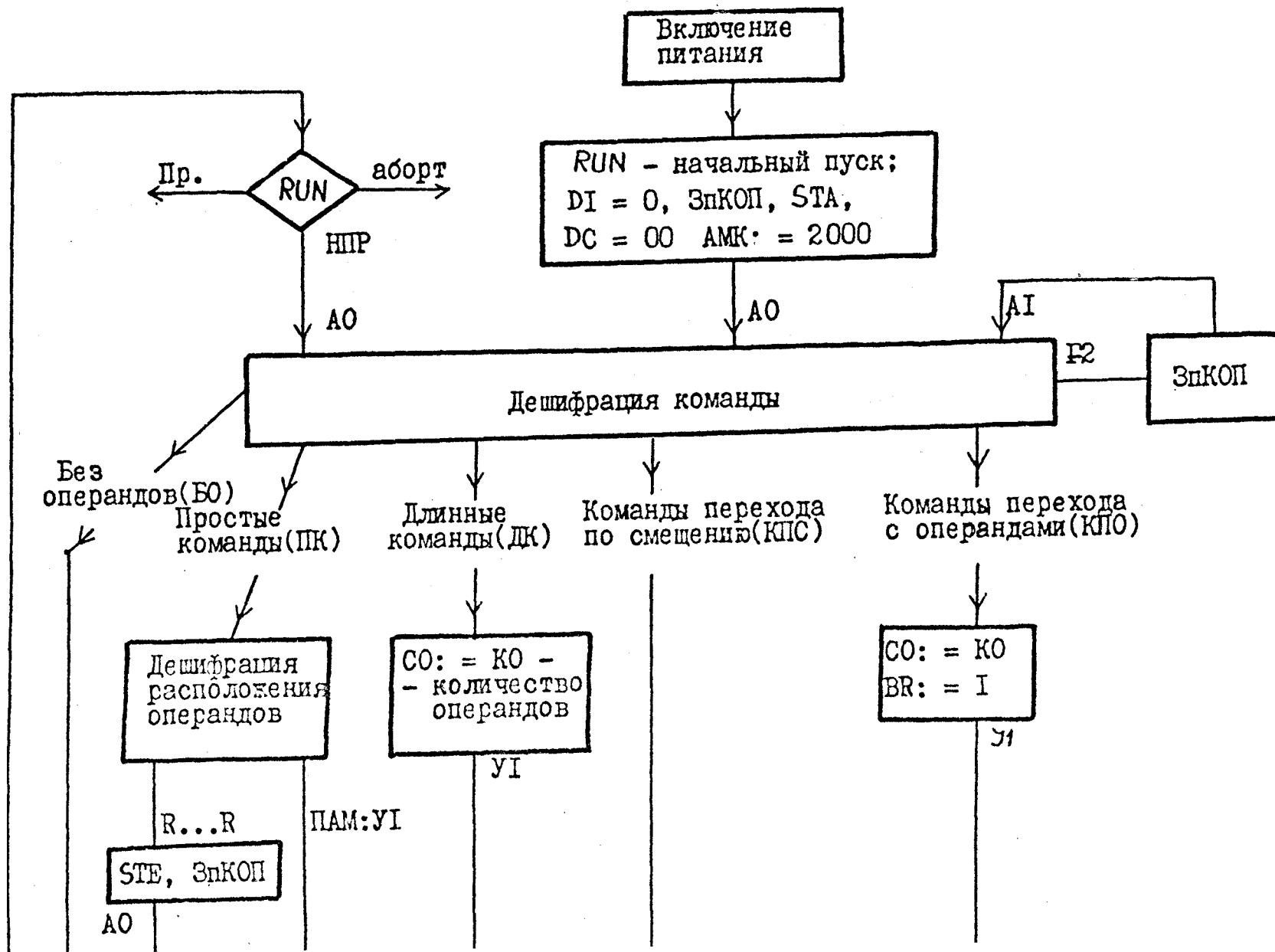
- ПУ — передача управления;
ЗпКОП — загрузить регистр — "поле КОП" на входе ПЛМ-ДИК;
RUN — поле кода режима на входе ПЛМ;
ПР — прерывание;
НПР — нет прерываний;
F2 — признак кода команды из пространства расширения;
Б0 — команды без операндов;
ПК — простые команды;
ДК — длинные команды;
КПО — команды перехода с операндами;
КПС — команды перехода по смещению (байт или слово);
РР — расположение operandov — все на регистрах;
П — расположение operandов — хотя бы один в памяти;
СО — счетчик operandов;
КО — количество operandов;
STA — старт адресации.

I.4. Операционный блок

Операционный блок (ОБ) предназначен для формирования адресов команд и operandов, для выполнения арифметических и логических операций над operandами, а также для хранения адресов, operandов и результатов выполнения операций на регистрах. Оп-

Лист №100 к ум.	160н.	дат	

Алгоритм дешифрации команд



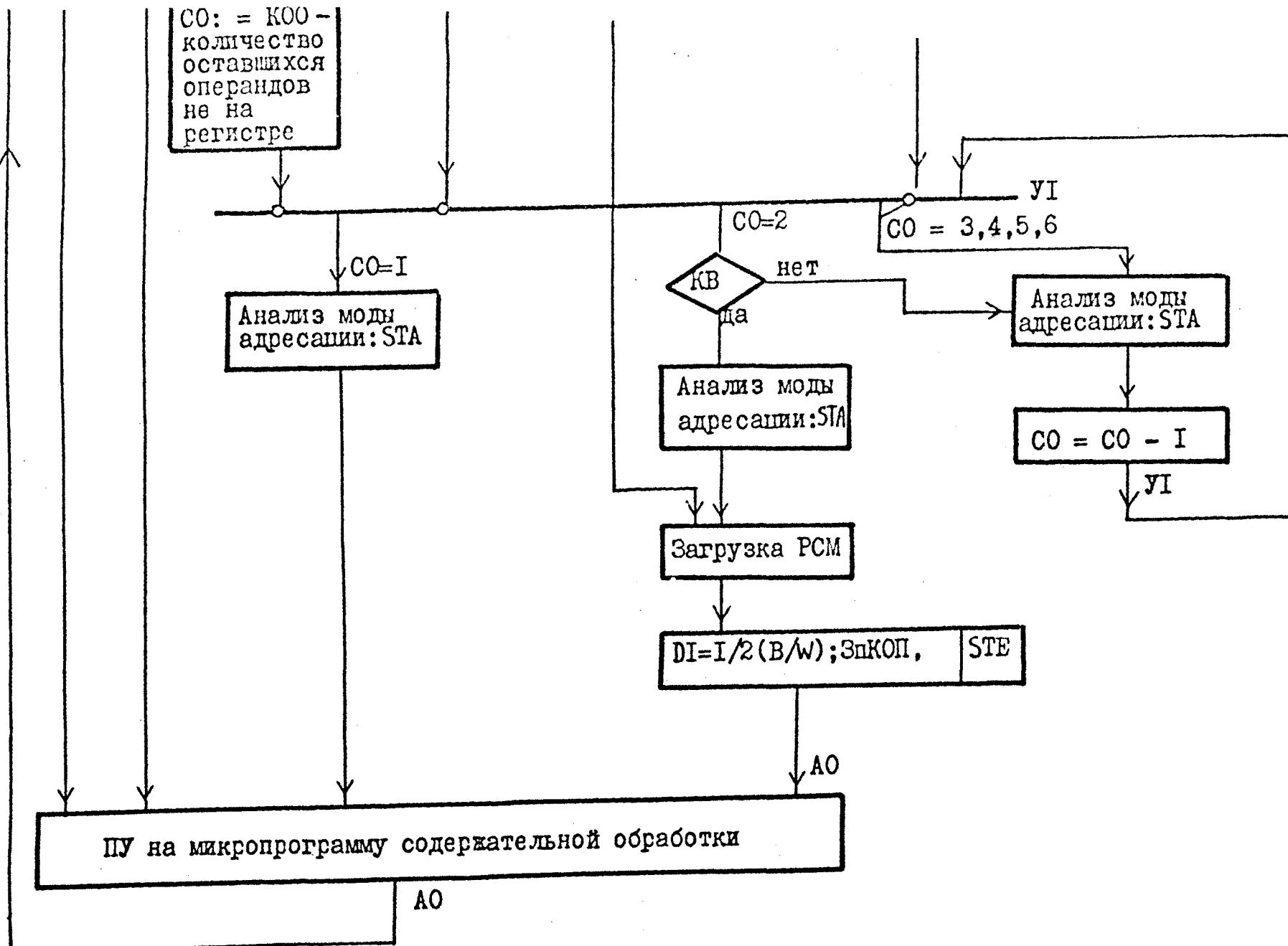


Рис. 4.

рационный блок состоит из операционной части, регистровой части и схем управления.

Операционная часть операционного блока

Операционная часть операционного блока выполняет арифметические и логические операции над операндами, осуществляет временное хранение результатов операций, формирует состояния N , Z , V , S по результатам операций. Операционная часть ОБ состоит:

- 1) Буферные регистры;
- 2) Арифметико-логическое устройство (АЛУ);
- 3) Регистр результата операции (РРОП);
- 4) Схема формирования состояний (СФС);
- 5) Декодаторы.

Работой ОБ управляет блок синхронизации ОБ (БСОБ). Временная диаграмма работы операционной части ОБ представлена на рис. 5.

Структурная схема операционной части ОБ представлена на рис. 6.

По сигналу $SRD1\& \varphi$ (чтение I) на шину X и \bar{Y} в соответствии с адресами, указанными в поле чтения ($RFX0 - RFX4$, $RFY0 - RFY4$) тела микрокоманды, читаются регистры (в одноадресной микрокоманде на шину \bar{Y} читается ячейка блока констант). Информация на шинах X и \bar{Y} защелкивается квазистатическими триггерами, стоящими на входе буферных регистров, после чего чтение регистров (блока констант) прекращается. Подброс шин X и \bar{Y} , а следовательно и сброс защелки осуществляется сигналом $SETXY$. Одновременно с сигналами чтения регистров вырабатываются сигналы записи, по которым операнды с шин X и \bar{Y} записываются в буферные регистры. В соответствии с кодом операции (OCIR-OCBR) в

Временная диаграмма работы операционной
части ОБ

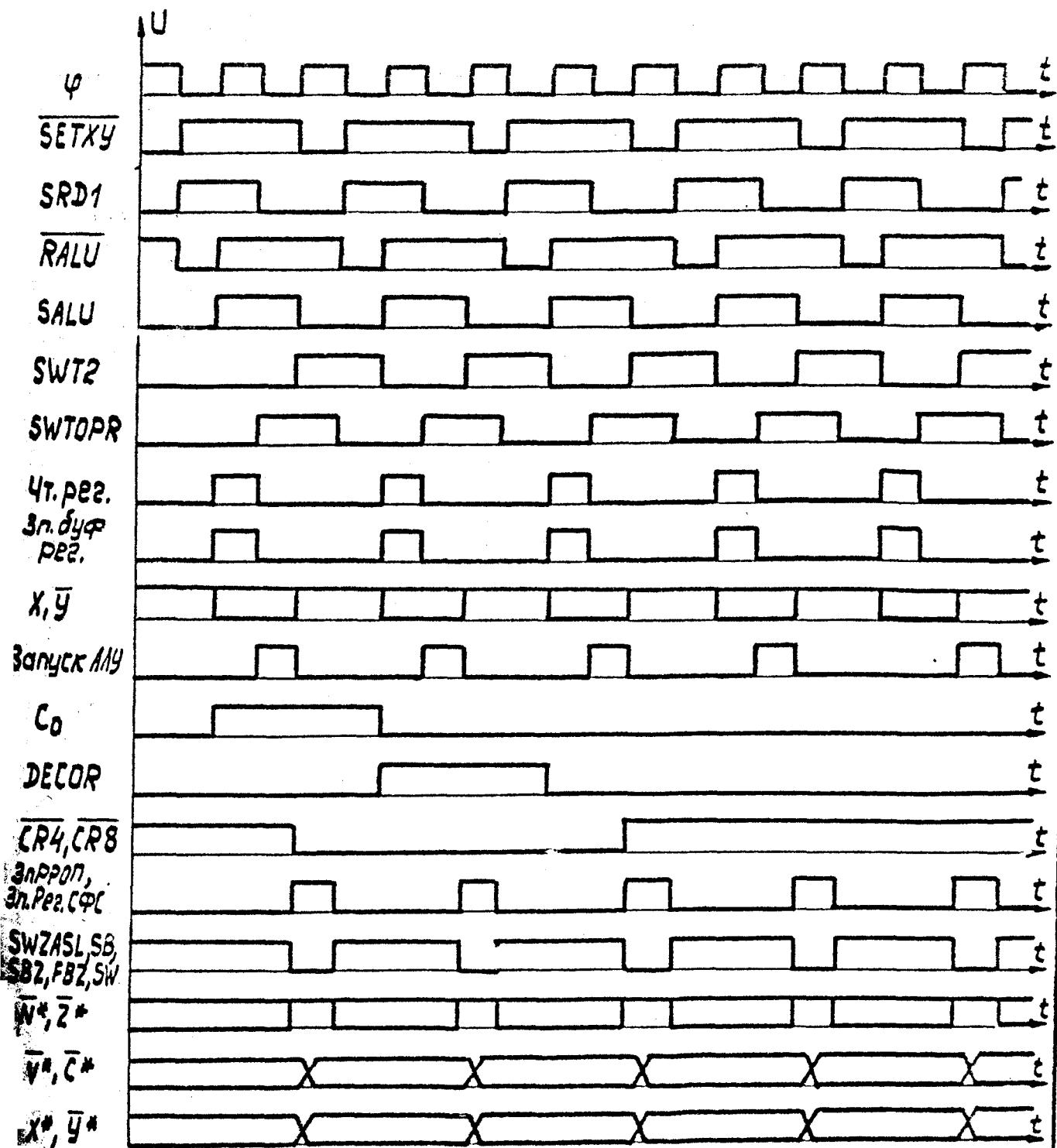
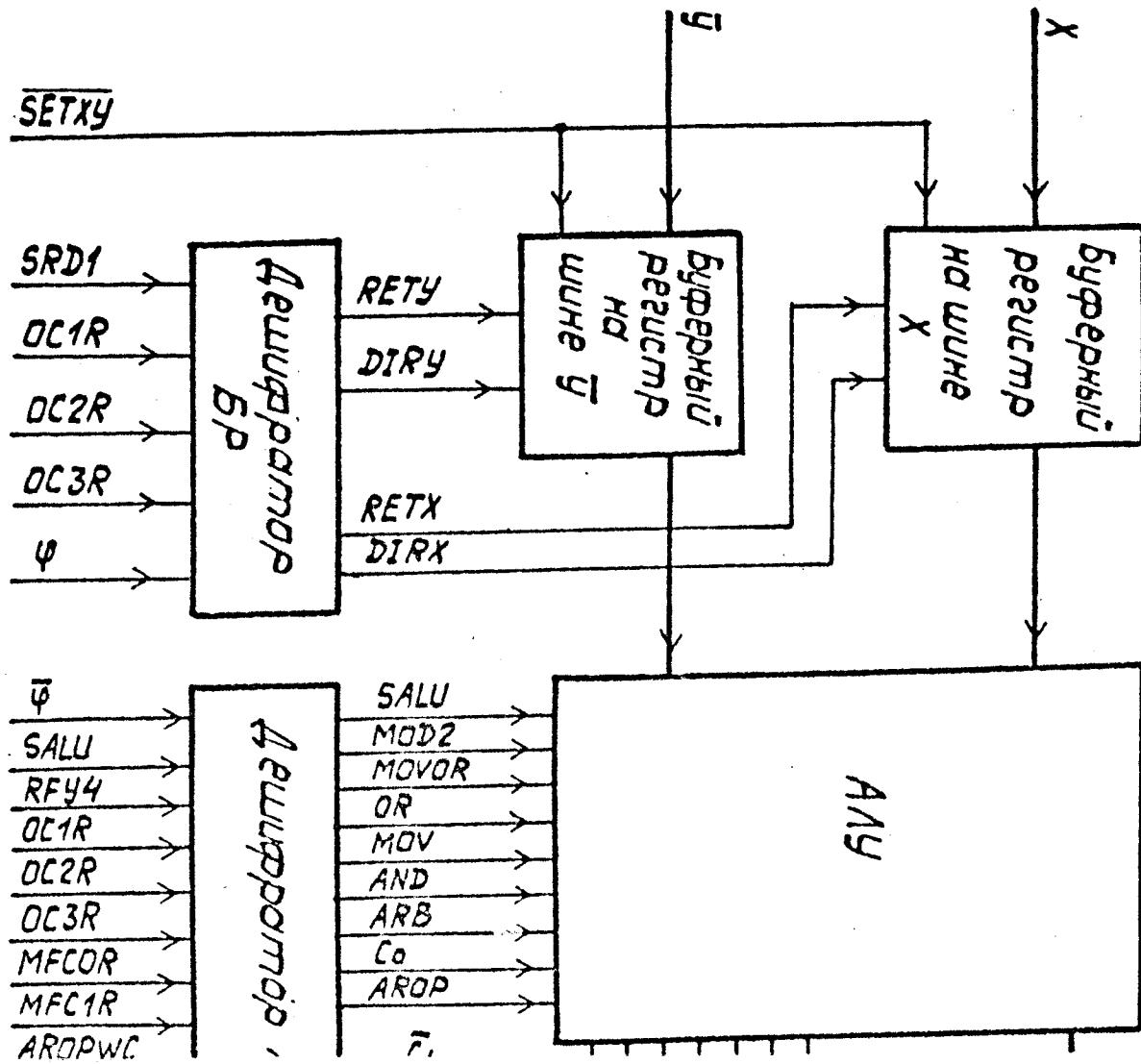
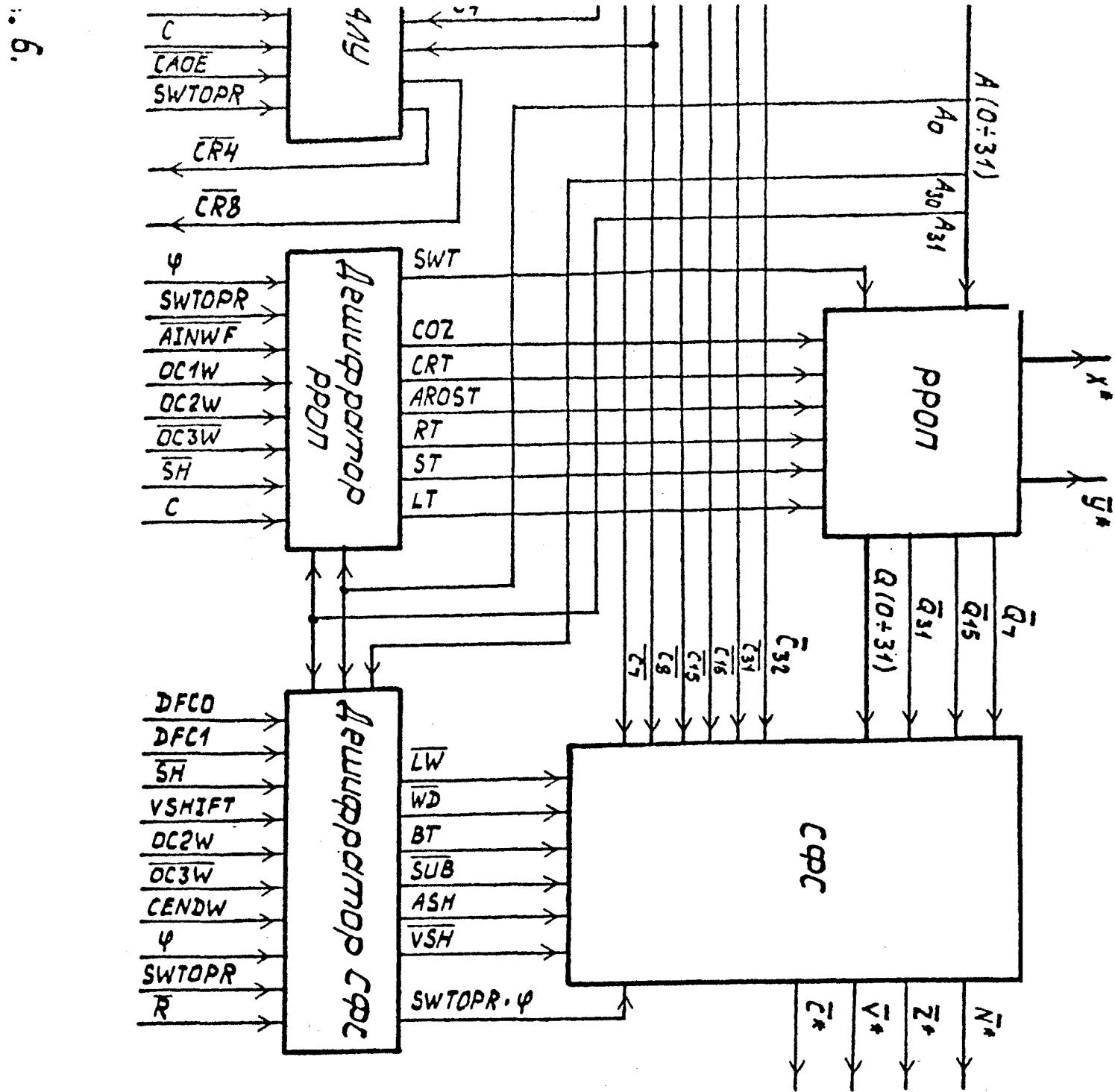


Рис. 5.

Структурная схема





буферные регистры записываются прямые или инверсные значения операндов. По сигналу $SALU\&\bar{Q}$ (разрешение запуска АЛУ) запускается арифметико-логическое устройство, которое в зависимости от кода операции ($RFY4$, $OC1R$ - $OC3R$, $AROPWC$) и формата микрокоманды ($MFCOR$, $MFC1R$) выполняет над операндами ту или иную операцию. Результат выполнения операции сигналом $SWTOPR\&Q$ записывается в регистр результата операции. РРОП кроме функции временного хранения результата выполняет функцию сдвигателя в микрокомандах сдвига. На основе полученного результата СФС формирует признаки \bar{N}^* , \bar{Z}^* , \bar{V}^* , \bar{C}^* , которые подаются на регистр состояний (РСП) и на схему образования условий ветвления.

Результат выполнения операции с РРОП через усилители подается на параллельную шину X^* , \bar{y}^* , откуда по сигналу $SWT2\&\bar{Q}$ (запись 2), в соответствии с адресом, указанным в поле записи ($WF0$ - $WF4$) тела микрокоманды, записывается в регистр.

I.5. Прерывания и исключительные состояния

I.5.1. При работе системы в ней могут возникать особые ситуации, которые приводят к изменениям в текущем выполняемом процессе. При этом процессор выполняет два основных действия:

1) сохраняет информацию о текущем процессе (в основном, это делается путем занесения в стек счетчика команд (*program counter* - PC) и длинного слова состояния процессора (*processor status Longword* - PSL) с целью последующего возобновления данного процесса);

2) передает управление от потока команд текущего процесса к определенным участкам программного обеспечения для обслуживания возникших событий.

Эти действия выполняются микропрограммно. Существует два основных типа подобных событий: прерывания (*Interrupt*)

МП-8000	МП-8000	МП-8000

и исключительные состояния (*exceptions*).

Прерывания – это асинхронные события, которые являются главными по отношению к другим (не текущим процессам) или даже по отношению к целой системе и они, соответственно, обслуживаются в контексте всей системы.

Исключительные состояния – это синхронные события, которые являются главными по отношению к текущему процессу и обычно обслуживаются в контексте текущего процесса.

Прерывания являются асинхронными событиями, поэтому для возможности обработки одновременно возникших запросов прерываний, в процессоре существует логика приоритетов. Приоритет, связанный с прерыванием, называется "уровнем приоритета прерывания" (*IPL - interrupt priority Level*). В любой момент времени будет обслуживаться прерывание с наивысшим приоритетом. *IPL* изменяется от 0 до 15, то есть существует 32 уровня приоритета прерывания.

1.5.2. Для исключительных состояний (кроме особых фатальных ситуаций, влияющих на работу всей системы – им сопоставляется наивысший *IPL* , равный 15) логика приоритетов не нужна, так как эти события синхронны по своей природе, и поэтому, возникают только в определенных, обусловленных набором выполняющихся команд, точках.

Существует три типа исключительных состояний:

I) "ловушка" (*trap*) – это исключительное состояние, которое возникает в конце команды, которая ее вызвала. Поэтому РС, сохраняемый в стеке, указывает на следующую команду, которая должна была бы нормально выполняться. Любая программа может разрешить или запретить некоторые из ловушек для какой-то конкретной команды;

2) "нарушение" (*fault*) - это исключительное состояние, которое возникает во время выполнения команды и оставляет регистры и память в таком состоянии, что при устранении условий для возникновения этой ситуации и рестарте, эта команда будет давать правильные результаты;

3) "прекращение" (*abort*) - это исключительное состояние, также возникающее во время выполнения команды, но, в отличие от нарушений, оставляющее регистры и память в неопределенном состоянии. Поэтому команда не может быть правильно завершена (даже при условии ее рестарта).

В табл. 4 приведены возможные уровни приоритета прерываний процессора и соответствующие им аппаратные события.

I.6. Состояние процессора

При обслуживании особых ситуаций состояние процессора должно быть сохранено для того, чтобы прерванный процесс мог нормально продолжаться. В основном это делается путем сохранения в стеке PC (который указывает либо на команду, которая выполнялась, либо на следующую) и PSL . PC и PSL в дальнейшем восстанавливаются по команде возврата из прерывания или исключительной ситуации (REJ).

Одной из главных характеристик текущего процесса или текущего состояния процессора является длинное слово состояния процессора. Это слово состоит из *processor status word* (PSW) - слова состояния процессора (разряды <0-15> PSL), которое доступно пользователю и слово привилегированного состояния (разряды <16-31> PSL), которое доступно привилегированному программному обеспечению. PSL приводится на рис. 7.

т. н. 0000000000000000	0000000000000000	0000000000000000

Уровни приоритета прерываний

Прерывания	Уровень приоритета прерывания IPL_{16}
Контроль аппаратных средств Недействительность стека ядра Недействительность стека прерываний	IF
Сбой питания	IE
Ошибки процессора, памяти или шины	ID IC IB IA I9
Таймер	I8
Прерывания устройств	I7 I6 I5 I4 I3 I2 II IO
События математического обеспечения	
Резерв	OF OE OD OC

Продолжение табл. 4.

П р е р ы в а н и я	Уровень приоритета прерывания IPL_{16}
Драйверы устройств	08 09 0A 0B
Таймер процесса	07
Очередь <i>AST</i>	06
Резерв	05
Почта ввода/вывода	04
Планировщик процесса	03
Передача <i>AST</i>	02
Резерв	01
Уровень пользовательского процесса	00

PSL

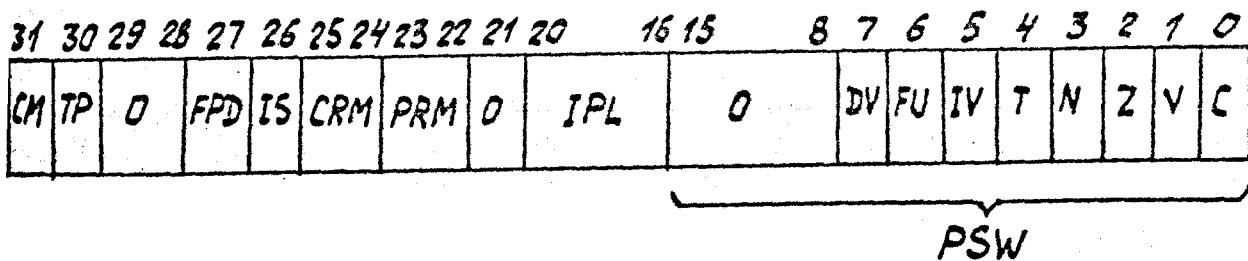


Рис. 7.

Значения полей PSL следующие:

Разряды	Поле	Значение
⟨3-0⟩	<i>N,Z,V,C</i>	Коды условий
⟨4⟩	Т	Бит разрешения трассы
⟨5⟩	IV	Бит разрешения ловушки пелочисленного переполнения
⟨6⟩	FU	Бит разрешения ловушки плавающего недополнения
⟨7⟩	DV	Бит разрешения ловушки десятичного переполнения
⟨15-8⟩	0	Резерв, должны быть нулями
⟨26-16⟩	IPL	Уровень приоритета прерывания
⟨21⟩	0	Резерв, должен быть нуль
⟨23-22⟩	PRM	Режим предыдущего доступа <i>(Previous access mode)</i>
⟨25-24⟩	CRM	Режим текущего доступа <i>(Current access mode)</i>
⟨26⟩	IS	Бит использования стека прерываний <i>(Interrupt stack)</i>
⟨27⟩	FPD	Первая часть выполнена <i>(First part done)</i>

<29-28>	0	Резерв, должны быть нулями
<30>	TP	Запрос трассы
<31>	CM	Признак режима совместимости

Микропрограммная процедура обработки прерываний и исключительных состояний использует в основном поля **IS**, CRM и **IPL**, поэтому рассмотрим их подробнее:

1) **IS**. Если **IS** = 1, то процессор работает со стеком прерываний. При этом необходимо, чтобы CRM = 0 и **IPL** > 0. Если **IS** = 0, то процессор работает со стеком, который определен полем CRM в **PSL**.

2) CRM. Возможны следующие режимы доступа из текущего выполняемого процессса:

- 0 - ядро (*kernel*)
- 1 - исполнитель (*executive*)
- 2 - супервизор (*supervisor*)
- 3 - пользователь (*user*)

3) PRM. Это поле загружается из поля текущего режима при исключительных состояниях и командами CHMX, сбрасывается прерываниями и восстанавливается командой REI.

4) **IPL**. Текущий приоритет процессора меняется в пределах от 0 до 1F. Процессор будет разрешать прерывания только на уровнях больших, чем текущий уровень процессора.

№ документа	Подпись	Дата

1.7. Различия между прерываниями и исключительными состояниями

В целом, по-своему влиянию на работу всей системы, прерывания и исключительные состояния очень похожи. Но существует ряд отличий между ними, которые можно свести к следующим:

- 1) Исключительное состояние (синхронное событие) возникает при выполнении текущей команды, в то время как прерывание (асинхронное событие) возникает из-за чьей-либо активности в вычислительной системе и это действие может быть независимо от текущей команды.
- 2) Исключительное состояние, как правило, обрабатывается в контексте процесса, в котором она возникла, в то время как прерывание обслуживается независимо от текущего выполняющегося процесса.
- 3) IPL процессора обычно не изменяется, когда возникает исключительное состояние. При прерывании IPL всегда повышается.
- 4) Подпрограммы, обслуживающие исключительные состояния, обычно работают со стеком процесса (стеком режима ядра - KSP), а подпрограммы, обслуживающие прерывания, со стеком процессора (стеком прерываний - ISP).
- 5) Разрешенные исключительные состояния всегда инициируются немедленно, независимо от IPL процессора, а прерывания откладываются до тех пор, пока IPL процессора не станет ниже IPL запрашивающего прерывания.
- 6) Большинство исключительных состояний не может быть запрещено. Однако, если причина этого состояния возникает, когда эта ситуация запрещена, то инициация данного исключи-

Лист №005. Учебн. Дата			

тельного состояния не происходит даже после того, когда запрет отменяется. Если же возникает условие прерывания в тот момент, когда оно запрещено или IPL запроса меньше или равно IPL процессора, то данное прерывание в конечном итоге будет удовлетворено, как только возникнет подходящая ситуация, если условие прерывания еще существует.

7) При прерывании поле PRM в PSL всегда устанавливается в режим ядра, а при исключительном состоянии в него переписывается поле CRM .

8) Вследствие асинхронного характера условия прерывания всегда фиксируется аппаратно и переход на соответствующую микропрограмму обработки осуществляется с помощью аппаратных средств. Для исключительного состояния, наряду с таким способом перехода на микропрограмму обработки, возможен и чисто микропрограммный переход. Это связано с синхронностью исключительных состояний — можно заранее определить точки в микропрограммах, из которых возможен переход на микропрограмму обработки соответствующего исключительного состояния.

I.8. Основные параметры прерываний и исключительных состояний

Векторы

Переход на подпрограммы обслуживания особых ситуаций осуществляется с помощью специальных векторов, которые представляют собой длинные слова, интерпретируемые следующим образом:

I) Биты $\langle I:0 \rangle$ содержат интерпретируемый код:

0 — об служить событие через стек ядра. Если работа уже выполняется через стек прерываний, то об служить событие через стек прерываний.

т.н.б.с.ум.	г.с.оп.	дата

I - обслужить событие через стек прерываний. Если это событие является исключительным состоянием, то IPL повышается до FF;

2 - обслужить событие в перезаписываемой управляющей памяти. Если эта память не используется, то процессор переходит на команду останова HALT;

3 - процессор переходит на HALT.

2) Биты <31:2> содержат виртуальный адрес подпрограммы обслуживания - для кодов 0 и I.

Системный управляющий блок (SCB).

Все векторы обслуживания прерываний и исключительных состояний находятся в системном управляющем блоке (SCB - *System Control Block*), который представляет собой две смежных страницы памяти. Физический адрес первой страницы содержится в привилегированном регистре - базе системного управляющего блока SCBB (*SCB Base*).

Векторы устройств на *UNIBUS* находятся во второй странице, адрес которой получается добавлением к адресу, содержащемуся в SCBB, константы $(200)_{16}$. Разрешенными для этих устройств считаются только векторы $(200\text{--}3FC)_{16}$.

Основные параметры прерываний и исключительных состояний приведены в табл.5.

Подгум.	Годп.	Дата

Таблица 5

Таблица основных параметров особых ситуаций

Наименование особых ситуаций	Тип ситуации	Уровень приоритета прерывания IPL_{16}	Вектор 16	Источник (а/МП)*	Код 16	Параметры, загружаемые в стек
1. Машинный сбой	Прекращение/ловушка	IF**	04	Контроллер (а)	4 ⁶	См.Примечание I
2. Недостоверный стек ядра	Прекращение	IF**	08	Процессор (а)	D ⁷	РНК***, PSL
3. Недостоверный стек прерываний	Прекращение	IF**	-	Процессор (а)	I ⁹	РНК, PSL и причина останова остаются на регистрах
4. Сбой питания	Прерывание	IE	0C	Адаптер (а)	5D	PC,PSL
5. Резервная/привилегированная команда	Нарушение	-	10	Процессор (МП)	-	РНК, PSL
6. Резервная команда пользователя	Нарушение	-	I4	Процессор (МП)	-	РНК, PSL
7. Резервный операнд	Нарушение/прекращение	-	I8	Процессор (а/МП)	E ¹¹	РНК, PSL
8. Резервный метод адресации	Нарушение	-	I0C	Процессор (МП)	-	РНК, PSL

Продолжение табл.5

Наименование особых ситуаций	Тип ситуации	Уровень приоритета прерывания IPL ₁₆	Вектор 16	Источник (а/МП)*	Код 16	Параметры, загружаемые в стек
9. Нарушение управления доступом	Нарушение	-	20	Процессор (а)	24	См.Примечание 2
10. Запрет трансляции	Нарушение	-	24	Процессор (а)	35	См.Примечание 2
II. Трассировка	Нарушение	-	28	Процессор (а)	7E	РНК, PSL
12. Команда ВРТ	Нарушение	-	2C	Процессор (МП)	-	РНК, PSL
13. Арифметические ловушки	Ловушка	-	34	Процессор (а/МП)	8-B4	См.Примечание 3
I4. Команды: СНМК СНМЕ СНМС СНМИ	Ловушка	-	40 44 48 4C	Процессор (МП)	62,A	См.Примечание 4
15. Зависание по записи в память	Прерывание	ID	60	Контроллер	C	PC, PSL
16. Исправимые ошибки памяти	Прерывание	IA	54	Контроллер (а)	C	PC,PSL

Продолжение табл.5

Наименование особых ситуаций	Тип ситуации	Уровень приоритета прерывания IPL ₁₆	Вектор I ₁₆	Источник (а/мп) [*]	Код I ₁₆	Параметры, загружаемые в стек
I7. Таймер	Прерывание	I8	CO	Адаптер (а)	5D	PC, PSL
I8. Консольный терминал (приемник)	Прерывание	I4	F8	Адаптер (а)	5D	PC, PSL
I9. Консольный терминал (передатчик)	Прерывание	I4	FC	Адаптер (а)	5D	PC, PSL
20. Программные прерывания	Прерывание	I...F	84...BC	Адаптер (а)	DQ	PC, PSL
21. Прерывания устройств	Прерывание	I4-I7	200-3FC	Адаптер (а)	56 D	PC, PSL

* - аппаратный/микропрограммный характер соответствующего события;

** - для этих событий IPL = IF, если I:0 разряды вектора равны I;

*** - РНК - регистр начала команды. Он указывает на код операции команды, при выполнении которой произошло нарушение или прекращение.

П р и м е ч а н и я к табл.5:

1. При машинном сбое в стеке сохраняются следующие параметры:

- (SP) Параметр длины (равен 28₁₆).
- (SP)+4 Код ошибки (для ошибок памяти, например, равен 2).
- (SP)+8 Регистр виртуального адреса (PKVA).
- (SP)+C PC во время сбоя.
- (SP)+10 Регистр данных памяти (MDR).
- (SP)+14 Сохраненный регистр режима (SMR).
- (SP)+18 Регистр зависания по чтению-модификации-записи (RLTO).
- (SP)+1C Регистр групповой четности буфера трансляции (TBGPR).
- (SP)+20 Регистр ошибок КЭШ-памяти (CAER).
- (SP)+24 Регистр ошибок шины (BER).
- (SP)+28 Регистр суммарных ошибок при машинном сбое (MCESR).
- (SP)+2C PC, указывающий на код операции.
- (SP)+30 PSL.

