

РАБОТА № 1

СТРУКТУРА ПРОЦЕССОРА И СОСТАВ МИКРОКОМАНД

Цель работы: изучение процессора на уровне структурной схемы, ознакомление со структурой микрокоманд (МК) и порядком ввода данных, кодирование и выполнение МК.

1. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

По материалам разделов 2 и 3 изучить структуру процессора и основных блоков, структуру МК и функции ее полей. По материалам раздела 4 ознакомиться с порядком ввода микропрограмм (МП) и исходных данных.

Выполнить упражнения из раздела 5 по указанию преподавателя.

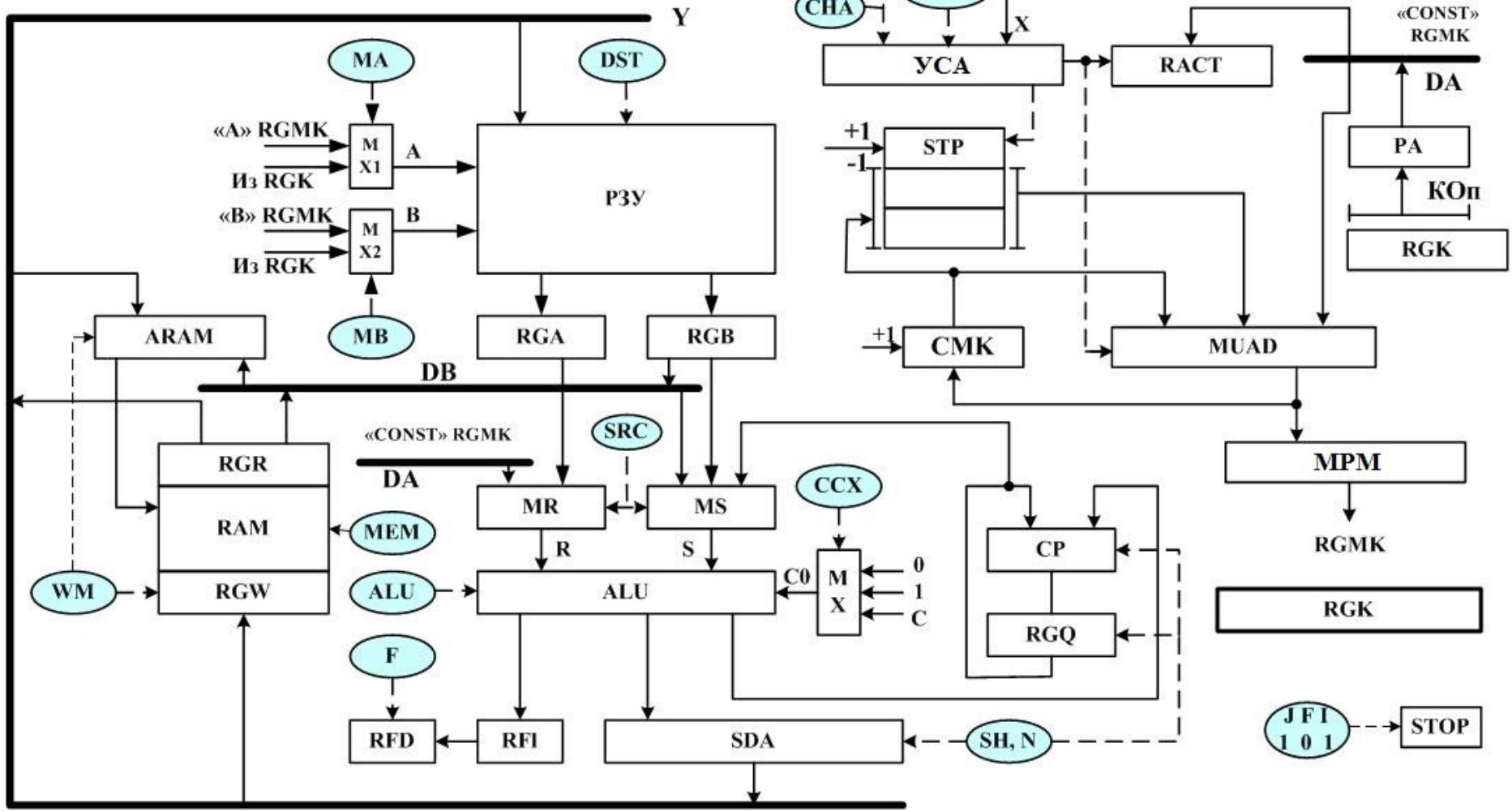
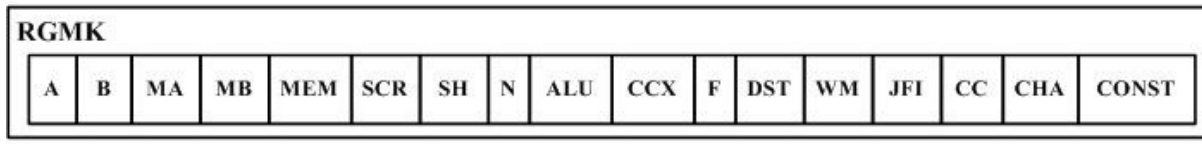
2. СТРУКТУРА ПРОЦЕССОРА

