МГТУ им. Н.Э.Баумана

Методические указания по выполнению лабораторных работ на эмуляторе CUU-2016 по курсу «Вычислительные средства АСОИУ»

2016

Оглавление

1. Теоретическая часть.

1.1. Используемые обозначения.

ОП – оперативная память.
КОП – код операции.
РК – регистр команд.
РОН – регистр(ы) общего назначения.
А – аккумулятор.
В – буферный регистр.
РС – программный счетчик.
SP – указатель стека.
РИ – индексный регистр.
РА – регистр адреса.
РВ – регистр возврата.
R – адрес(номер) одного из РОН / адрес слова в ОП – для модификации адресов.
R1/R2 – адрес (номер) одного из РОН.
S1/S2/S3 – адрес в ОП.
X4, X3, X2, X1 – флаги разрядов КОП.
Z, N, C, P – флаги результата операции.
Y<число> - управляющий сигнал.
a<число> - состояние.
D6 – D1 – сигналы возбуждения.
ПЛМ – программируемая логическая матрица.

Каждое из указанных понятий далее рассмотренно подробно.

1.2. Введение.

Любое операционное устройство всегда можно представить состоящим из двух частей: операционной и управляющей.

Центральное устройство управления (ЦУУ) – управляющая часть процессоров. ЦУУ процессоров различных ЭВМ отличаются друг от друга структурой и выполняемыми функциями.

Типовые функции ЦУУ:

1. Формирование начального(загрузочного) адреса программы.
2. Выборка команды из ОП и ее хранение до выборки следующей команды.
3. Расшифровка КОП для выбора соответствующей микропрограммы.
4. Выборка операндов из А, РОН или ОП.
5. Запись результата в РОН или ОП.
6. Формирование адреса следующей команды.
7. Контроль выполнения команд (ошибки в КОП, переполнение и т.д.).
8. Остановка процессора.
9. Обработка прерываний.

1.3. Состав ЦУУ.

- **Регистры**
  - **Регистр команд(РК)** – для приема и хранения машинных команд(см. п.1.4. – Машинные команды). Размер – 4 байта.
  - **Программный счетчик(РС)** – для хранения адреса ОП текущей команды (см. п.1.4. – Машинные команды).
  - **Регистр адреса(РА)** – для хранения адреса ОП, по которому происходит чтение и запись некоторой информации.
  - **Регистр возврата(РВ)** – для хранения адреса возврата в основную программу при обращении к подпрограмме. Иногда для этого используется РОН, стек или ОП.
  - **Индексный регистр(РИ)** – для хранения индекса массива. Иногда для этого используется РОН или ОП.
  - **Аккумулятор(А), буферный регистр(В)** – для хранения адресов, операндов и результатов операций.
  - **Указатель стека(SP)** – адрес первой свободную ячейку в стеке.
  - **Регистры общего назначения(РОН)** – для хранения любых возможных данных. Имеют номера от 0 до 15.

  Регистры РС, РА, РВ, РИ, А, В, SP и РОН занимают 1 байт. Следует понимать, что физически они ничем не отличаются, в каждом из них технически возможно хранить любые данные, а их назначение обусловлено давним соглашением.
- **Стек** – для запоминания текущего состояния процессора. ЦУУ знает только адрес первой свободной ячейки стека (хранится в SP).
- **Схема инкремента–декремента(СИД)** – используется для увеличения или уменьшения значения операнда на 1.
- **Управляющий автомат(УА)**

Взаимодействие устройств между собой и с ОП происходит через шины Ш1, Ш2, Ш3. Структурная схема ЦУУ представлена на рисунке 1.

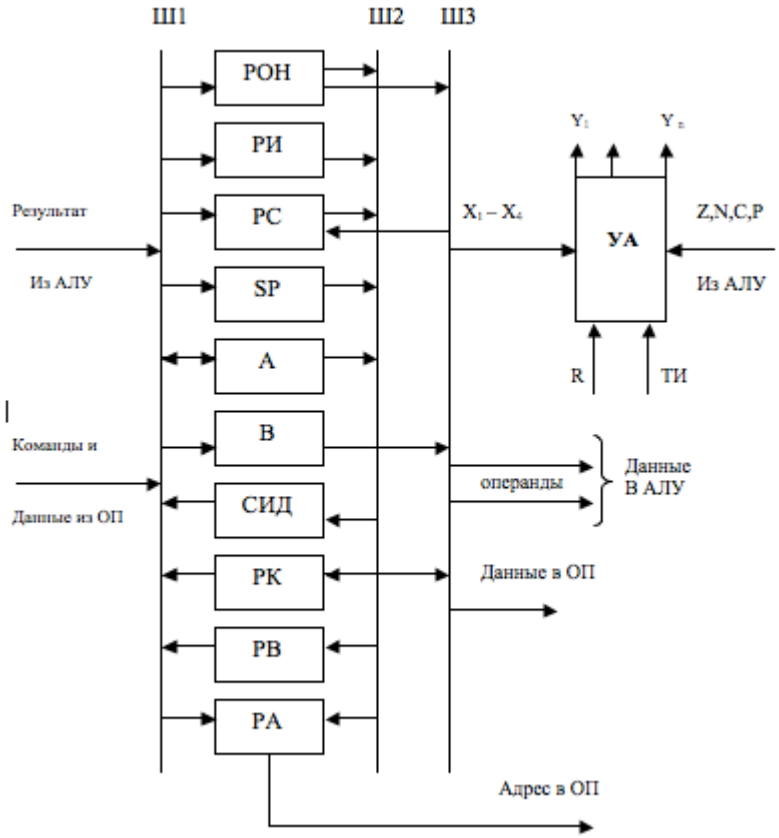


Рис.1. Структурная схема ЦУУ.

1.4. Машинные команды.

Машинная команда представляет собой некоторое количество байтов в ОП(объем ОП – 256 байт).

Первый байт каждой команды состоит из **КОП**(4 бита) и **R**(4 бита). При чтении команды из ОП флаги **X4, X3, X2, X1** устанавливаются соответственно разрядам КОП в двоичном коде, где **X4** – старший разряд, **X1** – младший.

R используется для модификации адресов.

- Если индекс хранится в РОН, в **R** указывается номер РОН.
- Если индекс хранится в ОП, в **R** указываются последние 4 бита адреса(индекс хранится по адресу **F.R**).
- Если для хранения индекса используется индексный регистр, в **R** записывается 0.

В последующих байтах хранятся адреса или операнды. Количество адресов/операндов определяет адресность команды. Команды могут быть одно-, двух-, трехадресными и безадресными.

Типы адресации:

- **Прямая** – указываются адреса операндов в ОП.
- **Непосредственная** – указываются операнды.
- **Косвенная регистровая** – указываются адреса(номера) РОН, в которых хранятся адреса операндов в ОП.
- **Прямая регистровая** – указываются адреса(номера) РОН, в которых хранятся операнды.

Команды считываются из ОП в регистр команд(РК) с помощью одного из управляющих сигналов:

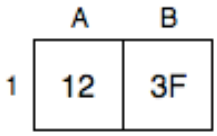
Y62 (РК[31:16] := ОП[РА]) – для одноадресных команд.

Y63 (РК[31:8] := ОП[РА]) – для двухадресных команд.

Y64 (РК[31:16] := ОП[РА]) – для трехадресных команд.

Для безадресных команд регистр команд не используется.

Пример: Рассмотрим 2 байта, расположенных в ОП по адресам 1A и 1B, в первом из которых записано число 12, а во втором 3F (в оперативной памяти используются **только шестнадцатиричные числа**).



Данные байты в совокупности могут представлять собой следующие команды:

1. Одноадресную команду с прямой адресацией (3F – адрес операнда в ОП).
2. Одноадресную команду с непосредственной адресацией (3F – операнд).
3. Двухадресную команду с косвенной регистровой адресацией(в РОН[3] хранится адрес ОП первого операнда, в РОН[15] – второго).
4. Команду с прямой регистровой адресацией(в РОН[3] хранится первый операнд, в РОН[15] – второй)

R = 2, следовательно используется команда с модификацией адресов и индекс хранится либо в РОН[2], либо в ОП[2F].

