

Л. М. Гольденберг
В. А. Малев
Г. Б. Малько

ЦИФРОВЫЕ УСТРОЙСТВА И МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

ЗАДАЧИ И УПРАЖНЕНИЯ

*Рекомендовано Комитетом по высшей школе
Миннауки России в качестве
учебного пособия для студентов вузов,
обучающихся по специальностям:
«Автоматическая электросвязь»,
«Радиосвязь, радиовещание и телевидение»,
«Многоканальная электросвязь»*



Москва
«Радио и связь»
1992

ББК 32.97

Г60

УДК 621.325.5-181.4(075)

Рецензенты: кафедра вычислительной техники МИЭТ (зав. кафедрой д-р техн. наук проф. В. А. Бархоткин), кафедра импульсной и вычислительной техники МИС (зав. кафедрой д-р техн. наук проф. Э. В. Евреинов)

Редакция литературы по информатике и вычислительной технике

Гольденберг Л. М. и др.

Г60 Цифровые устройства и микропроцессорные системы. Задачи и упражнения: Учеб. пособие для вузов / Л. М. Гольденберг, В. А. Малев, Г. Б. Малько.— М.: Радио и связь, 1992.— 256 с.: ил.

ISBN 5-256-00994-X.

Приводятся задачи и упражнения по анализу, синтезу и применению цифровых устройств различной сложности и назначения: цифровых логических элементов, кодопреобразователей, программируемых логических матриц, мультиплексоров и т. д., а также микропроцессоров и микропроцессорных систем. Каждый раздел содержит методические указания, ответы и решения. Имеются справочные материалы по используемой элементной базе: от простейших элементов серий 134, 155 до микропроцессора типа TMS 32010.

Для студентов вузов связи, обучающихся по специальностям «Автоматическая электросвязь», «Многоканальная электросвязь», «Радиосвязь, радиовещание и телевидение».

Г 2404000000-109
55-92
046(01)-92

ББК 32.97

Учебное издание

ГОЛЬДЕНБЕРГ Лев Моисеевич

МАЛЕВ Владимир Абрамович

МАЛЬКО Георгий Борисович

ЦИФРОВЫЕ УСТРОЙСТВА И МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ
Задачи и упражнения

Учебное пособие

Заведующая редакцией *Г. И. Козырева* Редактор *Т. М. Бердичевская*
Переплет художника *Ю. В. Архангельского* Художественный редактор *Н. С. Шенин*.
Технический редактор *Т. Г. Родина*. Корректор *Н. В. Козлова*.

ИБ № 1943

Сдано в набор 21.07.92. Подписано в печать 02.10.92. Формат 60×82_{1/16}. Бумага кн.-журн. Гарнитура «Таймс». Печать офсетная. Усл. печ. л. 15,68. Усл. кр.-отг. 16,05. Уч.-изд. л. 16,90. Тираж 12 500 экз. Изд. № 22671. Зак. № 775. С-109

Издательство «Радио и связь» 101000 Москва, Почтамт, а/я 693

Ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО «Первая Образцовая типография» Министерства печати и информации Российской Федерации. 113054, Москва, Валовая, 28

© Гольденберг Л. М., Малев В. А.,
Малько Г. Б., 1992

ISBN 5-256-00994-X

ПРЕДИСЛОВИЕ

Для современного этапа развития цифровой техники (ЦТ) характерным является то, что в основе построения цифровых устройств (ЦУ) лежит сравнительно небольшое число принципов, а на практике применяется огромное число различных технических решений, чрезвычайно разнообразна элементная база ЦТ и для описания работы ЦУ используется значительное число различных параметров и характеристик. В этих условиях разобраться в работе того или иного ЦУ и тем более спроектировать устройство с заданными характеристиками способен лишь весьма квалифицированный специалист. С одной стороны, он должен быть хорошо знаком с элементной базой — микросхемами различных типов и различных уровней интеграции, а также с методами логического проектирования; с другой стороны, для эффективного использования микропроцессоров (МП) он должен владеть методами алгоритмизации и программирования соответствующих технических задач. При этом необходимо учитывать конструктивные, технологические и экономические требования.

Многолетний опыт преподавания показывает, что только при решении задач анализа и синтеза ЦУ можно достичь глубокого понимания проблем ЦТ, умения творчески применять теоретические знания.

В соответствии с типовыми программами дисциплины базовой подготовки по цифровой технике радиотехнических и связанных специальностей ориентированы в первую очередь на изучение аппаратных средств. В связи с этим авторы не ставили своей целью обучить читателя методам алгоритмизации задач и программированию микроЭВМ на языках высокого уровня; эти цели ставятся в иных дисциплинах и иных учебных пособиях. Однако использование МП в отличие от цифровых микросхем малого и среднего уровней интеграции невозможно без знания методов их программирования. Поэтому в книгу включено довольно много задач по программированию МП и микропроцессорных систем (МПС) в машинных кодах, мнемокодах и на языке ассемблера. Тематика этих задач подобрана таким образом, чтобы акцентировать внимание читателя на функциях и характеристиках основных программно-доступных узлов МП и МПС, а также на особенностях команд, микрокоманд и микроопераций.

Учебное пособие содержит более 500 задач, вопросов и упражнений. Большинство из них снабжены подробными решениями и ответами. Среди задач имеются как чисто учебные, так и исследовательские; некоторые могут использоваться при курсовом проектировании, многие — при самостоятельной работе. Необходимые справочные сведения приведены в начале каждого параграфа, в условиях задач или в приложениях.

В решениях большинства задач приводятся краткие пояснения, но, разумеется, перед решением необходимо изучить соответствующий раздел по учебнику [2] или иной рекомендованной литературе.

Количество рассматриваемых в задачах элементов и устройств намеренно ограничено небольшим числом типов, и используются, как правило, не все возможные режимы работы. По элементам и устройствам приводятся лишь те справочные сведения, которые нужны для решения помещенных в книге задач.

Авторами книги являются преподаватели Ленинградского электротехнического института связи им. проф. М. А. Бонч-Бруевича. Главы 1, 6, 7 и приложения 3, 4 написаны В. А. Малевым, гл. 8 — В. А. Малевым (задачи 8.1—8.35) и Л. М. Гольденбергом (задачи 8.36—8.43), главы 2, 4, 5 и приложения 1, 2 — Г. Б. Малько, гл. 3 — Г. Б. Малько (задачи 3.1—3.37) и В. А. Малевым (задачи 3.38—3.67).

Глава 1. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ И АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ В ЦИФРОВЫХ УСТРОЙСТВАХ

1.1. ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

Для представления чисел в цифровых устройствах при выполнении арифметических операций, а также в текстах программ используются помимо десятичной двоичная, восьмеричная, шестнадцатеричная и двоично-десятичная системы счисления¹.

ЗАДАЧИ И УПРАЖНЕНИЯ

1.1. Определить значения следующих выражений²:

$$\log_2 16; \lfloor \log_2 7 \rfloor; \lceil \log_2 7 \rceil; \lceil \log_2 100 \rceil; \lfloor \log_2 1000 \rfloor; \\ \lceil \log_2 65000 \rceil; \lceil \log_2 10^6 \rceil; \lceil \log_2 7^2 \rceil; \lfloor \log_2 7 \rfloor; \lfloor \log_2 17 \rfloor.$$

1.2. Сколько различных чисел можно записать с помощью n -разрядного кода с основанием q . Произвести расчет при $n=4$; 8 и 16 и $q=10$; 2; 8; 16.

1.3. Обосновать известный алгоритм перевода целого числа из одной позиционной системы счисления в другую: число делится на основание новой системы счисления p и определяется остаток; частное вновь делится на p и вновь определяется остаток; так продолжается до тех пор, пока частное не окажется меньше p ; после этого выписывается последнее частное и все остатки—это и будет представление числа в системе счисления с основанием p .

1.4. Обосновать известный алгоритм перевода дробного числа из одной позиционной системы счисления в другую; исходное число умножается на основание новой системы

¹ Основание системы счисления указывается либо цифровым индексом ($11100_{(2)}$, $28_{(10)}$, $1C_{(16)}$), либо буквой (соответственно $11100B$, $28D$, $1CH$). После десятичного числа индекс 10 или буква D иногда не ставится. В некоторых случаях, если система счисления понятна из контекста, индексы и буквы не ставятся и в других системах счисления.

² $\lfloor A \rfloor$ —целая часть A ; $\lceil A \rceil$ —ближайшее к A целое число, не меньшее A ; $\{A\}$ —ближайшее к A целое число, например $\lfloor 2, 3 \rfloor = 2$, $\lceil 2, 3 \rceil = 3$, $\{2, 3\} = 2$, $\{2, 6\} = 3$.

счисления p , затем дробная часть произведения вновь умножается на p ; так поступают столько раз, сколько разрядов числа в новой системе счисления желательно получить. Затем выписываются целые части всех произведений — это и будет представление числа в системе счисления с основанием p .

1.5. Чем определяется точность числа в позиционной системе счисления? Какое число разрядов необходимо иметь, чтобы получить заданную точность?

1.6. Дробное число, содержащее n разрядов, переводится из системы счисления с основанием q в систему счисления с основанием p . Какое число разрядов m необходимо взять, чтобы сохранить точность исходного числа? Произвести расчет для $q=10$, $p=2$.

1.7. Перевести следующие числа из десятичной системы счисления в двоичную и из двоичной в шестнадцатеричную:

а) $27_{(10)}$; б) $127_{(10)}$; в) $74_{(10)}$ (см. табл. П.3.3).

1.8. Перевести следующие числа из шестнадцатеричной системы в двоичную и из двоичной в десятичную, а также непосредственно из шестнадцатеричной в десятичную:

а) $D5_{(16)}$; б) $127_{(16)}$; в) $E1_{(16)}$; г) $CAB_{(16)}$; д) $11C2_{(16)}$.

1.9. Перевести следующие числа из двоичной системы счисления в восьмеричную и из восьмеричной в десятичную с точностью $0,0001_{(10)}$:

а) $10101_{(2)}$; б) $1,101_{(2)}$; в) $10,0101_{(2)}$; г) $1,11001_{(2)}$.

1.10. Перевести следующие числа из десятичной системы в шестнадцатеричную без промежуточного перевода в двоичную с точностью до трех разрядов после запятой:

а) $1125_{(10)}$; б) $53127_{(10)}$; в) $0,216_{(10)}$.

1.11. Перевести следующие числа из двоичной системы в десятичную с погрешностью до $0,001_{(10)}$:

а) $1,001_{(2)}$; б) $0,1011_{(2)}$; в) $101,1001_{(2)}$.

1.12. Перевести следующие числа из десятичной системы счисления в восьмеричную без промежуточного перевода в двоичную систему:

а) $23_{(10)}$; б) $1,25_{(10)}$; в) $100,57_{(10)}$ с погрешностью не более $0,001_{(10)}$.

1.13. Не переводя указанные числа в десятичную систему счисления, оценить, какое из них самое большое и какое самое маленькое: $101110_{(2)}$; $101110_{(8)}$; $101110_{(16)}$?

1.14. Перевести указанные числа в десятичную систему счисления:

а) $11111_{(2)}$; б) $FFF_{(16)}$; в) $7777_{(8)}$; г) $\underbrace{NNN \dots N}_{n \text{ разрядов}}_{(N+1)}$.

1.15. Перевести указанные числа в двоичную систему счисления:

а) $2^8_{(10)}$; б) $(2^6 - 1)_{(10)}$; в) $(2^k - 1)_{(10)}$.

1.16. Перевести следующие числа из десятичной системы в двоичную с точностью до четырех разрядов после запятой:

а) $0,65_{(10)}$; б) $23,625_{(10)}$; в) $1,217_{(10)}$.

1.17. Представить в двоично-десятичном коде 8421 следующие числа:

а) $27_{(10)}$; б) $316_{(10)}$; в) $4571_{(10)}$; г) $01011001_{(2)}$.

1.18. Перевести из двоично-десятичного кода 8421 в десятичный и двоичный коды следующие числа:

а) $00100110_{(2/10)}$; б) $10010010_{(2/10)}$.

1.2. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ С ФИКСИРОВАННОЙ И ПЛАВАЮЩЕЙ ТОЧКОЙ

В вычислительной технике используются две основные формы представления чисел: с фиксированной и плавающей точкой [5, 14, 28]. При одинаковом числе разрядов вторая форма обеспечивает запись чисел более широкого диапазона и с меньшей погрешностью, но аппаратная реализация соответствующего арифметического устройства оказывается сложнее, чем при записи с фиксированной точкой.

ЗАДАЧИ И УПРАЖНЕНИЯ

1.19. Записать в 8-разрядной ячейке в прямом коде с фиксированной точкой следующие двоичные числа:

а) $0,1011010$; б) $-0,0101011$; в) $-0,111100101$.

1.20. Записать в 16-разрядной ячейке в прямом коде с плавающей точкой следующие числа:

а) $0,1010010$; б) $-11011, 1010111$; в) $0,0001011011101$; для мантииссы и порядка выделить по восемь разрядов.

1.21. Сколько двоичных разрядов необходимо использовать для записи с фиксированной точкой числа, находящегося в диапазоне от $-100_{(10)}$ до $+100_{(10)}$, с абсолютной погрешностью не хуже:

а) $0,1_{(10)}$; б) $0,001_{(10)}$; в) $0,000001_{(10)}$?

1.22. Сколько требуется разрядов, чтобы записать с плавающей точкой точные значения следующих двоичных чисел:

а) $0,0001011101$; б) $11,0101101$; в) $-0,011010111$; г) $-11011, 000001$; д) $-1,0011110001$?

1.23. Ячейка памяти, содержащая n двоичных разрядов, может использоваться для записи чисел с фиксированной или плавающей точкой. Как следует распределить разряды между мантииссой и порядком в случае плавающей точки, чтобы обеспечить запись чисел более широкого диапазона, чем при фиксированной точке? Произвести расчет при $n=16$.

1.24. В чем преимущества и недостатки способов записи чисел с фиксированной и плавающей точкой?

1.3. КОДИРОВАНИЕ ЧИСЕЛ И АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ¹

Арифметические операции обычно выполняются с применением прямого, обратного и дополнительного кодов. Арифметические основы цифровой техники подробно излагаются в [5, 14, 28].

ЗАДАЧИ И УПРАЖНЕНИЯ

1.25. Обосновать известное соотношение между дополнительным и обратным кодами отрицательного числа A : $\langle A \rangle_{\text{доп}} = \langle A \rangle_{\text{обр}} + 1$ младшего разряда.

1.26. Обосновать известное правило перехода от отрицательного числа к его дополнительному коду и от дополнительного кода отрицательного числа к самому числу: необходимо инвертировать все разряды исходного числа или кода, после чего прибавить к результату единицу младшего разряда.

1.27. Обосновать известное правило сложения двух чисел, каждое из которых может быть и положительным, и отрицательным: для того чтобы найти сумму двух чисел, необходимо сложить дополнительные коды этих чисел; полученный результат является дополнительным кодом искомой суммы. (Если при сложении дополнительных кодов образуется перенос из знакового разряда, то он не учитывается.)

1.28. Обосновать известное правило определения разности двух чисел, каждое из которых может быть и положительным, и отрицательным: чтобы вычесть из числа A число B , не используя операцию вычитания, необходимо сложить дополнительные коды чисел A и $-B$; полученная сумма является дополнительным кодом искомой разности. (Если при сложении дополнительных кодов получается перенос из знакового разряда, то он не учитывается.)

1.29. Записать в прямом коде следующие числа:

а) 0,101; б) $-0,011$; в) 0,0001; г) $-0,0101$; д) 1101;
е) -1011101 ; ж) 1011101^2 .

1.30. Записать в дополнительном коде следующие числа:

а) 0,1011; б) $-0,1011$; в) 0,1101; г) $-0,1001$; д) $-0,1000$;
е) $-0,0001$; ж) $-0,111111$; з) 1101; и) -1011101 ; к) 1011101.

1.31. Каким числам соответствуют следующие дополнительные коды:²

а) 0,1100; б) 1,0001; в) 1,0111; г) 1,10000; д) 1,0101; е) 0,1010;
ж) 1,01001; з) 01011; и) 101111; к) 1010110?

¹ Задачи на составление и обоснование алгоритмов арифметических операций приводятся также в гл. 6—8.

² Запятая в кодах дробных чисел условно отделяет знаковый разряд от дробной части, в кодах целых чисел запятая не ставится.

1.32. Воспользовавшись дополнительными кодами чисел, рассчитать $S = A + B$:

- а) $A = 0,11010$; $B = -0,10001$;
- б) $A = -0,11010$; $B = 0,10001$;
- в) $A = 0,1010$; $B = -0,1101$;
- г) $A = -1010$; $B = 1000$;
- д) $A = 1011010$; $B = -1001001$.

Проверить полученные результаты простым вычислением.

1.33. Воспользовавшись дополнительными кодами чисел, рассчитать $R = A - B$, если

- а) $A = 0,11101$; $B = 0,01011$;
- б) $A = -0,0101$; $B = 0,1010$;
- в) $A = -0,1001$; $B = -0,0111$;
- г) $A = 1101010$; $B = 1010101$;
- д) $A = 1010101$; $B = 1101010$.

1.34. Рассчитать $\Pi = A \cdot B$, если

- а) $A = -0,1011$; $B = 0,1101$;
- б) $A = 0,01011$; $B = -0,10111$;
- в) $A = B = 0,1111$;
- г) $A = -1010111$; $B = -101$.

1.35. Рассчитать $D = A : B$ с точностью до четвертого знака после запятой, если

- а) $A = -0,10011$; $B = 0,10101$;
- б) $A = -0,011001$; $B = 0,101$;
- в) $A = 0,1110001$; $B = -0,1111$;
- г) $A = 1010111$; $B = 101$.

1.36. Сколько двоичных разрядов (без учета знака) содержит:

- а) сумма двух целых n -разрядных двоичных чисел;
- б) разность двух целых n -разрядных двоичных чисел;
- в) произведение двух целых n -разрядных двоичных чисел;
- г) произведение целого n -разрядного двоичного числа и целого m -разрядного двоичного числа?

1.37. Перевести нижеуказанные числа A и B в двоично-десятичный код 8421, сложить и результат перевести обратно в десятичный код:

- а) $A = 0,597$; $B = 0,346$;
- б) $A = 0,2098$; $B = 0,3729$;
- в) $A = 0,3721$; $B = 0,5683$.

ОТВЕТЫ И РЕШЕНИЯ

Р1.1. $\lfloor \log_2 16 \rfloor = 4$; $\lfloor \log_2 7 \rfloor = 2$; $\lceil \log_2 7 \rceil = 3$;
 $\lceil \log_2 100 \rceil = 7$; $\lfloor \log_2 1000 \rfloor = 9$; $\lceil \log_2 65000 \rceil = 16$;
 $\lceil \log_2 10^6 \rceil = 20$; $\lceil \log_2 7^2 \rceil = 6$; $\lfloor \log_2 7 \rfloor = 3$; $\lfloor \log_2 17 \rfloor = 4$.

Р1.2. В позиционном коде с основанием q используются q различных цифр — от 0 до $q-1$. Поскольку в каждой позиции

n -разрядного кода может находиться любая из этих цифр, такой код позволяет записать $N=q^n$ различных чисел.

P1.3. Целое число A в системе с основанием p имеет вид $A=a_n a_{n-1} \dots a_i \dots a_2 a_1$, где $a_i \in \{0, 1, \dots, p-1\}$, $i=1, \dots, n$; a_i — неизвестные пока значения разрядов, т. е.

$$A = \sum_{i=1}^n a_i p^{i-1} = a_n p^{n-1} + a_{n-1} p^{n-2} + \dots + a_i p^{i-1} + \dots a_1.$$

Последовательно деля A и полученные частные B_j на p , имеем

$$A/p = a_n p^{n-2} + a_{n-1} p^{n-3} + \dots + a_3 p + a_2 = B_1, \text{ остаток } a_1;$$

$$B_1/p = a_n p^{n-3} + a_{n-1} p^{n-4} + \dots + a_3 = B_2, \text{ остаток } a_2;$$

$$B_{n-1}/p = a_n, \text{ остаток } a_{n-1}.$$

Таким образом, в остатке каждый раз получается соответствующий разряд числа в системе счисления с основанием p .

P1.4. Дробное число A в системе счисления с основанием p имеет вид $A=0, a_1 a_2 \dots a_i \dots a_m$, где a_i — неизвестные пока значения разрядов, причем $a_i \in \{0, 1, \dots, p-1\}$, $i=1, \dots, m$. Следовательно,

$$A = a_1 p^{-1} + a_2 p^{-2} + \dots + a_m p^{-m}.$$

Последовательно умножая это выражение и дробные части полученных произведений C_j на p , имеем

$$Ap = a_1 + a_2 p^{-1} + a_3 p^{-2} + \dots + a_m p^{-m+1} = a_1 + C_1;$$

$$C_1 p = a_2 + a_3 p^{-1} + \dots + a_m p^{-m+2} = a_2 + C_2;$$

$$C_m p^m = a_m.$$

Таким образом, целые части произведений являются соответствующими разрядами числа в системе счисления с основанием p .

