

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Лиодинович
Должность: Ректор
Дата подписания: 06.10.2025 22:12:05
Уникальный программный ключ:
5cf0d6f89e80f49a334f6a4ba78e91f37326b9926

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Дагестанский государственный технический университет»

УЧЕБНО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ по дисциплине «Организация ЭВМ,
вычислительных систем и комплексов» для студентов направления подготовки
бакалавров 09.03.01 «Информатика и вычислительная техника»

Махачкала 2020 г.

УДК 381.3

Методические указания к выполнению лабораторных работ по дисциплине «Организация ЭВМ, вычислительных систем и комплексов» для студентов направления подготовки бакалавров 09.03.01 «Информатика и вычислительная техника». - Махачкала: ДГТУ, 2020. - 38 с.

Методические указания предназначены для выполнения цикла лабораторных работ с использованием учебного эмулятора простейшей ЭВМ, реализованного в виде пакета прикладных программ (разработчик пакета СПб НИУ ИТМО).

Составитель: Фейламазова С. А., ст. препод. каф УиИТСиВТ

Рецензенты:

Рецензенты: доцент кафедры «Прикладной математики и информатики»,
к.т.н., Канаев М.М.

Программист 1 категории отдела информатизации образования ИВЦ ДГУ
Магомедова С.Р.

Печатается согласно постановлению Ученого совета Дагестанского государственного технического университета от

1. ОБЩАЯ ХАРАКТЕРИСТИКА БАЗОВОЙ ЭВМ.

1.1. Назначение базовой ЭВМ.

Базовая ЭВМ - это простая гипотетическая машина, обладающая типичными чертами многих конкретных микро-ЭВМ. Знание принципов построения и функционирования этой ЭВМ будет хорошей базой для освоения микропроцессорных систем любых типов и моделей. Прототипом базовой ЭВМ была выбрана простая модель, так как начинать изучение ЭВМ лучше всего с простых моделей. Базовая ЭВМ моделируется с помощью программы "basepc", разработанной в ЛИТМО (ныне СПб НИУ ИТМО) на кафедре "Вычислительной техники".

1.2. Структура базовой ЭВМ.

На рис.1.1. приведена укрупненная структура базовой ЭВМ. Это одноадресная ЭВМ, работающая с 16-ти разрядными словами. В ней реализованы два вида адресации: прямая и косвенная.

Сначала рассмотрим составные части базовой ЭВМ и их характеристики.

Оперативная память состоит из 2048 ячеек (16-ти битовых) с шестнадцатиричными адресами: 000, 001, 002, ..., 7FF. Восемь ячеек памяти с адресами: 008, 009, 00A, ..., 00F - несколько отличаются от остальных. Эти ячейки лучше использовать в циклических программах. Если произвести косвенное адресование к какой-либо из этих ячеек, то сначала ее содержимое будет использовано в качестве адреса операнда, а затем оно автоматически увеличится на единицу. При прямом адресовании индексные ячейки работают как обычные ячейки.

Процессор состоит из ряда регистров, арифметико-логического устройства и устройства управления (УУ).

Счетчик команд (СК) служит для организации обращений к ячейкам памяти, в которых хранятся команды программы. После исполнения любой команды СК автоматически увеличивает свое значение на 1 и указывает адрес ячейки памяти, содержащий следующую команду программы. СК имеет 11 двоичных разрядов.

Регистр адреса (РА) содержит адрес ячейки памяти, к которой в данный момент обращается ЭВМ за командой или данными. РА также имеет 11 двоичных разрядов.

Регистр команд (РК) имеет 16 двоичных разрядов и используется для хранения кода команды, непосредственно исполняемой машиной.

Регистр данных (РД) используется для временного хранения 16-ти разрядных слов при обмене информацией между памятью и процессором.

Аккумулятор (А) используется для хранения одного из операндов и результата операции при выполнении арифметических и логических операций. Он имеет 16 двоичных разрядов и является одним из главных элементов процессора.

Регистр переноса (С) - это одноразрядный регистр, выступающий в качестве продолжения аккумулятора, и заполняющийся при переполнении А. Этот регистр используется при выполнении сдвигов.

Арифметико-логическое устройство (АЛУ) может выполнять такие арифметические операции, как сложение и сложение с учетом переноса, полученного в результате выполнения предыдущей операции. Кроме того, оно способно выполнять операции логического умножения, инвертирования, циклического сдвига.

Клавишный регистр (КР) – это шестнадцатиразрядный регистр для набора кодов адресов и данных.

1.3. Система команд базовой ЭВМ

ЭВМ способна понимать и выполнять точно определенный набор команд. При составлении программы пользователь ограничен этими командами. Команды можно разделить на три группы:

- обращения к памяти (адресные команды);
- обращения к регистрам (регистровые или безадресные команды);
- команды ввода-вывода.

Команды обращения к памяти предписывают машине производить действия с содержимым ячейки памяти, адрес которой указан в адресной части команды.

Безадресные команды выполняют различные действия без ссылок на ячейку памяти. В качестве примера можно привести команды: CLA (очистить аккумулятор), HLT (остановить выполнение программы) и др.

Команды ввода-вывода осуществляют обмен данными между процессором и внешними устройствами ЭВМ.



Рис. 1.1

В табл. 1.1. приведен полный перечень команд базовой ЭВМ.

СИСТЕМА КОМАНД БАЗОВОЙ ЭВМ

Таблица 1.1.

Наименование	Мнемон. обозн.	Код	Описание
1	2	3	4
Адресные команды			
Логическое умножение	AND M	1XXX	(M) & (A) → A
Пересылка	MOV M	3XXX	(A) → M
Сложение	ADD M	4XXX	(M)+(A) → A
Сложение с переносом	ADC M	5XXX	(M)+(A) +(C)→A

Продолжение таблицы 1.1

1	2	3	4
Вычитание	SUB M	6XXX	$(A)-(M) \rightarrow A$
Переход, если перенос	BCS M	8XXX	Если $(C)=1$, то $M \rightarrow CK$
Переход, если плюс	BPL M	9XXX	Если $(A) \geq 0$, то $M \rightarrow CK$
Переход, если минус	BMI M	AXXX	Если $(A) < 0$, то $M \rightarrow CK$
Переход, если ноль	BEQ M	BXXX	Если $(A)=0$, то $M \rightarrow CK$
Безусловный переход	BR M	CXXX	$M \rightarrow CK$
Обращение к п/программе	JSR M	2XXX	$(CK) \rightarrow M, M+1 \rightarrow CK$
Приращение и пропуск	ISZ M	0XXX	$(M)+1 \rightarrow M$; если $(M) \geq 0$, то $(CK)+1 \rightarrow CK$
Безадресные команды			
Очистка аккумулятора	CLA	F200	$0 \rightarrow A$
Очистка регистра переноса	CLC	F300	$0 \rightarrow C$
Инверсия аккумулятора	CMA	F400	$NE(A) \rightarrow A$
Инверсия регистра переноса	CMC	F500	$NE(C) \rightarrow C$
Циклический сдвиг влево на 1 разряд	ROL	F600	Содержимое A и C сдвигается влево, $A(15) \rightarrow C$, $(C) \rightarrow A(0)$
Циклический сдвиг вправо на 1 разряд	ROR	F700	Содержимое A и C сдвигается вправо, $A(0) \rightarrow C$, $(C) \rightarrow A(15)$
Инкремент аккумулятора	INC	F800	$(A)+1 \rightarrow A$
Декремент аккумулятора	DEC	F900	$(A)-1 \rightarrow A$
Останов	HLT	F000	
Нет операции	NOP	F100	
Разрешение прерывания	EI	FA00	
Запрещение прерывания	DI	FB00	
Команды ввода-вывода			
Очистка флага	CLF B	E0XX	$0 \rightarrow \text{флаг устройства B}$
Опрос флага	TSF B	E1XX	Если (флаг устр. B)=1, то $(CK)+1 \rightarrow CK$
Ввод	IN	E2XX	$(B) \rightarrow A$
Вывод	OUT	E3XX	$(A) \rightarrow B$
Примечание: (M), (A), (CK), (C), (B) – содержимое: ячейки с адресом M, аккумулятора, счетчика команд, регистра переноса и регистра данных устройства ввода-вывода с адресом B соответственно. XXX - адрес ячейки памяти. XX - адрес устройства ввода-вывода. Описание команд приводится для случая прямой адресации.			

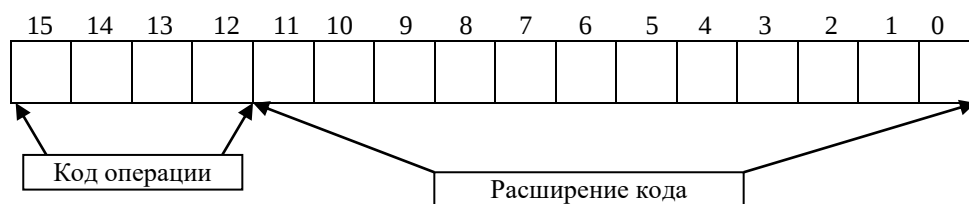
1.3.1. Форматы команд и способы адресации.

В базовой ЭВМ предусмотрены три формата 16-ти битовых однословных команд с 4-х битовым кодом операции (рис. 1.2.).

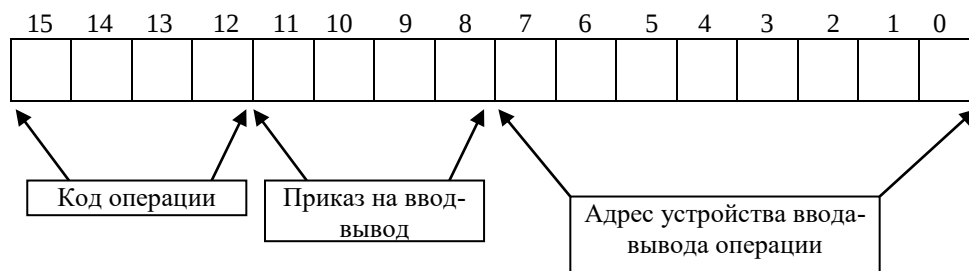
В командах обращения к памяти на адрес отведено 11 бит, что позволяет прямо адресоваться к любой из 2048 ячеек памяти ($2^{11}=2048$). Бит вида адресации при прямой адресации должен содержать 0. Если же в этом бите содержится 1, то адрес, размещенный в адресной части команды, указывает на ячейку памяти, в которой находится адрес операнда (косвенная адресация).



а)



б)



в)

Рис. 1.2. Форматы команд: а) - адресных, б) - безадресных, в) - команд ввода-вывода.

При косвенной адресации вокруг адреса добавляются скобки и в мнемонике и в описании команды. Например, команда ADD 25 приказывает ЭВМ сложить содержимое А с содержимым ячейки 25 (прямая адресация), а команда ADD (25) - сложить содержимое А с содержимым ячейки, адрес которой хранится в ячейке 25 (косвенная адресация).

ADD 25	$(A) + (025) \rightarrow A$	синтаксис для прямой адресации
ADD (25)	$(A) + ((025)) \rightarrow A$	синтаксис для косвенной адресации

1.4. Арифметические операции.

В базовой ЭВМ можно выполнять по командам сложение и вычитание положительных целых чисел и целых чисел со знаком. Умножение и деление можно осуществить программным путем.

Целые неотрицательные числа (положительные числа и нуль) в базовой ЭВМ могут лежать в диапазоне от 0 до $(2^{16}-1)=65535$.

Для представления целых чисел со знаком используются разряды от 0 до 14, а старший разряд (15-тый) служит для кодирования знака. Таким образом, в базовой ЭВМ можно представлять целые числа со знаком в диапазоне от -32768 до +32767 (см. табл.1.2.). Отрицательные числа представляются в дополнительном коде, так как это упрощает выполнение арифметических операций: вычитание заменяется сложением уменьшаемого с дополнительным кодом вычитаемого. Дополнительный код отрицательного целого числа получается по правилу:

- 1) инвертируется код числа;
- 2) к инверсии добавляется единица.

Например, прямой двоичный код единицы 0000000000000001. Чтобы получить дополнительный код нужно к инверсии этого кода добавить единицу:

$$\begin{array}{r}
 1111111111111110 - \text{инверсия } 1 \\
 + \quad 0000000000000001 \\
 \hline
 1111111111111111 - \text{дополнительный код } 1
 \end{array}$$

Десятичные эквиваленты 16-ти битовых двоичных чисел. Табл. 1.2.

Двоичное число	Десятичное число	Двоичное число	Десятичное число
0000 0000 0000 0000	0	1111 1111 1111 1111	-1
0000 0000 0000 0001	+1	1111 1111 1111 1110	-2
0000 0000 0000 0010	+2	1111 1111 1111 1101	-3
0000 0000 0000 0011	+3	1111 1111 1111 1100	-4
.....			
0111 1111 1111 1110	+32766	1000 0000 0000 0001	-32767
0111 1111 1111 1111	+32767	1000 0000 0000 0000	-32768

Сложение целых двоичных чисел со знаком и без знака выполняется в базовой ЭВМ с помощью команды **ADD**.

По команде **INC** к содержимому аккумулятора прибавляется единица, а по команде **DEC** - единица вычитается. Если при этом возникает перенос из старшего разряда А, то в регистр переноса (С) заносится 1, в противном случае в него заносится 0.

