Микропроцессоры серии К1814

Интегральные микросхемы серии К1814, выполненные по *p*-МДП-технологии, представляют собой однокристальные 4-разрядные микро-ЭВМ, предназначены для построения микропроцессорных систем управления. В серию входят:

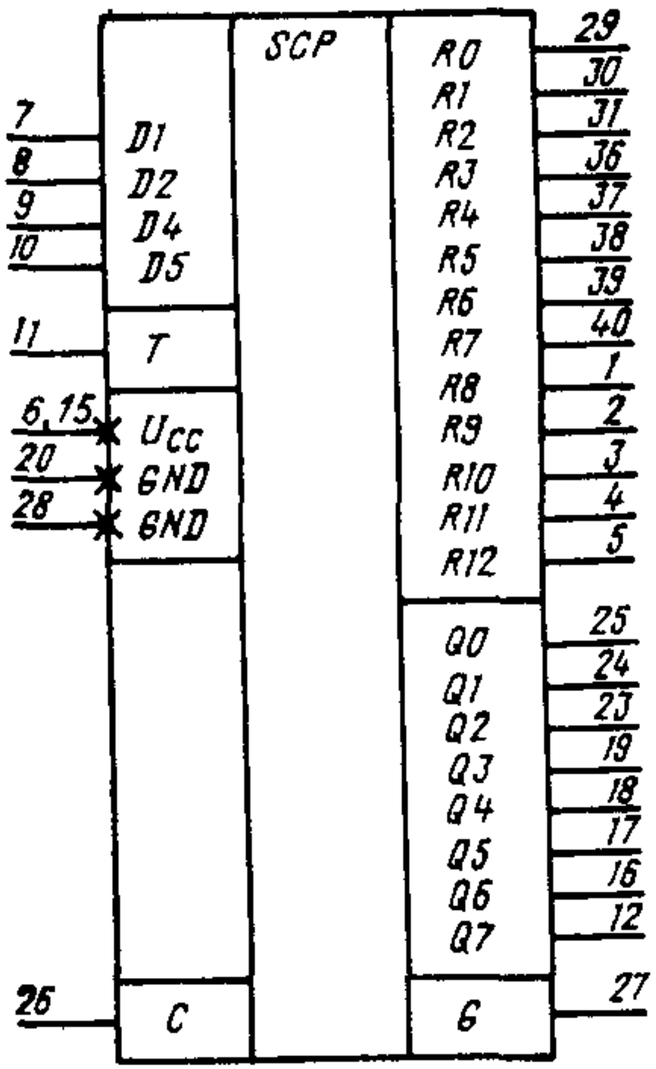

Рис. 1. Условное графическое обозначение однокристальной микроЭВМ

Таблица 1

Параметр	Обозначение	Значения параметров	
		мин.	макс.
Напряжение питания, В	U_{CC}	—9,9	—8,1
Входное напряжение низкого уровня, В	U_{IL}	—	—4
Входное напряжение высокого уровня, В	U_{IH}	—0,75	—
Потребляемая мощность, мВт	P_{CC}	—	70
Тактовая частота, кГц	—	100	350
Рабочий диапазон температур, °С	T	—10	+70

Таблица 2

Вывод	Обозначение	Тип вывода	Функциональное назначение выводов
1—5, 29—31, 36—40	*R8—R12, R0—R2, R3—R7*	Выход	Порт *R*
7—10	*D1, D2, D4, D5*	Вход	Порт *D*
12, 16—19, 23—25	*Q7—Q0*	Выход	Порт *Q*
11	*T*	Вход	Сброс, тест
27	*G*	Выход	Тактовый сигнал
26	*C*	Вход	Тактовый сигнал
6, 15	*Ucc*	—	Напряжение питания
20, 28	*GND*	—	Общий