P1.5. Для получения абсолютной погрешности, не превышающей Δ , необходимо иметь в системе с основанием p такое число разрядов m , чтобы выполнялось соотношение

$$1/p^m \leq \Delta, \text{ т. е. } m \geq \left\lceil \frac{\log 1/\Delta}{\log p} \right\rceil.$$

P1.6. Считаем, что абсолютная погрешность числа определяется единицей младшего разряда. Тогда для сохранения точности должно быть

$$1/p^m \leq 1/q^n, \text{ т. е. } m \geq n \frac{\log q}{\log p}.$$

При $q=10$, $p=2$ должно быть $m \geq 3,3n$.

$$\text{P1.7. a) } 27_{(10)} = 11011_{(2)} = 1B_{(16)};$$

$$\text{б) } 127_{(10)} = 1111111_{(2)} = 7F_{(16)};$$

$$\text{в) } 74_{(10)} = 1001010_{(2)} = 4A_{(16)}.$$

$$\text{P1.8. a) } D5_{(16)} = 1101\ 0101_{(2)} = 213_{(10)}; \quad D5_{(16)} = 13 \cdot 16 + 5 = 213_{(10)};$$

$$\text{б) } 127_{(16)} = 0001\ 0010\ 0111_{(2)} = 295_{(10)}; \quad 127_{(16)} = 1 \cdot 16^2 + 2 \cdot 16 + 7 = 295_{(10)};$$

$$\text{в) } E1_{(16)} = 1110\ 0001_{(2)} = 225_{(10)};$$

$$\text{г) } CAB_{(16)} = 1100\ 1010\ 1011_{(2)} = 3243_{(10)};$$

$$\text{д) } 11C2_{(16)} = 0001\ 0001\ 1100\ 0010_{(2)} = 4546_{(10)}.$$

$$\text{P1.9. a) } 10101_{(2)} = 25_{(8)} = 2 \cdot 8 + 5 = 21_{(10)};$$

$$\text{б) } 1,101_{(2)} = 1,5_{(8)} = 1 \cdot 8^0 + 5 \cdot 8^{-1} = 1 \frac{5}{8} = 1,6250_{(10)};$$

$$\text{в) } 10,0101_{(2)} = 2,24_{(8)} = 2 \cdot 8^0 + 2 \cdot 8^{-1} + 4 \cdot 8^{-2} = 2,0625_{(10)};$$

$$\text{г) } 1,11001_{(2)} = 1,62_{(8)} = 1 \cdot 8^0 + 6 \cdot 8^{-1} + 2 \cdot 8^{-2} = 1,7813_{(10)}.$$

$$\text{P1.10. a) } 1125 \lfloor 16 \quad 1125_{(10)} = 465_{(16)}$$

$$\begin{array}{r} 1120 \\ \hline 5 \end{array} \quad \begin{array}{r} 70 \lfloor 16 \\ 64 \quad 4 \\ \hline 6 \end{array}$$

$$\text{б) } 53127 \lfloor 16$$

$$\begin{array}{r} 48 \\ \hline 51 \end{array} \quad \begin{array}{r} 3320 \lfloor 16 \\ 32 \\ \hline 120 \end{array} \quad \begin{array}{r} 207 \lfloor 16 \\ 16 \\ \hline 47 \end{array} \quad 12 = C$$

$$\begin{array}{r} 48 \\ \hline 32 \end{array} \quad \begin{array}{r} 112 \\ \hline 8 \end{array} \quad \begin{array}{r} 32 \\ \hline 15 = F \end{array}$$

$$53127_{(10)} = CF87_{(16)}$$

$$\text{в) } 0,216$$

$$\quad 16$$

$$+ \quad 1\ 296$$

$$\quad 2\ 16$$

$$\hline 3,456$$

$$\quad 16$$

$$+ \quad 2\ 736$$

$$\quad 4\ 56$$

$$\hline 7,296$$

$$\quad 16$$

$$+ \quad 1\ 776$$

$$\quad 2\ 96$$

$$\hline 4,736$$

$$0,216_{(10)} \approx 0,374_{(16)}$$

P1.11. а) $1,001_{(2)} = 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 1,125_{(10)}$;

б) $0,1011_{(2)} \approx 0,688_{(10)}$;

в) $101,1001_{(2)} = 5,563_{(10)}$.

P1.12. а) $23_{(10)} = 27_{(8)}$;

б) $1,25_{(10)} = 1,2_{(8)}$;

в) $100,57_{(10)} = 144,4436_{(8)}$

(для получения заданной точности $0,001_{(10)}$ необходимо иметь в дробной части восьмичисленного числа четыре разряда).

P1.13. $101110_{(16)} > 101110_{(8)} > 101110_{(2)}$.

P1.14. а) $11111_{(2)} = 2^5 - 1 = 31_{(10)}$;

б) $FFF_{(16)} = 16^3 - 1 = 4095_{(10)}$;

в) $7777_{(8)} = 8^4 - 1 = 4095_{(10)}$;

г) $(N+1)^n - 1$.

P1.15. а) $2^8_{(10)} = 100000000_{(2)}$;

б) $(2^6 - 1)_{(10)} = 111111_{(2)}$;

в) $2^k - 1_{(10)} = \underbrace{11 \dots 1}_{k \text{ разрядов}}_{(2)}$

P1.16. а) $0,65_{(10)} \approx 0,1010_{(2)}$;

б) $23,625_{(10)} = 10111,1010_{(2)}$;

в) $1,217_{(10)} \approx 1,0011_{(2)}$.

P1.17. Каждая цифра переводится независимо от других в 4-разрядный двоичный код:

а) $27_{(10)} = 0010 \ 0111_{(2/10)}$;

б) $316_{(10)} = 0011 \ 0001 \ 0110_{(2/10)}$;

в) $4571_{(10)} = 0100 \ 0101 \ 0111 \ 0001_{(2/10)}$;

г) $0101 \ 1001_{(2)} = 89_{(10)} = 1000 \ 1001_{(2/10)}$.

P1.18. Число делится на тетрады, и каждая тетрада независимо от других переводится в десятичный код:

а) $0010 \ 0110_{(2/10)} = 26_{(10)} = 11010_{(2)}$;

б) $1001 \ 0010_{(2/10)} = 92_{(10)} = 1011100_{(2)}$.

P1.19. а) 01011010 ; б) 10101011 ; в) 11111001 .

P1.20. Предварительно число должно быть нормализовано.

а) $0 \ \underline{1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0} \quad 0 \quad \underline{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0}$

Знак
мантиссы

Модуль мантиссы

Знак
порядка

Модуль
порядка

В ячейке записано число $0,1010010 \cdot 2^0 = 0,1010010$.

б) $1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0.0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1$

В ячейке записано число $-0,1101110 \cdot 2^5 = -11011,10$.

в) $0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1$

В ячейке записано число $0,1011011 \cdot 2^{-3} = 0,0001011011$.

P1.21. а) Для целой части числа необходимо выделить $v = \lceil \log_2 100_{(10)} \rceil = 7$ двоичных разрядов, для дробной части $\mu = \lceil |\log_2 0,1_{(10)}| \rceil = 4$ разряда. Учитывая знаковый разряд, всего получаем 12 разрядов:

б) 18; в) 28.

P1.22. С учетом знаковых разрядов мантииссы и порядка:
а) 11; б) 13; в) 11; г) 16; д) 14.

P1.23. Максимальное n -разрядное число при записи с фиксированной точкой — это целое положительное число $A_{\text{макс}}^{\Phi} = 2^{n-1} - 1$ (один разряд — знаковый). Минимальное число $A_{\text{мин}}^{\Phi} = -A_{\text{макс}}^{\Phi} = -(2^{n-1} - 1)$. Диапазон чисел $D^{\Phi} = A_{\text{макс}}^{\Phi} - A_{\text{мин}}^{\Phi} = 2(2^{n-1} - 1)$.

Максимальное число при записи с плавающей точкой $A_{\text{макс}}^{\text{пл}} = (1 - 2^{-(m-1)}) 2^{p-1} - 1$, где p — число разрядов порядка (включая один знаковый); m — число разрядов мантииссы (включая один знаковый); $p + m = n$. Минимальное число $A_{\text{мин}}^{\text{пл}} = -A_{\text{макс}}^{\text{пл}}$. Диапазон чисел $D^{\text{пл}} = A_{\text{макс}}^{\text{пл}} - A_{\text{мин}}^{\text{пл}} = (1 - 2^{-(m-1)}) \cdot 2^{2^{p-1}}$.

По условию задачи должно быть $D^{\text{пл}} > D^{\Phi}$, т. е.

$$(1 - 2^{-(m-1)}) \cdot 2^{2^{p-1}} > 2(2^{n-1} - 1).$$

При $n, m \gg 1$ получим $2^{2^{p-1}-1} > 2^{n-1}$ и после двукратного логарифмирования

$$p > 1 + \log_2(n - 1).$$

Это неравенство и является условием, при котором способ с плавающей точкой обеспечивает запись чисел более широкого диапазона, чем способ с фиксированной точкой.

При $n = 16$ получим $p > 1 + \log_2 15$, т. е. для целочисленных p и m $p \geq 5$ и $m = n - 5 \leq 11$. Пусть, например, $p = 7$ и $m = 9$. Тогда $A_{\text{макс}}^{\Phi} = 2^{15} - 1 \approx 3 \cdot 10^4$; $D^{\Phi} \approx 6 \cdot 10^4$; $A_{\text{макс}}^{\text{пл}} = (1 - 2^{-8}) \cdot 2^{2^6-1} \approx 2^{63} \approx 10^{19}$; $D^{\text{пл}} \approx 2 \cdot 10^{19} \gg D^{\Phi}$.

P1.24. При использовании фиксированной точки арифметические операции над числами проще (например, при суммировании не нужно выравнивать порядки слагаемых), но диапазон представимых чисел существенно меньше, чем при использовании плавающей точки. (Сравниваются, конечно, одинаковые по числу разрядов форматы.)

P1.25. По определению дополнительного кода для целого отрицательного числа имеем

$$\langle A \rangle_{\text{доп}} = 2^n - |A|,$$

где n — число двоичных разрядов, включая знаковый.

Следовательно,

$$\langle A \rangle_{\text{доп}} = \underbrace{111 \dots 11}_{n \text{ разрядов}}_{(2)} + 1 - |A| = (111 \dots 11_{(2)} - |A|) + 1 = \langle A \rangle_{\text{обр}} + 1.$$

Аналогично для дробного отрицательного числа

$$\begin{aligned} \langle A \rangle_{\text{доп}} &= 2 - |A| = 10,00 \dots 00_{(2)} - |A| = (1,11 \dots 11_{(2)} - |A|) + \\ &+ 0,00 \dots 01_{(2)} = \langle A \rangle_{\text{обр}} + 0,00 \dots 01_{(2)}. \end{aligned}$$

P1.26. Правило для перехода от целого отрицательного числа к его дополнительному коду следует из соотношения

$$\langle A \rangle_{\text{доп}} = \langle A \rangle_{\text{обр}} + 1,$$

доказанного в задаче 1.25.

Правило для обратного перехода следует из соотношений

$$\langle A \rangle_{\text{доп}} = 2^n - |A|; |A| = 2^n - \langle A \rangle_{\text{доп}} = (111 \dots 1 - \langle A \rangle_{\text{доп}}) + 1.$$

Аналогичные правила имеют место и для дробных чисел.

P1.27. Рассмотрим различные случаи сложения двух дробных чисел A и B :

1) $A, B > 0$;

$$\langle A \rangle_{\text{доп}} + \langle B \rangle_{\text{доп}} = A + B = \langle A + B \rangle_{\text{доп}},$$

т. е. дополнительный код суммы равен сумме дополнительных кодов слагаемых:

2) $A > 0, B < 0$;

$$\langle A \rangle_{\text{доп}} + \langle B \rangle_{\text{доп}} = A + (2 + B) = 2 + (A + B).$$

Если $A > |B|$, то $A + B > 0$, $A + B = \langle A + B \rangle_{\text{доп}}$ и $\langle A \rangle_{\text{доп}} + \langle B \rangle_{\text{доп}} = 2 + \langle A + B \rangle_{\text{доп}}$, т. е. образуется лишняя двойка (перенос из знакового разряда), которая не должна учитываться.

Если $A < |B|$, то $A + B < 0$; $2 + (A + B) = \langle A + B \rangle_{\text{доп}}$ и $\langle A \rangle_{\text{доп}} + \langle B \rangle_{\text{доп}} = \langle A + B \rangle_{\text{доп}}$;

3) $A, B < 0$;

$$\langle A \rangle_{\text{доп}} + \langle B \rangle_{\text{доп}} = 2 + A + 2 + B = 2 + (2 + A + B) = 2 + \langle A + B \rangle_{\text{доп}},$$

т. е. опять образуется лишняя двойка.

Таким образом, сумма дополнительных кодов слагаемых всегда равна (с точностью до несущественного переноса из знакового старшего разряда) дополнительному коду суммы. Рассмотрен случай дробных чисел, для целых чисел результат аналогичен.

P1.28. Поскольку $A - B = A + (-B)$, можно применить то же доказательство, что и в предыдущей задаче.

P1.29. а) 0,101; б) 1,011; в) 0,0001; г) 1,0101; д) 01101; е) 11011101; ж) 01011101.

P1.30. а) Поскольку число положительное, $\langle 0,1011 \rangle_{\text{доп}} = 0,1011$;

б) для отрицательного числа $\langle -0,1011 \rangle_{\text{доп}} = \langle 0,1011 \rangle_{\text{обр}} + 0,0001 = 1,0100 + 0,0001 = 1,0101$;

в) 0,1101; г) 1,0111; д) 1,1000; е) 1,1111; ж) 1,000001; з) 01101; и) 10100011; к) 01011101.

P1.31. а) Поскольку знаковый разряд 0, число положительное; оно совпадает с дополнительным кодом, т. е. 0,1100;

б) поскольку знаковый разряд 1, число отрицательное. Поэтому инвертируем все разряды, после чего добавляем единицу в младшем разряде. Получаем—0,1111;

- в) $-0,1010$; г) $-0,10000$; д) $-0,1011$; е) $0,1010$; ж) $-0,10111$;
 з) 1011 ; и) -01000 ; к) -0101010 .

P1.32. а) $\langle A \rangle_{\text{доп}} = 0,11010$; $\langle B \rangle_{\text{доп}} = 1,01111$; $\langle A \rangle_{\text{доп}} + \langle B \rangle_{\text{доп}} = 10,01001$;

$\langle S \rangle_{\text{доп}} = 0,01001$ (образовавшийся перенос в знаковом разряде не учитываем); $S = 0,01001$;

б) $\langle A \rangle_{\text{доп}} = 1,00110$; $\langle B \rangle_{\text{доп}} = 0,10001$; $\langle S \rangle_{\text{доп}} = 1,10111$; $S = -0,01001$;

в) $\langle A \rangle_{\text{доп}} = 0,1010$; $\langle B \rangle_{\text{доп}} = 1,0011$; $\langle S \rangle_{\text{доп}} = 1,1101$; $S = -0,0011$;

г) $\langle A \rangle_{\text{доп}} = 10110$; $\langle B \rangle_{\text{доп}} = 01000$; $\langle S \rangle_{\text{доп}} = 11110$; $S = -0,0010$;

д) $\langle A \rangle_{\text{доп}} = 01011010$; $\langle B \rangle_{\text{доп}} = 10110111$; $\langle S \rangle_{\text{доп}} = 00010001$; $S = 0010001$.

P1.33. а) $\langle A_{\text{доп}} \rangle = 0,11101$; $\langle -B \rangle_{\text{доп}} = 1,10101$; $\langle R \rangle_{\text{доп}} = 0,10010$; $R = 0,10010$;

б) $\langle A \rangle_{\text{доп}} = 1,1011$; $\langle -B \rangle_{\text{доп}} = 1,0110$; $\langle R \rangle_{\text{доп}} = 1,0001$; $R = -0,1111$;

в) $\langle A \rangle_{\text{доп}} = 1,0111$; $\langle -B \rangle_{\text{доп}} = 0,0111$; $\langle R \rangle_{\text{доп}} = 1,1110$; $R = -0,0010$;

г) $\langle A \rangle_{\text{доп}} = 01101010$; $\langle -B \rangle_{\text{доп}} = 10101011$; $\langle A \rangle_{\text{доп}} + \langle -B \rangle_{\text{доп}} = 100010101$; $\langle R \rangle_{\text{доп}} = 00010101$; $R = 0010101$;

д) $\langle A \rangle_{\text{доп}} = 01010101$; $\langle -B \rangle_{\text{доп}} = 10010110$; $\langle A \rangle_{\text{доп}} + \langle -B \rangle_{\text{доп}} = 11101011 = \langle R \rangle_{\text{доп}}$; $R = -0010101$.

P1.34. а) $-0,10001111$; б) $-0,0011111101$; в) $0,11100001$;
 г) 110110011 .

P1.35. а) $\begin{array}{r} 10011 \quad \underline{10101} \\ 100110 \quad 0,1110 \end{array}$

$\begin{array}{r} 10101 \\ \underline{100010} \end{array}$

$\begin{array}{r} 10101 \\ \underline{11010} \end{array}$

$\begin{array}{r} 10101 \\ \underline{1010} \end{array}$

$D \approx 0,1110$; остаток $0,00000101$;

б) $-0,1010$, остатка нет;

в) $-0,1111$, остатка нет;

г) 10001 , 011 , остаток есть

P1.36. а, б) Не более $n+1$; в) не более $2n$; г) не более $n+m$.

P1.37. Суммирование чисел, представленных в двоично-десятичном коде 8421, выполняется в два этапа [28]. На 1-м этапе вычисляется предварительная сумма S_n , причем суммирование производится по обычным правилам двоичной арифметики, но со следующими уточнениями и дополнениями:

если в i -й тетраде S_{pi} предварительной суммы образуется естественный перенос, то он учитывается в следующей $(i+1)$ -й тетраде;

если в i -й тетраде S_{pi} естественного переноса нет, но $S_{pi} \geq 10$, то считается, что есть искусственный перенос, и он опять же учитывается в $(i+1)$ -й тетраде;

при $S_{pi} < 10$ переноса нет.

На 2-м этапе каждая тетрада S_{pi} корректируется по следующему правилу:

если $S_{pi} \geq 10$, т. е. в данной тетраде образовался естественный или искусственный перенос, то к ней прибавляется число 6; образующийся при этом перенос не учитывается. Ниже в примерах естественный перенос обозначается 1^e , искусственный — 1^u .

а)	0,597	0,0101	1001	0111	
	+	+			
	0,346	0,0011	0100	0110	
	<u>0,943</u>	0,1001	1110	1101	предварительная сумма
		+			
		0000	0110	0110	коррекция
		0,1001	0100	0011	сумма

б)	0,2098	0,0010	0000	1001	1000
	+	+			
	0,3729	0,0011	0111	0010	1001
	<u>0,5827</u>	0,0101	1000	1100	0001
		+			
		0000	0000	0110	0110
		0,0101	1000	0010	0111

в)	0,3721	0,0011	0111	0010	0001
	+	+			
	0,5683	0,0101	0110	1000	0011
	<u>0,9404</u>	0,1001	1110	1010	0100
		+			
		0000	0110	0110	0000
		0,1001	0100	0000	0100

Глава 2. КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

2.1. КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА НА ИНТЕГРАЛЬНЫХ ЛОГИЧЕСКИХ МИКРОСХЕМАХ

В данный раздел включены задачи и упражнения, связанные с анализом и синтезом комбинационных цифровых устройств на интегральных логических микросхемах [2, 3, 21].

При решении задач используются следующие соотношения булевой алгебры:

$$\begin{array}{lll} x \vee 0 = x, & x \cdot 0 = 0, & \bar{0} = 1, \\ x \vee 1 = 1, & x \cdot 1 = x, & \bar{1} = 0, \\ x \vee x = x, & x \cdot x = x, & \bar{\bar{x}} = x, \\ x \vee \bar{x} = 1, & x \cdot \bar{x} = 0, & \end{array} \quad \begin{array}{l} (2.1) \\ (2.2) \end{array} \quad (2.3)$$

Основные законы:

переместительный закон

$$x \vee y = y \vee x, \quad xy = yx; \quad (2.4)$$

сочетательный закон

$$\begin{aligned} x \vee y \vee z &= (x \vee y) \vee z = x \vee (y \vee z); \\ xy \cdot z &= x(yz) = (xy)z; \end{aligned} \quad (2.5)$$

распределительный закон

$$\begin{aligned} x(y \vee z) &= xy \vee xz; \\ x \vee yz &= (x \vee y)(x \vee z); \end{aligned} \quad (2.6)$$

закон поглощения

$$x \vee xy = x, \quad x(x \vee y) = x; \quad (2.7)$$

закон склеивания

$$xy \vee \bar{x}y = y; \quad (x \vee y)(\bar{x} \vee y) = y. \quad (2.8)$$

Правило де Моргана

$$\overline{x \vee y} = \bar{x} \bar{y}, \quad \overline{xy} = \bar{x} \vee \bar{y}. \quad (2.9)$$

Некоторые полезные соотношения:

$$(x \vee \bar{y})y = xy, \quad (2.10)$$

$$x\bar{y} \vee y = x \vee y, \quad (2.11)$$

$$x\bar{y} \vee \bar{x}y = \bar{x}y(x \vee y), \quad (2.12)$$

$$x\bar{y} \vee \bar{x}z = \bar{x}y(x \vee z). \quad (2.13)$$

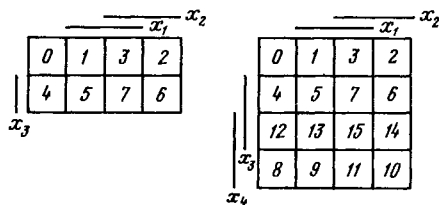


Рис. 2.1. Карты Карно для трех и четырех переменных

Минимизацию булевых функций в большинстве случаев целесообразно осуществлять при помощи карт Карно (рис. 2.1). Далее буквой Φ на картах обозначается набор, на котором функция не определена (факультативное условие).