КОП = 1 (0001 в двоичном коде).

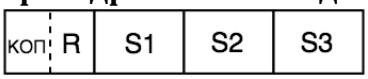
Для считывания команды из ОП запишем в регистр адреса (РА) адрес первого байта(1A) и выставим управляющий сигнал Y62. Будет выполнена микрооперация РК[31:16] = ОП[1A], в результате которой:

РК = 12 3F 00 00, X4 = 0, X3 = 0, X2 = 0, X1 = 1.

<u>Типовые машинные команды, используемые в работе:</u>

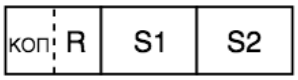
S1,S2,S3 – занимают 1 байт, хранят операнды/адреса в ОП.
R1,R2 – занимают ½ байта, хранят адреса(номера) РОН.
* - некоторая операция.
И – индекс, хранится в РИ, ОП[F.R] или РОН[R] в зависимости от варианта.
Программный счетчик – хранится в РС или РОН в зависимости от варианта.
Адрес возврата – хранится в SP, РВ, ОП[F.R] или РОН[R] в зависимости от варианта.

1. **Трехадресная команда с прямой адресацией и модификацией.**



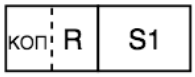
ОП[S3 + И] := ОП[S1 + И] * ОП[S2 + И]

2. **Двухадресная команда с прямой адресацией и модификацией.**



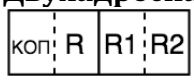
ОП[S1 + И] := ОП[S1 + И] * ОП[S2 + И]

3. **Одноадресная команда с прямой адресацией и модификацией.**



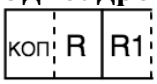
(ОП[S1 + И] и/или А) := А * ОП[S1 + И]

4. **Двухадресная команда с косвенной адресацией и модификацией.**



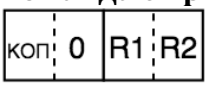
ОП[РОН[R1] + И] := ОП[РОН[R1] + И] * ОП[РОН[R2] + И]

5. **Одноадресная команда с косвенной адресацией и модификацией.**



(ОП[РОН[R1] + И] и/или А) := А * ОП[РОН[R1] + И]

6. **Команда с прямой регистровой адресацией.**

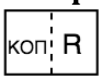


(РОН[R1] и/или А) := РОН[R1] * РОН[R2]

7. **Команда с непосредственной адресацией.**

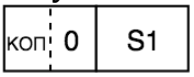
Используется для установки начальных значений адресов, индексов и т.д. S1 записывается в РОН[R]/РИ/ОП[F.R]/… в зависимости от предназначения команды.

8. **Возврат из подпрограммы(безусловный переход по косвенному адресу).**



Программный счетчик := адрес возврата.

9. **Безусловный переход по прямому адресу.**



Программный счетчик := S1

10. **Переход с возвратом(обращение к подпрограмме).**

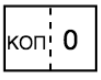
Адрес возврата := РС,
РС := S1

11. **Условный переход.**

Программный счетчик := S1, если условие выполнено.
Переход к следующей команде, если условие не выполнено(программный счетчик увеличивается на 1).

(См п. 1.5. - Условия перехода.). Аналогично можно использовать двух- и трехадресные команды для более сложных ветвлений.

12. **«СТОП»**



Завершение программы.

1.5. Условия перехода.

Для организации ветвления используются 9 флагов:

- **X4, X3, X2, X1** – соответствуют битам КОП текущей команды.
- **Z** – нулевой результат.
- **N** – отрицательный результат.
- **C** – перенос.
- **P** – переполнение.
- **B** – запуск.

Z, **N**, **C**, **P** устанавливаются в соответствии с результатом операции в АЛУ.

Условия перехода задаются различными комбинациями флагов. Условием может быть как положительное, так и отрицательное значение флага.

B может использоваться только отдельно от других флагов.

Для условного перехода необходимо указать исходное состояние(состояние, в котором ЦУУ проверит условия), следующее состояние(состояние, в которое перейдет ЦУУ при выполнении условий) и сами условия.

Пример 1: В данном фрагменте ЦУУ перейдет в состояние 2 при положительном результате операции, в состояние 3 – при нулевом, в состояние 4 – при отрицательном.

Исх. сост.	След. сост.	Условия перехода	Управляющие сигналы
1	2	-z,-n	
1	3	z	
1	4	n	

Рис.2. Условия перехода, пример 1.

Пример 2. В данном фрагменте ЦУУ перейдет в состояние 2 при КОП = 1, в состояние 3 – при КОП = 2.

Исх. сост.	След. сост.	Условия перехода	Управляющие сигналы
1	2	-x4,-x3,-x2,x1	
1	3	-x4,-x3,x2,-x1	

Рис.3. Условия перехода, пример 2.

При одновременном задании условий перехода и управляющих сигналов, сначала будут выполнены микрооперации, соответствующие управляющим сигналам, затем проверены условия.

Примеры использования машинных команд для организации ветвления:

PC:= { S1, если РИ > 0;
PC+1, если РИ < 0;

PC:= { S1, если ОП [F.R]<0;
S2, если ОП [F.R]>0;
PC+1, если ОП [F.R]=0;

PC:= { S1, если РОН [R]>0;
S2, если РОН [R]<0;
S3, если РОН [R]=0;

1.6. Микрооперации.

ЦУУ определяет, какую микрооперацию выполнять по выставленным управляющим сигналам. Соответствие управляющих сигналов и микроопераций, используемых в данной работе, приведено в таблице 1.

При указании нескольких управляющих сигналов, соответствующих микрооперациям, микрооперации будут выполнены в порядке возрастания номеров управляющих сигналов.

Таблица 1.

УС	Микрооперации	УС	Микрооперации
y20	A:=R	Y49	PC: =0
Y21	A: =S1	Y50	PC: =PC*A
Y22	A: =S2	…	
Y23	A: =S3	Y52	SP: =A
Y24	A: =PC	Y53	SP: SP*A
Y25	A: =PB	…	
Y26	A: =SP	Y55	РОН[R]:=S1
Y27	A: =РОН [R1]	Y56	РОН[R1]:=A
Y28	A: =РОН [R2]	Y57	РОН[R]:=РОН[R]*A
Y29	A: =РОН [T]	Y58	РОН[T]:=РОН[T]*A
Y30	A: =ОП [PA]	…	
Y31	A: =A*B	Y60	ОП[PA]:=A
…		…	
Y35	B: =РИ	Y62	РК[31:16]:=ОП[PA]
Y36	B: =РОН[R]	Y63	РК[31:8]:=ОП[PA]
Y37	B: =A*B	Y64	РК[31:0]:=ОП[PA]
Y38	B: =F.R	…	
…		Y66	A:=Дисплей
Y41	PA: =РОН [T]	Y67	Дисплей:= A
Y42	PA: =A*B	Y68	«Переполнение»
Y43	PA: =0	Y69	«Ошибка в ОП»
…		Y70	«Стоп»
Y45	PB: =A		
…			
Y47	РИ:=РИ+/-1		
Y48	РИ:=A		

*** - операция в АЛУ**. Операция выбирается с помощью управляющих сигналов y11,y12,y13,y14,y15(см. таблицу 2)

T – адрес(номер) РОН. T задается с помощью управляющих сигналов y7,y8,y9,y10(см. таблицу 3).

Сигнал Y47 используется как для увеличения, так и для уменьшения РИ на 1. Управление схемой инкремента-декремента осуществляется с помощью сигнала Y6(см. таблицу 4).

Сигнал Y66 используется для ввода байта данных с клавиатуры, **сигнал Y67** – для вывода байта данных на экран.

Сигналы Y68 и Y69 используются для вывода сообщений об ошибках на экран.

Сигнал Y70 соответствует микрооперации, которая выводит на экран сообщение о завершении программы и завершает ее на следующем такте.

Пример: Данный набор управляющих сигналов соответствует микрооперации РОН[3] := РОН[3] + A:

y7,y8,y11,y14,y58

Таблица 2.