Примечание Выводы *13—15 21, 22, 32—35* микросхемы не задействованы

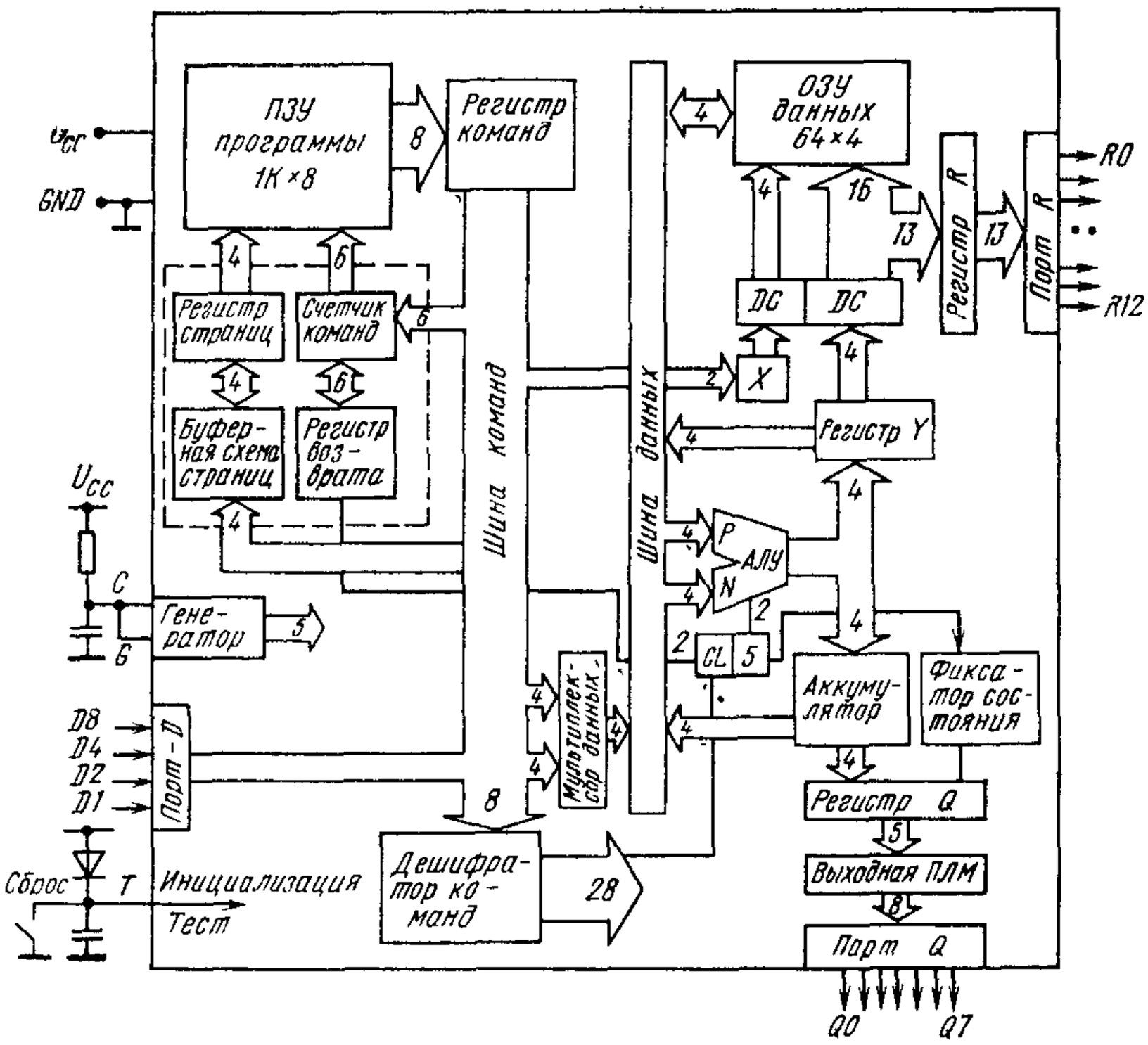

Рис .3 Архитектура однокристальной микро-ЭВМ

однокристальная микро-ЭВМ (ОМ ЭВМ), которая содержит 4-разрядный процессор, ПЗУ программы емкостью 1К 8-разрядных команд, ОЗУ данных емкостью 64 4-разрядных слова, схемы ввода/вывода и встроенный тактовый генератор (КР1814ВЕ4);

Однокристальные микро-ЭВМ выполняют функции специализированных микроконтроллеров в соответствии с программой, хранящейся в масочно-программируемом ПЗУ, и могут быть применены в периферийных устройствах вычислительной техники, различной контрольно-измерительной, медицинской и бытовой аппаратуре.

Условные графические обозначения микросхем ОМ ЭВМ приведены на рис. 1, функциональное назначение выводов микросхем ОМ ЭВМ показано в табл. 2.

Электрические параметры и режимы эксплуатации микросхем серии К1814 приведены в табл. 1.

Структурная схема ОМ ЭВМ приведены на рис. 3, Микро-ЭВМ построена с разделением шин команд и данных, что позволяет использовать данные и команды независимой разрядности, а также реализовать конвейерный принцип выполнения операции. Четырехразрядная организация достаточна для выполнения большинства контроллерных функций и особенно удобна при использовании двоично-десятичной арифметики.

Разделение шин команд и данных обусловливает аппаратное и программное разделение адресации памяти и ОЗУ данных.