ЗАДАЧИ И УПРАЖНЕНИЯ

2.1. Записать структурную формулу, которая реализуется комбинационной схемой, приведенной на рис. 2.2. Упростить полученную структурную формулу и построить новую схему на элементах И, ИЛИ и НЕ.

2.2. Упростить схему комбинационного устройства, приведенную на рис. 2.3.

2.3. Синтезировать в базисе И, ИЛИ, НЕ устройство, сигнал на выходе которого равен 1 только в том случае, когда на его двух входах (x_1 и x_2) действуют различные сигналы (узел неравнозначности, сумматор по модулю два).

2.4. Решить задачу 2.3, используя только элементы:

а) И-НЕ: б) ИЛИ-НЕ.

2.5. На вход сумматора по модулю два поступает серия импульсов, приведенная на рис. 2.4. Построить временную диаграмму выходного сигнала $y(t)$ без учета задержек и фронтов, создаваемых логическими элементами сумматора.

А В С D

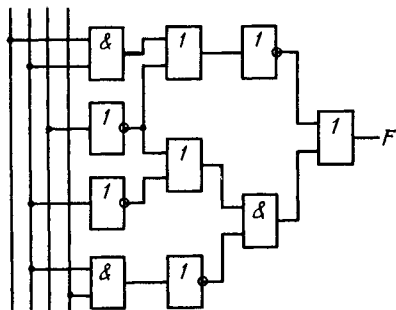


Рис. 2.2. Схема устройства к задаче 2.1

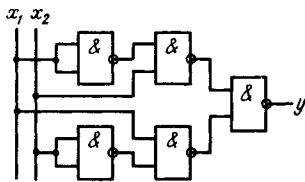


Рис. 2.3. Схема устройства к задаче 2.2

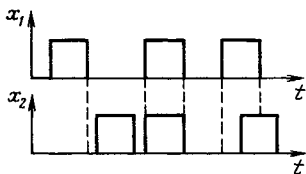


Рис. 2.4. Временные диаграммы к задаче 2.5

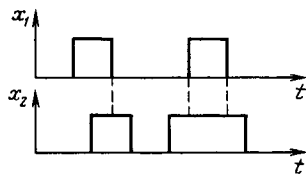


Рис. 2.5. Временные диаграммы к задаче 2.7

2.6. Синтезировать в базисе И, ИЛИ, НЕ устройство, сигнал на выходе которого равен 1, только в том случае, когда на его двух входах (x_1 , x_2) действуют одинаковые сигналы (узел равнозначности).

2.7. На вход узла равнозначности подается серия импульсов, приведенная на рис. 2.5. Построить временную диаграмму выходного сигнала без учета задержек и фронтов, создаваемых логическими элементами устройства.

2.8. Устройство с четырьмя входами (x_1 , x_2 , x_3 , x_4) должно работать так, чтобы на выходе появлялся сигнал 1, когда не менее чем на трех входах будут одновременно сигналы 1. Синтезировать устройство на элементах И, ИЛИ, НЕ.

2.9. Устройство с четырьмя входами (x_1 , x_2 , x_3 , x_4) должно работать так, чтобы на выходе появлялся сигнал 1, когда на трех входах будут одновременно сигналы 1. На всех четырех входах сигнал 1 никогда не появляется. Синтезировать устройство на элементах И, ИЛИ, НЕ.

2.10. Синтезировать мажоритарный элемент на три входа (x_1 , x_2 , x_3): а) в базисе И-НЕ; б) в базисе ИЛИ-НЕ. У такого элемента значение выходного сигнала совпадает с значением большинства входных.

2.11. Синтезировать устройство в базисе ИЛИ-НЕ, булева функция которого задана в числовом виде:

$$y(x_1, x_2, x_3, x_4) = \sum_{\text{обязат.}} (1, 4, 8, 10, 11, 14) + \sum_{\text{факульт}} (0, 2, 7, 12, 13).$$

2.12. Синтезировать устройство в базисе И-НЕ, булева функция которого задана в числовом виде:

$$y(x_1, x_2, x_3, x_4) = \sum (0, 1, 2, 3, 5, 7, 8, 9, 10, 11, 15).$$

2.13. Синтезировать полный дешифратор на три входа на элементах И, ИЛИ, НЕ. Как необходимо изменить схему дешифратора, чтобы превратить его в дешифратор-демультиплексор (стробируемый дешифратор)?

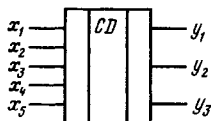


Рис. 2.6. Шифратор на пять входов

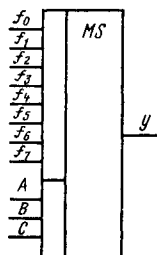


Рис. 2.7. Мультиплексор на восемь каналов

2.14. Как из дешифраторов-демультиплексоров типа ИДЗ серии 155 построить полный дешифратор на пять входов (см. приложение)?

2.15. Синтезировать шифратор на пять входов (рис. 2.6): а) на элементах ИЛИ-НЕ; б) на элементах И-НЕ.

2.16. Синтезировать преобразователь позиционного десятичного кода в двоично-десятичный на элементах И-НЕ.

2.17. Синтезировать мультиплексор на восемь каналов на элементах И, ИЛИ, НЕ (рис. 2.7).

2.18. Составить схему устройства для сравнения двух 5-разрядных двоичных чисел, используя только двухвходовые сумматоры по модулю два и ячейку ИЛИ. (При равенстве чисел выходной сигнал 0.)

2.19. Синтезировать узел, осуществляющий суммирование двух 1-разрядных двоичных чисел (полусумматор), на элементах И, ИЛИ, НЕ и на элементах И-НЕ (рис. 2.8).

2.20. На вход полусумматора поступает серия импульсов, приведенная на рис. 2.9. Построить временную диаграмму выходных сигналов без учета задержек, создаваемых логическими элементами полусумматора.

2.21. Составить схему полного сумматора (рис. 2.10), используя полусумматоры.

2.22. Синтезировать полный сумматор на элементах И-ИЛИ-НЕ.

2.23. Построить полный сумматор на мультиплексорах К155КП7 (рис. 2.11). Таблица истинности для мультиплексора приведена в решении задачи Р2.17.

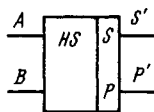


Рис. 2.8. Полусумматор

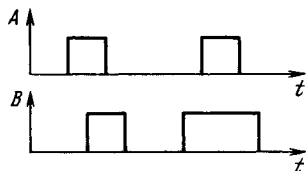


Рис. 2.9. Временные диаграммы к задаче 2.20

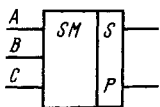


Рис. 2.10. Полный сумматор

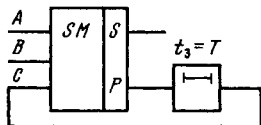


Рис. 2.12. Сумматор последовательного действия

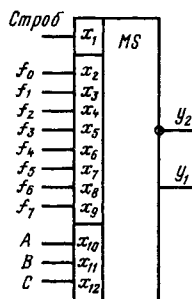


Рис. 2.11. Мультиплексор K155КП7

2.24. На вход сумматора последовательного действия (рис. 2.12) поступают коды двух чисел: $A=111(7)$, $B=001(1)$. Построить временные диаграммы входных и выходных сигналов.

2.25. Начертить схему сумматора параллельного действия с последовательным переносом для сложения двух 4-разрядных двоичных чисел, используя полные сумматоры. Ответить на вопросы:

а) каково максимальное время выполнения операции, если время задержки одного сумматора $t_{3\text{макс}}=100$ нс;

б) можно ли снимать информацию с суммирующего устройства в моменты времени $t < t_{3\text{макс}}$ после подачи входных сигналов;

в) для каких целей может использоваться третий вход первого сумматора;

г) как можно ускорить выполнение операции сложения?

2.2. ЦИФРОВЫЕ УСТРОЙСТВА НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ МАТРИЦАХ

В данный раздел включены задачи и упражнения, связанные с построением комбинационных цифровых устройств на программируемых логических матрицах (ПЛИМ) [15]. Для получения однозначного ответа следует ориентироваться на матрицы, приведенные в данном параграфе. На рис. 2.13 приведена упрощенная схема двухуровневой ПЛИМ.

На уровне I выполняется операция И над переменными x и их инверсиями. (Инверторы на схеме обозначены кружками.) На уровне II выполняется операция ИЛИ над переменными p , являющимися выходными сигналами первого уровня матрицы. Крестиками на шинах обозначены элементы связи, которые включаются во время программирования путем прожигания либо во время изготовления путем наложения маски.

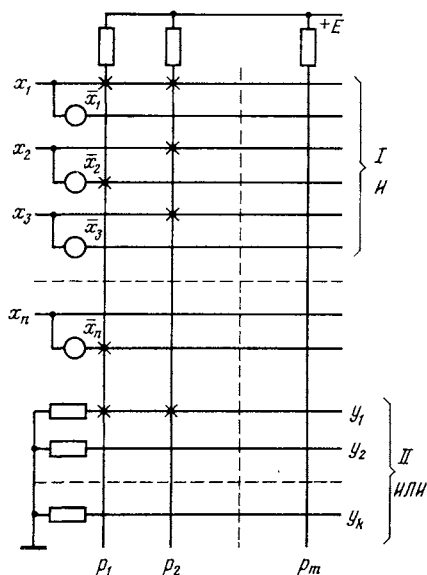


Рис. 2.13. Двухуровневая ПЛМ

На рис. 2.14 приведена упрощенная схема программируемой матрицы вентилей (ПМВ). Резисторы на схеме не показаны. Элементы связи включаются так же, как и в двухуровневой ПЛМ. Кругами, как и раньше, обозначены инверторы.

В данном примере (см. рис. 2.14)

$$y_3 = \overline{x_1 \bar{x}_2 \bar{x}_3} = \bar{x}_1 \vee x_2 \vee x_3, \quad y_4 = \bar{x}_1 x_2 x_3.$$

Очевидно, что матрица вентилей не позволяет получать все возможные функции от переменных x .

При решении задач на схемах необходимо показывать только используемые шины и их соединение элементами связи. После этого можно выбрать матрицу из имеющихся в распоряжении проектировщика.

ЗАДАЧИ И УПРАЖНЕНИЯ

2.26. Построить шифратор на пять входов на ПМВ.

2.27. Построить преобразователь позиционного десятичного кода в двоично-десятичный на ПМВ.

2.28. Решить задачу 2.27, используя двухуровневую ПЛМ.

2.29. Построить полный дешифратор на три входа на ПМВ.

2.30. Построить мультиплексор на восемь каналов на двухуровневой ПЛМ.

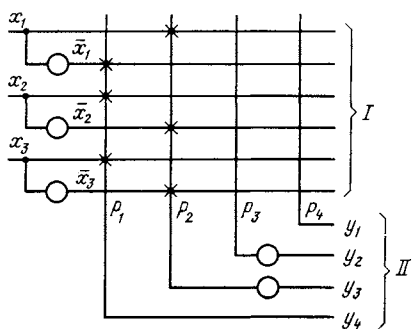


Рис. 2.14. Программируемая матрица вентилей

В рассматриваемом примере (рис. 2.13)

$$p_1 = x_1 \bar{x}_2 \bar{x}_n, \quad p_2 = x_1 x_2 x_3, \quad y_1 = p_1 \vee p_2 = x_1 \bar{x}_2 \bar{x}_n \vee x_1 x_2 x_3.$$

При решении задач резисторы на схемах можно не изображать.

2.31. Построить устройство для сравнения двух 3-разрядных чисел на двухуровневой ПЛМ. При неравенстве чисел на выходе устройства должен появляться сигнал нуль.

2.32. Построить полный сумматор на двухуровневой ПЛМ.

ОТВЕТЫ И РЕШЕНИЯ

P2.1. $F = \overline{AB \vee \bar{C} \vee BD} (\bar{C} \vee \bar{B}) = \overline{AB \vee \bar{C} \vee CB BD} = \overline{AB \vee \bar{C} \vee CB \vee BD} = (\overline{AB \vee \bar{C}}) \overline{B(C \vee D)} = \overline{B(A \vee \bar{C})} (\overline{C \vee D})$. Схема устройства приведена на рис. P2.1. Число входов у элементов исходной схемы $n_1 = 16$. Число входов у полученной схемы $n_2 = 9$.

P2.2. Для приведенной схемы (см. рис. 2.3) $y = \overline{\overline{\overline{x_1 x_2} \bar{x}_2 x_1}}$.

Упрощение: $y = x_1 x_2 x_2 x_1 x_2 x_1$.

Схема устройства приведена на рис. P2.2.

P2.3. Составляем таблицу истинности в соответствии со словесным описанием работы устройства (табл. P2.1);

записываем структурные формулы на основе единичных наборов:

$$y = x_1 \bar{x}_2 \vee \bar{x}_1 x_2 = x_1 \oplus x_2$$

Таблица P2.1

x_2	x_1	y
0	0	0
0	1	1
1	0	1
1	1	0

и на основе нулевых наборов [2]:

$$y = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2) = (x_1 \vee x_2)(\overline{x_1 x_2}).$$

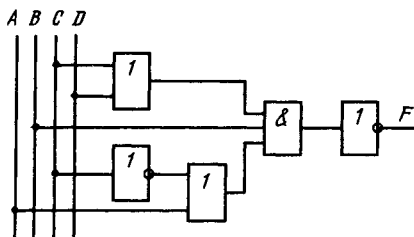


Рис. P2.1. Схема устройства к задаче 2.1

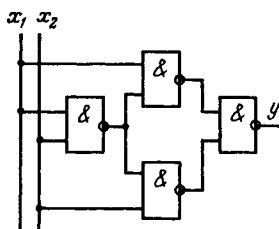


Рис. P2.2. Схема устройства к задаче 2.2

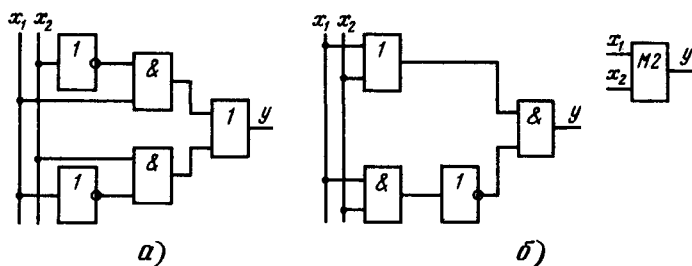


Рис. P2.3. Схемы устройств к задаче 2.3

Варианты схем устройства приведены на рис. P2.3.

P2.4. а) $y = x_1 \bar{x}_2 \vee \bar{x}_1 x_2 = \overline{x_1 \bar{x}_2 \vee \bar{x}_1 x_2} = \overline{x_1 \bar{x}_2} \bar{x}_1 x_2 = \overline{x_1 \bar{x}_2} \bar{x}_1 x_2 = \overline{x_1 \bar{x}_2} \bar{x}_1 x_2 = \overline{x_1 \bar{x}_2} \bar{x}_1 x_2$.
Схема устройства приведена на рис. P2.2;

б) $y = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2) = \overline{(x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2)} = \overline{x_1 \vee x_2} \vee \bar{x}_1 \vee \bar{x}_2$.
Схема устройства приведена на рис. P2.4.

P2.5. Временные диаграммы для схемы на рис. P2.3 приведены на рис. P2.5.

P2.6. Составляем таблицу истинности в соответствии со словесным описанием работы устройства (табл. P2.2). Записываем структурные формулы на основе единичных и нулевых наборов:

$$y = \bar{x}_1 \bar{x}_2 \vee x_1 x_2 = \overline{\bar{x}_1 x_2 \vee x_1 \bar{x}_2} = \overline{x_1 \oplus x_2}, \quad y = (x_2 \vee \bar{x}_1)(\bar{x}_2 \vee x_1).$$

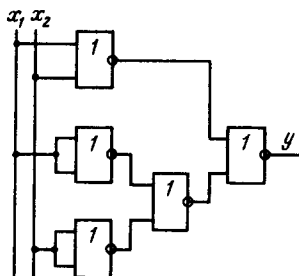


Рис. P2.4. Схема устройства к задаче 2.4

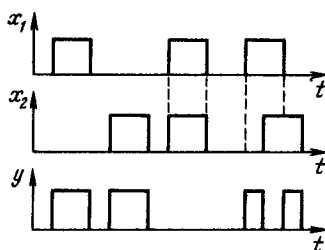


Рис. P2.5. Временные диаграммы к задаче 2.5

Таблица P2.2		
x_2	x_1	y
0	0	1
0	1	0
1	0	0
1	1	1

P2.7. Временные диаграммы приведены на рис. P2.6.

P2.8. Составляем таблицу истинности в соответствии со словесным описанием работы устройства (табл. P2.3). Наносим функцию на карту Карно (рис. P2.7) и осуществляем минимизацию:

$$y = x_1 x_2 (x_3 \vee x_4) \vee x_3 x_4 (x_1 \vee x_2).$$

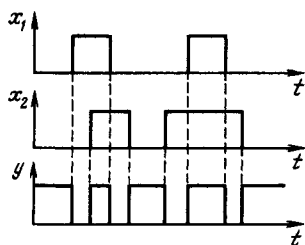


Рис. P2.6. Временные диаграммы к задаче 2.7

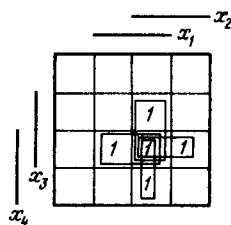


Рис. P2.7. Карта Карно к задаче 2.8

Таблица P2.3

Номер набора	x_4	x_3	x_2	x_1	y	Номер набора	x_4	x_3	x_2	x_1	y
0	0	0	0	0	0	8	1	0	0	0	0
1	0	0	0	1	0	9	1	0	0	1	0
2	0	0	1	0	0	10	1	0	1	0	0
3	0	0	1	1	0	11	1	0	1	1	1
4	0	1	0	0	0	12	1	1	0	0	0
5	0	1	0	1	0	13	1	1	0	1	1
6	0	1	1	0	0	14	1	1	1	0	1
7	0	1	1	1	1	15	1	1	1	1	1

P2.9. Задача 2.9 отличается от задачи 2.8 только тем, что на 15-м наборе функция не определена. Поэтому на карте Карно (рис. P2.7) на 15-м наборе вместо 1 ставится буква Φ . При минимизации целесообразно считать $\Phi=1$. При этом выражение для y оказывается таким же, как и в P2.8.

P2.10. Составляем таблицу истинности в соответствии со словесным описанием работы элемента (табл. P2.4). Наносим функцию на карту Карно (рис. P2.8) и осуществляем минимизацию:

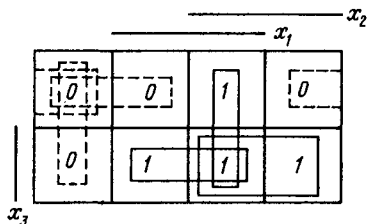


Рис. P2.8. Карта Карно к задаче 2.10

Таблица P2.4

Номер набора	x_3	x_2	x_1	y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$$\begin{aligned}
 а) \quad y &= x_1 x_2 \vee x_2 x_3 \vee x_1 x_3 = \overline{x_1 x_2 \vee x_2 x_3 \vee x_1 x_3} = \overline{x_1 x_2} \cdot \overline{x_2 x_3} \cdot \overline{x_1 x_3}; \\
 б) \quad y &= (x_1 \vee x_2)(x_2 \vee x_3)(x_1 \vee x_3) = \overline{x_1 \vee x_2} \vee \overline{x_2 \vee x_3} \vee \overline{x_1 \vee x_3}.
 \end{aligned}$$

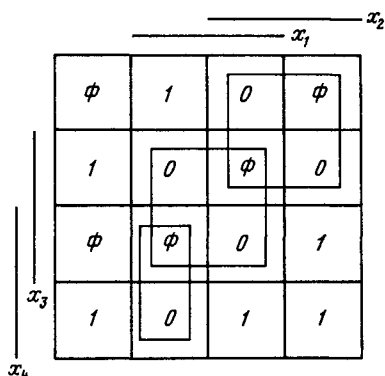


Рис. P2.9. Карта Карно к задаче 2.11

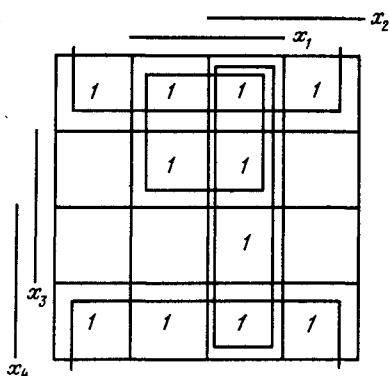


Рис. P2.10. Карта Карно к задаче 2.12

P2.11. Наносим функцию на карту Карно (рис. P2.9) и осуществляем минимизацию:

$$\begin{aligned}
 y &= (\bar{x}_2 \vee x_4)(\bar{x}_1 \vee \bar{x}_3)(\bar{x}_1 \vee x_2 \vee \bar{x}_4) = \\
 &= \overline{(\bar{x}_2 \vee x_4) \vee (\bar{x}_1 \vee \bar{x}_3) \vee (\bar{x}_1 \vee x_2 \vee \bar{x}_4)} = \\
 &= \overline{x_2 \vee x_4 \vee x_4 \vee \bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_1 \vee x_2 \vee x_2 \vee x_4} \\
 &= (x \vee \bar{y} = x \vee x \vee y).
 \end{aligned}$$

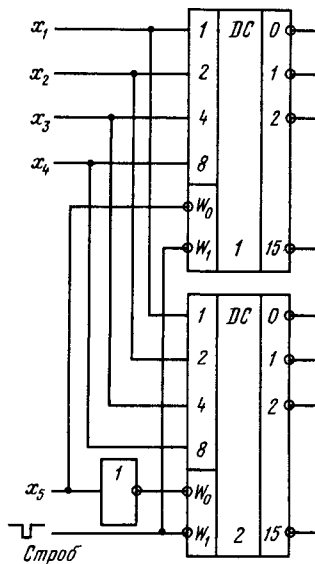


Рис. P2.11. Схема соединения дешифраторов к задаче 2.14

P2.12. Наносим функцию на карту Карно (рис. P2.10) и осуществляем минимизацию:

$$\begin{aligned}
 y &= \bar{x}_3 \vee x_1 x_2 \vee x_1 \bar{x}_4 = \overline{\bar{x}_3 \vee x_1 x_2 \vee x_1 \bar{x}_4} \\
 &= x_3 x_1 x_2 \vee x_1 \bar{x}_4, \\
 &\text{или} \quad y = \bar{x}_3 \vee x_1 (x_2 \vee \bar{x}_4) = \\
 &= x_3 x_1 (x_2 \vee \bar{x}_4) = x_3 x_1 \bar{x}_2 x_4.
 \end{aligned}$$

P2.13. Составляем таблицу истинности (табл. P2.5) и переходим к структурным формулам:

$$\begin{aligned}
 y_1 &= \bar{x}_1 \bar{x}_2 \bar{x}_3, & y_5 &= \bar{x}_1 \bar{x}_2 x_3, \\
 y_2 &= x_1 \bar{x}_2 \bar{x}_3, & y_6 &= x_1 \bar{x}_2 x_3, \\
 y_3 &= \bar{x}_1 x_2 \bar{x}_3, & y_7 &= \bar{x}_1 x_2 x_3, \\
 y_4 &= x_1 x_2 \bar{x}_3, & y_8 &= x_1 x_2 x_3.
 \end{aligned}$$

Для получения стробируемого дешифратора необходимо использовать элементы И на четыре входа. Четвертый вход всех элементов И следу-

Таблица P2.5

x_3	x_2	x_1	y_8	y_7	y_6	y_5	y_4	y_3	y_2	y_1
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

ет использовать для подачи на него стробирующего импульса.