Управляющие сигналы				Логические операции в АЛУ	Арифметические операции в АЛУ
y11	y12	y13	y14	y15=1	y15=0
0	0	0	0	$\overline{X}$	X+1
0	0	0	1	$\overline{X \vee Y}$	X << 1 (1)
0	0	1	0	$\overline{X} \wedge Y$	X+2
0	0	1	1	0	-
0	1	0	0	$\overline{X \wedge Y}$	X >> 1 (0)
0	1	0	1	$\overline{Y}$	X+3
0	1	1	0	$X \oplus Y$	X-Y
0	1	1	1	$\overline{X} \wedge Y$	-
1	0	0	0	$\overline{X} \vee Y$	X >> 1 (1)
1	0	0	1	$\overline{X \oplus Y}$	X+Y
1	0	1	0	Y	Y-X
1	0	1	1	$X \wedge Y$	X+4
1	1	0	0	FF	X << 1 (0)
1	1	0	1	$\overline{X} \vee Y$	-
1	1	1	0	$X \vee Y$	-
1	1	1	1	X	X-1

Операции X >> 1 и X << 0 означают сдвиг на один разряд вправо и влево соответственно. В скобках указано значение, которое после сдвига устанавливается у старшего(при сдвиге вправо) или младшего(при сдвиге влево) бита.

Таблица 3.

Управляющие сигналы	Адрес РОН [Т]															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
y7	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
y8	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
y9	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
y10	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Таблица 4.

Управляющий сигнал	Выполняемая микрооперация
У6 =1	Декремент (-1)

У6 = 0	Инкремент (+1)

2. Работа с макетом.

2.1 Запуск макета.

Windows – скачайте архив “cuu-windows.zip”, распакуйте его в отдельную папку, найдите в ней файл «cuu.exe», запустите.
Mac OS X –
Linux –

2.2 Начало работы.

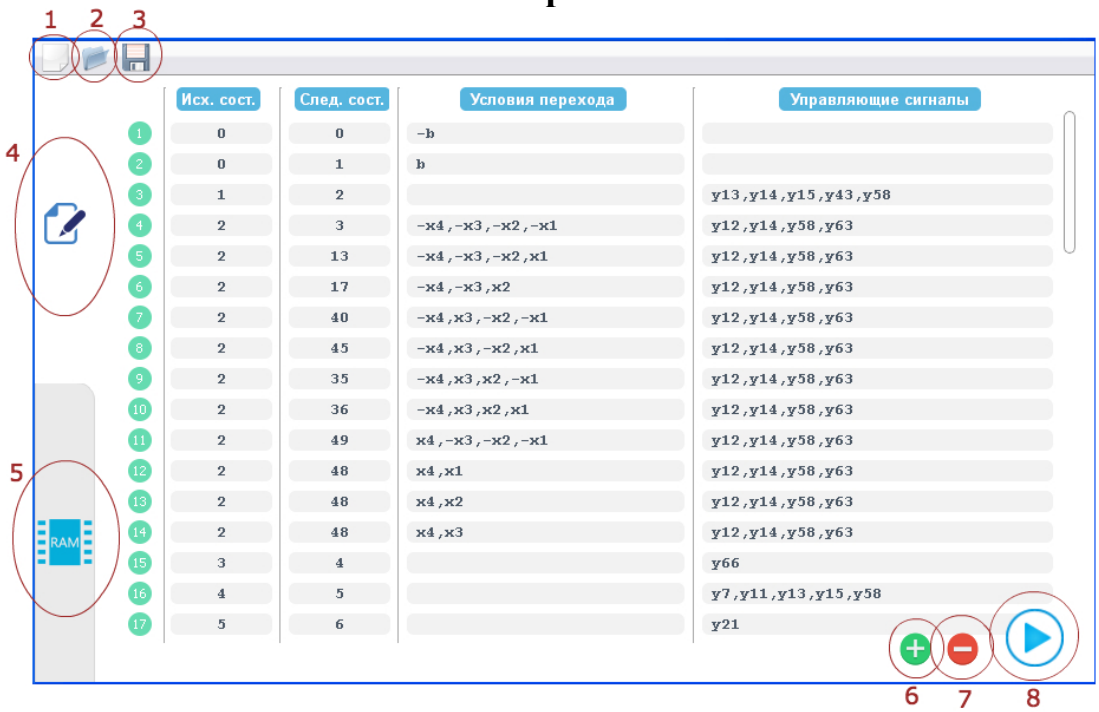


Рис.4. Главное окно макета.

Запустив макет, слева вы увидете две вкладки. Одна из них(**4**) соответствует таблице переходов, другая(**5**) – оперативной памяти. Для переключения между ними просто кликните на нужную вкладку.

Сверху расположено основное меню, с помощью которого вы можете создать новый файл(**1**), открыть существующий файл(**2**) или сохранить файл(**3**).

Кнопки **6** и **7** предназначены для добавления/удаления строки в таблице переходов.

Кнопка **8** запустит программу.

2.3 Создание нового файла.

Нажмите на кнопку **1**. Таблица переходов и ОП примут свое исходное состояние (при запуске макета они также находятся в исходном состоянии).

2.4 Открытие существующего файла.

Нажмите на кнопку **2**. В диалоговом окне вам будет предложено выбрать файл, видны будут только файлы с расширением «.cuu». Файлы, созданные в предыдущей версии макета, не поддерживаются.

2.5 Сохранение файла.

Нажмите на кнопку 3, введите название файла и выберите папку для сохранения.

2.6 Работа с таблицей переходов.

Каждая строка таблицы переходов состоит из четырех полей:

1. Исходное состояние.
2. Следующее состояние.
3. Условия перехода.
4. Управляющие сигналы.

Исходное и следующее состояние должны быть указаны обязательно. Условия перехода и управляющие сигналы могут быть не указаны вообще, указаны оба или может быть заполнено только одно из этих полей. Также допускаются пустые строки, которые игнорируются при выполнении программы.

Исходное и следующее состояния задаются числами от 0 до 999. В документации состояния указываются в формате a<номер состояния>, например, a2.

Условия перехода задаются комбинацией флагов x4, x3, x2, x1, z, n, c, p и b, перечисленных через запятую. Флаг b может задаваться только отдельно от других. Условием может быть как значение флага 1, так и значение флага 0. Для задания отрицательного условия перед соответствующим флагом ставится знак “–“ или “!”.

Управляющие сигналы перечисляются через запятую. Управляющий сигнал задается буквой “y” и следующим за ней номером сигнала. После окончания ввода, сигналы автоматически сортируются по возрастанию, а также удаляются сигналы, не указанные в данном методическом пособии.

Буквы “x” и “y” набираются на английской раскладке клавиатуры. Количество запятых и пробелов в полях Условия перехода и Управляющих сигналы не ограничено , строка будет автоматически приведена к правильному формату. Также допускается использование верхнего регистра, который будет автоматически изменен на нижний.

Если в поле Условия перехода или Управляющие сигналы введены неверные данные, после окончания ввода соответствующее поле подсветится красным. После исправления данных красный контур исчезнет.

Добавление строки:

Выберите строку, кливнув на любое поле в ней и нажмите клавишу «Enter» или кнопку **6**. Новая строка будет добавлена после выбранной строки. Для добавления строки перед первой строкой поместите курсор в начало первой строки(на нулевую позицию поля Исходное состояние) и нажмите клавишу «Enter» или кнопку **6.**

Удаление строки:

Выберите строку и нажмите кнопку **7** ИЛИ поместите курсор в начало строки(на нулевую позицию поля Исходное состояние) и нажмите клавишу «Backspace».

Перемещение между полями осуществляется клавишами со стрелками или клавишами WASD(в последнем случае при перемещении вправо/влево позиция курсора в поле не учитывается), также можно выбрать поле с помощью курсора.

2.7 Работа с ОП.

Слева от таблицы ОП располагаются адреса страниц, сверху – адреса слов в странице. Таким образом, чтобы ввести байт данных по адресу(к примеру) 3A , нужно выбрать строку, соответствующую странице памяти с адресом 3, столбец, соответствующий слову с адресом A в странице и ввести нужный байт в ячейку на их пересечении.

Ввод данных можно производить непрерывно, после ввода байта данных курсор автоматически переместится в начало следующего байта. Вручную перемещение между байтами осуществляется с помощью клавиш со стрелками или курсора.

