Министерство высшего и среднего специального
образования РСФСР

Куйбышевский ордена Трудового Красного Знамени
авиационный институт имени С.П.Королева

ИССЛЕДОВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ НА МИКРОПРОЦЕССОРАХ

Лабораторная работа

Куйбышев 1980

УДК 681.3

Составители: С.Г. Акимов, В.Г. Иоффе,
П.А. Соколов

Утверждена на редакционно-издательском
совете института 9.01.80 г.

Ц е л ь р а б о т ы - исследование типовой структуры (архитектуры) центральной части вычислительной системы на микропроцессорных БИС.

I. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИССЛЕДОВАНИЯ

Современные ВС ("большие", типа ЕС ЭВМ, БЭСМ-6, мини-ЭВМ, типа СМ ЭВМ, а также микро-ЭВМ на основе микропроцессоров) в своей центральной части (ядре) содержат в общем случае следующие модули (рис. I):

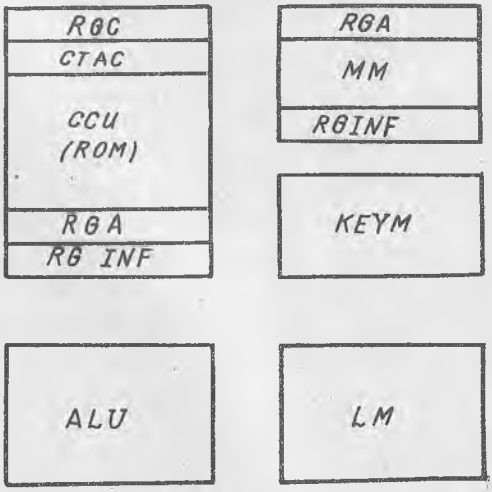
ALU - арифметическое-логическое устройство;

CCU - центральное управляющее устройство;

LM - локальная или местная память процессора (эти три модуля образуют центральный процессор)

MM - основная (оперативная) память;

KEYM - память ключей защиты.



Р и с. I.

Рассмотрим вкратце функции и взаимосвязи этих компонент.

ALU перерабатывает информацию, т.е. выполняет операции над данными и адресами, адресными компонентами. Чтобы выполнить операцию в *ALU*, следует подать в него операнды (операнд), сообщить ему операцию (код операции) и запустить его. *ALU* содержит операционную часть (регистры и сумматоры) и управляющую часть, локальное управление *ALU, LCALU*.

В настоящее время электронная промышленность выпускает БИС, которые содержат либо целиком *ALU*, либо отдельно операционную часть и отдельно управляющую часть, либо микропроцессоры, состоящие из *ALU* и локальной памяти *LM*, либо, наконец, микропроцессоры, содержащие, кроме *ALU*, многие другие компоненты. Разрядность БИС - 4, 8, 16 бит. В настоящей работе в качестве *ALU* используются микропроцессоры, имеющие разрядность 4 бит. Этим микропроцессорам, кроме функций *ALU*, приданы и некоторые другие функции.

CCU организует взаимодействие, регламентирует во времени работу всех центральных модулей ЭВМ. С другой точки зрения, можно сказать, что *CCU* реализует рабочий цикл ЭВМ. За один цикл ЭВМ исполняет одну команду программы (при простой организации цикла).

Исполнение команды ЭВМ, т.е. цикл, распадается на следующие этапы.

Во-первых, из основной памяти (*MM*) выбирается код очередной команды, и он принимается на специальный регистр центрального управляющего устройства - регистр команд (*RGC*). На счетчике адреса команды (*CTAC*), принадлежащем также *CCU*, вычисляется адрес следующей команды (для следующего цикла).

Во-вторых, вычисляются адреса операндов. Они, как правило, определяются несколькими компонентами, которые даны соответствующими полями команды. *CCU* выбирает компоненты адресов и засылает их в *ALU*, где происходит вычисление адресов. Простые команды, например формата *RR*, содержат в своих полях готовые адреса.