По команде **SUB** выполняется операция вычитания. Уменьшаемое находится в аккумуляторе, а вычитаемое - в ячейке с указанным адресом.

1.5. Управление вычислительным процессом, сдвиги и логические операции.

Задача управления вычислительным процессом, т.е. требуемой последовательностью выполнения команд, решается в базовой ЭВМ при помощи команд переходов (BCS, BPL, BMI, BEQ, BR), команд "Приращение и пропуск" (ISZ) и "Останов" (HLT). Все эти команды являются адресными (кроме HLT), т.е. в них указывается адрес той ячейки памяти, из которой должна быть выбрана следующая команда программы при выполнении того или иного условия. Если же условие не выполняется, то должна исполняться команда, расположенная вслед за данной командой управления. Как и в других адресных командах, здесь можно использовать косвенную адресацию. Команды переходов не изменяют состояния аккумулятора и регистра переноса. Они могут лишь изменять содержимое счетчика команд, поместив в него адрес, определяемый адресной частью команды.

BCS M (Переход, если перенос). Переход к команде, расположенной в ячейке с адресом М, если содержимое регистра переноса равно 1.

BPL M (Переход, если плюс). Переход к команде, расположенной в ячейке с адресом М, если содержимое аккумулятора больше или равно нулю, т.е. в его старшем (знаковом) разряде содержится 0.

BMI M (Переход, если минус). Переход к команде, расположенной в ячейке с адресом М, если содержимое аккумулятора меньше нуля, т.е. в его старшем (знаковом) разряде содержится 1.

BEQ M (Переход, если нуль). Переход к команде, расположенной в ячейке с адресом М, если содержимое аккумулятора равно нулю.

BR M (Переход безусловный). Переход к команде, расположенной в ячейке с адресом М, осуществляемый при любых значениях А и С или других регистров базовой ЭВМ.

Команды переходов широко применяются для организации циклических программ, которые используются в тех случаях, когда требуется несколько раз выполнить набор одинаковых действий с различными наборами данных.

Побитовая обработка данных обеспечивается в базовой ЭВМ командами логического умножения, циклических сдвигов, а также командами инвертирования и очистки регистра переноса.

Команда AND M (Логическое умножение) выполняет над каждой парой разрядов содержимого аккумулятора и ячейки М булеву операцию "&".

Результат выполнения команды для каждой пары битов операндов равен единице только тогда, когда оба бита равны единице, а в остальных случаях бит результата равен нулю. Команда позволяет выделить или очистить определенные биты слова.

Команды ROL (циклический сдвиг влево на один разряд) и ROR (циклический сдвиг вправо на один разряд) замыкают аккумулятор и регистр переноса в кольцо и сдвигают все биты кольца на один разряд влево или вправо (см. рис. 1.3.).

		Регистр переноса	Аккумулятор
а)	До сдвига	0	1011100000101011
	После сдвига	1	0111000001010110
б)	До сдвига	0	1011100000101011
	После сдвига	1	0101110000010101

Рис 1.3. Циклические сдвиги: а) - влево, б) - вправо.

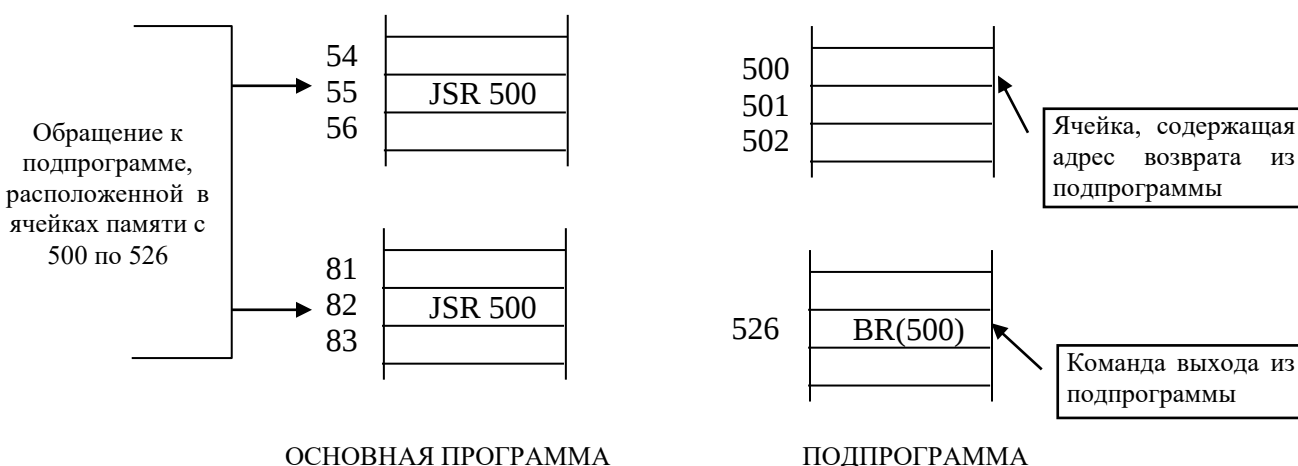
Сдвигами числа влево или вправо можно реализовать операции умножения или деления на два (один сдвиг), на четыре (два сдвига), на восемь (три) сдвига и т.д.

1.6. Подпрограммы.

Достаточно часто встречаются ситуации, когда отдельные части программы должны выполнять одни и те же действия по обработке данных (например вычисление тригонометрической функции). В подобных случаях повторяющиеся части программы выделяют в подпрограмму, а в соответствующие места программы заносят лишь команды обращения к этой подпрограмме. В базовой ЭВМ для этой цели используется команда JSR (Обращение к подпрограмме). Ниже показана часть основной программы, содержащая две команды JSR 500, с помощью которых осуществляется переход к выполнению команд подпрограммы.

По команде JSR 500, расположенной в ячейке 55, выполняется запись числа $55+1=56$ (текущего значения счетчика команд) в ячейку с адресом 500 и запись числа $500+1=501$ в

счетчик команд (адрес первой команды подпрограммы). Таким образом осуществляется переход к выполнению команд подпрограммы. Далее начинается процесс выполнения команд подпрограммы, который завершается командой BR (500), расположенной в ячейке 526. Это команда безусловного перехода с косвенной адресацией предписывает ЭВМ выполнить переход к команде, расположенной по адресу, хранящемуся в 500 ячейке (пересылает содержимое 500 ячейки в СК). Так как в эту ячейку ранее было записано число 56, то будет выполняться команда, находящаяся в ячейке 56, т.е. следующая за обращением к подпрограмме. Аналогично выполняется команда JSR 500, расположенная в ячейке 82 (после выполнения команд подпрограммы будет выполнен переход к ячейке 83).



Таким образом, при оформлении подпрограммы перед ее первой командой следует разместить ячейку, в которую будет пересылаться адрес возврата из подпрограммы. В команде обращения к подпрограмме указывается адрес именно этой ячейки (например адрес М в команде JSR М). Последней командой подпрограммы должна быть команда выхода (команда BR (М) для подпрограммы, размещенной начиная с ячейки М). По ней осуществляется переход к команде, адрес которой сохраняется в начальной ячейке тела подпрограммы.

1.7. Выполнение машинных команд.

В процессе выполнения команд устройство управления ЭВМ производит анализ и пересылку команды, отдельных ее частей (кода операции, признака адресации и адреса) или операнда из одного регистра ЭВМ в другой ее регистр, АЛУ, память или устройства ввода-вывода. Эти действия (микрооперации) протекают в определенной временной последовательности и скоординированы между собой. Для обеспечения такой последовательности в ЭВМ используется генератор тактовых импульсов.

Цикл команды. Для реализации одной команды требуется выполнить определенное количество микрокоманд, каждая из которых инициируется одним тактовым импульсом. Общее число тактовых импульсов, требуемых для выполнения команды, определяет время ее выполнения, называемое циклом команды. Цикл команды обычно включает один или несколько машинных циклов. Устройство управления базовой ЭВМ может находиться в четырех возможных состояниях: выборки команды, выборки адреса, исполнения и прерывания. Длительность каждого из этих четырех состояний определяет время выполнения соответствующего

машинного цикла. Основные действия, выполняемые ЭВМ во время каждого из машинных циклов описаны ниже.

Выборка команды. В данном машинном цикле выполняется чтение команды из памяти и ее частичное декодирование.

1. Содержимое ячейки памяти, на которую указывает регистр адреса, куда заносится содержимое счетчика команд, читается из памяти в регистр данных.

2. Содержимое счетчика команд увеличивается на 1.

3. Содержимое регистра данных пересылается в регистр команд, код операции команды частично декодируется для выявления типа команды (адресная, безадресная или ввода-вывода), анализируется бит признака адресации и происходит подготовка цепей, необходимых для выполнения команды.

Безадресные команды и команды ввода-вывода окончательно исполняются в этом же цикле, т.е. это одноцикловые команды.

4. Выполняются действия по завершению одноцикловой команды.

Выборка адреса. Этот машинный цикл следует за циклом выборки команды для адресных команд с косвенной адресацией (бит вида адресации равен 1). Цикл используется для чтения из памяти адреса операнда, результата или перехода и состоит из следующих шагов.

1. Адресная часть команды пересылается из регистра данных, где пока еще сохраняется копия команды, в регистр адреса.

2. Содержимое ячейки памяти, указываемой регистром адреса, читается в регистр данных. Теперь в этом регистре находится либо адрес операнда, либо адрес результата, либо адрес перехода, который будет использоваться в цикле исполнения команды.

Если косвенно адресуется одна из индексных ячеек (адреса 8, 9,...,F), то цикл выборки адреса операнда (результата) продолжается.

3. Содержимое регистра данных увеличивается на единицу.

4. Измененное содержимое регистра данных пересылается в ячейку памяти по адресу, указываемому регистром адреса.

5. Содержимое регистра данных уменьшается на единицу.

После этой операции в регистре данных восстанавливается значение адреса, находившегося в индексной ячейке до выполнения шага 3. Содержимое же индексной ячейки увеличилось на 1 и при следующем обращении к ней будет выбран новый адрес операнда (результата).

Исполнение. Последовательность действий, выполняемых в этом цикле, определяется типом выполняемой адресной команды.

1. Для команд, при выполнении которых требуется выборка операнда из памяти ЭВМ (AND, ADD, ADC, SUB, ISZ), состояние исполнения используется для чтения операнда в регистр данных и выполнения операции, указываемой кодом операции команды.

2. По команде пересылки (MOV) в этом машинном цикле производится запись содержимого аккумулятора в ячейку памяти с адресом, расположенным в регистре данных. Для этого содержимое регистра данных пересылается в регистр адреса, а содержимое аккумулятора - в регистр данных и далее в ячейку памяти, указываемую регистром адреса.

3. При исполнении команд переходов (BCS, BPL, BMI, BEQ) производится проверка соответствующего условия и пересылка адреса из регистра данных в счетчик команд при выполнении этого условия. Иначе будет выбрана команда, расположенная вслед за командой перехода. При исполнении команды безусловного перехода (BR) пересылка адреса перехода в счетчик команд выполняется без какой-либо проверки.

4. Для команды обращения к подпрограмме (JSR) во время этого машинного цикла осуществляется пересылка содержимого счетчика команд в ячейку памяти, адрес которой содержится в регистре данных (код команды из регистра данных заносится в регистр адреса, и при этом код операции теряется, а остается адрес), и занесение в счетчик команд увеличенного

на единицу содержимого регистра данных (код операции теряется, а остается адрес, увеличенный на единицу).

1.8 ПОРЯДОК РАБОТЫ С МОДЕЛЬЮ БАЗОВОЙ ЭВМ

Базовая ЭВМ моделируется с помощью специальной программы, работающей в среде MS DOS (при работе в операционной системе Windows 7 используется эмулятор DOS). Техническое обеспечение - IBM PC.

Для запуска программы необходимо войти в каталог BEWM и запустить программу "basepc.bat". Программа предусматривает три режима работы: "Базовая ЭВМ", "Работа с ВУ", "Работа с МПУ". После запуска программы необходимо с помощью клавиш "Shift + F4" войти в меню и выбрать режим "Базовая ЭВМ", используя для перемещения курсора клавиши "стрелка вверх", "стрелка вниз" и для выбора режима - "Enter". На экране дисплея появится картинка, соответствующая рис.1.1. В левом нижнем углу имитируется клавишный 16-ти разрядный регистр для набора и ввода данных в ЭВМ. На картинке можно просматривать содержимое памяти, регистра данных, регистра адреса, регистра команд, аккумулятора, регистра переноса и счетчика команд. В нижней и верхней строке экрана выведены подсказки какими клавишами пользоваться для работы с моделью базовой ЭВМ.

Для перемещения в клавишном регистре используются следующие клавиши:

→ - Перемещение указателя на одну позицию вправо. ← - Перемещение указателя на одну позицию влево. 1 - Занесение 1 по текущему положению указателя и перемещение его на следующую позицию.

0 - Занесение 0 по текущему положению указателя и перемещение его на следующую позицию.

В процессе работы также используются клавиши:

F4 - Ввод адреса. По этой клавише содержимое клавишного регистра заносится в счетчик команд.

F5 - Запись. Информация из клавишного регистра заносится в память по текущему содержимому счетчика команд.