2. При исключительных состояниях от диспетчера памяти в стеке сохраняются:

- (SP) Регистр суммарных ошибок диспетчера памяти.
- (SP)+4 Виртуальный адрес в странице нарушения (PKVA).
- (SP)+8 РНК.
- (SP)+C PSL.

3. При арифметических ловушках в стеке, кроме PC, также сохраняется код арифметической ловушки, который имеет следующие значения:

Код	Тип арифметической ловушки
1	Целочисленное переполнение
2	Целочисленное деление на 0
3	Плавающее переполнение
4	Плавающее/десятичное деление на 0
5	Плавающее обнуление
6	Десятичное переполнение
7	Диапазон приписки

4. При выполнении команд смены режима в стек, кроме РС, PSL загружается операнд команды (*code*) с расширенным знаком.

Вся аппаратура, связанная с обработкой прерываний и исключительных состояний, располагается в контроллере памяти, адаптере магистрали и в процессоре.

Контроллер фиксирует аппаратные ошибки, связанные с работой памяти и сообщает об этом процессору с помощью сигналов INRC (прерывание контроллера) и MCS (машинный сбой). Адаптер фиксирует все остальные прерывания, проводит разбор приоритетов между ними и сообщает процессору о наличии запроса прерывания сигналом INRA (прерывание адаптера). Кроме этого в адаптере находится часть внутренних регистров системы, которые связаны с обработкой прерываний.

В процессоре находится, собственно блок прерываний (БПР), который наряду с прерываниями контроллера и адаптера, обрабатывает все аппаратные исключительные состояния, проводит окончательный разбор приоритетов и формирует код особой ситуации, с помощью которого осуществляется формирование начального адреса соответствующей микропрограммы обработки.

I.9. Блок прерываний БПР

I.9.1. На рис.8 приведена структурная схема БПР процессора. Ниже перечислены основные блоки, входы и выходы БПР.

- 1) БСБПР - блок синхронизации БПР.
- 2) БРП - блок разбора приоритетов.
- 3) ДШИРП - дешифратор сброса источника прерываний и исключительных состояний.
- 4) РИП - регистр источников прерываний и исключительных состояний.
- 5) Входы:

Из.480.319 ТО

Цв.Лист №10001 Учебн. Часы

Лист

43

Структурная схема БПР

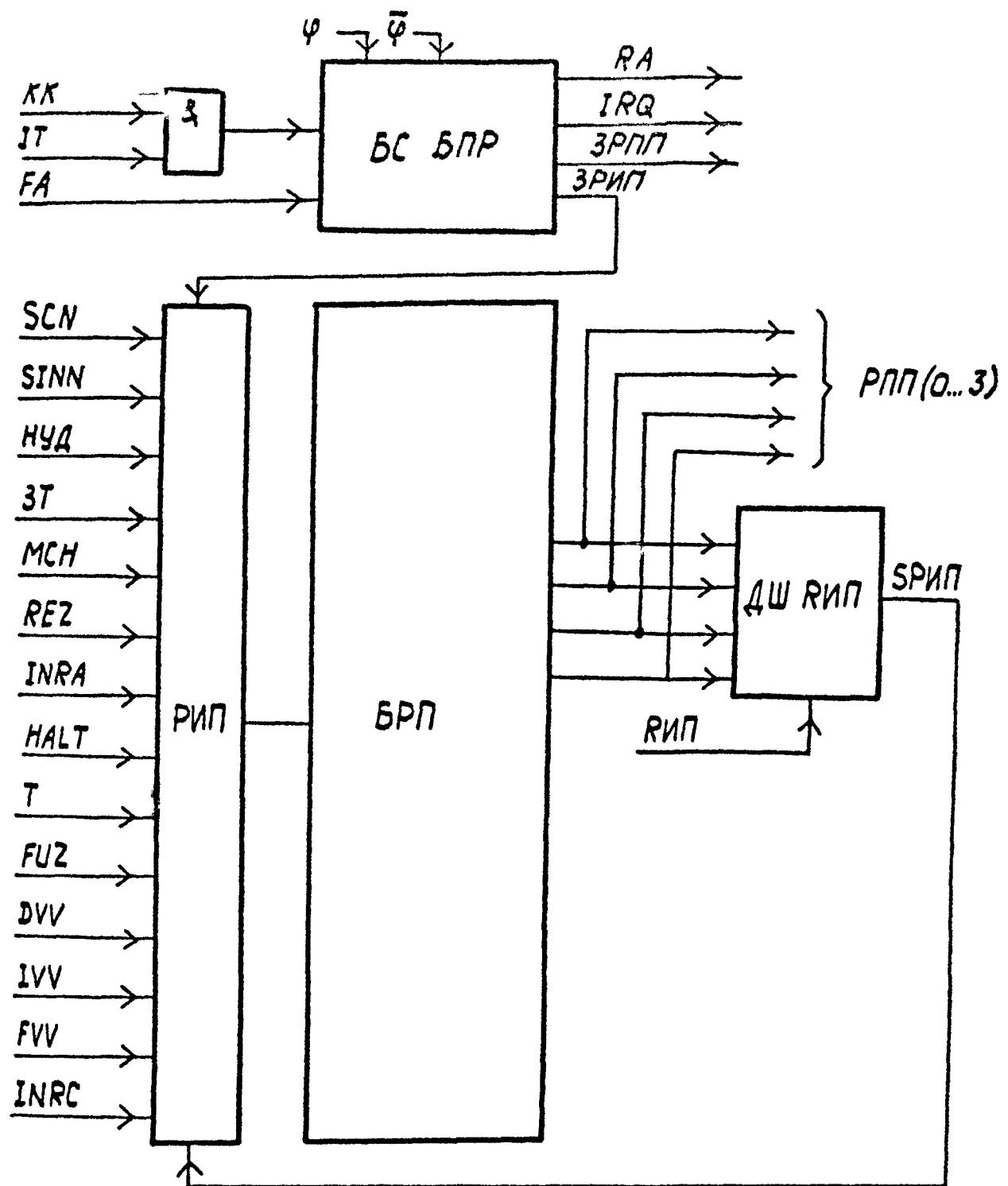


Рис. 8.

- | | |
|------|---|
| KK | - признак конца команды; |
| FA | - выход элемента ИЛИ, на входах которого собраны все нарушения и прекращения; |
| IT | - выход элемента ИЛИ, на входах которого собраны все прерывания и ловушки; |
| SCN | - запрос недостоверного стека ядра; |
| STNN | - запрос недостоверного стека прерываний; |
| НУД | - запрос нарушения управления доступом; |
| ЗТ | - запрос запрета трансляции; |
| MCH | - запрос машинного сбоя; |
| REZ | - запрос резервного операнда; |
| INRA | - запрос адаптера; |
| HALT | - запрос останова; |
| T | - запрос трассировки; |
| FUZ | - запрос ловушки плавающего недополнения; |
| DVV | - запрос ловушки десятичного переполнения; |
| IVV | - запрос ловушки целочисленного переполнения; |
| PVV | - запрос ловушки плавающего переполнения; |
| INRC | - запрос прерывания контроллера; |
| RMI | - сигнал сброса источника прерывания. |

6) Выходы:

- | | |
|----------|---|
| RA | - сигнал запроса сбоя; |
| IRQ | - сигнал запроса прерывания; |
| ЗРПИ | - сигнал записи регистра признака прерывания и исключительного состояния (РПИ); |
| ЗРИП | - сигнал записи РИП; |
| РПИ(0-3) | - код прерываний и исключительных состояний; |
| SPRIP | - сигнал сброса конкретного источника (в соответ- |

ствии с кодом).

1.9.2. Алгоритм работы БПР

При поступлении в БПР сигналов прерываний – ловушек или сигналов нарушений-прекращений запускается БСБПР – по сигналу IT или FA соответственно. FA может возникать в любой момент выполнения команды, IT возникает лишь после окончания команды, перед выполнением следующей команды. БСБПР вырабатывает сигналы RA или IRQ, которые поступают на ПМ ДИК, ЗРИП, по которому производится запись в РИП всех имеющихся к этому времени запросов. В БПР, в соответствии с самым высокоприоритетным запросом, формируется код, который поступает на входы РИП (находящегося в блоке управления ПЗУ) и на входы ДП КИП для последующего сброса источника запроса по сигналу RIP.

Запись в РИП производится сигналом ЗРИП, также выработанным БСБПА.

Содержимое РИП затем используется микрокомандой перехода по регистру для выхода на конкретные микропрограммы обработки прерываний и исключительных состояний.

В табл. 5 в графе "Код₁₆" приведены шестнадцатеричные коды аппаратных особых ситуаций.

Приоритетность между прерываниями и ловушками в БПР принята следующая (в порядке убывания приоритетов):

- 1) Арифметические ловушки.
- 2) Прерывания контроллера.
- 3) Прерывания адаптера.
- 4) Ловушка трассы.

На рис. 9 приведена временная диаграмма работы БПР.

Чертёжный лист №	Номенклатурный №	Год	Дат
------------------	------------------	-----	-----

Временная диаграмма работы БПР

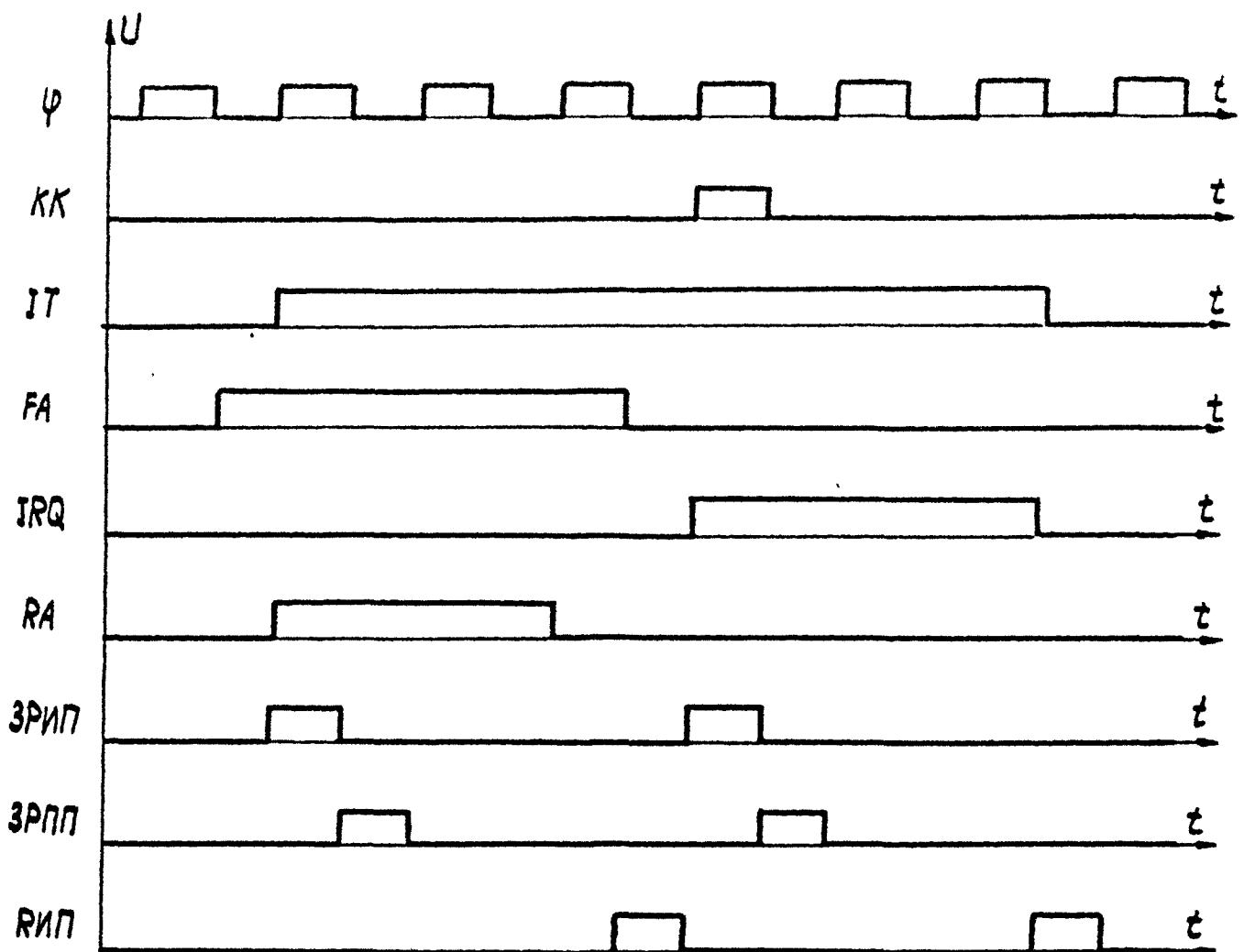
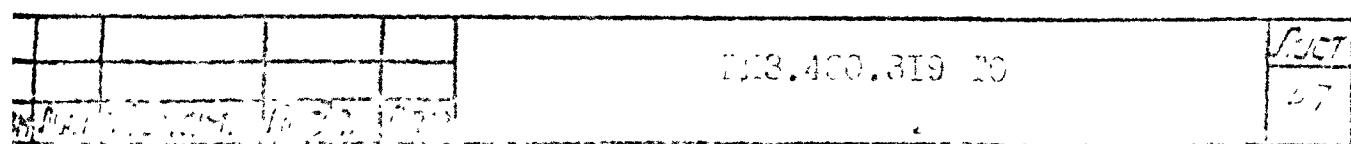


Рис. 9.



I.9.3. Алгоритм обработки прерываний и исключительных состояний

Микропрограммная процедура обработки ситуаций сводится, в основном, к двум действиям:

- 1) сохранению текущего состояния (РС(РНК), PSL , параметры (если нужно));
- 2) формированию нового состояния (загрузка РС начальным адресом программы обработки особой ситуации, формирование нового PSL).

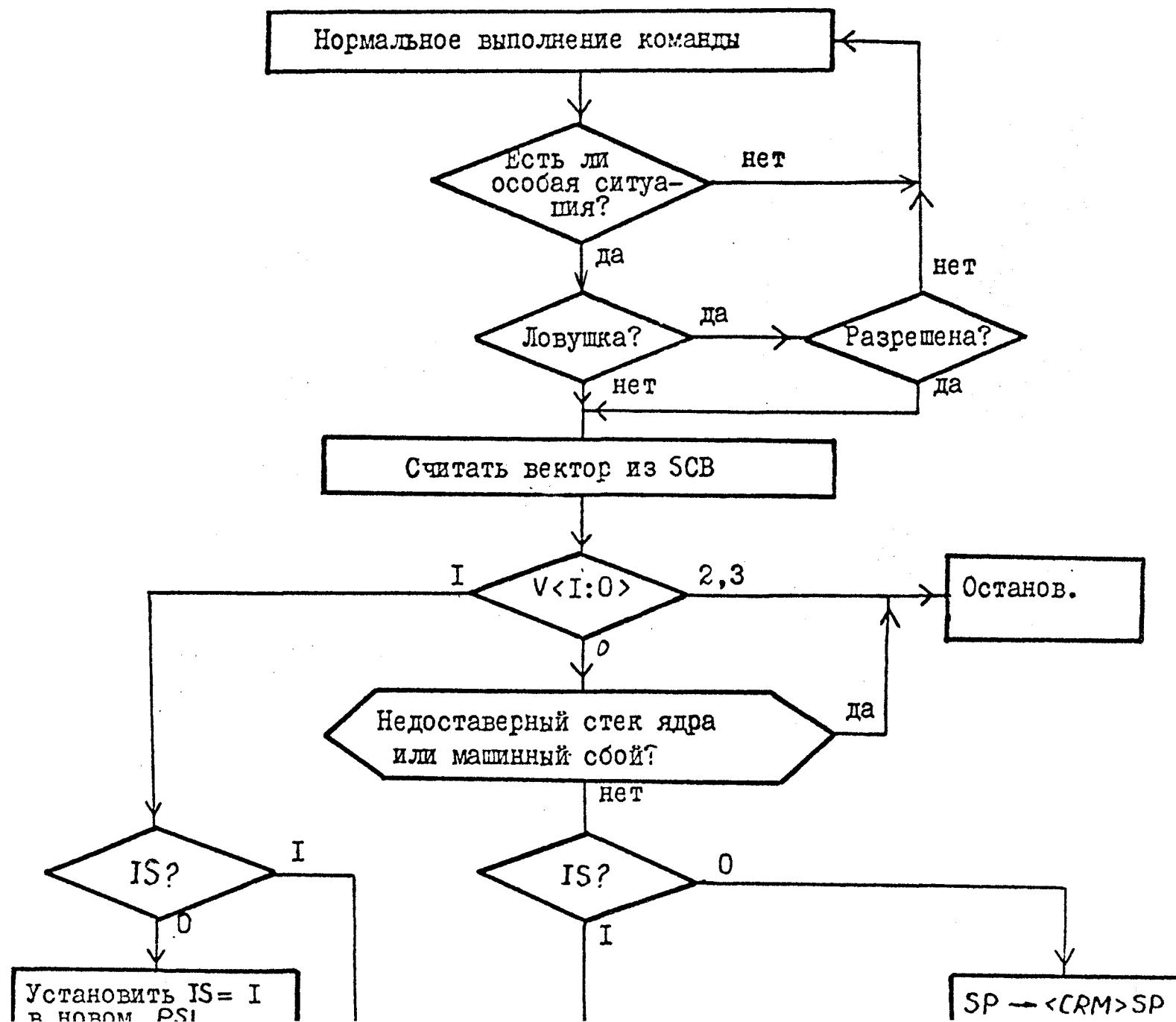
Более подробно эти два действия, и другие описаны в алгоритме обработки прерываний и исключительных состояний на рис.10.

I.10. Временные диаграммы процессора, идентификаторы обменов

I.10.1. Ко всем временным диаграммам относится следующее:

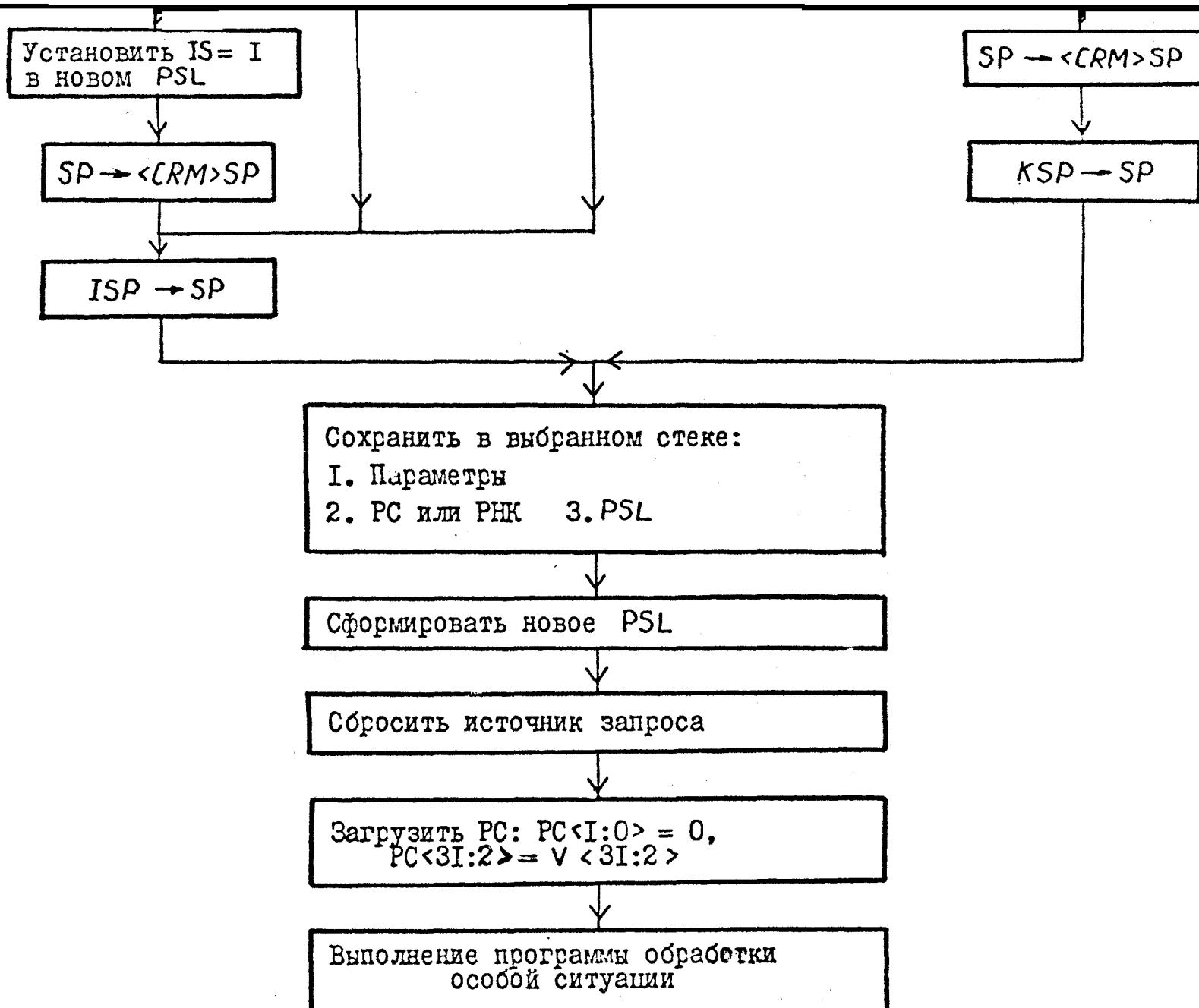
I) Если на входе тестирования $TEST$ подан низкий уровень, то новый обмен по микрокомандной магистрали не начнется, пока ответный сигнал $MSDS$ от накопителя микрокоманд не закончится. По магистрали $AD(0-31)$ не начнется новый обмен, пока не закончатся ответные сигналы SAS и SDS от предыдущего обмена. Если на входе $TEST$ высокий уровень, то новые

Алгоритм обработки прерываний и исключительных состояний



ИЗМ РСТ
№ 5524М РДЭМ
1970

ИМ 480.319 ТО



- $V <...>$ - разряды вектора обработки прерываний и исключительных состояний;
 $<CRM>SP$ - указатель стека текущего режима (содержимое одного из четырех регистров : KSP, ESP, SSP или USP);
 ISP - содержимое регистра указателя стека прерываний.

Рис.10.

обмены по магистралям начинаются независимо от окончания ответных сигналов. При этом окончание сигналов SAS и SDS должно происходить асинхронно с минимальным временем задержки по окончании сигналов MAS и MDS. Представленные ниже временные диаграммы справедливы для высокого уровня на входе TEST.

2) Интерфейсные сигналы при магистрали AD(0-3I):

MAS, PGB, PCRQ, MDS, SEL, DMAK вырабатываются выходами с тремя состояниями. На эти выходы необходимо ставить резисторы 2,7 кОм к шине источника питания + Ucc. После окончания сигналов в процедурах обменов на этих выходах осуществляется "подброс в "1" на время $0,5 T_{CIC}$ ", причем на выходах MAS, PCRQ, DMAK в отрицательной фазе CIC, а на выходах PGB, MDS и SEL - в положительной. Также в состояние активного высокого уровня эти выходы приводятся от низкого уровня на входе SR при начальном пуске ЦПР, причем для активного высокого уровня на выходе MAS необходимо наличие высокого уровня на входе ACTR.

3) Выходы AD(0-3I) также имеют три состояния. В третьем состоянии выходы AD(0-3I) находятся при приеме команд или данных, от низкого уровня на входе SR, в процедуре "Запись" с перехватом" и при предоставлении прямого доступа к памяти.

В состояние активного высокого уровня выходы AD(0-3I) приводятся между обменами по окончании сигнала MDS.

Временная диаграмма процедуры "Чтение микрокоманды" представлена на рис. 11.

Минимальный цикл приема микрокоманды составляет $2 T_{CIC}$. Ответный сигнал MSDS от накопителя микрокоманд может быть также статическим низким уровнем в отличие от предоставленного на рис. 11.

Временная диаграмма процедуры "Чтение микрокоманды"

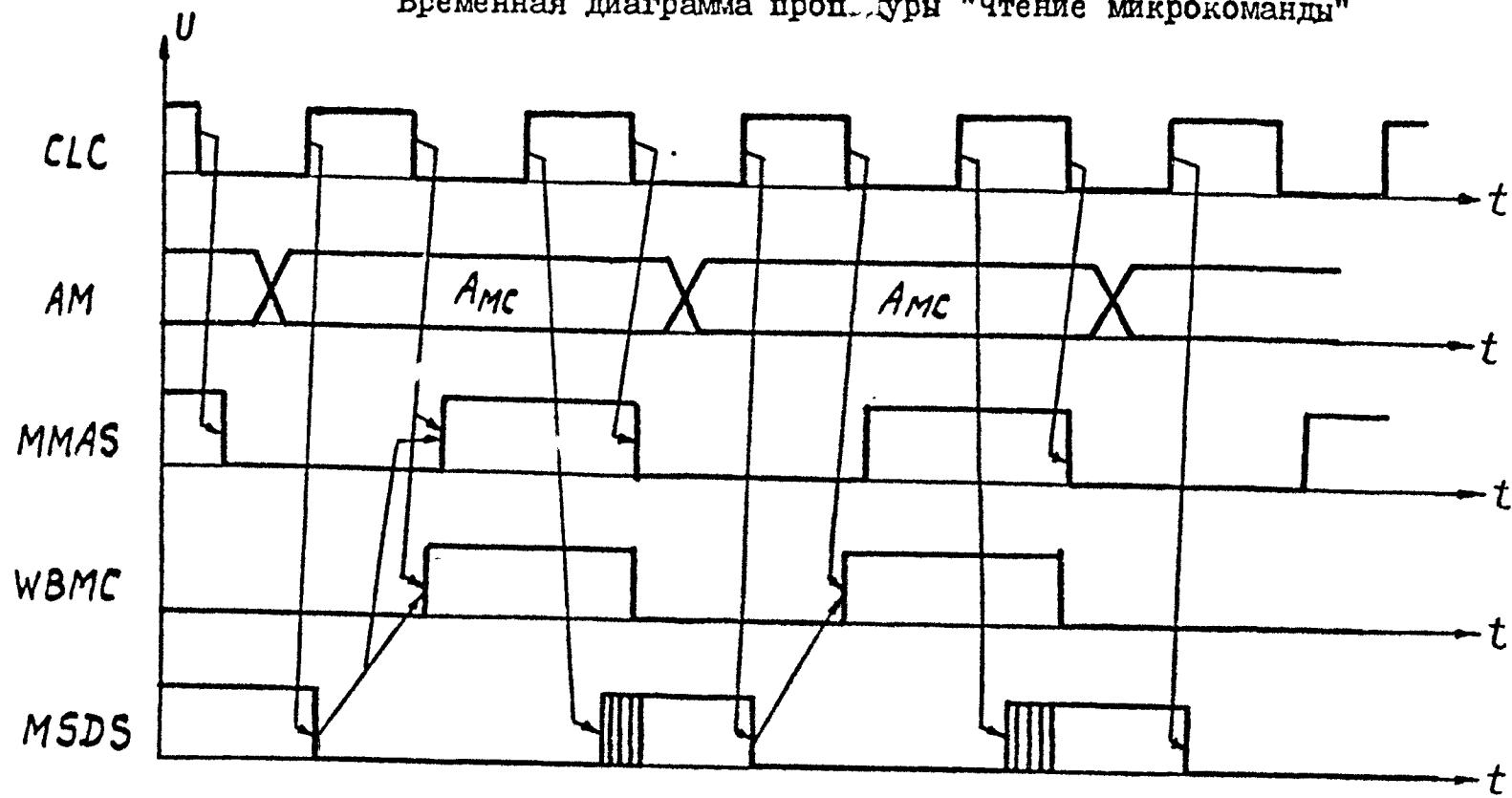


Рис. 11.

На временной диаграмме представлен дополнительно внутренний сигнал ЦПР $WBMС$ – запись в буфер микрокоманд.

Временная диаграмма процедуры "Адресное чтение" представлена на рис. 12. Минимальный цикл адресного чтения $3T_{CLC}$. Низкий уровень сигнала SAS принимается в отрицательной фазе CLC , затем выдается сигнал MDS низкого уровня и открывается внутренняя магистраль $MD(0-3I)$ для приема данных. В целях обеспечения быстродействия на частоте 10 МГц низкий уровень сигнала SAS необходимо подавать в течение первых 25 нс отрицательной фазы CLC . Низкий уровень сигнала SDS принимается также в отрицательной фазе CLC , и в следующей положительной фазе CLC заканчивается сигнал MDS . Для обеспечения быстродействия необходимо выполнить такие же требования, как и для сигнала SAS .

Временная диаграмма "Безадресное чтение команды" представлена на рис. 13. Минимальный цикл процедуры $2 T_{CLC}$. Прием сигнала SDS происходит также, как и при адресном чтении и требования для обеспечения быстродействия те же.

Временная диаграмма процедуры "Запись" представлена на рис. 14. Минимальный цикл процедуры – $4T_{CLC}$. Прием сигналов SAS и SDS и требования по обеспечению быстродействия такие же как и при адресном чтении.

Временная диаграмма процедуры "Чтение-модификация-запись" представлена на рис. 15. Минимальный цикл процедуры – $7 T_{CLC}$. Прием сигналов SAS и SDS и требования по обеспечению быстродействия такие как и при адресном чтении.

Временная диаграмма процедуры передачи адреса при пересечении данными границы страницы представлена на рис. 16.

Сигнал PGB при первом появлении сопровождает адрес перед

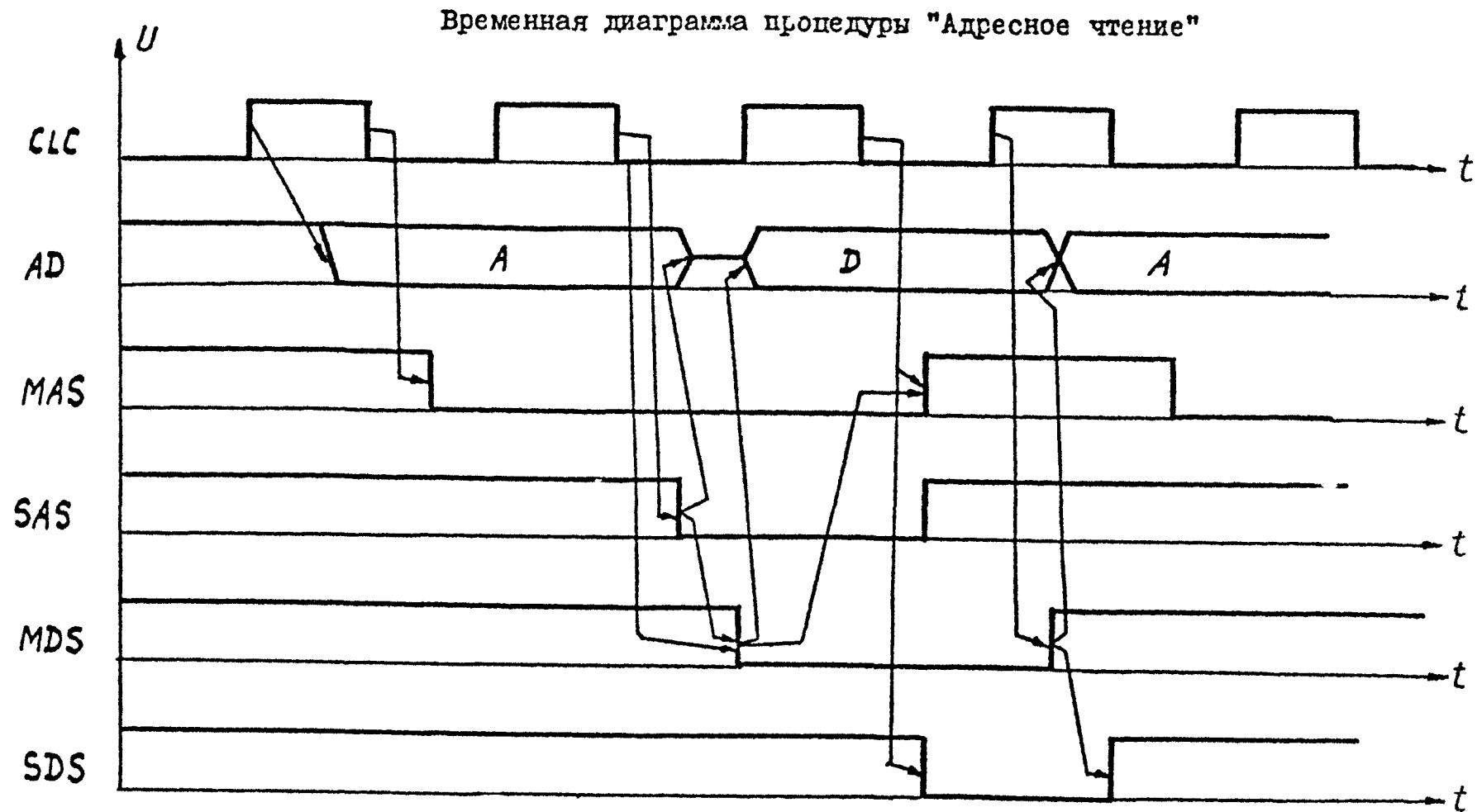


Рис. 12,

Временная диаграмма "Безадресное чтение команды"

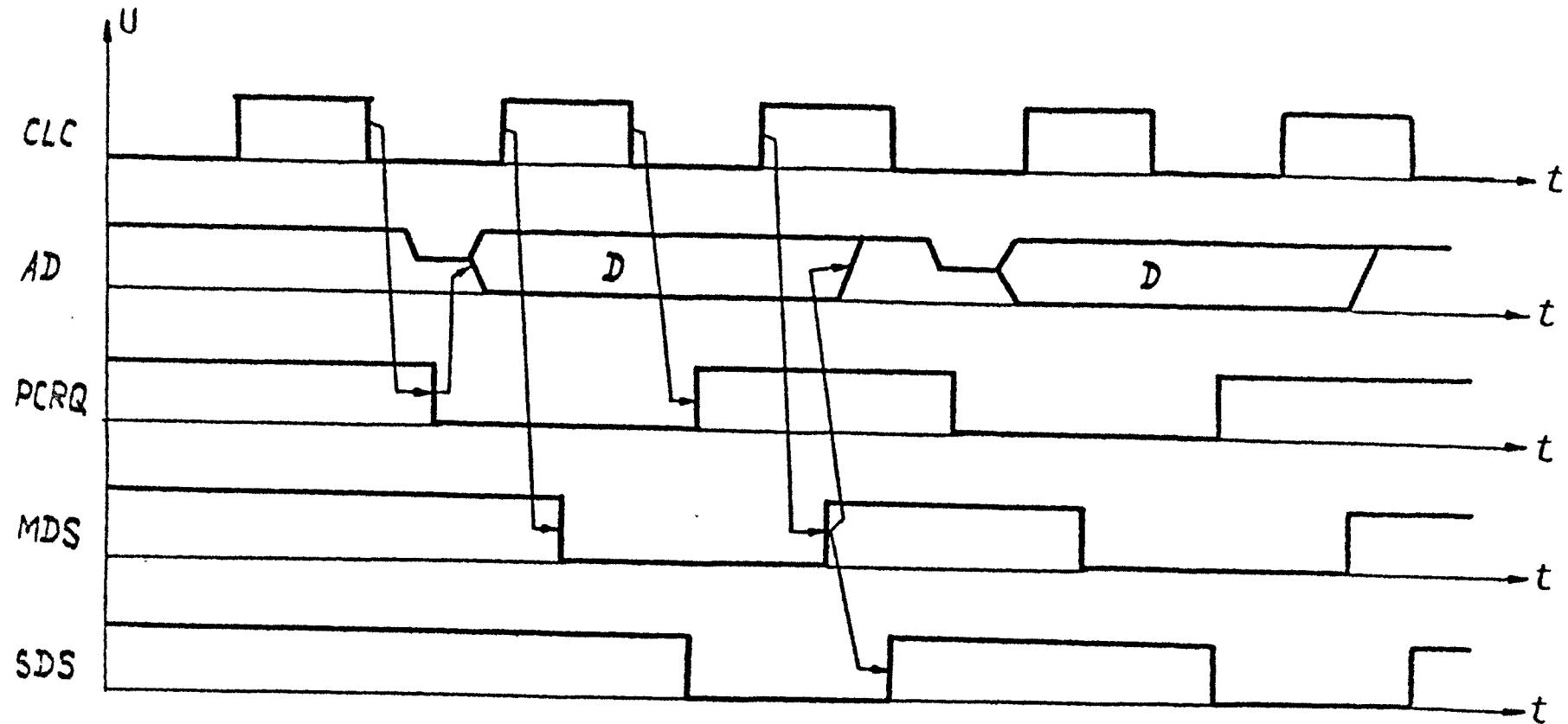


Рис. 13,

Временная диаграмма процедуры "Запись"