Адресное поле 1К разбито на 16 страниц по 64 байта. Адрес страницы задается 4-разрядным регистром адреса страницы (РА), а 6-разрядный счетчик команд (РС) адресует 64 команды внутри каждой страницы. РС не является последовательным счетчиком. Счет осуществляется по псевдослучайному закону, приведенному в табл. 4. Изменение порядка выборки команд программы осуществляется командами условного перехода *BR*, возврата из подпрограммы *RETN* По командам *BR* и *CALL* осуществляется занесение РС 6-разрядного адрсса перехода из адресного поля *W* этих команд. Одновременно в РА загружается 4-разрядный адрес страницы перехода из буферного регистра адреса страницы РВ.

Для реализации длинных ветвлений в РВ перед командами *BR* или *CALL* загружают страницы перехода из поля команды *LDP* Если содержимое РА и РВ одинаково, то происходит короткое ветвление программы (переход на той же странице), если в РА из РВ загружается содержимое новой страницы, то

Таблица 4

Номер команды	Восьмеричное значение РС	Номер комаиды	Восьмеричное значение РС	Номер команды	Восьмеричное значение РС	Номер команды	Восьмеричное значение РС
0	00	16	47	32	34	48	42
1	01	17	16	33	70	49	04
2	03	18	35	34	61	50	11
3	07	19	72	35	43	51	23
4	17	20	65	36	06	52	46
5	37	21	53	37	15	53	14
6	77	22	26	38	33	54	31
7	76	23	54	39	66	55	62
8	75	24	30	40	55	56	45
9	73	25	60	41	32	57	12
10	67	26	41	42	64	58	25
11	57	27	02	43	51	59	52
12	36	28	05	44	22	60	24
13	74	29	13	45	44	61	50
14	71	30	27	46	10	62	20
15	63	31	56	47	21	63	40

происходит длинное ветвление программы с переходом на другую страницу памяти. В случае команды *CALL* в регистре возврата из подпрограммы *SR* запоминается адрес возврата в основную программу, а в РВ запоминается адрес страницы осиовной программы.

По команде *RETN* происходит возврат в исходную программу; при этом содержимое РВ пересылается в РА, а содержимое *SR* — в РС. Команды *BR* и *CALL* условны по состоянию триггера состояния *S* и выполняются, если $S=1$. В противном случае ветвления не происходит и в следующем цикле РС адресует следующую команду программы. Такая структура адресации памяти программ проста в программировании, ио не позволяет осуществлять вложение подпрограмм и длиные ветвления в подпрограмме.

Структура адресации ОЗУ данных также страничного типа. ОЗУ емкостью 64×4 бит разбито на четыре страницы по 16 4-разрядных слова. Адрес страницы задается 2-разрядным регистром *X*, адрес слова внутри каждой страницы задается регистром *Y*. В отличие от регистра *X* регистр *Y* участвует в операциях АЛУ и является регистром общего назначения (РОН). Изменение содержимого регистра *X* осуществляется по командам «Загрузить регистр *X*» *(LDX)* и «Образовать дополнение содержимого регистра *X*» *(COMX)*

Четырехразрядное АЛУ выполняет операции сложения, вычитания, арифметического и логического сравнения над операциями, поступающими с шины данных через входные мультиплексоры. Результат арифметической операции через выходной селектор в соответствии с микрокомандами отсылается в один из РОН, аккумулятор или регистр *Y*. Одновременно с арифметическими операцнями АЛУ производит сравиение входных величин и выработку признака равенства или неравенства операндов. Результат сравнения *(NE)* или содержимое разряда переноса *(CR)* соответствующими микрокомаидами могут пересылаться в триггер состояния.

Входными операндами АЛУ могут быть выходные коды регистра *Y*, ОЗУ, мультиплексора даниых СДВ, прямой или иверсиый код аккумулятора, константа 15.

Мультиплексор данных СДВ представляет собой логическую схему, предназначенную для выбора и передачи на шину данных входной иформации из порта *D*, коистанты *C* с шины команд или битовой маски для маскирования разрядов при выполнении операций иад битами.

Пересылки данных в РОН осуществляются через АЛУ путем сложения их с нулями.

Базовая система команд (табл. 5) содержит 43 комаиды. Выполняемые операции: пересылка, арифметические, арифметическое и логическое сравнение, поразрядиая обработка слов памяти, загрузка коистант, ввода/вывода, адресации. Все команды однобайтовые и имеют один из четырех форматов согласно табл. 6. Декодирование кодов команд и формироваиие сигналов управления осуществляются дешифратором команд. На выходе дешифратора формируется горизонтальный микрокод, разряды которого инициируют выполнение микрооперации. Часть дешифратора выполиена на базе ПЛМ мощиостью в 30 произведений, причем матрица «И» ПЛМ осуществляет непосредствеиное декодироваиие кода команды, а матрица «ИЛИ» выполияет функции памяти микропрограмм. Управляющие сигналы, возбуждаемые иа выходе ПЛМ, инициируют выполнение комбинаций из 16 программируемых микроопераций. Для обеспечения возможиости формирования сложных команд типа «Чтение—операция—запись» выбрана оргаиизация выполнения микроопераций, обычная для горизонтального микропрограммирования.