F6 - Чтение. Из ячейки памяти (по адресу расположенному в счетчике команд) информация читается в регистр данных.

F7 - Пуск. Действие этой клавиши различно в режимах "РАБОТА" и "ОСТАНОВ". В режиме "РАБОТА" по ней происходит обнуление всех регистров, кроме счетчика команд, и происходит запуск программы на выполнение. В режиме "ОСТАНОВ" происходит очистка регистров, кроме счетчика команд, а запуск не производится.

F8 - Продолжение. В режиме "ОСТАНОВ" происходит исполнение одной команды, адрес которой определяется содержимым СК, а в режиме "РАБОТА" продолжение выполнения программы с адреса в счетчике команд.

F9 - Клавиша, управляющая переключением режима работы базовой ЭВМ. Производит переключение режимов "РАБОТА" и "ОСТАНОВ".

F10 - Выход из базовой ЭВМ.

Shift+F4 - Смена маски.

ЛАБОРАТОРНАЯ РАБОТА № 1

Исследование работы ЭВМ при выполнении линейных программ.

Цель работы - изучение приемов работы на базовой ЭВМ и исследование порядка выполнения арифметических команд и команд пересылки.

Порядок выполнения работы.

1. Изучить раздел 1 данных методических указаний.
2. Занести в память модели ЭВМ заданный вариант программы и, выполняя ее по командам, заполнить таблицу результатов (по форме табл. 1.3.).

Табл. 1.3.

Выполняемая команда		Содержимое регистров процессора после выполнения команды						Ячейка, содержимое которой изменилось после вып. команды	
Адрес	Код	СК	РА	РК	РД	А	С	Адрес	Новый код
XXX	XXXX	XXX	XXX	XXXX	XXXX	XXXX	X	XXX	XXXX

Содержание отчета по работе

1. Титульный лист.

2. Цель работы.

Текст исходной программы по следующей форме:

"Адрес"	"Код команды"	"Мнемоника"	"Комментарии"
21	4015	ADD 15	(A)+(15)--> A

3. Таблица с результатами выполнения программы.

4. Описание программы.

ВАРИАНТЫ ПРОГРАММ

(первая команда программы помечена знаком "+")

Адрес	В а р и а н т ы п р о г р а м м					
	1	2	3	4	5	6
017	0000	0000	+F200	0000	0000	0000
018	01AA	+F200	4023	0255	01AA	0000
019	7099	4021	4021	7099	7099	+F200
01A	0255	1022	3020	0000	+F200	4023
01B	0000	3024	F200	+F200	4018	6024
01C	+F200	F200	4022	4024	1024	3018
01D	4018	4023	1020	6018	3023	F200
01E	401A	4024	3020	301A	F200	4022
01F	301B	3024	F000	F200	4019	1018
020	F200	F000	0000	4019	6023	3018
021	4019	01AA	7099	101A	3023	F000
022	101B	0255	01AA	301A	F000	01AA
023	301B	7099	0255	F000	0000	0255
024	F000	0000	0000	03CF	0255	7099

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Структура базовой ЭВМ и функции ее компонентов (ОП, РА, РД, А, С, РК, КР, АЛУ, УУ, СК).
2. Форматы команд базовой ЭВМ.
3. Форматы данных базовой ЭВМ.
4. Порядок выполнения работы.
5. Объяснение работы программы.

ЛАБОРАТОРНАЯ РАБОТА № 2

Исследование работы ЭВМ при выполнении разветвляющихся программ.

Цель работы - изучение команд переходов, способов организации разветвляющихся программ и исследование порядка функционирования ЭВМ при выполнении таких программ.

Подготовка к выполнению работы.

1. Переписать заданную программу, дополнив ее мнемоническими кодами команд и комментариями.
2. Заполнить теоретически таблицу (типа табл.1.3), мысленно выполняя за базовую ЭВМ заданный вариант программы.

Порядок выполнения работы.

Занести в память ЭВМ заданный вариант программы и заполнить таблицу результатов (типа табл.1.3), выполняя эту программу по командам. При этом экспериментальная и теоретическая таблицы выполнения программы должны совпасть. В случае несовпадения найти ошибки.

Содержание отчета по работе.

Титульный лист, цель работы, текст программы с мнемоническими обозначениями команд и комментариями, таблица с результатами выполнения программы.

ВАРИАНТЫ ПРОГРАММ

(первая команда программы помечена знаком "+")

Адрес	В а р и а н т ы п р о г р а м м					
	1	2	3	4	5	6
016	0255	+C01A	0F0A	0000	0000	0000
017	0FA7	FFD0	F0F6	+C01B	5417	+C01B
018	+F200	002F	F000	0019	+F200	001B
019	4016	0000	+F200	F000	4022	FFD0
01A	4017	F200	4016	1000	4023	002F
01B	9020	4017	4017	F200	9020	F200
01C	F200	4018	B020	4019	F200	4019
01D	3022	A020	F200	401A	3017	401A
01E	F100	F200	3018	8022	F100	A021
01F	F000	F100	F000	F200	F000	F200
020	3022	3022	4016	3018	3017	F100
021	C01F	F000	3018	301A	C01F	3023
022	1111	CCCC	C01F	F000	FFD1	F000
023	0000	0000	0000	0000	002F	CCCC

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите команды, с помощью которых организуется разветвление программы и объясните как они работают.
2. Порядок выполнения работы.
3. Объяснение работы программы.

ЛАБОРАТОРНАЯ РАБОТА №3

Исследование работы ЭВМ при выполнении циклических программ.

Цель работы - изучение способов организации циклических программ и исследование порядка функционирования ЭВМ при выполнении циклических программ.

Подготовка к выполнению работы. Переписать заданную программу, дополнив ее мнемоническими кодами команд и комментариями.

Порядок выполнения работы. Занести в память ЭВМ заданный вариант программы и заполнить таблицу результатов (типа табл.1.3), выполняя эту программу по командам.

ВАРИАНТЫ ПРОГРАММ

(первая команда программы помечена знаком "+")

Адрес	В а р и а н т ы п р о г р а м м					
	1	2	3	4	5	6
00A	0000	0000	0000	0011	0000	0000
00B	0000	0000	0000	0000	001D	0000
00C	0000	0000	001A	0000	0000	0000
00D	0000	0000	0000	0000	0000	0012
00E	001B	0000	0000	0000	0000	0000
00F	0000	001B	0000	0000	0000	0000
010	0000	0000	0000	0000	0000	0000
011	0000	0000	+F200	71BC	FFFD	0000
012	FFFD	FFFD	480C	ABBA	+F200	0000
013	+F200	+F200	9016	63CD	480B	0707
014	480E	480F	401D	FFFD	9019	0000
015	B018	A018	301D	0000	F200	FFFD
016	4011	4011	0019	+F200	F800	+F200
017	3011	3011	C011	480A	401C	480D
018	0012	0012	F000	A01D	301C	B01A
019	C013	C013	FFFD	F200	0011	C01D
01A	F000	F000	8778	F800	C012	F800
01B	0375	7F01	7777	4015	F000	4011
01C	0000	DECA	8788	3015	0000	3011
01D	F0EB	30AE	0000	0014	B0BA	0015
01E	0000	0000	0000	C016	5B1B	C016
01F	0000	0000	0000	F000	CF01	F000

Содержание отчета по работе.

Титульный лист, цель работы, текст программы с комментариями, таблица с результатами выполнения программы и описание программы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как организуются программные циклы в базовой ЭВМ?

2. Порядок выполнения работы.
3. Объяснение работы программы.

ЛАБОРАТОРНАЯ РАБОТА № 4

Исследование работы ЭВМ при выполнении комплекса программ.

Цель работы - изучение способов связи между программными модулями, команды обращения к подпрограмме и исследование порядка функционирования ЭВМ при выполнении комплекса взаимосвязанных программ.

Подготовка к выполнению работы. Переписать заданную программу и подпрограмму, дополнив их мнемоническими кодами команд и комментариями.

Порядок выполнения работы. Занести в память ЭВМ заданный вариант программного комплекса и заполнить таблицу результатов (типа табл. 1.3), выполняя этот комплекс по командам.

Содержание отчета по работе.

Текст программы и подпрограммы с мнемоническими обозначениями и комментариями, таблица с результатами выполнения комплекса программ и описания программы и подпрограммы.

ВАРИАНТЫ КОМПЛЕКСА ПРОГРАММ
(первая команда программы помечена знаком "+")

Адрес	В а р и а н т ы п р о г р а м м					
	1	2	3	4	5	6
00A	0011	0000	0000	0000	0000	0000
00B	0000	001A	0000	0000	0000	0000
00C	0000	0000	0012	0000	0000	0000
00D	0000	0000	0000	0019	0000	0000
00E	0000	0000	0000	0000	0011	0000
00F	0000	0000	0000	0000	0000	0010
010	0000	0000	0000	+F200	0000	F200
011	ABBA	FFFD	FFFD	480D	6789	4816
012	63CD	+F200	0000	B014	5ACA	F800
013	71BC	480B	0707	2045	8A7C	+F200
014	FFFD	9016	0000	0018	FFFD	480F
015	0000	2045	0000	C010	+F200	9017
016	+F200	0011	+F200	F000	480E	2045
017	480A	C012	480C	0000	A019	001A
018	A01A	F000	B01A	FFFD	2045	C013
019	2045	0000	C01B	801B	0014	F000
01A	0014	CF01	2045	0000	C015	FFFD
01B	C016	B0BA	0011	11FF	F000	0000
01C	F000	5B1B	C016	0000	0000	0000
01D	0000	0000	F000	0000	0000	0000
045	0000	0000	0000	0000	0000	0000
046	F200	F200	F200	F200	F200	F200
047	F800	F800	F800	F800	F800	F800
048	4015	4019	4015	4017	4010	401B
049	3015	3019	3015	3017	3010	301B
04A	C845	C845	C845	C845	C845	C845

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как организуется вызов подпрограмм в базовой ЭВМ?
2. Порядок выполнения работы.

3. Объяснение работы программы.

2. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА В БАЗОВОЙ ЭВМ

Переход к моделированию режима ввода-вывода базовой ЭВМ осуществляется вхождением в меню (нажатием клавиш "Shift+F4") и выбором из меню режима "Работа с ВУ" (клавиши "стрелка вниз", "стрелка вверх" - для перемещения курсора и клавиша "Enter" - для выбора режима), после чего на экране появляется картинка, соответствующая рис. 2.1..

2.1. Устройства ввода-вывода базовой ЭВМ.

В базовой ЭВМ используются простейшие внешние устройства: одно устройство вывода (ВУ1), одно устройство ввода (ВУ2) и одно устройство ввода-вывода (ВУ3). Между внешними устройствами и процессором включены простейшие контроллеры, каждый из которых содержит: регистр данных для обмена данными между ВУ и процессором; дешифратор адреса и приказов, позволяющий выделить обращение к данному ВУ и декодирующий приказы от процессора на выполнение тех или иных операций; регистр состояния, в котором хранится информация о готовности ВУ к обмену данными с процессором. Здесь используются однобитовые регистры готовности, называемые флагом. Контроллеры ВУ связаны с процессором шинами, служащими для передачи информации, адресов, сигналов управления и сведений о состоянии внешних устройств.

2.2. Программно-управляемая передача данных.

При использовании программно-управляемого обмена должна быть составлена программа, обеспечивающая пересылку данных из памяти ЭВМ в аккумулятор и далее в регистр данных контроллера ВУ или из регистра данных контроллера ВУ в аккумулятор и затем в память ЭВМ.

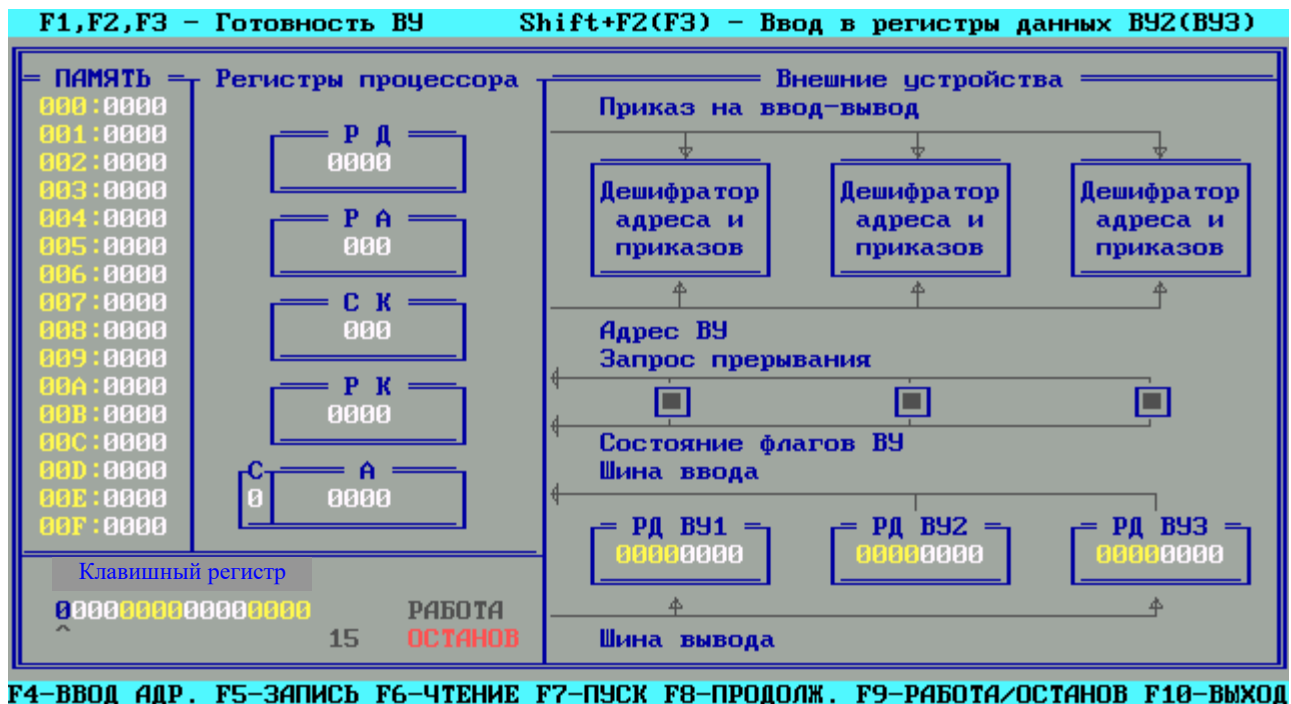


Рис. 2.1.