P2.14. Схема соединений дешифраторов приведена на рис. P2.11. При $x_5=0$ отключен дешифратор 2. На его выходах — сигнал 1. При $x_5=1$ отключен дешифратор 1.

P2.15. Составляем таблицу истинности (табл. P2.6) и переходим к структурным формулам.

Таблица P2.6

x_5	x_4	x_3	x_2	x_1	y_3	y_2	y_1
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	0
1	0	0	0	0	1	0	1

С учетом факультативных условий получаем

$$y_1 = x_1 \vee x_3 \vee x_5 = \overline{\bar{x}_1 \bar{x}_3 \bar{x}_5},$$

$$y_2 = x_2 \vee x_3 = \overline{\bar{x}_2 \bar{x}_3},$$

$$y_3 = x_4 \vee x_5 = \overline{\bar{x}_4 \bar{x}_5}.$$

P2.16. Составляем таблицу истинности (табл. P2.7) и переходим к структурным формулам.

Таблица P2.7

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_4	y_3	y_2	y_1
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0	1
									0	0	1	0
									0	0	1	1
									0	1	0	1
									0	1	0	1
									0	1	1	0
									0	1	1	1
									1	0	0	0
									1	0	0	1

С учетом факультативных условий получим

$$y_1 = x_1 \vee x_3 \vee x_5 \vee x_7 \vee x_9 = \overline{\bar{x}_1 \bar{x}_3 \bar{x}_5 \bar{x}_7 \bar{x}_9},$$

$$y_2 = x_2 \vee x_3 \vee x_6 \vee x_7 = \overline{\bar{x}_2 \bar{x}_3 \bar{x}_6 \bar{x}_7},$$

$$y_3 = x_4 \vee x_5 \vee x_6 \vee x_7 = \overline{\bar{x}_4 \bar{x}_5 \bar{x}_6 \bar{x}_7},$$

$$y_4 = x_8 \vee x_9 = \overline{\bar{x}_8 \bar{x}_9}.$$

P2.17. Составляем таблицу истинности (табл. P2.8) и переходим к структурной формуле

Таблица P2.8

<i>C</i>	<i>B</i>	<i>A</i>	<i>y</i>
0	0	0	f_0
0	0	1	f_1
0	1	0	f_2
0	1	1	f_3
1	0	0	f_4
1	0	1	f_5
1	1	0	f_6
1	1	1	f_7

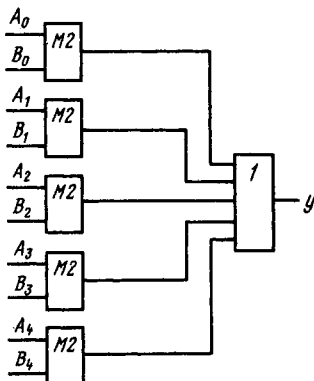


Рис. P2.12. Схема устройства для сравнения двух 5-разрядных двоичных чисел

$$y = f_0 \bar{A} \bar{B} \bar{C} \vee f_1 A \bar{B} \bar{C} \vee f_2 \bar{A} B \bar{C} \vee f_3 A B \bar{C} \vee f_4 \bar{A} \bar{B} C \vee f_5 A \bar{B} C \vee f_6 \bar{A} B C \vee f_7 A B C.$$

P2.18. Схема устройства приведена на рис. P2.12.

P2.19. Составляем таблицу истинности (табл. P2.9) и переходим к структурным формулам

Таблица P2.9

<i>B</i>	<i>A</i>	<i>S'</i>	<i>p'</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S' = A \bar{B} \vee \bar{A} B = A \oplus B, \quad p' = AB,$$

$$S' = (A \vee B)(\bar{A} \vee \bar{B}) = (A \vee B) \overline{AB}.$$

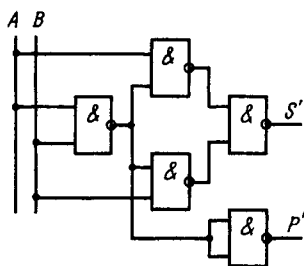


Рис. P2.13. Полусумматор на элементах И-НЕ

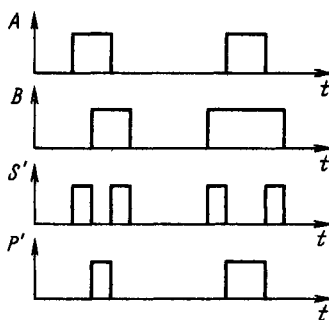


Рис. P2.14. Временные диаграммы к задаче 2.20

Переходим к базису И-НЕ:

$$S' = A\bar{B} \vee \bar{A}B = \overline{\overline{A\bar{B}} \cdot \overline{\bar{A}B}} = \overline{\overline{A} \cdot \overline{\bar{B}}} \cdot \overline{\overline{\bar{A}} \cdot \overline{B}} = \overline{\overline{A} \cdot \overline{\bar{B}}} \cdot \overline{\overline{\bar{A}} \cdot \overline{B}} = A \bar{B} \bar{A} B B.$$

Схема полусумматора на элементах И-НЕ приведена на рис. P2.13.

P2.20. Временные диаграммы выходных сигналов приведены на рис. P2.14.

P2.21. Схема полного сумматора на полусумматорах приведена на рис. P2.15. После суммирования двух 1-разрядных двоичных чисел (A, B) при помощи полусумматора к сумме прибавляется третье число. Так как при сложении трех 1-разрядных двоичных чисел может образовываться перенос только в одном полусумматоре, то переносы объединяются ячейкой ИЛИ.

P2.22. Составляем таблицу истинности (табл. P2.10) и переходим к структурным формулам:

Таблица P2.10

C	B	A	S	P
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

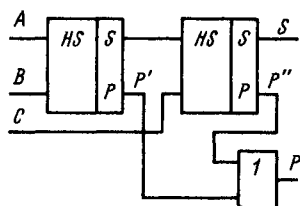


Рис. P2.15. Полный сумматор на полусумматорах

$$S = A\bar{B}\bar{C} \vee \bar{A}B\bar{C} \vee \bar{A}\bar{B}C \vee ABC,$$

$$P = A\bar{B}C \vee \bar{A}BC \vee \bar{A}\bar{B}C \vee ABC = AB \vee AC \vee BC.$$

Преобразуем выражение для S :

$$S = ABC \vee F,$$

где $F = A\bar{B}\bar{C} \vee \bar{A}B\bar{C} \vee \bar{A}\bar{B}C$;
 $\vee ABC^1 = \bar{A}\bar{B}\bar{C} \vee P$. $\bar{F} = \bar{A}\bar{B}\bar{C} \vee \bar{A}BC \vee A\bar{B}C \vee ABC \vee$

Следовательно, $F = \overline{A}\overline{B}\overline{C} \vee P$ и

$$S = ABC \vee \overline{A}\overline{B}\overline{C} \vee P = ABC \vee \overline{A}\overline{B}\overline{C}\overline{P} = ABC \vee (A \vee B \vee C)\overline{p},$$

где $p = AB \vee AC \vee BC$.

Схема полного сумматора на элементах И-ИЛИ-НЕ приведена на рис. P2.16.

P2.23. Мультиплексоры используем в качестве универсальных логических модулей: один для получения суммы, другой для получения переноса. Входы A, B, C используем в качестве информационных входов, а входы $f_0, f_1, \dots, f_7$ — в качестве настроечных (см. табл. P2.11).

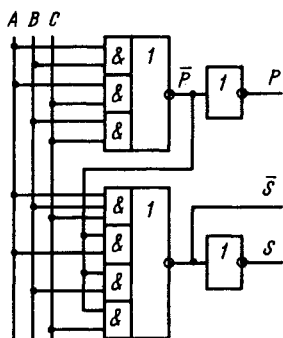


Таблица Р2.11

C	B	A	y	S	P
0	0	0	f_0	0	0
0	0	1	f_1	1	0
0	1	0	f_2	1	0
0	1	1	f_3	0	1
1	0	0	f_4	1	0
1	0	1	f_5	0	1
1	1	0	f_6	0	1
1	1	1	f_7	1	1

Рис. Р2.16. Полный сумматор на элементах И-ИЛИ-НЕ

Из табл. P2.11 следует, что для получения суммы необходимо на входы f_1, f_2, f_4 и f_7 первого мультиплексора подать сигнал 1, а на остальные 0 (заземлить). Для получения переноса необходимо подать сигнал 1 на входы f_3, f_5, f_6 и f_7 второго мультиплексора, а остальные заземлить.

P2.24. Временные диаграммы входных и выходных сигналов приведены на рис. P2.17.

P2.25. Схема сумматора приведена на рис. P2.18:

- а) $t_{3, \max} = nt_{31} = 400$ нс;
 б) нельзя, так как процесс не установится из-за задержек, создаваемых сумматорами, и ответ будет неправильным;
 в) третий вход может быть использован для подачи сигнала

¹ Логическая сумма всех наборов тождественно равна 1. Так как $F \vee \overline{F} = 1$, то члены, не вошедшие в F , дают $\overline{F}$.

Рис. Р2.17. Временные диаграммы к задаче 2.24

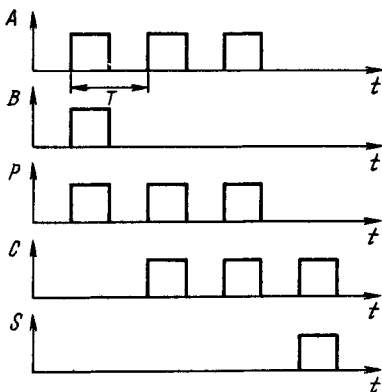
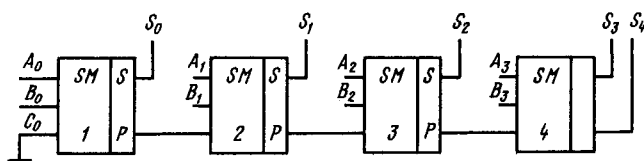


Рис. Р2.18. Сумматор параллельного действия с последовательным переносом



циклического переноса при выполнении операции вычитания в обратном коде. Этот вход может быть использован для прибавления единицы при образовании дополнительного кода отрицательного числа;

г) путем использования схем

ускоренного переноса.

Р2.26. Структурные формулы для шифратора на пять входов получены в Р2.15. Схема шифратора приведена на рис. Р2.19.

Р2.27. Структурные формулы для преобразователя кода получены в Р2.16.

Схема преобразователя кода приведена на рис. Р2.20.

Р2.28. Структурные формулы для преобразователя:

$$\begin{aligned} y_1 &= x_1 \vee x_3 \vee x_5 \vee x_7 \vee x_9, \\ y_2 &= x_2 \vee x_3 \vee x_6 \vee x_7, \\ y_3 &= x_4 \vee x_5 \vee x_6 \vee x_7, \\ y_4 &= x_8 \vee x_9. \end{aligned}$$

Схема преобразователя кода приведена на рис. Р2.21.

Р2.29. Структурные формулы для дешифратора на три входа

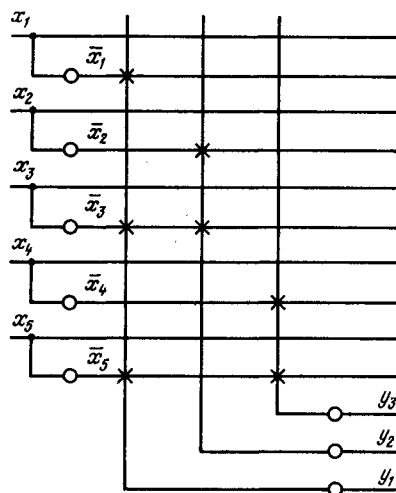


Рис. Р2.19. Шифратор на программируемой матрице вентилей

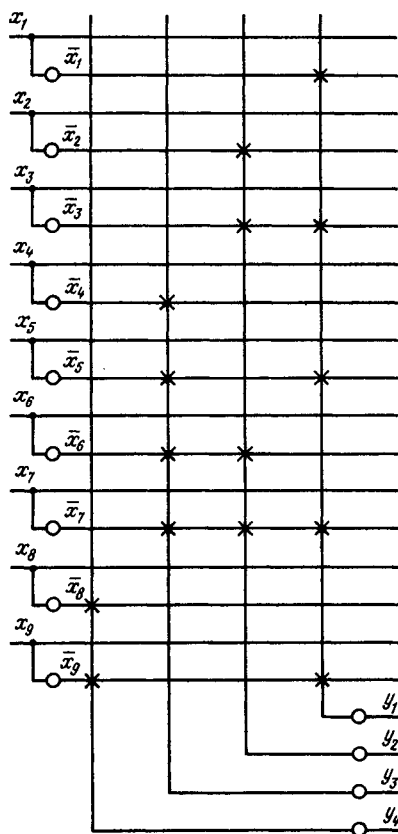


Рис. Р2.20. Преобразователь позиционного десятичного кода в двоично-десятичный на программируемой матрице вентилей

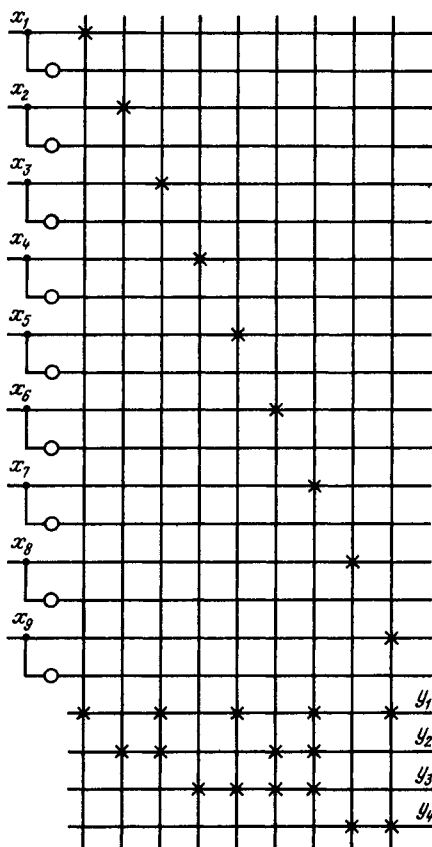


Рис. Р2.21. Преобразователь кода на двухуровневой ПЛМ

получены в Р2.13. Схема дешифратора на три входа приведена на рис. Р2.22.

Р2.30. Структурные формулы для мультиплексора получены в Р2.17. Принципиальная схема мультиплексора на ПЛМ приведена на рис. Р2.23.

Р2.31. Для построения устройства используем как основу схему на рис. Р2.24:

$$y_1 = \bar{A}_0 B_0 \vee A_0 \bar{B}_0, \quad y_2 = \bar{A}_1 B_1 \vee A_1 \bar{B}_1, \\ y_3 = A_2 B_2 \vee A_2 \bar{B}_2.$$

Схема устройства на ПЛМ для сравнения чисел приведена на рис. Р2.25.

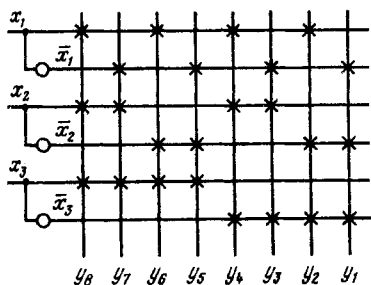


Рис. P2.22. Дешифратор на ПМВ

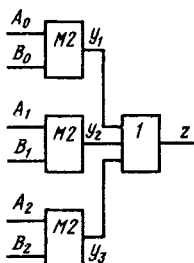


Рис. P2.24. Схема устройства к задаче 2.31

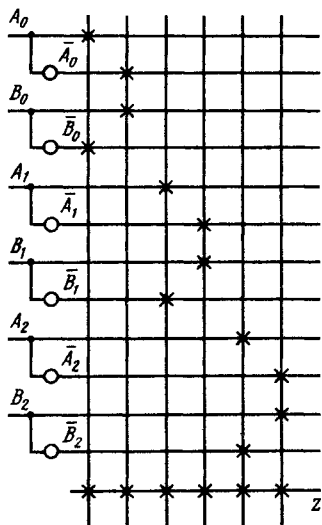


Рис. P2.25. Сравнивающее устройство на двухуровневой ПЛМ

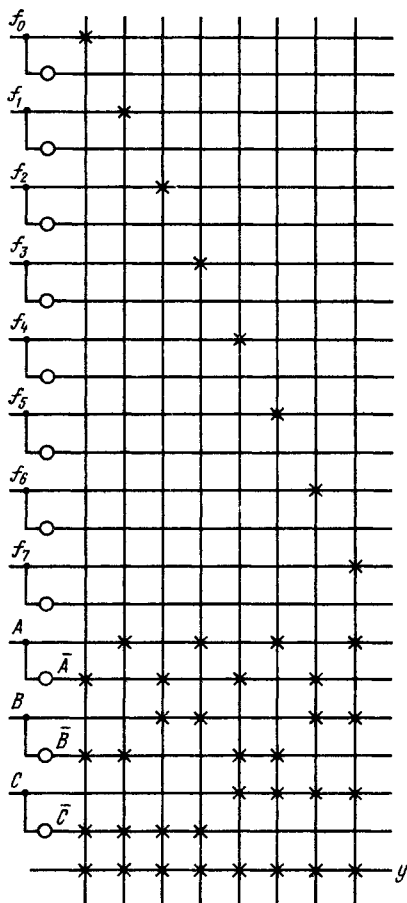


Рис. P2.23. Мультиплексор на двухуровневой ПЛМ

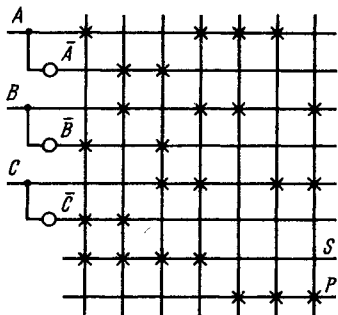


Рис. P2.26. Полный сумматор на двухуровневой ПЛМ

P2.32. Структурные формулы для полного сумматора:

$$S = A\bar{B}\bar{C} \vee \bar{A}B\bar{C} \vee \bar{A}\bar{B}C \vee ABC,$$

$$P = AB \vee AC \vee BC.$$

Схема сумматора приведена на рис. P2.26.

Глава 3. ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

3.1. ТРИГГЕРЫ

В данный раздел включены задачи и упражнения, связанные с анализом работы различных типов триггеров [2—4].

Для получения однозначного ответа следует ориентироваться на микросхемы ТТЛ-типа (см. приложение 1).

При построении временных диаграмм можно не учитывать там, где это специально не оговорено, фронты входных сигналов и задержки, создаваемые элементами триггера.

ЗАДАЧИ И УПРАЖНЕНИЯ

3.1. На рис. 3.1 приведена схема асинхронного *RS*-триггера на элементах И-НЕ и его условное обозначение:

- пояснить принцип работы триггера;
- составить таблицу переключений триггера;
- записать собственную функцию *RS*-триггера;
- построить временные диаграммы, поясняющие работу триггера, с учетом задержек, создаваемых его элементами ($t_{зср}$);

д) определить минимальную длительность управляющих импульсов $t_{\text{мин}}$, при которой триггер функционирует нормально, а также разрешающее время триггера $t_{\text{мин}}$;

е) на вход триггера поданы сигналы $S=R=0$ — запрещенная комбинация. Затем эти сигналы изменяются на $S=R=1$. В каком состоянии окажется триггер?