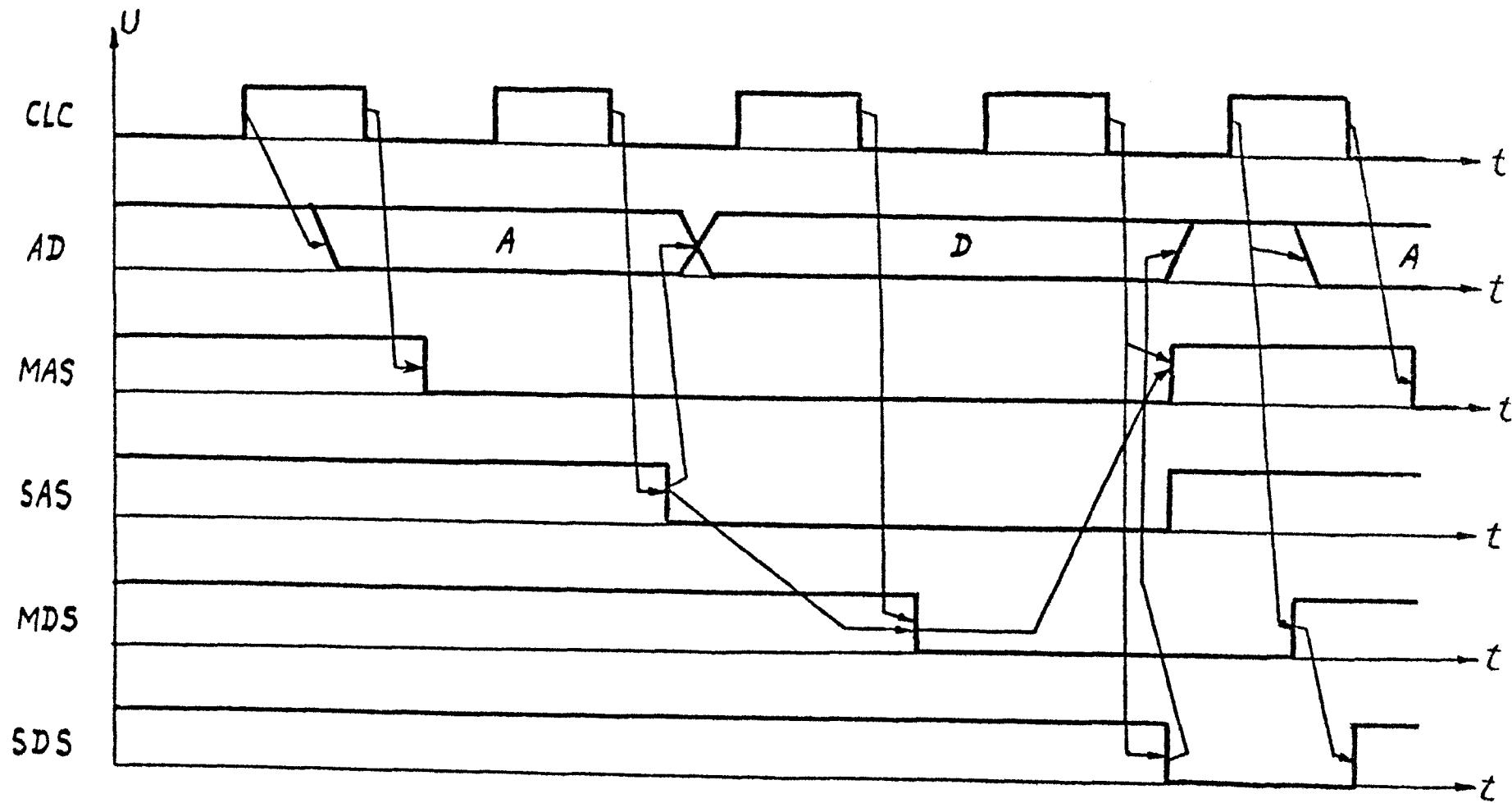


Рис. 14,

Временная диаграмма процедуры "Чтение-модификация-запись"

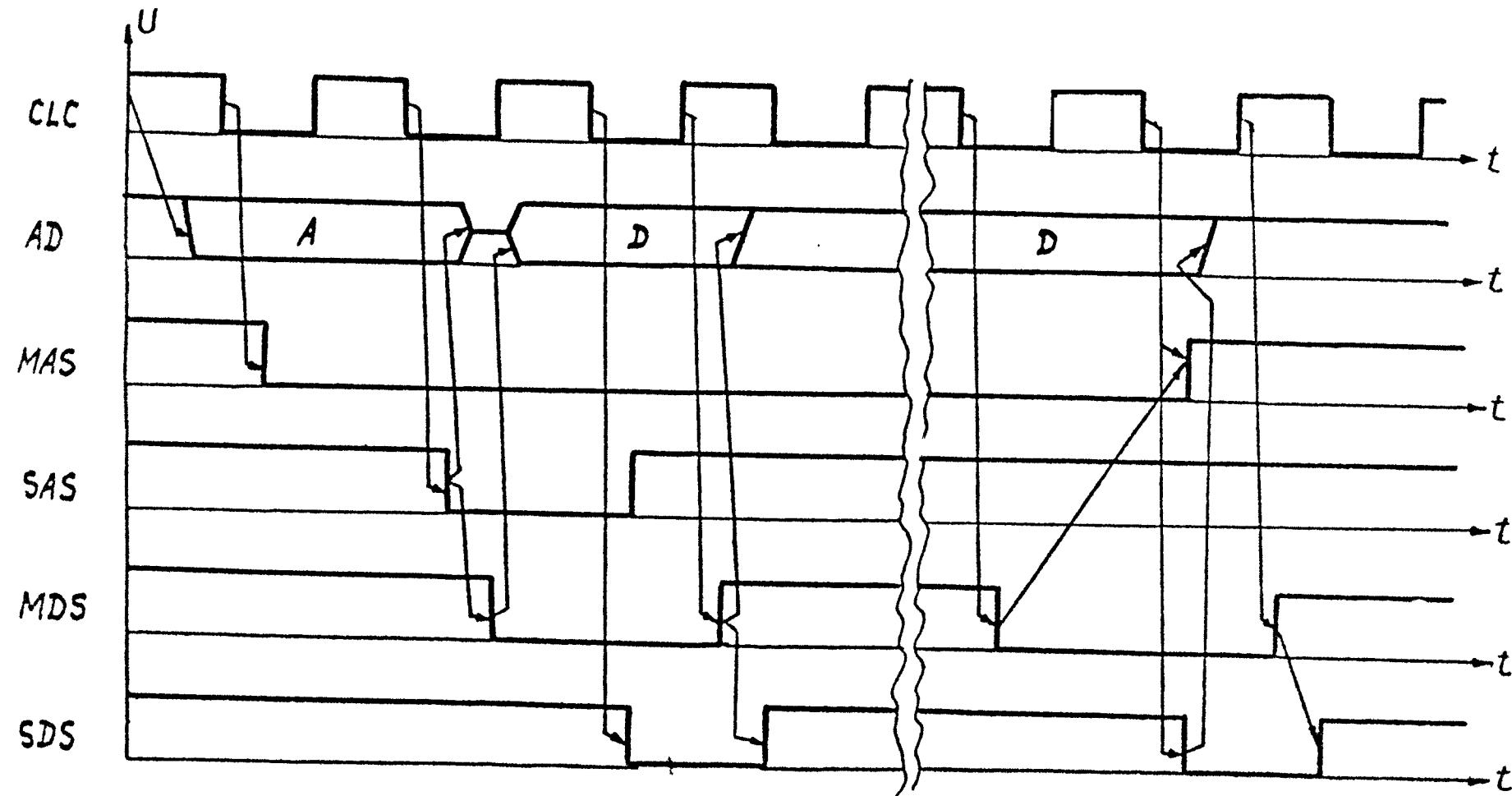


Рис. 15.

Временная диаграмма процедуры передачи адреса при пересечении
данными границы страницы

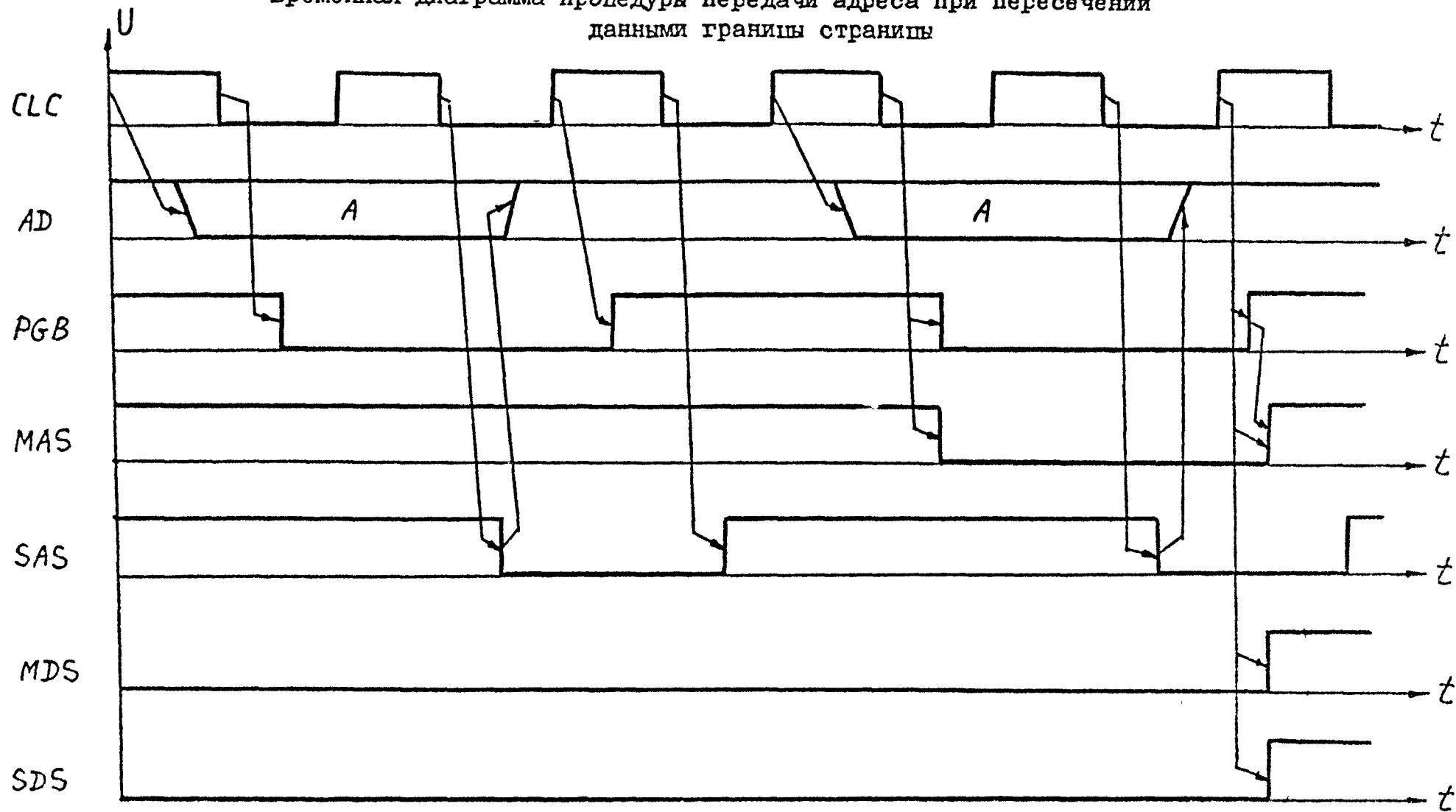


Рис. I6.

границей страницы, при втором появлении сопровождает адрес на -
чала новой страницы. Между двумя адресами (рис. 16) возможно
выполнение обменов, например, "Чтение РТЕ", если пуст "Буфер
трансляции", или "Безадресное чтение команды". Для обеспечения бы-
стродействия на частоте 10 МГц требования к сигналам \overline{SAS} и \overline{SDS}
такие же, как и при адресном чтении.

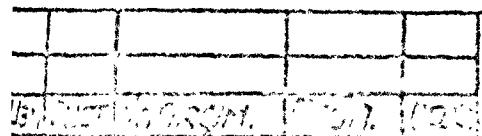
Временная диаграмма отказа от записи в процедуре "Чтение-
модификация-запись РТЕ" представлена на рис. 17. Отказ от запи-
си РТЕ происходит, когда страница уже модифицировалась (бит
модификации M < 26 > регистра РТЕ установлен в "1").

Временные диаграммы процедур чтения и записи внутренних ре-
гистров IPR , размещенных в процессоре (P0BR , P0LR , P1BR ,
P1LR , SBR , SLR , MAPEN , TBIA , TBIS , ACCS)
представлены на рис. 18,19. Внешние сигналы \overline{SAS} , \overline{SDS}
не требуются, вместо них в процессоре вырабатываются внутренние
сигналы такого же назначения.

Временная диаграмма чтения внешнего регистра начальных ус-
ловий представлена на рис. 20 . Для чтения этого регистра пред-
назначен выход SEL . Сигнал SEL возникает при чтении
системной памяти (A3I, A30 = "1") и при A23 = "1". На временной
диаграмме сигнал SEL аналогичен сигналу \overline{MDS} .

Временная диаграмма сигналов сопровождения процедуры захва-
та магистрали при прямом доступе к памяти представлена на рис. 21.

Сигнал DMR магистрали Q-BUS преобразуется микросхемой
адаптера магистрали КЛ1839ВВ1 в сигнал \overline{DMRQ} , поступаю-
щий на процессор. При отсутствии обменов процессор принимает сиг-
нал \overline{DMRQ} в низкой фазе CLC и через период CLC выдает
сигнал разрешения прямого доступа к памяти DMAK , который
микросхема АМ преобразует в сигнал \overline{DMGO} магистрали Q-BUS.
Окончание сигнала DMAK происходит в той же фазе низкого уров-



Временная диаграмма отказа от записи в процедуре "Чтение-модификация-запись РТЕ"

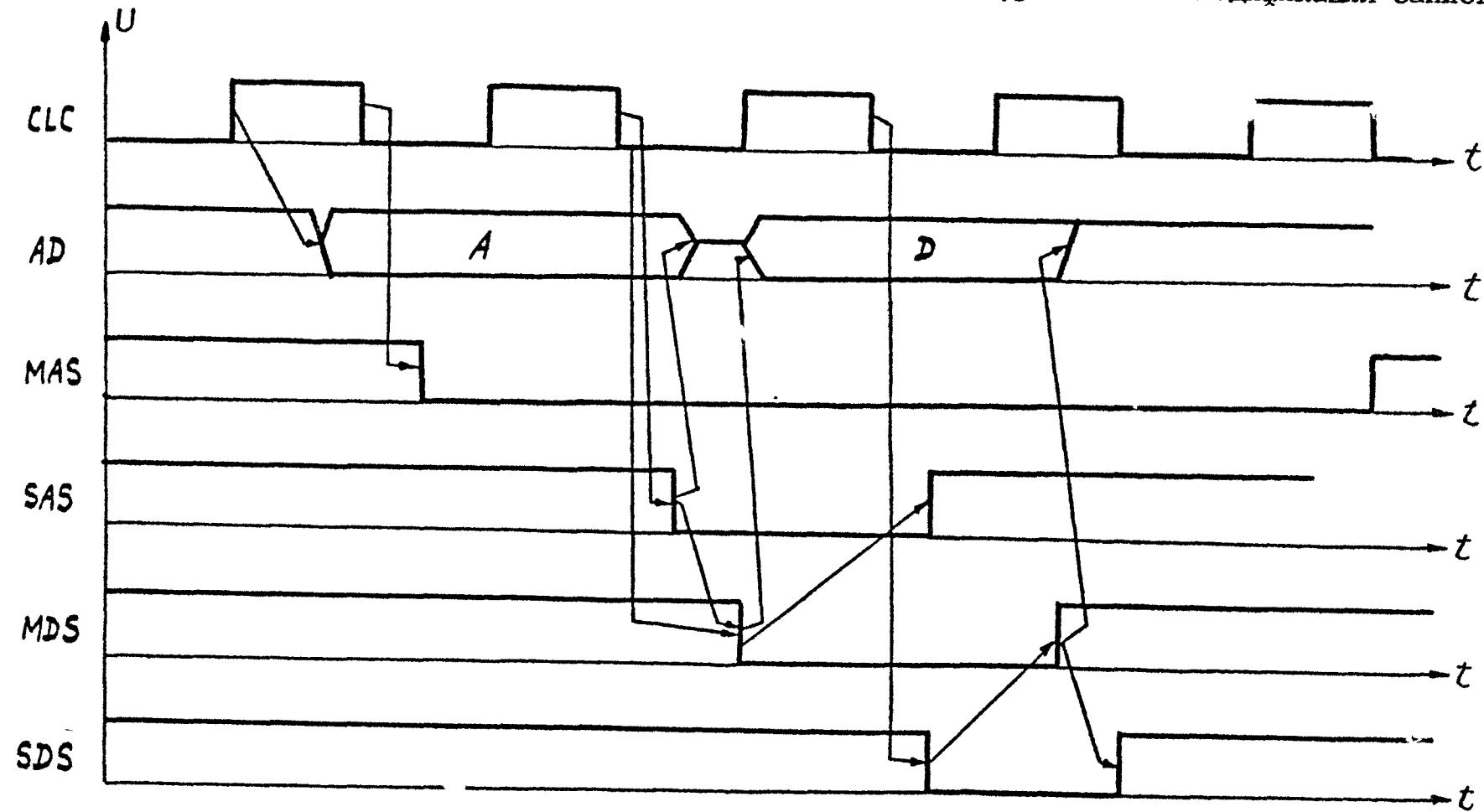


Рис. 17,

Временная диаграмма процедуры чтения внутренних регистров IPR,
размещенных в процессоре

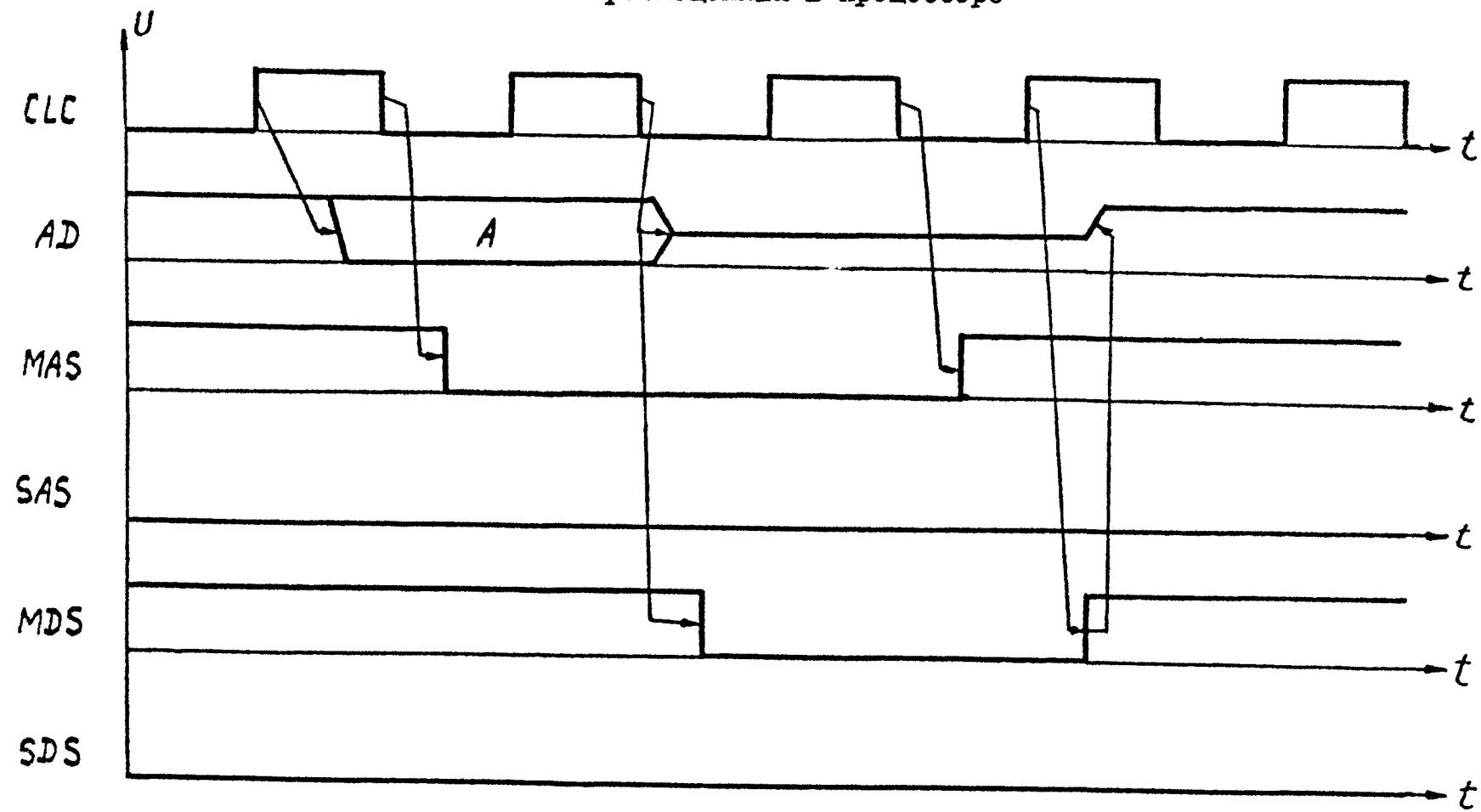


Рис. 18,

Временная диаграмма процедуры записи внутренних регистров IPR,
размещенных в процессоре

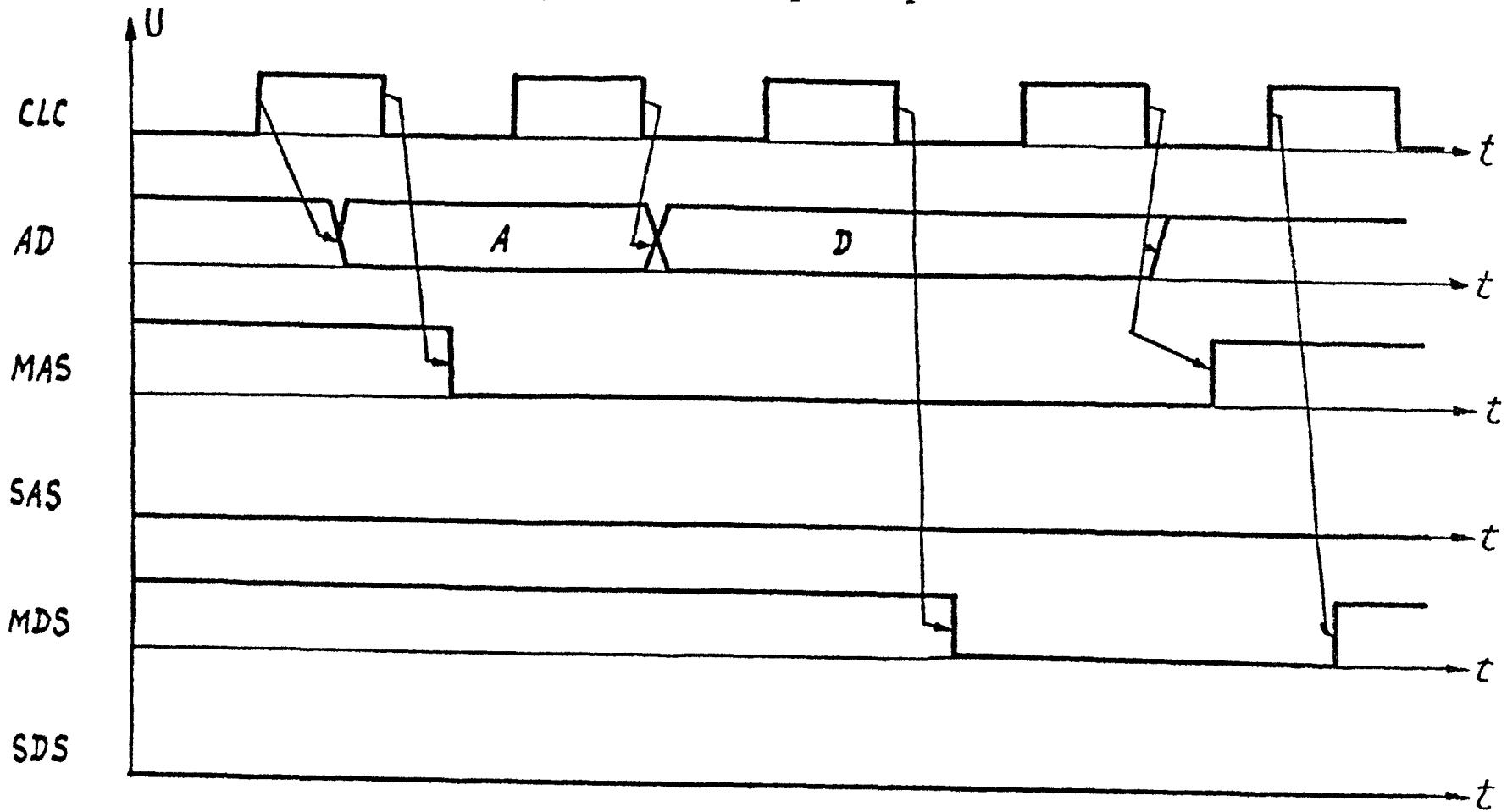


Рис. 19,

STRUCT
ROUTINE
MAIN
MAIN
MAIN

19.460.3TS TO

STRUCT
61

Временная диаграмма чтения внешнего регистра начальных условий

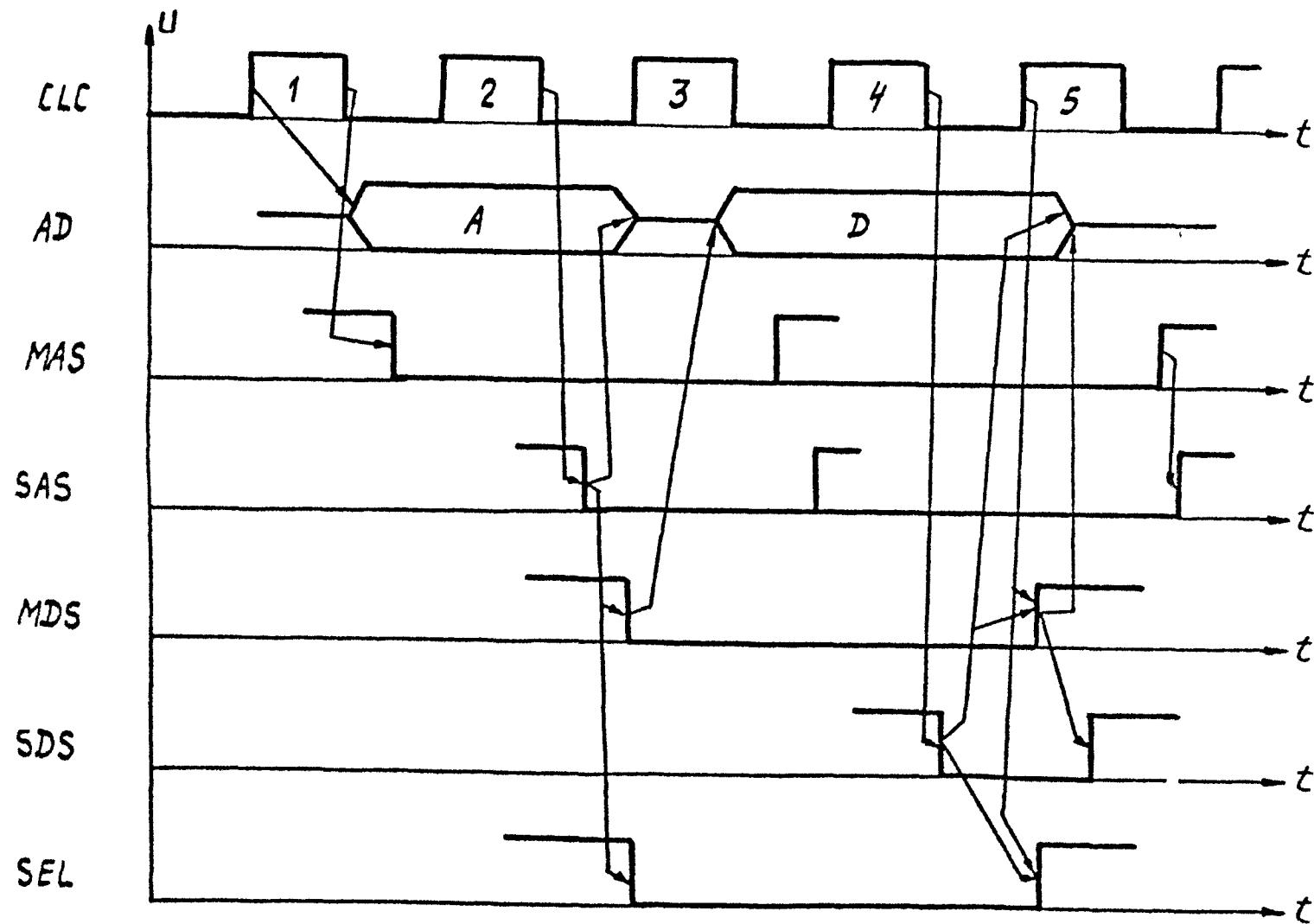
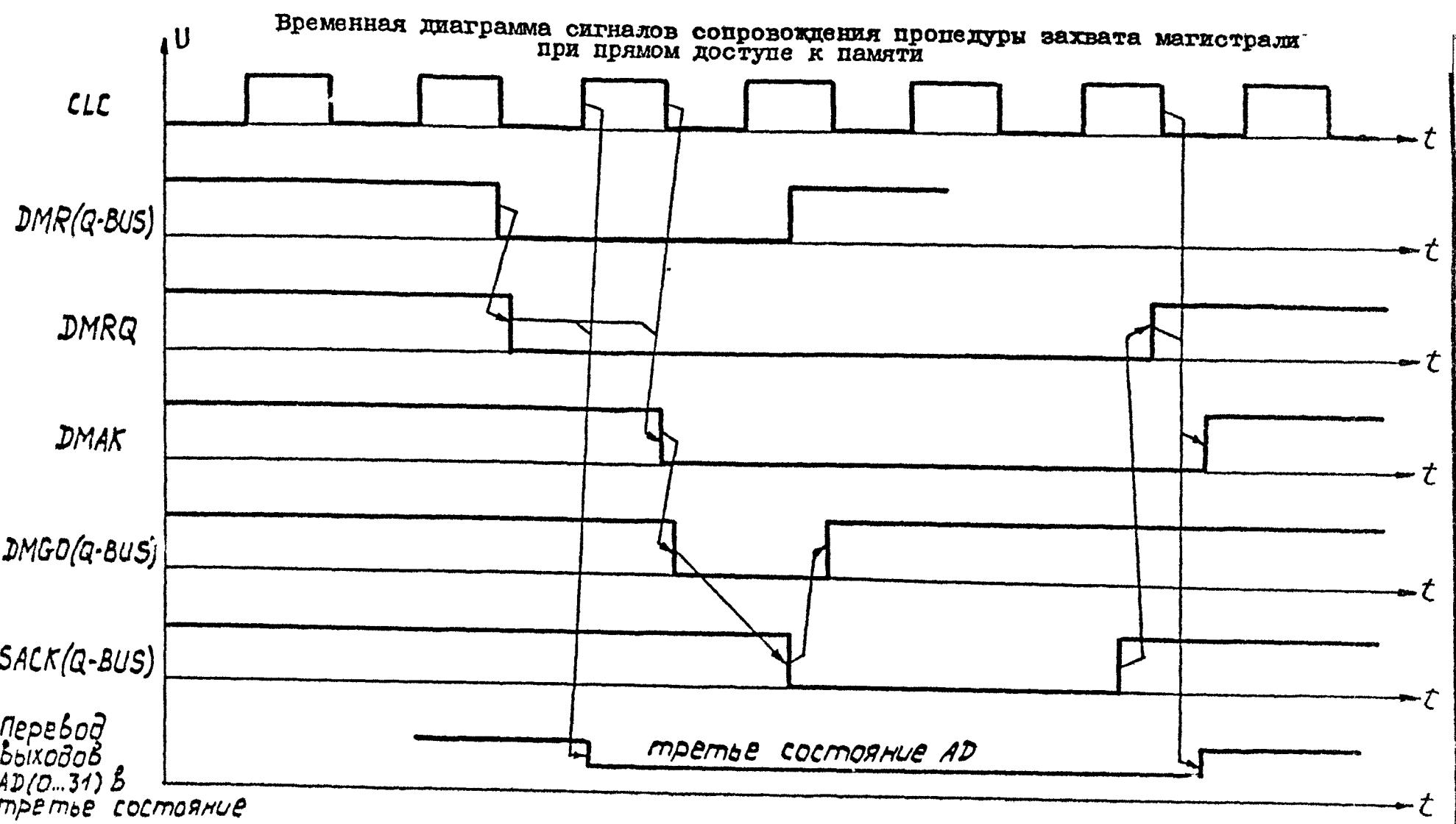


Рис. 20,



Со стороны АМ прием и выдача всех сигналов прямого доступа асинхронны

Рис. 21.

ия CLC в какой принимается окончание сигнала DMRQ. На временной диаграмме также представлен перевод выходов матрицы AD(0-3I) в третье состояние и выход из третьего состояния. При наличии обмена сигнал DMAK появится только по его окончании.

Временные диаграммы адресного чтения и адресной записи с перехватом сопроцессором обменов с памятью представлены на рис. 22, 23. Обмен с перехватом происходит при A3I = "0", A30 = "1". О том, что микросхема сопроцессора начинает обмен с памятью, свидетельствует сигнал ACTR. Приняв асинхронно этот сигнал, процессор переводит выход MAS в третье состояние, предоставляя возможность сопроцессору выставить активный низкий уровень MAS_{СПРЦ} и далее сопроцессор читает данные из памяти или записывает данные в память также, как выполняет эти процедуры процессор. При записи с перехватом выходы AD(0-3I) процессора переводятся в третье состояние по окончании адреса в низкой фазе CLC и выходят из третьего состояния (подброс в "1") по окончании низкого уровня нашине MDS.

Временная диаграмма процедуры "Чтение-модификация-запись с перехватом" представлена на рис. 24.

Временная диаграмма процедуры "Двойное адресное чтение" представлена на рис. 25.

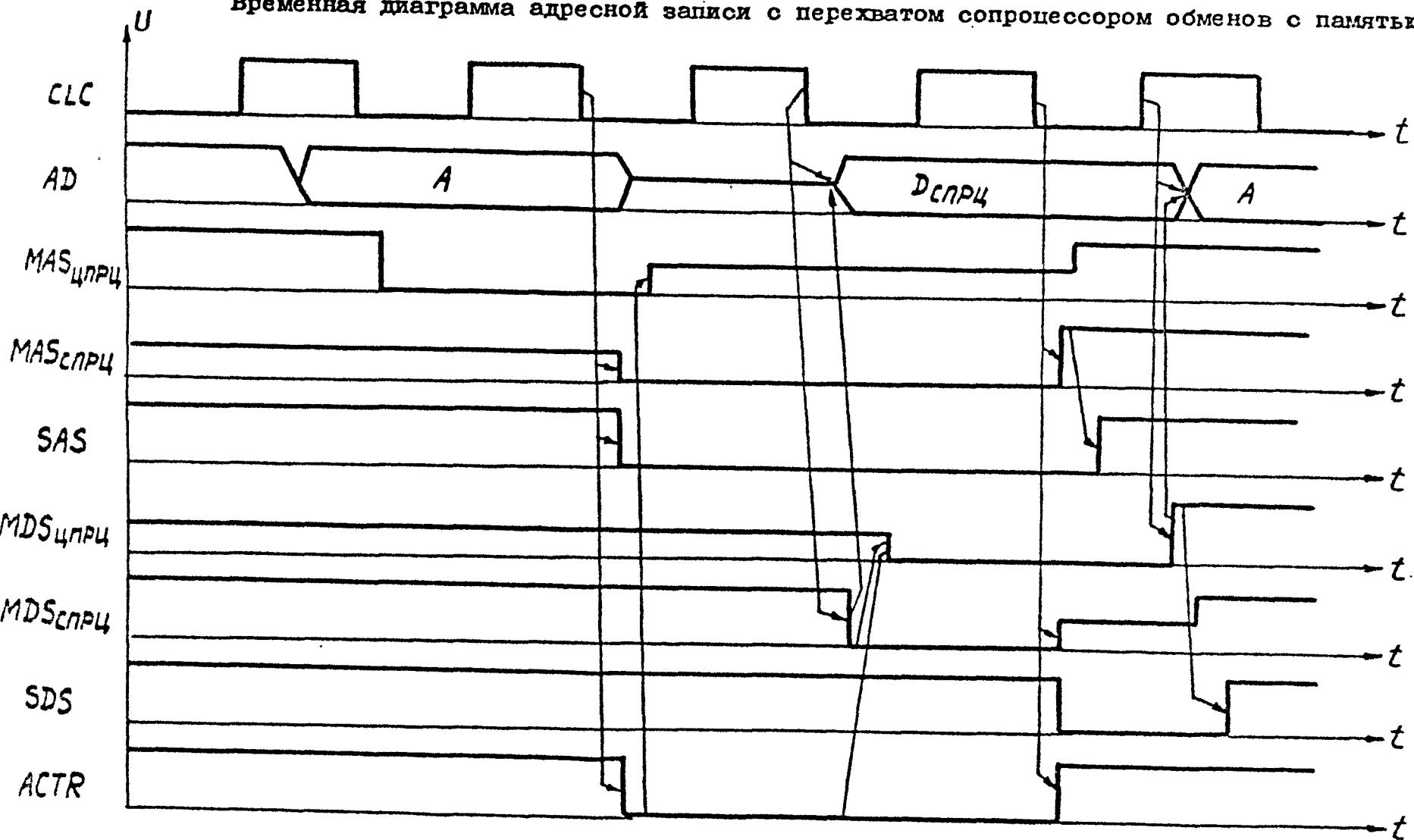
Временная диаграмма процедуры "Двойная запись" представлена на рис. 26.

Двойное чтение и запись выполняются при обменах с четырьмя словами.

На рис. 27-30 представлены временные диаграммы различных обменов с внешними и внутренними сигналами процессора.



