5. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МИКРОКОМПЬЮТЕРОВ МЕТОДАМИ ПРОГРАММНОЙ ЭМУЛЯЦИИ

Система программного обеспечения представляет собой комплекс программных средств, предназначенных для повышения эффективности использования микропроцессорной техники, облегчения ее эксплуатации, т.е. снижения трудоемкости подготовительной работы, упрощения связи человека с системой при решении задач. Отличительной особенностью большей части микропроцессорных вычислительных средств является то, что они ориентированы на конкретные применения. В подавляющем большинстве случаев это специализированные устройства, предназначенные для управления в реальном масштабе времени различными объектами. При проектировании таких систем характерно стремление получить минимальную стоимость аппаратных и программных средств при ограниченном объеме памяти микро-ЭВМ.

5.I. Состав и основные функции программного обеспечения микропроцессорных систем

Программное обеспечение (ПО) современных микрокомпьютеров разделяется на три составные части: резидентное, кросс-обеспечение и прикладное [9,27]. Структура одного из вариантов ПО для микро-ЭВМ на базе МП 8080 приведена на рис.5.I.

Резидентное ПО реализуется на самой микро-ЭВМ. В состав резидентных средств ПО обычно входят простейшая операционная система (диспетчер), ассемблер, редактор и загрузчик. Диспетчер предназначен для загрузки и управления прохождением задач пользователя в различных режимах работы: разделение времени, интерактивной обработки и т.п. Программа ассемблер осуществляет трансляцию исходной программы в символической форме в двоичные коды команд микро-ЭВМ, а также поиск и индикацию ошибок.

Ассемблеры современных микро-ЭВМ, как правило, являются макроассемблерами, т.е. разрешают применение макрокоманд. Это позволяет повысить эффективность исходных программ, облегчить программирование и отладку.

Редактор программ позволяет осуществлять исправление исход-

ных программ, написанных на языке ассемблера, путем изъятия или замены отдельных символов из текстового файла.

Программа загрузчик осуществляет ввод исходной программы с перфоленты или другого носителя в ОЗУ микро-ЭВМ.

В зависимости от размеров и состава памяти и периферийных устройств, входящих в состав микропроцессорной системы, резидентное ПО может находиться на следующих носителях информации:

- перфоленте ;
- храниться в ПЗУ ;
- гибких магнитных дисках ;
- кассетных накопителях.

Ясно, что для успешного использования резидентного ПО требуется определенное количество внешних устройств (например,дисплей или электрическая пишущая машинка, перфоратор, считыватель с перфоленты и т.д.). Это не всегда осуществимо в микропроцессорных специализированных системах, обладающих ограниченной номенклатурой подключаемых устройств ввода-вывода и малым объемом памяти. В этом случае целесообразно использовать кросс-ПО.

Кросс-ПО предназначено для подготовки и отладки программ микро-ЭВМ с помощью универсальных ЭВМ, которые называют инструментальными [9,28]. В состав кросс-ПО (рис.5.I) входят:

- эмулятор ;
- кросс-ассемблер ;
- система документирования.

Эмулятор предоставляет пользователю возможность моделировать на инструментальной ЭВМ работу микро-ЭВМ и таким образом осуществлять отладку рабочей программы на языке микро-ЭВМ.

Кросс-ассемблер обеспечивает трансляцию на универсальной ЭВМ программ, составленных на языке ассемблера микро-ЭВМ, в коды команд микро-ЭВМ.

Система документирования ооуществляет подготовку необходимой информации для записи отлаженных программных модулей с помощью программатора в ПЗУ, а также при необходимости выдачи их на перфоленту для последующего ввода в ОЗУ микро-ЭВМ. Здесь же может осуществляться выдача листингов исходных программ, таблиц прошивки ПЗУ и т.д.

Прикладное ПО содержит набор программных модулей, ориентиро-

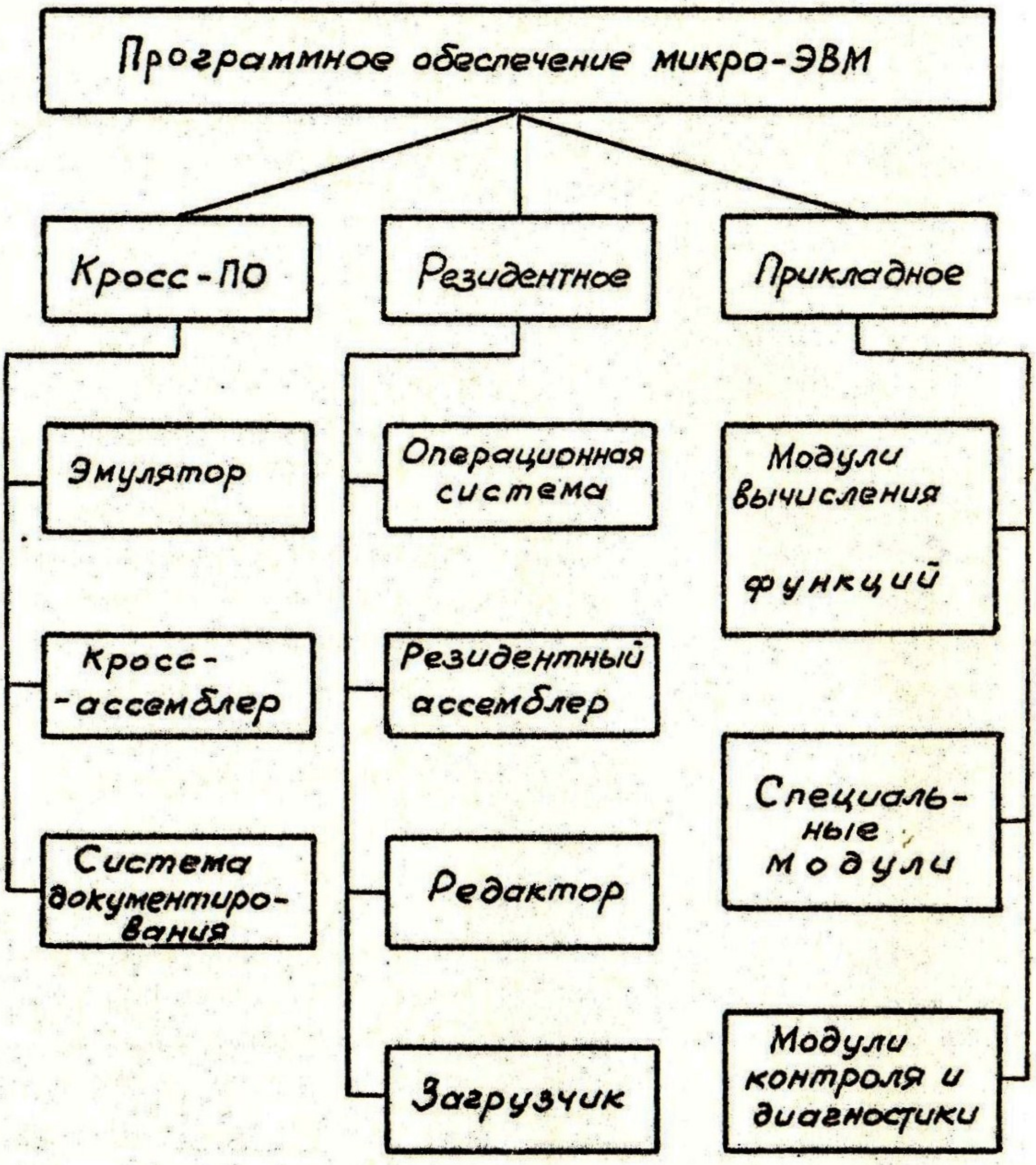


Рис. 5.1. Структура программного обеспечения микрокомпьютеров на базе МП8080

ванных на пользователей микропроцессорной системы. Сюда могут быть отнесены следующие программные средства:

I. Модули вычисления основных элементарных функций, например, умножение, деление, извлечение корня квадратного при различных форматах операндов.

2. Программные модули, обеспечивающие специализацию данного микропроцессорного вычислителя, например, для микропроцессорных регуляторов - это модули реализации П-, ПИ-, ПИД- законов регулирования.

3. Модули контроля и диагностики, предназначенные для фиксации ошибок в работе и поиска неисправностей в микрокомпьютерной системе.

В настоящее время с целью повышения эффективности и качества проектирования и отладки аппаратных средств и программного обеспечения микропроцессорных вычислителей находят широкое применение отладочные системы - комплексы автоматизации проектирования микро-ЭВМ.