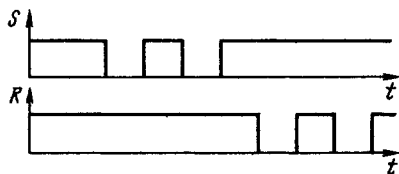
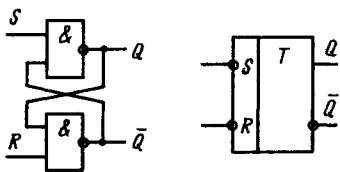


Рис. 3.1. Асинхронный *RS*-триггер на элементах И-НЕ

Рис. 3.2. Временные диаграммы к задаче 3.1

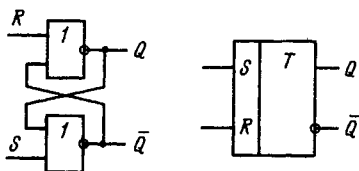


Рис. 3.3. Асинхронный RS-триггер на элементах ИЛИ-НЕ

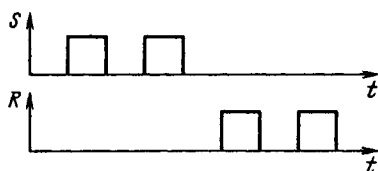
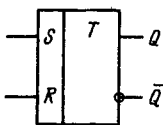


Рис. 3.4. Временные диаграммы к задаче 3.2

ж) начертить временные диаграммы выходных сигналов по временным диаграммам входных, приведенных на рис. 3.2. Исходное состояние триггера $Q=0$.

3.2. На рис. 3.3 приведена схема асинхронного RS-триггера на элементах ИЛИ-НЕ и его условное обозначение;

- пояснить принцип работы триггера;
- составить таблицу переключений триггера;
- записать собственную функцию RS-триггера;
- построить временные диаграммы, поясняющие работу триггера, с учетом задержек, создаваемых его элементами $t_{зср}$;
- определить минимальную длительность управляющих импульсов, при которой триггер функционирует нормально, а также разрешающее время триггера $t_{мин}$;

е) на вход триггера поданы сигналы $S=R=1$. Затем эти сигналы изменяются на $S=R=0$. В каком состоянии окажется триггер?

ж) начертить временные диаграммы выходных сигналов по временным диаграммам входных (рис. 3.4). Исходное состояние триггера $Q=0$.

3.3. Построить схему синхронного одноступенчатого RS-триггера на элементах И-НЕ. Составить таблицу переключений. Используя одноступенчатые синхронные RS-триггеры, начертить схему двухступенчатого RS-триггера. Пояснить различие в работе.

3.4. Построить одноступенчатый D-триггер, используя синхронный RS-триггер. Начертить временные диаграммы выходных сигналов Q и $\bar{Q}$ по временным диаграммам входных (рис. 3.5). В исходном состоянии $Q=0$. Чем будет отличаться работа двухступенчатого RS-триггера от работы одноступенчатого?

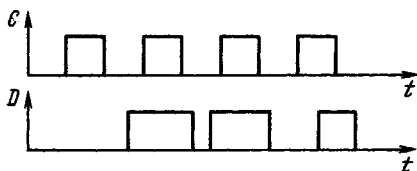


Рис. 3.5. Временные диаграммы к задаче 3.4

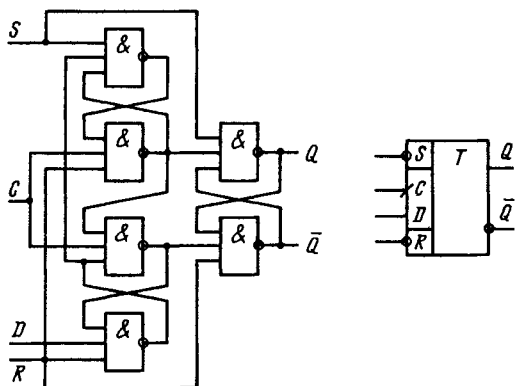


Рис. 3.6. D -триггер K155TM2

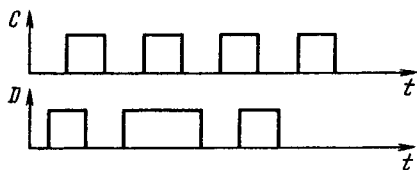


Рис. 3.7. Временные диаграммы к задаче 3.5

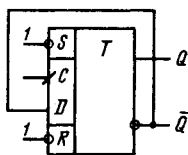


Рис. 3.8. Схема триггера к задаче 3.6

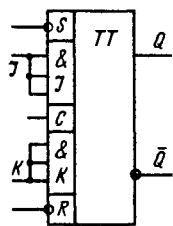


Рис. 3.9. JK -триггер K155TB1

3.5. На вход шестизадающего D -триггера типа TM2 (рис. 3.6) подаются сигналы, изображенные на рис. 3.7. Построить временные диаграммы выходных сигналов. В исходном состоянии $Q=0$.

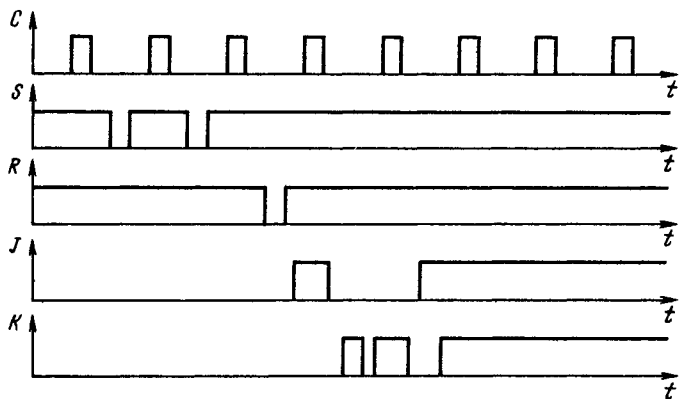


Рис. 3.10. Временные диаграммы к задаче 3.9

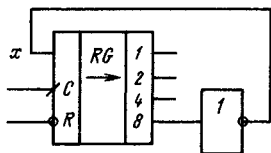


Рис. 3.12. Кольцевой регистр с перекрестной связью

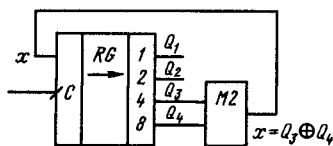


Рис. 3.13. Кольцевой регистр с логической обратной связью

а) пояснить принцип работы регистра на примере продвижения одной единицы;

б) построить временные диаграммы сигналов на выходах регистра Q при подаче шести продвигающих импульсов C и $X=1$. Исходное состояние регистра $Q_1=Q_2=Q_3=Q_4=0$;

в) определить максимальную задержку выходных сигналов относительно продвигающего импульса C ?

г) можно ли осуществлять запись новой информации при считывании старой в последовательном коде?

д) можно ли в регистре заменить триггеры K155TM2 триггерами K155TM7 или K155TM5?

е) можно ли в регистре использовать JK -триггеры?

3.13. Как можно использовать сдвигающий регистр (см. рис. 3.11) в качестве распределителя импульсов? (Сигнал единица должен появляться поочередно на каждом выходе.)

3.14. На рис. 3.12 приведена схема кольцевого регистра с перекрестной связью. Составить таблицу переключений, считая, что в исходном состоянии во все разряды записаны нули. Определить коэффициент пересчета (длину цикла) k .

3.15. На рис. 3.13 приведена схема кольцевого регистра с логической обратной связью. Составить таблицу переключений, считая, что в исходном состоянии единица записана в первый разряд регистра. Определить коэффициент пересчета (длину цикла) k .

3.16. В задаче 3.15 кольцевой регистр с логической обратной связью после сбоя, когда во всех разрядах оказываются нули, перестает работать. Как устранить этот недостаток?

3.17. Как использовать кольцевые регистры в качестве делителей числа импульсов? Какова задержка выходного сигнала относительно входного в этих делителях? Какова связь коэффициента деления с числом триггеров в регистре?

3.18. В качестве делителя числа импульсов используется два кольцевых 4-разрядных регистра, соединенных так, как показано на рис. 3.14. Определить коэффициент деления (пересчета).

3.19. В цифровом устройстве необходимо передавать информацию в параллельном коде от нескольких регистров одному приемнику по одной и той же многоразрядной шине. Ответить на вопросы:

а) можно ли к этой шине присоединить регистр, изображенный на рис. 3.11? Если нет, то почему?

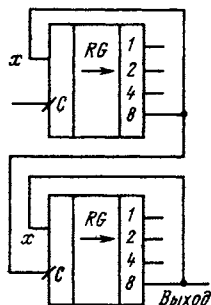


Рис. 3.14. Схема к задаче 3.18

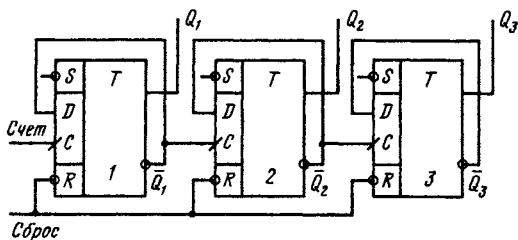


Рис. 3.15. Суммирующий двоичный счетчик на D -триггерах

б) можно ли к этой шине присоединить регистр ИР1, обладающий двумя состояниями выхода? Если нет, то почему?

в) как можно присоединить указанные регистры к общей шине?

3.20. На рис. 3.15 приведена схема суммирующего 3-разрядного двоичного счетчика на D -триггерах 155ТМ2:

а) пояснить принцип работы счетчика;

б) составить таблицу переключений счетчика. Определить коэффициент пересчета (модуль счета) k ;

в) построить временные диаграммы сигналов, снимаемых с выходов Q_1 , Q_2 и Q_3 ;

г) определить максимальную задержку, создаваемую счетчиком. Задержка, создаваемая одним триггером, $t_{зтр} = 55$ нс;

д) какие необходимо произвести изменения в схеме суммирующего счетчика (рис. 3.15), чтобы он стал вычитающим?

3.21. Можно ли в схеме счетчика на рис. 3.15 заменить триггеры ТМ2 триггерами ТМ5 или ТМ7?

3.22. Перевести схему суммирующего счетчика (рис. 3.15) на JK -триггеры 155ТВ1 (см. рис. 3.9). Построить временные диаграммы выходных сигналов для девяти входных импульсов.

3.23. Какие изменения необходимо произвести в схеме суммирующего счетчика, полученного в предыдущей задаче, чтобы он стал вычитающим?

3.24. Суммирующий счетчик (рис. 3.15) находится в нулевом состоянии. В каком состоянии он будет находиться после подачи 64 и 67 входных импульсов?

3.25. На рис. 3.16 приведена схема суммирующего двоичного счетчика с параллельным переносом на JK -триггерах 155ТВ1:

а) пояснить принцип работы счетчика;

б) составить таблицу переключений счетчика;

в) построить временные диаграммы сигналов, снимаемых с выходов Q_1 , Q_2 и Q_3 ;

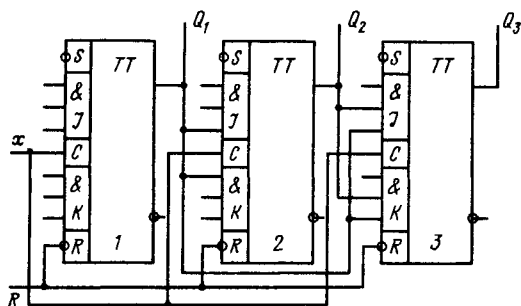


Рис. 3.16. Суммирующий двоичный счетчик с параллельным переносом на JK -триггерах с входной логикой И

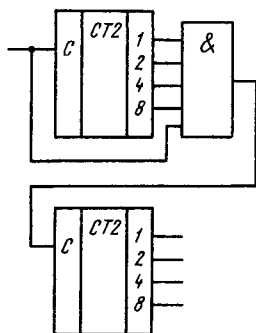


Рис. 3.17. Схема счетчика к задаче 3.27

г) определить задержку, создаваемую счетчиком $t_{\text{сч}}$. Задержка, создаваемая одним триггером, $t_{\text{тр}} = 40$ нс;

д) определить, какое максимальное число разрядов счетчика с параллельным переносом можно получить на триггерах ТВ1 без дополнительных элементов.

3.26. Построить 8-разрядный суммирующий счетчик на микросхемах К155ИЕ5 (см. приложение 1). Определить задержку, создаваемую счетчиком. Задержка, создаваемая одной микросхемой, $t_{\text{з1}} = 135$ нс. В микросхеме ИЕ5 (счетчик с последовательным переносом) выход первого триггера Q_1 не соединен со входом C_2 второго триггера.

3.27. Два счетчика с параллельным переносом соединены так, как показано на рис. 3.17. Задержка, создаваемая каждым счетчиком, $t_{\text{з1}} = 50$ нс. Задержка, создаваемая элементом И, $t_{\text{зр}} = 20$ нс. Определить задержку, создаваемую всем счетчиком, и его коэффициент пересчета (модуль счета).

3.28. Построить суммирующий счетчик с коэффициентом пересчета $k=10$ на микросхеме К155ИЕ5 (см. приложение 1 и задачу 3.26).

3.29. Построить суммирующий счетчик с коэффициентом пересчета $k=120$ на микросхемах К155ИЕ5 (см. приложение 1 и Р3.28).

3.30. Построить суммирующий счетчик на микросхеме К155ИЕ5 (см. Р3.28) с возможностью получения любого коэффициента пересчета от 2 до 15.

3.31. В схеме на рис. 3.17 отключили выход Q_4 от входа ячейки И. Каков будет при этом коэффициент деления счетчика, если в схеме используются ТТЛ-элементы?

3.32. На рис. 3.18 приведена схема распределителя импульсов, в которой используется счетчик К155ИЕ5 (см. задачу 3.26) и дешифратор-демультиплексор К155ИД3 (см. приложение 1);

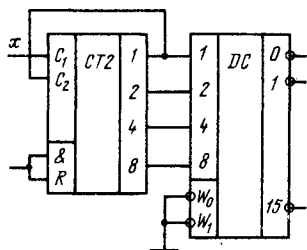


Рис. 3.18. Распределитель импульсов к задаче 3.32

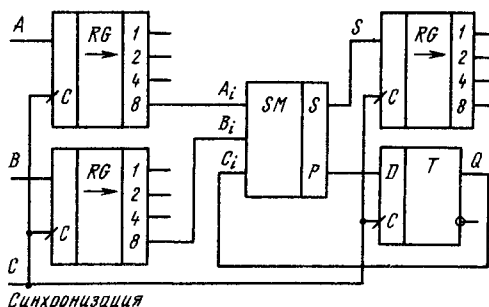


Рис. 3.19. Сумматор последовательного действия

дешифратор К155ИДЗ — стробируемый; выходные сигналы появляются при $W_0 = W_1 = 0$:

- пояснить работу распределителя импульсов;
- определить максимальное число выходов у распределителя при использовании счетчика К155ИЕ5;
- происходят ли опасные состязания в рассматриваемой схеме?

3.33. Как необходимо изменить схему распределителя импульсов (рис. 3.18), чтобы устранить опасные состязания (появление ложных сигналов на выходе дешифратора)?

3.34. Как необходимо изменить схему распределителя импульсов (рис. 3.18), чтобы длительность выходных импульсов распределителя была равна заданной?

3.35. Как необходимо изменить схему распределителя импульсов (рис. 3.18), чтобы получить распределитель на 10 выходов?

3.36. На рис. 3.19 изображена схема сумматора последовательного действия. Пояснить принцип работы устройства. Построить временные диаграммы всех сигналов для сложения двух чисел $A=1011$ и $B=1110$, предварительно введенных в регистры A и B .

3.37. На рис. 3.20 приведена функциональная схема устройства управления с жесткой логикой:

- пояснить принцип работы устройства управления;
- предложить варианты построения распределителя импульсов;
- определить, какое максимальное число выходов должен иметь дешифратор, если в устройстве используется 4-разрядный код операции;
- в устройстве управления используется распределитель импульсов на счетчике с последовательным переносом и дешифраторе (см. рис. 3.18)? Будет ли при этом нормально

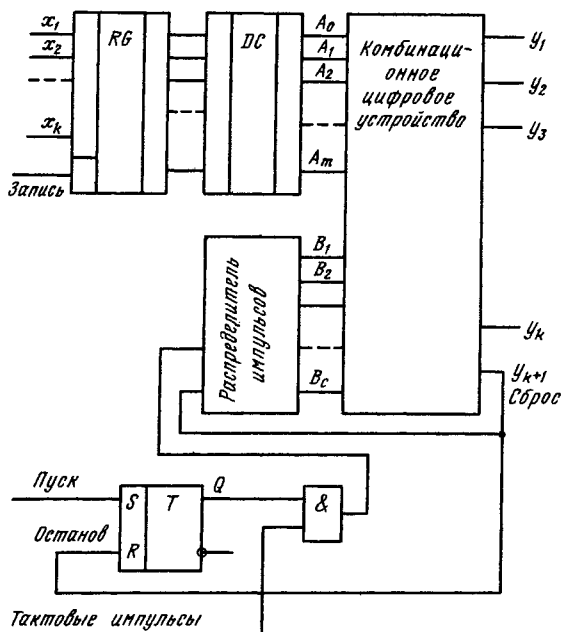


Рис. 3.20. Функциональная схема устройства управления с жесткой логикой

функционировать устройство управления? Указать причину сбоев, если они появляются, и способы устранения;

д) объяснить, как необходимо изменить функциональную схему устройства управления, чтобы при некоторых кодах операции сигналы управления повторялись n раз?

3.3. АНАЛИЗ И СИНТЕЗ ПОСЛЕДОВАТЕЛЬНОСТНЫХ УСТРОЙСТВ

В данном разделе приведены задачи и упражнения, в которых рассматриваются вопросы анализа и синтеза последовательностных устройств небольшой сложности: счетчиков, управляющих автоматов и т. д. Методика синтеза подобных устройств описана, например, в [3]. В качестве элементной базы используются в большинстве случаев микросхемы серии K155.

ЗАДАЧИ И УПРАЖНЕНИЯ

3.38. Оценить быстродействие двоичного счетчика с последовательным переносом, выполненного на D -триггерах типа K155TM2, у которых разрешающее время $\theta_{\text{ртр}} = 100$ нс и время задержки $t_{\text{зтр}} = 60$ нс (см. рис. 3.15).

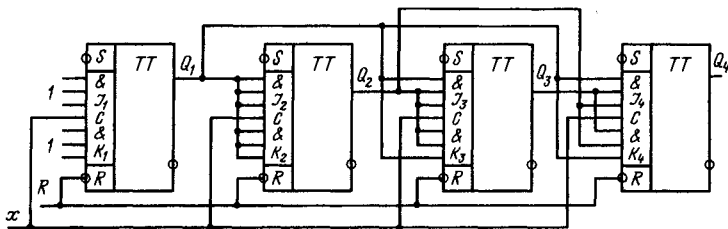


Рис. 3.21. Схема суммирующего двоичного счетчика

Указание. Быстродействие счетчика характеризуется разрешающим временем $\theta_{\text{рсч}}$ — минимальным временным интервалом между двумя переключающими импульсами, при котором счетчик надежно переключается и в каждом устойчивом состоянии находится в течение времени $\theta \geq \theta_{\text{мин}}$, где $\theta_{\text{мин}}$ — заданная величина (принять $\theta_{\text{мин}} = 40$ нс), а также разрешающей способностью $F_{\text{рсч}} = 1/\theta_{\text{рсч}}$.

3.39. Оценить быстродействие счетчика из задачи 3.38 при его работе в качестве делителя частоты.

3.40. Суммирующий двоичный счетчик с параллельным переносом (рис. 3.21) находится в состоянии 0011. Объяснить, как происходит переход счетчика в следующее состояние при поступлении на вход x одного импульса. Нарисуйте временные диаграммы и составьте таблицу переходов этого счетчика, определите модуль сета.

3.41. Оценить быстродействие двоичного счетчика с параллельным переносом, выполненного на JK -триггерах типа К155ТВ1 ($\theta_{\text{ртр}} = 100$ нс, $t_{\text{зтр}} = 40$ нс, $\theta_{\text{мин}} = 40$ нс) (см. задачу 3.40).

3.42. Определить функции возбуждения и синтезировать схемы суммирующих счетчиков с параллельным переносом и модулем сета (коэффициентом пересчета) k , равным: а) 6 и б) 10. В счетчиках должны быть реализованы состояния 0, 1, 2, ..., $(k-1)$. Использовать JK -триггеры типа К155ТВ1.

3.43. Оценить быстродействие кольцевого счетчика с прямыми связями, выполненного на D -триггерах типа К155ТМ2. Счетчик нагружен на устройство с временем срабатывания $t_{\text{сраб}} = 40$ нс (см. задачу 3.38).

3.44. Оценить быстродействие кольцевого счетчика на D -триггерах типа К155ТМ2 с одной перекрестной связью (рис. 3.12) при начальной установке 000. Длительность пребывания счетчика в каждом устойчивом состоянии должна быть не менее $\theta_{\text{мин}} = 40$ нс (см. задачу 3.38).

3.45. Синтезировать двоичный счетчик, используя JK -триггеры типа К155ТВ1. Минимальная длительность выходных сигналов $\theta_{\text{мин}} = 40$ нс. Выбрать схему, если счетчик должен иметь следующие параметры: а) $k = 4$, $F_{\text{рсч}} \geq 16$ МГц; б) $k = 16$, $F_{\text{рсч}} \geq 8$ МГц; в) $k = 2^{12}$, $F_{\text{рсч}} \geq 1$ МГц.