Временная диаграмма адресной записи с перехватом сопроцессором обменов с памятью



ЦПРЦ – центральный процессор

СПРЦ – сопроцессор

Рис. 23,

Временная диаграмма про-

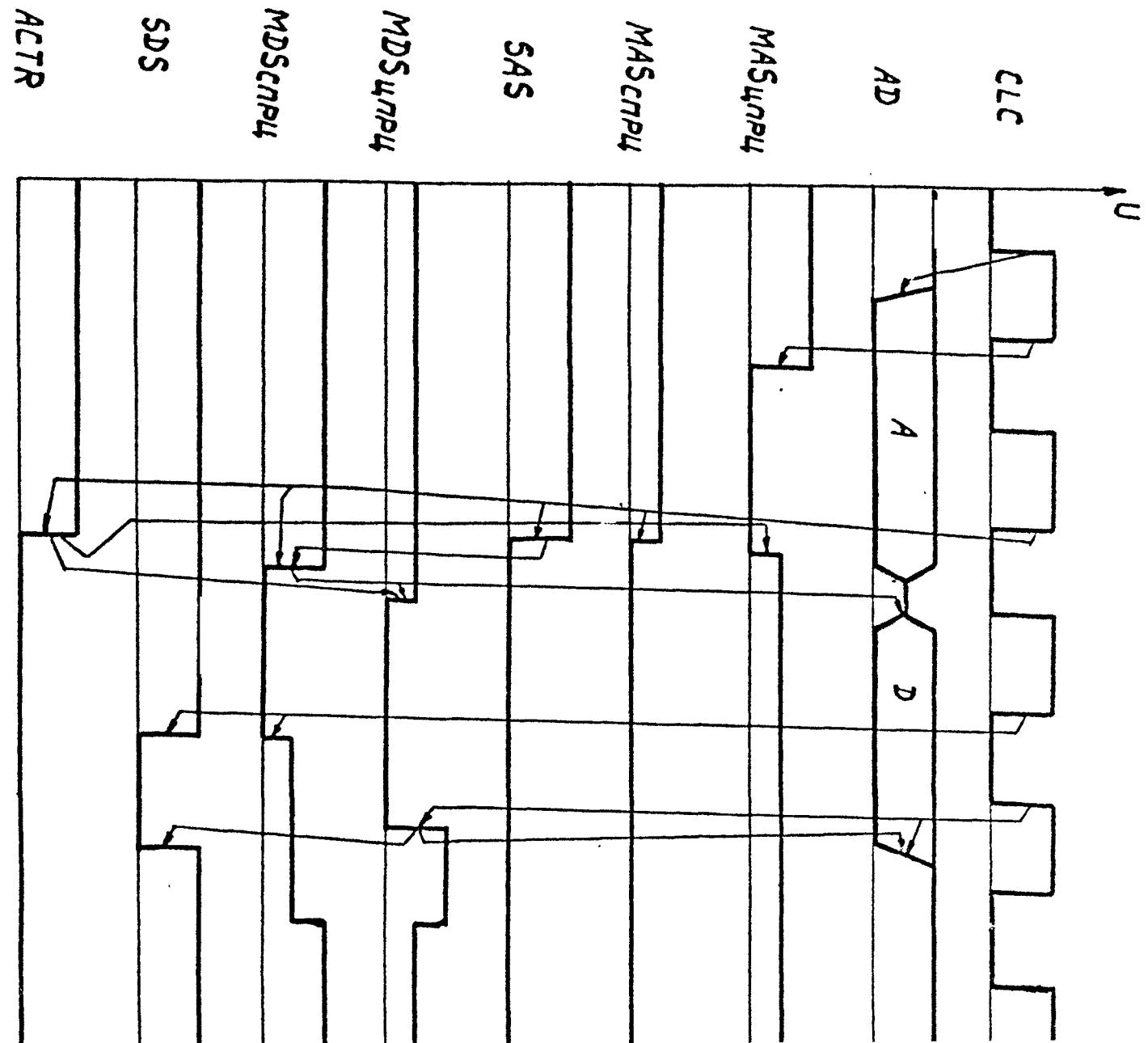
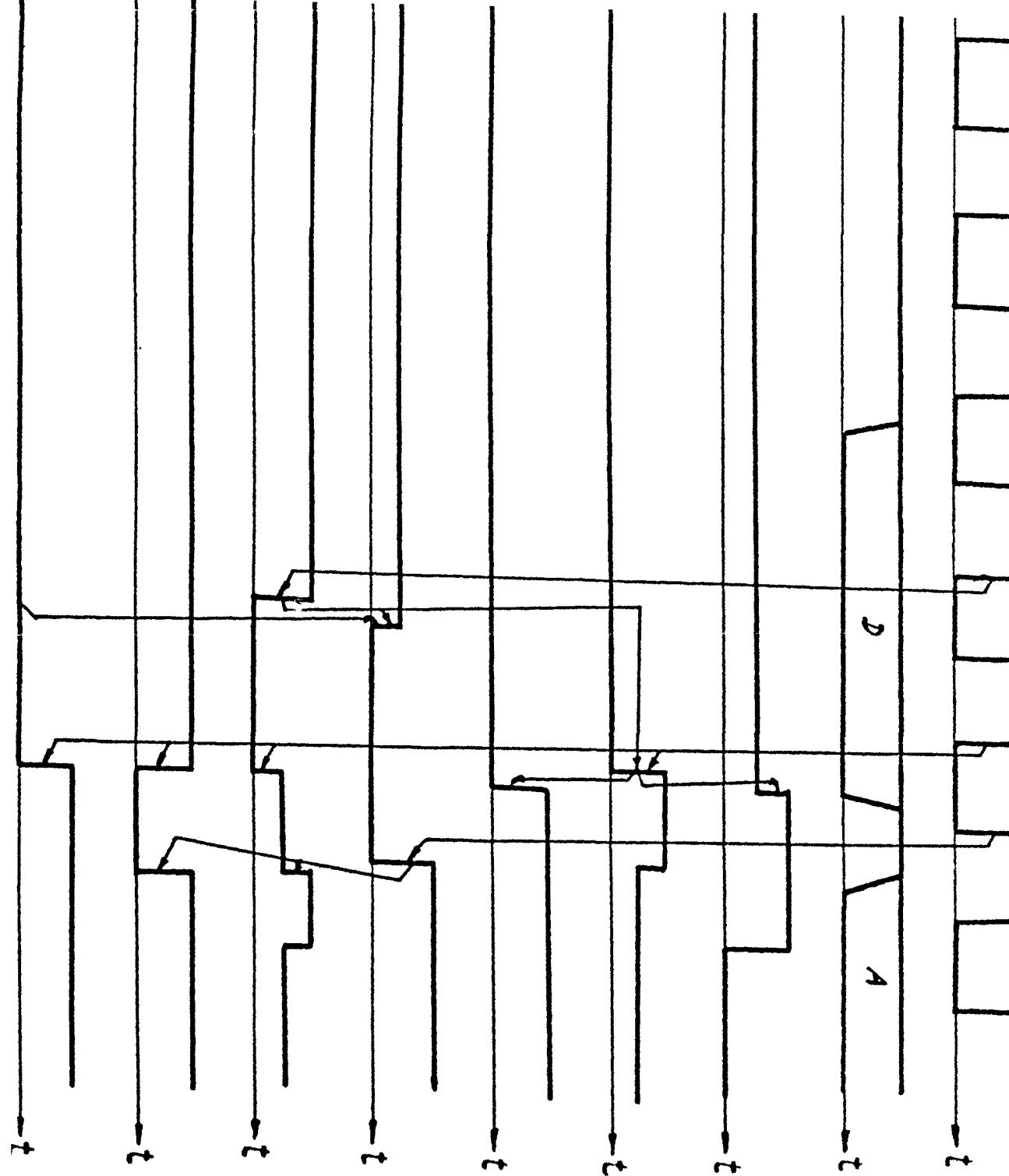


Рис. 24.

Диаграмма "Чтение-модификация-запись с перехватом"



Временная диаграмма процедуры "Люди"

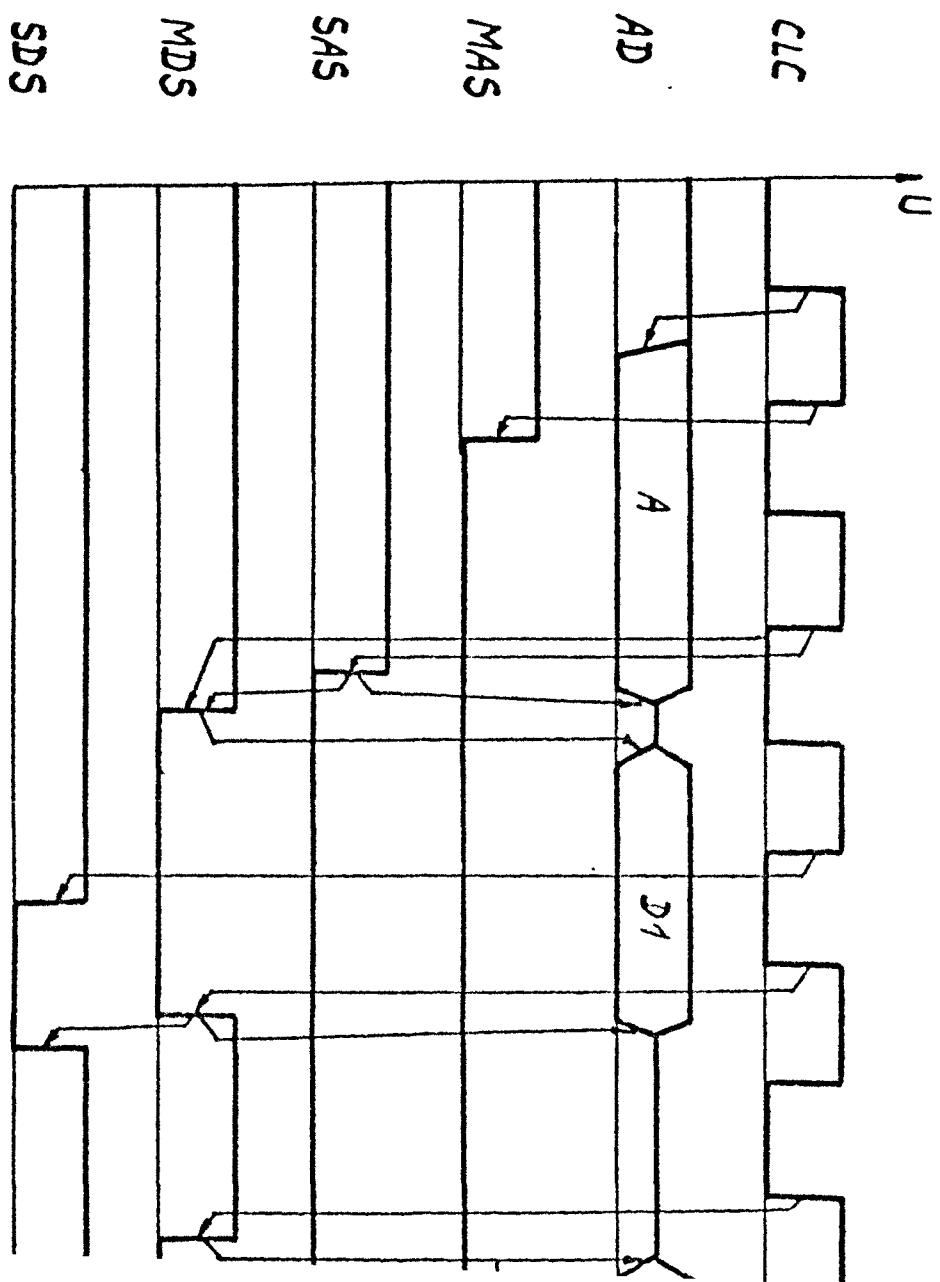
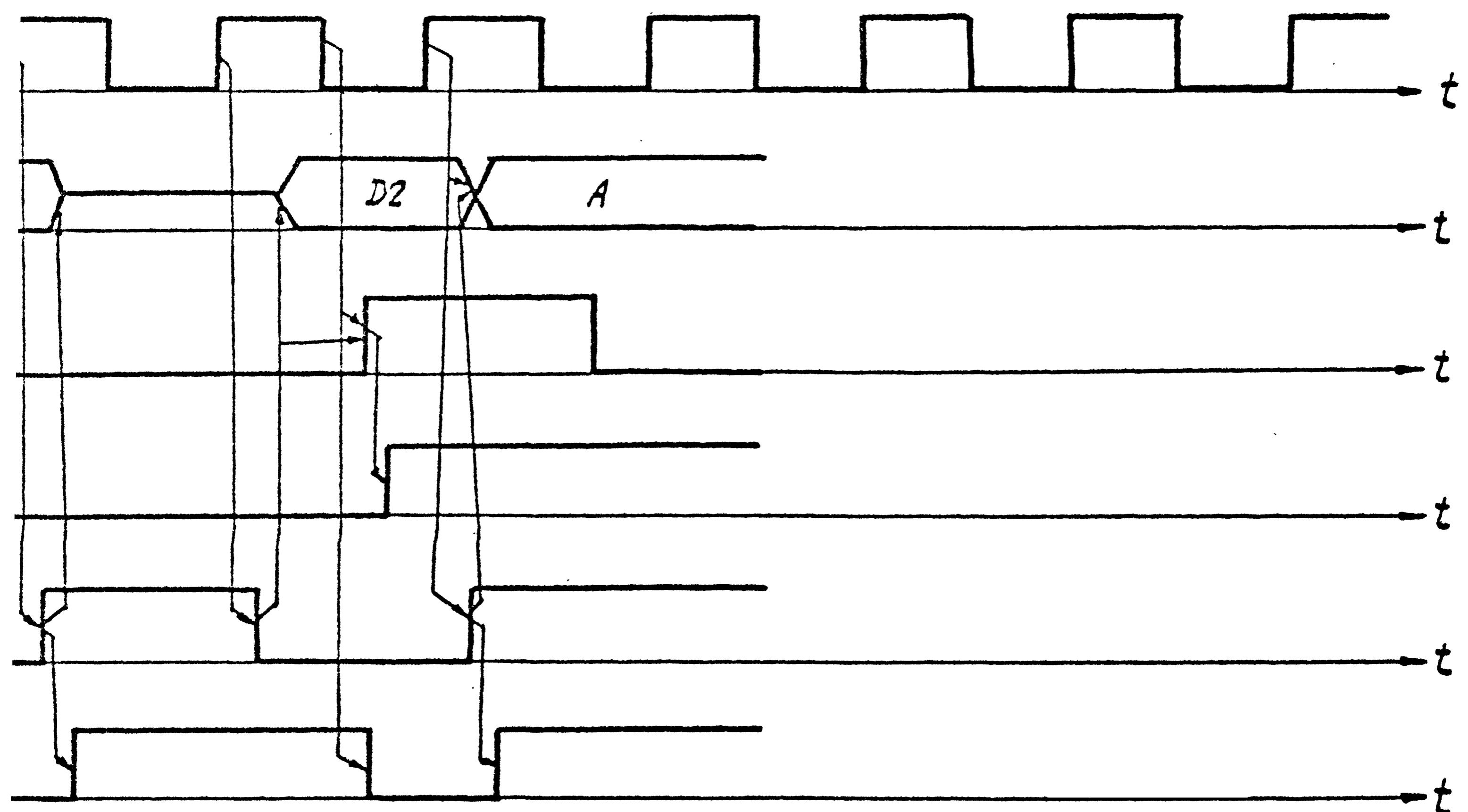


Рис. 25.

процедуры "Двойное адресное чтение"



25.

Временная диаграмма проекции "Двойная з

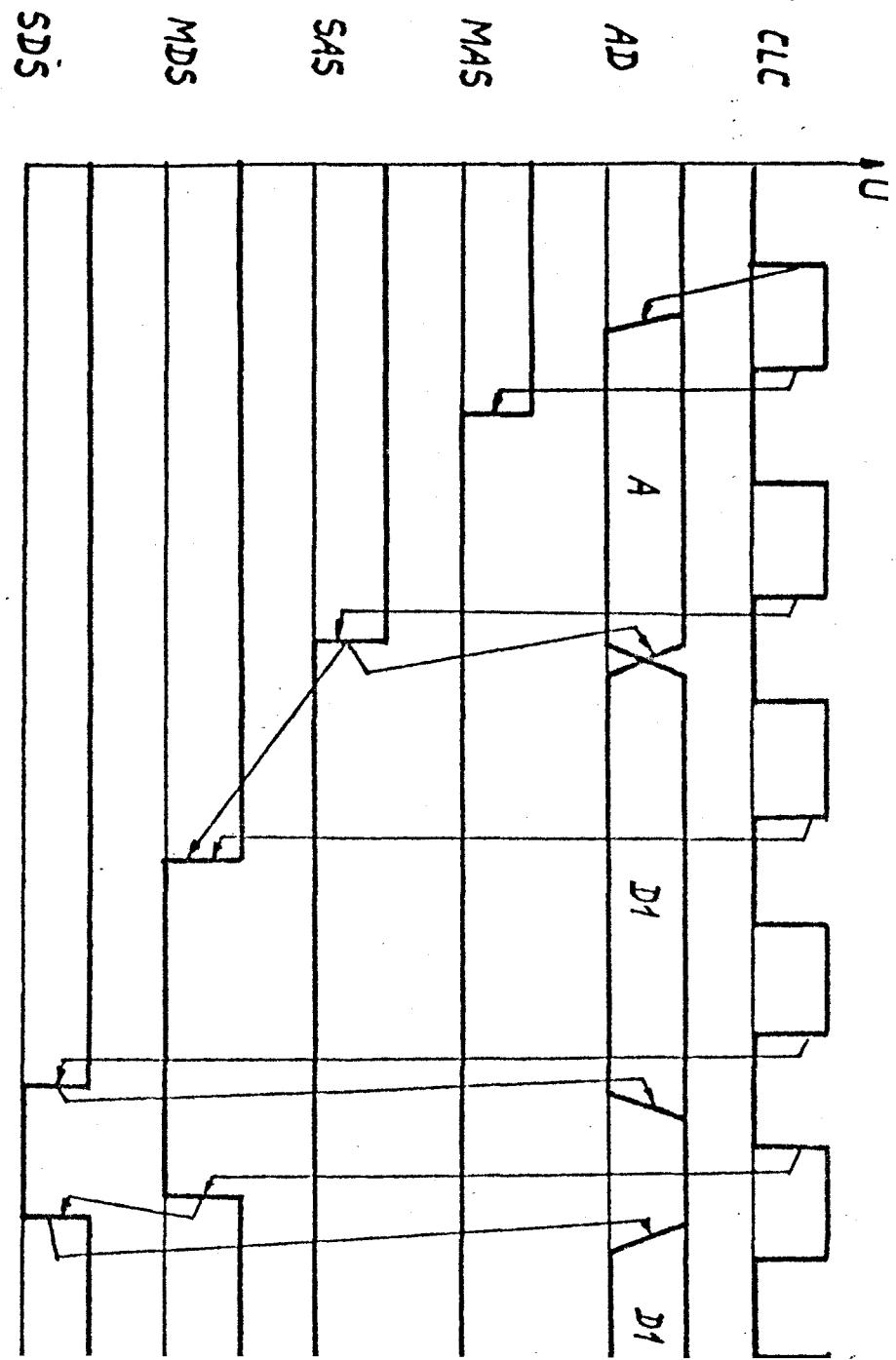
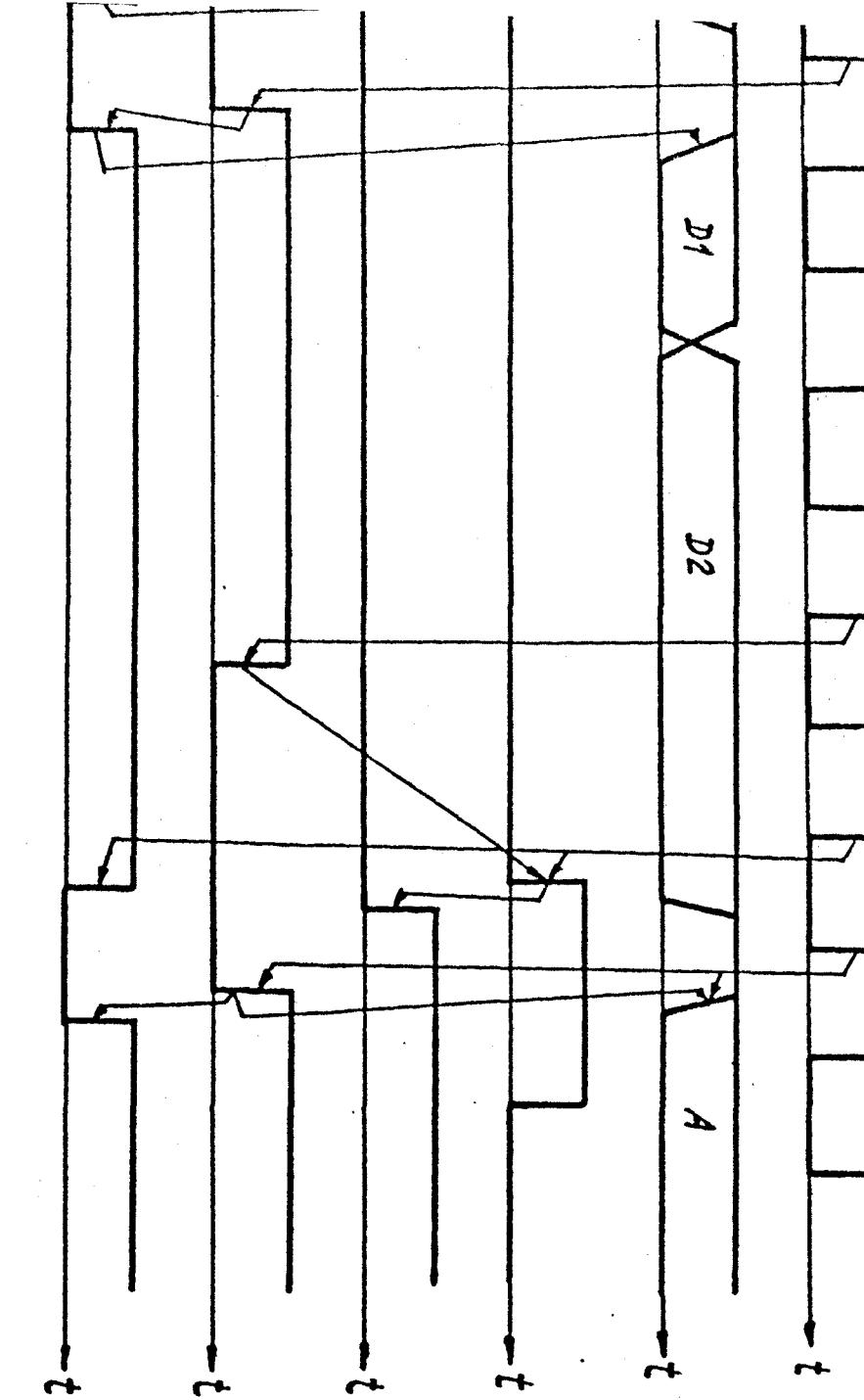


Рис. 26.



И "двойная запись"



26.

Запись № 5054м. поэт. Ариада

ЛМ3.480.319 ТО

69

Временные диаграммы адресного и безадр.

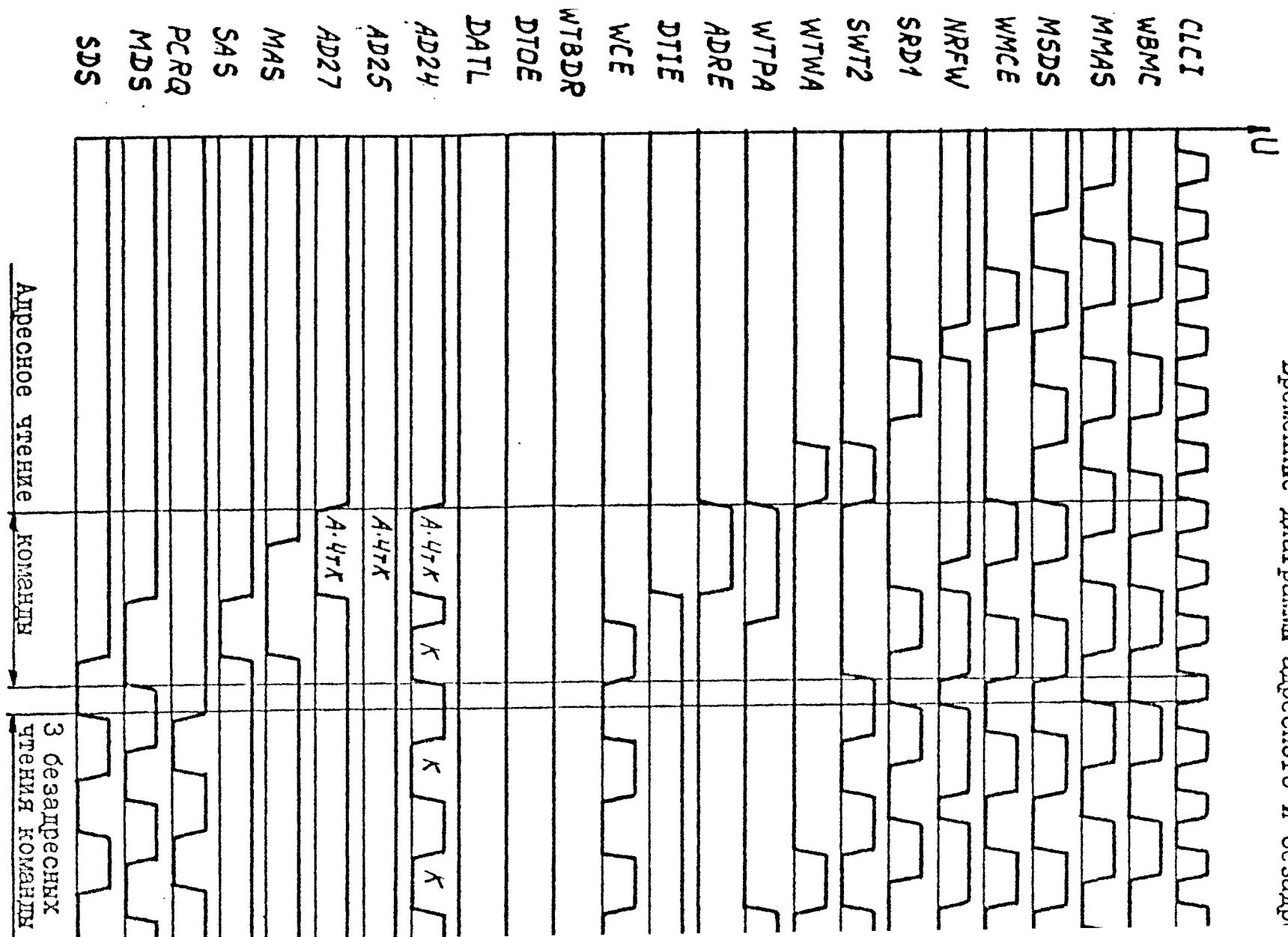
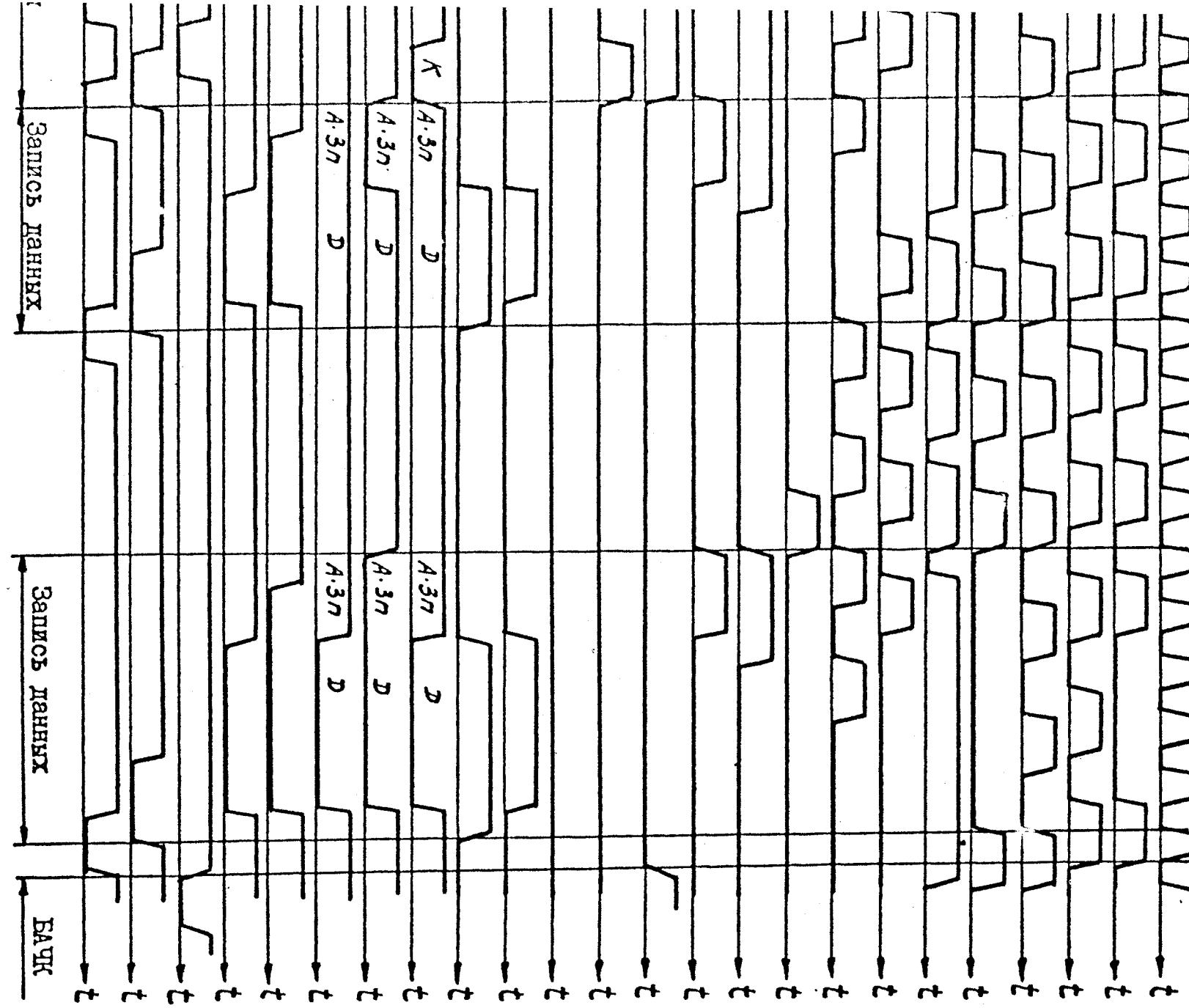


Рис. 2



Временные диаграммы записи и чтения.

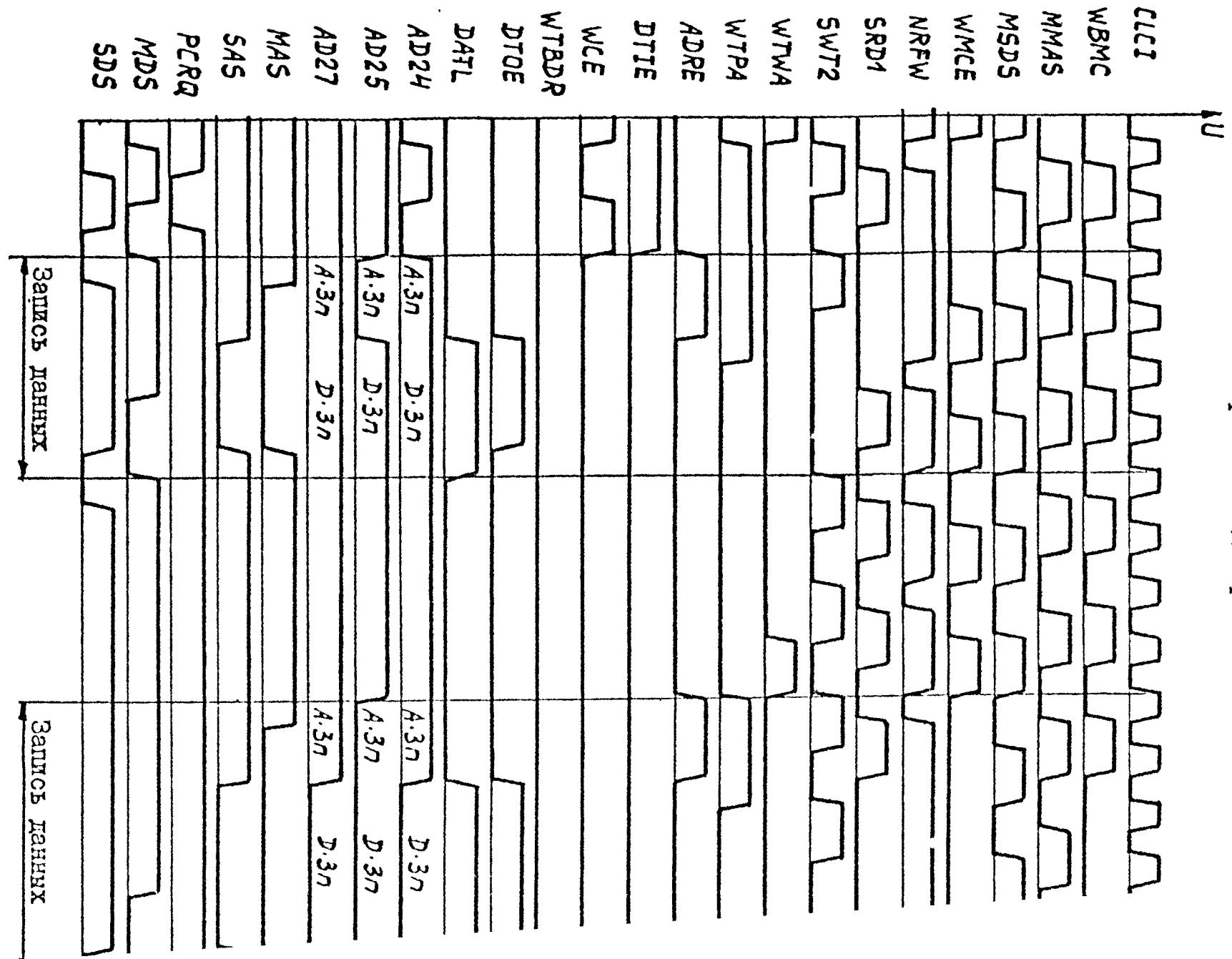
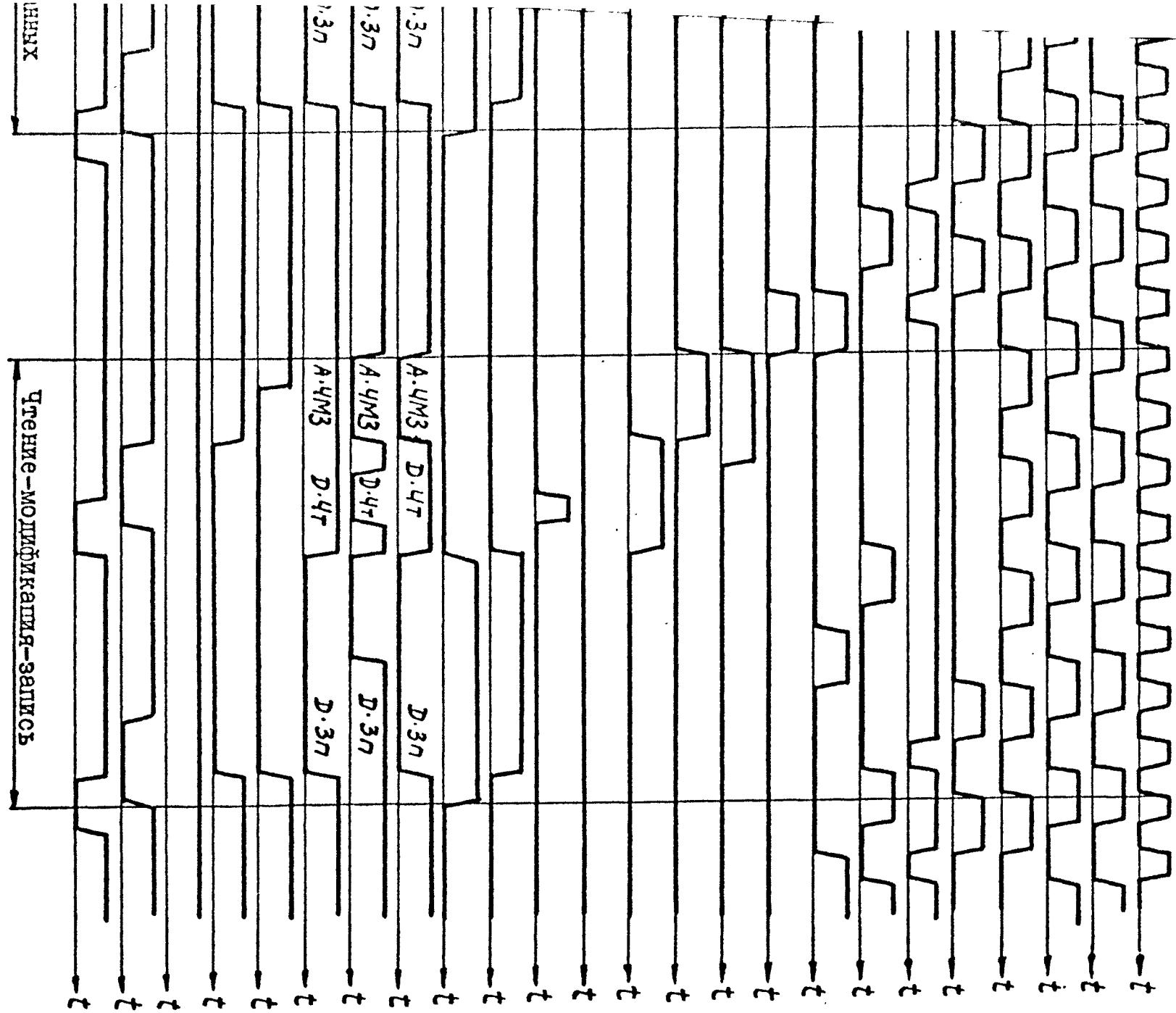


Рис. 28.