2.8 Выполнение программы.

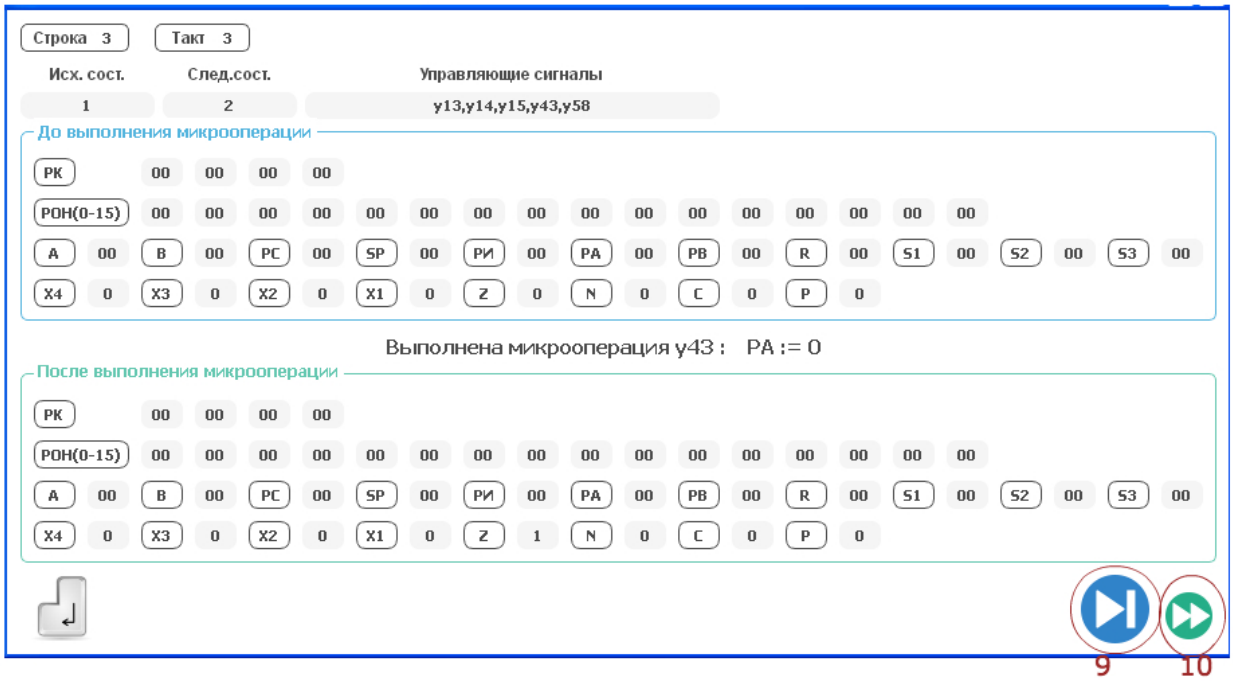


Рис.5. Окно выполнения программы.

Для запуска программу нажмите кнопку **8**. Если программа содержит ошибки (есть поля, подсвеченные красным), то выдастся сообщение об ошибках и программа запущена не будет.

После запуска программы будет сразу выполнен первый такт. В окне выполнения программы содержится информация о текущей строке программы, выполняемой микрооперации, а также информация о состоянии регистров и флагов до и после выполнения микрооперации.

Переход к следующему такту осуществляется с помощью клавиши «Enter» или кнопки **9**.

Перемотка в конец программы осуществляется с помощью кнопки **10**. Перемотка возможна только до 1000-го такта. Если программа выполняется циклически или содерит слишком большое количество тактов, после 1000-го такта можно воспользоваться пошаговым выполнением.

Вы в любой момет можете закрыть окно выполнения программы клавишей «Esc» или стандартной кнопкой закрытия окна.

2.9 Выход.

Закрытие макета осуществляется с помощью стандартной кнопки закрытия окна или любым другим методом, предусмотренным конкретной операционной системой. При закрытии будет вызвано диалоговое окно, уточняющее намерение выйти из программы. При выборе «Да» окно с макетом закроется, «Нет» закроет только диалоговое окно.

Данные при выходе из программы НЕ сохраняются автоматически.

3. Порядок выполнения работы.

1. **Получить у преподавателя вариант задания**(см. п. 4 - Варианты).
2. **Разработать алгоритм решения задачи**(см. рис.6.)
3. **Разработать систему машинных команд**(см п. 1.4 – Машинные команды и таблицу 8).
4. **Разработать микропрограммы машинных операций**(см. п 1.6. – Микрооперации и рис.7).
5. **Разработать обобщенную программу работы ЦУУ**(см. рис. 8), используя результаты выполнения пнуктов 2-4.
   В микропрограмме необходимо предусмотреть:
   - Выборку машинных команд из оперативной памяти в регистр команд
   - Расшифровку кода операции выбранной команды и выполнение соответствующей операции
   - Переход к выбору и выполнению следующей команды программы.
   - Загрузку начального адреса в программный счетчик
   - Окончание вычислений после выполнения программы
6. **Построить закодированный граф микропрограммы**(см. рис.9), используя списки микроопераций и логических условий. Состояния указываются в формате **a<номер состояния>.**
7. **Разработать управляющий автомат** (в соответствии с закодированным графом) по схеме Мура или Мили **и составить таблицу переходо**(см таблицу 9). Сигналы возбуждения **D6-D1** соответствуют битам следующего состояния, **D6** – старший бит, **D1** – младший. Сигнал возбуждения указывается в том случае, если соответствующий ему бит равен 1.
8. **Синтезировать управляющий автомат**(см. рис. 10) на основе ПЛМ и регистра, включенного в режим записи по тактовому сигналу:
   a. Определить разрядность памяти управляющего автомата:
      $K = \log_2 N$ (N – число состояний автомата).
   b. Опредилить число **m** использующихся в микропрограмме управляющих сигналов.
   c. Определить необходимое число ПЛМ (каждая ПЛМ имеет 16 входов, 8 выходов и может реализовать до 68 конъюнкций).
   d. Распределить между ПЛМ управляющие сигналы и сигналы возбуждения.
   e. Для каждой ПЛМ построить таблицу соединений(см. таблицу 10).
      На входы ПЛМ подаются коды состояния и осведомительные сигналы(условия), а с выхода снимаются сигналы возбуждения D6-D1 и управляющие сигналы. У первой ПЛМ на выходе будут только сигналы возбуждения.
9. **Составить машинную программу в мнемокодах** на основании алгоритма и набора машинных команд (см рис. 11).
10. **Составить карту памяти**(распределить страницы памяти между данными, командами, программой и подпрограммой, см рис. 12.)**.**

11. **Закодировать машинные команды в шестнадцатеричной системе счисления и составить программу в машинных кодах** (в соответствии с картой памяти и выбранными машинными командами, см. таблицу 11).
12. **Промоделировать работу ЦУУ с помощью электронного макета.**

Содержание отчета.

1. Постановка задачи и исходные данные.
2. Алгоритм решения задачи.
3. Программа в мнемокодах.
4. Программа в машинных кодах.
5. Карта памяти.
6. Фотография памяти.
7. Обобщенная микропрограмма.
8. Закодированный граф микропрограммы.
9. Список переходов.
10. Таблицы соединений ПЛМ.
11. Протокол работы ЦУУ.

Пример отчета по лабораторной работе находится в файле ЦУУ.doc.

Примеры отдельных компонентов можно посмотреть в конце данного пособия.

4. Варианты.

Таблица 5.