В-третьих, *CCU* организует выборку операндов из основной (или локальной) памяти, т.е. *CCU* передает адрес операнда на адресный регистр памяти (*RGAM* или *RGAL*) и запускает последнюю для выборки. После срабатывания памяти *CCU* пересылает операнд из

информационного регистра памяти (*RGINFM* или *RCINFL*) в определенный приемный регистр *ALU*.

В-четвертых, после засылки всех необходимых операндов в *ALU*, *CCU* запускает *ALU*, которое выполняет операцию.

В-пятых, *CCU* определяет адрес засылки результата операции, передает адрес на адресный регистр соответствующей памяти, передает результат из *ALU* в информационный регистр той же памяти и запускает память для записи. На этом исполнение команды и машинный цикл заканчивается. Процессор переходит на новый цикл, т.е. повторяет перечисленные этапы. При этом исполняется следующая команда программы и т.д. Паспортное быстродействие ЭВМ, например 100000 оп/с, означает именно количество машинных циклов (в среднем), выполняемое в 1 секунду.

Перечисленная последовательность шагов в цикле относится к общему, "полному" случаю; многие команды не требуют всех шагов цикла. Например, команды с непосредственной адресацией *SI* содержат один операнд в коде команды. Следовательно, при исполнении такой команды из памяти будет выбираться только один операнд, этап выборки второго операнда отсутствует.

Центральное управление запускает модули процессора – *ALU, LM, MM* и др., и после запуска эти модули работают автономно от *CCU* под управлением собственного, локального устройства управления (*LC*). По окончании работы модуля, его локальное устройство управления возвращает управление центральному устройству.

Современные устройства управления строятся на основе исполнения записанных в их памяти микропрограмм – алгоритмов работы управляемых устройств (принцип Уилкса). Мы вернемся к подробному рассмотрению принципа Уилкса после перечисления остальных модулей ЭВМ.

Локальная, местная память (*LM*) процессора называется еще сверхоперативной и вводится в состав процессора с целью повышения его производительности. При работе программ замечена следующая статистическая закономерность. Если программа обратилась в память за данными по адресу i, то наиболее вероятными адресами обращения к памяти в ближайших командах являются соседние адреса $(i+1), (i+2)$ и т.д. Другими словами, в среднем действия программы в каждом интервале времени локализуются в некотором компактном участке памяти. Эта закономерность используется при выборе различных конст-

руктивных решений. В частности, если поместить очередной участок (сегмент) данных в небольшую и, следовательно, быструю память, то весьма вероятно, что текущий участок программы будет в основном обращаться именно к этой малой памяти и, следовательно, исполнится быстрее. Периодически выполняется "подкачка" информации из основной памяти в локальную. Кроме данных, в *LM* хранится другая, заведомо "популярная" информация, - индексы, базы, информация операционной системы. Применение локальной памяти дает еще одно преимущество. Поскольку емкость *LM* мала (она находится в пределах 8-256 слов), постольку величина адресных полей для ее адресации также мала. Поэтому длина (формат) команд, предназначенных для работы с *LM* существенно меньше, чем длина команд, работающих с основной памятью. Например, в ЕС ЭВМ команды формата *RR*, работающие с *LM*, имеют длину 2 байта, а команды, работающие с основной памятью - *RX*, *SS* и др., имеют длину 4 и 6 байт. Отсюда получается существенное сокращение объема текста программы (вследствие применения *LM*) и соответствующее сокращение емкости памяти, потребной для хранения программы.

В машинах ЕС ЭВМ, а затем и в других разработках, в частности в микропроцессорах, локальная память называется еще регистровой памятью или памятью на РОНах, регистрах общего назначения.

В лабораторном макете локальная память *LM* не предусмотрена, однако микропроцессоры содержат РОНы. Следует иметь в виду, что эти РОНы не адресуются командами программы, т.е. не используются в вышеописанном качестве, используются лишь микрокомандами.

Основная память (*MM*) предназначена для хранения программ и данных. Она прямо адресуема, т.е. выдает и принимает информацию по адресу, поданному на ее вход, адресный регистр *RCAM*. Считанная из памяти информация фиксируется на ее информационном регистре (*RGINFM*). При записи информации в память, записываемая информация подается на *RGINFM*. Кроме адреса и информации (при записи), на вход памяти из центрального управления должен быть подан сигнал, определяющий цель обращения - считывание или запись. Обращение к памяти инициируется (запускается) центральным управлением. После запуска работа памяти выполняется под управлением собственного локального устройства управления (*LCM*). По окончании обращения управление возвращается от локального устройства к центральному.