2.1. Состав процессора

В состав микропрограммируемого процессора (МП) входят операционный блок, блок микропрограммного управления (БМУ), оперативная память (RAM) и микропрограммная память (МРМ). Он может использоваться для выполнения алгоритмов в форме микропрограмм, а также для эмуляции операций в заданной системе команд. В последнем случае для выполнения каждой команды вызывается соответствующая микропрограмма.

Структурная схема микропрограммируемого процессора в укрупненном виде приведена на рисунке. В овалах показаны названия полей микрокоманды, которые управляют тем или иным устройством МП.

Микропрограммная память содержит 64 микрокоманды по 64 бит с адресами 0-3F.



2.2. Оперативная память

Оперативная память (ОП) содержит 512 16-разрядных слов (1024 байта) с 16-ричными адресами 0-3FF. ОП связана с операционным блоком через регистр чтения RGR, регистр записи RGW и адресный регистр ARAM. Чтобы прочитать информацию из ОП, необходимо в ARAM записать адрес и на управляющий вход подать сигнал «Чтение ОП» (поле MEM микрокоманды). В результате в RGR будет помещен результат чтения ОП: $RGR := RAM[ARAM]$. Возможно чтение как слова, так и отдельно старшего или младшего байта. С выхода RGR возможны передачи в регистр команд RGK, на мультиплексор MS через шину DB, как код одного из операндов, и на шину Y для записи в регистровое запоминающее устройство (РЗУ).

Чтобы записать информацию в ОП, необходимо в адресный регистр ARAM записать адрес, а в регистр записи RGW занести данные для записи с шины Y и на управляющий вход ОП подать сигнал «Запись в ОП» (поле MEM).

2.3. Операционный блок

В операционный блок входят блок внутренней памяти, арифметико-логический блок, блок рабочего регистра.

Блок внутренней памяти включает в себя регистровое запоминающее устройство (РЗУ), состоящее из шестнадцати 16-разрядных регистров, предназначенных для хранения операндов и результатов (названия регистров приведены в табл. 2), мультиплексоров адреса MX1 и MX2, регистры RGA и RGB.

Чтение РЗУ происходит одновременно по адресам А и В в регистры RGA и RGB соответственно. Запись происходит по адресу В. Адреса А и В поступают на РЗУ из одноименных полей микрокоманды или из полей reg1, reg2, r/m регистра команд RGK. Выбор источника адреса осуществляется схемами мультиплексоров адреса MX1 и MX2 под управлением полей МА и МВ микрокоманды.