Известно три подхода к созданию таких комплексов. При первом подходе проектирование микропроцессорной системы ведется на универсальной ЭВМ с помощью кросс-средств. Отладка программ осуществляется с помощью программной модели микро-ЭВМ-эмулятора. Имеется возможность подготовки необходимой документации: для прожигания ПЗУ, вывода информации на перфоноситель и т.д. Кроме того, создание или приобретение кросс-средств намного проще и дешевле, чем приобретение микро-ЭВМ с резидентным программным обеспечением. Недостатком такого подхода является трудность воспроизведения реальных временных соотношений, необходимых для полной проверки работы системы в реальном масштабе времени.

При втором подходе используются специальные отладочные комплексы, содержащие реальную микро-ЭВМ или ее прототип, реальное внешнее оборудование и, желательно, мини-ЭВМ для управления режимами отладки и расширения памяти и номенклатуры устройств ввода-вывода. В этом случае отладка программ ведется с помощью резидентного ПО, что дает возможность исследовать характеристики объектов управления в реальном масштабе времени. К недостаткам таких комплексов следует отнести большие затраты на их проектирование и изготовление, сложность обслуживания.

Наиболее часто прменяют комбинированный подход к проектированию микропроцессорной техники, когда рабочая программа отлаживается с помощью кросс-средств, а окончательная проверка и доводка производятся на отладочном комплексе. В связи с этим указанные выше два подхода следует считать взаимодополняющими, а не исключающими друг друга.

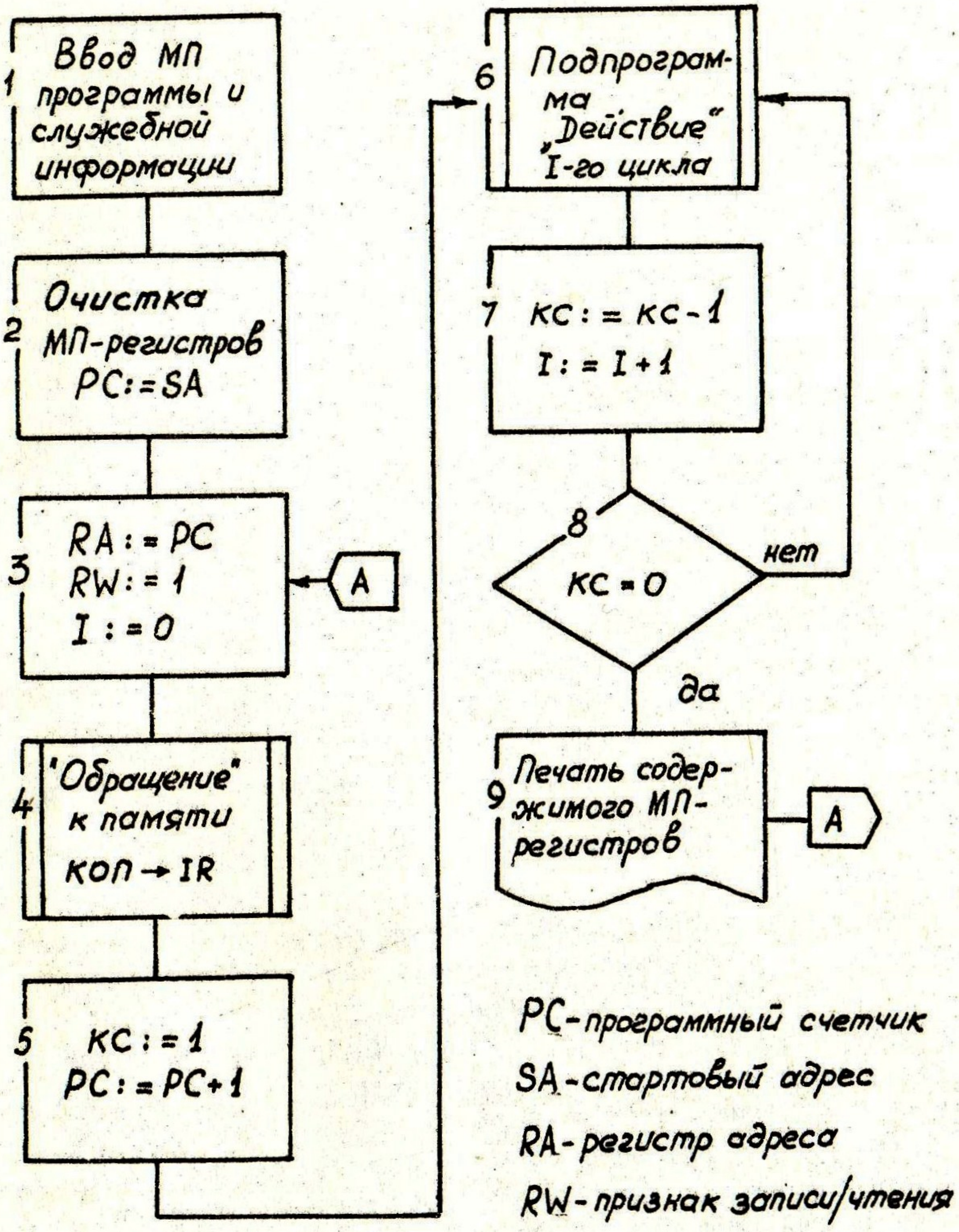
5.2. Эмулятор микропроцессора 8080

Разработанный эмулятор МП 8080 относится к кросс-обеспечению и предназначен для обучения студентов программированию микровычислителей и для обеспечения научно-технических разработок в области микропроцессорных средств ВТ.

Укрупненная блок-схема эмулятора приведена на рис.5.2. В основу алгоритма эмуляции положен принцип открытой системы, т.е. программа состоит из управляющего блока и набора подпрограмм, которые реализуют выполнение отдельных циклов команд микропроцессора. Это обеспечивает простое изменение и увеличение состава эмулируемых команд. Каждая команда МП 8080 эмулируется набором от одной до пяти подпрограмм в зависимости от количества циклов, которое требуется для ее выполнения в микропроцессоре. Формирования адреса передачи управления на подпрограмму моделирования I-го цикла команды (подпрограммы "действие") осуществляется с помощью специальных таблиц. В описываемом варианте эмулятора принято, что микропроцессор работает с постоянным запоминающим устройством ROM и оперативным запоминающим устройством RAM по 8 К байт каждое.

Входная информация для эмулятора - это микропроцессорная программа (МП-программа), написанная пользователем или полученная ассемблированием, которая сопровождается служебной информацией. МП-программа записывается в кодах микропроцессора и может содержать несколько зон. Служебная информация задает режимы моделирования отдельных участков МП-программы. Она содержит:

- начальные адреса постоянной и оперативной памяти;
- адреса участков программ и соответствующие им режимы эмуляции: нет печати, печать после каждого цикла, печать после каждой команды, печать после последней команды участка;
- стартовый адрес программы SA .



PC-программный счетчик

SA-стартовый адрес

RA-регистр адреса

RW-признак записи/чтения

IR-регистр команд

КС-количество циклов команд

Рис. 5.2. Укрупненная блок-схема эмулятора микропроцесса 8080

Листинг эмулятора в зависимости от заданной служебной информации для данного участка МП-программы содержит сведения о состоянии регистров и операндах команд. В каждом такте "работы" микропроцессора на печать может выдаваться содержимое следующих триггеров и регистров:

- A - накапливающий регистр микропроцессора (аккумулятор);
- B, C, D, E, H, L - регистры общего назначения;
- S - триггер знака;
- Z - триггер нуля аккумулятора;
- X - десятичный перенос;
- P - триггер паритета;
- C - триггер переноса;
- IR - регистр кода операции;
- PC - регистр счетчика адреса команд (программный счетчик);
- SP - указатель стека;
- Z, W - временные регистры микропроцессора;

Дополнительно на печать может выводиться:

- AOP - адрес операнда;
- OPER - значение операнда;
- RW - признак записи/чтения (I - чтение, 0 - запись);
- RA - фиктивный регистр, содержащий адрес ячейки памяти, к которой происходит обращение;
- RD - фиктивный регистр, содержащий байт, которым МП обменивается с ЗУ.

Кроме этого, выходной информацией эмулятора являются количество тактов и циклов выполнения команд, что позволяет судить о времени реализации программы микропроцессором.

Работа эмулятора начинается с ввода МП-программы и служебной информации и очистки всех ячеек, имитирующих регистры МП. Затем по заданному стартовому адресу из памяти извлекается код операции первой команды, который записывается в регистр команды IR. По выбранному коду из таблицы извлекается адрес подпрограммы, моделирующей действия МП в первом цикле выполнения команды. Происходит обращение к этой подпрограмме и ее исполнение, т.е. эмуляция работы МП в первом цикле. Если команда однобайтная, на этом ее эмуляция заканчивается (КС=0) и эмулятор извлекает сле-

дующий код операции. Если команда многобайтная, после эмуляции первого цикла вызывается подпрограмма, моделирующая извлечение второго байта и выполнение соответствующих действий, затем (если необходимо) происходит извлечение третьего байта. Максимально возможно последовательное обращение к пяти подпрограммам, что соответствует пяти циклам выполнения команды в МП 8080. Эмулятор нормально прекращает работу при извлечении команды HLT (останов). Возможно также аварийное прекращение работы эмулятора с выдачей диагностической информации. Ряд команд (ввод, вывод, загрузка вектора прерывания RST) не эмулируются.