I.I. Принцип микропрограммного управления

Рассмотрим теперь подробнее принцип Уилкса реализации функций устройства управления. Этот принцип – альтернатива традиционному методу, когда управляющий автомат создается на логических (комбинационных) и регистровых элементах. Этот последний метод называется аппаратным в отличие от метода Уилкса, который называется микропрограммным. Принцип Уилкса заключается в следующем. Управляющие сигналы на управляемые элементы получаются посредством трансформации слов, считываемых из памяти микропрограммы *(ROM)*. Все управляемые элементы управляются посредством управляющих шин, например, сдвигающий регистр выполняет сдвиг при подаче сигнала на управляющую шину сдвига, счетчик выполняет операцию прибавления единицы при подаче сигнала на шину *INC*, запоминающее устройство выполняет обращение при подаче сигнала на шину запуска ЗУ и т.д. При управлении по методу Уилкса все управляющие шины берут начало от выхода памяти микропрограмм *(ROM)*. Если на j -м шаге (микрокоманде) следует подать сигнал на i -ю и k -ю управляющие шины, то в j -й микрокоманде записываются единицы в i -й и k -й разряды, в остальные – нули. Буду-

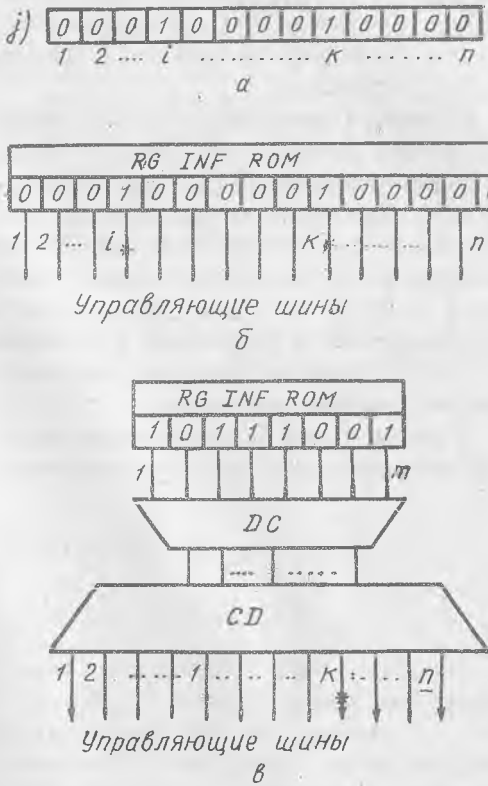


Рис. 2.

чи считанным из памяти, код j -й микрокоманды превращается в сигналы на управляющих шинах, т.е. в единичные сигналы на i -й и k -й шины и нулевые - на остальных (рис. 2,а,б). В следующем такте считывается микрокоманда из следующей ячейки *ROM*, она содержит свои управляющие сигналы, и т.д.

Практически описанное здесь унитарное кодирование микрокоманд применяется редко, поскольку оно неэкономично, требует большой длины слова микрокоманды (она равна, очевидно, числу всех управляющих шин в центральных устройствах, n). Чаще применяется позиционное кодирование микрокоманд. В этом случае каждая микрокоманда (т.е. совокупность микроопераций, управляющих сигналов) получает уникальный позиционный код. Выходное слово *ROM* подается на дешифратор, выходы которого подаются на шифратор, кодер. Выходы кодера соединены с управляющими шинами (рис. 2,в). В этом случае длина слова микрокоманды m равна двоичному логарифму числа различных микрокоманд.