Арифметико-логический блок включает в себя мультиплексоры операндов MR и MS, управляемые полем микрокоманды SRC, арифметико-

логическое устройство ALU, сдвигатель SDA и регистры флажков RFI и RFD. На входы ALU поступают данные R и S соответственно с мультиплексоров MR и MS, а также входной перенос C0, значения которого определяется полем микрокоманды CCX. Источниками операндов ALU для входа R являются RGA и CONST из соответствующего поля микрокоманды. Источниками операндов ALU для входа S являются RGB, RGR и RGQ. Сдвигатель SDA управляется полями микрокоманды SH и N.

Флажки формируются и хранятся только на время исполнения текущей микрокоманды в регистре флажков RFI. При необходимости они переписываются в регистр длительного хранения RFD. Обозначение флажков: N - знак минус (старший бит результата), Z - признак нуля, V - признак переполнения, C - перенос из старшего бита, M - признак переноса при выполнении операции умножения на два разряда, P – признак паритета (нечетное число единиц в результате).

Блок рабочего регистра состоит, собственно, из рабочего регистра RGQ и сдвигателя CP. RGQ используется при умножении, делении, двойном сдвиге, а также как аккумулятор и один из источников операнда S. Запись в него производится с выхода ALU со сдвигом вправо, влево или без сдвига. Управляет сдвигом CP поля SH и N.

2.4. Блок микропрограммного управления

В состав данного блока входят:

- схема управления последовательностью микрокоманд (УПМ);
- микропрограммная память (МРМ) емкостью 64 64-битных микрокоманд;
- регистр микрокоманд RGMK;
- схема формирования признака STOP.

Основная функция УПМ – формирование последовательности адресов микрокоманд и организация циклических вычислительных процессов. В схеме управления последовательностью микрокоманд можно выделить шесть основных блоков: мультиплексор адреса MUAD, счетчик микрокоманд

СМК, счетчик циклов RACT, аппаратный стек, схему управления следующим адресом УСА. Все элементы схемы имеют разрядность 16.

Мультиплексор адреса MUAD выбирает в качестве адреса содержимое СМК, стека или прямой выход с шины DA. Адрес на эту шину может поступить из поля CONST микрокоманды или с преобразователя начального адреса РА. Выход MUAD соединен с адресным входом микропрограммной памяти МРМ и входом СМК.

Счетчик микрокоманд СМК содержит адрес следующей микрокоманды.

Регистр счетчика циклов RACT используется для записи и хранения числа циклов, поступающего с шины DA.

В аппаратном стеке емкостью 2 слова хранятся адреса возврата из подпрограмм. Указатель вершины стека STP содержит адрес последнего занесенного в стек значения.

Схема управления следующим адресом (УСА) формирует этот адрес под управлением поля СНА. В состав УСА входят мультиплексор кода условия (МКУ) и инвертор кода условия (ИКУ). Мультиплексор МХ определяет какой регистр флажков RFI или RFD будет использоваться для формирования следующего адреса микрокоманды (бит F=1 в поле JFI). Мультиплексор кода условия (МКУ) в зависимости от поля СС выбирает флажок или флажки, которые формируют код условия X. ИКУ инвертирует сигнал X, если бит I поля JFI равен 1. Далее в зависимости от сигнала X, признака безусловного перехода (бит J=1 в поле JFI) и инструкции УПМ (поле СНА) определяется что является источником следующего адреса микрокоманды.

3. СТРУКТУРА МИКРОКОМАНДЫ И ОПИСАНИЕ ПОЛЕЙ

Микрокоманда содержит 64 бит и включает в себя 17 полей. Структура микрокоманды представлена в табл. 1.

Поля А и В задают адреса регистров РЗУ. Табл. 2 задает соответствие между адресами РЗУ и регистрами микропроцессора 8086. В регистре Е хранится копия регистра команд RGK. Регистр F используется как рабочий

регистр RW. При работе в микропрограммном режиме регистры РЗУ могут использоваться произвольным образом.