Вари-ант	Код задания	Вари-ант	Код задания	Вари-ант	Код задания	Вари-ант	Код задания	Вари-ант	Код задания
I	I-П3-I	21	5-П2-7	41	9-П3-9	61	13-П2-5	81	1-Р-8
2	2-П2-2	22	6-П1-8	42	10-КР1-7	62	14-П3-6	82	2-П3-9
3	3-П1-3	23	7-КР2-9	43	11-П1-7	63	15-КР1-4	83	3- П3-8
4	4-КР2-4	24	8-КР1-1	44	12-П3-6	64	16-Р-3	84	4-Р-7
5	5-КРI-5	25	9-Р-2	45	13-КР1-5	65	1-П2-2	85	5-Р-6
6	6-Р-6	26	10-П3-3	46	14-Р-4	66	2 - Р-3	86	6-П3 -5
7	7-П3-7	27	11-П2-4	47	15-П2-3	67	3-КР2-4	87	7-П2-4
8	8-П2-8	28	12-П1-5	48	16-КР1-2	68	4-П1-5	88	8-Р - 3
9	9-П1-9	29	13-КР2-6	49	1-КР1-1	69	5-П1-6	89	9-КР2-2
10	10-КР2 -1	30	14-КР1-7	50	2-П1-2	70	6-КР1-7	90	10-П1-1
11	11-КРI-2	31	15-Р-8	51	3-П2-3	71	7-Р-8	91	11-Р-2
12	12-Р-3	32	16-П1-9	52	4-КР1-4	72	8-П1-9	92	12-КР2-3
13	13-П3-4	33	1-П1-1	53	5-КР2-5	73	9-П2-8	93	13-П1-4
14	14-П2-5	34	2-КР2-2	54	6-П2-6	74	10-Р-9	94	14-П1-5
15	15-П1-6	35	3-КР1-3	55	7-КР1-7	75	11-П3-8	95	15-КР2-6
16	16 -П3-8	36	4-П2-4	56	8-П3-8	76	12-П2-7	96	16-Р-7
17	1-КР2-3	37	5-П3-5	57	9-КР1-9	77	13-Р-6	97	1-КР2-6
18	2-КР1-4	38	6-КР2-6	58	10-П2-8	78	14- КР2 - 5	98	2-КР1-7
19	3-Р-5	39	7-П1-7	59	11-КР2-7	79	15-П3-6	99	3-Р-9
20	4-П3-6	40	8-КР2-8	60	12-КР1-6	80	16-Р-7	100	4-КР2-8
101	5-КР1-3	107	11-КР1-7	113	1-П1-9	119	7-П1-2	125	13-П1-2
102	6-Р-2	108	12-Р-8	114	2-П2-6	120	8-П2-4	126	14-П2-3
103	7-П3-1	109	13-П3-9	115	3-П3-7	121	9-П3-6	127	15-П3-4
104	8-КР1-9	110	14-КР1-1	116	4-КР1-6	122	10-КР1-5	128	16-КР1-5
105	9-Р-5	111	15-Р-2	117	5-КР2-1	123	11-КР2-3	129	1-КР2-6
106	10-КР2-6	112	16-КР2-7	118	6-Р-8	124	12-Р-1	130	2-Р - 7

Расшифровка кода задания:

Код задания состоит из трех частей, разделенных знаком “-“.
Первая часть – номер задачи, которую должна выполнять программа(см. перечень задач на следующей странице).
Вторая часть – формат команды и способ адресации(см. таблицу 6).
Третья часть – параметры структуры ЦУУ(см.таблицу 7).
Пример: 16-П3-8
Номер задачи – 16.
П3 – используются трехадресные команды с прямой адресацией, длина команды – 4 байта.
8 – программный счетчик хранится в РОН, адрес возврата – в ОП, индекс – в РИ.

Перечень задач.

*** -** означает, что размерность массивов n задается с клавиатуры.
Результаты всех задач необходимо вывести на дисплей.

1. Вычислить:
   $C_i = 2A_i - B_i$, i=1, n

2. Вычислить:
   Ci = Ai + 2Bi , i=1,n

3. Переписать из массива A в массив B все числа, большие единицы.

4. Поменять местами в массивах A и B элементы с одинаковыми порядковыми номерами*.

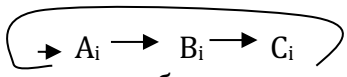
5. Вычислить
   $$C=\sum_{i=1}^{n}(A_i+B_i),\ \overline{i=1,n^*}$$

6. Найти максимальный элемент в массиве A и записать его в оперативную память*.

7. Найти минимальный элемент в массиве A и записать его в оперативную память*.

8. В массиве A все отрицательрые элементы преобразовать в дополнительный код и найти их сумму*.

9. Поменять местами соответствующие элементы в массивах A, B и C по схеме

   
   , $i=\overline{1,n^*}$

10. Выполнить преобразование Ci (7:0) = Ai(7:4).0000, i=1,n*.

11. Вычислить полную сумму четных элементов двух массивов A и B.

12. Вычислить полную сумму нечетных элементов двух массивов A и B.

13. Подсчитать число четных элементов в массивах A и B*.

14. Подсчитать число нечетных элементов в массивах A и B*.

15. В массивах A и B найти сумму элементов , лежащих в интервале 3…9*.

16. Вычислить Ci= Ai + Bi, i=1,n. Вывести на дисплей n и Ci.

Таблица 6.

Обозначение	Тип команд	Длина команды, байт
П3	Трехадресная с прямой адресацией	4
П2	Двухадресная с прямой адресацией	3
П1	Одноадресная с прямой адресацией	2
КР2	Двухадресная с косвенной регистровой адресацией	2
КР1	Одноадресная с косвенной регистровой адресацией	2
Р	Прямая регистровая адресация (двухадресные форматы RR и RS)	2

Таблица 7.

	Местонахождение компонент и код структуры ЦУУ								
Компоненты ЦУУ	1	2	3	4	5	6	7	8	9
Программный счетчик	РОН	РОН	РОН	РОН	РС	РС	РС	РОН	РС
Индексный регистр	РОН	РОН	РОН	РОН	РИ	РИ	РОН	РИ	РИ
Адрес возврата	РВ	ОП	Стек	Стек	Стек	Стек	ОП	ОП	РВ
Указатель стека	-	-	РОН	-	РОН	-	-	-	-

5. Примеры.

Пример полного выполнения работы можно найти в файле ЦУУ.doc.

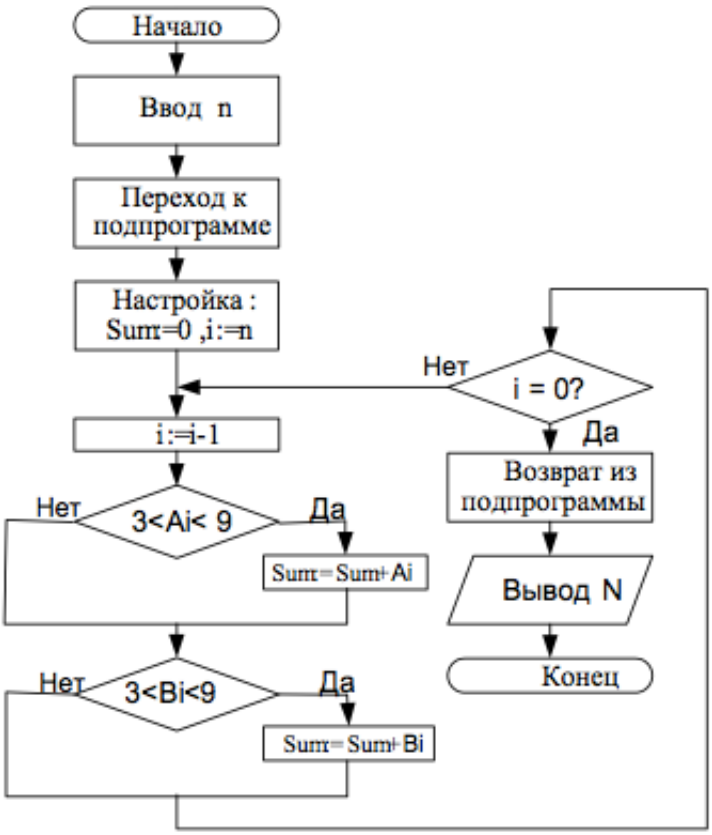


Рис.6. Пример алгоритма решения задачи.