Наконец, используется еще один способ кодирования микрокоманд, который занимает промежуточное положение между вышеописанными крайними способами. Все управляющие шины делятся на категории, так, чтобы в каждой категории одновременно возбуждалось не более одной шины. Каждой категории шин ставится в соответствие отдельное поле в слове микрокоманды. Таким образом, слово микрокоманды превращается в совокупность автономных полей. В каждом поле осуществляется позиционное кодирование и для каждого поля на выходе *ROM* ставится отдельный дешифратор. Выходы дешифраторов управляют управляющими шинами.

В данном случае в лабораторном макете, слово микрокоманды имеет несколько полей, которые кодируются позиционными кодами.

2. ОПИСАНИЕ ЛАБОРАТОРНОГО МАКЕТА

2.I. С о с т а в с и с т е м ы

Структура лабораторного макета представлена на рис. 3, (см. вкладку) вид лицевой панели - на рис. 4.

MPU содержит два микропроцессора (или центральных процессорных элементов) серии К584ИК1, соединенных последовательно. Микропроцессоры получают на вход по Шин.КОП код микрокоманды. на

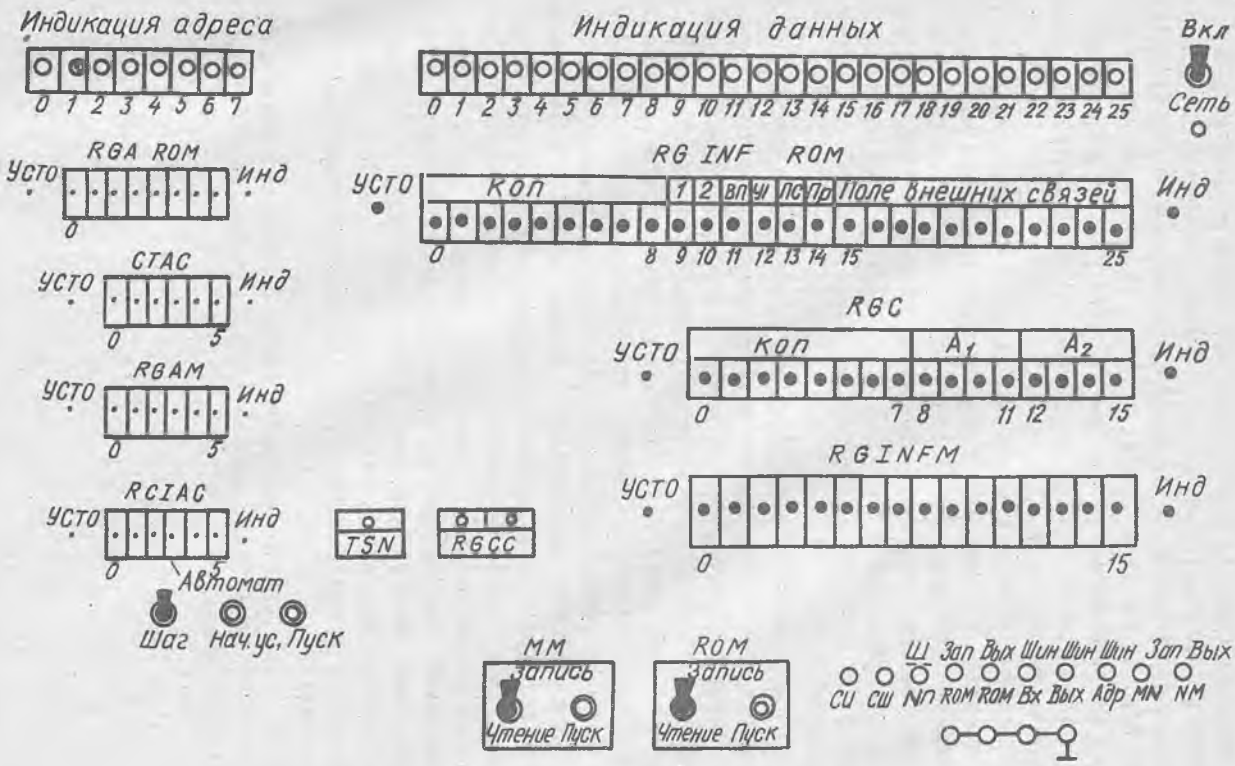

Индикация адреса
0 1 2 3 4 5 6 7
Индикация данных
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Вкл
Сеть
RGA ROM
Усто
Инд
0
RG INF ROM
Усто
КОП
1
2
ВП
У
ПС
Пр
Поле внешних связей
Инд
0
8 9 10 11 12 13 14 15
25
СТАС
Усто
Инд
0
5
RGAM
Усто
Инд
0
5
RGC
Усто
КОП
A1
A2
Инд
0
7 8
11 12
15
RCIAC
Усто
Инд
0
5
Автомат
TSN
RGCC
RGINFM
Усто
Инд
0
15
Шаг
Нач.ус.
Пуск
MM
Запись
Чтение
Пуск
ROM
Запись
Чтение
Пуск
Ш
Зап
Вых
Шин
Шин
Шин
Зап
Вых
СИ
СШ
NП
ROM
ROM
Вх
Вых
Адр
MN
NM

Рис. 4.

Шин.Вх. - входное слово информации, на Вх.ПАЛУ - входной сигнал переноса в младший разряд, на вход Упр.ИНКР. - сигнал для управления инкрементором, на вход П.СТ. - инверсный входной сигнал программного счетчика, на вход p - сигнал приоритета. Микропроцессоры выдают на Шин.Вых. выходное слово информации, на Шин.АДР - слово адреса. Это слово непосредственно используется для адресации памяти микропрограмм (*ROM*). Таким образом, на микропроцессорах реализованы функции *ALU* и функция *CCU* формирования адреса следующей микрокоманды.