Структура микрокоманды

Таблица 1

Поле	A	B	MA	MB	MEM	SRC	SH	N	ALU
Бит	4	4	2	2	3	3	4	4	4
Def	0	0	0	0	0	1	0	0	6
Поле	CCX	F	DST	WM	JFI	CC	CHA	CONST	
Бит	2	1	3	2	3	3	4	16	
Def	0	0	0	0	0	0	7	0	

Обозначения:

Поле - наименование поля микрокоманды;

Бит - число бит в поле;

Def - значение поля по умолчанию.

Выбор регистров РЗУ

Таблица 2

Поле A/B	Значение	Поле A/B	Значение
0	AX	8	CS
1	CX	9	SS
2	DX	A	DS
3	BX	B	ES
4	SP	C	IP
5	BP	D	PSW
6	SI	E	RGK
7	DI	F	RW

Поля MA и MB управляют выбором адресов РЗУ из микрокоманды или регистра команд RGK. По умолчанию MA=MB=0, что соответствует выбору адресов из микрокоманды. Значения полей MA и MB приведены в табл. 3.

Управление адресом РЗУ

Таблица 3

Поле MA/MB	0	1	2	3
Источник адреса	Поле A/B МК	reg1	reg2	r/m

Поле MEM управляет чтением и записью памяти. Оно состоит из трех бит. Первый бит - признак обращения к памяти, второй - определяет режим чтение/запись, третий - размер данных (байт или слово). Значения поля MEM приведены в табл. 4.

Управление памятью

Таблица 4

Поле MEM	Значение	Поле MEM	Значение
0	NOP	4	Чтение байта
1	NOP	5	Чтение слова
2	NOP	6	Запись байта
3	NOP	7	Запись слова

Поле SRC управляет выбором источников операндов. По умолчанию SRC=1, что соответствует выбору операндов из RGA и RGB. Значения поля SRC приведены в табл. 5.

Выбор источников операндов

Таблица 5

Поле SRC	0	1	2	3	4	5	6	7
Операнд R	0000	RGA	RGA	RGA	RGA*2	CONST	CONST	CONST
Операнд S	0000	RGB	RGQ	RGR	RGB	RGB	RGR	RGQ

Поле SH управляет работой сдвигателей SDA и CP. Поле N указывает количество разрядов, на которое надо сдвигать содержимое SDA и/или CP. По умолчанию SH=0 и N=0 - передача с выхода ALU на шину Y без сдвига. Значения поля SH приведены в табл. 6.

Управление сдвигателями

Таблица 6

Поле SH	Операция
0	Без сдвига
1	АС АЛУ вправо
2	ЛС АЛУ вправо
3	АС АЛУ, RGQ вправо
4	ЛС АЛУ, RGQ вправо
5	ЛС RGQ вправо
6	$RGQ \leftarrow ALU$
8	ЛС АЛУ влево
A	ЛС АЛУ, RGQ влево
E	Расширение знака

Примечание:

- ЛС и АС - логический и арифметический сдвиги соответственно.
- В операции «Расширение знака» все биты старшего байта (8-15) устанавливаются равными биту 7.

Поле ALU управляет операциями АЛУ. По умолчанию $ALU=6$, что означает сложение операнда R с входным переносом C0, определяемым полем CCX. Значения поля ALU приведены в табл. 7. Среди операций АЛУ интерес представляет операция умножения на 2 бита. Данная операция позволяет в два раза повысить скорость выполнения операции умножения по сравнению с обычным методом, основанным на анализе одного бита. Операция умножения на 2 бита имеет следующий алгоритм: анализируются 2 младших бита множителя, и в зависимости от них выполняются операции сложения и сдвига. Если биты равны 00, то происходит сдвиг множителя и суммы частичных произведений (СЧП) вправо на 2 бита. Если биты равны 01, то к СЧП прибавляется значение множимого и опять происходит сдвиг на 2 бита. Если биты равны 10, то происходит сложение СЧП с удвоенным множимым и сдвиг на два бита. Если биты равны 11, то из СЧП вычитается множимое и происходит перенос единицы в следующие два разряда.