В такой программе можно реализовать один из трех видов обмена: синхронный, асинхронный и по прерыванию. Синхронный обмен не рассматривается в данных методических указаниях.

К командам ввода-вывода относятся: CLF, TSF, IN, OUT. Команда CLF В служит для установки в нуль флага ВУ с адресом В.

Команда TSF В служит для проверки готовности к обмену ВУ с адресом В. Если флаг этого ВУ сброшен (ВУ не готово к обмену), то выполняется команда, расположенная вслед за TSF В. В противном случае эта команда пропускается и выполняется команда, расположенная через одну за TSF В.

Команда IN В служит для пересылки содержимого регистра данных контроллера ВУ с адресом В в восемь младших разрядов аккумулятора.

Команда OUT В служит для пересылки содержимого восьми младших разрядов аккумулятора в регистр данных контроллера ВУ с адресом В.

Для организации обмена с ВУ в состав устройства управления базовой ЭВМ включены два устройства: регистр состояния внешних устройств и контроллер прерываний. Связь контроллеров ВУ с этими устройствами осуществляется по линиям "Состояние флагов ВУ" и "Запрос прерывания". Данные передаются по шинам ввода и вывода.

2.3. Асинхронный обмен.

Программа такого обмена строится следующим образом: сначала проверяется готовность ВУ к обмену и, если оно готово, дается команда на обмен. ВУ сообщает о готовности установкой флага.

Пример 2.1. С помощью ВУ-2 записать в ячейку 006 коды символов слова "ДА".

Для обозначения символов будем использовать следующие коды:

Символ	А	Б	Д	Е	И	Й	К	М	Н	О	П	Р	Т	Ч	Ш	Ъ	Ы	Я
Код	E1	E2	E4	E5	E9	EA	EC	ED	EE	EF	F0	F2	F4	FE	FB	F8	F9	F1

Программа для выполнения этого задания имеет вид:

Адрес	Содержимое		Комментарий
	Код	Мнемоника	
05	FFF8		Константа -8, используемая для сдвига Ячейка для записи слова "ДА".
06	0000		
..			
20	E102	TSF 2	Опрос флага контроллера ВУ-2 и повторение этой операции, если ВУ-2 не готово к обмену (флаг=0).
21			
	C020	BR 20	Эта команда выполняется, если флаг=1 (команда BR 20 в этом случае пропускается). Содержимое РД ВУ-2 пересылается в 8 младших разрядов ак- кумулятора.
22	E202	IN 2	
23	E002	CLF 2	Сброс готовности ВУ-2 (очистка флага ВУ-2)
24	F600	ROL	Код первого символа сдвигается на 8 разрядов влево и освобождает место для ввода следующе- го символа.
25	0005	ISZ 5	
26	C024	BR 24	Опрос флага контроллера ВУ-2 и повторение этой операции, если ВУ-2 не готово к обмену (флаг=0).
27	E102	TSF 2	
28	C027	BR 27	Ввод кода символа содержащегося в РД ВУ-2. Эта команда выполняется, если флаг=1.
29	E202	IN 2	

2A	E002	CLF 2	Сброс готовности ВУ-2.
2B	3006	MOV 6	Пересылка кода слова "ДА" в ячейку 006.
2C	F000	HLT	Останов ЭВМ.

Две первые команды этой программы "заставляют" ЭВМ ожидать готовности ВУ-2 к выдаче данных. Поэтому до первого нажатия клавиши F2 необходимо занести в РД ВУ-2 код символа "Д" (для ввода данных в РД ВУ-2 необходимо нажать клавиши Shift+F2 и осуществить ввод с клавиатуры, после чего нажать клавишу "Enter"). Затем нажать клавишу F2 и (после сброса флага ВУ-2) приступить к набору символа "А". В процессе набора этого кода ЭВМ занята сдвигом кода символа "Д" в старшие разряды аккумулятора, чтобы подготовиться к приему символа "А", и ожиданием поступления нового сигнала готовности ВУ-2 к выдаче информации. После набора кода символа "А" (набирается аналогично символу "Д") можно сразу нажимать клавишу F2. Теперь в аккумулятор переписывается все слово "ДА", а затем оно переписывается в ячейку 006 и выполнение программы прекратится.

Легко заметить, что при асинхронном обмене ЭВМ должна тратить время на ожидание момента готовности, а так как готовность проверяется программным путем (команда TSF), то в это время ЭВМ не может выполнить никакой другой работы по преобразованию данных.

2.4. Обмен по прерыванию программы.

Этот вид обмена отличается от асинхронного тем, что сигнал готовности ВУ к обмену анализируется не программным, а аппаратным путем. ЭВМ может выполнять любую не связанную с обменом программу (будем называть ее основной), а когда из ВУ по линии "Запрос прерывания" (рис. 2.1.) поступит сигнал готовности ВУ к приему или выдаче информации, прервать (приостановить) выполнение этой программы на время выполнения программы обмена данными. Все эти действия осуществляются с помощью контроллера прерываний, входящего в состав устройства управления базовой ЭВМ.

Команды EI (Разрешение прерывания) и DI (Запрещение прерывания) переводят контроллер прерывания в одно из двух состояний, в которых он соответственно реагирует или не реагирует на сигналы готовности ВУ, передаваемые по линии "Запрос прерывания".

Если контроллер прерываний установлен в состояние разрешения прерывания, то после выполнения каждой команды программы управление передается контроллеру прерываний, который анализирует состояние линии "Запрос прерывания". При отсутствии на линии сигнала запроса прерывания начинается выборка и исполнение следующей команды и данный шаг повторяется. При наличии запроса прерывания контроллер прерывания переходит в состояние запрещения прерывания, в ячейку с адресом 000 заносится содержимое СК (адрес следующей команды основной программы, которая выполнялась бы при отсутствии запроса прерывания), и управление передается команде, расположенной в ячейке 001. Так происходит переход к программе обработки прерывания, первая команда которой должна располагаться в ячейке 001.

Пример 2.2. Составить программу, которая периодически (с периодом в три цикла команды) наращивает на 1 содержимое аккумулятора. Восемь младших разрядов аккумулятора должны выводиться на ВУ-1 по его запросу (нажатие клавиши F1), а по запросу ВУ-2 код, набранный в регистре данных ВУ-2 должен помещаться в ячейку 25.

Основная программа решения задачи примера 2.2.

Адрес	Содержимое		Комментарии
	Код	Мнемон.	
20	FA00	EI	Установка состояния разрешения прерывания
21	F200	CLA	Очистка аккумулятора
...			
22	F800	INC	Цикл для наращивания содержимого аккумулятора
23	F100	NOP	
24	C022	BR 22	
...			
25	0000		Ячейка для хранения кодов, поступающих с ВУ-2

Если команды этой программы занести в память базовой ЭВМ, установить в СК пусковой адрес 20 и нажать клавишу F7 (в режиме "РАБОТА", то начнет выполняться бесконечный цикл наращивания содержимого аккумулятора. Когда же будет нажата любая из трех клавиш F1, F2, F3, то будет выполнен переход к программе обработки прерываний.

Программа обработки прерываний для примера 2.2.

Адрес	Содержимое		Комментарии
	Код	Мнемон.	
00	0000		Ячейка для хранения адреса возврата (адрес заносится, когда происходит прерывание).
01	C030	BR 30	Первая команда подпрограммы - переход к основному ее тексту, размещенному в ячейках 30...4C.
30	304B	MOV 4B	Сохранение в буферных ячейках 4B и 4C содержимого аккумулятора и регистра переноса
31	F600	ROL	
32	304C	MOV 4C	Опрос флага ВУ-2. Если он сброшен, то переход к опросу флага ВУ-1. В противном случае <u>переход на ввод данных из ВУ-2.</u>
33	E102	TSF 2	
34	C036	BR 36	Опрос флага ВУ-1. Если он сброшен, то переход к сбросу флага ВУ-3. В противном случае <u>переход на вывод данных в ВУ-1.</u>
35	C039	BR 39	
36	E101	TSF 1	Ввод данных из ВУ-2, пересылка их в ячейку 25 сброс флага ВУ-1, переход к восстановлению содержимого регистров и выходу из подпрограммы.
37	C043	BR 43	
38	C03E	BR 3E	Пересылка в аккумулятор содержимого буферной ячейки 4B, вывод на ВУ-1 восьми младших разрядов аккумулятора, сброс флага ВУ-1, переход к восстановлению А и С и выходу из подпрограммы.
39	F200	CLA	
3A	E202	IN 2	Восстановление содержимого регистра переноса и аккумулятора.
3B	E002	CLF 2	
3C	3025	MOV 25	Возобновление состояния разрешения прерывания и <u>выход из подпрограммы.</u>
3D	C044	BR 44	
3E	F200	CLA	Ячейки для сохранения содержимого аккумулятора и регистра переноса.
3F	404B	ADD 4B	
40	E301	OUT 1	
41	E001	CLF 1	
42	C044	BR 44	
43	E003	CLF 3	
44	F200	CLA	
45	404C	ADD 4C	
46	F700	ROR	
47	F200	CLA	
48	404B	ADD 4B	
49	FA00	EI	
4A	C800	BR (0)	
4B	0000		
4C	0000		

ЛАБОРАТОРНАЯ РАБОТА № 5

Исследование работы ЭВМ при асинхронном обмене данными С ВУ

Цель работы - изучение организации системы ввода-вывода базовой ЭВМ, команд ввода-вывода и исследование процесса функционирования ЭВМ при обмене данными по сигналам готовности внешних устройств.

Подготовка к выполнению работы. Закодировать заданную программу и составить ее описание. Команды программы надо разместить, начиная с ячейки 10, а коды символов - начиная с ячейки 20.

Порядок выполнения работы.

1. Занести программу в память базовой ЭВМ.
2. Запустить ЭВМ в режиме автоматического выполнения программы ("Работа") с 10 адреса, и ввести в память четыре первых символа заданного слова.

Примечание. Ввод информации в регистр данных ВУ-2 производится с клавиатуры после нажатия клавиш "Shift+F2" и подтверждается нажатием клавиши "Enter". Флаг готовности ВУ-2 устанавливается клавишей F2.

3. Перевести ЭВМ в режим покомандного выполнения программы ("Останов") и ввести в ее память еще два символа заданного слова, заполняя таблицу результатов (по форме табл. 1.3.).

Содержание отчета по работе.

Титульный лист, цель работы, текст программы с мнемоническими обозначениями и комментариями, заданное слово и коды его символов, таблица с результатами и описание программы.

Исходные данные к лабораторной работе.

1. Программа асинхронного обмена данными:

A: TSF 2	Опрос флага ВУ-2 и повторение этой операции, если
BR A	ВУ-2 не готово к обмену (флаг=0)
IN 2	Ввод данных в аккумулятор, если флаг=1
CLF 2	Сброс флага ВУ-2
MOV (B)	Пересылка содержимого аккумулятора в память и увеличение на 1 адреса элемента массива (B)+1→B
ISZ C	Нарастивание на 1 содержимого счетчика элементов
BR A	массива и переход по адресу A, пока (C)<0.
HLT	Останов ЭВМ.

Примечание. Здесь условные обозначения: А - адрес начала программы; В - адрес ячейки с начальным адресом массива (любая индексная ячейка); С - адрес ячейки, содержащей счетчик количества еще не введенных символов.

2. Варианты вводимых слов:

- 1) ОПТИКА; 2) РАБОТА; 3) МАШИНА; 4) ПАМЯТЬ; 5) ЯЧЕЙКА;
- 6) ДАННЫЕ.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие устройства ввода-вывода имеются в базовой ЭВМ?
2. Что из себя представляет асинхронный обмен данными с устройствами ввода-вывода в базовой ЭВМ?
3. Порядок выполнения работы.
4. Объяснение работы программы.

ЛАБОРАТОРНАЯ РАБОТА № 6

Исследование работы ЭВМ при обмене данными с ВУ в режиме прерывания программы

Цель работы - изучение организации процесса прерывания программы и исследование порядка функционирования ЭВМ при обмене данными в режиме прерывания программы.