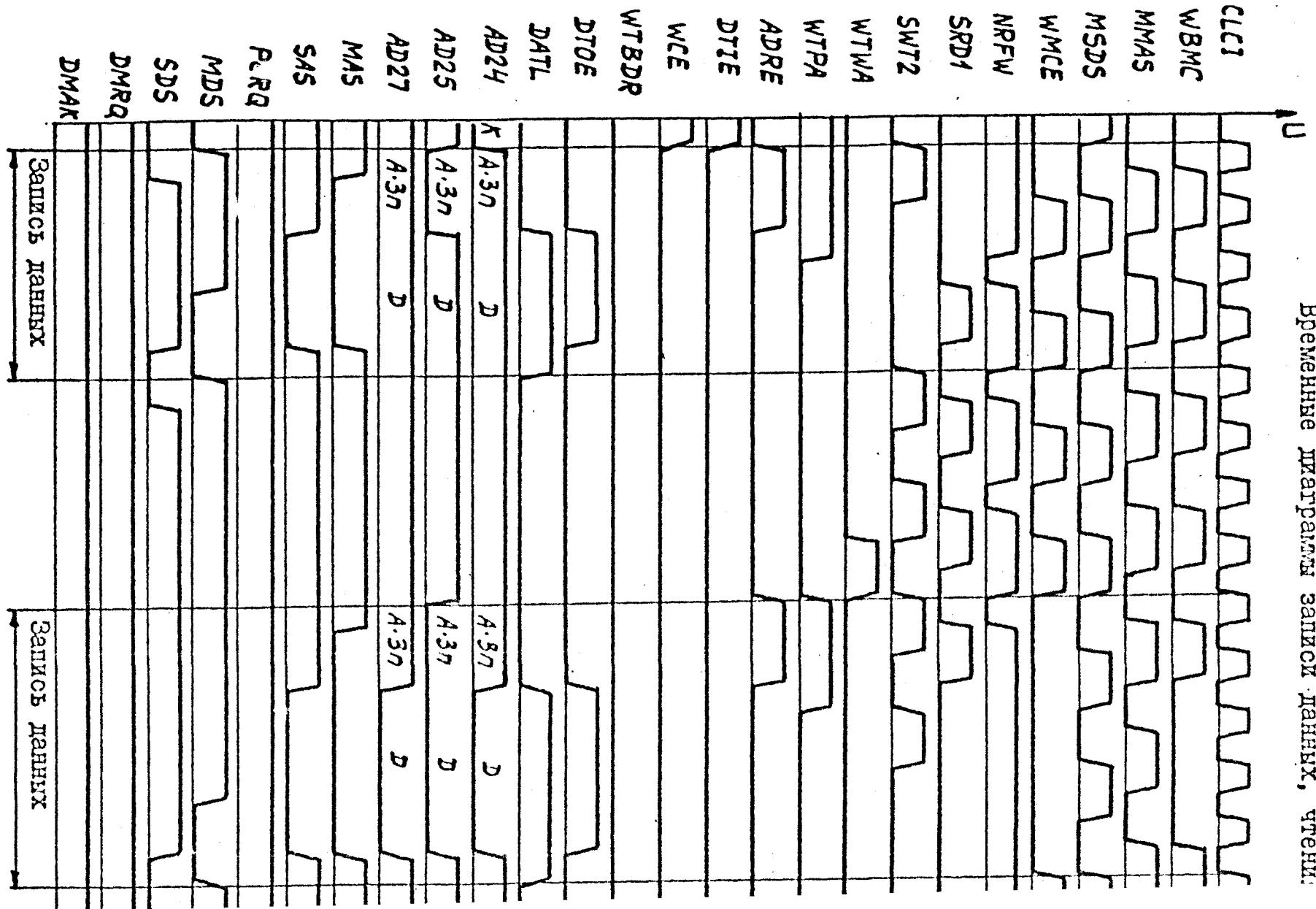
Модификации-записи



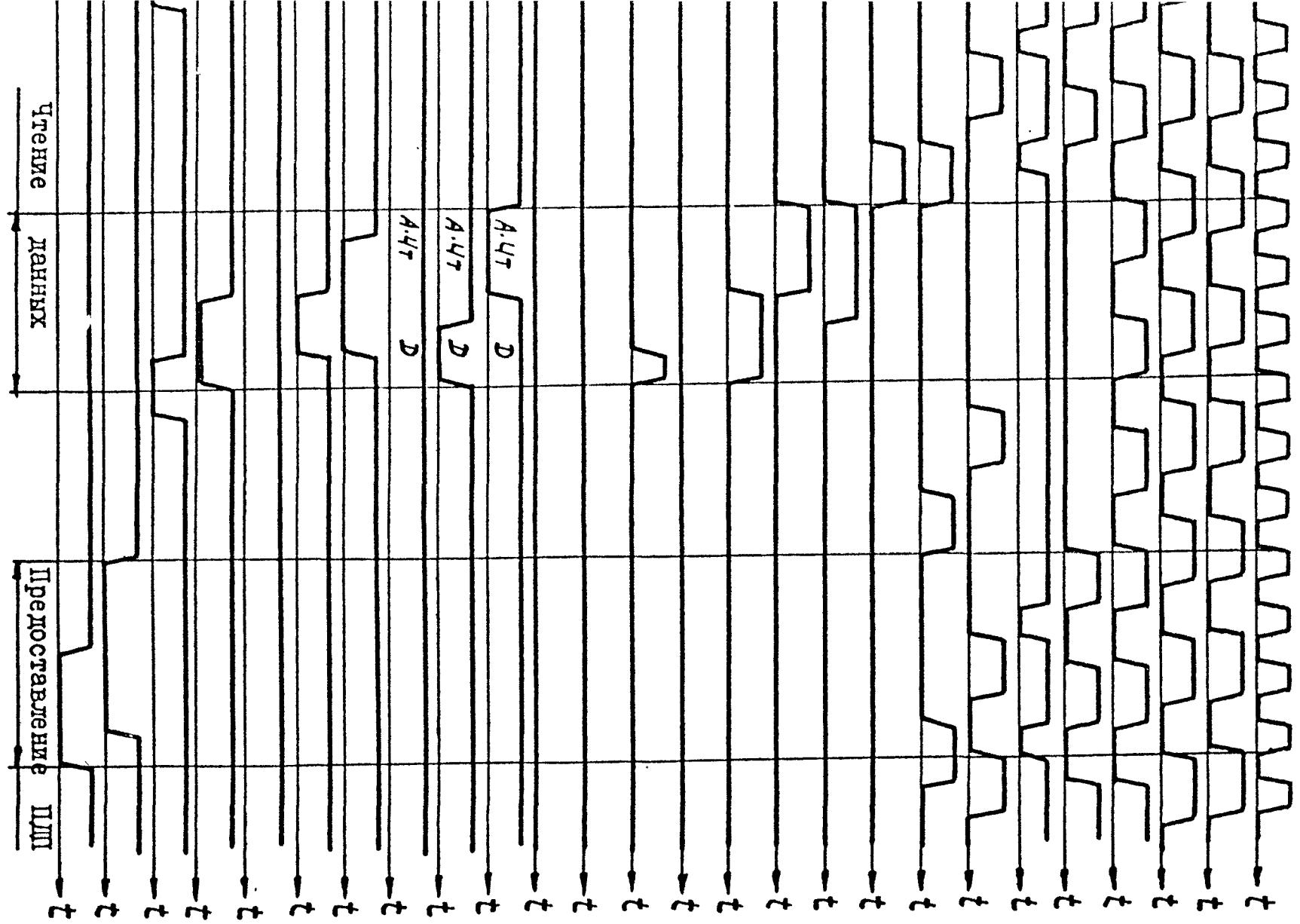
ШВ3.480.319 ТО

УЗОЛ	№ 504УМ	ПОДР.	4010

Временные диаграммы записи данных, чтения



данных и предоставления ПДИ



М.3.480.319 Т0

72

Бланк
№ 5005. Поясн. 1000

К внутренним сигналам процессора относятся:

- WBMC - запись в буфер микрокоманд;
WMCE - запись в регистр микрокоманд;
NRFW - сигнал записи поля чтения;
SRDI - сигнал чтения источника;
SWT2 - сигнал записи в приемник;
WTVA - запись в регистр виртуального адреса;
WTPA - запись в регистр физического адреса;
ADRE - сигнал разрешения выдачи адреса в системную магистраль AD(0-31);
DTIE - сигнал приема команд и данных на внутреннюю магистраль MD(0-31);
WCE - сигнал записи в буферные регистры команд;
WTBDR - сигнал записи в буферные регистры данных;
DTOE - сигнал выдачи данных в системную магистраль AD(0-31) при записи;
DATL - сигнал выдачи данных с выходного буферного регистра данных во внутреннюю магистраль MD(0-31).

I.IO.2. Идентификаторы обменов

Старший байт физического адреса содержит идентификаторы обменов. В таблицах, приведенных ниже, даны логические значения кодов и электрические уровни на магистрали AD .

31	30	29	28	27	26	25	24
Направление обмена			Формат данных			Код обмена	

Временные диаграммы торможения обмена из-за ПЛ и торможения ПЛ

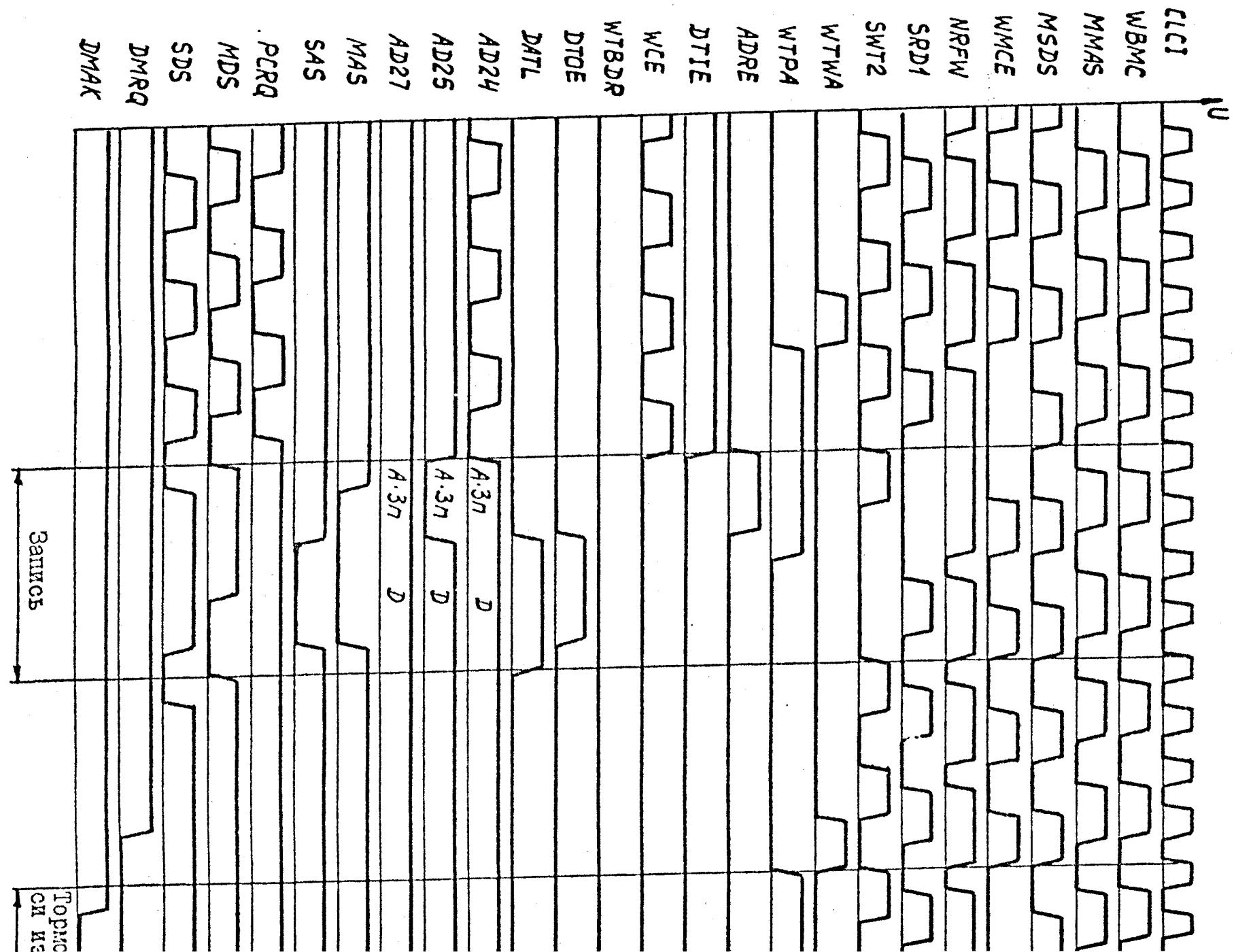
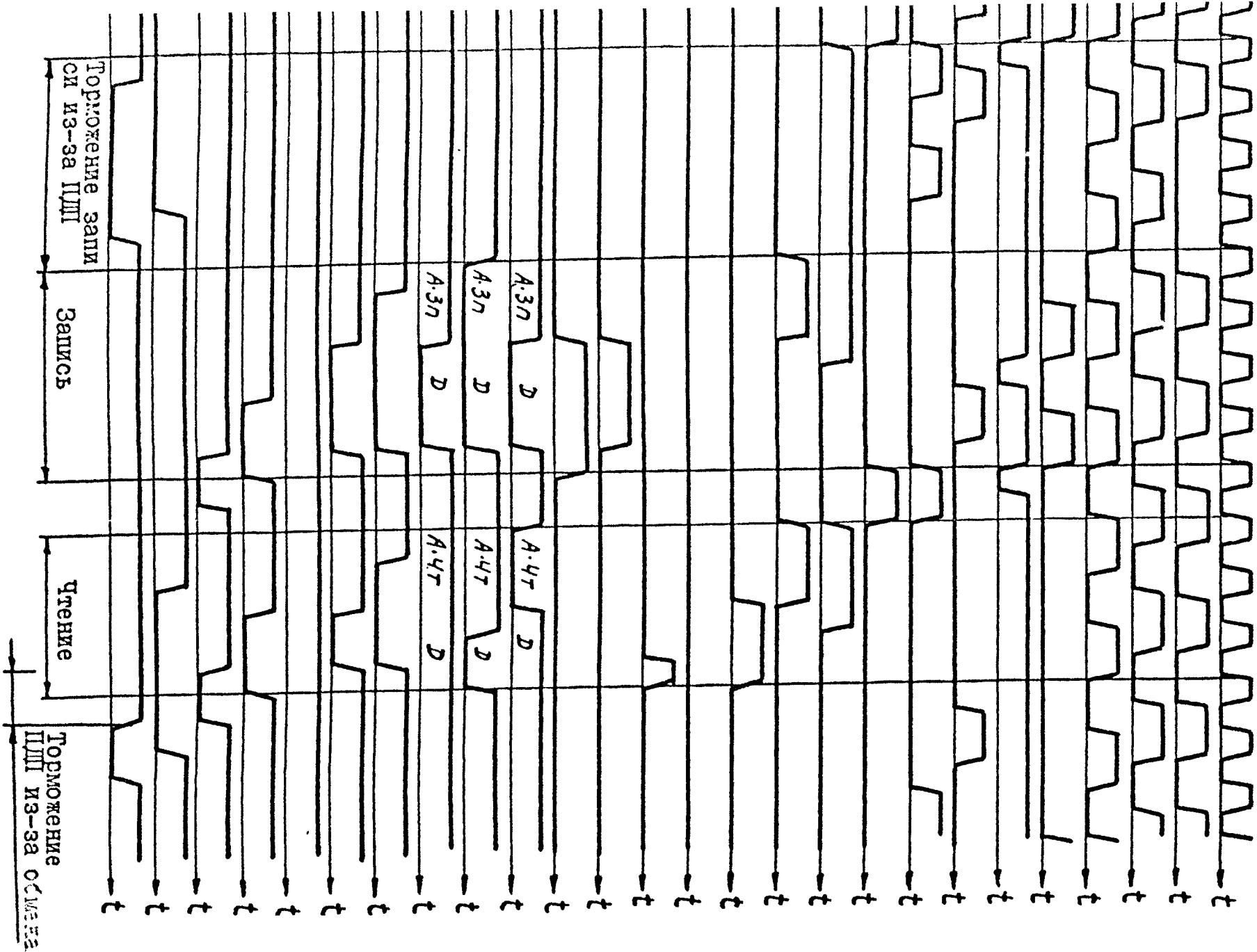


Рис. 30.

ложе-
ния ПДП из-за обмена



I.10.2.1. Направление обмена

31 разряд		30 разряд		Направление обмена
логические значения	уровень	логические значения	уровень	
0	высокий	0	высокий	Основная память, регистры внешних устройств
0	высокий	1	низкий	Основная память, с перехватом магистрали сопроцессором
1	низкий	0	высокий	Внутренние регистры
1	низкий	1	низкий	Системная память

I.10.2.2. Формат данных

29 разряд		28 разряд		Формат данных
логические значения	уровень	логические значения	уровень	
0	высокий	0	высокий	Двойное слово
0	высокий	1	низкий	Слово
1	низкий	0	высокий	Четверное слово
1	низкий	1	низкий	Байт

I.10.2.3. Код обмена

27 разряд		26 разряд		25 разряд		24 разряд		Обмен
лог. знач	уровень							
0	высокий	X	X	0	высокий	1	низкий	Чтение данных
1	низкий	X	X	0	высокий	1	низкий	Чтение команд
0	высокий	X	X	1	низкий	0	высокий	Запись
0	высокий	X	X	1	низкий	1	низкий	Чтение-модификация - запись

где X – безразличное значение, остальные комбинации 24-27 разрядов – резервные.

I.II. Функциональные и электрические схемы входных и выходных каскадов

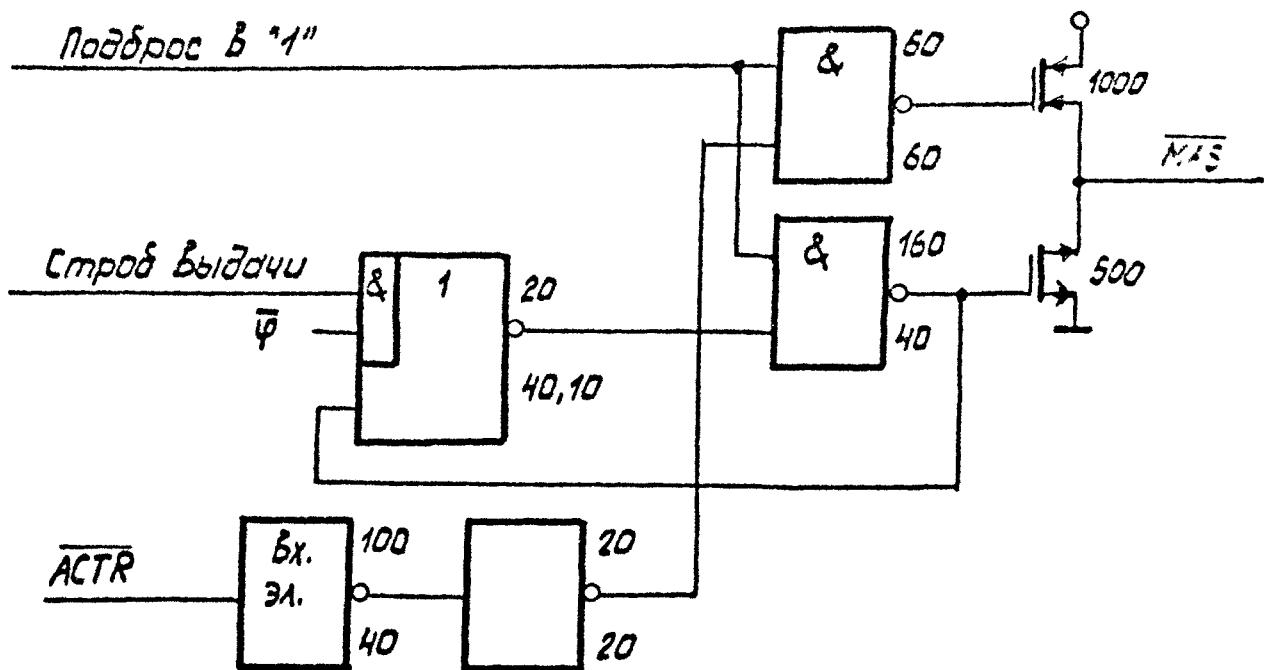
Функциональные схемы входных и выходных каскадов представлены на рис. 3I-39.

На всех рисунках численное значение около логических элементов обозначает ширину транзисторов элементов в мкм: сверху р-типа, снизу п-типа.

Внутренние тактовые сигналы синхронизации φ , $\bar{\varphi}$ соответствуют следующим фазам входного сигнала CLCI: φ соответствует CLCI; $\bar{\varphi}$ – \overline{CLCI} .

Электрические схемы входов и входов/выходов представлены на рис. 40,4I.

Схема выходного каскада MAS



\overline{ACTR} – сигнал захвата магистрали акселератором

Рис. 31.

Схема выходного каскада \overline{PGB} , \overline{PCRQ} , \overline{DMAK}

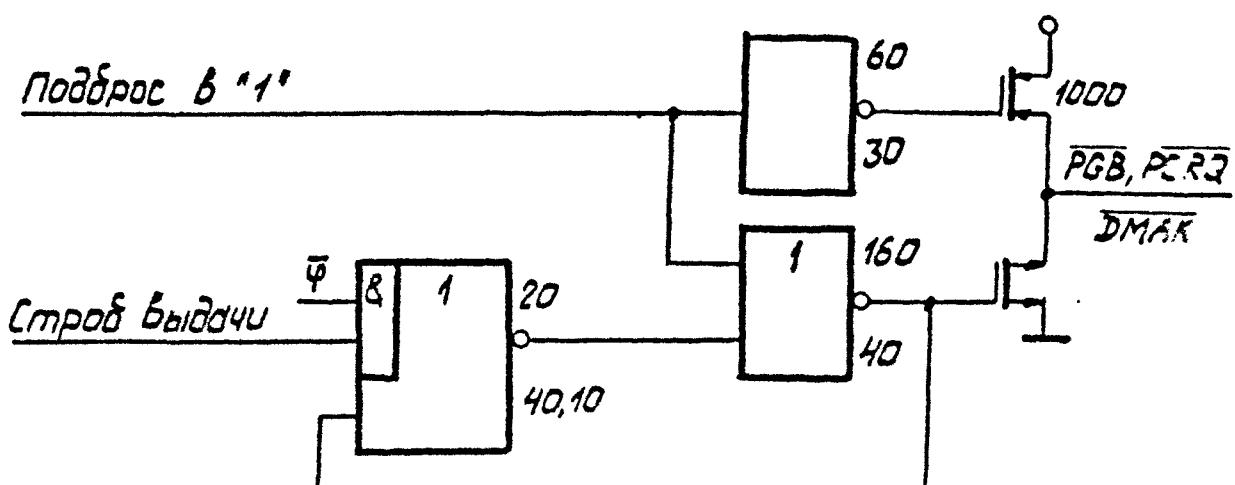
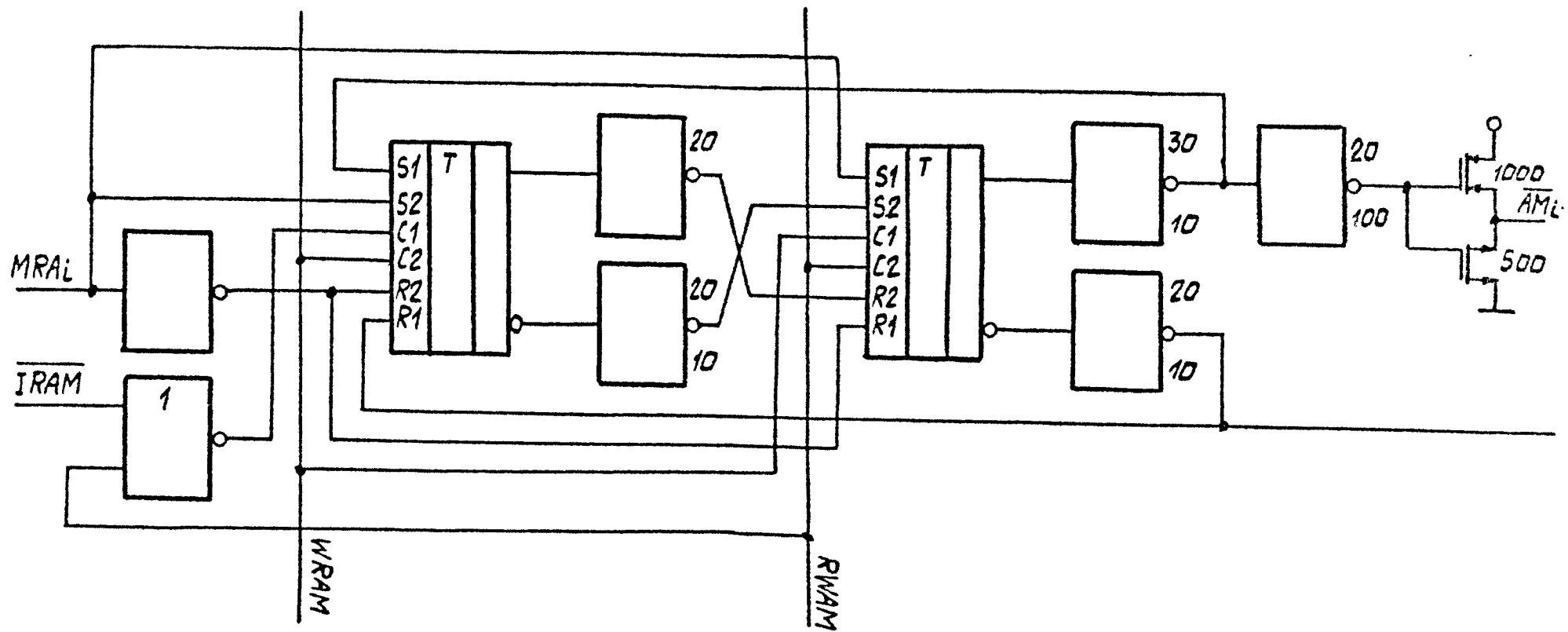


Рис. 32.

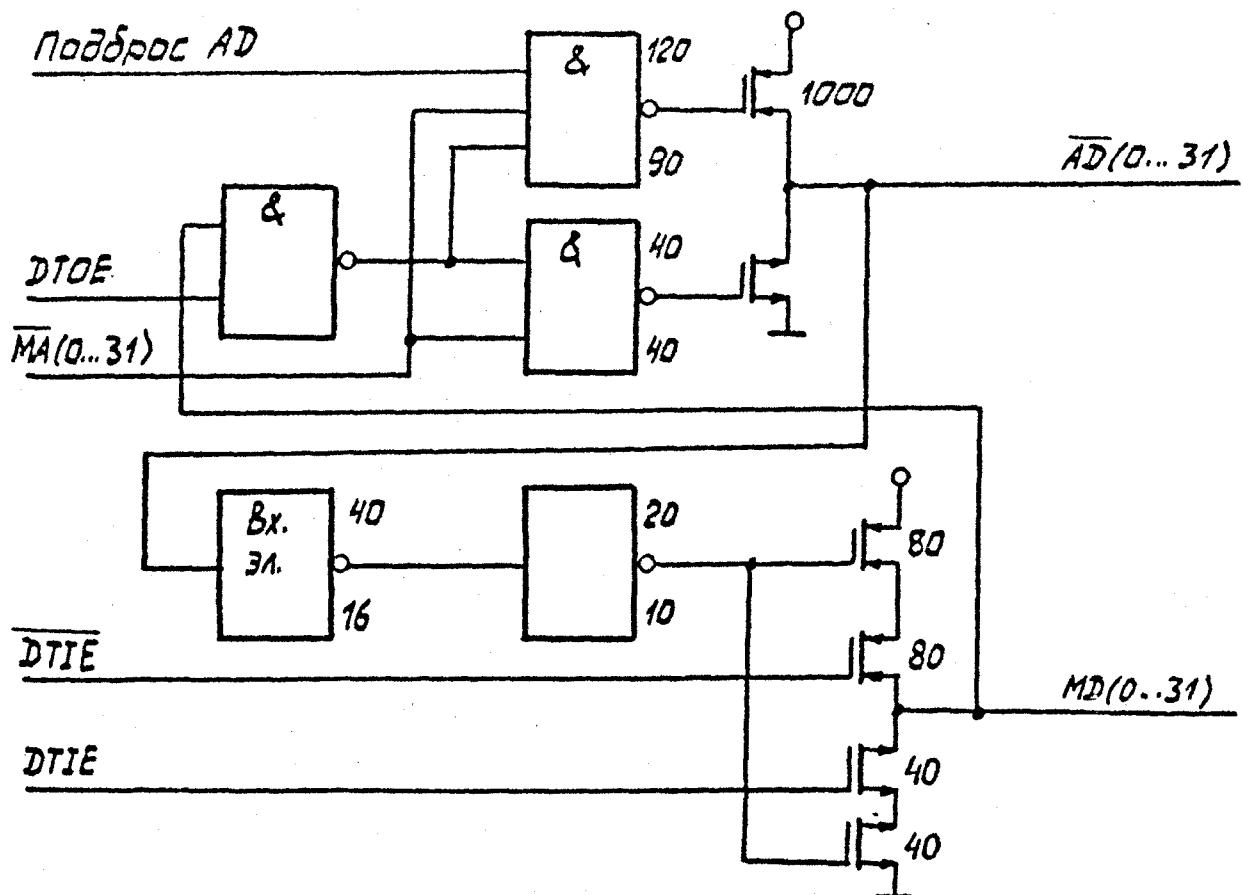
Схема одного разряда регистра и выходного каскада адреса микрокоманды AM(0...I3)



- MRAI** - *i*-разряд магистрали регистра адреса микрокоманды
- IRAM** - сигнал инкремента регистра адреса микрокоманды
- WRAM** - сигнал записи в регистр адреса микрокоманды
- RWAM** - сигнал переписи во второй каскад регистра адреса микрокоманды

Рис. 33,

Схема каскада вход/выход AD(0...3I)



$\overline{MA}(0..3I)$ - магистраль адреса

$\overline{MD}(0..3I)$ - магистраль данных

\overline{DTIE} , \overline{DTIE} - сигналы стробирования приема данных с AD на MD при чтении

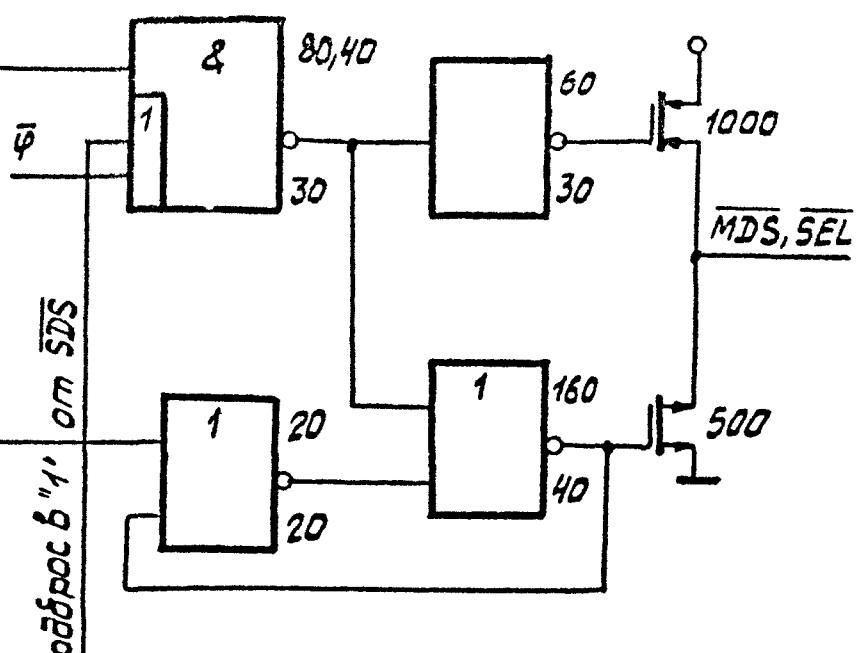
\overline{DTOE} - сигнал стробирования выдачи данных с MD на AD при записи

Подброс AD - сигнал подброса магистрали $\overline{AD}(0..3I)$ в электрическую "1" по окончании обмена

Рис. 34

Схема выходного каскада \overline{MDS} , \overline{SEL}

Подбрас в "1" от $\hat{R}/23$



Строй выдачу

Подбрас в "1" от SDS

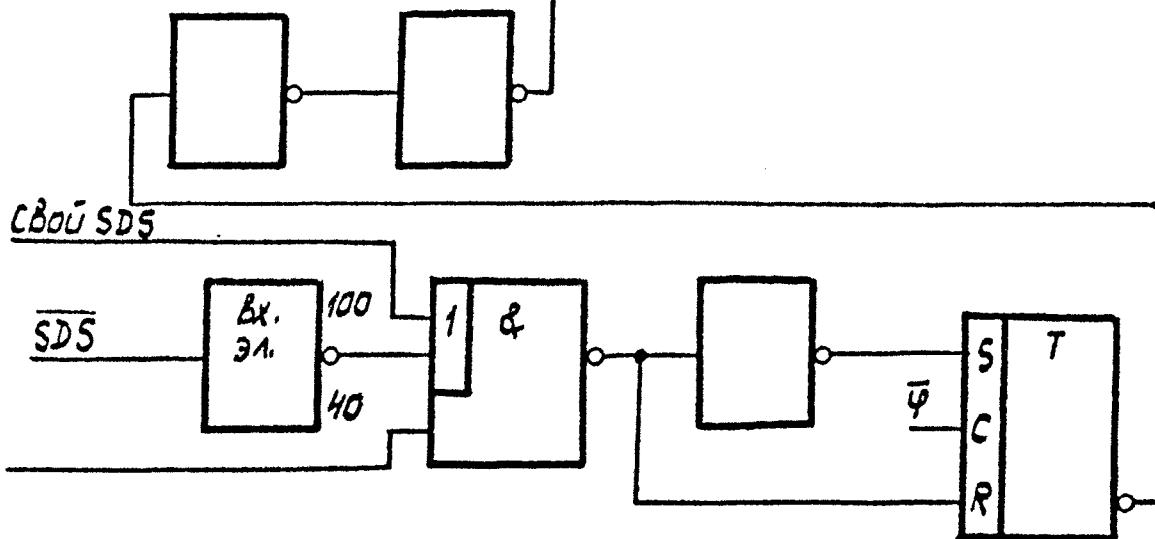
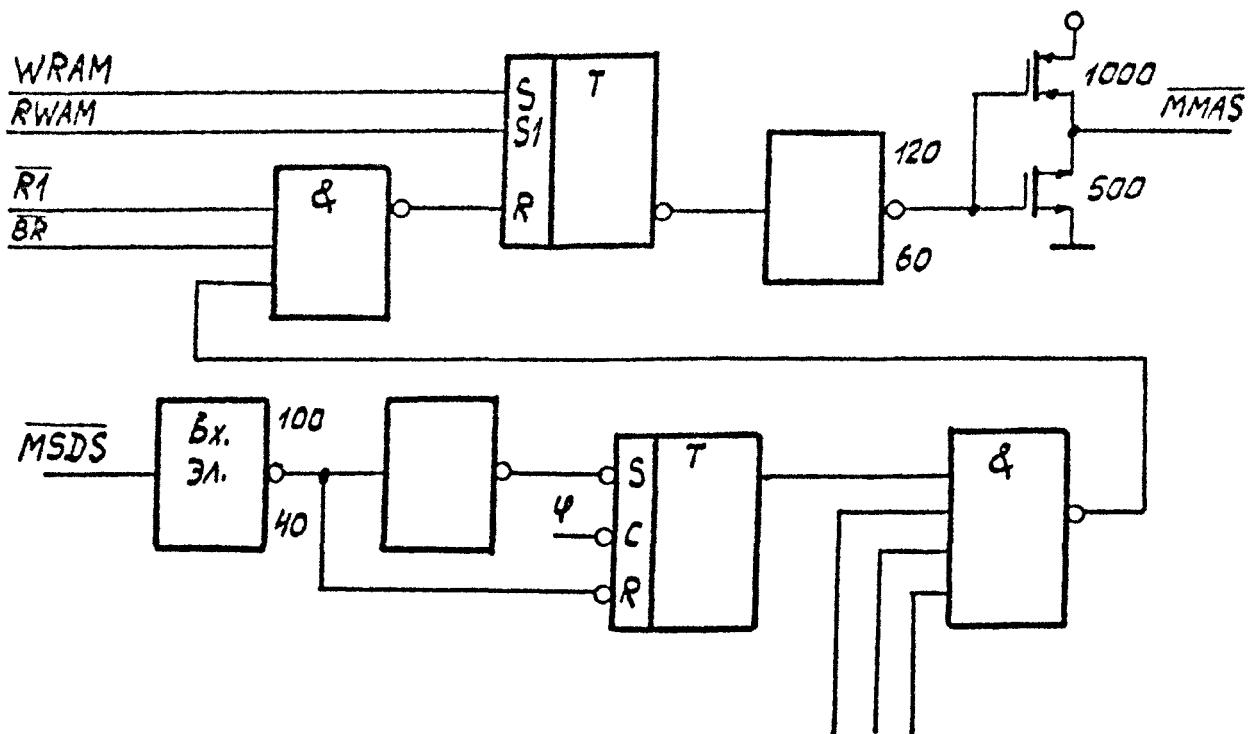


Рис. 35.

Схема выходного каскада MMAS и входного каскада MSDS



WRAM - сигнал записи в регистр адреса микрокоманды

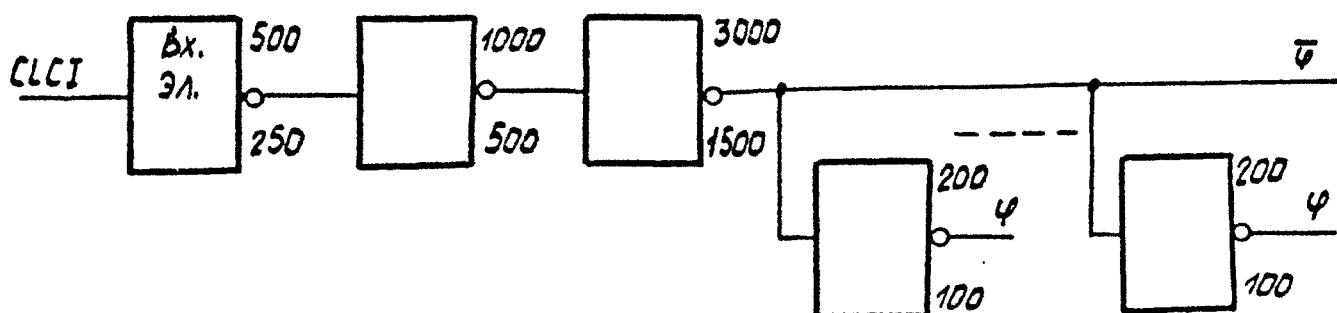
RWAM - сигнал переписи во второй каскад регистра адреса микрокоманды

\overline{RI} - сигнал общего сброса

\overline{BR} - сигнал ветвления

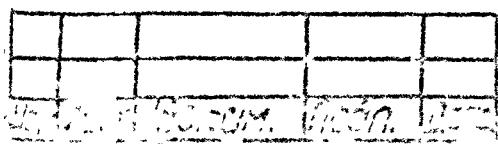
Рис. 36.