Операции АЛУ

Таблица 7

Поле ALU	Операция АЛУ	Флажки				Поле ALU	Операция АЛУ	Флажки			
		N	Z	V	C			N	Z	V	C
0	На всех выходах «0»	0	1	0	0	8	Умножение на 2 бита	+	+	+	+
1	$S - R - 1 + C0$	+	+	+	+	9	$R \& S$	+	+	0	0
2	$R - S - 1 + C0$	+	+	+	+	A	$R \& \overline{S}$	+	+	0	0
3	$R + S + C0$	+	+	+	+	B	$\overline{R \& S}$	+	+	0	0
4	$S + C0$	+	+	+	+	C	$R \vee S$	+	+	0	0
5	$\overline{S} + C0$	+	+	+	+	D	$\overline{R \vee S}$	+	+	0	0
6	$R + C0$	+	+	+	+	E	$R \oplus S$	+	+	0	0
7	$\overline{R} + C0$	+	+	+	+	F	$\overline{R \oplus S}$	+	+	0	0

В модели это реализовано следующим образом: в регистр RGQ заносится множитель. В поле A заносится адрес РЗУ множимого, в поле B - адрес СЧП. В поле сдвигателя задается сдвиг на 2 бита. При правильном написании микропрограммы результат будет получен в RGQ (младшие цифры) и в

регистре СЧП (старшие цифры). Для реализации переноса введен специальный флажок М. Фактически происходит анализ не двух, а трех бит - двух младших бит множителя и флажка М.

Поле ССХ управляет входным переносом. По умолчанию значение ССХ=0. Значения поля ССХ приведены в табл. 8.

Управление входным переносом			Таблица 8	
Поле ССХ	0	1	2	3
СО	0	1	С	С

Поле F - поле фиксации флажков. При значении F=1 текущее значение флажков заносится в RFD.

Поле DST управляет записью данных с выходов SDA и RGR в PЗУ по адресу В. По умолчанию DST=0 (без записи). Значения поля DST приведены в табл. 9, где RGRL и RGRH соответственно младший и старший байты RGR, а PЗУL и PЗУH - младший и старший байты адресуемого регистра PЗУ.

Управление записью в PЗУ				Таблица 9	
Поле DST	0	1	2	3	4
Источник	Без записи	RGR	RGRL	RGRH	SDA
Приемник	Без записи	PЗУ	PЗУH	PЗУL	PЗУ

Поле WM управляет записью в память. Значения поля WM приведены в табл. 10.

Управление записью в память			Таблица 10	
Поле WM	0	1	2	3
Источник	Без записи	SDA	SDA	RGB
Приемник	Без записи	RGW	ARAM	ARAM

Поле JFI участвует совместно с полем СС в формировании условий перехода. Старший бит J=1 - признак безусловного перехода, бит F указывает, что условие перехода определяется флажками из RFD, бит I означает инверсию формируемого условия. При JFI=101 происходит остановка работы и формирование признака STOP. По умолчанию JFI=0.

Поле СС управляет формированием условий перехода. Обозначения переходов соответствуют условным переходам МП i8086. Значения поля СС

приведены в табл. 11. Для переходов, помеченных звездочкой, устанавливается бит инверсии I в поле JFI.

Формирование условий
перехода

Таблица 11

Поле CC	Вид перехода	Условие перехода
0	JP, JNP*	P=1
1	JZ, JNZ*	Z=1
2	JS, JNS*	N=1
3	JO, JNO*	V=1
4	JC, JNC*	C=1
5	JL, JNL*	$N \oplus V=1$
6	JLE, JNLE*	$Z \vee (N \oplus V)=1$
7	JBE, JNBE*	$C \vee Z=1$

Поле СНА обеспечивает формирование адреса следующей микрокоманды и содержит набор инструкций схемы УПМ. Значения поля СНА приведены в таблице 12. По умолчанию СНА=7 (функция CONT - продолжить). Столбец Y таблицы соответствует адресу следующей микрокоманды, столбец X - значению проверяемого условия (X=1, если условие выполняется, X=0, если оно не выполняется).