Эмулятор записан на языке АССЕМБЛЕРА для операционной системы ОС ЕС ЭВМ.

Инструкция по использованию эмулятора приведена в приложении.

5.3. Пример эмуляции программы

Рассмотрим простейший пример подготовки и эмуляции программы. Пусть необходимо разработать программу, которая вычисляет сумму частного и остатка деления двух целых положительных шестнадцати-разрядных чисел. Если, например, задано:

делимое - $II699_{IO} = 2ДВ3_{I6} = 0010\ 1101\ 1011\ 0011_2$,

делитель - $364_{IO} = 016C_{I6} = 0000\ 0001\ 0110\ 1100_2$,

то в результате работы программы должно быть получено:

частное - $32_{IO} = 0020_{I6} = 0000\ 0000\ 0010\ 0000_2$

остаток - $5I_{IO} = 0033_{I6} = 0000\ 0000\ 0011\ 0011_2$

сумма - $83_{IO} = 0053_{I6} = 0000\ 0000\ 0101\ 0011_2$

Основной элемент программы, несомненно, операция деления. Для составления подпрограммы деления I6-разрядных чисел воспользуемся примером из работы [32]. Блок-схема подпрограммы деления двух целых положительных шестнадцатиразрядных чисел приведена на рис.5.3. Делимое располагается в паре регистров BC, делитель - в паре ДЕ. После окончания работы подпрограммы частное находится в паре BC, остаток - в паре ДЕ.

Подпрограмма работает следующим образом. После передачи ей управления выполняется преобразование содержимого пары регистров ДЕ в дополнительный код (блок I) с помощью команд межрегистрового обмена и инверсии аккумулятора. Затем (блок 2) происходит

очистка пары HL и задание количества повторения тела цикла ($17_{10}=11_{16}$) в счетчик - аккумулятор А. Блок 3 - первый блок тела цикла - содержимое пары HL записывается в стек. После этого из содержимого пары HL вычитается делитель с помощью команды суммирования пар регистров DAD D (в ДЕ - делитель в дополнительном коде). Если при выполнении операции произошел перенос (остаток больше делителя), то новый остаток заносится в стек. В блоке 7 содержимое вершины стека (старый остаток, если нет переноса, новый остаток, если есть) записывается в пару HL . Здесь же аккумулятор- счетчик запоминается в стеке. Далее в блоке 8 происходит сдвиг влево пары BC, а в блоке 9 - сдвиг влево пары HL . Сдвиги выполняются так, что в крайний правый разряд BC заносится содержимое триггера переноса, а в правый разряд HL - спадающий разряд пары BC. Сдвиг HL влево эквивалентен сдвигу делителя вправо на один разряд. В блоках 10,11,12 организуется проверка на окончание цикла. Если A≠0, то управление передается в блок 3, в противном случае происходит передача содержимого HL в ДЕ со сдвигом вправо на один разряд. Далее выполняется выход из подпрограммы.

Полный текст программы на языке ассемблера МП 8080 [32] и в кодах микропроцессора приведен на рис.5.4. Программа состоит из трех частей (зон). Для работы программы принято следующее распределение памяти микропроцессора: оперативное ЗУ - ячейки 0000_{16} - $1FFF_{16}$, постоянное ЗУ - ячейки 2000_{16} - $3FFF_{16}$. Стек расположен в ячейках ОЗУ с 0040_{16} по $03FF_{16}$ включительно.

Первая часть программы (ячейки 0000 - 0006) имитирует пуск микропроцессора со стартового адреса 0000. Команда LXI SP задает начальное содержимое указателя стека, а JMP 0400H передает управление в основную программу.

В основной программе (ячейки 0400 - 0411) сначала также восстанавливается указатель стека, затем задаются операнды программы: LXI B 2083 задает делитель, LXI D 016C - делимое. Команда CALL DIV - обращение к подпрограмме деления. Последняя расположена в области ПЗУ, начиная с ячейки 2010. После выхода из подпрограммы управление возвращается в основную программу на команду LXI H,0 - очистка пары HL . Далее двумя командами DAD B и DAD D в паре HL накапливается сумма частного и остатка. Последняя команда программы HLT - останов.

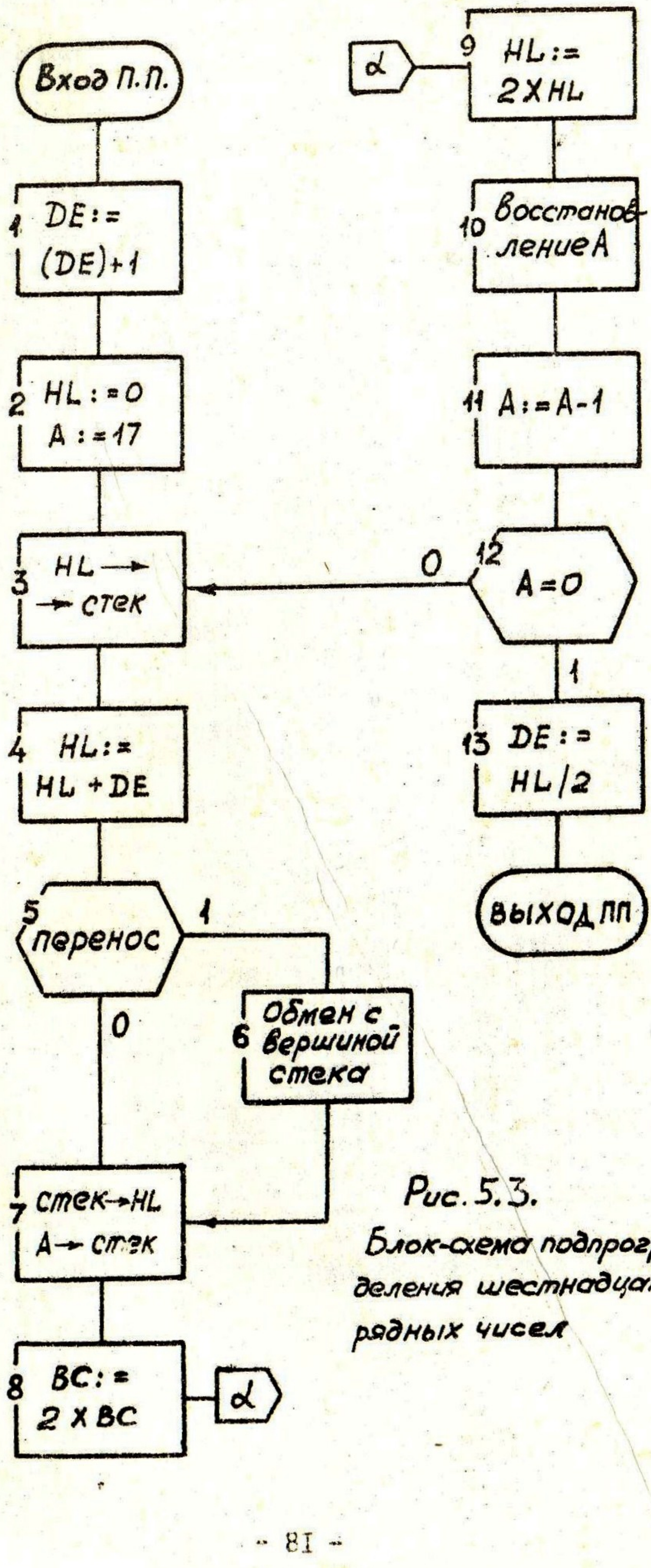


Рис. 5.3.