К *MPU* относятся две схемы, реализующие аппаратные функции. Как известно, для выполнения арифметических операций над алгебраическими числами широко используется модифицированный код, содержащий два знаковых разряда. В данном случае второй (старший) знаковый разряд реализуется на схеме *TSN* - ЛСI. Первый (младший) знаковый разряд *SN* передается по Шин.Вх.0 и Шин.Вых.0 старшего микропроцессора.

Признак результата операции (*CC*) формируется аппаратно дешифратором *DC* и фиксируется на двухразрядном регистре *RGCC*.

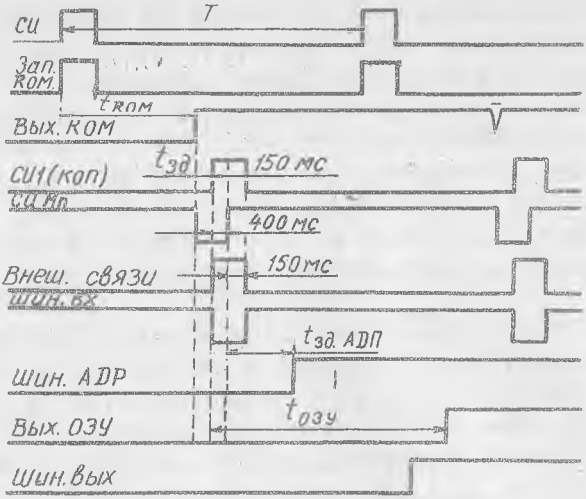
Память микропрограмм *ROM* построена на оперативном запоминающем устройстве. Очередная микрокоманда, считанная из *ROM*, хранится в течение такта на *RGINF ROM* Структура микрокоманды ясна из рисунка. Поле внешних связей микрокоманды содержит код управляющих сигналов, используемых для связи различных элементов структуры. Например, чтобы передать слово информации с *RGKOП* на Шин.Вх., необходимо выдать управляющий сигнал, замыкающий эту магистраль; данный сигнал относится к категории управления внешними связями.

Поля микрокоманды, обозначенные цифрами I, 2 (однобитные) предназначены для указания микропроцессоров, участвующих в данной микрокоманде. Микропроцессор № I - старший (левый), № 2 - младший.

К *ROM* примыкает регистр команд (*RGC*), имеющий простейшую структуру, и счетчик адреса команд (*СТАС*), вычисляющий адрес следующей команды.

RGIAC - регистр начального адреса команды - заполняется вручную и указывает адрес первой команды, которая будет исполнена после пуска макета.

2.2. Временная диаграмма системы



Р и с. 5.