Инструкции схемы УПМ

Таблица 12

Поле СНА	Мнемоника	X=0		X=1		РАСТ
		Y	Стек	Y	Стек	
0	JZ	0	Очистка	0	Очистка	Хранение
1	CJS	CMK	Хранение	CONST	Загрузка	Хранение
2	JMAP	PA	Хранение	PA	Хранение	Хранение
3	CJP	CMK	Хранение	CONST	Хранение	Хранение
4	RPCT	CMK	Хранение	CONST	Хранение	Декремент
5	CRTN	CMK	Хранение	Стек	Выгрузка	Хранение
6	LDCT	CMK	Хранение	CMK	Хранение	Загрузка
7	CONT	CMK	Хранение	CMK	Хранение	Хранение
8-F	Резерв	-	-	-	-	-

Рассмотрим подробнее выполняемые инструкции:

JZ – управление передается МК с адресом 0, и происходит очистка стека путем установки STP в состояние 0.

CJS – условный переход к подпрограмме. При $X=1$ продвинутый адрес МК запоминается в стеке и происходит переход по адресу подпрограммы, заданному в поле CONST.

JMAP – переход по адресу из PA.

CJP – условный переход по адресу из поля CONST при $X=1$.

RPCT – повторение цикла (переход по счетчику). Если RACT не равен нулю, из него вычитается 1 и выполняется переход по адресу, расположенному в поле CONST. Если $RACT=0$, следующая микрокоманда выбирается в естественном порядке.

CRTN – условный возврат из подпрограммы. При $X=1$ происходит переход по адресу из стека и выполняется декремент STP.

LDCT – загрузка счетчика и продолжение. Из поля CONST в RACT загружаются адрес или число циклов.

CONT – продолжить выборку микрокоманд в естественном порядке. Этот код устанавливается по умолчанию.

Поле CONST содержит 16-битовую константу, подключаемую к шине DA операционного блока, или адрес перехода, подаваемый на шину DA блока микропрограммного управления. По умолчанию $CONST=0$.

4. ДИАЛОГ ПОЛЬЗОВАТЕЛЯ С МОДЕЛЬЮ

Приложение имеет имя MICRO и его ярлык находится на РАБОЧЕМ СТОЛЕ. При запуске приложения перед пользователем появляется основная форма, состоящая из двух страниц: «Работа» и «Данные». На странице «Работа» отображаются схема процессора и текущая микрокоманда. Страница имеет четыре кнопки: «Рестарт», «Продолжить», «Стоп», «Настройка».

При нажатии на кнопку «Рестарт» происходит сброс моделируемого процессора.

При нажатии на кнопку «Продолжить» в режиме МИКРОКОМАНДА происходит выполнение одной микрокоманды и остановка до следующего нажатия на эту кнопку. В режиме КОМАНДА происходит выполнение

микрокоманд, пока в текущей МК не будет признака конца команды. В режиме АВТОМАТ после рестарта происходит выполнение микрокоманд, пока не будет признака STOP или не будет выполнена последняя МК.

Кнопка «СТОП» доступна только в режиме АВТОМАТ. При нажатии на эту кнопку происходит остановка выполнения МК до нажатия на кнопку «Продолжить».

Кнопка «Настройка» вызывает окно настройки, в котором задается задержка при выполнении микрокоманд в режиме АВТОМАТ.

Страница «Данные» содержит четыре подчиненные страницы: «Регистры», «Память», «Микропрограммная память», «Таблица преобразования адресов».

На каждой из этих страниц имеются кнопки «Сохранить» и «Загрузить». При нажатии на них открываются стандартные диалоги сохранения файла и загрузки файла.

Страница «Регистры» позволяет задавать и изменять содержимое регистровой памяти микропроцессора и изменять режим отладки. Страница «Регистры» содержит также кнопки «ОК» и «Отмена». При нажатии на кнопку «ОК» внесенные в регистры изменения передаются в схему, в противном случае изменения игнорируются.