Блок-схема подпрограммы деления шестнадцатиразрядных чисел

Метка	Мне-мо-код	Операнд	Адрес	Код I6 с/с	Код 2с/с	Комментарий
I	2	3	4	5	6	7
	ORA	A	0000	B7	10110111	ОЧИСТКА A
	LXI	SP 0400H	0001	31	10110001	
			0002	00	00000000	
			0003	04	00000100	
	JMP	0400H	0004	C3	11000011	ПЕРЕХОД К ПРОГРАММЕ
			0005	00	00000000	
			0006	04	00000100	
; ПРОГРАММА						
	LXI	SP 0400H	0400	31	00110001	ЗАДАНИЕ УКАЗАТЕЛЯ
			0401	00	00000000	СТЕКА
			0402	04	00000100	
	LXI	B 2DB3H	0403	01	00000001	ЗАГРУЗКА
			0404	B3	10110011	
			0405	2D	00101101	
	LXI	D 016CH	0406	11	00010001	ЗАГРУЗКА
			0407	66	01101100	
			0408	01	00000001	
	CALL	DIV	0409	CD	11001101	ОВРАЩЕНИЕ К ПП
			040A	10	00010000	
			040B	20	00100000	
	LXI	H OH	040C	21	00100001	ОЧИСТКА HL
			040D	00	00000000	
			040E	00	00000000	
	DAD	B	040F	09	00001001	НАКОПЛЕНИЕ
	DAD	D	0410	19	00011001	НАКОПЛЕНИЕ
	HLT		0411	76	01110110	ОСТАНОВ
; ПОДПРОГРАММА ДЕЛЕНИЯ						
DIV	MOV	A,D	2010	7A	01111010	БЛОК 1
	CMA		2011	2F	00101111	
	MOV	D,A	2012	57	01010111	
	MOV	A,E	2013	7B	01111011	
	CMA		2014	2F	00101111	
	MOV	E,A	2015	5F	01011111	
	INX	D	2016	13	00010111	
	LXI	H,O	2017	21	00100001	БЛОК 2
			2018	00	00000000	
			2019	00	00000000	
	MVI	A,17	201A	3E	00111110	
			201B	11	00010001	
DVO	PUSH	H	201C	E5	11100101	БЛОК 3
	DAD	D	201D	19	00011001	БЛОК 4
	JNC	DV1	201E	D2	11010010	БЛОК 5
			201F	22	00100010	

Рис. 5.4. Текст программы (начало)

I	2	3	4	5	6	7
			2020	20	00100000	
	XTHL		2021	E3	11100011	БЛОК 6
	POP	H	2022	E1	11100001	БЛОК 7
	RUSH	PSW	2023	F5	11110101	
	MOV	A,C	2024	79	01111001	БЛОК 8
	RAL		2025	17	00010111	
	MOV	C,A	2026	4F	01001111	
	MOV	A,B	2027	78	01111000	
	RAL		2028	17	00010111	
	MOV	B,A	2029	47	01000111	
	MOV	A,L	202A	7D	01111101	БЛОК 9
	RAL		202B	17	00010111	
	MOV	L,A	202C	6F	01101111	
	MOV	A,H	202D	7C	01111100	
	RAL		202E	17	00010111	
	MOV	H,A	202F	67	01100111	
	POP	PSW	2030	F1	11110001	БЛОК 10
	DCR	A	2031	3D	00111101	БЛОК 11
	JNZ	DVO	2032	C2	11000010	БЛОК 12
			2033	1C	00011100	
			2034	20	00100000	
	ORA	A	2035	B7	10110111	БЛОК 13
	MOV	A,H	2036	7C	01111100	
	RAR		2037	1F	00011111	
	MOV	D,A	2038	57	01010111	
	MOV	A,L	2039	7D	01111101	
	RAR		203A	1F	00011111	
	MOV	E,A	203B	5F	01011111	
	RET		203C	C9	11001001	ВЫХОД ИЗ ПП
			203D			
			203E			
			203F			

Рис. 5.4. Текст программы (окончание)

Текст задания на эмуляцию программы приведен на рис.5.5. Начальный и конечный участки листинга эмуляции программы показаны на рис.5.6. Результаты эмуляции совпадают с рассчитанными значениями: BC - (частное) - 0020_{16}, DE (остаток) - 0033_{16}, HL (сумма) - 0053_{16}. Время эмуляции программы на ЭВМ ЕС 1022 составило 4 секунды.

```
//Y21101 JOB 'ЧАЙКА А. ГР.ВТ-13А БАШКОВ Е.А.'
// EXEC PGM=INTES
//STEPLIB DD DSN=USEDPI.LIBLINTE,DISP=OLD
//PR1 DD SYSOUT=A
//VVCARD DD *
00000000000000000000300002000
02000000100204000420022000 2040
***
00030000000040020100000000000000000000000000000000000000000000000000000000000000000000
***
B7310004C300040000000000000000000000000000000000000000000000000000000000000000000000
***
31000401B32D116C01C,Д10202100000919760000000000000000000000000000000000000000000000
***
7A2F577B2F5F132100003E11E519D22220E3E1F579174F7817477D176F7C1767F13DC21C20B77C1F
577D1F5FC900000000000000000000000000000000000000000000000000000000000000000000000
***
```

Рис. 5.5. Задание на эмуляцию подпрограммы деления

Количество циклов	Количество тактов	IR	ФЛАГИ SZXPC	B	C	Д	E	H	L	A	SP	PC	AOP	OPER
1	4	B7	01010	00	00	00	00	00	00	00	0000	0001		
4	14	31	01010	00	00	00	00	00	00	00	0400	0004	0003	04
7	24	C3	01010	00	00	00	00	00	00	00	0400	0400	0006	04
10	34	31	01010	00	00	00	00	00	00	00	0400	0403	0402	04
13	44	01	01010	2D	B3	00	00	00	00	00	0400	0406	0405	2D
16	54	11	01010	2D	B3	01	6C	00	00	00	0400	0409	0408	01
21	65	CD	01010	2D	B3	01	6C	00	00	00	03FE	2010	03FE	0C
22	70	7A	01010	2D	B3	01	6C	00	00	01	03FE	2011		
23	74	2F	01010	2D	B3	01	6C	00	00	FE	03FE	2012		
24	79	57	01010	2D	B3	FE	6C	00	00	FE	03FE	2013		
25	84	7B	01010	2D	B3	FE	6C	00	00	6C	03FE	2014		

Количество циклов	Количество тактов	IR	ФЛАГИ SZXPC	B	C	Д	E	H	L	A	SP	PC	AOP	OPER
582	2446	57	01010	00	20	00	94	00	66	00	03FE	2039		
583	2451	7D	01010	00	20	00	94	00	66	66	03FE	203A		
584	2455	1F	01010	00	20	00	94	00	66	33	03FE	203B		
585	2460	5F	01010	00	20	00	33	00	66	33	03FE	203C		
588	2470	C9	01010	00	20	00	33	00	66	33	0400	040C	03FF	04
591	2480	21	01010	00	20	00	33	00	00	33	0400	040F	040E	00
592	2490	09	01010	00	20	00	33	00	20	33	0400	0410		
593	2500	19	01010	00	20	00	33	00	53	33	0400	0411		

Рис. 5.6. Начальный и конечный участки листинга эмуляции примера

6. МИКРОПРОЦЕССОРНЫЙ РЕГУЛЯТОР ЭЛЕКТРОПРИВОДА ПОСТОЯННОГО ТОКА

Развитие микропроцессорной техники приводит к тому, что наряду с традиционными аналоговыми узлами в системах электроприводов все чаще находят применение программируемые микроконтроллеры, реализующие сложные законы оптимального и адаптивного управления. Производство новых серий больших интегральных схем позволяет проектировать цифровые и цифроаналоговые системы управления, отличающиеся повышенной точностью, быстродействием, помехозащищенностью, компактностью, надежностью, улучшенными энергетическими показателями, наглядностью отображаемой информации, низкой стоимостью.

Появление микроконтроллеров предопределяет построение сложных иерархических систем управления. Нижний уровень иерархии должны составить микропроцессорные контроллеры, обеспечивающие регулирование отдельных технологических параметров процесса. Такая организация позволяет создавать системы локального управления объектом, в которых микроконтроллер функционально связан с датчиками и исполнительными органами управляемого объекта и занят решением определенной задачи. Наличие стандартного интерфейса обеспечивает простую организацию связи с мини-ЭВМ вышестоящих уровней.

Основными недостатками традиционных систем локального управления с "жесткой" логикой следует считать: индивидуальное проектирование аппаратуры для конкретной системы, трудности внесения изменений в уже изготовленное изделие, большие сроки разработки и высокую стоимость проектирования. Этих недостатков лишены микропроцессорные программируемые контроллеры. В них алгоритм функционирования хранится в памяти. В этом случае при переходе от одной задачи к другой меняется в основном программное обеспечение. Аппаратная часть системы для решения самых разнообразных задач остается без существенных изменений.