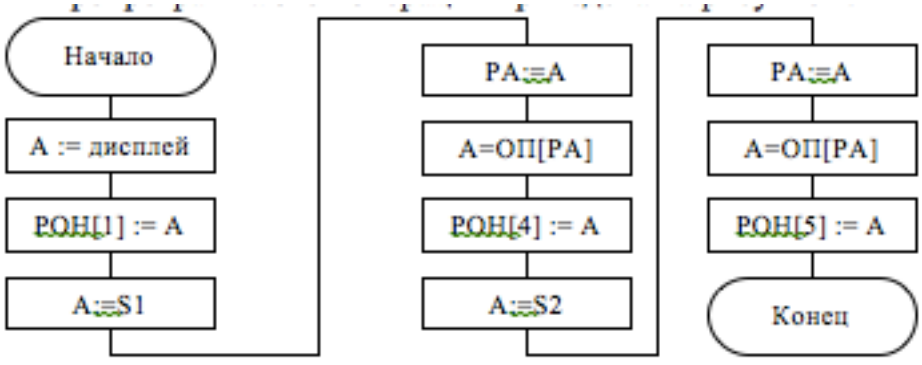


Рис.7. Пример алгоритма микропрограммы «Установка индекса»

Таблица 8. Пример системы команд.

Формат команды	Мнемокод	КОП	Примечание
КОП | - | S1 | S2	УИ	0000	«Установка индексов» Дисплей→РОН[1] ОП[S1] →РОН[4] ОП[S1] →РОН[5]
КОП | - | S1 | -	ПВ	0001	«Переход с возвратом» (переход к подпрограмме) РОН[0] → SP S1 → РОН[0]
КОП | - | S1 | -	СРА	0010	«Сравнение из массива А» PC := ⎧ PC + 4, если i<=3 или i>=9 ⎩ PC + 2, если 3<i<9
КОП | - | S1 | -	СРВ	0011	«Сравнение из массива В» PC := ⎧ PC + 4, если i<=3 или i>=9 ⎩ PC + 2, если 3<i<9
КОП | - | S1 | -	УП	0100	«Условный переход по ненулевому индексу» РОН[0] = РОН[0]+3, при i = 0 S1, при i ≠ 0
КОП | - | - | -	ПБК	0101	«Переход безусловный» SP → РОН[0]
КОП | - | - | -	ДИ	0110	«Декремент индекса» РОН[1]:=РОН[1]-1
КОП | - | - | -	ВЫВ	0111	«Вывод на экран» РОН[6] →Дисплей РОН[7] →Дисплей
КОП | - | - | -	Стоп	1000	«Остановка» «Стоп» → Дисплей

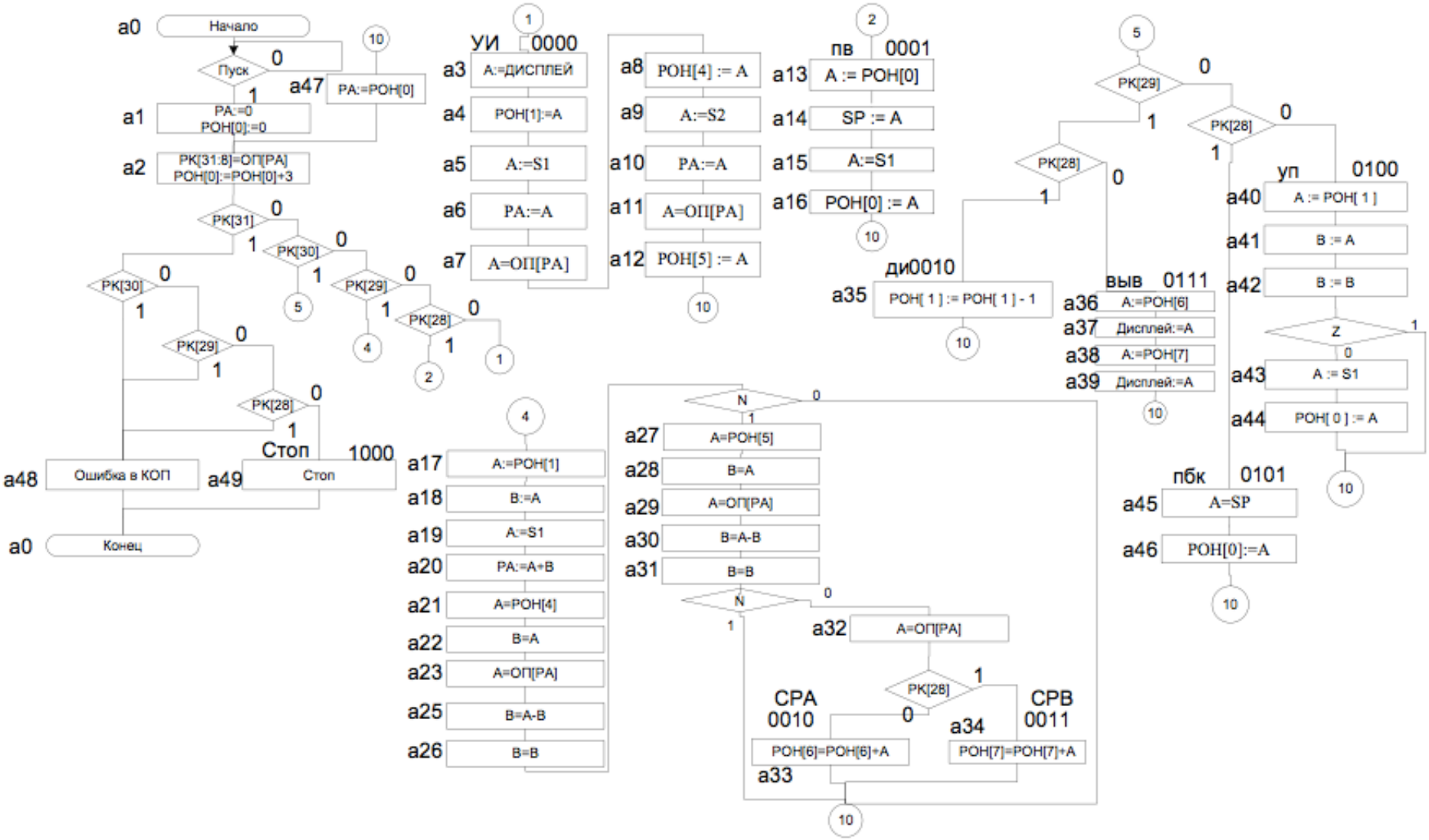


Рис.8. Пример схемы обобщенной микропрограммы.