Вкладка «Память» позволяет вводить в оперативную память данные по словам или побайтно, начиная с определенного адреса. В столбце указаны старшие цифры адреса ОП, а в строке – младшие. Причем информация может вводиться как в символьном, так и в цифровом виде. Что особенно важно при решении задач криптографического кодирования, когда необходимо работать с текстом.

Страница «Микропрограммная память» содержит структуру микропрограммной памяти в виде таблицы, где столбцы – поля микрокоманды, а строки – адреса микрокоманд. При первом обращении к микропрограммной памяти в ней занесены значения, которые используются по умолчанию. Если значение поля не соответствует значению по умолчанию, необходимо указатель «мыши» установить на

требуемое поле и ввести нужное значение. При отладке микропрограммы можно вставлять и удалять микрокоманды. Для этого указатель мыши надо установить на микрокоманду и нажать правую кнопку мыши. При этом будет вызвано контекстное меню и Вам необходимо будет только выбрать нужное действие.

Страница «Таблица преобразования адресов» используется в качестве преобразователя начального адреса при эмуляции команд процессора Intel x86.

5. ПРИМЕРЫ

5.1. Составить и выполнить МК вычисления выражения $(BX - CX) * 4 \rightarrow CX$

a	b	ma	mb	mem	src	sh	n	alu	ccx	f	dst	wm	jfi	cc	cha	const
3	1	0	0	0	1	8	2	2	1	0	4	0	0	0	7	0000

	BX	CX
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1
26	1	1
27	1	1
28	1	1
29	1	1
30	1	1
31	1	1
32	1	1
33	1	1
34	1	1
35	1	1
36	1	1
37	1	1
38	1	1
39	1	1
40	1	1
41	1	1
42	1	1
43	1	1
44	1	1
45	1	1
46	1	1
47	1	1
48	1	1
49	1	1
50	1	1
51	1	1
52	1	1
53	1	1
54	1	1
55	1	1
56	1	1
57	1	1
58	1	1
59	1	1
60	1	1
61	1	1
62	1	1
63	1	1
64	1	1
65	1	1
66	1	1
67	1	1
68	1	1
69	1	1
70	1	1
71	1	1
72	1	1
73	1	1
74	1	1
75	1	1
76	1	1
77	1	1
78	1	1
79	1	1
80	1	1
81	1	1
82	1	1
83	1	1
84	1	1
85	1	1
86	1	1
87	1	1
88	1	1
89	1	1
90	1	1
91	1	1
92	1	1
93	1	1
94	1	1
95	1	1
96	1	1
97	1	1
98	1	1
99	1	1
100	1	1

R-S-1+C0

SDA:=ALU * 4

$$\text{SDA} \rightarrow \text{P3Y}(\text{CX})$$

5.2. Составить и выполнить МК логического умножения регистра CX с константой равной Fh, и результат записать в ОП по адресу 120h:

(CX & Fh) \rightarrow RAM[120h].

МК0. В ARAM записать адрес 120h.

a	b	ma	mb	mem	src	sh	n	alu	ccx	f	dst	wm	jfi	cc	cha	const
0	0	0	0	0	5	0	0	6	0	0	0	2	0	0	7	0120

CONST, RGB

SDA → ARAM

MKL, CX & Fh \rightarrow RGW

a	b	ma	mb	mem	src	sh	n	alu	ccx	f	dst	wm	jfi	cc	cha	const
0	1	0	0	7	5	0	0	9	0	0	0	1	5	0	7	000F

$$\text{CX}$$

CONST, RGB

R&S

SDA \rightarrow RGW

3π ОП

STOP

5.3. Прочитать слово из RAM по адресу 40h в регистр DX: $\text{RAM}[40\text{h}] \rightarrow \text{DX}$
и вычислить выражение $(\text{DX} + 100\text{h}) / 8 \rightarrow \text{DX}$

МК0. В ARAM записать адрес 40h.

a	b	ma	mb	mem	src	sh	n	alu	ccx	f	dst	wrm	jfi	cc	cha	const
0	0	0	0	0	5	0	0	6	0	0	0	2	0	0	7	0040