6.I. Схемное построение регулятора

На рис.6.I приведена структурная схема микропроцессорного регулятора для систем управления электроприводами постоянного тока. На регулятор могут быть возложены некоторые из функций: задатчика интенсивности ; регулятора положения, скорости, тока, ЭДС, контроля и сигнализации ; защиты. В принципе возможно пост-

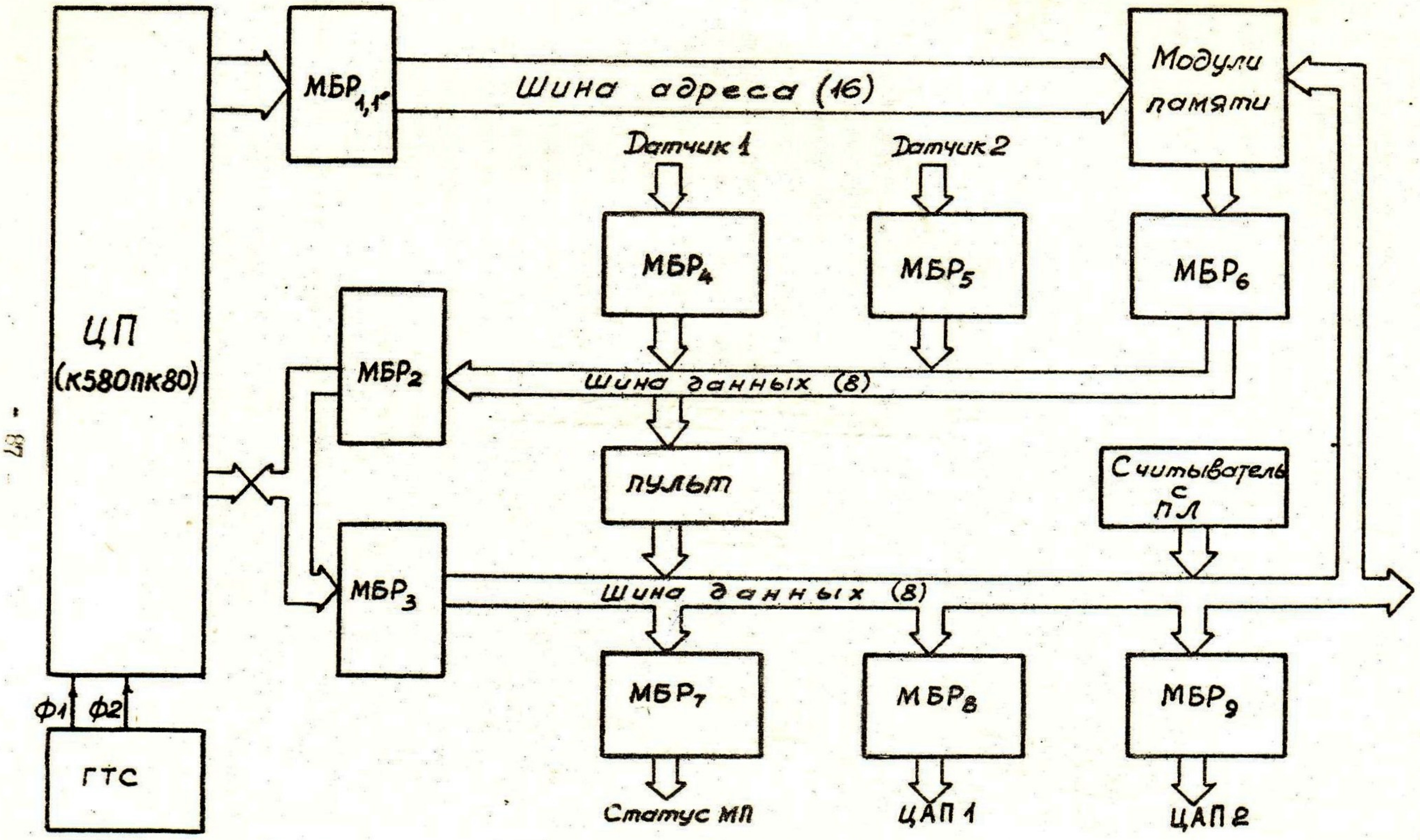


Рис. 6.1. Структура микропроцессорного регулятора

роение системы, в которой каждую задачу реализует свой регулятор.

Основным элементом микропроцессорного регулятора является центральный процессор ЦП, выполненный на БИС К580ИК80 [7]. Система команд этого микропроцессора позволяет производить широкий набор операций для решения самых разнообразных задач.

Кроме центрального процессора, в состав регулятора входят следующие блоки:

- пульт ;
- модули запоминающих устройств ;
- генератор тактовых сигналов ГТС ;
- считыватель с перфоленты ;
- набор многорежимных буферных регистров.

Пульт микропроцессорного регулятора (рис.6.2) позволяет задавать режим работы системы, вводить в ОЗУ программы, коэффициенты настройки цифровых регуляторов, производить индикацию содержимого блоков памяти.

Задание адресов и данных производится с помощью десятичных наборных переключателей. Содержимое шины данных отображается на индикаторной панели. Кнопка "сброс" управляет пуском-сбросом микро-контроллера.

Тумблер ТПИ необходим для задания автоматического или ручного режима работы системы. В ручном режиме операции записи, чтения можно выполнить с пульта при работе с модулем памяти. Для этого используется кнопка З/Ч. В автоматическом режиме работы управление берет на себя ЦП.

С помощью тумблера готовности ТГОТ можно осуществлять синхронизацию работы ЦП и модулей памяти с различным быстродействием.

Тумблер ТHOLD позволяет перевести МП в состояние HOLD , когда его шины адреса и данных переключены в третье состояние. Это используется при прямом доступе к памяти со стороны внешних устройств.

Генератор тактовых сигналов формирует две последовательности тактовых импульсов ФI и Ф2, подаваемых непосредственно в микропроцессор и сигнал ФI (ТТЛ), поступающий в регистр слова состояния микропроцессора.

Считыватель с перфоленты типа СП-3 позволяет вводить программы и данные микроконтроллера при работе с модулями оперативной памяти в режиме отладки программ.

Набор многорежимных буферных регистров (МБР) служит для организации интерфейса связи МП с модулями памяти, объектом управления, внешними устройствами. В качестве базового элемента для построения МБР принята интегральная микросхема К589ИР12 [25], представляющая собой 8-разрядный регистр на D -триггерах с выходными буферными схемами трех состояний.

В МП типа 8080 принят раздельный способ передачи адресов и данных, причем шина данных двунаправленная. В микроконтроллере шина данных с помощью МБР разделена на две подшины: входную и выходную. Так как МБР имеет выход трех состояний, все источники информации подключаются к входной шине данных непосредственно, образуя схему "проводное ИЛИ".

В зависимости от режимов работы микропроцессорного регулятора в качестве модулей памяти используются ОЗУ на микросхемах К565РУ2АА (см.4.2) или ПЗУ на базе К556РТ4 или К556РТ5 (см.4.3). Емкость модуля памяти составляет 2К 8-разрядных слов. При необходимости емкость памяти может быть расширена до 64К. Так как время обращения к модулям памяти этих типов меньше времени такта работы МП, составляющего 500 нс, не возникает необходимости формирования сигнала готовности, на вход микропроцессора READY может быть подан уровень логической I.

Для вывода управляющих сигналов при ограниченном количестве контактов в МП 8080 использован принцип временного мультиплексирования статуса и данных. Поэтому часть управляющих сигналов заносится на МБР с шины данных по сигналу СИНХРО ΛФI (ТТЛ). На выходах регистра состояний формируются сигналы "Подтверждение запроса прерывания", "Чтение", "Стек", "Подтверждение останова", "Вывод", "МI", "Ввод", "Запись/Чтение", указывающие на тип данного цикла работы МП. Вторая часть управляющих сигналов ("Синхро", "Прием", "Выдача", "Ожидание", "Подтверждение захвата", "Разрушение прерывания") поступает непосредственно с выходов МП и после умощнения используется для организации работы регулятора.

Микропроцессорный регулятор ориентирован на взаимодействие с объектом управления. Для этого имеются два канала ввода ($МБР_4$, $МБР_5$) и вывода ($МБР_7$, $МБР_8$) информации. Сбор информации о состоянии объекта производится с помощью цифровых датчиков положения или скорости. Воздействие на объект с целью изменения его состояния в соответствии с алгоритмом осуществляется путем выдачи информации

в цифроаналоговые преобразователи.

При необходимости вести обработку информации с более высокой точностью или управлять более чем по двум переменным, количество каналов связи с объектом может быть увеличено.

Управление работой всех МБР производится путем подачи на управляющие входы микросхем сигналов в соответствии с табл.6.I.

Таблица 6.I

Управляющие сигналы МБР

Номер МБР	Сигнал $\overline{R}$	$ВК_1$	$ВК_2$	ВР	С
I-I	сброс	HOLD	$\overline{HOLD}$	+5В	+5В
2	сброс	общий	DBIN	общий	+5В
3	сброс	общий.	+5В	общий	+5В
4	сброс	$A_0 \cdot DBIN$	INP*	общий	+5В
5	сброс	$\overline{A_1 \cdot DBIN}$	INP*	общий	+5В
6	сброс	общий	общий	MEMR*	WAIT
7	сброс	$\overline{\phi 1 (TTL)}$	SYNC	+5В	+5В
8	сброс	$\overline{A_0 \cdot OUT}$*	$\overline{WR}$	+5В	+5В
9	сброс.	$\overline{A_1 \cdot OUT}$	$\overline{WR}$	+5В	+5В

Звездочкой в таблице помечены сигналы, поступающие с регистра статуса $МБР_7$.