Схема формирователя внутренних тактовых сигналов
 ψ , $\bar{\psi}$ от входного сигнала CLCI



один элемент на 30 эквивалентных нагрузок типа
 инвертора с $W_p=20$, $W_n=10$

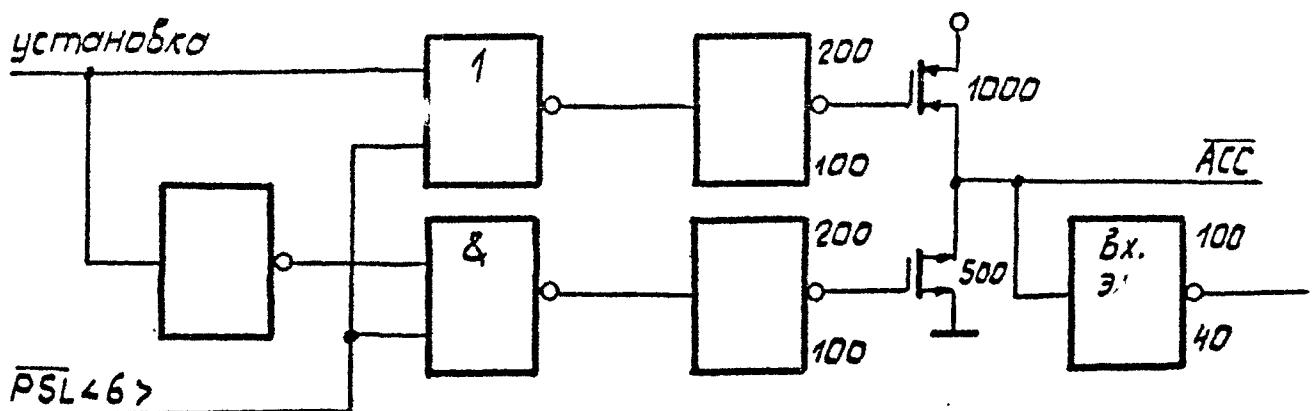
Рис. 37.



1.13.420.319 TO

Subject
S1

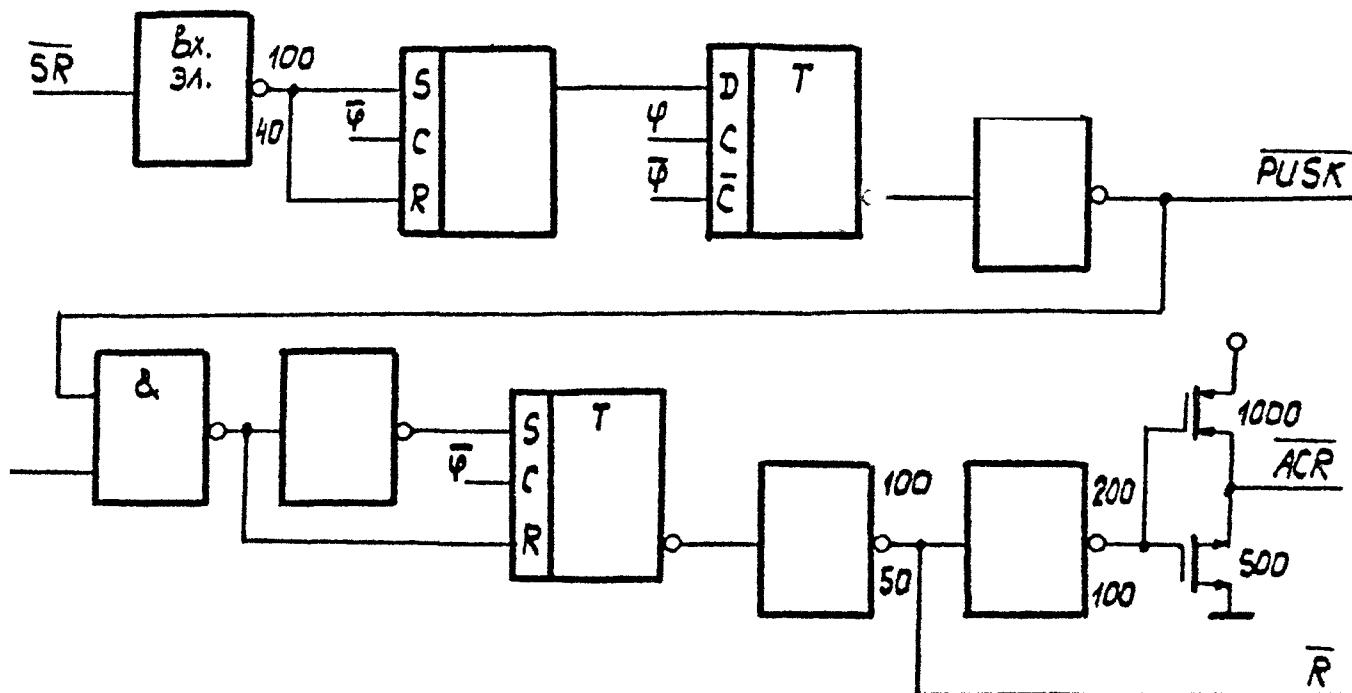
Схема каскада вход/выход АСС



PSL <6> - бит разрешения ловушки плавающего недополнения FU регистра состояний PSL

Рис. 38.

Схема входного каскада \overline{SR} , каскада формирования сигнала общего сброса \overline{R} и выходного каскада ACR

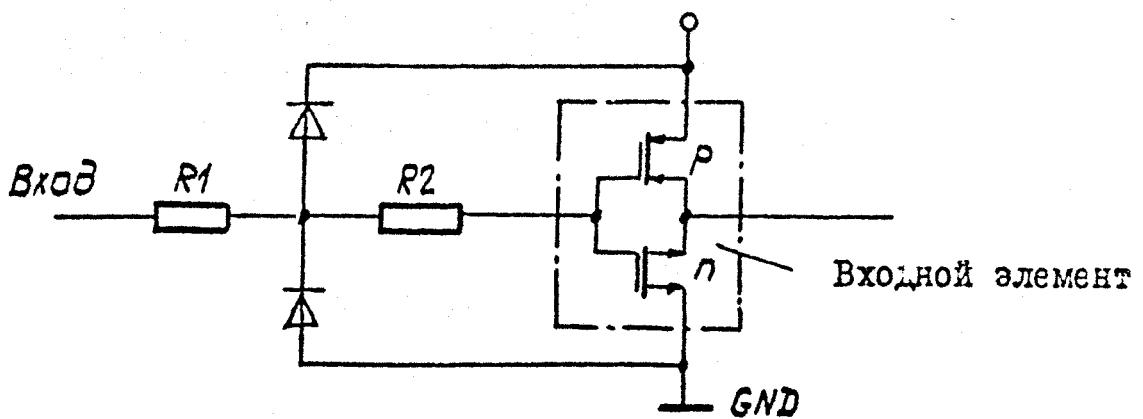


PUSH - сигнал пуска

Рис. 39.

				KMS.480.319 TO	<i>SAC</i>
STRUCTUREM.	ROON.	DATE			82

Электрическая схема входов MC(0...3I), SAS, SDS, DMRQ, ACTR, INRA, INRC, MCH, HALT, MSDS, ACRA, ACN, ACZ, ACV, TTEST, SR, CLCI

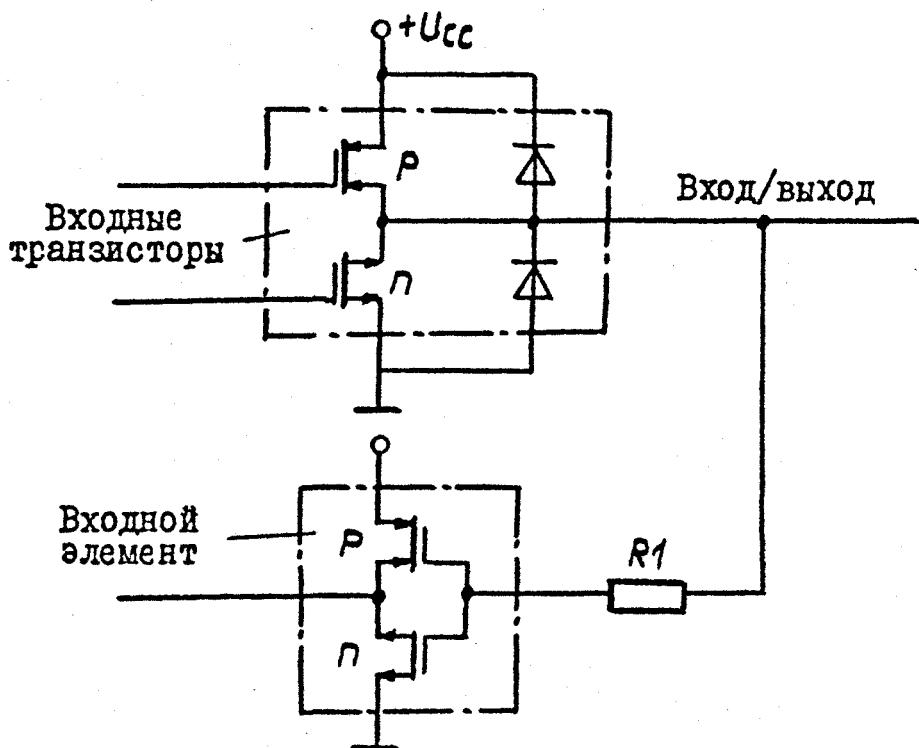


RI - резистор поликремниевый Si* 15,7 \square \approx 1,5 кОм

R2 - резистор диффузионный p+ 3Ω ≈ 350 Ом

Рис. 40.

Электрическая схема входов/выходов AD(0...3I), ACC



RI - резистор диффузионный p^+ $3\Omega \approx 350\text{ Ом}$

Рис. 47.

I.12. Система микрокоманд

В микропроцессорной системе "Электроника 32" имеется несколько форматов микрокоманд.

Форматы MA, MB, MC это содержательные микрокоманды операционного блока (ОБ), остальные форматы микрокоманд не воспринимаются ОБ и служат для управления выборкой микрокоманд из ПЗУ-микрокода, а также для управления ПЛМ-десифратора команд (ПЛМ ДШК).

Форматы микрокоманд операционного блока МА, МВ, МС приведены на рис. 42.

Форматы микрокоманд операционного блока

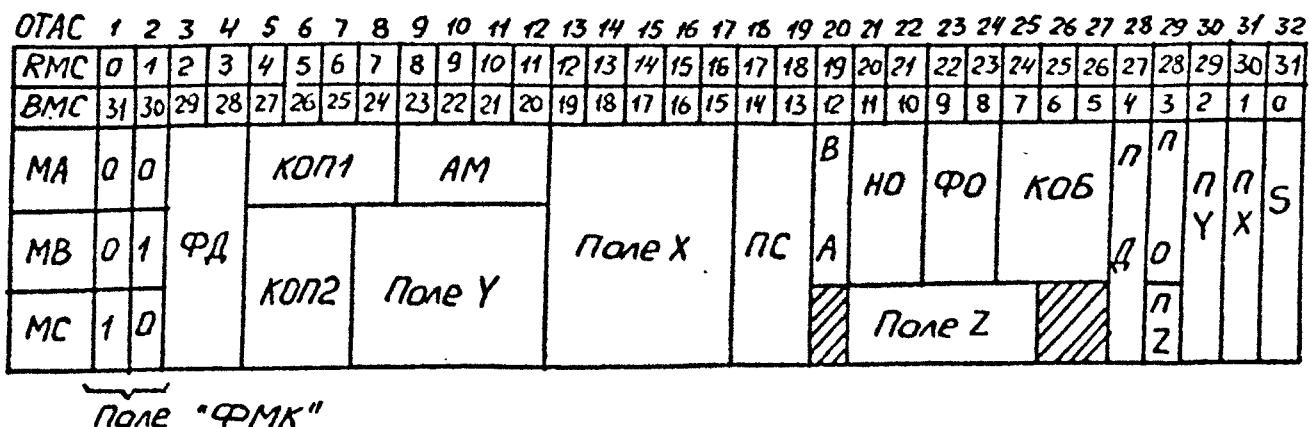


Рис. 42.

Расшифруем значение полей содержательных микрокоманд.

Поле "ФМК" – формат микрокоманды

Разряды МК		Наименование формата микрокоманды	Обобщенное обозначение
31	30		
0	0	Одноадресная ОБ	МА
0	I	Двухадресная ОБ	МВ
I	0	Трехадресная ОБ	МС

Поле "ФД" - формат данных

Разряды МК		Формат данных в ОБ
29	28	
0	0	Двойное слово (ДС) - 32 разряда
0	I	Слово (СЛ) - 16 разрядов (возможно изменение только мл. части РСП)
I	I	Байт (БТ) - 8 разрядов
I	0	Резервный код "ФД", признак подсветки приема в регистр косвенной адресации (РКА)

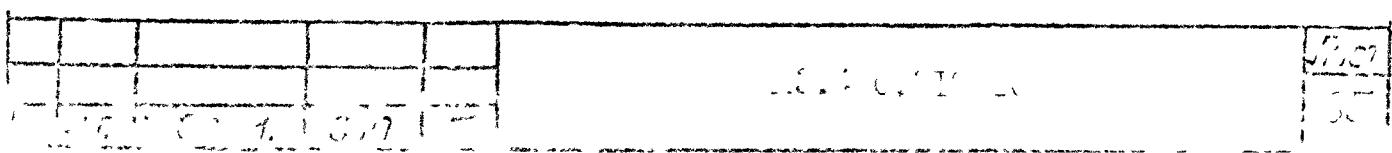
Поле "КОП2" - коды операций двухадресных и трехадресных микрокоманд

КОП2			Обозначение	Обозначение	Операция
Разряды МК					
27	26	25			
0	0	0	$Y \rightarrow X$	$X \rightarrow Z$	Пересылка
0	0	I	$X+Y \rightarrow X$	$X+Y \rightarrow Z$	Сложение
0	I	0	$X-Y \rightarrow X$	$X-Y \rightarrow Z$	Вычитание
0	I	I	$Y-X \rightarrow X$	$Y-X \rightarrow Z$	Обр.вычитание
I	0	0	$X \oplus Y \rightarrow X$	$X \oplus Y \rightarrow Z$	$MOD2$
I	0	I	$X \vee Y \rightarrow X$	$X \vee Y \rightarrow Z$	Дизъюнкция
I	I	0	$X \& \bar{Y} \rightarrow X$	$X \& \bar{Y} \rightarrow Z$	Конъюнкция с отрицанием
I	I	I	$X \& Y \rightarrow X$	$X \& Y \rightarrow Z$	Конъюнкция

Здесь: Для двухадресных микрокоманд:

X - операнд в поле X (МК I9-I5),

Y - операнд в поле Y (МК 24-20),



Для трехадресных микрокоманд:

X - операнд в поле X (МК 19-15),

Y - операнд в поле Y (МК 24-20),

Z - операнд в поле Z (МК 11-7).

Поле "КОПІ" - коды операций одноадресных микрокоманд

КОПІ				Обозначение	О п е р а ц и я
Разряды МК					
27	26	25	24		
0	0	0	0	$M \rightarrow X$	Пересылка
0	0	0	I		Резерв
0	0	I	0	$X + M \rightarrow X$	Сложение
0	0	I	I	$X \rightarrow X$	Пересылка
0	I	0	0	$X - M \rightarrow X$	Вычитание
0	I	0	I		Резерв
0	I	I	0	$M - X \rightarrow X$	Обратное вычитание
0	I	I	I	$X \rightarrow \bar{X}$	Инверсия
I	0	0	0	$X \oplus M \rightarrow X$	MOD2
I	0	0	I	АЛ(Х)	Арифметический сдвиг влево
I	0	I	0	$X \vee M \rightarrow X$	Дизъюнкция
I	0	I	I	ЦЛ(Х)	Циклический сдвиг влево
I	I	0	0	$X \& \bar{M} \rightarrow X$	Конъюнкция с отрицанием
I	I	0	I	АП(Х)	Арифметический сдвиг вправо
I	I	I	0	$X \& M \rightarrow X$	Конъюнкция
I	I	I	I	ЦП(Х)	Циклический сдвиг вправо

Здесь X - операнд по адресу в поле X (МК 19-15),

M - модификатор, или константа, по адресу в поле АМ (МК 23-20),

(т.е. в одноадресной микроинструкции константа читается по полу "Y").

Поле "ПС" – прием состояния

ПС		Прием состояния	
разряды МК			
I4	I3		
0	0	Запись признаков N, Z, V, C	(ЗС)
I	0	Запись N, Z, V ; Сохранение C	(МС)
0	I	Запись N, Z , обнуление V, C	(НВ)
I	I	Нет записи состояния	(НЗС)

Поле "ВА" – запись в регистр виртуального адреса (РВА)

VA (МК I2) = 0 – адрес в фазе чтения

VA (МК I2) = I – адрес в фазе записи.

Если поле VA (МК I2):=0 в одноадресных микроинструкциях циклических сдвигов, то сдвиг осуществляется минуя разряд "C" в регистре состояния процессора (РСП) (см. рис. 43).

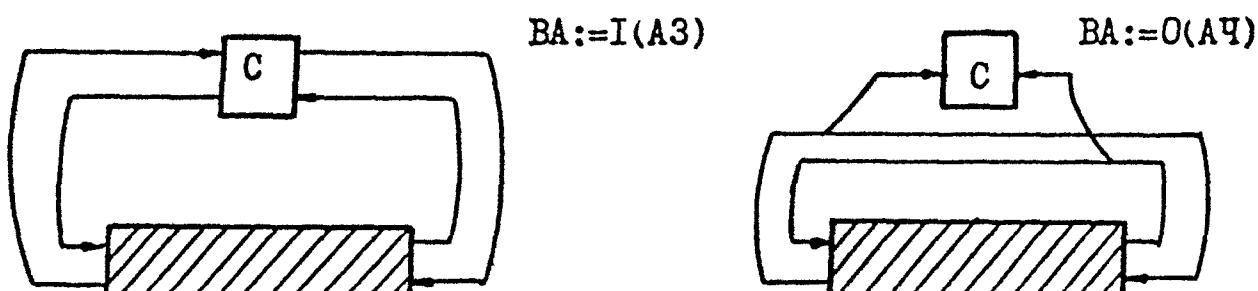


Рис. 43.

Однако, при любых микрооперациях сдвигов, арифметических либо циклических, выдвигаемый разряд заносится в "C" РСП. Кроме того, необходимо учесть, что при арифметическом сдвиге влево, если поле VA:=0, то данная микрооперация трансформируется в циклический левый сдвиг.

Поле "Н0" – направление обмена

Н0		Тип обмена
Разряды МК		
II	I0	
0	0	Обмен с основной памятью (ПМ)
0	I	Обмен с памятью с перехватом в БРА (ПП)
I	0	Обмен с системными регистрами (СР)
I	I	Обмен с системной памятью (СП)

Поле "Ф0" – формата обмена

Ф0		Тип формата обмена
Разряды МК		
9	8	
0	0	Длинное слово обмена (ДСО)
0	I	Слово обмена (СЛО)
I	I	Байт обмена (БО)
I	0	Четверное слово обмена (ЧСО)

1) Если в микрокоманде поле "Ф0" ($\text{МК} <9,8>$) : = $\langle I,0 \rangle$, то за одну такую микрокоманду можно осуществить сложение/вычитание с переносом (бит "С" РСЛ), используя выходной признак CARRY $\langle == \rangle$ $\text{МК} <9,8> := \langle I,0 \rangle$ (см. синтаксические таблицы микроассемблера). Сложение/вычитание с переносом блокируется, если в данной микрокоманде запрещен обмен.

2) Если в микрокоманде поле "Ф0" ($\text{МК} <9,8>$) : = $\langle I,0 \rangle$, то при использовании микроопераций сдвигов влево (АЛ, ЦЛ) в специальном флагке запоминается признак "V" – переполнение. Логика установления данного флагка такова: если любой из выдвигаемых

(на место знакового) разрядов не совпадает со знаковым разрядом, то $V := I$, иначе $V := 0$.

По флагу записи состояния соответствующим образом (в соответствии с данным флагом) устанавливается бит "V" в РСП.

Накопленный признак "V" (флаг переполнения) сбрасывается по концу команды (КК). В противном случае (если поле "Ф0" ($MK <9,8>$) : $\neq <I,0>$) блокируется установка "V" при сдвигах.

Рассмотрим теперь поле кода обмена ("КОБ") в формате микрокоманды. Для осуществления двойных обменов (обмены 64-разрядными данными) необходимо использовать формат обмена ("Ф0") ЧСО ($MK <9,8>$: = $<I,0>$).

Поле "КОБ" – кода обмена

КОБ			Тип обмена	
Разряды МК				
7	6	5		
0	0	0	Нет обмена (НОБ)	
0	0	I	Чтение данных (ЧД)	
0	I	0	Запись данных (ЗД)	
0	I	I	Чтение-модификация-запись (ЧМЗ)	
I	0	0	Запись в регистр-аккумулятор R(I7) (WDAK)	
I	0	I	Чтение команды (ЧК)	
I	I	0	Резерв	
I	I	I	Запись смещения в БРД (ЗпЧТ)	

Процедура чтение-модификация-запись (ЧМЗ) для работы с 64-разрядными данными имеет свои особенности, а именно:

ЧМЗ трансформируется в чтение 64-разрядных данных, а адрес записи аппаратно помещается в регистр-аккумулятор процессора R(I7) и для завершения процедуры ЧМЗ необходимо проинициировать запись по адресу, находящемуся в R(I7), т.е. тип обмена ЧМЗ 64-разрядными данными фактически не является неделимой операцией.

Разряд (МКО) служит для останова выборки микрокоманд из ПЗУ микрокода (S -бит). Если в микрокоманде $MKO:=I$, то после выборки этой микрокоманды выборка микрокоманд из ПЗУ приостанавливается до следующей записи в регистр адреса ПЗУ, т.е. если установлен S -бит и заполнены хотя бы две ячейки буфера команд (готовность буфера команд), то блок синхронизации дешифратора команд (БС ПЛМ ДШК) запускает новый цикл ПЛМ.

Разряд (МК 4) – признак приема данных – ПД; используется при чтении с буферного регистра данных (БРД). ПД : = I в случае использования в переменных полях ПХ, ПУ микрокоманды номера регистра – ПР (БРД). ($PD:=I$ – признак продвижения БРД).

Разряды (МК 1-3) управляют выдачей переменных полей микрокоманды. Если признак равен I, то соответствующее поле микрокоманды выдается не из буфера ПЗУ, а из соответствующего регистра переменного поля, который загружается при дешифрации команды.

Признаки: $PX:=I$ – заменить переменное поле X;

$PU:=I$	<i>то же</i>	у;
$PZ:=I$	$"$	Z;
$PO:=I$	$"$	обмена от ПЛМ.

Рассмотрим формат микрокоманды безусловного ветвления и ветвления по коду прерывания (см.рис. 44)

Формат микрокоманды безусловного ветвления

OTAC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
RME	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
BMC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
форм- мат БП	1	1	1	0	0	0																										
"МК"							1																									

Модификация 6-ти мл.р. адреса перехода из R(18)ОБ - I;
Полн. адрес из "МК" - 0

0 - запоминание адреса перехода в РВ; I - нет

I - призн. перехода по коду прерывания (РПП)

формат "МК" БП (УПР_КП)

- нули

Рис. 44.

Расшифруем значение полей данного формата.

Поле "ФМК" – формат микрокоманды

Разряды МК							Наименование формата микрокоманды	Обобщенное обозначение
Поле "ФМК"			Нули		"Пр"			
31	30	29	28	27	26	25		
I	I	I	0	0	0	0	Безусловное ветвление	БП
I	I	I	0	0	0	I	Ветвление по коду прерывания	УДР_КП

Микрокоманды данного формата в ДБ не передаются (Сигнал ЗПМК – не вырабатывается). Назначение этих микрокоманд – безусловный переход по микроадресу, указанному в разрядах I5-2 микрокоманды. Если разряд модификации микроадреса **MADR** – установлен (МК I6:=1), то адрес микроперехода формируется сле-

дующим образом:

разряды 5-0 адреса заменяются разрядами 5-0 из регистра ОБ с адресом I8, а разряды I3-6 - непосредственно из тела ми - крокоманды. Далее регистр адреса ПЗУ микрокода (РА) загружается адресом перехода.

Далее, если разряд RETURN установлен ($\text{МК } \text{I7}:=0$), то одновременно с загрузкой регистра адреса ПЗУ микрокода в регистре возврата запоминается адрес микрокоманды, следующей за микрокомандой безусловного перехода. Это предусматривает использование механизма микроподпрограмм с единичным уровнем вложенности. Возврат из микропрограммы осуществляется микроинструкцией типа RTS, которая будет описана ниже. Если же $\text{МК } \text{I7}:=1$, то в регистр возврата записи не происходит.

Тот факт, что разряд МК 25:=1 указывает на то, что в качестве индикатора младшей части микроадреса используется четырехразрядный код из регистра - кода прерываний (РКП). Четвертый и пятый разряды адреса микрокоманды - нули. Разряды I3-6 - из тела микрокоманды, т.е. в этом случае осуществляется переход по регистру кода прерываний (УПР _ КП).

Рассмотрим формат микрокоманды перехода по условиям (см. рис. 45). Поле "ФМК": МК <31, 30, 29> := <1, 1, 0>

Формат микрокоманды условного перехода (УП)

OTAC:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
PMC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
BMC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
формат МК "УП"	1	1	0	§	N	Z	V	C	FP	FPD	PXB	PST	RD	RETUR	*	13	Адрес перехода в М17												0		S	

Рис. 45,

При выборке микрокоманды перехода по условию в ОБ не выполняются никаких содержательных операций. Происходит только анализ признаков N , Z , V , C , вырабатываемых в ОБ, и анализ некоторых флагов состояния и выработка сигнала ветвления (\overline{BR}). Сигнал \overline{BR} вырабатывается схемой образования условия ветвления и затем поступает на блок управления приемом и обработкой микрокоманд (БПМК), где и формируется сигнал СВМС, который настраивает мультиплексор регистра адреса пуска ПЗУ-микрокода на определенное направление. После выдачи микрокоманды условного перехода она тормозится до появления сигнала готовности ОБ. Если при появлении сигнала готовности ОБ сигнал \overline{BR} переходит в низкий уровень, то следующая микрокоманда выбирается по адресу, указанному в поле микрокоманды (МК 15-2) и далее из адресов, следующих за этим адресом по порядку. Если сигнал \overline{BR} при появлении сигнала готовности ОБ остается на высоком уровне, то следующая микрокоманда выбирается по адресу, следующему за микрокомандой ветвления.

Приведем расшифровку разрядов в данном формате микрокоманды: "УВ" (МК 28):=I/0 – условие ветвления по признакам и флагам;

"N" (МК 27):=I – ветвление по признаку отрицательного результата;

"Z" (МК 26):=I – ветвление по признаку нулевого результата;

"V" (МК 25):= I – ветвление по признаку переполнения;

"C" (МК 24):=I – ветвление по признаку переноса.

Т.е. коды условий (N, Z, V, C) задаются в поле признаков (МК 27-24) установкой в I соответствующего разряда. Если единицами заданы сразу несколько признаков, то они объединяются по схеме "ИЛИ" при анализе, если разряд УВ:=I (если разряд УВ:=0, то результатирующее условие ветвления формируется по конъюнкции).

- "FP" (МК 23):=I - ветвление по универсальному флагу перехода в микропрограмме (Установка/сброс флага перехода реализуется микропрограммно при помощи микрокоманды "SRF", которая будет описана ниже);
- "FPD" (МК 22):=I - ветвление по флагу разрешения прерывания в "длинных" командах (*First Part Done*);
- "PRB" (МК 21):=I - ветвление по флагу PRB для команды PROBEX (флаг PRB взводится аппаратурой диспетчера памяти процессора, если запрещена процедура преобразования адреса);
- "RST" (МК 20):=I - ветвление по флагу рестарта;
- "RD" (МК I9):=I - ветвление по признаку "регистр-приемник" (Признак "RD" взводится ПЛМ-ДИК при дешифрации команды и используется также для модификации адреса микропрограммы содержательной обработки);
- "RETURN" (МК I7):=I - нет записи в регистр возврата;
 (МК I7):=0 - запись в регистр возврата (Подробнее см. описание формата микрокоманды безусловного ветвления);
- "XC" (МК I6):=I - признак "ждать состояния" кодов условий N, Z, V, C.

Т.е. поле "XC" (МК I6) задает режим анализа. Если XC:=0, то анализируются те признаки, которые хранятся в РСП в момент приема микрокоманды ветвления. Если XC:=I, то анализируются признаки, выработанные в предыдущей микрокоманде, независимо от того, за-писываются они в РСП или нет (признаки берутся из схемы формиро-

вания состояния). В микрокомандах условного ветвления нет возможности модификации младшей части микроадреса в отличии от микропрограмм безусловного перехода и некоторых других, которые будут описаны ниже.

Рассмотрим остальные форматы микрокоманд (см.рис. 4б).

Форматы микрокоманд: ЗСЧ, ПСЧ, NOP, RTS , ACC, ЧКА,
ЧЛК, (KA), SRF, FLAG, REST

OTAC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
RMC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
BMC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ЗСЧ	1	1	1	0	1	1																											
NOP	1	1	1	0	1	1																											
ПСЧ	1	1	1	0	0	1																											
RTS	1	1	1	1	0	0																											
ACC	1	1	1	0	1	0																											
ЧКА	1	1	1	1	0	1																											
ЧЛК	1	1	1	1	1	0	23	Литера ПЗУ (Const 24 разр.)																									
BRF	1	1	1	1	1	1																											
FLAG	0	0	0	0	0	0	1																										
REST	0	0	0	0	0	1	0	1																									
KA	Константа адресная KA=KI {поле I - 32 разряда}																																

Рис. 4б.

Для организации управления циклами в микропрограмме слушат микрокоманды форматов ЗСЧ (загрузка микропрограммного счетчика) и ПСЧ (ветвление по счетчику). Микропрограммный счетчик находится в блоке управления ПЗУ микрокода. Этот счетчик загружается микрокомандой формата ЗСЧ (см. рис. 4б). После выборки микрокоманды этого формата в счетчик загружается константа, содержащаяся в разрядах 9-2 микрокоманды. Если разряд модификации константы установлен, т.е. МСС (МК I6):=I, то разряды 5-0 константы заменяются разрядами 5-0 регистра R(18)ОБ. Разряды 7-6 константы берутся непосредственно из микрокоманды.

При выборке микрокоманды ветвления по счетчику (ПСЧ) анализируется состояние счетчика. После анализа состояния счетчика его содержимое декрементируется на единицу. Если содержимое счетчика не равно нулю при анализе, то происходит ветвление по адресу, указанному в разрядах I5-2 микрокоманды. При этом, если разряд модификации адреса микроперехода установлен, т.е. MADP (МК I6):=I, то разряды 5-0 микроадреса перехода заменяются разрядами 5-0 регистра R (I8)ОБ. Разряды микроадреса I3-6 берутся непосредственно из микрокоманды. Если содержимое счетчика равно нулю при анализе, то выбирается следующая микрокоманда.

При выборке микрокоманды NOP (см.рис.4б) не производится никаких содержательных операций (пустой заголовок в микро-программе).

Микрокоманды формата АСС (см.рис. 4б) - микрокоманды, выполняющие аппаратным ускорителем команд расширенной арифметики и плавающей точки (формат АСС будет раскрыт в другом источнике – отчет по НИР "Электроника 32 С"). После выборки микрокоманд этого формата в блоке управления приемом и обработкой микроко-

манд вырабатывается сигнал торможения приема микрокоманды (блокируется сигнал **WMCE**). Это торможение сбрасывается внешним сигналом готовности ускорителя (**ACCE**).

После выборки микрокоманды формата **RTS** (см.рис. 4б) – возврат из микроподпрограммы, в регистр адреса ПЗУ – микрокода загружается содержимое регистра возврата (РВ), т.е. адрес микрокоманды, следующей за последней выполняемой микрокомандой безусловного/условного ветвления.

После выборки микрокоманды формата ЧКА (см.рис. 4б) по адресу, указанному в разрядах I5-2 микрокоманды читается 32-разрядная адресная константа и помещается в регистре констант ПЗУ (RC), который доступен микропрограммно по адресу, указанному по полю чтения константы (поле "Y" в одноадресном формате). Если разряд модификации адреса константы установлен, т.е. **MnDR** (МК I6):=I, то адрес константы формируется следующим образом:

разряды 5-0 адреса заменяются разрядами 5-0 регистра I8 ОБ;
разряды I3-6 берутся непосредственно из микрокоманды.

После выборки микрокоманды формата ЧЛК (см.рис. 4б) в регистр констант <RC> (в разряды 23-0) загружается содержимое микрокоманды из разрядов 25-2. Разряды 3I-24 <RC> загружаются нулями, если признак СБ:=0 (МК I), и единицами, если СБ:=I.

Регистр констант доступен микропрограммно по адресу, указанному по полю чтения константы. Для чтения литеральных и адресных констант используется адрес <6> в блоке констант (БК).

Микрокоманды формата **SRF** (см.рис. 4б) используются для установки/сброса некоторых функциональных флагов:

Наименование флага (вых. признак)	Состояние	Назначение
WTMM	МК 25:=I	Устанавливает торможение по приему микрокоманды (блокирует WMCE) до конца работы диспетчера памяти
SFM	МК 21:=I	Установка универсального флага перехода
RFM	МК 20:=I	Сброс универсального флага перехода
ИНК	МК I:=I	Инкремент номера регистра
ДЕК	МК 5:=I	Декремент номера регистра
ОСТ	МК 4:=I	Сброс конвейера приема команд, буфера команд, останов ДШК
ЖДЗРА	МК 2:=I	Устанавливает торможение по приему микрокоманды (блокирует WMCE) до конца выполнения микрокоманды загрузки регистра адреса (R(I8)OB).

При использовании микроинструкций ветвлений и некоторых других допускающих модификацию младшей части микроадреса (УП, ЧКА, БП, ПСЧ), а также микроинструкций, допускающих модификацию младшей части константы, загружаемой в микропрограммный счетчик, необходимо загрузить регистр-модификатор адреса OB, который имеет адрес записи $(I \cdot 100X)_2$, т.е. $I8_{16}$ или $I9_{16}$. Флагок торможения ЖДЗРА сбрасывается по факту записи в

$R(I9)OB$, т.е. по окончании выполнения данной микроинструкции OB. Поясним алгоритм установки/адреса такого типа торможения по приему микрокоманды следующей временной диаграммой (см. рис. 47 ·). Временная диаграмма установки/сброса торможения по флагу ЖДЗРА

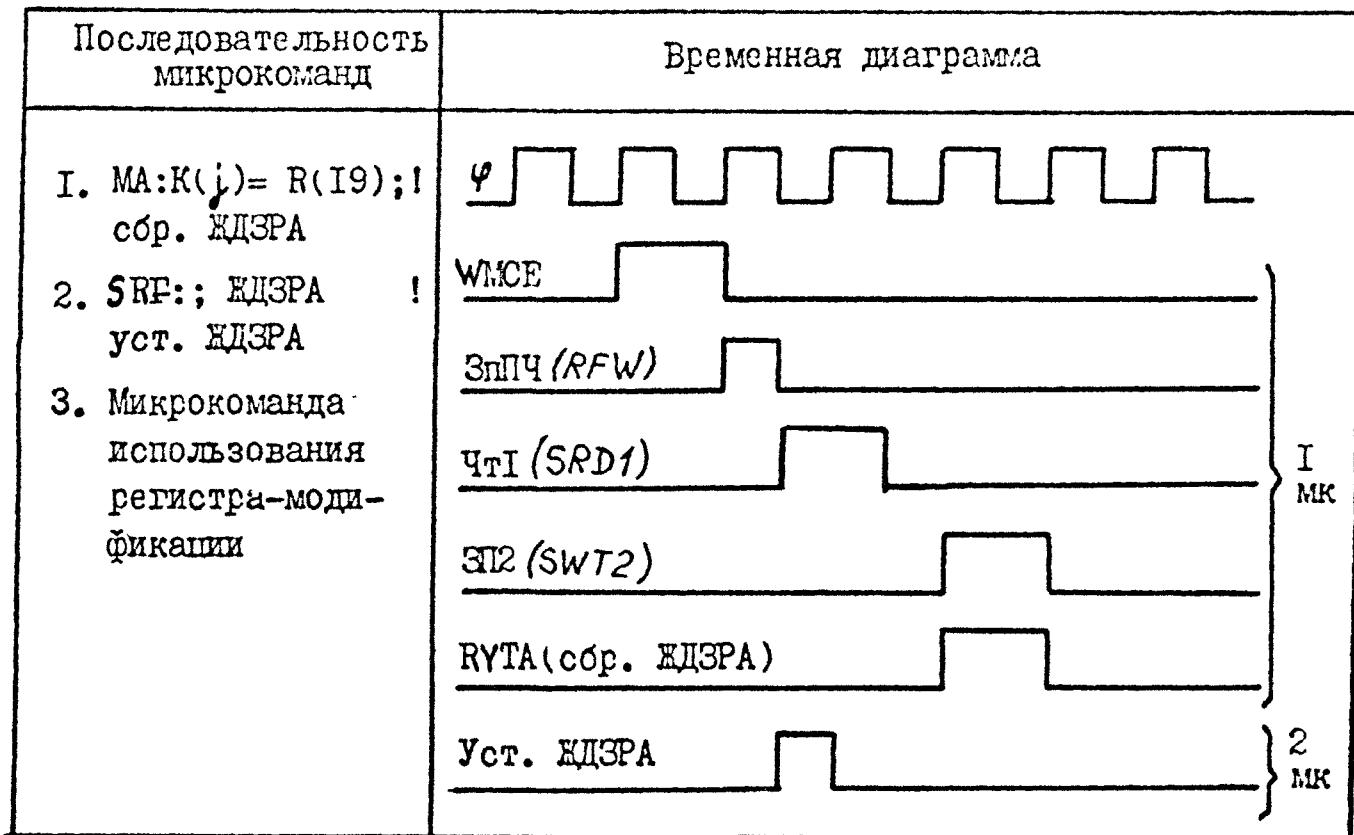


Рис. 47,

Кроме того регистр-модификатор $R(I8)OB$ может быть использован в качестве обычного аккумулятора; нулевой разряд $R(I8)OB$ можно прочитать по адресу OF_{I6} в блоке констант, но на месте второго разряда:



В отличие от формата SRF , формат микрокоманд $FLAG$ (см. рис. 46) служит для установки некоторых функциональных флагов, привязанных к числу операционного блока (OB):

Наименование флага (вых.признаки)	Состояние	Назначение
INTMS	МК 23:=I	Подсветка микропрограммы обработки прерываний
RCPU	МК 22:=I	Микропрограммный сброс процессора
SIVV	МК 21:=I	Блокировка целочисленного переполнения (сброс по "КК")
CCFPU	МК 20:=I	Принять состояние от аппаратного ускорителя команд расширенной арифметики и плавающей точки
PROB	МК 19:=I	Флаг для команды PROBE
SIU	МК 17:=I	Микропрограммный опрос блока прерываний (БПР)
TOU	МК 16:=I	Торможение операционного блока (ОБ) до конца работы диспетчера памяти (ДП)
COPWT	МК 15:=I	Перенос копий РСП и СК

INTMS – устанавливает два аппаратных флага:

INVRD – чтение вектора прерываний по физическому адресу. Флаг **INVRD** – сбрасывается первой одноадресной микроинструкцией.