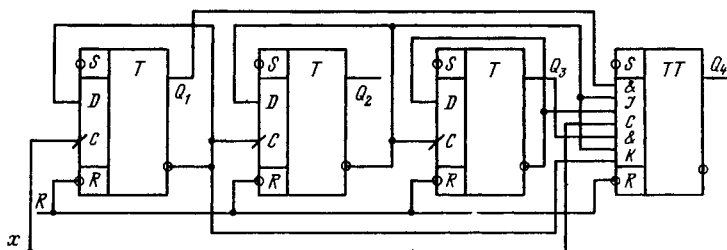


Рис. 3.22. Схема цифрового автомата к задаче 3.46

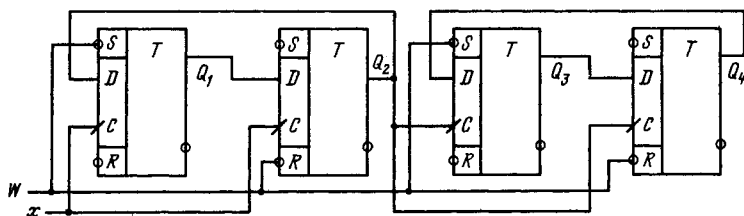


Рис. 3.23. Схема цифрового автомата к задаче 3.47

3.46. На вход x схемы на рис. 3.22 поступает периодическая последовательность положительных прямоугольных импульсов длительностью $t_{\text{нвх}} = 2$ мкс (период $T_{\text{вх}} = 6$ мкс). Пренебрегая задержками в элементах схемы, построить временные диаграммы напряжений на выходах Q_1 , Q_2 , Q_3 и Q_4 . Определить длительность импульса $t_{\text{нвых}}$ и длительность периода $T_{\text{вых}}$ на выходе Q_4 .

3.47. Составить таблицу переходов для схемы на рис. 3.23, в которой начальное состояние 0101 устанавливается отрицательным импульсом по входу W . Определить длину цикла. Предложить более простую схему, реализующую ту же таблицу переходов.

3.48. Построить управляющий автомат-распределитель импульсов на 16 направлений. Сравнить по объему оборудования (количеству корпусов микросхем) различные варианты схем,

используя триггеры, регистры, счетчики, дешифраторы серии К155 (см. приложение 1). Построить принципиальную схему простейшего устройства и временные диаграммы.

3.49. Определить функции возбуждения элементов памяти и функции выходов синхронного автомата, граф которого показан на рис. 3.24 (автомат управляется сигналами синхронизации). В автомате использовать D -триггеры

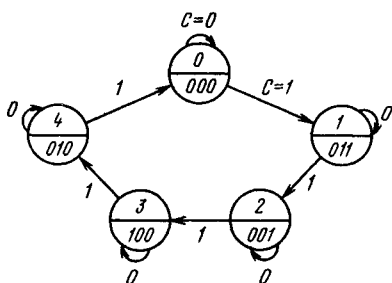


Рис. 3.24. Граф автомата к задаче 3.49

типа К155ТМ2 и естественное кодирование состояний (состоянию 0 соответствует код 000, состоянию 1—код 001 и т. д.). Составить принципиальную схему автомата, используя элементы серии КМ155.

3.50. Решить задачу 3.49, используя иное кодирование состояний, при котором код состояния совпадает с кодом выходных сигналов: $y_3 = Q_3$; $y_2 = Q_2$; $y_1 = Q_1$. Сравнить результаты решения задач 3.50 и 3.49.

3.51. Решить задачу 3.49, используя JK -триггеры типа К155ТВ1 и такое кодирование состояний, при котором код состояния совпадает с кодом выходных сигналов. Сравнить результаты решения задач 3.50 и 3.51. Составить принципиальную схему автомата.

Указание: при составлении таблицы для функций возбуждения JK -триггеров воспользуйтесь обращенной таблицей переходов (табл. 3.1). (Соответствующие переходы имеют место только при наличии сигнала синхронизации $C=0 \rightarrow 1 \rightarrow 0$.)

Таблица 3.1

Q^n	Q^{n+1}	J	K
0	0	0	Ф
0	1	1	Ф
1	0	Ф	1
1	1	Ф	0

3.52. Определить функции возбуждения и синтезировать из элементов серии КМ155 суммирующий счетчик с параллельным переносом и модулем счета $k=3$, используя: а) D -триггеры; б) JK -триггеры.

3.53. Определить функции возбуждения и синтезировать схемы вычитающих счетчиков с параллельным переносом и модулем счета, равным: а) 8; б) 5, используя JK -триггеры типа КМ155ТВ1 и логические элементы 2И-НЕ типа КМ155ЛАЗ.

Указание: в счетчиках реализуются состояния $(k-1)$, $(k-2)$, ..., 0.

3.54. Определить функции возбуждения и синтезировать схему управляемого реверсивного счетчика с модулем счета $k=4$, который при управляющем сигнале $a=1$ является суммирующим, а при $a=0$ —вычитающим. Использовать JK -триггеры типа КМ155ТВ1 и логические элементы 2И-2ИЛИ-НЕ типа КМ155ЛР1.

3.55. Суммирующий счетчик с модулем счета $k=15$ находился в состоянии $0101_{(2)}$, после чего на него были поданы $N=5437$ импульсов. В каком состоянии окажется счетчик?

3.56. Двоичный реверсивный счетчик с модулем счета $k=16$ находился в состоянии $1111_{(2)}$, после чего на него были поданы

$N_+ = 142$ импульса в режиме суммирования и $N_- = 83$ импульса в режиме вычитания. В каком состоянии окажется счетчик?

3.57. Реверсивный десятичный счетчик находился в состоянии $N^0 = 1001_{(2)}$, после чего на него были поданы $N_+ = 151$ импульс в режиме суммирования и $N_- = 211$ импульсов в режиме вычитания. Определить конечное состояние счетчика.

3.58. Синтезировать схему автомата, который имел бы цикл из четырех тактов и выдавал на шести выходах последовательности импульсов, указанные в табл. 3.2. Использовать D -триггеры и логические элементы серии K155.

Таблица 3.2

Номер такта	y_1	y_2	y_3	y_4	y_5	y_6
0	1	0	1	0	1	0
1	0	1	0	1	1	0
2	0	1	1	0	0	0
3	0	0	0	1	0	1

3.59. Синтезировать схему автомата, который имел бы цикл из восьми тактов и выдавал на двух выходах последовательности импульсов, указанные в табл. 3.3.

Таблица 3.3

Номер такта	y_1	y_2
0	0	0
1	1	0
2	1	0
3	0	1
4	1	0
5	0	1
6	0	1
7	1	1

3.60. Синтезировать схему перестраиваемого автомата, который имел бы цикл из четырех тактов и выдавал на выходе при управляющем сигнале $z=0$ последовательность y_1 , а при $z=1$ — последовательность y_2 (табл. 3.4).

Таблица 3.4

Номер такта	y_1	y_2
0	0	0
1	0	1
2	0	0
3	1	1

3.61. Синтезировать схему управляемого суммирующего счетчика с параллельным переносом, в котором при управляющем сигнале $a=0$ модуль счета $k=8$, а при $a=1$ $k=5$, используя JK -триггеры и логические элементы серии КМ155.

3.62. Синтезировать схему управляемого реверсивного 8-разрядного кольцевого счетчика с начальной установкой 00000001 и модулем счета $k=8$, в котором при управляющем сигнале $a=1$ осуществляется сдвиг единицы вправо, а при $a=0$ — влево. Использовать синхронные D -триггеры и ЛЭ серии КМ155.

3.63. Синтезировать схему реверсивного счетчика с параллельным переносом и модулем счета $k=4$, который должен работать по следующим правилам: при поступлении единицы только на вход x_1 к показаниям счетчика должна прибавляться единица; при поступлении единицы только на вход x_2 должна вычитаться единица; при поступлении на входы x_1 и x_2 одинаковых сигналов показания счетчика не должны меняться.

3.64. Синтезировать схему управляемого делителя частоты, в котором при подаче 3-разрядной команды $z_3z_2z_1$ устанавливается коэффициент деления частоты $k=2^{m+1}$, где $m=4z_3+2z_2+z_1$, причем $m=0, \dots, 7$. В схеме необходимо предусмотреть защиту от опасных состязаний.

3.65. Синтезировать схему многофункционального автомата, который имеет цикл из десяти тактов и выдает на выходе y при подаче соответствующей команды $z_3z_2z_1$ одну из восьми последовательностей $y_0, \dots, y_7$, приведенных в табл. 3.5. Необходимо предусмотреть защиту от опасных состязаний.

Таблица 3.5

Номер такта	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	0	0	0	1	1	0
1	0	0	0	0	1	1	1	0
2	0	0	1	1	0	0	0	1
3	1	0	1	1	1	0	0	1
4	0	0	0	1	0	1	0	0
5	0	0	0	1	1	1	0	0
6	0	1	1	0	0	0	1	1
7	1	1	1	0	1	0	1	1
8	0	0	0	0	0	1	1	1
9	0	0	0	0	0	1	1	1

3.66. Синтезировать схему автомата, который имеет цикл из восьми тактов и выдает на выходе при подаче соответствующей 3-разрядной команды одну из восьми последовательностей, указанных в табл. 3.6 (эти последовательности называются последовательностями Уолша третьего порядка).

Таблица 3.6

n	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1
2	0	0	1	1	1	1	0	0
3	1	1	0	0	1	1	0	0
4	1	0	0	1	1	0	0	1
5	0	1	1	0	1	0	0	1
6	0	1	0	1	1	0	1	0
7	1	0	1	0	1	0	1	0

3.67. Синтезировать схему устройства, которое имеет цикл из 16 тактов и выдает на выходе при подаче соответствующей 2-разрядной команды 00, 01, 10, 11 одну из четырех меандровых последовательностей: 01010101010101, 0011001100110011, 0000111100001111, 0000000011111111. Использовать элементы серии К155.

ОТВЕТЫ И РЕШЕНИЯ

Р3.1. б) Работа триггера описывается в табл. Р3.1;

Таблица Р3.1

S^n	R^n	Q^n	Q^{n+1}	$\overline{Q^{n+1}}$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Запрещенная комбинация

в) $Q^{n+1} = \overline{S^n} \vee R^n Q^n = \overline{S^n R^n} Q^n$;

г) временные диаграммы с учетом задержки приведены на рис. Р3.1;

д) $t_{\text{и мин}} = 2t_{\text{з ср}}$, $t_{\text{мин}} = 2t_{\text{з ср}}$. Для получения выходного сигнала с длительностью $2t_{\text{з ср}}$ следует увеличить разрешающее время до $3t_{\text{з ср}}$;

е) $Q=0$, $\overline{Q}=1$ либо $Q=1$, $\overline{Q}=0$;

ж) временные диаграммы входных и выходных сигналов приведены на рис. Р3.2.

Р3.2. б) Работа триггера описывается табл. Р3.2;

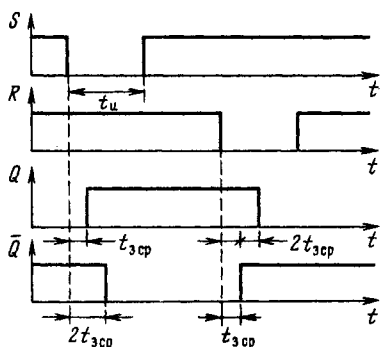


Рис. Р3.1. Временные диаграммы к задаче 3.1г

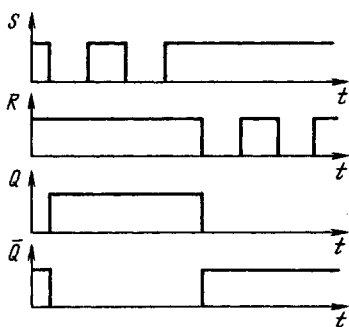


Рис. Р3.2. Временные диаграммы к задаче 3.1ж

Таблица Р3.2

S^n	R^n	Q^n	Q^{n+1}	$\overline{Q^{n+1}}$
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0

Запрещенная комбинация

в) $Q^{n+1} = Q^n \overline{R^n} \vee S^n = \overline{Q^n} \vee R^n \vee S^n$;

г) временные диаграммы с учетом задержек приведены на рис. Р3.3;

д) $t_{\text{н мин}} = 2t_{3\text{cp}}$, $t_{\text{мин}} = 2t_{3\text{cp}}$. Для получения выходного сигнала с длительностью $2t_{3\text{cp}}$ следует увеличить разрешающее время до $3t_{3\text{cp}}$;

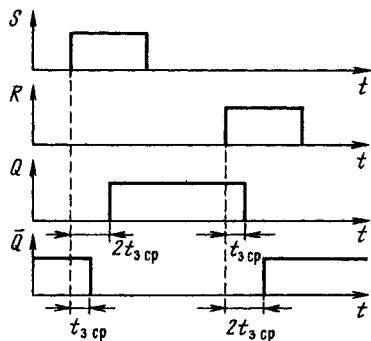


Рис. Р3.3. Временные диаграммы к задаче 3.2г

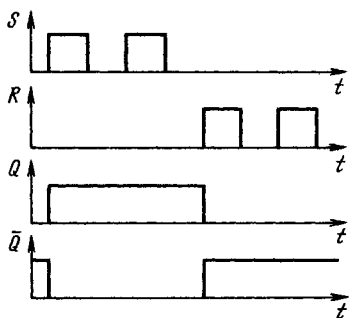


Рис. Р3.4. Временные диаграммы к задаче 3.2ж

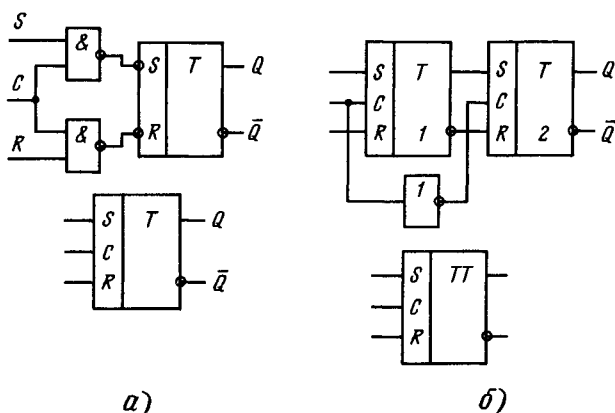


Рис. Р3.5. Синхронные RS-триггеры

е) $Q=0$, $\bar{Q}=1$, либо $Q=1$, $\bar{Q}=0$;

ж) временные диаграммы входных и выходных сигналов приведены на рис. Р3.4.

Р3.3. Схема одноступенчатого синхронного RS-триггера приведена на рис. Р3.5, а, двухступенчатого — на рис. Р3.5, б. Работа триггеров описывается табл. Р3.3.

Таблица Р3.3

C^n	S^n	R^n	Q^{n+1}
0	0	0	Q^n
0	0	1	$\bar{Q}^n$
0	1	0	Q^n
0	1	1	Q^n
1	0	0	$\bar{Q}^n$
1	0	1	0
1	1	0	1
1	1	1	Запрещенная комбинация

Одноступенчатый RSC-триггер срабатывает по фронту сигнала, поступающего на вход С. Двухступенчатый триггер срабатывает по срезу. Новая информация появляется на выходе после окончания действия импульса на входе С.

Р3.4. Схема D-триггера приведена на рис. Р3.6, временные диаграммы, поясняющие работу, — на рис. Р3.7. Двухступенчатый D-триггер срабатывает по срезу импульса, поступающего на вход С. Новая информация появляется на выходе после окончания действия импульса на входе С.

Р3.5. Временные диаграммы входных и выходных сигналов приведены на рис. Р3.8. Триггер ТМ2 имеет динамический вход

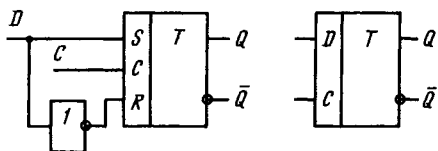


Рис. Р3.6. Одноступенчатый D -триггер

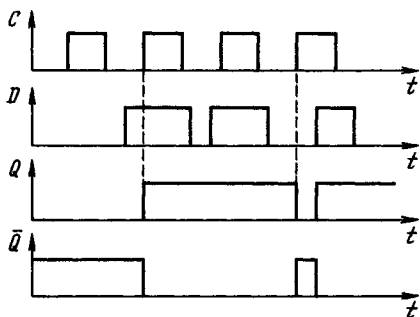


Рис. Р3.7. Временные диаграммы к задаче 3.4

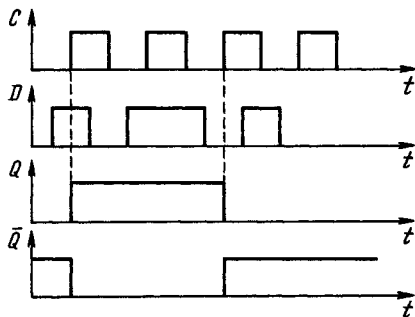


Рис. Р3.8. Временные диаграммы к задаче 3.5

C . Срабатывает триггер по фронту импульса C и в дальнейшем не реагирует на изменение сигнала на входе D при $C=1$.

Р3.6. При указанном соединении (рис. 3.8) образуется T -триггер. Временные диаграммы входных и выходных сигналов приведены на рис. Р3.9.

Р3.7. Образование T -триггера из D -триггера ТМ2 приведено в Р3.6. D -триггеры типа ТМ5 и ТМ7 одноступенчатые, со статическим входом C . По этой причине их нельзя использовать в качестве T -триггера, как это сделано на рис. 3.8.

Р3.8. Работа JK -триггера описывается табл. Р3.4.

Таблица Р3.4

C^n	J^n	K	Q^{n+1}
0	0	0	Q^n
0	0	1	$\bar{Q}^n$
0	1	0	Q^n
0	1	1	$\bar{Q}^n$
1	0	0	$\bar{Q}^n$
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}^n$

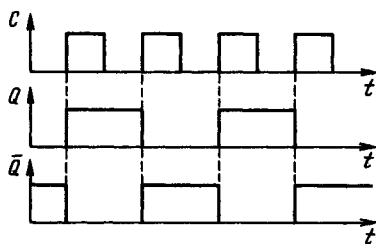


Рис. Р3.9 Временные диаграммы к задаче 3.6

Р3.9. Временные диаграммы входных и выходных сигналов приведены на рис. Р3.10.

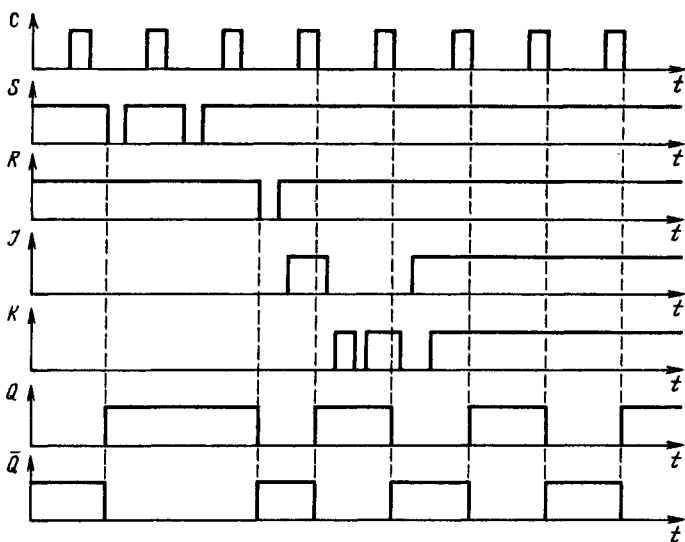


Рис. Р3.10. Временные диаграммы к задаче 3.9

Р3.10. Для получения T -триггера из JK -триггера необходимо на все входы, кроме входа C , подать сигнал «1». (Можно оставить входы разомкнутыми.) Вход C использовать как вход T . Можно входной сигнал T подавать на все входы J , K и C . Образование D -триггера из JK -триггера показано на рис. Р3.11.

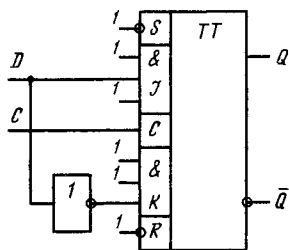
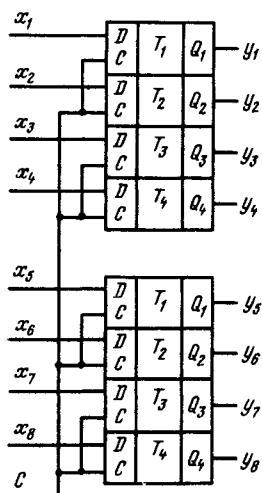


Рис. Р3.11. D -триггер

Рис. Р3.12. Схема регистра к задаче 3.11



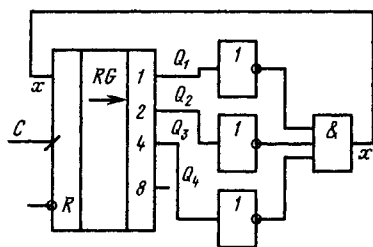
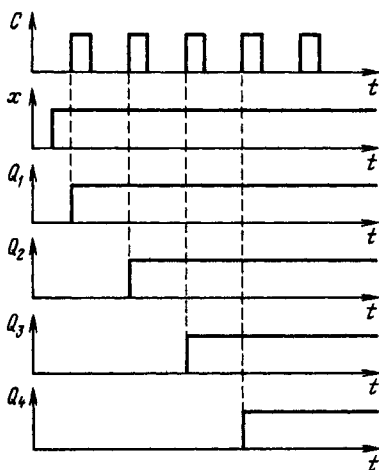


Рис. Р3.14. Кольцевой регистр с самовосстановлением после сбоя

Рис. Р3.13. Временные диаграммы к задаче 3.12

Р3.11. а) Схема регистра приведена на рис. Р3.12;
 в) нет необходимости стирать старую информацию;
 г) в регистр запишется новая информация, так как вход C у микросхем К155ТМ5 статический;
 д) не изменится, так как вход C у микросхемы К155ТМ2 динамический. Запись происходит только по фронту синхронизирующего импульса.