Конструктивно регулятор выполнен на двух платах размером 230х240 мм. На одной плате располагаются ЦП, ГТС и все МБР, образующие интерфейс. На другой плате располагается модуль памяти емкостью 2Кх8 разрядов.

Работу с регулятором можно подразделить на два этапа. На первом этапе ведется разработка и отладка управляющих алгоритмов, причем сложные программы целесообразно вначале отлаживать на универсальной ЭВМ методом программной эмуляции. Окончательная отладка алгоритма производится с помощью реального микропроцессорного регулятора совместно с управляемым объектом. Рабочие программы в такой системе загружаются с перфоленты в модуль оперативной памяти. Наличие ОЗУ позволяет вносить необходимые изменения в программу.

На втором этапе работы отлаженный алгоритм зашивается в модуль

ПЗУ, который заменяет ОЗУ. Замена модулей памяти не влечет за собой дополнительных изменений в аппаратной части регулятора.

6.2. Программа пропорционально-интегрального регулирования

Передаточную функцию ПИ регулятора можно представить в виде суммы передаточных функций пропорциональной и интегральной частей.

$$D_{ПИ}(z) = D_П(z) + D_И(z), \qquad (6.1)$$

причем $D_П(z) = K_П$, а $D_И(z)$ реализуется одним из методов численного интегрирования. Рассмотрим организацию интегрирования по методу Адамса-Бомфорта. Фазовая частотная характеристика программы интегрирования по этому методу совпадает с фазой идеального интегрирования и не вносит дополнительных запаздываний в систему [29].

Передаточная функция программы интегрирования в этом случае имеет вид:

$$D_И(z) = \frac{2 \cdot T \cdot z^{-1}}{1 - z^{-2}}, \qquad (6.2)$$

рекурентное соотношение для интегрирования:

$$y^*[nT] = y^*[n\overline{T-2}] + 2 \cdot T \cdot \varepsilon^*[n\overline{T-1}], \qquad (6.3)$$

где T – период квантования системы;
n – натуральное число;
ε^* – ошибка рассогласования;
y^* – выходной сигнал.

Для программной реализации (6.1) воспользуемся методом параллельного программирования (рис.6.3). Этот метод имеет существенные преимущества при отладке и изменении разработанных программ. На рис.6.3 введены следующие обозначения: $\omega_З^*$, $\omega_Д^*$ – сигналы задания и опроса датчика обратной связи, $K_И$ – коэффициент усиления интегральной составляющей

$$K_И = 2 \cdot T.$$

Непосредственно из рисунка следует

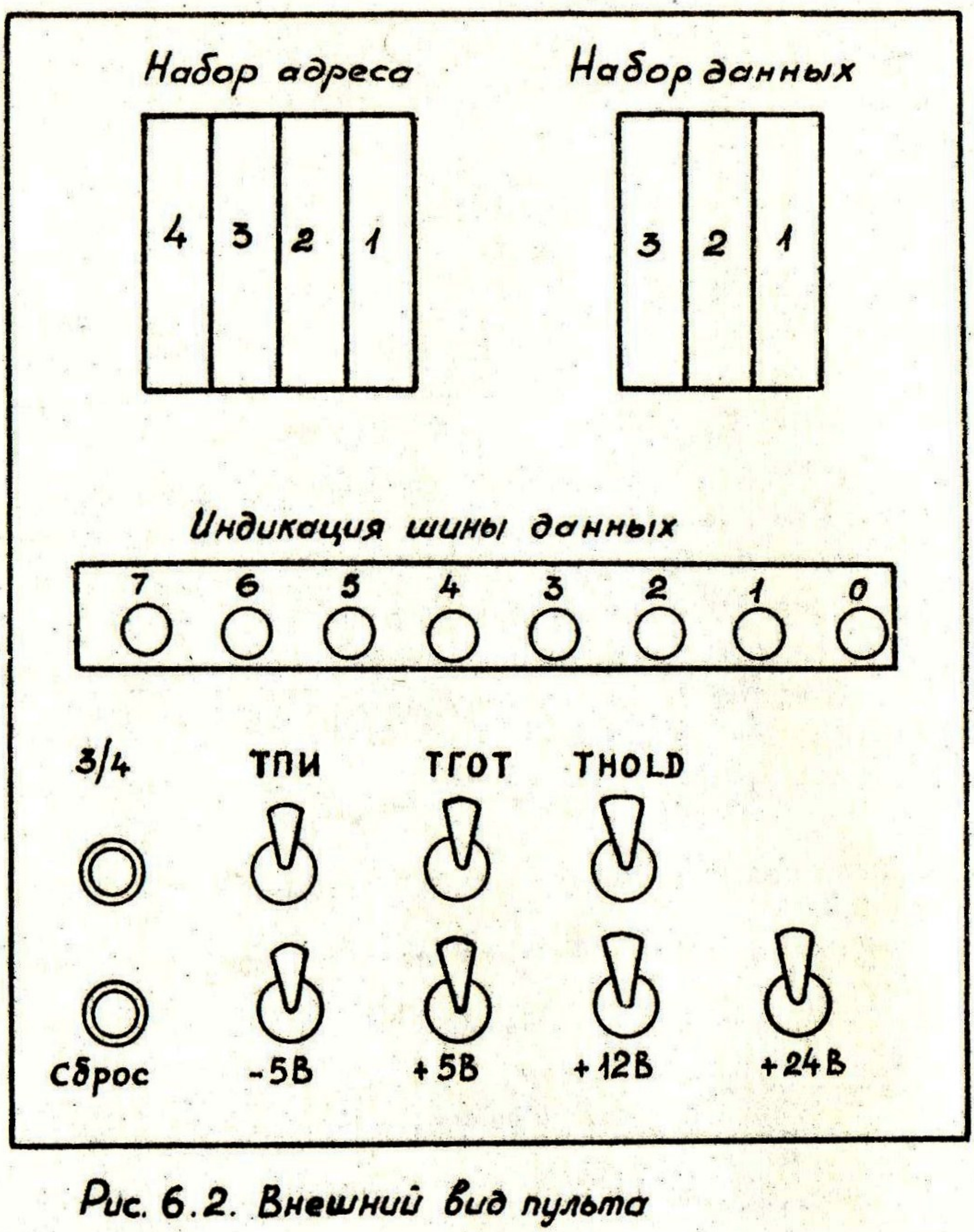


Рис. 6.2. Внешний вид пульта

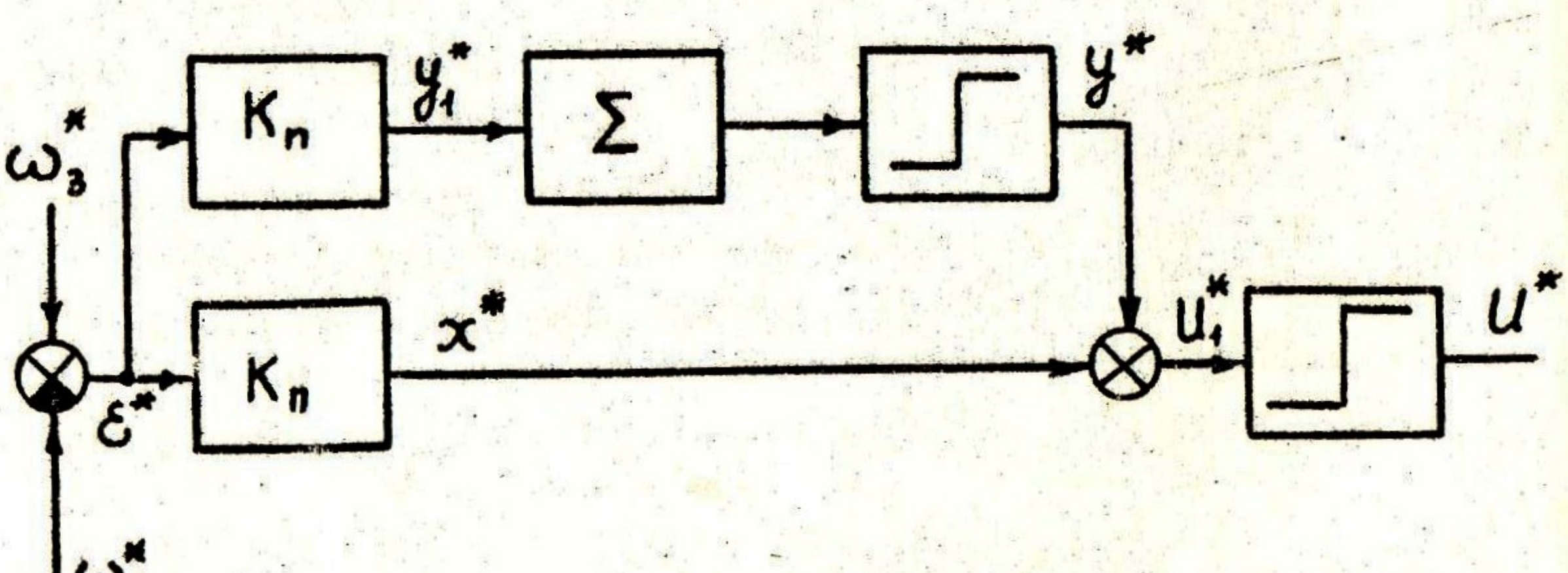


Рис. 6.3. Структурная схема микропроцессорного ПИ-регулятора

$$\varepsilon^*[иT] = \omega^*_з[иT] - \omega^*_д[иT],$$
$$U^*_1[иT] = y^*[иT] - x^*[иT],$$
$$y^*_1[иT] = K_и \cdot \varepsilon^*[и\overline{T-1}], \qquad (6.4)$$
$$y[иT] = y^*[и\overline{T-2}] + y[иT].$$