Для некоторых операций, связанных с адресацией, вводом и модификацией разрядов, достаточно произвести одну микрооперацию, причем эти микрооперации не используются в совокупности с другими микрооперациями в одном командном цикле. Поэтому с целью экономии микропрограммной ПЛМ формирование сигналов управления для данных операций осуществляется непосредственно на выходе дешифратора (матрица «И» ПЛМ).

Слово состояния программы определяется содержанием триггера признака подпрограммы *CL* и триггера состояния *S*. Триггер признака подпрограммы устанавливается командой *CALL* и определяет занесение адреса возврата в регистр возврата и буфер страницы и блокировку длинных переходов в режиме вызова. Таким образом, длина подпрограммы не может превышать 64 команды. Сброс триггера производится при возвращении в основную программу по команде *RETN*.

Таблица 5

Тип команд	Мнемоника команд	Код операции (двоичный)	Состояние PC	Описание команды
Пересылки	*TAY*	00100100	1	Переслать содержимое аккумулятора в регистр *Y*
	TYA	00100011	1	Переслать содержимое регистра *Y* в аккумулятор
	TAM	00000011	1	Переслать содержимое аккумулятора в память
	TAMIY	00100000	1	Переслать содержимое аккумулятора, инкремент регистра *Y*
	TAMZA	00000100	1	Переслать содержимое аккумулятора в память, очистить аккумулятор
	TMY	00100010	1	Переслать содержимое памяти в регистр *Y*
	TMA	00100001	1	Переслать содержимое памяти в аккумулятор
	XMA	00101110	1	Поменять содержимое памяти и аккумулятора
	CLA	00101111	1	Очистить аккумулятор
Засылки констант	*TCY*	0010(C)	1	Загрузить констаиту в регистр *Y*
	TCMIY	0110(C)	1	Загрузить констаиту в память, увеличить регистр *Y*
Арифметические операции	*AMMAC*	00100101	*CR*	Сложить содержимое аккумулятора и памяти, результат переслать в аккумулятор
	SAMAN	00100111	*CR*	Вычесть содержимое аккумулятора из памяти, результат переслать в аккумулятор
	IMAC	00101000	*CR*	Инкремент содержимого памяти и загрузить в аккумулятор
	DMAN	00101010	*CR*	Декремент содержимого памяти и загрузить в аккумулятор
	IA	00001110	1	Инкремент содержимого аккумулятора
	DAN	00000111	*CR*	Декремеит содержимого аккумулятора
	IYC	00101100	*CR*	Инкремент содержимого регистра *Y*
	DYN	00101100	*CR*	Декремеит содержимого регистра *Y*
	A6AAC	00000110	*CR*	Увеличить содержимое аккумулятора иа 6
	A8AAC	00000001	*CR*	Увеличить содержимое аккумулятора на 8
	A10AAC	00000101	*CR*	Увеличить содержимое аккумулятора на 10
	CPAIZ	00101101	*CR*	Образовать дополнеиие аккумулятора до 2, устаиовить триггер состояния
Сравнения	*ALEM*	00101001	*CR*	Если аккумулятор меньше или равен памяти, то установить триггер состояния
	ALEC	0111(C)	*CR*	Если аккумулятор меньше или равен константе, то установить триггер состояиия
	MNEZ	C0100110	*NE*	Если содержимое памяти не равно 0, то установить триггер состояния
	YNEA	00000010	*NE*	Если содержимое регистра *Y* ие равно аккумулятору, то установить триггер состояния, бит состояния переслать в фиксатор состояния
	YNEC	0101(C)	*NE*	Если содержимое регистра *Y* не равно константе, то установить триггер состояния
Побитовой обработки памяти	*SBIT*	001100B	1	Установить бит памяти
	RBIT	001101B	1	Сбросить бит памяти
	TBITI	001110B	*NE*	Проверить бит памяти, и если 1, то устаиовить триггер состояния
Адресации страниц ОЗУ	*LDX*	001111B	1	Загрузить константу в регистр *X*
	COMX	00000000	1	Иивертировать содержимое регистра *X*
	LDP	0001(C)	1	Загрузить коистанту в буфер адреса страницы
	BR	10(W)	1	Переход, условно по состояиию
	CALL	11(W)	1	Обращение к подпрограмме, условно по состоянию

Окоичаиие табл. 5

Тип команд	Мнемоника команд	Код операции (двоичный)	Состояние PC	Описаиие команды
Ввод/вывод	*RETN*	00001111	1	Возврат из подпрограммы
	DNEZ	00001001	*NE*	Если входные данные не равиы 0, то установить триггер состояния
	TDA	00001000	1	Переслать входные данные в аккумулятор
	SETR	00001101	1	Устаиовить разряд регистра *R*
	TIQ	00001010	1	Переслать содержание аккумулятора и фиксатора состояния в регистр *Q*
	CLQ	00001011	1	Очистить регистр *Q*
	PSTR	00001100	1	Сбросить разряды регистра *R*

Примечаиие. *CR* — перенос в результате арифметической операции, перенос-заем; *NE* — результат компарацин (*NE*=1, если операнды ие равны); (C) — код константы; (W) — 6-разрядное поле адреса перехода.