На рис. 5 представлена временная диаграмма макета. Здесь принят синхронный метод тактирования и используются три серии синхронизирующих импульсов. Основная серия – СИ, по ней происходит запуск *ROM* . Через время t_{ROM} на выходе ЗУ появляется новая микрокоманда. Она передается в исполнительные цепи, т.е. в микропроцессоры и внешние связи, с помощью смещенной серии СИ1. Для тактирования микропроцессоров используется третья серия СИМП, которая несколько опережает СИ1. Минимальный период синхронизации T определяется из анализа следующих трех возможностей:

$$T = макс \begin{cases} t_{ROM} + t_{СИОМП} + t_{зд\ АДР} \\ t_{СИОМП} + t_{СИ1МП} \\ t_{зд} + t_{озу} \end{cases}$$

ЛС2, ЛС3 – логические комбинационные схемы, формирующие сигналы на Вх.ПАЛУ2 и Упр.ИНКР2.

2.3. Порядок работы макета

Из ММ извлекается очередная команда программы. Она принимается на регистр команд *RCC* . Она исполняется посредством обращения к соответствующей микропрограмме, хранящейся в *ROM*. КОП команды определяет адрес начала специальной части микропрограммы. Очередная микрокоманда считывается из *ROM* и принимается на *RGINF ROM*. В текущем такте она управляет всеми процессами в микро-ЭВМ. Поле КОП микрокоманды передается на Шин.КОП микропроцессоров, указанных в полях I, 2. Управляющие сигналы через логические схемы 2, 3 поступают на соответствующие входы микропроцессоров.

Поле внешних связей имеет четыре отдельных компоненты, четыре "элементарных" поля, кодируемых позиционным кодом. Поле, состоящее из одного I5-го разряда, кодирует способ интерпретации поля внешних связей. Если в I5-м разряде записана единица, разряды I6-25 микрокоманды интерпретируются как константа, которая передается на Шин.Вх.

Если в I5-м разряде записан ноль, разряды I6-25 микрокоманды интерпретируются как три поля. Поле I6-I9 кодирует различные управляющие сигналы, одноразрядные микрооперации. Например, код 0010 означает *TSN:=TSN*. Шин.Вх. *SN* , код 1001 - *RGCC:=DC* и т.п., (табл. I и 2).

Поля 20-22 и 23-25 управляют передачами кодов по магистрали от различных источников к различным приемникам. Поле 20-22 содержит код источника информации, например, код 001 означает *RG* КОП. Поле 23-25 содержит код приемника информации, например, код 110 означает Шин.Вх. (см. табл. I). Тогда код 001 110 в полях 20-25 означает микрооперацию Шин.Вх.: = *RGКОП*.

Значения кодов микроопераций можно узнать из блок-схемы макета (см. рис. 3). Цифрой в круге обозначен код для поля источника 20-23, цифрой в квадрате обозначен код для поля приемника 23-25, просто цифры означают коды однобитных операций в поле I6-I9 их полный перечень в табл. 2.

Итак, в текущей микрокоманде в общем случае параллельно исполняется микрооперация в микропроцессорах (или одном) и микрооперации во вне микропроцессоров, например передачи кодов. Кроме того, параллельно же в микропроцессорах вычисляется адрес сле-

Таблица I

Поля внешних связей

Ключ	Управление				Источник			Приемник			Обозначение
I5	I6	I7	I8	I9	20	2I	22	23	24	25	
0	0	0	0	0	0	0	I	I	I	0	Шин.Вх.: = *RG* КОП
0	I	0	0	I	I	I	0	I	0	I	*RG* СС:=ДС; *RGINF*М:=Шин.Вых
0	0	0	0	0	0	0	0	I	I	I	*WRITE* М ;
I	0	0	0	0	0	I	I	0	0	0	Шин.Вх.: = 0II000
0	0	0	I	0	I	I	I	I	I	0	*TSN*: = *TSN* + Шин.Вх *SN*; Шин.Вх: = *RGINFM*