SKF – признак использования стека ядра/прерыва- ний. Флаг **SKF** – сбрасывается по концу команда (КК).

Микропрограммный сброс процессора по флагу **RCPU** осущест- вляется по следующей схеме:

....; НВ:; A=RESET	Безусловный переход с загрузкой регистра возврата
.....	
RESET; FLAG:; RCPU	Сброс ПРЦ, выход на прерывание
.....	
A=(2100); УПР_КП:; A=	Ветвление по коду прерывания
= (1000)	
A=(1003); RTS ::	Возврат

По факту установки флага PROB обеспечивается холостой цикл диспетчера памяти для пробирования адреса . Если адрес недоступен, то аппаратура диспетчера памяти взводит флагок PRB, по которому производится ветвление.

По факту установки флага COPWT (и по "КК") обеспечивается аппаратная перепись копий РСП и СК:

РНК [R(IC)] $\xrightarrow[\text{переход}]{\text{копия}} \text{R(OF)},$
 КРСП [R(ID)] $\xrightarrow[\text{переход}]{\text{копия}} \text{R(IE)}.$

При выборке микрокоманды REST (см. рис. 4б) происходит восстановление РОНа из стека-рестарта и декрементируется счетчик рестарта (Запись с стека рестарта происходит аппаратно в фазе адресации операндов, при использовании одноадресной микроинструкции, микроопераций инкремент/декремент, употребления по полю чтения-Х микрокоманды 4-х разрядного номера регистра общего назначения).

Алгоритм восстановления регистров-указателей операндов из стека рестарта приведен на рис. 48 . (глубина стека рестарта равна 6, в соответствии с максимальным количеством операндов в команде).

Алгоритм восстановления регистров-указателей
операндов из стека рестарта

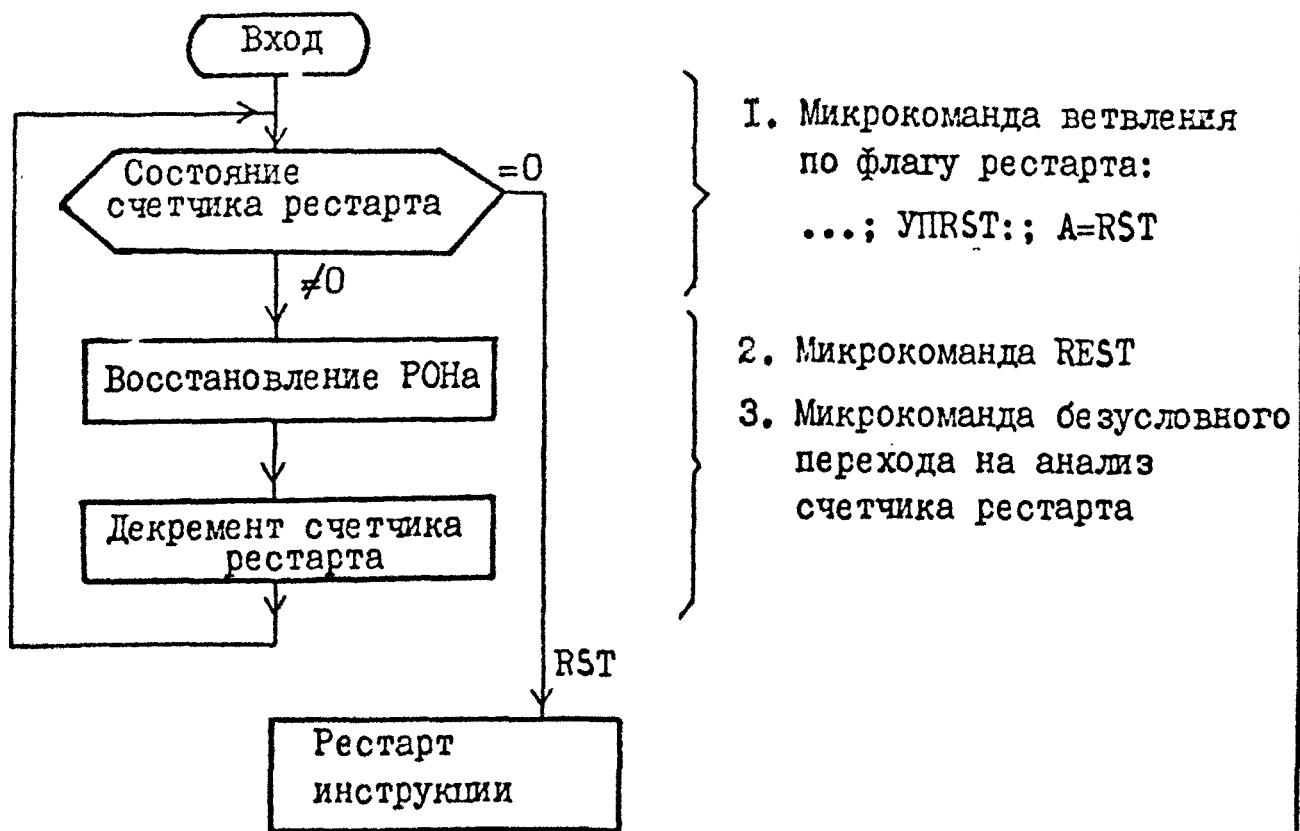


Рис. 48.

I.13. Описание ресурсов микропрограммного уровня процессора

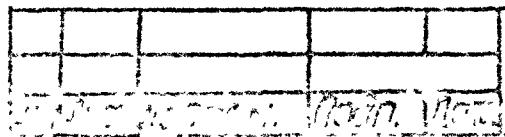
К ресурсам микропрограммиста будем относить такие регистры операционного блока (ОБ), как регистры общего назначения (РОНы), регистр-указатель стекового кадра, регистр-указатель аргумента, регистр-указатель стека, регистры-аккумуляторы, регистры виртуального адреса, счетчик команд, слово состояния процессора, буферный регистр данных и т.д., а также некоторые внутренние регистры процессора. Все регистры, используемые в проекте "Электроника 32", сведены в табл. 6.

Микропрограммно доступен также блок констант операционного блока (БК). Константы ОБ приведены в табл. 7.

Микропрограммно доступна любая ячейка виртуального адресного пространства. Виртуальный адрес содержит 32 разряда и определяет ячейку длиной в байт в адресном пространстве.

Для программиста этот адрес обеспечивает адресацию линейной матрицы объемом в 4 294 967 296 байтов. Виртуальное адресное пространство разделено на страницы по 512байтов каждая. Страница является объектом перемещения и защиты. Виртуальное адресное пространство показано на рис. 49 . Виртуальная память разделена на 2 части. Первая половина, занимающая младшую часть адресов, предназначена для распределения процессов и для каждого процесса, выполняемого в комплексе, она своя. Вторая половина виртуальной памяти, занимающая старшую часть адресов, предназначена для размещения системы, и эта область разделяется всеми процессами.

Область процессов в свою очередь делится на две равные части: программную область (РО – регион) и управляющую область (РІ – регион).



ЛСС.730.1-12



Таблица 6

Регистры процессора

Мнемоника	Адрес(16)	Назначение	Доступ	Сфера
R0-RB	00-03	Регистры общего назначения (РОН)	ч на X, \bar{y} / з с X*, \bar{y}^*	
AP	0C	Указатель аргумента		
FP	0D	Указатель кадра		
SP	0E	Указатель стека		
PC	0F	Счетчик команд		
AK6	I0	Регистры-аккумуляторы	ч на X, \bar{y} / з с X*, \bar{y}^*	ОБ про-цессора
AK4	I1			
AK3	I2			
AK5	I3			
AK0	I4			
AK1	I5			
AK2	I6			
AK7	I8 (адрес записи II $\emptyset\emptyset$ X ₂)			
AK8	I7			
PKA	ID (адрес чтения)	Регистр косвенной адресации	ч на X/з с МД	
PKVA	I9 (адрес чтения)	Регистр копия виртуального адреса	ч на \bar{y} / з с PVA	

Мнемоника	Адрес (16)	Назначение	Доступ	Сфера
PVA	IA(адрес чтения)	Регистр виртуального адреса	ч на X, Y / з с X, Y*, PBAK	
PBAK	IB(адрес чтения) OF(адрес записи)	Регистр виртуального адреса команды	ч на Y / з с X*, Y*	
PHK	IC(адрес чтения)	Регистр начала команды	ч на Y / з с СК	
KRSP	ID(адрес чтения)	Копия регистра состояния процессора	ч на Y / з с РСП	
PSL	IE	Слово состояния процессора	ч на X, Y / з с X*, Y*	OB про-цессора
БО	IF	Буферный регистр данных	ч на X, Y / з с X*, Y*; МД	
<u>Внутренние регистры процессора</u>				
KSP	Ø0	Указатель стека режима ядра	ч/з	
ESP	Ø1	Указатель стека режима исполнения	ч/з	Адаптер Q-BUS
SSP	Ø2	Указатель стека режима супервизора	ч/з	
USP	Ø3	Указатель стека режима пользователя	ч/з	
ISP	Ø4	Указатель стека прерываний	ч/з	(область сист. рег.)
	Ø5...Ø7	Резерв	ч/з	
PØBR	Ø8	Регистр базы PØ	ч/з	

Продолжение табл. 6

Мнемоника	Адрес (16)	Назначение	Доступ	Сфера
POLR	09	Регистр длины РО	ч/з	Диспетчер памяти процессора
PIBR	0A	Регистр базы PI	ч/з	
PILR	0B	Регистры длины PI	ч/з	
SBR	0C	Регистр базы системы	ч/з	
SLR	0D	Регистр длины системы	ч/з	
	0E..0F	Резерв	ч/з	
PCBB	I0	Регистр базы блока управления процесса	ч/з	Адаптер Q-BUS
SCBB	I1	Регистр базы блока управления системы	ч/з	Адаптер Q-BUS (обл.сист.рег.)
IPL	I2	Уровень приоритета прерывания	ч/з	Процессор (разряды 20+16 PSL)
ASTR	I3	Уровень AST	ч/з	
SIRR	I4	Регистр запроса на программное прерывание	з	Адаптер Q-BUS
SISR	I5	Регистр программного прерывания	ч/з	(обл.сист.рег.)
	I6	Резерв		
MCSR	I7	Регистр состояния сбоя машины	ч	Контроллер памяти

Продолжение табл. 6.

Мнемоника	Адрес (16)	Назначение	Доступ	Сфера
ICCS	I8	Регистр состояния таймера	ч/з	
NICR	I9	Регистр счета следующего интервала	з	Адаптер Q-BUS
ICR	IA	Регистр-счетчик интервалов	ч	(обл. сист. рег.)
TODR	IB	Регистр времени дня	ч/з	
	IC+IF	Резерв		
RXCS	20	Регистр состояния консольного приемника	ч/з	
RXDB	21	Регистр данных консольного приемника	ч	
TXCS	22	Регистр состояния консольного передатчика	ч/з	
TXDB	23	Регистр данных консольного передатчика	з	
TEDR	24	Отключение групп буфера трансляции(при обращении читаются нули)	ч/з	
CADR	25	Регистр отключения КЭШ-памяти	ч/з	Контроллер памяти
MCESR	26	Регистр ошибок при машинном сбое	ч/з	
CAER	27	Регистр ошибок КЭШ-памяти	ч/з	

Продолжение табл. 6

Мнемоника	Адрес I(6)	Назначение	Доступ	Сфера
ACCS	28	Регистр состояния ускорителя команд П.З.	ч	БИС со - процес - сора
	29...36	Резерв		
IORESET	37	Регистр сброса UNIBUS	з	Адаптер Q-BUS
MME (MAPEN)	38	Регистр включения диспетчера памяти	ч/з	Диспетчер памяти про-цессора
TBIA	39	Регистр обнуления всего буфера трансляции	з	Диспетчер памяти процессора
TBIS	3A	Регистр обнуления записи в буфере трансляции	з	Диспетчер памяти процессора
	3B...3C	Резерв		
PMC	3D	Регистр включения исполнительного монитора	ч/з	Адаптер Q-BUS
SID	3E	Регистр идентификации системы	ч/з	Процессор
		Резерв		

Таблица 7

Константы ОБ

Адрес (I6)	Значение (I6)	Назначение
00	00000004	Инкремент
01	00000002	Инкремент
02	00000008	Инкремент
03	00000001	Инкремент
04	00000000	Обнуление
05	-	PCM
06	-	Литера ПЗУ, адресные константы
07	00000066	Константа перевода в код с избыtkом 6
08	0000000*	$[1:0] = \text{PCM}$ CRM [27-19] (Чтение CRM из PCM)
09	00000000	Маски CRM в PCM
0A	FFFFFFFFFF	
0B	001F0000	Маски IPL в PCM
0C	04000000	Маски IS в PCM
0D	0000000*	$[2:0] = \text{MPL}$
0E	000000**	Десятичная коррекция $[7:0] = \begin{cases} AA, A0 \\ OA, OO \end{cases}$
0F	0000000*	второй разряд = нулевой разряд RI8 выбор стека при прерываниях

Виртуальное адресное пространство

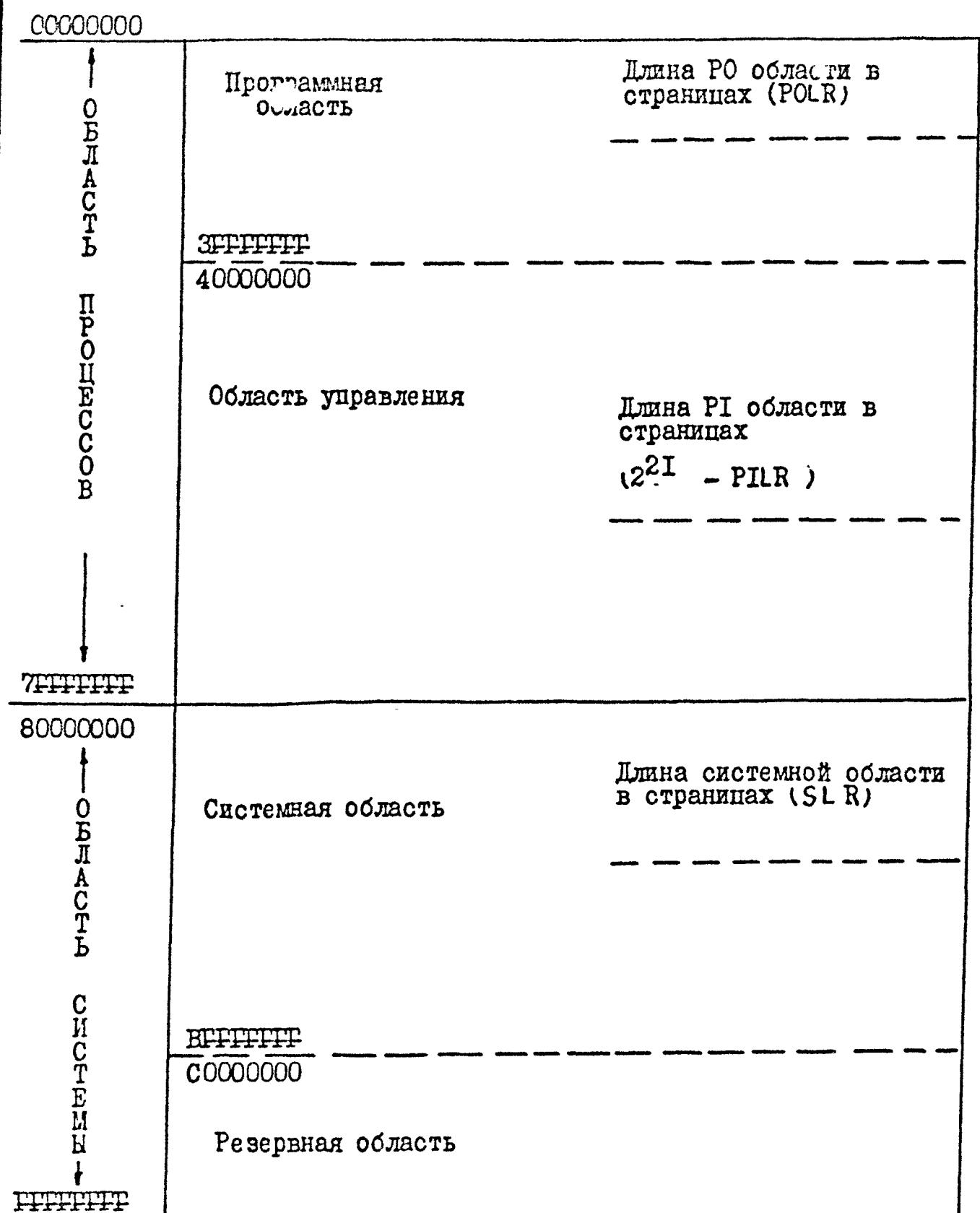


Рис. 49.

Для организации системного ОЗУ (СОЗУ), некоторых внутренних системных регистров, хранения карты адресов *Q-BUS* БИС адаптера магистралей обеспечивает управление (чтение, запись) четырех микросхем статического ЗУ с организацией 2Kx8.

Системное ОЗУ и системные регистры доступны микропрограммно.

I.I4. Система команд

В данном разделе очень кратко будут описаны те инструкции, которые используются в любом программном обеспечении при всех применениях вычислительной системы "Электроника 32". Главная цель архитектуры вычислительной системы - обеспечить набор команд, который симметричен по отношению к типам данных. Например, существуют команды сложения (*ADD*) для семи типов данных (байт, слово, длинное слово, *F*, *D*, *G* и *H-floating*), причем каждая команда имеет варианты для двух и трех операндов. Симметричными командами являются также команды пересылки, преобразования тестирования и вычислений (в микропроцессорной системе "Электроника 32" команды работы со 128-разрядными данными, такими как *H-floating* и *Octal integer*, выполняются эмуляцией на программном уровне).

В число команд пересылок входят также команды пересылки с арифметическим отрицанием и логическим дополнением. Здесь же команды отрицания и дополнения включают в себя пересылку и не изменяют исходный операнд.

Команды арифметических вычислений включают в себя сложение, вычитание, умножение и деление. Логические команды работают с тремя типами данных (байт, слово, длинное слово) и включают в себя: установку и сброс битов, логическое сложение MOD 2. Команды арифметических и логических вычислений доступны как в двух, так и в трехоперандной формах для каждого допустимого типа дан-

ных. Трехоперандная форма принимает первые два операнда и сохраняет результат по адресу третьего операнда.

Для задач целочисленного программирования имеется команда записи в стек длинного слова.

Для каждого типа целочисленных данных существуют операции инкремента и декремента.

Существуют специальные команды сложения, вычитания, умножения и деления с многократной точностью.

Для реализации математических функций введены команды вычисления показателей степени и полиномов.

В систему команд входят также команды обработки чисел в формате с плавающей запятой. Команды с плавающей запятой работают с данными четырех типов (*F*, *D*, *G* и *H*-формат). Для всех четырех типов данных с плавающей запятой имеются стандартные арифметические операции сложения, вычитания, умножения и деления (*ADD*, *SUB*, *MUL*, *DIV*). Кроме того имеются две составные операции *EMOD* и *POLY*, также применимые ко всем четырем типам данных с плавающей запятой. *EMOD* вырабатывает произведение двух операндов, а затем разделяет результат на целую и дробную части. *POLY* подсчитывает значение полинома по заданной степени, аргументу и указателю таблицы коэффициентов.

Имеется полный набор команд преобразования из целочисленного типа данных во все форматы с плавающей запятой, и наоборот. Также имеется набор команд преобразования из одного типа данных с плавающей запятой в другой, за исключением преобразований из формата **D** в формат **G** и наоборот.

Для всех типов данных с плавающей запятой существует класс команд пересылок (*MOV*, *MNEG*), очистки (*CLR*), сравнения (*CMP*) и тестирования (*TST*). И, наконец, существует класс команд *ACB* (сложение, сравнение и переход).

STRUCT NODORUM. Noch. 112

В процессоре имеется широкий набор команд ветвлений и переходов (команды управления). Довольно часто требуется передача управления, основанная на значении переменной, для поддержки таких конструкций в языках высокого уровня как "GOTO" в языке *FORTRAN* и "CASE" в языке *PASCAL*. Вследствии этого система команд включает в себя команду *CASE*, позволяющую эффективно реализовать эти структуры управления. Команда *CASE* не только производит передачу управления, но также управляет инициализацией и контролем границ для переменной типа *INDEX*. Цель команды *CASE* состоит в передаче управления одной из *n* ячеек на основании значения целочисленного операнда-селектора.

Существует также ряд важных команд, предназначенных для работы со специфическими типами данных, такими как поле бит переменной длины, символьная строка, строка десятичных цифр, строка упакованных десятичных цифр, очередь (типы данных приведены на рис. 50).

В системе команд имеется несколько категорий команд, которые обладают в точно определенных условиях большими привилегиями, чем обычное программное обеспечение.

Это привилегированные команды и особые (специальные) команды, которые не попадают под какую-либо классификацию. Некоторые из этих инструкций предназначены для работы со специализированной частью архитектуры вычислительной системы (например: диспетчеризация памяти, прерывания и исключительные ситуации, диспетчеризация процессов, внутренние регистры процессора).

Для наглядности представления системы команд, на основании изложенного выше, разделим набор инструкций на 13 основных разделов:

I) Команды для работы с целыми числами и логические команды.

Типы данных

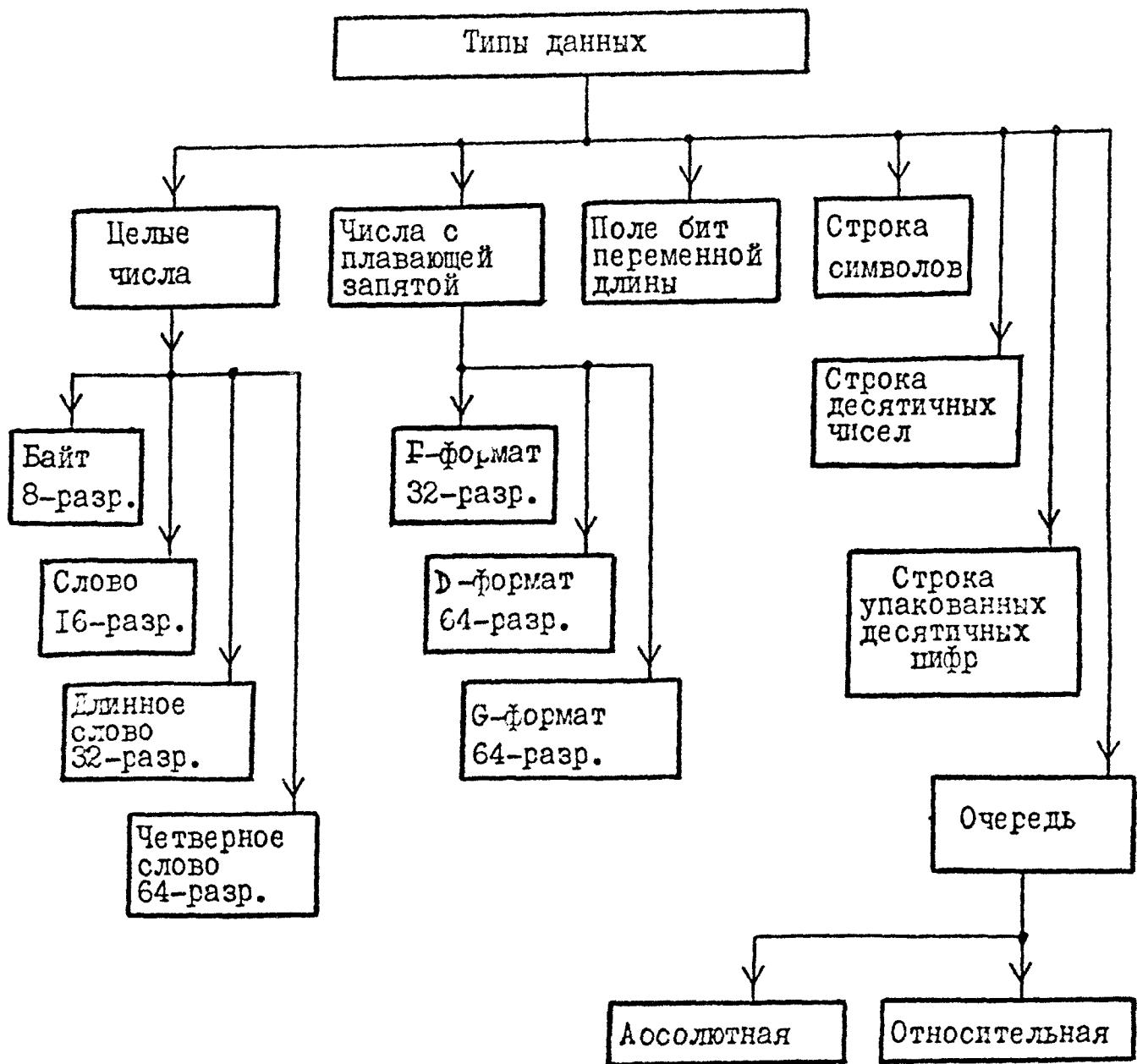


Рис.50.

- 2) Команды для работы со стеками и адресами памяти.
- 3) Команды управления.
- 4) Команды для работы с переменными битовыми полями.
- 5) Команды вызова процедур и подпрограмм.
- 6) Привилегированные команды.
- 7) Специальные команды.
- 8) Команды работы с очередями.
- 9) Команды плавающей арифметики.
- 10) Команды обработки строк символов.
- 11) Команды десятичной строки цифр.
- 12) Команды контроля циклической избыточности.
- 13) Команды редактирования.

Ниже будет приведено краткое описание инструкций (см. табл. 8).

Внутри каждого раздела близкие по свойствам инструкции объединены в группы и вместе описываются. Описание групп инструкций состоит из следующих пунктов: название группы; код команды; формат каждой инструкции в группе.

В табл.8 приводятся мнемоника команды, определяются название и тип каждого спецификатора операнда инструкции и порядок, в котором они появляются в инструкции.

СИСТЕМА КОМАНД

Код	Мнемоника	Формат	Назначение
I. Команды для работы с целыми числами и логические команды			
58	ADAWI	КОП add.rw,sum.mw	Сложение выровненных слоев с блокировкой памяти
80	ADD82	КОП add.rx,sum.mx	Сложение байтов двух - операндное
A0	ADDW2	"	Сложение слов двухоперандное
C0	ADDL2	"	Сложение двойных слов двухоперандное
81	ADD83	КОП add1.rx,add2.rx,sum.mx	Сложение байтов трех - операндное
A1	ADDW3	"	Сложение слов трехоперандное
C1	ADDL3	"	Сложение двойных слов трехоперандное
D8	ADWC	КОП add.r ^b ,sum.m ^b	Сложение с переносом
78	ASHL	КОП cnt.r ^b ,src.rx,dst.wx	Арифметический сдвиг двойного слова
79	ASHQ	"	Арифметический сдвиг квадрослова
8A	BICB2	КОП mask.rx,dst.mx	Очистка разрядов байта двухоперандная
AA	BICW2	"	Очистка разрядов слова двухоперандная
CA	BICL2	"	Очистка разрядов двойного слова двухоперандная

Код	Мнемоника	Формат	Назначение
8B	<i>BICB3</i>	КОП mask. <i>xx</i> , <i>sfc.</i> <i>xx</i> , <i>dst.</i> <i>mx</i>	Очистка разрядов байта трехоперандная
AB	<i>BICW3</i>	-"-	Очистка разрядов слова трехоперандная
CB	<i>BICL3</i>	-"-	Очистка разрядов двойно- го слова трехоперандная
88	<i>BISB2</i>	КОП <i>mask.</i> <i>xx</i> , <i>dst.</i> <i>mx</i>	Установка разрядов байта двухоперандная
A8	<i>BISW2</i>	-"-	Установка разрядов слова двухоперандная
C8	<i>BISL2</i>	-"-	Установка разрядов двойно- го слова двухоперандная
89	<i>BISB3</i>	КОП <i>mask.</i> <i>xx</i> , <i>sfc.</i> <i>xx</i> , <i>dst.</i> <i>mx</i>	Установка разрядов байта трехоперандная
A9	<i>BISW3</i>	-"-	Установка разрядов слова трехоперандная
C9	<i>BISL3</i>	-"-	Установка разрядов двой- ного слова трехоперандная
93	<i>BITB</i>	КОП <i>mask.</i> <i>xx</i> , <i>sfc.</i> <i>xx</i>	Проверка разрядов байта
B3	<i>BITW</i>	-"-	Проверка разрядов слова
D3	<i>BITL</i>	-"-	Проверка разрядов двойно- го слова
94	<i>CLRB</i>	КОП <i>dst.</i> <i>wx</i>	Очистка байта
B4	<i>CLRW</i>	-"-	Очистка слова
D4	<i>CLRL</i>	-"-	Очистка двойного слова
7C	<i>CLRQ</i>	-"-	Очистка квадрослова
9I	<i>CMPB</i>	КОП <i>sfc1.</i> <i>xx</i> , <i>sfc2.</i> <i>xx</i>	Сравнение байтов
BI	<i>CMPW</i>	-"-	Сравнение слов
DI	<i>Cmpl</i>	-"-	Сравнение двойных слов

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
99	<i>CVTBW</i>	KOP <i>src.rx, dst.wy</i>	Преобразование байта в слово
98	<i>CVTBL</i>	"	Преобразование байта в двойное слово
33	<i>CVTWB</i>	"	Преобразование слова в байт
32	<i>CVTWL</i>	"	Преобразование слова в двойное слово
F6	<i>CVTLB</i>	"	Преобразование двойного слова в байт
F7	<i>CVTLW</i>	"	Преобразование двойного слова в слово
97	<i>DEC8</i>	KOP <i>dif.mx</i>	Декремент на один байт
B7	<i>DECW</i>	"	Декремент на одно слово
D7	<i>DECL</i>	"	Декремент на одно двойное слово
86	<i>DIVB2</i>	KOP <i>divr.rx, quo.mx</i>	Деление байтов двухоперандное
A6	<i>DIVW2</i>	"	Деление слов двухоперандное
C6	<i>DIVL2</i>	"	Деление двойных слов двухоперандное
87	<i>DIVB3</i>	KOP <i>divr.rx, divd.rx, quo.wx</i>	Деление байтов трехоперандное
A7	<i>DIVW3</i>	"	Деление слов трехоперандное
C7	<i>DIVL3</i>	"	Деление двойных слов трехоперандное
7B	<i>EDIV</i>	KOP <i>divr.r8, divd.r9, quo.w8, tem.w8</i>	Расширенное деление

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
7A	<i>EMUL</i>	<i>KOP t1c.r1, r2, t1d.r2, add.r2, prod.wq</i>	Расширенное умножение
96	<i>INC B</i>	<i>KOP sum.mX</i>	Инкремент на один байт
B6	<i>INC W</i>	<i>-"-</i>	Инкремент на одно слово
D6	<i>INCL</i>	<i>-"-</i>	Инкремент на одно двойное слово
92	<i>MCOMB</i>	<i>KOP src.rX, dst.wX</i>	Пересылка байта с инверсией
B2	<i>MCOMW</i>	<i>-"-</i>	Пересылка слова с инверсией
D2	<i>MCOML</i>	<i>-"-</i>	Пересылка двойного слова с инверсией
8E	<i>MNEG B</i>	<i>-"-</i>	Пересылка байта с арифметическим отрицанием
AE	<i>MNEG W</i>	<i>-"-</i>	Пересылка слова с арифметическим отрицанием
CE	<i>MNEGL</i>	<i>-"-</i>	Пересылка двойного слова с арифметическим отрицанием
90	<i>MOV B</i>	<i>-"-</i>	Пересылка байта
B0	<i>MOV W</i>	<i>-"-</i>	Пересылка слова
D0	<i>MOV L</i>	<i>-"-</i>	Пересылка двойного слова
7D	<i>MOV Q</i>	<i>-"-</i>	Пересылка квадрослова
9B	<i>MOVZBW</i>	<i>KOP src.rX, dst.wY</i>	Пересылка байта с дополнением ведущими нулями до слова
9A	<i>MOVZBL</i>	<i>-"-</i>	Пересылка байта с дополнением ведущими нулями до двойного слова
9C	<i>MOVZWL</i>	<i>-"-</i>	Пересылка слова с дополнением ведущими нулями

Код	Мнемоника	Формат	Назначение
			до двойного слова
84	MULB2	КОП <i>mul.rx, prod.tx</i>	Умножение байтов двухоперандное
A4	MULW2	-"-	Умножение слов двухоперандное
C4	MULL2	-"-	Умножение двойных слов двухоперандное
85	MULB3	КОП <i>mul.rx, muld.wx, prod.wx</i>	Умножение байтов трехоперандное
A5	MULW3	-"-	Умножение слов трехоперандное
C5	MULL3	-"-	Умножение двойных слов трехоперандное
DD	PUSHL	КОП <i>src.rl, {-(SP).wl}</i>	Занесение двойного слова в стек
9C	ROTL	КОП <i>cnt.r6, src.rl, dst.wl</i>	Циклический сдвиг двойного слова
D9	SBWC	КОП <i>sub.rl, dif.m6</i>	Вычитание с переносом
82	SUBB2	КОП <i>sub.rx, dif.tx</i>	Вычитание байтов двухоперандное
A2	SUBW2	-"-	Вычитание слов двухоперандное
C2	SUBL2	-"-	Вычитание двойных слов двухоперандное
83	SUBB3	КОП <i>sub.rx, min.wx, dif.wx</i>	Вычитание байтов трехоперандное
A3	SUBW3	-"-	Вычитание слов трехоперандное

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
C3	SUBL3	КОП <i>sub.rx, t1l.rx, dif.wx</i>	Вычитание двойных слов трехоперандное
95	TSTB	КОП <i>src.rx</i>	Проверка разрядов байта
B5	TSTW	-"-	Проверка разрядов слова
D5	TSTL	-"-	Проверка разрядов двойно- го слова
8C	XORB2	КОП <i>mask.rx, dst.mx</i>	Сложение байтов по MOD2 двухоперандное
AC	XORW2	-"-	Сложение слов по MOD2 двухоперандное
CC	XORL2	-"-	Сложение двойных слов по MOD2 двухоперандное
8D	XORB3	КОП <i>mask.rx, src.rx, dst.wx</i>	Сложение байтов по MOD2 трехоперандное
AD	XORW3	-"-	Сложение слов по MOD2 трехоперандное
CD	XORL3	-"-	Сложение двойных слов по MOD2 трехоперандное

2. Команды работы со стеками и адресами памяти

9E	MOVAB	КОП <i>src.ax, dst.wl</i>	Пересылка адреса байта
3E	MOVAW	-"-	Пересылка адреса слова
DE	MOVAL; MOVAF	-"-	Пересылка адреса двойного слова /или слова формата F/

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
7E	MOVAQ, MOVAD, MOVAG	KOP <i>src.ax, dst.w</i>	Пересылка адреса четвертного слова /или слов форматов D, G /
7FFD	MOVAD, MOVAH	-"-	Пересылка адреса восьмого слова /или слова формата H/
BF	PUSHAB	KOP <i>src.ax, {-(SP).w}</i>	Пересылка адреса байта в стек
3F	PUSHAW	-"-	Пересылка адреса слова в стек
DF	PUSHAL, PUSHAF	-"-	Пересылка адреса двойного слова /или слова формата F / в стек
7F	PUSHAQ, PUSHAD, PUSHAG	-"-	Пересылка адреса четвертного слова /или слов форматов D ; G / в стек
7FFD	PUSHAO, PUSHAH	-"-	Пересылка адреса восьмого слова/или слова формата H/ в стек
3. Команды управления			
9D	ACBB	KOP <i>limit.rx, add rx,</i> <i>index.mx, dispL bw</i>	Сложение байтов со сравнением и переход
3D	ACBW	-"-	Сложение слов со сравнением и переход