Триггер состояния определяет разрешение передач управления. Содержимое триггера определяется командой, выполненной в предыдущем цикле. В зависимости от комаиды (см. табл. 5) триггер состояния принимает безусловное зиачеиие 1, значение разряда переиоса сумматора, либо результата сравиения операндов компаратором. Содержимое триггера состояния через выходиой фиксатор состояния *SL* может быть выведено в выходной регистр *Q*. Фиксатор состояния может быть модифицироваи только командой *YNEA*.

Микро-ЭВМ имеет раздельиые схемы ввода и вывода, позволяющие параллельно вводить входные данные и выводить содержимое двух выходных регистров *Q* и *R* на раздельные порты вывода.

Входные данные вводятся в схему через 4-разрядный входной порт *D* и далее через мультиплексор данных поступают на шину данных. Входная информация из порта *D* может пересылаться в аккумулятор по комаиде *TDA* или в АЛУ для проверки иа нуль по команде *DNEZ*. Выходной регистр *Q* используется для параллельиого вывода даиных. Запись в регистр *Q* производится по команде вывода *TIQ*; при этом содержимое аккумулятора и фиксатора состояния *SL* пересылается в регистр *Q*.

Для удобства представления выходной информации порта *Q*, например на 7-сегментном индикаторе или другом устройстве отображения ииформации, в схеме имеется выходной шифратор информации, выполиенный на базе ПЛМ мощностью 20 произведений. Матрица «ИЛИ» ПЛМ имеет восемь выходных лииий в соответствии с разрядностью порта *Q*. Программирование выходиой ПЛМ производится маской иа стадии изготовления в соответствии с требованиями пользователя. В универсальиой микро-ЭВМ выходная ПЛМ кодируется как повторитель.

Выходной регистр *R* имеет 13 индивидуально программируемых ячеек памяти. В одном командиом цикле может быть устаиовлеиа (командой *SETR*) или сброшена (командой *RSTR*) только одна ячейка регистра *R*, адресуемая текущим содержимым регистра *Y*.

Все ячейки регистра *R* имеют параллельвый выход на выводы порта *R*. Индивидуальное программирование выходиых линий порта *R* позволяет эффективио использовать их для непосредственного управления исполнительными устройствами, стробироваиия входиых и выходных данных, сканирования клавиатуры, дисплея и других применений.

Разделение шин комаид и данных позволяет организовать двухуровневый конвейер операций, совместив цикл выборки и исполнительный цикл соседних команд. При такой организации время выполнения операций определяется длительиостью машинного цикла микро-ЭВМ, а не полиого цикла команды. Длительиость машиного цикла составляет 6 тактов геиератора или 20 мкс на частоте 300 кГц и одинакова для всех команд, что очень удобно при программировании задач реального времени. Синхроиизация схемы в пределах цикла

Таблица 6

Формат команды	Разряд кода комаиды							
	K8	*K7*	*K6*	*K5*	*K4*	*K3*	*K2*	*K1*
I	КОП		MSB		(W)			LSB
II	КОП				LSB	(C)		MSB
III	КОП					LSB	(B)	MSB
IV	КОП							

Примечаиие. *K8—K1* — разряды кода команды, (W) — 6-разрядное поле адреса перехода; (C)—4-разрядиое поле коистанты или адреса страницы программной памяти; (B) — 2-разрядиое поле бита ячейкн ОЗУ.