Р3.12. б) Временные диаграммы приведены на рис. Р3.13;
 в) задержка выходных сигналов для всех выходов одинакова и равна задержке, создаваемой одним триггером;

г) можно. Новая информация записывается в освобождающиеся триггеры;

д) нельзя. Триггеры ТМ5 и ТМ7 одноступенчатые со статическими входами C . При сигналах $x=1$ и $C=1$ единица запишется во все триггеры;

е) можно. Передача информации от триггера к триггеру должна осуществляться по двум проводам (с выходов Q и $\bar{Q}$ поступать на входы J и K).

Р3.13. Для получения распределителя импульсов можно замкнуть регистр в кольцо [соединить выход Q_4 (рис. 3.11) со входом x] и осуществлять продвижение одной единицы. Такая схема на практике используется редко, так как после сбоя работоспособность схемы автоматически не восстанавливается. Для устранения указанного недостатка используется кольцевой регистр с самовосстановлением после сбоя. В регистр записывается единица только после того, как регистр «очистится» от единиц. Схема такого регистра приведена на рис. Р3.14.

Максимальное число выходов при использовании одной микросхемы регистра равно четырем. Для увеличения числа

выходов необходимо прибегнуть к последовательному соединению регистров.

Р3.14. Работа регистра описывается табл. Р3.5; $k=2n=8$.

Таблица Р3.5

Номер состояния	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

Р3.15. Работа регистра описывается табл. Р3.6; $k=15$. Рассматриваемое устройство используется для получения псевдослучайной последовательности нулей и единиц.

Таблица Р3.6

Номер состояния	Q_1	Q_2	Q_3	Q_4
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	1	0	0	1
4	1	1	0	0
5	0	1	1	0
6	1	0	1	1
7	0	1	0	1
8	1	0	1	0
9	1	1	0	1
10	1	1	1	0
11	1	1	1	1
12	0	1	1	1
13	0	0	1	1
14	0	0	0	1
15	1	0	0	0

Р3.16. Для устранения отмеченного недостатка необходимо обнаруживать кодовую комбинацию $Q_1=Q_2=Q_3=Q_4=0$ и полученным сигналом записывать единицу в первый разряд регистра так, как это делается в кольцевом регистре с самовосстановлением (см. Р3.13).

Р3.17. При использовании кольцевого регистра с самовосстановлением (Р3.13) выходной сигнал может сниматься с лю-

бого разряда регистра. Коэффициент деления равен числу триггеров в регистре $k=n$.

При использовании кольцевого регистра с логической обратной связью (рис. 3.13) необходимо обнаруживать какую-либо кодовую комбинацию (любую) для получения выходного сигнала. Максимальный коэффициент деления $k=2^n-1$. В рассматриваемом примере $k=15$.

Для уменьшения коэффициента деления можно обнаруживать необходимую комбинацию и сбрасывать регистр в начальное состояние, с которого начинается счет в следующем цикле.

При использовании кольцевых регистров в качестве делителей задержка выходного сигнала относительно входного не зависит от числа разрядов в регистре и определяется задержкой, создаваемой одним триггером регистра и логической схемой, которая служит для получения выходного сигнала.

Р3.18. При последовательном соединении кольцевых регистров каждый выходной импульс первого регистра осуществляет сдвиг единицы во втором регистре на один разряд, поэтому $k=k_1 \cdot k_2 = 4 \cdot 4 = 16$.

Р3.19. а) Нельзя. Информация с выходов всех регистров будет одновременно поступать на общую шину;

б) нельзя по той же причине, что и в п. а);

в) можно присоединить через шинный формирователь, обладающий тремя состояниями выхода. В третьем состоянии информация не передается и $R_{\text{вых}} = \infty$.

Р3.20. б) Работа счетчика описывается табл. Р3.7. Коэффициент пересчета $k=8$;

Таблица Р3.7

Номер состояния	Q_3	Q_2	Q_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

в) временные диаграммы приведены на рис. Р3.15;

г) $t_{\text{з макс}} = 3t_{\text{з тр}} = 165 \text{ нс}$;

д) необходимо переключить перемычки между триггерами с выходов $\bar{Q}$ на выходы Q .

Р3.21. Нельзя. Триггеры ТМ5 и ТМ7 — одноступенчатые со статическим входом S .

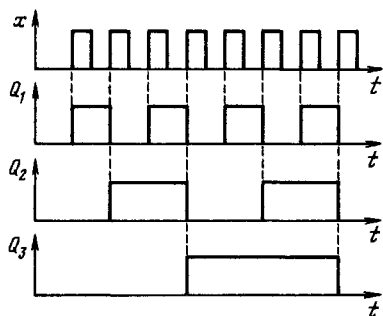


Рис. P3.15. Временные диаграммы к задаче 3.20

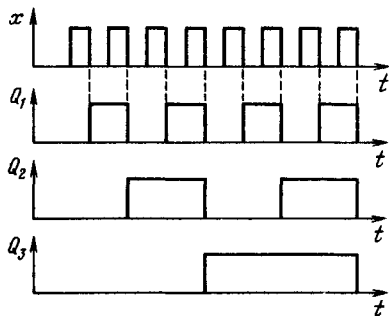


Рис. P3.16. Временные диаграммы к задаче 3.22

P3.22. В счетчике должны использоваться T -триггеры. Для образования T -триггера из JK -триггера необходимо на все входы J и K подать сигнал единица (в ТТЛ-элементах можно эти выходы оставить разомкнутыми). Так как JK -триггер двухступенчатый, то сигнал на вход C должен поступать с выхода Q предыдущего триггера. Опрокидывание триггера происходит по срезу импульса, подаваемого на вход C . Временные диаграммы приведены на рис. P3.16.

P3.23. Сигналы на вход C должны поступать с выхода $\bar{Q}$ предыдущего триггера.

P3.24. Коэффициент пересчета счетчика $k = 2^n = 2^3 = 8$. После каждых восьми импульсов счетчик будет переходить в нулевое состояние. Следовательно, после подачи 64 импульсов он окажется в нулевом состоянии, а после 67 в нем окажется записанным число 011(3).

P3.25. б) Работа счетчика описывает табл. P3.7;

в) временные диаграммы приведены на рис. P3.16;

г) $t_3 = t_{3\text{тр}}$;

д) максимальное число разрядов $n = 4$.

P3.26. Схема счетчика приведена на рис. P3.17. Задержка, создаваемая счетчиком, $t_3 = 2t_{31} = 270$ нс.

P3.27. Задержка, создаваемая счетчиком, $t_3 = t_{31} + t_{3\text{ср}} = 70$ нс. Коэффициент пересчета $k = k_1 \cdot k_2 = 16 \cdot 16 = 256$.

P3.28. Для получения $k = 10$ необходимо обнаруживать число десять (1010) и сбрасывать счетчик в нулевое состояние. Сигнал для сброса $A = Q_4 \bar{Q}_3 Q_2 \bar{Q}_1$. Если учесть факультативные условия, то $A = Q_4 Q_2$. Схема счетчика с $k = 10$ приведена на рис. P3.18.

P3.29. Необходимо обнаруживать число 120 (1111000) и сбрасывать счетчик в нулевое состояние. Сигнал для сброса $A = Q_7 Q_6 Q_5 Q_4 \bar{Q}_3 \bar{Q}_2 \bar{Q}_1$. Если учесть факультативные условия, то $A = Q_7 Q_6 Q_5 Q_4$. Необходим 7-разрядный счетчик. Используем счетчик, построенный на двух микросхемах K155ИЕ5 (см.

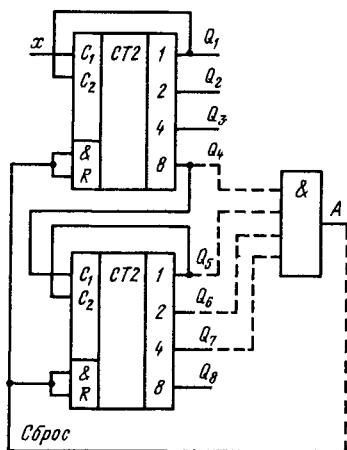


Рис. Р3.17. Восьмиразрядный суммирующий счетчик на микросхемах К155IE5

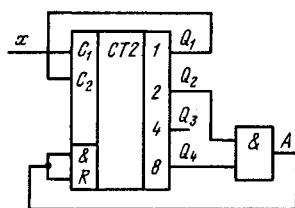


Рис. Р3.18. Счетчик с коэффициентом пересчета $k=10$

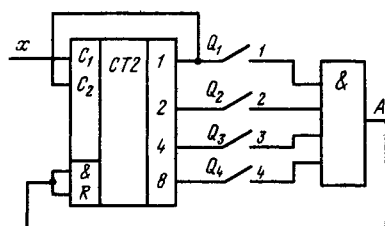


Рис. Р3.19. Перестраиваемый счетчик

рис. Р3.17). Подключение схемы для сброса показано штриховой линией.

Р3.30. Схема перестраиваемого счетчика приведена на рис. Р3.19. Так как в счетчике используются ТТЛ-элементы, то размыкание ключа соответствует подаче единицы на соответствующий вход элемента И. Например, при замыкании всех ключей $k=15$, при замыкании только ключа 2 $k=2$.

Р3.31. Отключение выхода Q_4 от входа ячейки И равносильно подаче на этот вход сигнала 1. При этом $k=k_1 k_2 = 8 \cdot 16 = 128$.

Р3.32. б) Максимальное число выходов $k=2^n=2^4=16$;

в) так как в схеме используется счетчик с последовательным переносом, то на выходе дешифратора могут появляться ложные импульсы во время перехода счетчика из одного состояния в другое. Например, при переходе счетчика из состояния 1111 в нулевое состояние на его выходе появляются дополнительно следующие кодовые комбинации: 1110, 1100, 1000. Эти комбинации вызывают ложные сигналы на выходе дешифратора.

Р3.33. Для устранения опасных состязаний необходимо на входы И дешифратора подавать стробирующие сигналы 0 с задержкой относительно сигнала x , равной или большей максимального времени установления сигналов в счетчике ($t_{1сч} = 135$ нс).

Р3.34. Для этой цели можно использовать два последовательно включенных ждущих мультивибратора. Один для созда-

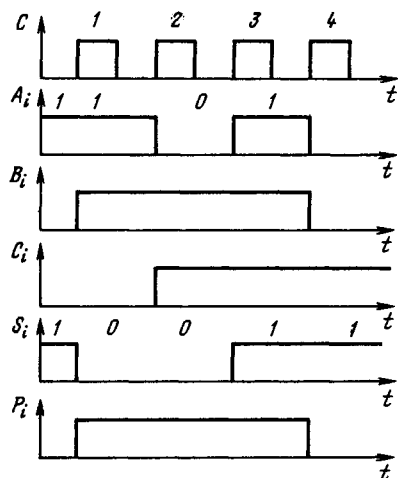


Рис. P3.20. Временные диаграммы к задаче 3.36

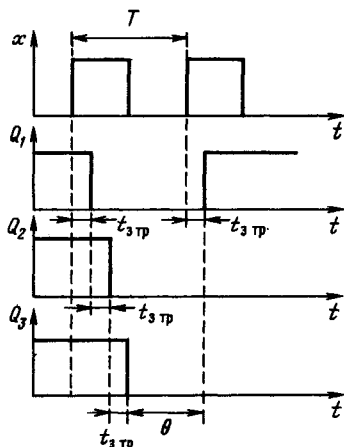


Рис. P3.21. Временные диаграммы для счетчика с последовательным переносом

ния задержки [устраняющий опасные состязания (см. P3.33)], другой для получения стробирующего импульса, длительность которого должна быть равна необходимой длительности выходных импульсов $t_{\text{вых}}$.

P3.35. Для этой цели следует использовать счетчик с коэффициентом пересчета $k=10$ (см. P3.28).

P3.36. Слагаемые A и B вводятся в регистры A и B в последовательном коде, начиная с младших разрядов. В это время может выполняться сложение над «выталкиваемыми» кодами чисел, которые были туда занесены ранее.

Если коды чисел A и B занесены в регистры, то перед приходом импульса C на выходе сумматора образуются сигналы суммы S и переноса P в результате сложения младших разрядов чисел A и B (рис. P3.20). В момент подачи импульса C перенос записывается в D -триггер, а сумма — в регистр суммы по фронту импульса C (входы C — динамические). В это же время на вход сумматора поступают следующие разряды слагаемых из сдвигающих регистров A и B и перенос, образованный в предыдущий момент, из D -триггера. Образуются снова сигналы суммы и переноса, и по фронту импульса C следующий разряд суммы и перенос записываются в сдвигающий регистр суммы и в D -триггер и т. д.

Временные диаграммы сигналов, характеризующие работу суммирующего устройства, приведены на рис. P3.20.

P3.37. а) Код операции записывается в регистр кода операции по сигналу «Запись», откуда поступает на дешифратор

кода операции. Появляется сигнал 1 на соответствующем выходе дешифратора. Каждому коду операции соответствует сигнал 1 на соответствующем выходе дешифратора.

По сигналу «Пуск» записывается 1 в RS-триггер, благодаря чему тактирующие импульсы начинают поступать на распределитель импульсов через ячейку И. На выходе распределителя начинают появляться импульсы в каждом такте на соответствующем выходе, начиная с первого.

Эти импульсы и сигналы A с выхода дешифратора поступают на комбинационное цифровое устройство. Это устройство создает необходимые сигналы управления в каждом такте. На выходе y_{k+1} создается сигнал для останова устройства после выполнения соответствующей операции, определяемой кодом операции. Этот сигнал переводит RS-триггер в нулевое состояние, что приводит к прекращению подачи синхроимпульсов на распределитель импульсов. Он устанавливается в исходное нулевое состояние;

б) распределитель импульсов может быть выполнен на кольцевом регистре, в котором продвигается одна единица, либо на счетчике и дешифраторе;

в) максимальное число выходов $2^4 = 16$;

г) не будет, так как на выходе дешифратора будут появляться ложные импульсы (см. Р3.33). Способы устранения опасных состязаний указаны в Р3.33;

д) ввести счетчик числа циклов. Останов должен осуществляться после заданного числа циклов сигналом, формируемым этим счетчиком.

Р3.38. В наихудшем случае, когда в исходном состоянии $Q_1 = Q_2 = \dots = Q_m = 1$, все триггеры в схеме на рис. 3.15 при подаче входного импульса переключаются один за другим (рис. Р3.21). Поэтому должно быть $T + t_{3\text{тр}} \geq m t_{3\text{тр}} + \theta$. Вместе с тем для надежного переключения первого триггера должно быть $T > \theta_{\text{ртр}}$. Таким образом,

$$\begin{aligned} \theta_{\text{рсч}} &= T_{\text{мин}} = \max \{ \theta_{\text{ртр}}, (m-1)t_{3\text{тр}} + \theta_{\text{мин}} \} = \\ &= \max \{ 100, 2 \cdot 60 + 40 \} = 160 \text{ нс}; F_{\text{рсч}} = 1/\theta_{\text{рсч}} \approx 6,3 \text{ МГц}. \end{aligned}$$

Р3.39. Для надежной работы делителя частоты достаточно выполнение условия $T \geq \theta_{\text{ртр}}$. Поэтому $\theta_{\text{рдел}} = 100 \text{ нс}$ и $F_{\text{рдел}} = 10 \text{ МГц}$.

Р3.40. В исходном состоянии $Q_1 = Q_2 = 1$ и $Q_3 = Q_4 = 0$, т. е. $J_2 = K_2 = 1$; $J_3 = K_3 = 1$ и $J_4 = K_4 = 0$. Поэтому при поступлении на вход x положительного импульса переключаются 1, 2 и 3-й триггеры, т. е. счетчик оказывается в состоянии 0100 ($k=16$).

Р3.41. При любых переходах все триггеры счетчика с параллельным переносом переключаются одновременно. Поэтому

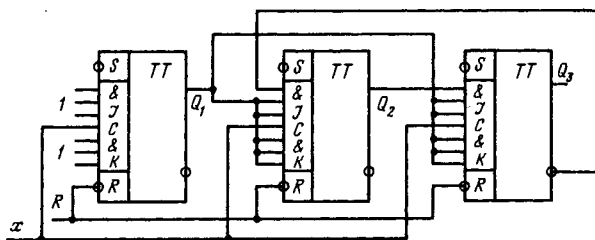


Рис. Р3.22. Схема счетчика с модулем счета $k=6$

разрешающее время и разрешающая способность такого счетчика не зависят от числа триггеров и соответственно равны

$$\theta_{\text{рсч}} = \max \{ \theta_{\text{ртр}}, \theta_{\text{мин}} \} = \max \{ 100, 40 \} = 100 \text{ нс};$$

$$F_{\text{рсч}} = 1/\theta_{\text{рсч}} = 10 \text{ МГц.}$$

Р3.42. Функции переходов и выходов счетчика с $k=6$ представлены в табл. Р3.8 (см. также табл. 3.1). После минимизации с помощью карт Карно получаем $J_1 = K_1 = 1$; $J_2 = Q_1 \bar{Q}_3$; $K_2 = K_3 = Q_1$; $J_3 = Q_1 Q_2$ (схема на рис. Р3.22). У десятичного счетчика $J_1 = K_1 = 1$; $J_2 = Q_1 \bar{Q}_4$; $K_2 = Q_1$; $J_3 = K_3 = Q_1 Q_2$; $J_4 = Q_1 Q_2 Q_3$; $K_4 = Q_1$.

Таблица Р3.8

Номер состояния	Q_3^*	Q_2^*	Q_1^*	Q_3^{*+1}	Q_2^{*+1}	Q_1^{*+1}	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	0	1	0	Ф	0	Ф	1	Ф
1	0	0	1	0	1	0	0	Ф	1	Ф	Ф	1
2	0	1	0	0	1	1	0	Ф	Ф	0	1	Ф
3	0	1	1	1	0	0	1	Ф	Ф	1	Ф	1
4	1	0	0	1	0	1	Ф	0	0	Ф	1	Ф
5	1	0	1	0	0	0	Ф	1	0	Ф	Ф	1
6	1	1	0	Ф	Ф	Ф	Ф	Ф	Ф	Ф	Ф	Ф
7	1	1	1	Ф	Ф	Ф	Ф	Ф	Ф	Ф	Ф	Ф

Р3.43. В кольцевом счетчике с прямыми связями при подаче входного импульса одновременно переключаются два соседних триггера (рис. Р3.23). Должны выполняться условия $T > \theta_{\text{рсч}}$ и $T + t_{\text{зтр}} > t_{\text{зтр}} + \theta$. Поэтому $\theta_{\text{рсч}} = \max \{ \theta_{\text{ртр}}, \theta_{\text{мин}} \} = \max \{ \theta_{\text{ртр}}, t_{\text{сраб}} \} = \max \{ 100, 40 \} = 100 \text{ нс.}$

Р3.44. Как показывает рис. Р3.24, в кольцевом счетчике с одной перекрестной связью каждый импульс вызывает переключение только одного триггера. Должны выполняться условия $mT \geq \theta_{\text{ртр}}$ и $T + t_{\text{зтр}} \geq t_{\text{зтр}} + \theta$. Следовательно, $\theta_{\text{рсч}} = T_{\text{мин}} = \max \{ \theta_{\text{ртр}}/m, \theta_{\text{мин}} \} = \max \{ 100/3, 40 \} = 40 \text{ нс};$ $F_{\text{рсч}} = 1/\theta_{\text{рсч}} \approx 25 \text{ МГц.}$

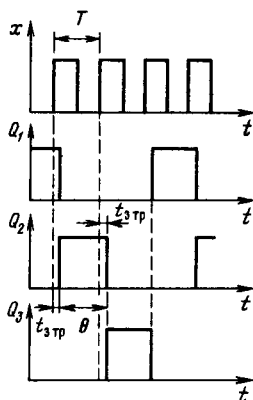


Рис. Р3.23. Временные диаграммы для кольцевого счетчика с прямыми связями

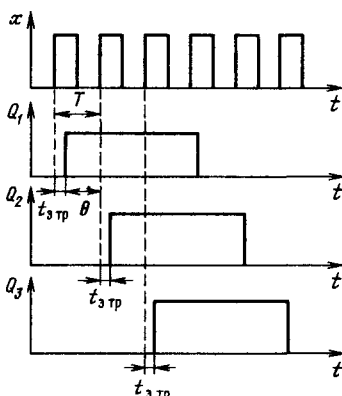


Рис. Р3.24. Временные диаграммы для кольцевого счетчика с одной перекрестной и остальными прямыми связями

Р3.45. Выбираем схемы счетчика:

а) кольцевого с одной перекрестной связью ($m=2$, $F_{\text{рсч}}=25$ МГц);

б) с параллельным переносом ($m=4$, $F_{\text{рсч}}=10$ МГц);

в) с последовательным переносом ($m=12$, $F_{\text{рсч}}=2$ МГц).

Р3.46. Временные диаграммы приведены на рис. Р3.25; $t_{\text{и вых}}=3T_{\text{вх}}=18$ мкс; $T_{\text{вых}}=8T_{\text{вх}}=48$ мкс.