Домашнее задание.

Написать комплекс программ, обеспечивающий обмен данными с ВУ в режиме прерывания программы. Основная программа должна наращивать на 1 (начиная с нуля) содержимое (обозначим его буквой X) какой-либо ячейки памяти в непрерывном цикле. Цикл для наращивания X не должен содержать более трех команд. Программа обработки прерывания должна реализовывать выбранный вариант задания.

Варианты задания:

1. По запросу ВУ-1 вывести $2X$, а по запросу ВУ-3 вывести X.
2. По запросу ВУ-1 вывести $X+2$, а по запросу ВУ-3 вывести $X-2$.
3. По запросу ВУ-1 вывести $2X+2$, а по запросу ВУ-3 вывести $X-5$.
4. По запросу ВУ-1 вывести $X+10$, а по запросу ВУ-3 вывести $X+90$.
5. По запросу ВУ-1 вывести $3X$, а по запросу ВУ-3 вывести X.
6. По запросу ВУ-1 вывести $2X-2$, а по запросу ВУ-3 вывести $X+50$.

Подготовка к выполнению работы. Завершить домашнее задание и составить описание разработанного комплекса программ для заданного варианта.

Порядок выполнения работы.

1. Занести комплекс программ в память базовой ЭВМ.
2. Запустить ЭВМ в режиме "РАБОТА" и проверить правильность функционирования разработанного комплекса программ.

Содержание отчета по работе. Титульный лист, цель работы, тексты основной программы и программы обработки прерываний с мнемоническими обозначениями и комментариями; результаты выполнения комплекса программ.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как организуется обмен с внешними устройствами базовой ЭВМ в режиме прерывания?
2. Порядок выполнения работы.
3. Объяснение работы программы.

3. МИКРОПРОГРАММНОЕ УСТРОЙСТВО УПРАВЛЕНИЯ

Переход к исследованию работы микропрограммного устройства управления модели базовой ЭВМ осуществляется вхождением в меню (нажатием клавиш "Shift +F4") и выбором из

меню режима "Работа с МПУ" (клавиши "стрелка вниз", "стрелка вверх" - для перемещения курсора и клавиша "Enter" - для выбора режима), после чего на экране появляется картинка, соответствующая рис. 3.1.

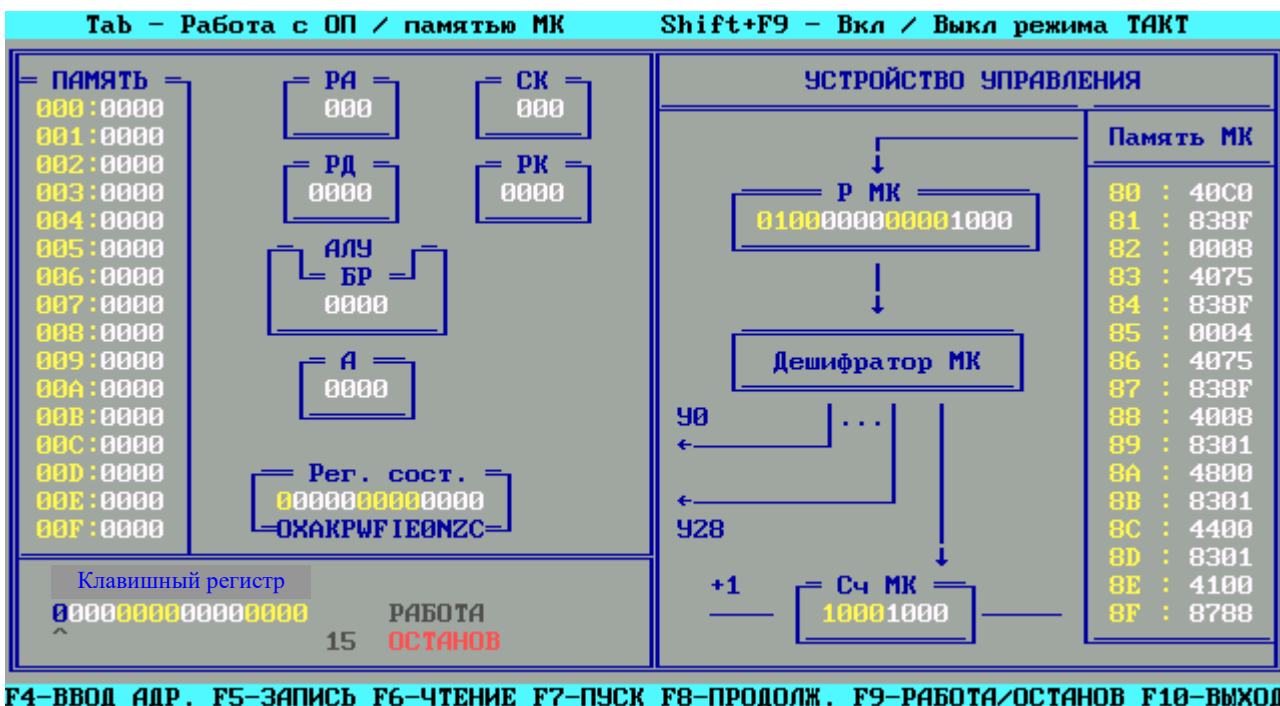


Рис. 3.1

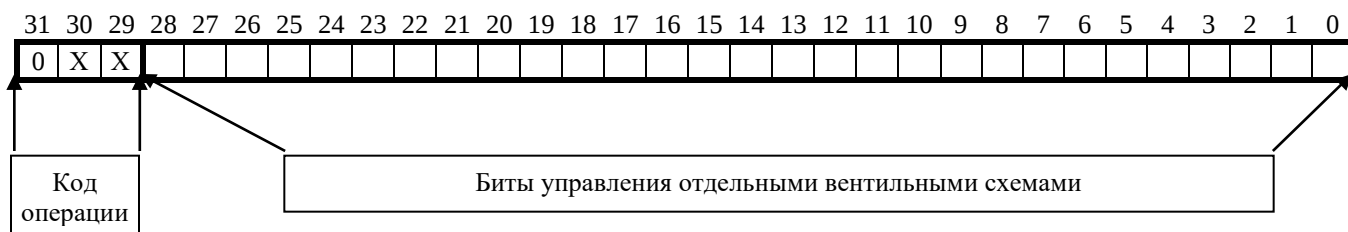
3.1. Микропрограммное управление вентильными схемами.

Процесс выборки, дешифрации и выполнения команд ЭВМ состоит из последовательности элементарных операций (например, пересылка содержимого одного регистра в другой регистр или проверка определенного бита в каком-либо регистре). Для выполнения таких микроопераций, как правило, достаточно подать открывающий сигнал на одну или несколько вентильных схем, связывающих между собой два регистра, регистр и АЛУ и (или) перестраивающих АЛУ на выполнение заданной операции (сложения, логического умножения и т.п.). Требуемая последовательность сигналов на вентильные схемы ЭВМ вырабатывается ее устройством управления, связанным с тактовым генератором.

Микропрограммное устройство управления (МПУ) базовой ЭВМ - это, в свою очередь, очень простая ЭВМ, для которой регистры и вентильные схемы процессора являются как бы устройствами ввода-вывода (рис. 3.2.). Программа работы такой ЭВМ называется микропрограммой, а ее команды, содержащие информацию об элементарных действиях, выполняемых в течение одного рабочего такта ЭВМ, - микрокомандами.

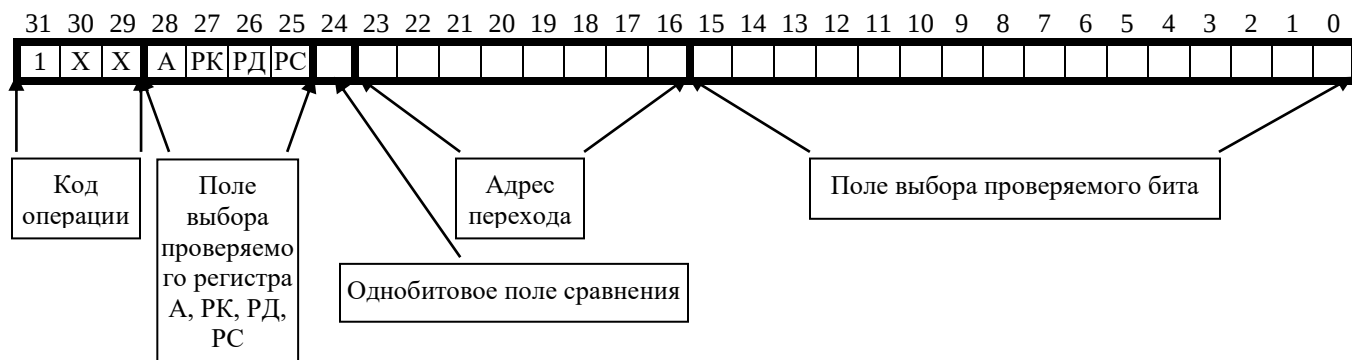
В данном варианте реализации МПУ используется всего два типа микрокоманд - операционная и управляющая:

Операционная микрокоманда (ОМК)





Управляющая микрокоманда



Микропрограмма хранится в постоянном запоминающем устройстве - памяти микрокоманд. В каждом такте работы ЭВМ из этой памяти в регистр микрокоманд (РМК) пересылается очередная микрокоманда, т. е. микрокоманда, на которую указывает счетчик микрокоманд (СчМК), одновременно выполняющий функции регистра адреса микрокоманд. Затем содержимое СчМК наращивается на единицу.

Если из памяти микрокоманд выбрана операционная микрокоманда, то в 31-й бит РМК записывается 0. Этот сигнал через инвертор открывает вентильную схему ВРО и обеспечивает передачу на В0-В28 состояний соответствующих битов РМК (управляющих сигналов У0-У28). Разряды РМК, содержащие 1, создают открывающий управляющий сигнал, а содержащие 0 - закрывающий. Подобная структура микрокоманды, где каждый бит используется для создания отдельного управляющего сигнала, называется горизонтальной.

Вентильные схемы В1, В2, В3 предназначены, соответственно, для передачи содержимого РД, РК, СК на правый вход АЛУ. Если все эти схемы закрыты ($У1=У2=У3=0$), то сигнал на правом входе АЛУ соответствует коду числа 0. Аналогично используются вентильные схемы В4, В5 и В6, позволяющие передать на левый вход АЛУ содержимое А, РС, клавишного регистра (КР) или кода числа 0.

Управляющие сигналы У7-У10 перестраивают АЛУ на выполнение различных микроопераций. При $У7=...=У10=0$ в 17 разрядный буферный регистр АЛУ (БР) записывается сумма входных сигналов АЛУ; при $У7=У8=У9=0$ и $У10=1$ к такой сумме добавляется 1; при $У7=У8=У10=0$ и $У9=1$ в БР записывается результат логического умножения входных сигналов АЛУ; при $У7=1$ и (или) $У8=1$ можно получить аналогичные результаты, но для инверсных значений одного или двух входных сигналов.

Рассмотрим несколько примеров операционных микрокоманд.

1. Для вычитания содержимого РД из содержимого А и записи результата в буферный регистр следует выполнить микрокоманду

$(0000\ 0000\ 0000\ 0000\ 0000\ 0101\ 0001\ 0010)_2 = (0000\ 0512)_{16}$, т.е. одновременно подать единичные управляющие сигналы на В1, В4, В8 и В10. Тогда к уменьшаемому прибавится обратный код вычитаемого и к этой сумме добавится единица, что эквивалентно суммированию уменьшаемого с дополнительным кодом вычитаемого.

2. Для вычитания 1 из содержимого аккумулятора надо выполнить микрокоманду $(0000\ 0110)_{16}$, т.е. подать единичные управляющие сигналы на В4 и В8 и сложить содержимое А с обратным кодом числа 0 или (что то же самое) с дополнительным кодом числа -1.

3. Для увеличения на 1 содержимого СК надо выполнить микрокоманду $(0000\ 0408)_{16}$, т.е. открыть В3 и В10.

Вентильные схемы В11 и В12 позволяют записать в БР сдвинутое на один разряд вправо или влево содержимое аккумулятора. При этом "лишний" разряд БР заполняется содержимым регистра переноса С.

Вентильные схемы В13-В15 используются для передачи в однобитные регистры С, N, Z признаков результата операции, выполненной в АЛУ: двух старших разрядов 17-ти битового БР (перенос и знак), а также выходного сигнала специальной схемы, который равен 1 лишь в том случае, когда содержимое БР равно 0. Управляющие сигналы У16 и У17 позволяют установить регистр С в 0 или 1 независимо от результата выполнения операции, сохраняемого на БР.

Вентильные схемы В18-В22 позволяют переписать содержимое 16 или 11 младших разрядов БР в РА, РД, РК, СК и А соответственно.

Вентильные схемы В23-В28 используются для организации обмена информацией между регистрами процессора и другими подсистемами ЭВМ (памятью и устройствами ввода-вывода). И, наконец, вентильная схема В0 используется для передачи сигнала прекращения выполнения программы (команда HLT).