Таблица 2

Перечень микроопераций управления

I6	I7	I8	I9	Обозначение
0	0	0	I	*TSN* : = Шин.Вх.
0	0	I	0	*TSN* : = *TSN* + Шин.Вх.
0	0	I	I	*TSN* : = *TSN* + 7 Шин.Вх.
0	I	0	0	З/О МПI → Вх. ПАЛУ МП2
0	I	0	I	7 З/О МПI → УпрИНКР МП2
0	I	I	0	7 З/О МП2 → Упр ИНКР МП2
0	I	I	I	Вых. ПАЛУ МПI → Упр ИНКР МП2
I	0	0	0	*TSN* → Упр ИНКР МП2
I	0	0	I	*RGCC* : = ДС

дующей микрокоманды в *ROM* . Стандартно адрес формируется на программном счетчике (П.СТ.) и выдается на Шин.АДР, откуда передается в *ROM* . В *ROM* возбуждается выборка и через известное время (t_{ROM}) код очередной микрокоманды появляется на выходе *ROM* , на *RGINF ROM*. Начинается исполнение новой микрокоманды и т.д. Так исполняется микропрограмма, соответствующая одной команде программы. После этого начинается исполнение новой микропрограммы, точнее общего участка микропрограмм, реализующего выборку следующей команды из ММ. Код операции выбранной команды определяет адрес начала спецмикропрограммы в *ROM* , отвечающей этой конкретной команде и т.д.

2.4. Описание лицевой панели макета

На лицевую панель выведены два ряда ("слова") светодиодов для индикации информации в статике. Слева вверху находится ряд светодиодов "Индикация адреса", предназначенных для индикации содержимого регистров, обозначенных на панели ниже, под ним. Индикация и занесение информации делается с помощью электрического "щупа". Если вы хотите вывести на индикацию, например, СТАС, следует коснуться щупом контакта "Инд" около регистра СТАС.

Чтобы занести новую информацию в некоторый регистр, следует, во-первых, погасить его, прикоснувшись щупом клеммы У"0" у соответствующего регистра, и, во-вторых, занести единицы, касаясь щупом клемм соответствующих разрядов на изображении данного регистра. Естественно, если при этом была возбуждена клемма "Инд" данного регистра, то все манипуляции с ним отражаются на светодиодах

Справа расположен второй ряд светодиодов, предназначенный для индикации слов информации, показанных ниже под ним. Оперировать с ними- точно так же, как с регистрами адреса.

Ниже на панели расположены тумблеры и кнопки управления. В шаговом режиме при нажатии кнопки "Пуск" исполняется одна очередная микрокоманда, точнее один период СИ (см. рис. 5). В режиме "Авт" непрерывно исполняется вся программа.

Программа и данные заносятся в память ММ с помощью соответствующих тумблера "ЗП" - "ЧТ" и кнопки "Пуск". Для того, чтобы занести слово в ММ, следует набрать на *RGAMM* адрес, (индицируя его на светодиодах), затем набрать на *RGINFM* слово ин-

формации (также с помощью индицирования его на "информационных" светодиодах), поставить тумблер ММ в положение "ЗП" и нажать кнопку "Пуск" ММ. В результате этих манипуляций слово запишется в ММ. Для того, чтобы убедиться в этом, следует погасить (У"0") *RGINFM*, затем переключить тумблер ММ в положение "ЧТ" и нажать "Пуск". Считанное слово индицируется на светодиодах.

Аналогичное управление ("ЗП" - "ЧТ", "Пуск") имеется для памяти *ROM* . Перед работой на макете в *ROM* с помощью этого управления следует занести необходимые микропрограммы.

Далее на панели имеются клеммы, выводящие основные сигналы временной диаграммы - СИ, СИI и т.д.

3. СОДЕРЖАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

Основным содержанием данной работы является занесение в вычислительную систему заданных или написанных по заданию микропрограмм, их исполнение, просмотр на осциллографе временной диаграммы. В программах I, 2 даны примеры микропрограмм некоторых команд. Микропрограмма "Выборка команды" исполняется для любой команды. Чтобы обратиться к этой микропрограмме, достаточно обнулить адресный регистр *ROM* и ПСТ микропроцессоров. Это достигается кнопкой "Нач.уст.".