Для более точного воспроизведения процесса интегрирования программа реализована с удвоенной точностью. Это позволяет избежать потерю малых приращений в процессе интегрирования.

Особенность программной реализации ПИ регулятора заключается во введении ограничения как выходной величины регулятора U^*_1, так и выходного значения интегральной составляющей y^*. Выходной сигнал регулятора, выдаваемый в ЦАП, имеет вид:

$$U[иT] = \begin{cases} U^*_1[иT], & \text{если } |U^*_1[иT]| \le U^*_{max} \\ U^*_{max}, & \text{если } U^*_1[иT] \ge U^*_{max} \\ -U^*_{max}, & \text{если } U^*_1[иT] \le -U^*_{max} \end{cases} \qquad (6.5)$$

Упрощенная блок-схема программы ПИ-регулятора приведена на рис.6.4, текст программы – на рис.6.5.

Программа занимает 150 ячеек памяти без учета подпрограммы умножения. Принято следующее распределение памяти (адреса в восьмеричной системе счисления):

$y^*_{mM}[иT-1]$	– 0640,	$y^*_{mC}[иT-1]$	– 0641,
$y^*_{mM}[иT-2]$	– 0642,	$y^*_{mC}[иT-2]$	– 0643,
$x^*_{mM}[иT]$	– 0644,	$x^*_{mC}[иT]$	– 0645,
$y^*_{mM}[иT]$	– 0646,	$y^*_{C}[иT]$	– 0647.
$\varepsilon^*_{mM}[иT]$	– 0650,		

Индексы М и С обозначают соответственно младший и старший байты переменных. При частоте тактовых импульсов МП 2 мГц время выполнения программы составит 1,318 мс.

Начальный участок листинго эмуляции программы ПИ регулирования приведен на рис.6.6. Время эмуляции программы составило 1,5 минуты. Операции ввода вывода при эмуляции заменялись командой NOP (нет операции).

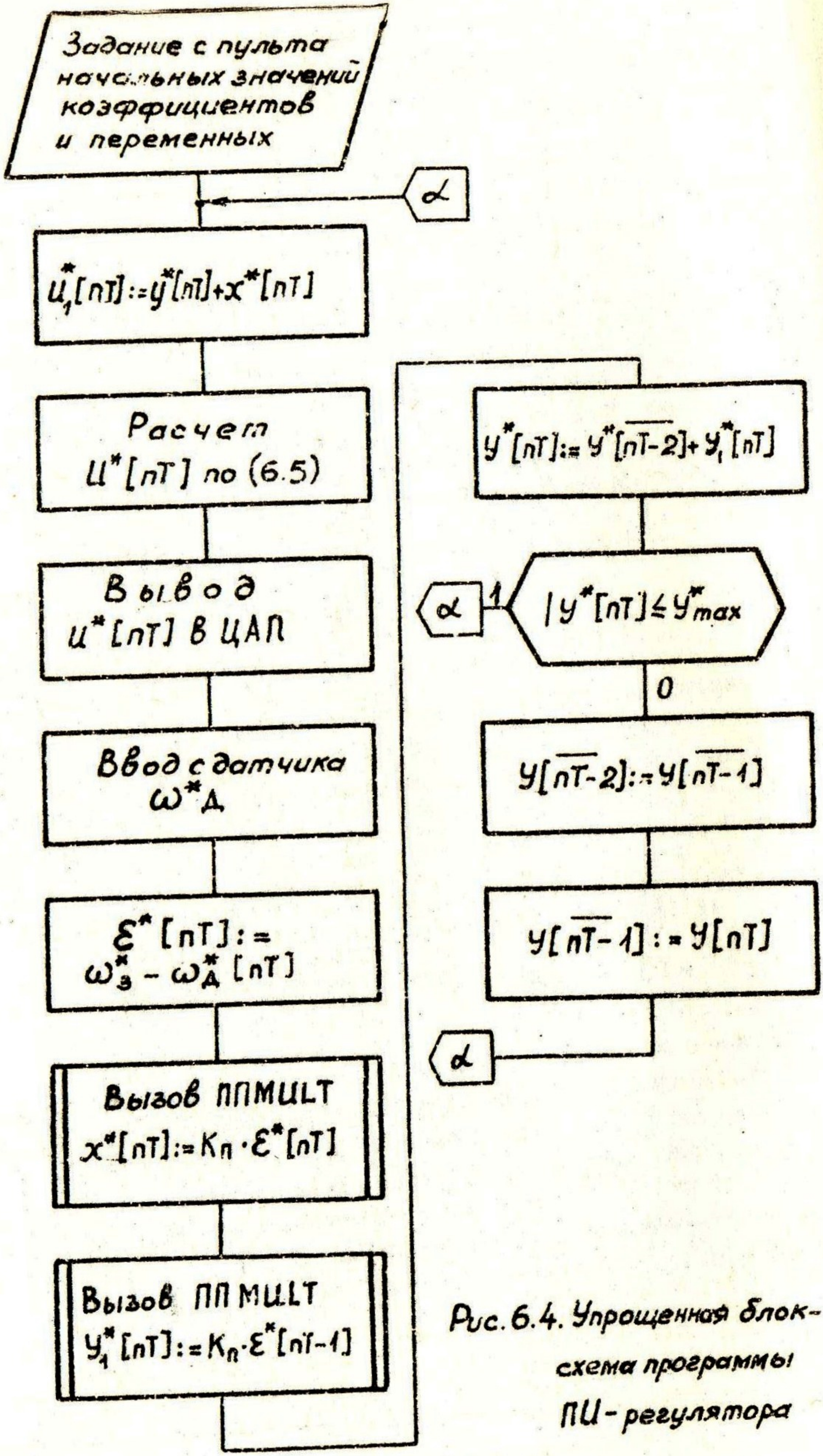


Рис. 6.4. Упрощенная блок-схема программы ПИ-регулятора

Метка	Мнемокод	Операнд	Адрес памяти	Программа в 8с.с	Программа в 2с.с.	Комментарий
I	2	3	4	5	6	7
PIAL	LXI	H 144	0400	041	00100001	
		003	0401	144	01100100	H,L:=644
			0402	003	00000011	
	MOV	M,E	0403	163	01110011	E:=
	INX	H	0404	043	00100011	ИНКР. H,L
	MOV	M,D	0405	162	01110010	D:=
	INX	H	0406	043	00100011	ИНКР. H,L
	MOV	A,M	0407	176	01111110	A:=
	ADD	E	0410	203	10000011	
	MOV	E,A	0411	137	01011111	E:=
	INX	H	0412	043	00100011	ИНКР. H,L
	MOV	A,M	0413	176	01111110	A:=
	ADC	D	0414	212	10001010	
	MOV	D,A	0415	127	01010111	D:=
	JM	M9	0416	372	11111010	УСЛОВНЫЙ ПЕРЕХОД
			0417	042	00100010	ПО ТЗ=1 (A<0)
			0420	001	00000001	
	JZ	M8	0421	312	11001010	УСЛОВНЫЙ ПЕРЕХОД
			0422	033	00011011	ПО ТО=1 (A=0)
			0423	001	00000001	
M6	MVI	A 177	0424	076	00111110	
			0425	177	01111111	A:=177
M7	OUT	001	0426	323	11010011	ВЫДАЧА A ПО 1-МУ КАНАЛУ В ЦАП
			0427	001	00000001	
	JMP	LOOP	0430	303	11000011	БП НА LOOP
			0431	067	00110111	
			0432	001	00000001	
M8	ADD	E	0433	203	10000011	A:=A+E
	JP	M7	0434	362	11110010	УСЛОВНЫЙ ПЕРЕХОД
			0435	026	00010110	
			0436	001	00000001	ПО ТЗ=0 (A>0)
	JMP	M6	0437	303	11000011	ВП НА M6
			0440	024	00010100	
			0441	001	00000001	
M9	DCX	D	0442	033	00011011	ДЕКР. (Д,Е)
	MOV	A,D	0443	172	01111010	A:=D
	ANA	A	0444	247	10100111	A:=A&A
	JZ	M10	0445	312	11001010	УСЛОВНЫЙ ПЕРЕХОД
			0446	055	00101101	ПО ТО=1(A=0)
			0447	001	00000001	
M11	MVI	A 377	0450	076	00111110	A:=377
			0451	377	11111111	
	JMP	M7	0452	303	11000011	ВП НА M7
			0453	026	00010110	
			0454	001	00000001	
M10	MOV	A,E	0455	173	01111011	A:=E
	CMA		0456	057	00101111	A:=7A
	JM	M11	0457	372	11111010	УСЛОВНЫЙ ПЕРЕХОД
			0460	050	00101000	ПО ТЗ=1 (A<0)
			0461	001	00000001	