Если из памяти микрокоманд выбрана управляющая микрокоманда, то в 31-й бит РМК записывается единица (код операции УМК). Этот сигнал открывает вентильную схему ВР1 и тем самым создает условия для выполнения УМК. Теперь по сигналу, создаваемому каким-либо битом поля выбора проверяемого регистра (У1, У2, У4 или У5) открывается одна из вентильных схем В1, В2, В4 или В5 и на вентили ВВ0-ВВ15 поступает через АЛУ содержимое соответствующего регистра (РД, РК, А или РС). Одновременно на эти же вентили поступает с РМК содержимое поля выбора проверяемого бита. Так как в этом поле записана только одна 1 (на месте, соответствующем проверяемому биту), то открывается лишь один из вентилях ВВ0-ВВ15, через который на схему сравнения поступает содержимое проверяемого бита из проверяемого регистра. На другой вход этой схемы поступает содержимое однобитового поля сравнения (24 бит УМК), в которое при кодировании УМК записали 0 или 1.

Если проверяемый бит и бит из поля сравнения идентичны, то схема сравнения формирует единичный сигнал, который открывает вентильную схему ВА и на СчМК пересылается адрес перехода (16-23 биты УМК). В противном случае на СчМК сохраняется адрес микрокоманды, расположенной вслед за исполняемой, так как после выборки текущей микрокоманды содержимое СчМК увеличивается на единицу.

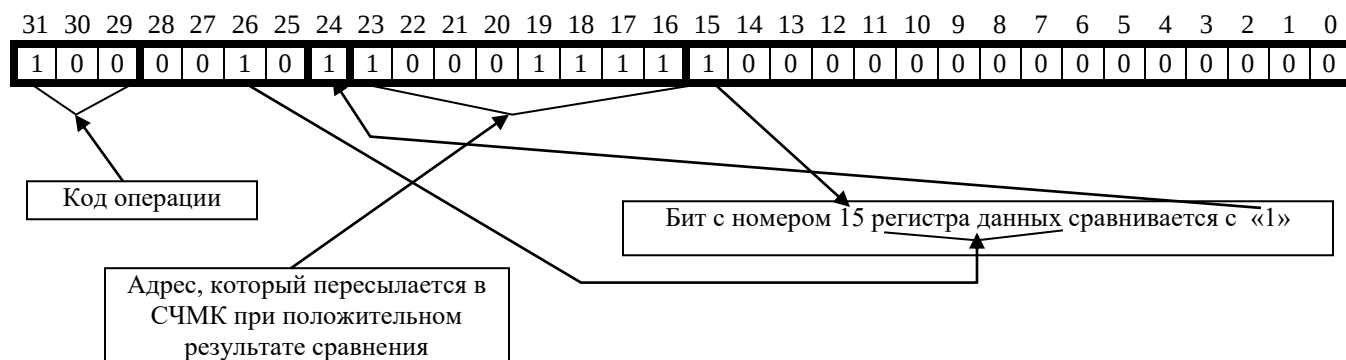
При организации разветвлений в микропрограмме используется содержимое регистра состояний, являющегося объединением однобитовых регистров признаков и состояний ЭВМ (табл. 3.1). Такое объединение сделано с целью формального уменьшения числа регистров, с которыми работает МПУ, что позволяет сократить разрядность УМК. На структурной схеме (рис. 3.2) РС изображен (для удобства описания) в виде самостоятельного регистра, хотя его разряды лишь дублируют состояние регистров С, N, Z и т.д.

Табл. 3.1

Раз- ряд	Содержимое	Раз- ряд	Содержимое	Раз- ряд	Содержимое
0	Перенос (С)	4	Разрешение пре- рывания	8	Программа
1	Нуль (Z)			9	Выборка команды
2	Знак (N)	5	Прерывание		
3	0 - используется для организации безусловных пе- реходов в МПУ	6	Состояние ВУ (Ф)	10	Выборка адреса
		7	Состояние РАБОТА/ОСТАНОВ (1 - РАБОТА)	11	Исполнение
				12	Ввод-вывод

Рассмотрим две управляющие команды:

1. После увеличения на 1 содержимого РД в команде ISZ надо проверить знаковый разряд РД (разряд с номером 15). Если этот разряд равен 1 (содержимое < 0), то выполнение команды ISZ завершается. В противном случае необходимо прибавить 1 к содержимому СК, т.е. организовать пропуск команды, следующей за ISZ. Это разветвление (переход по адресу 8F) осуществляется с помощью микрокоманды $(858F\ 8000)_{16}$, имеющей следующую структуру:



2. Для организации безусловного перехода (например, по тому же адресу 8F) используется 3-й бит регистра состояний, содержащий константу 0. Сравнение этого разряда с нулем, записанным в 24-й разряд УМК, всегда дает положительный результат и позволяет переслать в СчМК нужный адрес перехода. Микрокоманда, реализующая эту операцию, имеет вид: $(828F\ 0008)_{16}$.

3.2. Интерпретатор базовой ЭВМ

Полный текст микропрограммы (интерпретатора команд) приведен в табл. 3.2. В столбце "Верт." приведены варианты укороченной кодировки микрокоманд - вертикальные микрокоманды. Эти микрокоманды состоят из полей, в которых закодированы требуемые наборы управляющих сигналов (рис.3.3). Для декодирования таких команд используются дополнительные устройства – дешифраторы.

В модели базовой ЭВМ используется вариант интерпретатора с вертикальными микрокомандами.

Первые микрокоманды интерпретатора служат для выборки команды из основной памяти (ОП) базовой ЭВМ и определения ее типа: адресная, безадресная или ввода-вывода. Для этого содержимое СК (в котором хранится адрес исполняемой команды) пересылается через БР в РА (СК → БР и БР → РА). Затем из ячейки ОП, на которую указывает РА, пересылается в РД команда, а содержимое СК увеличивается на единицу и пересылается в БР: ОП(РА) → РД, СК + 1 → БР. Далее содержимое БР, т.е. адрес следующей команды, пересылается из БР в СК, а код команды из РД в РК, после чего начинается дешифрация команды.

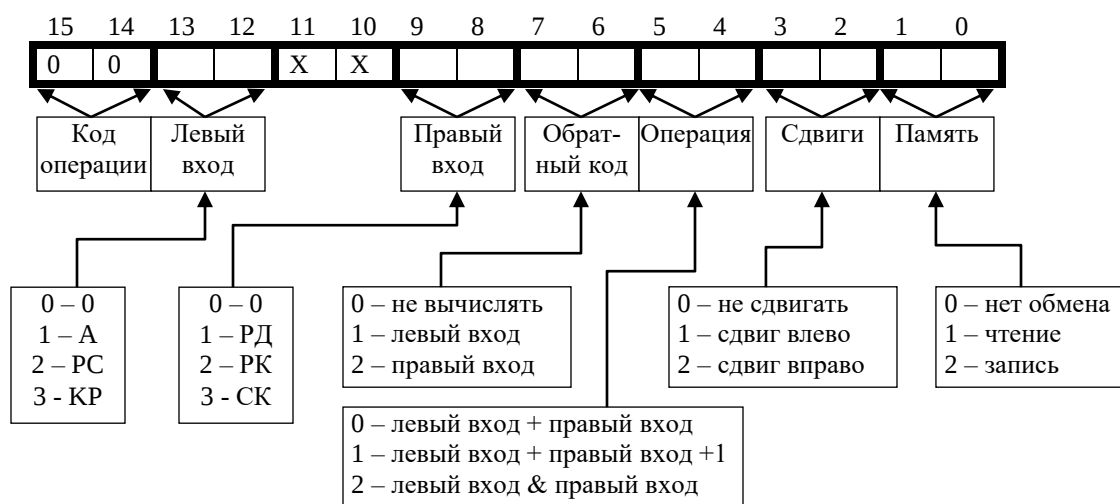
Так как адресные команды (команды с кодами операций от 0 до D) обязательно содержат ноль в 15, 14 или 13-м бите, то проверкой этих битов РК можно выделить адресную команду и перейти к проверке ее 11 бита (бита вида адресации). Для разделения команд ввода-вывода (код операции E) и безадресных команд (код операции F) достаточно проанализировать 12-й бит РК: если этот бит равен единице, то надо переходить к микрокомандам продолжения дешифрации

безадресных команд, расположенных, начиная с адреса 5E (метка БАД). В комментариях микрокоманда анализа 12-го бита записана в виде:

IF BIT(12,PK)=1 THEN БАД(5E).

В памяти микрокоманд нет полных микропрограмм для адресных команд с кодами операций 7 и D, а также для безадресных команд FC00, FD00 и FF00. Когда при декодировании команды выясняется, что выбрана команда 7xxx, управление передается ячейке с адресом B0. Начиная с этой ячейки могут располагаться микрокоманды какой-либо новой арифметической команды (например, умножения). Для микропрограмм реализации команды перехода и безадресных команд выделены участки памяти микрокоманд с начальными адресами D0 и E0.

Операционная микрокоманда (ОМК0)



Операционная микрокоманда (ОМК1)

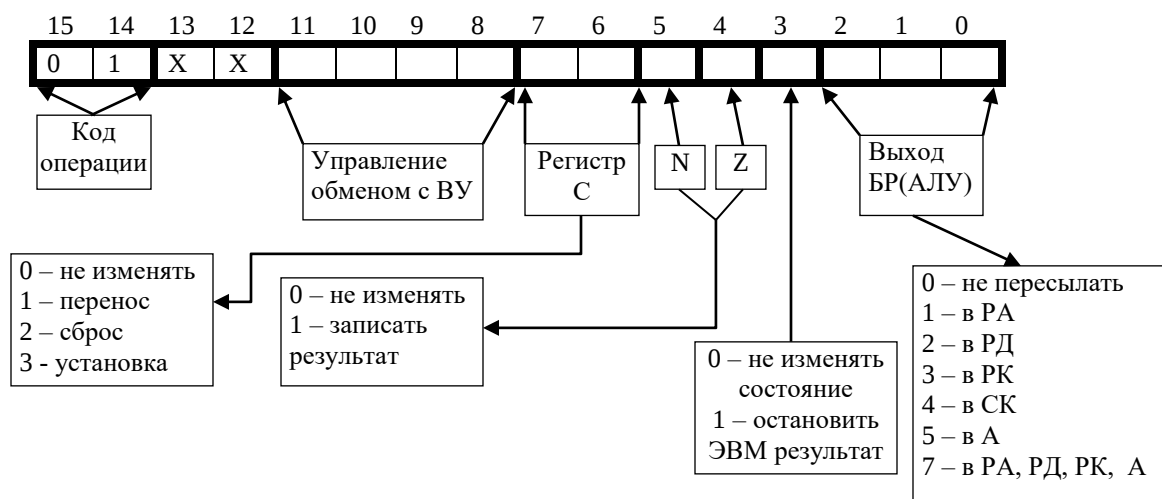


Рис. 3.3

Управляющая микрокоманда

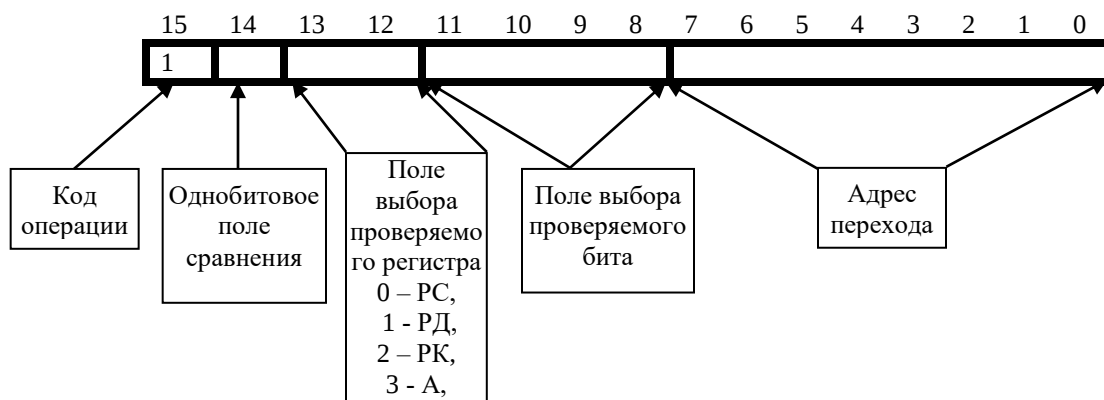


Рис. 3.3 (продолжение).