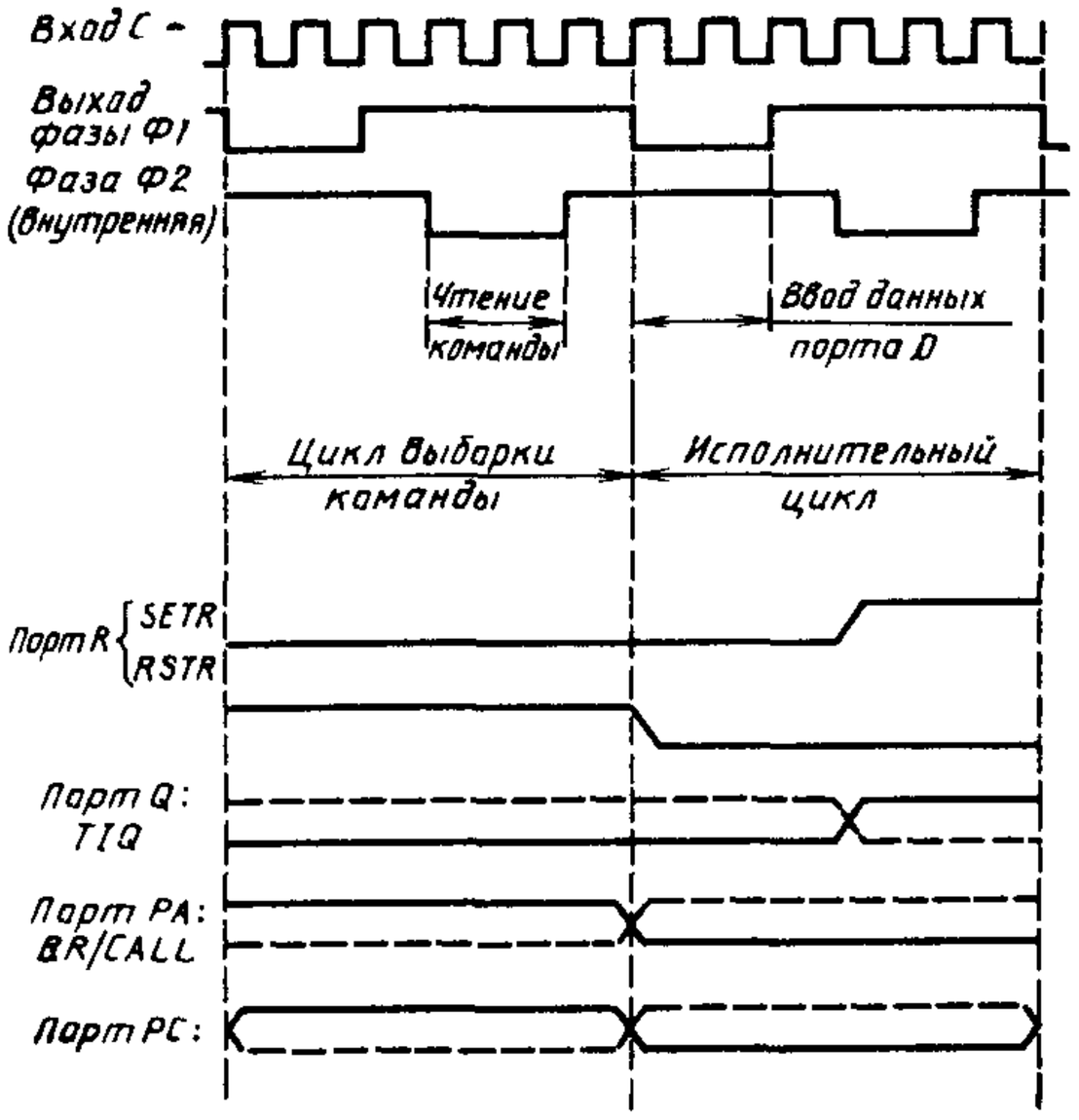

Рис. 4. Временные диаграммы работы микро-ЭВМ серии К1814

осуществляется пятью фазами, часть которых приведена на рис. 4. Здесь же указаны временные интервалы выполнения различных микроопераций, для команд вывода приведены временные диаграммы.

При включении питания специальной схемой производится начальная установка микросхемы в исходное состояние: $PA = PB = 15$, $PC = 0$, $Q = 0$, $R = 0$, $CL = 0$. Поддержка внутренней схемы инициализации при работе с источниками питания, не обеспечивающими крутой фронт нарастания напряжения в микросхеме, обеспечивается по выводу T путем создания задержки установки уровня нуля на данном выводе внешней емкостью. Разряд емкости при отключении питания производится внешним диодом. Сброс микро-ЭВМ в начальное состояние осуществляется путем установки на выводе T напряжения высокого уровня на время не менее шести машинных циклов при нулевой информации на входах порта D. Состояние схемы после сброса аналогично состоянию после начальной установки.

Микро-ЭВМ может работать с внешней и внутренней синхронизацией. В первом случае тактовые импульсы подаются на вход C с внешнего генератора, во втором при подключении RC-цепи к замкнутым выводам C и G внутренний релаксационный генератор формирует тактовые импульсы с частотой, определяемой параметрами RC-цепи. Типовое значение частоты составляет 300 кГц.