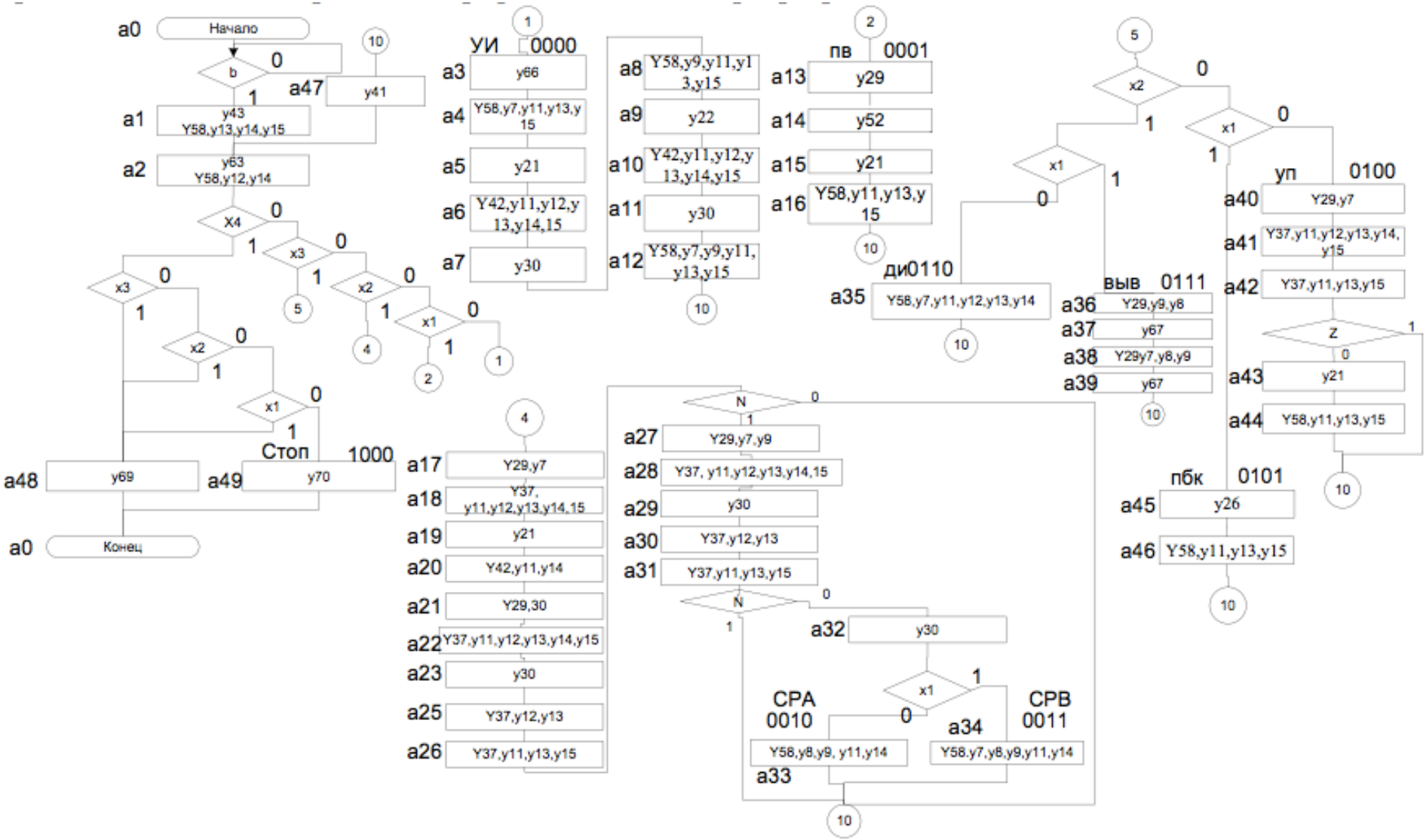


Рис. 9. Пример закодированного графа обобщенной микропрограммы.

Таблица 9. Пример списка переходов.

№ п п	Исх. сост	Код	След. сост.	Код	Входные сигналы	Ñèãíàëû âîçáóæäåíèÿ	Выходные сигналы
1	a0	000000	a0	000000	!B		
2			a1	000001	B	D1	
3	a1	000001	a2	000010		D2	y13, y14, y15, y43, y58
4	a2	000010	a3	000011	!x4,!x3, !x2, !x1	D2, D1	y12,y14, y58, y63
5			a13	001101	!x4, !x3, !x2, x1	D4, D3, D1	
6			a17	001010	!x4, !x3, x2	D4, D2	
7			a40	101000	!x4, x3, !x2, ⌉x1	D6, D5	
8			a45	101101	!x4, x3, !x2, x1	D6, D4, D3,D1	
9			a35	100011	!x4, x3, x2, !x1	D6, D2, D1	
10			a36	100100	!x4, x3, x2, x1	D6, D3	
11			a49	110001	x4, !x3, !x2, !x1	D6, D5, D1	
12			a48	110000	x4, x1	D6, D5	
13			a48	110000	x4, x2	D6, D5	
14			a48	110000	x4, x3	D6, D5	
15	a3	000011	a4	000100		D3	y66
16	a4	000100	a5	000101		D3, D1	y58,y7,y11,y13,y15
17	a5	000101	a6	000110		D3, D2	y21
18	a6	000110	a7	000111		D3, D2, D1	y42,y11,y12,y13,y14,y15
19	a7	000111	a8	001000		D4	y30
20	a8	001000	a9	001001		D4, D1	y58,y9,y11,y13,y15
21	a9	001001	a10	001010		D4, D2	y22
22	a10	001010	a11	001011		D4, D2, D1	y42, y11,y12,y13,y14,y15
23	a11	001011	a12	001100		D4, D3	Y30
24	a12	001100	a47	101111		D6, D4, D3, D2, D1	y58,y7,y9,y11,y13,y15
25	a13	001101	a14	001110		D4, D3, D2	y29
26	a14	001110	A15	001111		D4, D3, D2, D1	y52
27	a15	001111	a16	010000		D5	y21
28	a16	010000	a47	101111		D6, D4, D3, D2, D1	y58,y11,y13,y15
29	a17	010001	A18	010001		D5, D1	y29,y7
30	a18	010010	A19	010011		D5, D2, D1	y37, y11,y12,y13,y14,y15
31	a19	010011	a20	010100		D5, D3	y21
32	a20	010100	a21	010101		D5, D3, D1	y42,y11,y14
33	a21	010101	a22	010110		D5, D3, D2	Y29, y9
34	a22	010110	a23	010111		D5, D3, D2, D1	y37, y11,y12,y13,y14,y15
35	a23	010111	a25	011000		D5, D4	Y30

36	a25	011001	a26	011010		D5, D4, D2	y37,y12,y13
37	a26	011010	a27	011011	N	D5, D4, D2, D1	Y37, y11, y13, y15
38			a47	101111	!N	D6, D4, D3, D2, D1	
39	a27	011011	a28	011100		D5,D4,D3	y29,y7,y9
40	a28	011100	a29	011101		D5, D4, D3, D1	Y37, y11,y12,y13,y14,y15
41	a29	011101	a30	011110		D5, D4, D3, D2	Y30
42	a30	011110	a31	011111		D6, D3, D1	y37,y12,y13
43	a31	011111	a47	101111	N	D6, D4, D3, D2, D1	y37,y11,y13,y15
44			a32	100000	!N	D6	
45	a32	100000	a34	100010	X1	D6, D2	y30
46			a33	100001	!X1	D6,D1	
47	a33	100001	a47	101111		D6, D4, D3, D2, D1	y58,y8,y9,y11,y14
48	a34	100010	a47	101111		D6, D4, D3, D2, D1	y58,y7,y8,y9,y11,y14
49	a35	100011	a47	101111		D6, D4, D3, D2, D1	y58, y7, y11,y12,y13,y14
50	a36	100100	A37	100101		D6, D3, D1	y29,y9,y8
51	a37	100101	A38	100110		D6, D3, D2	y67
52	a38	100110	A39	100111		D6, D3, D2,D1	y29,y7,y8,y9
53	a39	100111	A47	101111		D6, D4, D3, D2, D1	y67
54	a40	101000	a41	101001		D6, D4, D1	y29,y7
55	a41	101001	a42	101010		D6,D4,D2	y37, y11,y12,y13,y14,y15
56	a42	101010	a43	101011	!Z	D6,D4,D2,D1	y37,y11,y13,y15
57			a47	101111	Z	D6,D4,D3,D2,D1	
58	a43	101011	a44	101100		D6,D4,D3	y21
59	a44	101100	a47	101111		D6,D4,D3,D2,D1	y58,y11,y13,y15
60	a45	101101	a46	101110		D6,D4,D3,D2	y26
61	a46	101110	a47	101111		D6,D4,D3,D2,D1	y58,y11,y13,y15
62	a47	101111	a2	000010		D2	y41
63	a48	110000	a0	000000			y69
64	a49	110001	a0	000000			y70