Таблица 3.2. Интерпретатор базовой ЭВМ

Адрес	Микрокоманды		Комментарии	
	Горизонт.	Верт.	Метка	Действие
1	2	3	4	5
Цикл выборки команды				
01	0000 0008	0300	нач	СК → БР
02	0004 0000	4001		БР → РА
03	0080 0408	0311		ОП(РА) → РД, СК+1 → БР
04	0020 0000	4004		БР → СК
05	0000 0002	0100		РД → БР
06	0010 0000	4003		БР → РК
Определение типа команды				
07	880C 8000	AF0C		IF BIT(15,PK)=0 THEN АДЦ(0C)
08	880C 4000	AE0C		IF BIT(14,PK)=0 THEN АДЦ(0C)
09	880C 2000	AD0C		IF BIT(13,PK)=0 THEN АДЦ(0C)
0A	895E 1000	EC5E		IF BIT(12,PK)=1 THEN БАД(5E)
0B	828E 0008	838E		GOTO B/B(8E)
Определение вида адресации				
0C	881D 0800	AB1D	АДЦ	IF BIT(11,PK)=0 THEN АДР(1D)
Цикл выборки адреса операнда				
0D	0000 0002	0100		РД → БР
0E	0004 0000	4001		БР → РА
0F	0080 0000	0001		ОП(РА) → РД
10	881D 0003	A31D		IF BIT(3,PK)=0 THEN АДР(1D)
11	891D 0010	E41D		IF BIT(4,PK)=1 THEN АДР(1D)
12	891D 0020	E51D		IF BIT(5,PK)=1 THEN АДР(1D)
13	891D 0040	E61D		IF BIT(6,PK)=1 THEN АДР(1D)
14	891D 0080	E71D		IF BIT(7,PK)=1 THEN АДР(1D)
15	891D 0100	E81D		IF BIT(8,PK)=1 THEN АДР(1D)
16	891D 0200	E91D		IF BIT(9,PK)=1 THEN АДР(1D)
17	891D 0400	EA1D		IF BIT(10,PK)=1 THEN АДР(1D)
18	0000 0402	0110		РД+1 → БР
19	0008 0000	4002		БР → РД
1A	0100 0000	0002		РД → ОП(РА)

1B	0000 0082	0149		РД + COM(0)=РД+1 → БР
1C	0003 0000	4002		БР → РД
Цикл исполнения адресных команд				
1D	892D 8000	EF2D	АДР	IF BIT(15,PK)=1 THEN ПРХ(2D)
1E	0000 0002	0100		РД → БР
1F	0004 0000	4001		БР → РА
20	8927 4000	EE27		IF BIT(14,PK)=1 THEN АРФ(27)
21	8824 2000	AD24		IF BIT(13,PK)=0 THEN А1(24)
22	8857 1000	AC57		IF BIT(12,PK)=0 THEN JSR(57)
23	8238 0008	8338		GOTO MOV(38)
24	0080 0000	0001	А1	ОП(РА) → РД
25	8850 1000	AC50		IF BIT(12,PK)=0 THEN ISZ(50)
26	8235 0008	8335		GOTO AND(35)
27	0080 0000	0001	АРФ	ОП(РА) → РД
28	882B 2000	AD2B		IF BIT(13,PK)=0 THEN СУМ(2B)
29	8843 1000	AC43		IF BIT(12,PK)=0 THEN SUB(43)
2A	8280 0008	8380		GOTO P-A(B0)
2B	883C 1000	AC3C	СУМ	IF BIT(12,PK)=0 THEN ADD(3C)
2C	823F 0008	833F		GOTO ADC(3F)
2D	8830 4000	AE30	ПРХ	IF BIT(14,PK)=0 THEN УПХ(30)
2E	8847 1000	AC47		IF BIT(12,PK)=0 THEN BR(47)
2F	82D0 0008	83D0		GOTO P-П(D0)
30	8833 2000	AD33	УПХ	IF BIT(13,PK)=0 THEN П1(33)
31	884C 1000	AC4C		IF BIT(12,PK)=0 THEN BMI(4C)
32	824E 0008	834E		GOTO BEQ(4E)
33	8846 1000	AC46	П1	IF BIT(12,PK)=0 THEN BCS(46)
34	824A 0008	834A		GOTO BPL(4A)
Исполнение адресных команд				
35	0000 0212	1120	AND	А & РД → БР
36	0040 C000	4035		БР → А,N,Z
37	828F 0008	838F		GOTO ПРЕ(8F)
38	0000 0010	1000	MOV	А → БР
39	0008 0000	4002		БР → РД
3A	0100 0000	0002		РД → ОП(РА)
3B	828F 0008	838F		GOTO ПРЕ(8F)
3C	0000 0012	1100	ADD	А + РД → БР
3D	0040 E000	4075		БР → А,C,N,Z
3E	828F 0008	838F		GOTO ПРЕ(8F)
3F	823C 0001	803C	ADC	IF BIT(0,PC)=0 THEN ADD(3C)
40	0000 0412	1110		А + РД +1 → БР
41	0040 E000	4075		БР → А,C,N,Z
42	828F 0008	838F		GOTO ПРЕ(8F)
43	0000 0512	1190	SUB	А + COM(РД) + 1 = А - РД → БР
44	0040 E000	4075		БР → А,C,N,Z
45	828F 0008	838F		GOTO ПРЕ(8F)
46	828F 0001	808F	BCS	IF BIT(0,PC)=0 THEN ПРЕ(8D)
47	0000 0002	0100	BR	РД → БР
48	0020 0000	4004		БР → СК
49	828F 0008	838F		GOTO ПРЕ(8F)
4A	838F 0004	C28F	BPL	IF BIT(2,PC)=1 THEN ПРЕ(8F)
4B	8247 0008	8347		GOTO BR(47)

4C	828F 0004	828F	BMI	IF BIT(2,PC)=0 THEN ПРЕ(8F)
4D	8247 0008	8347		GOTO BR(47)
4E	828F 0002	818F	BEQ	IF BIT(1,PC)=0 THEN ПРЕ(8F)
4F	8247 0008	8347		GOTO BR(47)
50	0000 0402	0110	ISZ	РД + 1 → БР
51	0008 0000	4002		БР → РД
52	0100 0000	0002		РД → ОП(РА)
53	858F 8000	DF8F		IF BIT(15,РД)=1 THEN ПРЕ(8F)
54	0000 0408	0310		СК + 1 → БР
55	0020 0000	4004		БР → СК
56	828F 0008	838F		GOTO ПРЕ(8F)
57	0000 0402	0110	JSR	РД + 1 → БР
58	0010 0000	4003		БР → РК
59	0000 0008	0300		СК → БР
5A	0008 0000	4002		БР → РД
5B	0100 0004	0202		РД → ОП(РА), РК → БР
5C	0020 0000	4004		БР → СК
5D	828F 0008	838F		GOTO ПРЕ(8F)
Продолжение цикла выборки команды				
Декодирование и исполнение безадресных команд				
5E	8861 0800	AB61	БАД	IF BIT(11,PK)=0 THEN Б0(61)
5F	886C 0400	AA6C		IF BIT(10,PK)=0 THEN Б1(6C)
60	82E0 0008	83E0		GOTO Р - Б(Е0)
61	8867 0400	AA67	Б0	IF BIT(10,PK)=0 THEN Б2(67)
62	8865 0200	A965		IF BIT(9,PK)=0 THEN Б3(65)
63	8882 0100	A882		IF BIT(8,PK)=0 THEN ROL(82)
64	8285 0008	8385		GOTO ROR(85)
65	887B 0100	A87B	Б3	IF BIT(8,PK)=0 THEN CMA(7B)
66	827E 0008	837E		GOTO CMC(7E)
67	886A 0200	A96A	Б2	IF BIT(9,PK)=0 THEN Б4(6A)
68	8876 0100	A876		IF BIT(8,PK)=0 THEN CLA(76)
69	8279 0008	8379		GOTO CLC(79)
6A	8888 0100	A888	Б4	IF BIT(8,PK)=0 THEN HLT(88)
6B	8287 0008	8387		GOTO NOP(87)
6C	886F 0200	A96F	Б1	IF BIT(9,PK)=0 THEN Б5(6F)
6D	888A 0100	A88A		IF BIT(8,PK)=0 THEN EI(8A)
6E	828C 0008	838C		GOTO DI(8C)
6F	8873 0100	A873	Б5	IF BIT(8,PK)=0 THEN INC(73)
70	0000 0110	1080	DEC	A + COM(0)=A - 1 → БР
71	0040 E000	4075		БР → A, C, N, Z
72	828F 0008	838F		GOTO ПРЕ(8F)
73	0000 0410	1010	INC	A + 1 → БР
74	0040 E000	4075		БР → A, C, N, Z
75	828F 0008	838F		GOTO ПРЕ(8F)
76	0000 0200	0020	CLA	0 → БР
77	0040 C000	4035		БР → A, N, Z
78	828F 0008	838F		GOTO ПРЕ(8F)
79	0001 0000	4080	CLC	0 → C
7A	828F 0008	838F		GOTO ПРЕ(8F)
7B	0000 0090	1040	CMA	COM(A) → БР; ИНВЕРСИЯ A
7C	0040 C000	4035		БР → A, N, Z
7D	828F 0008	838F		GOTO ПРЕ(8F)

7E	8280 0001	8080	СМС	IF BIT(0,PC)=0 THEN B6(80)
7F	8279 0008	8379		GOTO CLC(79)
80	0002 0000	40C0	Б6	1 → C
81	828F 0008	838F		GOTO ПРЕ(8F)
82	0000 1000	0008	ROL	RAL(A) → БР; СДВИГ ВЛЕВО
83	0040 E000	4075		БР → A, C, N, Z
84	828F 0008	838F		GOTO ПРЕ(8F)
85	0000 0800	0004	ROR	RAR(A) → БР; СДВИГ ВПРАВО
86	0040 E000	4075		БР → A, C, N, Z
87	828F 0008	838F	NOP	GOTO ПРЕ(8F)
88	0000 0001	4008	HLT	ОСТАНОВ МАШИНЫ
89	8201 0008	8301		GOTO НАЧ(01)
8A	1000 0000	4800	EI	РАЗРЕШЕНИЕ ПРЕРЫВАНИЯ
8B	8201 0008	8301		GOTO НАЧ(01)
8C	0800 0000	4400	DI	ЗАПРЕЩЕНИЕ ПРЕРЫВАНИЯ
8D	8201 0008	8301		GOTO НАЧ(01)
Продолжение цикла выборки команды Декодирование и исполнение команд ввода-вывода				
8E	0200 0000	4100	В/В	ОРГАНИЗАЦИЯ СВЯЗЕЙ С ВУ
ЦИКЛ ПРЕРЫВАНИЯ				
8F	8288 0080	8788	ПРЕ	IF BIT(7,PC)=0 THEN HLT(88)
90	8201 0020	8501		IF BIT(5,PC)=0 THEN НАЧ(01)
91	0000 0200	0020		0 → БР
92	0004 0000	4001		БР → РА
93	0000 0008	0300		СК → БР
94	0008 0000	4002		БР → РД
95	0100 0400	0012		РД → ОП(РА), 1 → БР
96	0020 0000	4004		БР → СК
97	0800 0000	4400		ЗАПРЕЩЕНИЕ ПРЕРЫВАНИЯ
98	8201 0008	8301		GOTO НАЧ(01)
Пульты операции				
Ввод адреса				
99	0000 0040	3000	В/А	КР → БР
9A	0020 0000	4004		БР → СК
9B	828F 0008	838F		GOTO ПРЕ(8F)
Чтение				
9C	0000 0008	0300	ЧТ	СК → БР
9D	0004 0000	4001		БР → РА
9E	0080 0408	4004		ОП(РА) → РД, СК + 1 → БР
9F	0020 0000	4004		БР → СК
A0	828F 0008	838F		GOTO ПРЕ(8F)
Запись				
A1	0000 0008	0300	ЗАП	СК → БР
A2	0004 0000	4001		БР → РА
A3	0000 0040	3000		КР → БР
A4	0008 0000	4002		БР → РД
A5	0100 0408	0312		РД → ОП(РА), СК + 1 → БР
A6	0020 0000	4004		БР → СК
A7	828F 0008	838F		GOTO ПРЕ(8F)
Пуск				
A8	0000 0200	0020	ПУС	0 → БР

A9	005C E000	4077		БР -> А, С, N, Z, РА, РД, РК
AA	0400 0000	4200		СБРОС ФЛАГОВ ВУ
AB	0800 0000	4400		ЗАПРЕЩЕНИЕ ПРЕРЫВАНИЯ
AC	828F 0008	838F		ГОТО ПРЕ(8F)
				
B0			Р-А	АРИФМЕТИЧЕСКАЯ КОМАНДА 7###
				
D0			Р-П	КОМАНДА ПЕРЕХОДА D###
				
E0			Р-Б	БЕЗАДРЕСНАЯ КОМАНДА FC##
				
FF				

Домашнее задание

Расширение системы команд базовой ЭВМ

Цель задания - изучение микрокоманд базовой ЭВМ, микропрограмм выполнения отдельных команд, а также овладение навыками составления микропрограмм для новых команд.

Часть 1. Написать последовательность адресов микрокоманд, которые должны быть выполнены при реализации заданного фрагмента программы, начинающегося с команды, расположенной по адресу 002 (предполагается, что регистр переноса и аккумулятор предварительно очищены).

Адрес	Номер варианта					
	1	2	3	4	5	6
002	СМА	INC	DEC	ADD 01	+BEQ 04	CMC
003	BMI 05	BPL 05	+BMI 05	+BPL 05	NOP	BCS 05
004	NOP	NOP	NOP	NOP	ADD 01	NOP
005	+MOV 01	+ADD 01	ADD 01	DEC	INC	+ADD 01

Результаты сводятся в таблицу вида:

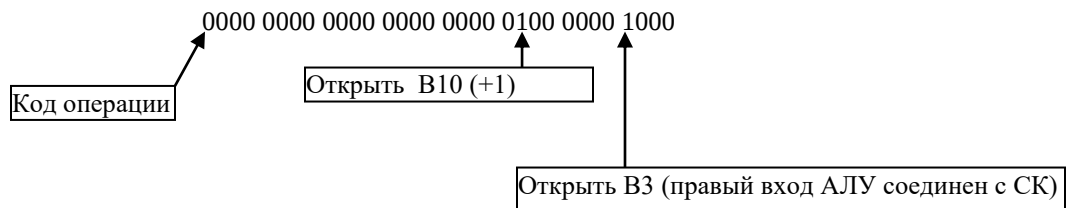
Команда	Машинный цикл	Последовательность адресов микрокоманд
AND 01 (1001)	- Выб. команды опред. типа опред. вида адр. Исполнение -	89 01,02,03,04,05,06, 07, 0C 1D,1E,1F,20,21,24,25,26,35,36,37,8F 88
CLC (1300)	- Выб. команды опред. типа Исполнение -	89 01,02,03,04,05,06, 07,08,09,0A, 5E,61,67,69, 79,7A,8F 88

В этой таблице символом "-" отмечены микрокоманды останова и перехода к циклу "Выборка команды", используемые при пошаговом выполнении программы.