Рис. 6.5. Текст программы ПИ-регулятора (начало)

```
 1 :  2  :  3   :  4  :  5 :    6    :                    7
        ADI   200      0462   306   11000110  A:=A+200
                       0463   200   10000000
        JMP   M7       0464   303   11000011  БП НА  M7
                       0465   026   00010110
                       0466   001   00000001
LOOP    IN    001      0467   333   11011011  ВВОД            В  А
                       0470   001   00000001
        MOV   B,A      0471   107   01000111  B:=A
        MVI   A, 101*  0472   076*  00111110  A:=
                       0473   101   01000001
        SUB   B        0474   220   10010000  A:=A-B
        STA   250,001  0475   062   00110010
                       0476   250   10101000  ЗАПИСЬ А В ПАМЯТЬ
                       0477   001   00000001
        MVI   L        0500   056*  00101110  L:=K
                       0501   010   00001000
        LXI   SP       0502   061   00110001  ЗАГРУЗКА УКАЗАТЕЛЯ
                       0503   100   01000000  СТЕКА
                       0504   002   00000010
        CALL  MULT     0505   315   11001101
                       0506   270   10111000
                       0507   001   00000001
        INX   H        0510   043   00100011  H,L:=H,L+1
        MOV   A,M      0511   176   01111110  A:=M
        MVI   L        0512   056*  00101110  L:=K
                       0513   015   00001101
        CALL  MULT     0514   315   11001101
                       0515   270   10111000
                       0516   001   00000001
        LXI   H        0517   041   00100001
                       0520   242   10100010  H,L:=0642
                       0521   001   00000001
        MOV   A,M      0522   176   01111110  A:=M
        ADD   E        0523   203   10000011  A:=A+E
        MOV   E,A      0524   137   01011111  E:=A
        INX   H        0525   043   00100011  H,L:=H,L+1
        MOV   A,M      0526   176   01111110  A:=M
        ADC   D        0527   212   10001010  A:=A+D
        MOV   D,A      0530   127   01010111  D:=A
        LDA   A        0531   072   00111010
                       0532   073   00111011  A:=
                       0533   001   00000001
        JZ    M12      0534   312   11001010  УСЛОВНЫЙ ПЕРЕХОД
                       0535   217   10001111  ПО  ТО=1
                       0536   001   00000001
        MVI   B        0537   006   00000110  B:=362
                       0540   362   11110010
        MVI   A        0541   076   00111110  A:=336
                       0542   336   11011110
M13     STA   M        0543   062   00110010  M:=A
                       0544   156   01101110
                       0545   001   00000001
```

Рис. 6.5. Текст программы ПИ регулятора (продолжение)

```
I  :2  :  3  :  4  :  5  :    6    :                    7
     STA    M        0546   062   00110010
                     0547   163   01110011   M:=A
                     0550   001   00000001
     MOV    A,B      0551   170   01111000   A:=B
     STA    M        0552   062   00110010   M:=A
                     0553   167   01110111
                     0554   001   00000001
     MOV    A,E      0555   173   01111011   A:=E
     NOP             0556   000   00000000
                     0557   377   11111111   A:=A∓377∓ТП
     INX    H        0560   043   00100011   H,L:=H,L+1
     ADD    M        0561   206   10000110   A:=A+M
     MOV    A,D      0562   172   01111010   A:=D
     NOP             0553   000   00000000
                     0564   177   01111111   A:=A∓177∓ТП
     INX    H        0565   043   00100011   H,L:=H,L+1
     ADD    M        0566   206   10000110   A:=A+M
     NOP             0567   000   00000000   УСЛОВНЫЙ ПЕРЕХОД
                     0570   000   00000000   ПО  ТЗ
                     0571   001   00000001
     INX    H        0572   043   00100011   H,L:=H,L+1
     MOV    C,M      0573   116   01001110   C:=M
     MOV    M,E      0574   163   01110011   M    :=E
     INX    H        0575   043   00100011   H,L:=H,L+1
     MOV    B,M      0576   106   01000110   B:=M
     MOV    M,D      0577   162   01110010   M    :=D
     LXI    H        0600   041   00100001
                     0601   240   10100000   H,L:=640
                     0602   001   00000001
     MOV    E,M      0603   136   01011110   E:=M
     MOV    M,C      0604   161   01110001   M    :=C
     INX    H        0605   043   00100011   H,L+1
     MOV    D,M      0606   126   01010110   D:=M
     MOV    M,B      0607   160   01110000   M    :=B
     INX    H        0610   043   00100011   H,L:=H,L+1
     MOV    M,E      0611   163   01110011   M    :=E
     INX    H        0612   043   00100011   H,L:=H,L+1
     MOV    M,D      0613   162   01110010   M    :=D
     JMP    PIAL     0614   303   11000011   ВП  НА  PIAL
                     0615   000   00000000
                     0616   001   00000001
M12  MVI    B        0617   006   00000110   B:=372
                     0620   372   11111010
     MVI    A        0621   076   00111110   A:=316
                     0622   316   11001110
     JMP    M13      0623   303   11000011
                     0624   143   01100011   БП  НА  M13
                     0625   001   00000001
```

Рис. 6.5. Текст программы ПИ регулятора (окончание).

```
//M06301 JOB 'ГУБАРЬ Ю.В.КАФ.ЭВМ'
// EXEC PGM=INTES
//STEPLIB DD DSN=USEDPI.LIBLINTE,DISP=OLD
//PR1 DD SYSOUT=A
//VVCARD DD *
00000000000000000000200000000
02010001950001B801F50000000000
***
00020100010001B800000000000000000000000000000000000000000000000000000000000000
***
216403732372237E835F237E8A57FA2201CA1B013E7F0000C337018 3F21601C314011B7AA7CA2D01
3EFFC316017B2FFA2801C680C316010000473F419032A8012E08 314002CDB8012 37E2E0DCDB80121
A2017F835F237E8A573A3B01CA8F0106F23EDE326E01327 30178 3277017B00FF23867A007F238600
0001234E732346722 1A0015E712356702373 2372C3000106FA3ECEC36 3010000000000000000000000
***
0E07110000A7F2F1013D2F0680677D07A76F7BF2D401845F7ACE00577B175F7A17570DC2C6017A1F
577B1F5F78A7F2F0017A2F577B2F5F17C90600C3C501000000000000000000000000000000000000
***
```

Количество циклов	Количество тактов	IR	ФЛАГИ SZXPC	B	C	Д	E	H	L	A	SP	PC	AOP	OPER
3	10	21	00000	00	00	00	00	03	64	00	0000	0103	0102	03
5	17	73	00000	00	00	00	00	03	64	00	0000	0104	0364	00
6	22	23	00000	00	00	00	00	03	65	00	0000	0105		
8	29	72	00000	00	00	00	00	03	65	00	0000	0106	0365	00
9	34	23	00000	00	00	00	00	03	66	00	0000	0107		
11	41	7E	00000	00	00	00	00	03	66	00	0000	0108	0366	00
12	45	83	01010	00	00	00	00	03	66	00	0000	0109		
13	50	5F	01010	00	00	00	00	03	66	00	0000	010A		
14	55	23	01010	00	00	00	00	03	67	00	0000	010B		
16	62	7E	01010	00	00	00	00	03	67	84	0000	010C	0367	84
17	66	8A	10010	00	00	00	00	03	67	84	0000	010D		
18	71	57	10010	00	00	84	00	03	67	84	0000	010E		
21	82	FA	10010	00	00	84	00	03	67	84	0000	0122	0110	01
22	87	1B	10010	00	00	83	FF	03	67	84	0000	0123		

Рис. 6.6. Начальный участок листинга эмуляции программы регулятора