Таблица 10. Пример матрицы ПЛМ-1

#	F6	F5	F4	F3	F2	F1	x1	x2	x3	x4	z	n	B	D6	D5	D4	D3	D2	D1	Y7	Y9
1	0	0	0	0	0	0	*	*	*	*	*	*	0								
2	0	0	0	0	0	0	*	*	*	*	*	*	1						1		
3	0	0	0	0	0	1	*	*	*	*	*	*	*					1			
4	0	0	0	0	1	0	0	0	0	0	*	*	*					1	1		
5	0	0	0	0	1	0	1	0	0	0	*	*	*			1	1		1		
6	0	0	0	0	1	0	*	1	0	0	*	*	*			1		1			
7	0	0	0	0	1	0	0	0	1	0	*	*	*	1	1						
8	0	0	0	0	1	0	1	0	1	0	*	*	*	1		1	1		1		
9	0	0	0	0	1	0	0	1	0	0	*	*	*	1				1	1		
10	0	0	0	0	1	0	1	1	1	0	*	*	*	1			1				
11	0	0	0	0	1	0	0	0	0	1	*	*	*	1	1				1		
12	0	0	0	0	1	0	1	*	*	1	*	*	*	1	1						
13	0	0	0	0	1	0	*	1	*	1	*	*	*	1	1						
14	0	0	0	0	1	0	*	*	1	1	*	*	*	1	1						
15	0	0	0	0	1	1	*	*	*	*	*	*	*				1				
16	0	0	0	1	0	0	*	*	*	*	*	*	*				1		1	1	
17	0	0	0	1	0	1	*	*	*	*	*	*	*				1	1			
18	0	0	0	1	1	0	*	*	*	*	*	*	*				1	1	1		
19	0	0	0	1	1	1	*	*	*	*	*	*	*			1					
20	0	0	1	0	0	0	*	*	*	*	*	*	*			1			1		1
21	0	0	1	0	0	1	*	*	*	*	*	*	*			1		1			
22	0	0	1	0	1	0	*	*	*	*	*	*	*			1		1	1		
23	0	0	1	0	1	1	*	*	*	*	*	*	*			1	1				
24	0	0	1	1	0	0	*	*	*	*	*	*	*	1		1	1	1	1	1	1
25	0	0	1	1	0	1	*	*	*	*	*	*	*			1	1	1			
26	0	0	1	1	1	0	*	*	*	*	*	*	*			1	1	1	1		
27	0	0	1	1	1	1	*	*	*	*	*	*	*		1						
28	0	1	0	0	0	0	*	*	*	*	*	*	*	1		1	1	1	1		
29	0	1	0	0	0	1	*	*	*	*	*	*	*		1				1	1	
30	0	1	0	0	1	0	*	*	*	*	*	*	*		1			1	1		
31	0	1	0	0	1	1	*	*	*	*	*	*	*		1		1				
32	0	1	0	1	0	0	*	*	*	*	*	*	*		1		1		1		
33	0	1	0	1	0	1	*	*	*	*	*	*	*		1		1	1			1
34	0	1	0	1	1	0	*	*	*	*	*	*	*		1		1	1	1		
35	0	1	0	1	1	1	*	*	*	*	*	*	*		1	1					
36	0	1	1	0	0	1	*	*	*	*	*	*	*		1	1		1			
37	0	1	1	0	1	0	*	*	*	*	*	1	*		1	1		1	1		
38	0	1	1	0	1	0	*	*	*	*	*	0	*	1		1	1	1	1		
39	0	1	1	0	1	1	*	*	*	*	*	*	*		1	1	1			1	1
40	0	1	1	1	0	0	*	*	*	*	*	*	*		1	1	1		1		
41	0	1	1	1	0	1	*	*	*	*	*	*	*		1	1	1	1			
42	0	1	1	1	1	0	*	*	*	*	*	*	*	1			1		1		
43	0	1	1	1	1	1	*	*	*	*	*	1	*	1		1	1	1	1		
44	0	1	1	1	1	1	*	*	*	*	*	0	*	1							
45	1	0	0	0	0	0	1	*	*	*	*	*	*	1				1			
46	1	0	0	0	0	0	0	*	*	*	*	*	*	1					1		
47	1	0	0	0	0	1	*	*	*	*	*	*	*	1		1	1	1	1		1

48	1	0	0	0	1	0	*	*	*	*	*	*	*	1		1	1	1	1		
49	1	0	0	0	1	1	*	*	*	*	*	*	*	1		1	1	1	1		
50	1	0	0	1	0	0	*	*	*	*	*	*	*	1			1		1		
51	1	0	0	1	0	1	*	*	*	*	*	*	*	1			1	1			
52	1	0	0	1	1	0	*	*	*	*	*	*	*	1			1	1	1		
53	1	0	0	1	1	1	*	*	*	*	*	*	*	1		1	1	1	1		
54	1	0	1	0	0	0	*	*	*	*	*	*	*	1		1			1	1	
55	1	0	1	0	0	1	*	*	*	*	*	*	*	1		1		1			
56	1	0	1	0	1	0	*	*	*	*	0	*	*	1		1		1	1		
57	1	0	1	0	1	0	*	*	*	*	1	*	*	1		1	1	1	1		
58	1	0	1	0	1	1	*	*	*	*	*	*	*	1		1	1				
59	1	0	1	1	0	0	*	*	*	*	*	*	*	1		1	1	1	1		
60	1	0	1	1	0	1	*	*	*	*	*	*	*	1		1	1	1			
61	1	0	1	1	1	0	*	*	*	*	*	*	*	1		1	1	1	1		
62	1	0	1	1	1	1	*	*	*	*	*	*	*					1			
63	1	1	0	0	0	0	*	*	*	*	*	*	*								
64	1	1	0	0	0	1	*	*	*	*	*	*	*								

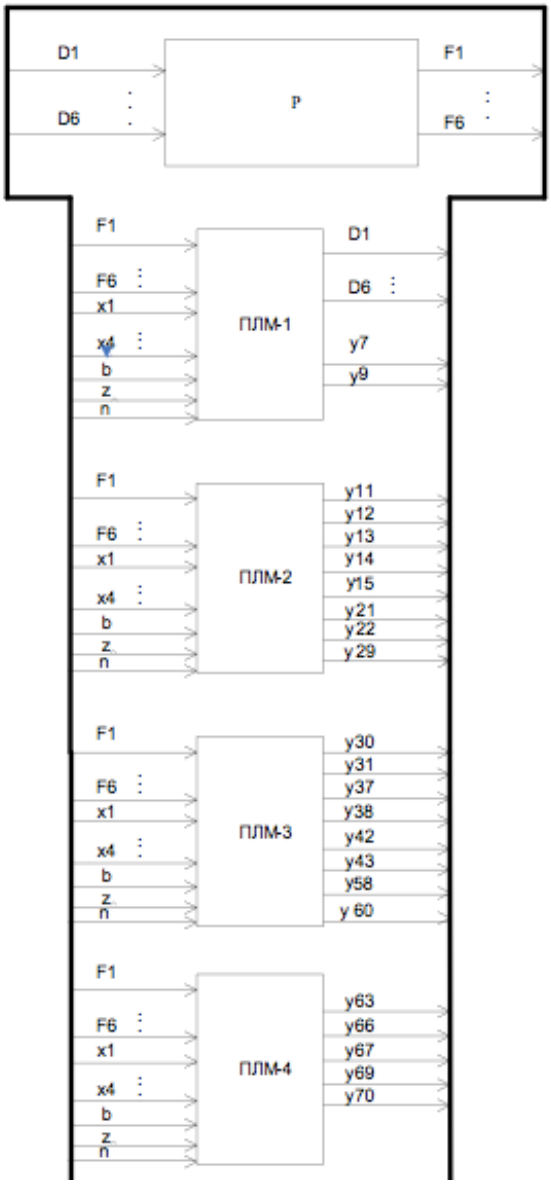


Рис. 10. Пример функциональной схемы управляющего автомата.

№	Адрес	Код команды			Примечание
пп.	ОП	Байт 1	Байт 2	Байт 3	
1	00	00	40	41	Установка индексов
2	03	10	60	00	Переход к подпрограмме
3	06	70	00	00	Вывод на экран
4	09	80	00	00	Остановка
Подпрограмма					
5	60	60	00	00	Декремент индекса
6	63	20	20	00	Сравнение из массива A
7	66	30	30	00	Сравнение из массива B
8	69	40	60	00	Условный переход по ненулевому индексу
9	6C	50	00	00	Переход безусловный

Рис.11. Пример машинной программы.

а) Страница 0 – основная программа;

б) Страница 2 – массив A;(сумма 1E)

в) Страница 3 – массив B;(сумма 22)

г) Страница 4 – границы(3 и 8)

д) Страница 6-7 – подпрограмма;

Рис.12. Пример карты памяти.

Таблица 11. Пример фотографии ОП.

Адрес страницы	**Адрес слова в странице**															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	40	41	10	60	00	70	00	00	80	00	00				
1																
2	01	09	11	02	03	04	05	06	00	09	07	08	00	03	30	10
3	0A	0C	01	05	06	07	04	01	01	01	00	0F	04	08	09	31
4	08	03														
5																
6	60	00	00	20	20	00	30	30	00	40	60	00	50	00	00	
7																
8																
9																
A																
B																
C																
D																
E																
F																