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
F1	ACBL	KOP <i>limit.7x, add.7x, index.mx, displ.6w</i>	Сложение двойных слов со сравнением и переход
4F	ACBF	"-	Сложение слов формата F со сравнением и переход
6F	ACBD	"-	Сложение слов формата D со сравнением и переход
4FFD	ACBG	"-	Сложение слов формата G со сравнением и переход
6FFD	ACBH	"-	Сложение слов формата H со сравнением и переход
F3	AOBLEQ	KOP <i>limit.7x, index.ml, displ.6b</i>	Прибавить единицу и переход, если меньше или равно
F2	AOBLS5	"-	Прибавить единицу и переход, если меньше
I9	BLSS	KOP <i>displ.6b</i>	Переход, если N = I
I5	BLEQ	"-	Переход, если {NORZ} = 1
I3	BEQL, BEQLU	"-	Переход, если Z = I
I2	BNEQ, BNEQU	"-	Переход, если Z = 0
I8	BGEQ	"-	Переход, если N = 0
I4	BGTR	"-	Переход, если {NORZ} = 0
IF	BLSSU, BCS	"-	Переход, если C = I
IB	BLEQU	"-	Переход, если {CORZ} = I
IE	BGEQU, BCC	"-	Переход, если C = 0
IA	BGTRU	"-	Переход, если {CORZ} = 0
ID	BVS	"-	Переход, если V = I
IC	BVC	"-	Переход, если V = 0

Продолжение табл. 3

Код	Мнемоника	Формат	Назначение
E0	BBS	KOP pos. $r_1^l, base.vb,$ $disp.l.66, \{field, rv\}$	Ветвление по биту /указанный разряд равен 1/
E1	BBC	-"-	Ветвление по биту /указанный разряд равен 0/
E2	BBS5	-"-	Ветвление по установленному биту и установить /без блокировки памяти/
E4	BBS C	-"-	Ветвление по установленному биту и очистить /без блокировки памяти/
E5	BBC C	-"-	Ветвление по очищенному биту и очистить /без блокировки памяти/
E3	BBC S	-"-	Ветвление по очищенному биту и установить /без блокировки памяти/
E6	BBS5I	-"-	Ветвление по установленному биту и установить с блокировкой памяти
E7	BBC CI	-"-	Ветвление по сброшенному биту и очистить с блокировкой памяти
E8	BLBS	KOP $src.r_1^l, disp.l.66$	Ветвление, если установлен младший бит
E9	BLBC	-"-	Ветвление, если сброшен младший бит
II	BRB	KOP $disp.l.6x$	Безусловный переход с базовым смещением
3I	BRW	-"-	Безусловный переход со смещением в одно слово

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
10	B58B	КОП <i>displ.6x,{-(SP).w}</i>	Переход к подпрограмме с байтовым смещением
30	B5BW	"-	Переход к подпрограмме со смещением в одно слово
8F	CASEB	КОП <i>selector.rr,base,rr,limit.rr,displ.6w-list</i>	Многофункциональный переход с байтовым смещением
AF	CASEW	"-	Многофункциональный переход со смещением в одно слово
CF	CASEL	"-	Многофункциональный переход со смещением в одно двойное слово
I7	JMP	КОП <i>dst.ab</i>	Абсолютный безусловный переход
I6	JSB	КОП <i>dst.ab,{-(SP).w}</i>	Абсолютный переход к подпрограмме
05	RSB	КОП <i>{(SP)+.rl}</i>	Возврат из подпрограммы
F4	S0BGEQ	КОП <i>index.ml,displ.6b</i>	Вычесть единицу и переход, если больше или равно
F5	S0BGTR	"-	Вычесть единицу и переход, если больше

4. Команды для работы с переменными битовыми полями

EC	CMPV	КОП <i>pos.rl,size.rl,base.w,{field.rv},src.rl</i>	Сравнение полей
ED	CMPZV	"-	Сравнение полей с расширением ведущими нулями

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
EE	EXTV	KOP pos.r ^b ,size.r ^b , base.v ^b ,{field.rv},dst.w ^b	Выделить поле
EF	EXTZV	" -	Выделить поле, расширенное ведущими нулями
EA	FFS	KOP startpos.r ^b , size.r ^b ,base.v ^b , {field.rv},findpos.w ^b	Поиск первого установленного разряда
EB	FFC	" -	Поиск первого сброшенного разряда
FO	INSV	KOP src.r ^b ,pos.r ^b , size.r ^b ,base.v ^b , {field.wv}	Вставка поля

5. Команды вызова процедур и подпрограмм

FA	CALLG	KOP arglist.ab, dst.ab,{-(SP).w*}	Вызов процедуры с произвольным списком аргументов
FB	CALLS	KOP numq.r9.r ^b , dst.ab,{-(SP).w*}	Вызов процедуры со списком аргументов в стеке
04	RET	KOP {(SP)+.r*}	Возврат из процедуры

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
6. Привилегированные команды			
OC	PROBER	КОП mode. $r^b, R_{en}.w^b, b_{ase}, a^b$	Проверка доступности по чтению
OD	PROBEW	... -" -	Проверка доступности по записи
BC	CHMK	КОП $r_{param}.w^b, \{-(ySP).w^b\}$, где: $y = MINU/x, PSL <$ текущий реж.>)	Смена режима на режим ядра
BD	CHME	... -" -	Смена режима на режим исполнения
BE	CHMS	-" -	Смена режима на режим супервизора
BF	CHMU	-" -	Смена режима на режим пользователя
02	REI	КОП $\{(SP)+, r^b\}$	Возврат из исключительной ситуации или прерывания
06	LDPCX	КОП $\{PCB.r^b, -(kSP).w^b\}$, где PCB – регистр управления процессором	Загрузка контекста процесса
07	SVPCTX	КОП $\{(SP)+, r^b, PCB.w^b\}$	Сохранение контекста процесса
DA	MTPR	КОП $src.r^b, procreg.r^b$	Пересылка в процессорный регистр
DB	MFPR	КОП $procreg.r^b, dst.w^b$	Пересылка из процессорного регистра

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
	7. Специальные команды		
B9	<i>BICPSW</i>	КОП <i>mask.zw</i>	Сброс разрядов в слове состояния процессора
B8	<i>BISPSW</i>	—“—	Установка разрядов в слове состояния процессора
03	<i>BRT</i>	КОП <i>{-(kSP).w*}</i>	Ошибка ветвления
00	<i>HALT</i>	—“—	Останов
0A	<i>INDEX</i>	КОП <i>subscript.rl,low.rl,high.rl,size.rl,indexin.rl,indexout.wl</i>	Вычисление индекса
DC	<i>MOVPSL</i>	КОП <i>dst.wl</i>	Пересылка из длинного слова состояния процессора
01	<i>NOP</i>	КОП	Пустой оператор
BB	<i>PUSHR</i>	КОП <i>mask.zw,{-(SP).w*}</i>	Пересылка содержимого регистров в стек
BA	<i>POPR</i>	КОП <i>mask.zw,{(SP)+,z*}</i>	Удаление содержимого регистров из стека
FC	<i>XFC</i>	КОП	Расширенный вызов функции
FEFF	<i>BUGW</i>	КОП <i>message.bx</i>	Проверить слово на наличие ошибок
FDFE	<i>BUGL</i>	—“—	Проверить двойное слово на наличие ошибок
	8. Команды работы с очередями		
5C	<i>INSQHI</i>	КОП <i>entry.ab,header.az</i>	Занесение элемента в "голову" очереди с блокировкой памяти

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
5D	INSQTI	КОП <i>entry.a6, header.aq</i>	Занесение элемента в "хвост" очереди с блокировкой памяти
0E	INSQUE	КОП <i>entry.a6, pred.a6</i>	Занесение элемента в очередь
5E	REMQHI	КОП <i>header.aq, addr.w6</i>	Удаление элемента из "головы" очереди с блокировкой памяти
5F	REMQTI	"—"	Удаление элемента из "хвоста" очереди с блокировкой памяти
0F	REMQUE	КОП <i>entry.a6, addr.w6</i>	Удаление элемента из очереди
9. Команды плавающей арифметики			
40	ADDF2	КОП <i>addr.rx, sum.mx</i>	Сложение двухоперандное в формате F
60	ADDD2	"—"	Сложение двухоперандное в формате D
40FD	ADDG2	"—"	Сложение двухоперандное в формате G
60FD	ADDH2	"—"	Сложение двухоперандное в формате H
4I	ADDF3	КОП <i>add1.rx, add2.rx, sum.wx</i>	Сложение трехоперандное в формате F
6I	ADDD3	"—"	Сложение трехоперандное в формате D

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
4IFD	<i>ADDG3</i>	КОП <i>add1.rx,add2.rx,Sum.wx</i>	Сложение трехоперандное в формате <i>G</i>
6IFD	<i>ADDH3</i>	-"-	Сложение трехоперандное в формате <i>H</i>
D4	<i>CLRL, CLRF</i>	КОП <i>dst.wx</i>	Очистка двойного слова /или слова формата <i>F</i> /
7C	<i>CLRQ, CLRD, CLRG</i>	-"-	Очистка четверного слова /или слов форматов <i>D, G</i> /
7CFD	<i>CLRO, CLRH</i>	-"-	Очистка октослова /или слова формата <i>H</i> /
5I	<i>CMPF</i>	КОП <i>src1.rx,src2.rx</i>	Сравнение слов формата <i>F</i>
7I	<i>CMPD</i>	-"-	Сравнение слов формата <i>D</i>
5IFD	<i>CMPG</i>	-"-	Сравнение слов формата <i>G</i>
7IFD	<i>CMPH</i>	-"-	Сравнение слов формата <i>H</i>
4B	<i>CVTRFL</i>	КОП <i>src.rx,dst.wl</i>	Преобразование слова формата <i>F</i> в двойное слово с округлением
6B	<i>CVTRDL</i>	-"-	Преобразование слова формата <i>D</i> в двойное слово с округлением
4BFD	<i>CVTRGL</i>	-"-	Преобразование слова формата <i>G</i> в двойное слово с округлением
6BFD	<i>CVTRHL</i>	-"-	Преобразование слова формата <i>H</i> в двойное слово с округлением
48	<i>CVTFB</i>	КОП <i>src.rx,dst.wy</i>	Преобразование слова формата <i>F</i> в байт

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
56	CVTFD	KOI <i>s7c.7x, dst.wy</i>	Преобразование слова формата F в слово формата D
99FD	CVTFG	-"-	Преобразование слова формата F в слово формата G
98FD	CVTFH	-"-	Преобразование слова формата F в слово формата H
4A	CVTFL	-"-	Преобразование слова формата F в двойное слово
49	CVTFW	-"-	Преобразование слова формата F в слово
48FD	CVTGB	-"-	Преобразование слова формата G в байт
33FD	CVTGF	-"-	Преобразование слова формата G в слово формата F
56FD	CVTGH	-"-	Преобразование слова формата G в слово формата H
4AFD	CVTGL	-"-	Преобразование слова формата G в двойное слово
49FD	CVTGW	-"-	Преобразование слова формата G в слово
68FD	CVTHB	-"-	Преобразование слова формата H в байт
F7FD	CVTHD	-"-	Преобразование слова формата H в слово формата D
56FD	CVTHF	-"-	Преобразование слова формата H в слово формата F
76FD	CVTHG	-"-	Преобразование слова формата H в слово формата G
6AFD	CVTHL	-"-	Преобразование слова формата H в двойное слово
69FD	CVTHW	-"-	Преобразование слова формата H в слово

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
68	CVTDB	KOP <i>s?c.??,dst.wy</i>	Преобразование слова формата D в байт
76	CVTDF	"-	Преобразование слова формата D в слово формата F
32FD	CVTDH	"-	Преобразование слова формата D в слово формата H
6A	CVTDL	"-	Преобразование слова формата D в двойное слово
69	CVTDW	"-	Преобразование слова формата D в слово
6C	CVTB <small>D</small>	"-	Преобразование байта в слово формата D
4C	CVTB <small>F</small>	"-	Преобразование байта в слово формата F
4CFD	CVTB <small>G</small>	"-	Преобразование байта в слово формата G
6CFD	CVTB <small>H</small>	"-	Преобразование байта в слово формата H
6D	CVTW <small>D</small>	"-	Преобразование слова в слово формата D
4D	CVTW <small>F</small>	"-	Преобразование слова в слово формата F
4DFD	CVTW <small>G</small>	"-	Преобразование слова в слово формата G
6DFD	CVTW <small>H</small>	"-	Преобразование слова в слово формата H
6E	CVTL <small>D</small>	"-	Преобразование двойного слова в слово формата D
4E	CVTL <small>F</small>	"-	Преобразование двойного слова в слово формата F
4EFD	CVTL <small>G</small>	"-	Преобразование двойного слова в слово формата G

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
6EF0	CVTLN	КОП <i>xc.zx,dst.wy</i>	Преобразование двойного слова в слово формата Н
46	DIVF2	КОП <i>divt.zx,qua.tx</i>	Деление двухоперандное в формате F
66	DIVD2	-"-	Деление двухоперандное в формате D
46FD	DIVG2	-"-	Деление двухоперандное в формате G
66FD	DIVH2	-"-	Деление двухоперандное в формате H
47	DIVF3	КОП <i>divt.zx,divd.zx,qua.wx</i>	Деление трехоперандное в формате F
67	DIVD3	-"-	Деление трехоперандное в формате D
47FD	DIVG3	-"-	Деление трехоперандное в формате G
67FD	DIVH3	-"--	Деление трехоперандное в формате H
54	EMODF	КОП <i>mulz.zx,mulzx.yb,muld.zx,int.wl,frac.wx</i>	Расширенное умножение в формате F с выделением целой части
74	EMODD	-"--	Расширенное умножение в формате D с выделением целой части
54FD	EMODG	КОП <i>mulz.zx,mulzx.yw,muld.zx,int.wl,frac.wx</i>	Расширенное умножение в формате G с выделением целой части
74FD	EMODH	-"--	Расширенное умножение в формате H с выделением целой части

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
52	MNEG F	КОП <i>src.1x,dst.wx</i>	Пересылка с арифметическим отрицанием в формате F
72	MNEG D	"	Пересылка с арифметическим отрицанием в формате D
52FD	MNEG G	"	Пересылка с арифметическим отрицанием в формате G
72FD	MNEG H	"	Пересылка с арифметическим отрицанием в формате H
50	MOV F	"	Пересылка слов формата F
70	MOV D	"	Пересылка слов формата D
50FD	MOV G	"	Пересылка слов формата G
70FD	MOV H	"	Пересылка слов формата H
44	MUL F2	КОП <i>mul.1x,prod.wx</i>	Умножение двухоперандное в формате F
64	MUL D2	"	Умножение двухоперандное в формате D
44FD	MUL G2	"	Умножение двухоперандное в формате G
64FD	MUL H2	"	Умножение двухоперандное в формате H
45	MUL F3	КОП <i>mul.1x,muld.1x,prod.wx</i>	Умножение трехоперандное в формате F
65	MUL D3	"	Умножение трехоперандное в формате D
45FD	MUL G3	"	Умножение трехоперандное в формате G
65FD	MUL H3	"	Умножение трехоперандное в формате H

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
55	POLYF	KOP $\arg.zf, \text{degree.zw}, t6laddr.ab, \{R0-3.w\}$	Вычисление значения полинома в формате F
75	POLYD	KOP $\arg.zd, \text{degree.zw}, t6laddr.ab, \{R0-5.w\}$	Вычисление значения полинома в формате D
55FD	POLYG	KOP $\arg.zg, \text{degree.zw}, t6laddr.ab, \{R0-5.w\}$	Вычисление значения полинома в формате G
75FD	POLYH	KOP $\arg.zh, \text{degree.zw}, t6laddr.ab, \{R0-5.w, -1(SP).w\}$	Вычисление значения полинома в формате H
42	SUBF2	KOP $sub.zx, dif.mx$	Вычитание двухоперандное в формате F
62	SUBD2	"	Вычитание двухоперандное в формате D
42FD	SUBG2	"	Вычитание двухоперандное в формате G
62FD	SUBH2	"	Вычитание двухоперандное в формате H
43	SUBF3	KOP $sub.zx, min.zx, dif.wx$	Вычитание трехоперандное в формате F
63	SUBD3	"	Вычитание трехоперандное в формате D
43FD	SUBG3	"	Вычитание трехоперандное в формате G
63FD	SUBH3	"	Вычитание трехоперандное в формате H
53	TSTF	KOP $stc.zx$	Тестирование в формате F
73	TSTD	"	Тестирование в формате D
53FD	TSTG	"	Тестирование в формате G

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
73ED	TSTH	КОП src.rw	Тестирование в формате H
		IO. Команды обработки строк символов	
29	CMPC3	КОП len.rw, src1addr.ab, src2addr.ab, {R0-3.wl}	Трехоперандное символьное сравнение
2D	CMPCS	КОП src1len.rw, src1addr.ab, file.r6, src2len.rw, src2addr.ab, {R0-3.wl}	Пятиоперандное символьное сравнение
3A	LOCc	КОП char.r6, len.rw, addr.ab, {R0-1.wl}	Поиск символа
39	MATCHC	КОП len1.rw, addr1.ab, len2.rw, addr2.ab, {R0-3.wl}	Поиск подстроки
28	MOV C3	КОП len.rw, srcaddr.ab, dstaddr.ab, {R0-5.wl}	Трехоперандная пересылка символов
2C	MOV C5	КОП srclen.rw, srcaddr.ab, file.r6, dstlen.rw, dstaddr.ab	Пятиоперандная пересылка символов
2E	MOVTC	КОП srclen.rw, srcaddr.ab, file.r6, t6laddr.ab, dstlen.rw, dstaddr.ab, {R0-5.wl}	Пересылка символов с трансляцией
2F	MOVTCUC	КОП srclen.rw, srcaddr.ab, esc.r6, t6laddr.ab, dstlen.rw, dstaddr.ab, {R0-5.ml}	Пересылка символов с трансляцией и выходом по указанному символу

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
2A	SCANC	KOP $\text{len}.\text{rw}, \text{addr}.\text{ab},$ $\text{tbladdr}.\text{ab}, \text{mask}.\text{rb}, \{\text{R0-3.w}\}$	Сканирование символов
3B	SKPC	KOP $\text{char}.\text{rb}, \text{len}.\text{rw}, \text{addr}.\text{ab},$ $\{\text{R0-1.w}\}$	Пропуск символов
2B	SPANC	KOP $\text{len}.\text{rw}, \text{addr}.\text{ab},$ $\text{tbladdr}.\text{ab}, \text{mask}.\text{rb},$ $\{\text{R0-3.w}\}$	Поиск символа
II. Команды десятичной строки цифр			
20	ADDP4	KOP $\text{addlen}.\text{rw}, \text{addaddr}.\text{ab},$ $\text{sumlen}.\text{rw}, \text{sumaddr}.\text{ab},$ $\{\text{R0-3.w}\}$	Четырехоперандное сложение в упакованном формате
21	ADDP6	KOP $\text{add1len}.\text{rw}, \text{add1addr}.\text{ab},$ $\text{add2len}.\text{rw}, \text{add2addr}.\text{ab},$ $\text{sumlen}.\text{rw}, \text{sumaddr}.\text{ab},$ $\{\text{R0-5.w}\}$	Шестиоперандное сложение в упакованном формате
F8	ASHP	KOP $\text{cnt}.\text{rb}, \text{src1len}.\text{rw},$ $\text{src1addr}.\text{ab}, \text{round}.\text{rb}, \text{dstlen}.\text{rw},$ $\text{dstaddr}.\text{ab}, \{\text{R0-3.w}\}$	Арифметический сдвиг и округление в упакованном формате
35	CMPP3	KOP $\text{len}.\text{rw}, \text{src1addr}.\text{ab},$ $\text{src2addr}.\text{ab}, \{\text{R0-3.w}\}$	Трехоперандное сравнение в упакованном формате
37	CMPP4	KOP $\text{src1len}.\text{rw}, \text{src1addr}.\text{ab},$ $\text{src2len}.\text{rw}, \text{src2addr}.\text{ab},$ $\{\text{R0-3.w}\}$	Четырехоперандное сравнение в упакованном формате

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
59	CVTLP	КОП $src.l, dstlen.rw,$ $dstaddr.ab, \{R0-3.w\}$	Преобразование двойного слова в упакованный формат
36	CVTPL	КОП $srcLen.rw, srcaddr.ab,$ $\{R0-3.w\}, dst.w$	Преобразование упакованного формата в двойное слово
08	CVTPS	КОП $srcLen.rw, srcaddr.ab,$ $dstLen.rw, dstaddr.ab,$ $\{R0-3.w\}$	Преобразование упакованного формата в числовый с выделенным ведущим знаком
24	CVTPT	КОП $srcLen.rw, srcaddr.ab,$ $tbladdr.ab, dstLen.rw,$ $dstaddr.ab, \{R0-3.w\}$	Преобразование из упакованного формата в усеченный числовый
09	CVTSP	КОП $srcLen.rw, srcaddr.ab,$ $dstLen.rw, dstaddr.ab,$ $\{R0-3.w\}$	Преобразование из числового формата с выделенным ведущим знаком в упакованный
26	CVTTP	КОП $srcLen.rw, srcaddr.ab,$ $tbladdr.ab, dstLen.rw,$ $dstaddr.ab, \{R0-3.w\}$	Преобразование из усеченного числового формата в упакованный
27	DIVP	КОП $divLen.rw, divaddr.ab,$ $divdlen.rw, divdaddr.ab,$ $quoLen.rw, quoaddr.ab,$ $\{R0-5.w, -16(SP):-1(SP).w\}$	Деление в упакованном формате
34	MOV P	КОП $len.rw, srcaddr.ab,$ $dstaddr.ab, \{R0-3.w\}$	Пересылка в упакованном формате
25	MULP	КОП $mulLen.rw, mulraddr.ab,$ $muldlen.rw, muldaddr.ab,$ $prodLen.rw, prodaddr.ab,$ $\{R0-5.w\}$	Умножение в упакованном формате

Продолжение табл. 8

Код	Мнемоника	Формат	Назначение
22	SUBP4	КОП sublen.zw,subaddr.ab, diflen.zw,difaddr.ab, {R0-3.wl}	Четырехоперандное вычитание в упакован- ном формате
23	SUBP6	КОП sublen.zw,subaddr.ab, minlen.zw,minaddr.ab, diflen.zw,difaddr.ab, {R0-5.wl}	Шестиоперандное вычитание в упакован- ном формате
I2. Контроль циклической избыточности			
0B	CRC	КОП t6L.ab,inccrcL, strlen.zw,stramt.ab	Вычисление циклического полинома для проверки
I3. Редактирование			
38	EDITPC	КОП strlen.zw, srcaddr.ab,pattern.ab, dstaddr.ab	Редактирование упако- ванной строки в символь- ную

Формат команды

В вычислительной системе "Электроника 32" используется формат команды переменной длины. Длина формата может быть от одного байта и выше в зависимости от типа команды. Обобщенная схема формата команды приведена на рис. 51.

Общая схема формата команды

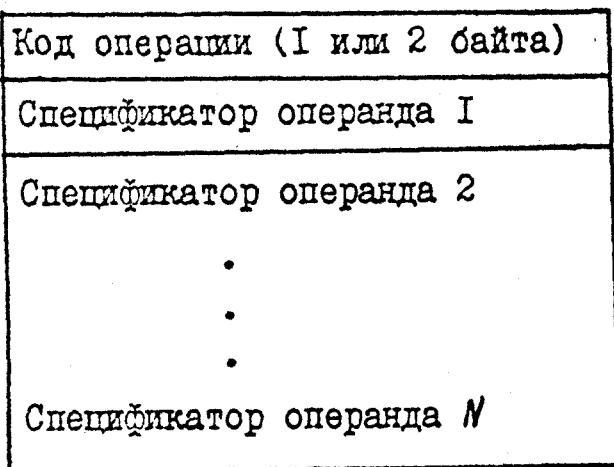


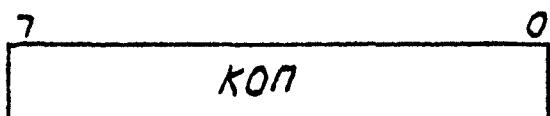
Рис. 51.

Каждая команда состоит из кода операции и следующих за ним от 0 до 6 спецификаторов operandов, количество и тип которых зависят от кода операции. Код операции точно определяет как требуемая операция будет выполняться. Под код операции может отводиться один или два байта, в зависимости от содержимого байта с адресом "A". Два байта будут использоваться под код операции при условии, что значение числа, содержащегося в этом байте, находится в диапазоне от FС(16) до FF(16). Формат кода операции приведен на рис. 52.

Каждый из спецификаторов операнда имеет одинаковый формат, т.е. метод адресации плюс дополнительная информация. Эта дополнительная информация содержит до двух переменных: указателя регистра и адрес; указатель регистра и данные; указатель регистра

Формат кода операции

Однобайтовый код операции



Двухбайтовый код операции

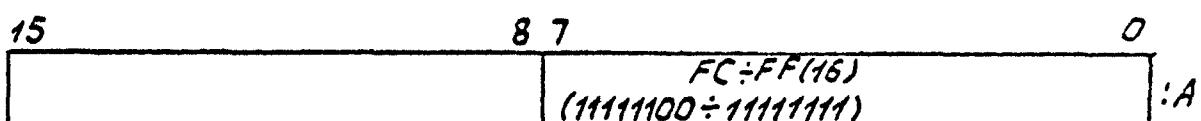


Рис. 52

и смещение. Использование операнда определяется косвенно из кода операции и называется типом операнда.

Спецификатор операнда описывается следующим образом:

<Название> , <тип доступа> <тип данных>
тип операнда

где:

<Название> – определяет operand в контексте инструкции.

<Тип доступа> – спецификатора операнда обозначается одной буквой:

- a – вычисляется эффективный адрес операнда. Адрес формируется в двойном слове, которое является фактическим операндом инструкции. Контекст вычисления адреса является типом данных, т.е. величиной, используемой для автонкрементации; автодекрементации и индексации;
- b – нет ссылок на operand. Спецификатор операнда представляет собой смещение перехода. Размер смещения перехода определяется типом данных;

- m* - операнд считывается, может быть модифицирован и записывается. Если операнд фактически не модифицируется, то он может и не записываться обратно. Несмотря на это, операнды с таким типом доступа всегда проверяются на возможность как чтения, так и записи;
- r* - операнд только читается;
- v* - вычисляется эффективный адрес операнда (доступ "поле"). Если эффективный адрес формируется в памяти, то он сохраняется в двойном слове, которое является фактическим операндом инструкции. Если эффективным адресом является регистр R_n , то операндом является $R[n]$ или $R[n+1]R[n]$;
- w* - операнд доступен только по записи.

<Тип данных> - обозначают следующим образом:

- b* - байт
- d* - плавающее слово типа *D*
- f* - *"-* *F*
- g* - *"-* *G*
- h* - *"-* *H*
- e* - двойное слово
- o* - восьмерное слово
- q* - четверное слово
- w* - слово
- x* - {первый тип данных определяется инструкцией}
- y* - {второй тип данных определяется инструкцией}

Адресация операндов

Методы адресации совместно с шестнадцатью регистрами общего назначения обеспечивают удобный метод работы с данными, так как они точно определяют, каким образом конкретные регистры используются с целью доступа, обработки и хранения данных и команд в памяти. Регистры общего назначения могут быть ис-

пользованы командой любым из следующих способов:

- как аккумуляторы;
- как указатели (содержимое регистра – адрес операнда, а не сам операнд);
- как индексные регистры.

Адресация может быть разделена на две категории: основные методы адресации и адресация по условному переходу. Табл. 9 дает краткое описание методов адресации через регистры общего назначения и через программный счетчик. В табл. 9 приняты следующие обозначения:

- | | |
|-----|---|
| D | – смещение; |
| I | – любой метод адресации, допускающий индексирование; |
| - | – логически невозможно; |
| PMA | – ошибка по резервной моде адресации; |
| AC | – адресация через счетчик команд; |
| НП | – непредсказуемый результат; |
| НЧ | – непредсказуемый результат для Q(0) слова, D(G,H) форматов, а также для поля бит переменной длины, если "позиция" плюс "размер" больше 32; |
| НВ | – непредсказуемый результат для Q-слова и формата H; |
| НИР | – непредсказуемый результат для индексного регистра, такого как базовый регистр; |
| ДА | – всегда действительный метод адресации; |
| Ч | – доступ по чтению; |
| З | – доступ по записи; |
| М | – доступ по чтению-модификации-записи; |
| А | – доступ по адресу; |
| БА | – доступ по базовому адресу поля бит переменной длины. |

Таблица 9

Краткий обзор методов адресации

Продолжение табл.9

Мо- да (16)	Название мето- да адресации	Ассемблер	Доступ						Исполь- зование		
			Ч	М	З	А	БА	РС	Р	ИНД	
<u>Общая регистровая адресация</u>											
8	Непосредствен- ный	<i>I # Const</i>	ДА	НП	НП	ДА	ДА	-	-	ДА	
9	Абсолютный	<i>@ # address</i>	ДА	ДА	ДА	ДА	ДА	-	-	ДА	
A	Относительный байтовый	<i>B # address</i>	ДА	ДА	ДА	ДА	ДА	-	-	ДА	
B	Косвенно-отно- сительный бай- товый	<i>@ B # address</i>	ДА	ДА	ДА	ДА	ДА	-	-	ДА	
C	Относительный словный	<i>w # address</i>	ДА	ДА	ДА	ДА	ДА	-	-	ДА	
D	Косвенно-отно- сительный словный	<i>@ w # address</i>	ДА	ДА	ДА	ДА	ДА	-	-	ДА	
E	Относительный двухсловный	<i>L # address</i>	ДА	ДА	ДА	ДА	ДА	-	-	ДА	
F	Косвенно-отно- сительный двухсловный	<i>@ L # address</i>	ДА	ДА	ДА	ДА	ДА	-	-	ДА	

ВРЕМЯ ВЫПОЛНЕНИЯ ТИПОВЫХ КОМАНД (TIMING)

Времена выполнения команд (групп команд) микропроцессором М1839ВМ1 определены в микроциклах (I микроцикл равен двум периодам тактовой частоты $2T_{CLC}$), при условии идеальной памяти, пренебрегая временем на сбои конвейера адресных/безадресных подкачек команд, не учитывая внутренних торможений, обусловленных особенностями конвейера команд (это торможение по занятости регистра смещения – TCM, торможение по занятости буфера обмена данных и торможения приема микрокоманд). Данная процедура определения времени выполнения групп команд проводится для случая регистровых мод адресации для всех операндов команды. В случае, если применение регистровой моды адресации недопустимо, не имеет смысла, либо не является показательным в плане оценки времени выполнения, приводится время выполнения (в микроциклах) только содержательной микрообработки, исключая этап микропрограммной адресации operandов, а также время аппаратной дешифрации команд, что обычно скрыто конвейером выполнения команд. Процедура TIMING – чисто качественная оценка времен выполнения типовых команд.

Сокращения, применяемые ниже по тексту:

- Р – регистр;
- П – память;
- РР – формат регистр-регистр;
- ПП – формат память-память;
- РП – формат регистр-память.

I. Команда ADAWI

Случай РР – один микроцикл, особого смысла не имеет, так как в этом случае ADAWI эквивалентно простому сложению ADDW, поскольку не реализуется блокировка памяти.

Случай ПП – восемь микроциклов.*

Изм	Лист	№докум	Подп	Дата

2. Группа команд ADDB2, ADDW2, ADDL2, ADDB3, ADDW3, ADDL3,
SUBB2, SUBW2, SUBL2, SUBB3, SUBW3, SUBL3

PP - один микроцикл.*

3. Группа команд ADWC, SBWC

PP - один микроцикл.*

4. Команда ASHL

Случай регистр-приемник и приемник-память почти эквивалентны.
Даже в случае регистровых мод адресации выполняется адресация
операндов.

Параметр сдвига равен I: ~ 20 микроциклов.*

При параметре сдвига, равном n (до 2^8 сдвигов):

$$T = 20 + 3(n - I) \text{ микроциклов.}$$

5. Команда ASHQ

Случай регистр-приемник и приемник-память почти эквивалентны.
Даже в случае регистровых мод адресации выполняется адресация
операндов.

Параметр сдвига равен I: 28 микроциклов.*

При параметре сдвига, равном n (до 2^8 сдвигов):

$$T = 28 + 4(n - I) \text{ микроциклов.}$$

6. Группа команд BICB2, BICW2, BICL2, BICB3, BICW3, BICL3,
BISB2, BISW2, BISL2, BISB3, BISW3, BISL3

PP, PPP - один микроцикл.*

7. Группа команд BITB, BITW, BITL

PP - один микроцикл.*

8. Группа команд CLRБ, CLRW, CLRL

P - один микроцикл.*

9. Команда CLRQ

Память-приемник: два микроцикла.*

Регистр-приемник: три микроцикла.*

Лист	№докум	Подп.	Дата

I0. Группа команд CMPB, CMPW, CMPL

PP - max пять микросекунд.*

I1. Группа команд CVTBW, CVTBL, CVTWL

PP, РП - max девять микросекунд.*

Случаи регистр-приемник и память-приемник почти эквивалентны.

I2. Группа команд CVTLB, CVTLW, CVTWB

PP, РП - max 13 микросекунд.*

Случаи регистр-приемник и память-приемник почти эквивалентны.

I3. Группа команд DECB, DECW, DECL, INCB, INCW, INCL

Р, П - один микросекунд.*

I4. Группа команд MCOMB, MCOMW, MCOML, MNGB, MNFW, MNGL

PP - три микросекунды.*

I5. Группа команд MOVB, MOVW, MOWL

PP - один микросекунд.*

I6. Команда MOVQ

Случаи РП, PP - почти эквивалентны;

- max восемь микросекунд.*

I7. Группа команд MOVZBL, MOVZBW, MOVZWL

PP - шесть микросекунд.*

I8. Команда PUSHL

Р - четыре микросекунды.

I9. Команда PUSHR

Упаковывание в стек всех регистров (РОнов), т.е. все разряды операнда mask<0...I4> установлены в единицу: 65 микросекунд.*

20. Команда POPR

Восстановление из стека всех регистров (РОнов), т.е. все разряды операнда mask<0...I4> установлены в единицу: 65 микросекунд.*

21. Команда ROTL

Случай регистр-приемник и память-приемник почти эквивалентны. Даже в случае регистровых мод адресации выполняется микропрограммно этап адресации операндов.

Параметр сдвига равен I : $I8$ микроциклов.*

При параметре сдвига, равном n (до 2^8 сдвига):

$$T = I8 + 3(n - I) \text{ микроциклов.}$$

22. Группа команд TSTB, TSTW, TSTL

P, P - один микроцикл.*

23. Группа команд XORB2, XORW2, XORL2, XORB3, XORW3, XORL3

PP, PPP - один микроцикл.*

24. Группа команд MOVAB, MOVAW, MOVAL, MOVAE, MOVAQ, MOVAD, MOVAG

Независимо от моды адресации операнда - источника выполняется микропрограммная адресация operandов (адресный тип доступа).

PP, PPP - один микроцикл.*

25. Группа команд PUSHAB, PUSHAW, PUSHAL, PUSHAE, PUSHAQ, PUSHAD, PUSHAG

Независимо от моды адресации operand-источника выполняется микропрограммная адресация operandов (адресный тип доступа). - четыре микроцикла.*

26. Группа команд CALLS, CALLG

Случай сохранения всех РОНов в кадре вызова процедуры:

~ $I15$ микроциклов.*

27. Команда RET

Случай загрузки всех РОНов из кадра вызова процедуры:

~ $I26$ микроциклов.*

28. Группа команд BICPSW, BISPSW

Независимо от моды адресации operand $\langle mask \rangle$ выполняется микропрограммная адресация operandов: ~ пять микроциклов.*

29. Команда INDEX

Независимо от моды адресации операндов выполняется микропрограммная адресация операндов:

~ 225 микроциклов (случай микропрограммной реализации операции умножения при отсутствии сопроцессора);

~ 45 микроциклов (случай реализации операции умножения на сопроцессоре).

30. Группа команд MOVPSL, NOP

Один микроцикл.

31. Команды работы с очередями

От 80 до 120 микроциклов.

32. Команды обработки строк символов

Мин 30 микроциклов.

33. Команды для работы с переменными битовыми полями

От 80 до 100 микроциклов.

34. Команды ветвлений (*Branches*):

Простые: AOBLEQ, AOBLS, BLSS, BLEQ, BEQL, BEQLU, BN EQ,
BNEQU, BGEQ, BGTR, BLSSU, BCS, BLEQU, BGEQU, BCC, BGTRU,
BVS, BVC, BLBS, BLBC, BRB, BRW, BSBB, BSBW, JMP, SOBGEQ,
SOBGTR - от пяти до восьми микроциклов.

Сложные: ACBV, ACBW, ACBL, ACBF, ACBD, ACBG, ACBH, BBS,
BBC, BBSS, BBSC, BBCS, BBSSI, BBCCI, CASEB, CASEW, CASEL,
JSB - от 10 до 60 микроциклов.

35. Время реакции на прерывания

Время реакции на прерывания - это время от момента восприятия центральным процессором запроса на прерывание до начала выполнения первой команды соответствующей программы обработки прерываний.

На тактовой частоте 10 МГц это время составляет 10-12 мкс. Наибольшее значение (для случая машинного сбоя) составляет 16 мкс. В основном, это связано с большим числом параметров (12), сохраняемых

в стеке (и соответственно, таким же числом обменов записей в ОЗУ) для данного типа прерываний.

Примечание:

* - только содержательная микрообработка, без учета адресации операндов.

Лист регистрации изменений

1 листов (страниц)		Всего листов (страниц)		Входящий № сопроводительного документа и дата	Подп	Дата
Замененных	новых	аннулированных	в докум.	№ докум.		
01,4,5,15, 16,17,18				ЩИБ33-92	Жасу	29/ХII/92.