На рис. 5 приведена схема включения однокристальной микро-ЭВМ для реализации простого секундомера. При нажатии на кнопку *1* начинается отсчет времени с отображением на 4-разрядном индикаторе десятых долей секунды, секунд, десятков секунд и минут. При нажатии на кнопки *2* или *3* происходит остановка счета времени или сброс секундомера соответственно. Для отсчета времени на вход *D8* микро-ЭВМ подаются импульсы частотой 50 кГц.

Разряды порта R *R0—R3* используются для сканирования клавиатуры и индикатора, причем сигнал высокого уровня на выходах R соответствует разрешению свечения соответствующего разряда индикатора. Предполагается, что выходная ПЛМ закодирована для преобразования содержимого регистра Q в коды семисегментного индикатора так, что нулевому значению регистра Q соответствует 0 на инди-

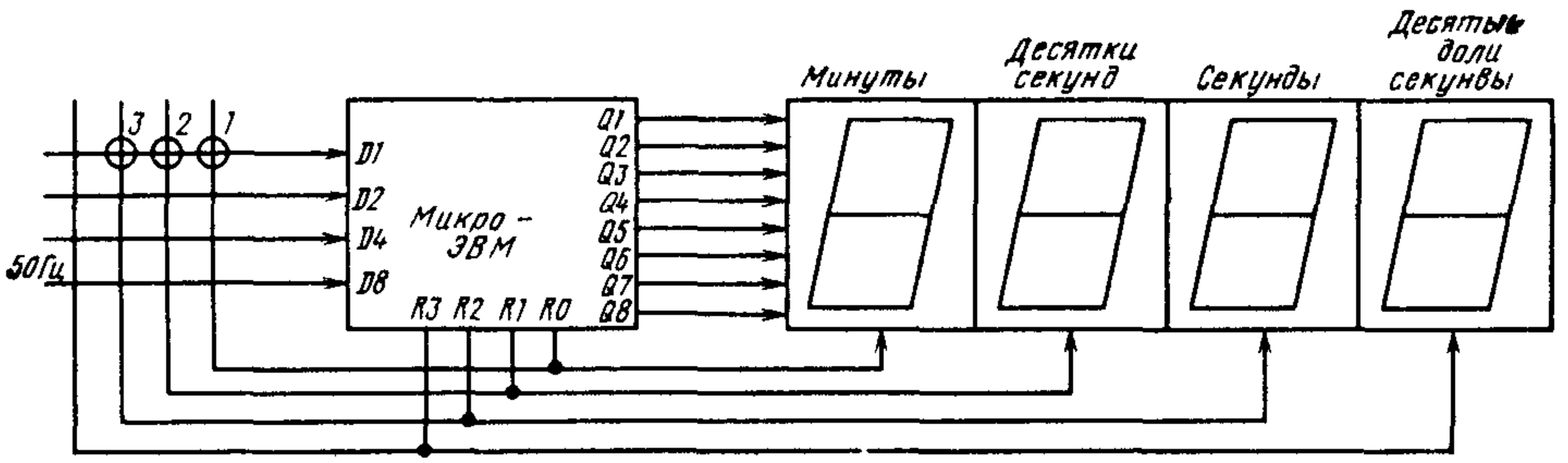

Рис. 5. Схема включения микро-ЭВМ при реализации секундомера:
1 — кнопка «Пуск»; *2* — кнопка «Стоп»; *3* — кнопка «Сброс»