МК1. Чтение ОП. Запись в РЗУ (DX) из RGR слова.

a	b	ma	mb	mem	src	sh	n	alu	ccx	f	dst	wrm	jfi	cc	cha	const
0	2	0	0	5	1	0	0	6	0	0	1	0	0	0	7	0

DX

Чт ОП

RGR → РЗУ(DX)

МК2. $DX := (DX + 100h) / 8$

a	b	ma	mb	mem	src	sh	n	alu	ccx	f	dst	wrm	jfi	cc	cha	const
0	2	0	0	0	5	1	3	3	0	0	4	0	5	0	4	0100

SDA:=ALU / 8 R + S + C0

5.4. Умножить содержимое BX на 10.

Умножение на некоторую константу можно свести к операциям сложения, вычитания и сдвигов, если константу представить степенями числа 2. В данном случае $BX * 10 = BX * (8 + 2) = (BX * 4 + BX) * 2$

МК0. $AX := 4 * BX$

a	b	ma	mb	mem	src	sh	n	alu	ccx	f	dst	wrm	jfi	cc	cha	const
3	0	0	0	0	1	8	2	6	0	0	4	0	0	0	7	0000

МК1. $BX := (AX + BX) * 2$

a	b	ma	mb	mem	src	sh	n	alu	ccx	f	dst	wrm	jfi	cc	cha	const
0	3	0	0	0	1	8	1	3	0	0	4	0	0	0	7	0000

6. ВАРИАНТЫ ЗАДАНИЙ

Написать и отладить микропрограммы, которые вычисляют следующие выражения:

1. $(AX - BX) / 16 \& [100h] + 31h \rightarrow [100h]$
2. $(DX \& F0h) * 2 + CX - [80h] \rightarrow [80h]$
3. $(BX - [200h]) \& F0h + AX * 16 \rightarrow [200h]$
4. $[300h] + (SI + 35h) / 2 - AX \rightarrow AX$
5. $([160h] \& F00h) + (DX - 10h) \rightarrow [160h]$
6. $(AX - DX) / 32 + ([200h] \vee F0F0h) \rightarrow [200h]$
7. $(DX * 2 + 47h) \bmod 2 (DI - [130h]) / 8 \rightarrow DI$
8. $CX - (\text{not } (AX \& FF0h) + [140h]) / 16 \rightarrow CX$
9. $32 * (DI \& F0F0h) + [150h] - BX \rightarrow BX$
10. $(([160h] \& \text{not}(SI)) + DI) * 16 - 35h \rightarrow [160h]$

$$11. DX - (FFh \& \text{not}(CX) + [170h]) / 2 \rightarrow DX$$

$$12. (F000h \& DI) / 64 + [100h] - SI \rightarrow SI$$

Примечание. В квадратных скобках указан адрес ячейки ОП, в которой находится операнд.

7. СОДЕРЖАНИЕ ОТЧЕТА

Отчет состоит из следующих разделов:

- 1) цель работы;
- 2) структура микропрограммируемого процессора;
- 3) структура микрокоманды с описанием назначения ее полей;
- 4) микропрограмма выполнения заданного выражения;
- 5) исходные данные для тестирования;
- 6) трасса выполнения выражения, т.е. протокол (в виде таблицы) изменения состояний основных элементов модели, выводимых на экран в режиме МИКРОКОМАНДА;
- 7) проверка вычисления выражения вручную по микрокомандам.

8. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите структуру микропрограммируемого процессора и назначение основных элементов.
2. Опишите структуру микрокоманды и функции ее полей.
3. Какие действия выполняются по МК, полностью сформированной из значений по умолчанию?
4. Как работать с оперативной памятью?
5. Что находится в СМК?
6. Что такое АЛУ? Как оно работает?
7. Зачем в микропроцессоре два регистра флажков?
8. Перечислите источники операндов для АЛУ
9. Что такое комбинационный сдвигатель и как он работает?
10. Каково назначение блока рабрчего регистра?