По окончании своей работы микропрограмма "Выборки" передает управление индивидуальной микропрограмме, соответствующей выбранной команде. Это достигается тем, что на программный счетчик ПСТ (РОН 7) передается код, равный КОП выбранной команды, (микрокоманда 6, программа I). Такой в следующем такте будет выбираться микрокоманда по адресу (КОП+I). Именно с этой ячейки в *ROM* записана индивидуальная микропрограмма.

По окончании индивидуальной микропрограммы следует передача на микропрограмму "Выборка команды". Если необходимо повторять циклически исполнение одной команды, чтобы, например, наблюдать временную диаграмму на осциллографе, то следует возвращаться к первой ячейке *ROM* (программа I). В этом случае к содержимому программного счетчика (K) следует добавить дополнительный код K, $[K]Д$, тогда на ПСТ образуется ноль и после стандартного приращения содержимого программного счетчика - единица.

Если же необходимо перейти к исполнению следующей по порядку команды, необходимо передать управление в микропрограмму "Выборки" во вторую ячейку. Это достигается тем, что к текущему значению ПСТ, равному K, добавляется $[K]Д + I$.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

I. Определить у преподавателя команду, подлежащую исполнению и исследованию.

2. Пользуясь записью соответствующих микропрограмм в Программах I, 2 и блок-схемой системы рис. 3, закодировать микропрограммы "Выборки команды" и индивидуальной части.

3. Определить место команды и операндов в ММ, закодировать команду, ввести ее и операнды в ММ. Проверить правильность ввода.

4. Ввести закодированные микропрограммы в *ROM*. Проверить правильность ввода.

5. Исполнить команду в автоматическом режиме. Убедиться в правильности результата. Снять временную диаграмму системы по образцу рис. 5. Синхронизировать осциллограф в режиме внешнего запуска от сигнала СИ.

6. Исполнить микропрограмму в шаговом режиме.

5. ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Отчет должен содержать:

I. Упрощенную блок-схему вычислительной системы лабораторного макета.

2. Мнемоническую и кодированную запись заданных микропрограмм.

3. Временную диаграмму системы.

4. В устном виде – ответы на контрольные вопросы.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

I. Типовая структура ВС и назначение ее элементов.

2. Комплексы микропроцессорных БИС - типовой состав и назначение отдельных БИС.

3. Принцип микропрограммного управления Уилкса - подробное описание.

4. Способы кодирования микрокоманд. Достоинства и недостатки этих способов.

5. Описать механизм определения адреса следующей микрокоманды.

6.Описать этапы выполнения команды в вычислительной системе.

7. Чем определяется минимальный ТАКТ СИСТЕМЫ?

8. Как технически (схемно) реализуется функция "Ключа" - I5-го разряда микрокоманды?

9. Как в микропрограмме организуется циклическое повторение микропрограммы, переход к исполнению следующей команды?

I0. Описать механизм условного и безусловного перехода в микропрограмме.

II. Как определяется быстродействие микропрограммы?

I2. В чем достоинства и недостатки применения запоминающих устройств разного типа - ПЗУ, ППЗУ, ОЗУ, в качестве управляющей памяти (памяти микропрограмм)?

I3. Вопросы, поставленные в задании на лабораторную работу (п. 4.).

Содержание

Составители: С.Г. А к и м о в, В.Г. И о ф ф е,
П.А.С о к о л о в

ИССЛЕДОВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ
НА МИКРОПРОЦЕССОРАХ

Лабораторная работа

Редактор Н.В. К а с а т к и н а
Техн. редактор Н.М. К а л е н ю к
Корректор С.С.Р у б а н

Подписано в печать 20.10.80. Формат 60x84 1/16
Бумага оберточная белая. Печать оперативная.
усл.п.л.1,2(+2 вкл.0,25 п.л.).Уч.-изд.л.1,0.Тираж 500 экз.
Заказ № 6603 Бесплатно.

Куйбышевский ордена Трудового Красного Знамени
авиационный институт имени С.П.Королева.
г. Куйбышев, ул. Молодогвардейская, 151.

Областная типография имени В.П. Мяги.
г. Куйбышев, ул. Венцека, 60.