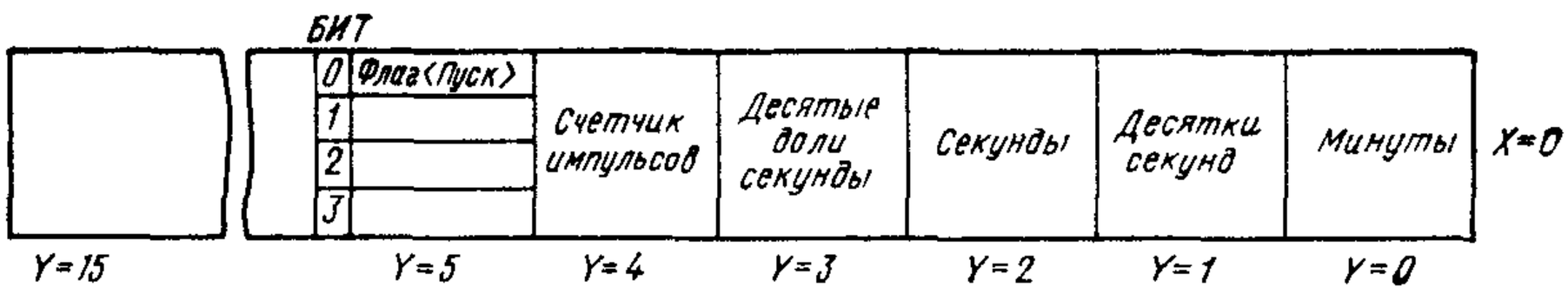

Рис. 6. Распределение памяти данных при реализации секундомера

Таблица 7

Адрес страницы ЗУ программы	Адрес команды	Код команды	Метка перехода	Мнемоника команды	Операнд	Примечание
15	0	00101111	*STA*	*CLA*		Старт программы 0→*M*
15	1	00111100		*LDX*	0	(*X*=0, *Y*=0—5)
15	3	01001010		*TCY*	5	
15	7	00000011	*CL*	*TAM*		
15	15	00101100		*DYN*		
15	31	10000111		*BR*	*CL*	
15	63	01000010	*DIS1*	*TCY*	4	Программа сканированиа
15	62	00101100	*DIS2*	*DYN*		клавиатуры и дисплея
15	61	00100001		*TMA*		
15	59	00101011		*IYC*		
15	55	00001100		*RSTR*		
15	47	00101100		*DYN*		
15	30	00001010		*TIQ*		
15	60	00001101		*SETR*		
15	57	00001001		*DNEZ*		
15	51	10111010		*BR*	*IMP*	
15	39	01011111	*DIS3*	*YNEC*	15	
15	14	10111110		*BR*	*DIS2*	
15	29	10111111		*BR*	*DIS1*	
15	58	00001000	*IMP*	*TDA*		Проверка нажатия кио-
15	53	01111000		*ALEC*	1	пок
15	43	10001010		*BR*	*F*	К определению кнопки
15	22	00100011		*TYA*		Проверка флага (пуск)
15	44	01001010		*TCY*	5	
15	24	00111000		*TBIT1*	0	
15	48	10000101		*BR*	*CLC*	Пуск — > иа счет вре-
						мени
15	33	00100100		*TAY*		
15	2	10100111		*BR*	*DIS3*	<Пуск> ие установлеи
15	5	00100100	*CLC*	*TAY*		Счет времени
15	11	00001100		*RSTR*		
15	23	01000010		*TCY*	4	
15	46	00101000		*IMAC*		
15	28	00000011		*TAM*		
15	36	01110010		*ALEC*	4	
15	49	10011001		*BR*	*WAIT*	
15	35	00101111	*INC*	*CLA*		
15	6	00000011		*TAM*		
15	13	00101100		*DYN*		
15	27	00101000		*IMAC*		
15	54	00000011		*TAM*		
15	45	00001111		*RETN*		
15	26	01111001		*ALEC*	9	
15	52	10011001		*BR*	*WAIT*	
15	41	11100011		*CALL*	*INC*	
15	18	01111001		*ALEC*	9	
15	36	10011001		*BR*	*WAIT*	
15	8	11100011		*CALL*	*INC*	
15	17	01111010		*ALEC*	5	
15	34	10011001		*BR*	*WAIT*	
15	4	11100011		*CALL*	*INC*	
15	9	01111001		*ALEC*	9	
15	19	10011001		*BR*	*WAIT*	
15	38	00101111		*CLA*		
15	12	00000011		*TAM*		
15	25	00001001	*WAIT*	*DNEZ*		Задержка до конца
						ИМП—СА
15	50	10011001		*BR*	*WAIT*	
15	37	10111111		*BR*	*DIS1*	

Окончаиие табл. 7

Адрес страницы ЗУ программы	Адрес команды	Код команды	Метка перехода	Мнемоника команды	Операнд	Примечание
15	10	00010000	*F*	*LDP*	0	
15	21	10000000		*BR*	*FILT*	
0	0	11000000		*MEMORY*	00 000	
0	0	00100011	*FILT*	*TYA*		Определение кнопки
0	1	01110000		*ALEC*	0	
0	3	10111110		*BR*	*SETFL*	
0	7	01111000		*ALEC*	1	
0	15	10101111		*BR*	*RSTFL*	
0	31	00011111		*LDP*	15	
0	63	10000000		*BR*	*STA*	
0	62	01001010	*SETFL*	*TCY*	5	Установка флага <пуск>
0	61	00110000		*SBIT*	0	
0	59	00100100		*TAY*		
0	55	10111001		*BR*	*DIS*	
0	47	01001010	*RSTFL*	*TCY*	5	Сброс <флага>
0	30	00110100		*RBIT*	0	<пуск>
0	60	00100100		*TAY*		
0	57	00011111	*DIS*	*LDP*	15	
0	51	10100111		*BR*	*DIS3*	
0	0	00000000		*END*		

каторе, единице соответствует 1 на индикаторе, и так до девяти. Программа микро-ЭВМ, реализующая секундомер, приведена ниже (табл. 7). В программе используются ячейки $Y=0—5$ нулевой страницы ОЗУ даиных. Назначение ячеек ОЗУ приведено на рис. 6.