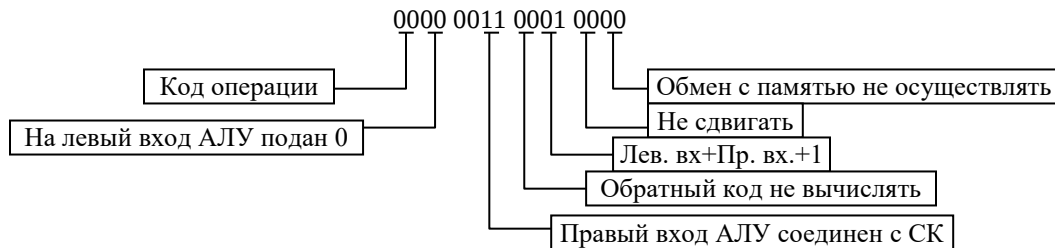
Кроме того, необходимо описать поля одной управляющей и одной операционной микрокоманд из шести последних микрокоманд цикла "Исполнение команды", отмеченной символом "+". Описания каждой микрокоманды выполнить в виде рисунков:

Микрокоманда: СК+1->БР

Горизонтальная: 0000 0408



Вертикальная: 0310



Часть 2

А. Написать завершающие вертикальные микрокоманды цикла "Исполнение" следующих команд:

Команда 7xxx

1 вариант - Загрузка (записать в аккумулятор содержимое ячейки памяти, на которую указывает адресная часть команды);

2 вариант - Пересылка со сбросом (записать содержимое аккумулятора в ячейку памяти, на которую указывает адресная часть команды, а затем очистить аккумулятор);

3 вариант - Сравнение (вычесть из содержимого аккумулятора содержимое ячейки памяти, на которую указывает адресная часть команды, и, не изменяя содержимое аккумулятора, установить признаки результата вычитания: C,N,Z);

4 вариант - Загрузка дополнительная (записать в аккумулятор дополнительный код содержимого ячейки, на которую указывает адресная часть команды);

5 вариант - Пересылка дополнительная (записать дополнительный код содержимого аккумулятора в ячейку памяти, на которую указывает адресная часть команды);

6 вариант - Пересылка удвоенная (записать в ячейку памяти, на которую указывает адресная часть команды, удвоенное содержимое аккумулятора).

Команда Dxxx

Организовать переход к команде, расположенной по адресу, на который указывает адресная часть команды, если:

1 вариант - аккумулятор содержит четное число;

2 вариант - аккумулятор содержит нечетное число;

3 вариант - аккумулятор содержит число, большее чем 16383;

4 вариант - аккумулятор содержит число, меньшее чем -16384;

5 вариант - 7-й бит аккумулятора (старший бит младшего байта) равен нулю;

6 вариант - 7-й бит аккумулятора равен единице.

Безадресные команды

1 вариант - циклический сдвиг влево на два разряда (FC00);

2 вариант - циклический сдвиг вправо на два разряда (FD00);

3 вариант - получение дополнительного кода аккумуля. (FE00);

4 вариант - запись единицы в аккумулятор (FC00);

5 вариант - цикл. сдвиг влево с очисткой рег. С (FD00);

6 вариант - цикл. сдвиг вправо с очисткой рег. С (FE00).

Б. Написать текстовые программы для проверки правильности исполнения всех трех синтезированных команд базовой ЭВМ и подготовиться к выполнению лабораторной работы №8.

В. При разработке микропрограмм заданных команд следует иметь ввиду:

1. В процессе дешифрации команды 7xxx в РА записывается адрес операнда (может использоваться для команд пересылки), а в РД - сам операнд (может использоваться для команд загрузки и сравнения). Затем осуществляется переход к ячейке памяти микрокоманд В0, где надо разместить первую синтезированную микрокоманду команды 7xxx.

2. После выборки команды перехода Dxxx в РД сохраняется адрес перехода (адресная часть команды), который может быть переписан в СК при выполнении условия перехода. Последняя микрокоманда дешифрации команды Dxxx передает управление в ячейку с адресом D0, где надо разместить первую синтезируемую микрокоманду команды Dxxx.

3. Когда в процессе дешифрации безадресных команд выясняется, что в 10-м и 11-м разрядах РК содержатся единицы (т.е. выбрана одна из команд: FC00, FD00, FE00 или FF00), управление передается в ячейку с адресом E0. Здесь должны начинаться микрокоманды дополнительной дешифрации, выделяющие заданную команду путем анализа 9-го и 8-го разрядов РК и передающие управление в свободную область памяти микрокоманд (от E2 до FF), где следует разместить микрокоманды реализации заданной безадресной команды.

4. Все микропрограммы синтезируемых команд должны заканчиваться микрокомандой 838F (GOTO ПРЕ(8F)), осуществляющей переход к микрокомандам, завершающим исполнение любой команды базовой ЭВМ.

Пример. Для создания команды FF00, которая осуществляет инвертирование содержимого аккумулятора и очистку регистра переноса, можно написать следующую последовательность микрокоманд:

Адрес ПМК	Микро- команды	Комментарии	
E0	A98F	IF BIT(9,PK)=0 THEN ПРЕ(8F)	К окончанию цикла исполнения, если дешифрируемая команда не FF00
E1	A88F	IF BIT(8,PK)=0 THEN ПРЕ(8F)	
E2	1040	COM(A)->БР	Инверсия А
E3	4035	БР->A,N,Z	Пересылка результата в А и регистр признаков
E4	4080	0->C	Очистка С
E5	838F	GOTO ПРЕ(8F)	Выход

Для проверки синтезированной команды можно выполнить пультовую команду "Пуск" и следующие команды: СМА (занесение 1 в С) и FF00 (очистка С и занесение 1 в А).

ЛАБОРАТОРНАЯ РАБОТА № 7

Исследование микропрограммного устройства управления

Цель работы - исследование микропрограмм выполнения нескольких команд базовой ЭВМ, способов программирования отдельных машинных циклов и дешифрации команд, а также принципа кодирования отдельных микрокоманд. Работа является завершением первой части домашнего задания. В ней проводится проверка правильности порядка выполнения микрокоманд команд заданной программы.

Подготовка к выполнению работы - завершить первую часть домашнего задания и подготовить следующие таблицы:

а) для записи последовательности микрокоманд, которые будут выполняться базовой ЭВМ при реализации фрагмента программы первой части домашнего задания (форма таблицы аналогична таблице этого задания);

б) для записи результатов выполнения шести последних микрокоманд цикла "Исполнение" команды, которая отмечена символом "+" в заданном фрагменте программы.

СчМК до выборки МК	Содержимое регистров после выборки и исполнения МК										
	РМК	СК	РА	РК	РД	А	С	БР	N	Z	СчМК
xx	xxx	xxx	xxxx	xxxx	xxxx	xxxx	x	xxxx	x	x	xx

Порядок выполнения работы

1. Занести в память машины заданный фрагмент программы, ввести его пусковой адрес, нажать клавишу F7 ("Пуск") и после завершения начальной установки устройств ЭВМ перевести ее в режим потактового выполнения программы (режим "ТАКТ" устанавливается нажатием клавиш Shift+F9 и отключается повторным нажатием).

2. Нажатием клавиши F8 последовательно выполнить все микрокоманды, записывая в подготовленные таблицы адреса выполняемых микрокоманд и для шести из них - содержимое регистров.

Содержание отчета по работе.

1. Титульный лист.
2. Цель работы.
3. Таблицы экспериментальных данных.
4. Схема алгоритма дешифрации команды, помеченной символом "+".

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Типы микрокоманд и форматы микрокоманд.
2. Порядок выполнения работы.
3. Объяснение работы программы на микрокомандном уровне.

ЛАБОРАТОРНАЯ РАБОТА № 8

Синтез команд базовой ЭВМ

Цель работы - практическое завершение второй части домашнего задания. В ней производится загрузка в память микропрограмм микрокоманд новых команд базовой ЭВМ, загрузка в память ЭВМ программы для проверки правильности выполнения синтезированных команд, а также проверка и отладка этих микропрограмм.

Подготовка к выполнению работы. Завершить домашнее задание и подготовить две таблицы по форме, приведенной в лабораторной работе № 7.

Строки первой из этих таблиц (теоретической) должны быть заполнены содержимым регистров базовой ЭВМ при пошаговом выполнении за нее тестовой программы (синтезированные команды должны выполняться по тактам, а остальные - по командам). Строку с содержимым регистров ЭВМ после исполнения команды (или первой микрокоманды новой команды) следует предварять заголовком:

КОМАНДА xxxx, РАСПОЛОЖЕННАЯ ПО АДРЕСУ xxx.

Вторая таблица (экспериментальная) заполняется в лаборатории.

Порядок выполнения работы.

1. Занести в память базовой ЭВМ текст тестовой программы.
2. Занести в память микрокоманд (ПМК) микрокоманды новых команд. Микрокоманды заносятся в ПМК также как и команды в память ЭВМ, но при этом предварительно устанавливается режим работы "МК" нажатием клавиши "Tab" (повторное нажатие возвращает предшествующий режим работы). Адрес ПМК, куда заносится код микрокоманды указывает счетчик микрокоманд (СчМК), содержимое которого автоматически увеличивается на 1 после

занесения очередной микрокоманды. Правильность записи микрокоманд можно проконтролировать путем занесения в СчМК адреса первой микрокоманды и последовательного нажатия клавиши F6 ("чтение").

3. Выполнить в пошаговом режиме тестовую программу, занося в таблицу содержимое регистров процессора после выполнения каждой микрокоманды (для синтезированных команд) или каждой команды (для остальных команд).

Содержание отчета по работе.

1. Титульный лист.
2. Цель работы.
3. Таблицы с результатами выполнения тестовой программы (теоретическая и экспериментальная).
4. Анализ расхождения между этими таблицами и описание процесса отладки программы и микропрограммы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объяснение работы синтезированных команд с помощью структурной схемы микропрограммного устройства управления (рис. 3.2).
2. Порядок выполнения работы.
3. Разъяснение процесса отладки программы и микропрограммы.

ЛИТЕРАТУРА

1. Попов А.Ю. Организация ЭВМ: учебно-методическое пособие / А. Ю. Попов. — Москва: МГТУ им. Н.Э. Баумана, 2010. — 48 с. — Текст : электронный // Лань : электронно-библиотечная система. <https://e.lanbook.com/book/52407>
2. Довгий П.С., Скорубский В.И. Организация ЭВМ / П. С. Довгий, В. И. Скорубский. — Санкт-Петербург: НИУ ИТМО, 2009. — 56 с. — Текст: электронный // Лань: электронно-библиотечная система. <https://e.lanbook.com/book/40706>
3. Журавлев, А. Е. Организация и архитектура ЭВМ. Вычислительные системы: учебное пособие для спо / А. Е. Журавлев. — 2-е изд., стер. — Санкт-Петербург: Лань, 2021. — 144 с. — ISBN 978-5-8114-8611-3. — Текст: электронный // Лань: электронно-библиотечная система. URL: <https://e.lanbook.com/book/179036>
4. Вотинов, М. В. Практикум по архитектуре вычислительных машин, комплексам защиты информации и протоколам передачи данных в компьютерных сетях: учебное пособие / М. В. Вотинов. — Мурманск: МГТУ, 2018. — 110 с. — ISBN 978-5-86185-968-4. — Текст: электронный // Лань: электронно-библиотечная система. URL: <https://e.lanbook.com/book/142640>
5. Орлов С.А., Цилькер Б.Я.. Организация ЭВМ и систем. — Спб.: Питер, 2007. — 672 с.
6. Пятибратов А.П., Гудыно Л.П., Кириченко А.А. Вычислительные системы, сети и телекоммуникации. - М.: Финансы и статистика, 2008.
7. Меркухин Е.Н. Организация ЭВМ и систем. — Махачкала: ДГТУ, 2010. — 208 с.

Фейламазова Светлана Абдулаевна

Методические указания к выполнению лабораторных работ по дисциплине «Организация ЭВМ, вычислительных систем и комплексов» для студентов направления подготовки бакалавров 09.03.01 «Информатика и вычислительная техника». - Махачкала: ДГТУ, 2020. - 38 с.

Редактор Штанчаева П.Х. _____

Формат 60x84 1/16. Бумага тип №1.

Печать ротاپринтная. Уч. изд. л.

Усл. печ. л.

Тираж 200 экз. Заказ №

ИПЦ ДГТУ 367015, Махачкала, пр. Имама Шамиля, 70.