Р3.47. Таблица переходов представлена в табл. Р3.9; $m=4$. Ту же таблицу переходов реализует двоичный счетчик на двух D-триггерах (при начальной установке 11), если сигналы Q_1 и Q_2 снимать с выходов первого триггера, а Q_3 и Q_4 — со второго.

Таблица Р3.9

Номер состояния	Q_4	Q_3	Q_2	Q_1
0	0	1	0	1
1	1	0	1	0
2	1	0	0	1
3	0	1	1	0

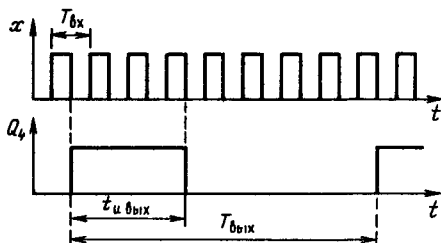


Рис. Р3.25. Временные диаграммы для автомата к задаче 3.46

Р3.48. Можно использовать либо кольцевой счетчик с прямыми связями и модулем счета 16 (варианты 1, 2 в табл. Р3.10), либо двоичный счетчик с модулем счета 16

и дешифратор на четыре входа (варианты 3, 4 в табл. P3.10).

Таблица P3.10

Вариант	D-триггер KM155TM2	Регистр KM155IP1	Двоичный счетчик KM155IE7	Дешифратор KM155ID3	Число корпусов
1	16	—	—	—	8
2	—	4	—	—	4
3	4	—	—	1	3
4	—	—	1	1	2

Таблица P3.11

Номер состояния	Q_3	Q_2	Q_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0

P3.49. Кодирование состояний показано в табл. P3.11, переключательные таблицы для функций переходов и выходов представлены в табл. P3.12. Минимизация с помощью карт Карно дает $D_1 = \bar{Q}_1 \bar{Q}_3$; $D_2 = \bar{Q}_1 Q_2 \vee Q_1 \bar{Q}_2$; $D_3 = Q_1 Q_2$; $y_1 = \bar{Q}_1 Q_2 \vee Q_1 \bar{Q}_2$; $y_2 = Q_1 \bar{Q}_2 \vee Q_3$; $y_3 = Q_1 Q_2$.

Таблица P3.12

Номер состояния	Q_3^*	Q_2^*	Q_1^*	$Q_3^{*+1}=D_3$	$Q_2^{*+1}=D_2$	$Q_1^{*+1}=D_1$	y_3	y_2	y_1
0	0	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1	1
2	0	1	0	0	1	1	0	0	1
3	0	1	1	1	0	0	1	0	0
4	1	0	0	0	0	0	0	1	0
5	1	0	1	Ф	Ф	Ф	Ф	Ф	Ф
6	1	1	0	Ф	Ф	Ф	Ф	Ф	Ф
7	1	1	1	Ф	Ф	Ф	Ф	Ф	Ф

P3.50. Кодирование состояний показано в табл. P3.13, а переключательные таблицы для функций переходов и выходов представлены в табл. P3.14. После минимизации получаем $D_1 = \bar{Q}_1 \bar{Q}_2 \bar{Q}_3 \vee Q_1 Q_2$; $D_2 = \bar{Q}_1 \bar{Q}_2$; $D_3 = Q_1 \bar{Q}_2$; $y_1 = Q_1$; $y_2 = Q_2$; $y_3 = Q_3$. Кодирование состояний по табл. P3.13 обеспечивает получение более простой схемы, чем по табл. P3.11.

Таблица Р3.13

Номер состояния	Q_3	Q_2	Q_1
0	0	0	0
1	0	1	1
2	0	0	1
3	1	0	0
4	0	1	0

Таблица Р3.14

Номер состояния	$Q_3^* = y_3$	$Q_2^* = y_2$	$Q_1^* = y_1$	$Q_3^{*+1} = D_3$	$Q_2^{*+1} = D_2$	$Q_1^{*+1} = D_1$
0	0	0	0	0	1	1
1	0	0	1	1	0	0
2	0	1	0	0	0	0
3	0	1	1	0	0	1
4	1	0	0	0	1	0
5	1	0	1	Φ	Φ	Φ
6	1	1	0	Φ	Φ	Φ
7	1	1	1	Φ	Φ	Φ

Р3.51. Кодирование состояний см. в табл. Р3.13, а функции возбуждения — в табл. Р3.15. После минимизации получаем $J_1 = \overline{Q_2} \overline{Q_3}$; $K_1 = \overline{Q_2}$; $J_2 = \overline{Q_1}$; $K_2 = 1$; $J_3 = Q_1 \overline{Q_2}$; $K_3 = 1$. Схема автомата приведена на рис. Р3.26. В отличие от схем на D -триггерах (см. задачи 3.49 и 3.50) она не содержит логических элементов.

Таблица Р3.15

Номер состояния	$Q_3^* = y_3$	$Q_2^* = y_2$	$Q_1^* = y_1$	Q_3^{*+1}	Q_2^{*+1}	Q_1^{*+1}	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	1	1	0	Φ	1	Φ	1	Φ
1	0	0	1	1	0	0	1	Φ	0	Φ	1	Φ
2	0	1	0	0	0	0	0	Φ	1	1	0	Φ
3	0	1	1	0	0	1	0	Φ	1	1	0	Φ
4	1	0	0	0	1	0	Φ	1	1	Φ	0	Φ
5	1	0	1	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ
6	1	1	0	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ
7	1	1	1	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ

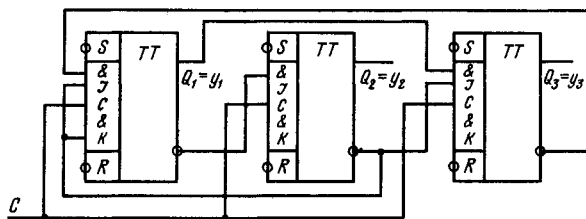


Рис. Р3.26. Схема автомата

P3.52. а) $D_2 = Q_1$; $D_1 = \bar{Q}_2 \bar{Q}_1$; б) $J_2 = Q_1$; $J_1 = \bar{Q}_2$; $K_1 = K_2 = 1$.

P3.53. а) $J_1 = K_1 = 1$; $J_2 = K_2 = \bar{Q}_1$; $J_3 = K_3 = \bar{Q}_1 \bar{Q}_2$;

б) $J_1 = \bar{Q}_2 \bar{Q}_3$; $K_1 = 1$; $J_2 = Q_3$; $K_2 = \bar{Q}_1$; $J_3 = \bar{Q}_1 \bar{Q}_2$; $K_3 = 1$.

P3.54. Функции переходов и функции возбуждения счетчика приведены в табл. P3.16.

Таблица P3.16

Номер состояния	a	Q_2^*	Q_1^*	Q_2^{*+1}	Q_1^{*+1}	J_2	K_2	J_1	K_1
0	0	0	0	1	1	1	Ф	1	Ф
1	0	0	1	0	0	0	Ф	Ф	1
2	0	1	0	0	1	Ф	1	1	Ф
3	0	1	1	1	0	Ф	0	Ф	1
4	1	0	0	0	1	0	Ф	1	Ф
5	1	0	1	1	0	1	Ф	Ф	1
6	1	1	0	1	1	Ф	0	1	Ф
7	1	1	1	0	0	Ф	1	Ф	1

После минимизации с помощью карт Карно получим $J_1 = K_1 = 1$, $J_2 = K_2 = a\bar{Q}_1 \vee \bar{a}Q_1$.

P3.55. Исходное состояние соответствует числу $N^0 = 5$, конечное состояние — остатку от деления $N^0 + N$ на модуль счета k , т. е. числу

$$N^k = N^0 + N - \left\lfloor \frac{N^0 + N}{k} \right\rfloor k = 5 + 5437 - \left\lfloor \frac{5 + 5437}{15} \right\rfloor \cdot 15 = 5442 - 362 \cdot 15 = 12 = 1100_{(2)}.$$

Таким образом, триггеры счетчика окажутся в следующих состояниях: $Q_1 = Q_2 = 0$, $Q_3 = Q_4 = 1$.

$$\text{P3.56. } N^k = N^0 + N_+ - N_- - \left\lfloor \frac{N^0 + N_+ - N_-}{k} \right\rfloor k = 10 = 1010_{(2)}.$$

P3.57. Конечное состояние совпадает с начальным.

P3.58. В схеме можно использовать двоичный счетчик на двух D -триггерах типа КМ155ТМ2. Выходы счетчика Q_1 , $\bar{Q}_1$, Q_2 и $\bar{Q}_2$ необходимо подключить ко входам комбинационных схем, реализующих следующие функции: $y_1 = \bar{Q}_1 \bar{Q}_2$; $y_2 = Q_1 Q_2 \vee \bar{Q}_1 \bar{Q}_2$; $y_3 = \bar{Q}_1$; $y_4 = Q_1$; $y_5 = \bar{Q}_2$; $y_6 = Q_1 Q_2$. Комбинационные схемы можно выполнить на одном элементе 2И-2ИЛИ-НЕ типа КМ155ЛР1 и трех элементах 2И-НЕ типа КМ155ЛА3 (один из них — в качестве инвертора) — всего два корпуса.

P3.59. В схеме можно использовать счетчик типа КМ155ИЕ7. Для реализации функций выходов $y_1 = Q_1 \oplus Q_2 \oplus Q_3$;

$y_2 = Q_1 Q_2 \vee Q_2 Q_3 \vee Q_3 Q_1 = \text{maj}(Q_1, Q_2, Q_3)$ можно использовать 1-разрядный комбинационный сумматор типа К155ИМ1.

Р3.60. В схеме можно использовать двоичный счетчик типа КМ155ИЕ7. Функция выхода автомата представлена в табл. Р3.17: $y = Q_1 z \vee Q_1 Q_2$; ее можно реализовать на элементе 2И-2ИЛИ-НЕ типа КМ155ЛР1 и инверторе типа К155ЛН1.

Таблица Р3.17

Номер состояния	z	Q_2	Q_1	y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Р3.61. Таблица переходов и функций возбуждения представлена в табл. Р3.18. Минимизированные функции возбуждения таковы: $J_1 = a Q_3$; $K_1 = 1$; $J_2 = \bar{a} Q_1 \vee Q_1 \bar{Q}_3$; $K_2 = Q_1 \vee a Q_3$; $J_3 = Q_1 Q_3$; $K_3 = a \vee Q_1 Q_2$. Схема может быть выполнена на трех JK-триггерах типа КМ155ТВ1 и восьми элементах типа КМ155ЛА3 (всего пять корпусов).

Таблица Р3.18

Номер состояния	a	Q_3^*	Q_2^*	Q_1^*	Q_3^{*+1}	Q_2^{*+1}	Q_1^{*+1}	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	0	0	1	0	Ф	0	Ф	1	Ф
1	0	0	0	1	0	1	0	0	Ф	1	Ф	Ф	1
2	0	0	1	0	0	1	1	0	Ф	Ф	0	1	Ф
3	0	0	1	1	1	0	0	1	Ф	Ф	1	Ф	1
4	0	1	0	0	1	0	1	Ф	0	0	Ф	1	Ф
5	0	1	0	1	1	1	0	Ф	0	1	Ф	Ф	1
6	0	1	1	0	1	1	1	Ф	0	Ф	0	1	Ф
7	0	1	1	1	0	0	0	Ф	1	Ф	1	Ф	1
8	1	0	0	0	0	0	1	0	Ф	0	Ф	1	Ф
9	1	0	0	1	0	1	0	0	Ф	1	Ф	Ф	1
10	1	0	1	0	0	1	1	0	Ф	Ф	0	1	Ф
11	1	0	1	1	1	0	0	1	Ф	Ф	1	Ф	1
12	1	1	0	0	0	0	0	Ф	1	0	Ф	0	Ф
13	1	1	0	1	0	0	0	Ф	1	0	Ф	Ф	1
14	1	1	1	0	0	0	0	Ф	1	Ф	1	0	Ф
15	1	1	1	1	0	0	0	Ф	1	Ф	1	Ф	1

Р3.62. В кольцевом счетчике все триггеры равноценны. Поэтому функции возбуждения для всех триггеров однотипны.

Например, для второго триггера при $a=1$ сигнал возбуждения формируется первым триггером ($D_2=Q_1$), а при $a=0$ — третьим ($D_2=Q_3$), т. е. $D_2=aQ_1 \vee \bar{a}Q_3=aQ_1 \vee \bar{a}Q_3$. Вообще для i -го триггера ($i=2, \dots, 7$) $D_i=aQ_{i-1} \vee \bar{a}Q_{i+1}$; для первого $D_1=aQ_8 \vee \bar{a}Q_2$; для последнего $D_8=aQ_7 \vee \bar{a}Q_1$. Схема будет содержать восемь D -триггеров типа КМ155ТМ2 и восемь элементов 2И-2ИЛИ-НЕ типа КМ155ЛР1 (всего восемь корпусов).

Р3.63. Таблица переходов и функций возбуждения JK -триггеров представлена в табл. Р3.19. Функции возбуждения:

$J_1=K_1=x_1x_2 \vee \bar{x}_1\bar{x}_2$; $J_2=K_2=Q_1x_1\bar{x}_2Q_1\bar{x}_1x_2$. Схема может быть выполнена на двух JK -триггерах типа КМ155ТВ1, двух элементах 2И-2ИЛИ-НЕ типа КМ155ЛР1, один из которых используется в качестве элемента 2И-НЕ, и двух элементах 4И-НЕ типа КМ155ЛА1 (всего четыре корпуса).

Таблица Р3.19

Номер состояния	x_2	x_1	Q_2^*	Q_1^*	Q_2^{*+1}	Q_1^{*+1}	J_2	K_2	J_1	K_1
0	0	0	0	0	0	0	0	Ф	0	Ф
1	0	0	0	1	0	1	0	Ф	Ф	0
2	0	0	1	0	1	0	Ф	0	0	Ф
3	0	0	1	1	1	1	Ф	0	Ф	0
4	0	1	0	0	0	1	0	Ф	1	Ф
5	0	1	0	1	1	0	1	Ф	Ф	1
6	0	1	1	0	1	1	Ф	0	1	Ф
7	0	1	1	1	0	0	Ф	1	Ф	1
8	1	0	0	0	1	1	1	Ф	1	Ф
9	1	0	0	1	0	0	0	Ф	Ф	1
10	1	0	1	0	0	1	Ф	1	1	Ф
11	1	0	1	1	1	0	Ф	0	Ф	1
12	1	1	0	0	0	0	0	Ф	0	Ф
13	1	1	0	1	0	1	0	Ф	Ф	0
14	1	1	1	0	1	0	Ф	0	0	Ф
15	1	1	1	1	1	1	Ф	0	Ф	0

Р3.64. Поскольку $k=2^{m+1}$, можно использовать типовые двоичные счетчики типа КМ155ИЕ7, а для коммутации выходов счетчиков — электронный коммутатор (мультиплексор) 8:1 типа КМ155КР7 (рис. Р3.27). Для устранения ложных выбросов, которые могут возникнуть в схеме в результате опасных сочетаний, используем стробирующий вход мультиплексора x_1 . Задержка стробирующих импульсов T' относительно входных импульсов T должна быть не меньше $t_3=2t_{3\text{сч}}+t_{3\text{ком}}=2 \cdot 24+20=68$ нс.

Р3.65. Используем двоично-десятичный счетчик типа КМ155ИЕ2, комбинационное устройство и электронный коммутатор 8:1 типа КМ155КР7 (рис. Р3.28). В КЦУ реализуются булевы функции: $y_0=Q_1Q_2$; $y_1=Q_2Q_3$; $y_2=Q_2$; $y_3=Q_2Q_3 \vee Q_2Q_3$;

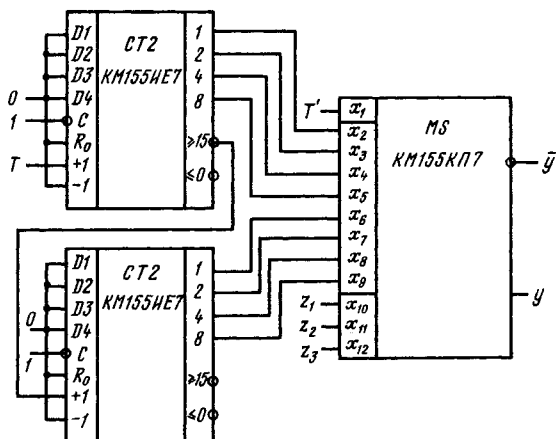


Рис. Р3.27. Схема управляемого делителя частоты

$y_4 = Q_1 \bar{Q}_4$; $y_5 = \bar{Q}_2$; $y_6 = \overline{Q_2 Q_3 \vee Q_2 Q_3}$; $y_7 = \overline{Q_2 Q_4}$. Схема КЦУ может быть выполнена на двух элементах 2И-2ИЛИ-НЕ типа КМ155ЛР1 (один корпус), четырех элементах 2И-НЕ типа КМ155ЛА3 (один корпус) и трех инверторах типа К155ЛН1 (1/2 корпуса). Для защиты от опасных состязаний используем стробирующие импульсы T' по входу x_1 электронного коммутатора, задержанные относительно тактовых импульсов T на время $t_3 \geq t_{3\text{сч}} + t_{3\text{КЦУ}} + t_{3\text{ком}} \approx 100 + 20 + 20 = 140$ нс. Для получения на выходе y последовательности y_i необходимо подать на входы z_3, z_2, z_1 двоичный код числа i .

Р3.66. Можно использовать схему типа рис. Р3.28 со счетчиком типа КМ155ИЕ7 и электронным коммутатором 8:1 типа КМ155КП7. В КЦУ реализуются следующие функции: $y_0 = \bar{Q}_1 Q_2 \vee Q_1 \bar{Q}_2$; $y_1 = \bar{Q}_2 \bar{y}_2 \vee Q_2 y_2$; $y_2 = \bar{Q}_1 \bar{Q}_3 \vee \bar{Q}_1 Q_3$; $y_3 = \bar{Q}_1$; $y_4 = 1$; $y_5 = \bar{Q}_3$; $y_6 = \overline{Q_2 Q_3 \vee Q_2 Q_3}$; $y_7 = \bar{Q}_2$. Схема КЦУ может быть выполнена на четырех элементах 2И-2ИЛИ-НЕ типа

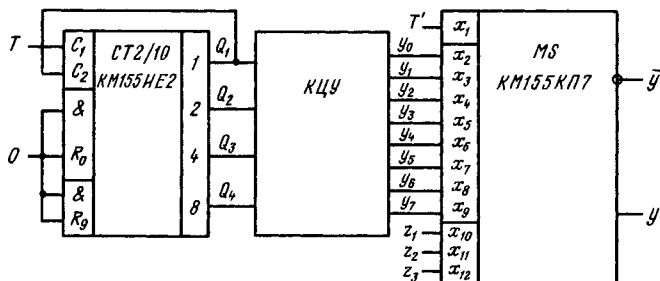


Рис. Р3.28. Схема многофункционального автомата

КМ155ЛР1 (два корпуса). Для исключения опасных состояний задержка стробирующих импульсов T' относительно тактовых импульсов T должна составлять $t_3 \geq t_{3\text{сч}} + t_{3\text{кцу}} + t_{3\text{ком}} = 24 + 2 \cdot 20 + 20 = 84$ нс. Для получения на выходе y последовательности y_i необходимо подать на входы z_3, z_2, z_1 двоичный код числа i .

Р3.67. Можно использовать двоичный счетчик К155ИЕ5 с модулем счета 16 и мультиплексор К155КП2 [21], причем выходы счетчика следует подключить к информационным входам мультиплексора, а на адресные входы последнего подать команду. При этом к общему выходу устройства (выходу мультиплексора) будет подключаться в зависимости от кода команды выход одного из триггеров счетчика, на котором формируется требуемая последовательность.

Глава 4. ФОРМИРОВАТЕЛИ, ГЕНЕРАТОРЫ И СЕЛЕКТОРЫ ИМПУЛЬСОВ

4.1. ОГРАНИЧИТЕЛИ

Данный раздел содержит задачи и упражнения по ограничителям импульсов на диодах и ТТЛ-элементах. В этот же раздел введены задачи, связанные с работой ограничителей, включенных после разделительной цепи, и фиксаторов уровня [3, 4].

При построении передаточных характеристик диодных ограничителей не следует учитывать внутреннее сопротивление генераторов импульсов и сопротивление нагрузки. В схеме замещения открытый диод представлять линейным сопротивлением R_d . Сопротивление закрытого диода принимать равным бесконечности. При построении временных диаграмм не учитывать инерционность диода.

ЗАДАЧИ И УПРАЖНЕНИЯ

4.1. Построить статические характеристики передачи ограничителей $u_{\text{вых}} = f(u_{\text{вх}})$, схемы которых приведены на рис. 4.1.

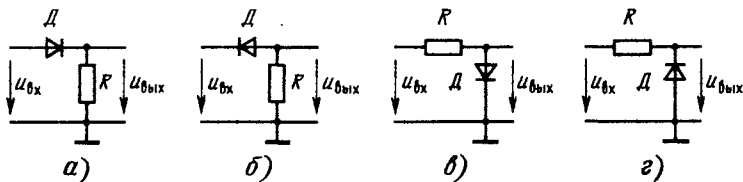


Рис. 4.1. Схемы ограничителей к задаче 4.1

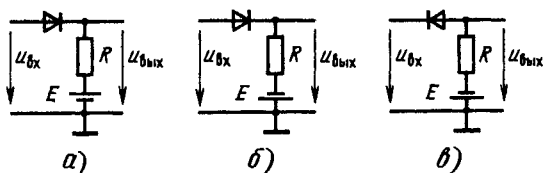


Рис. 4.2. Схемы ограничителей к задаче 4.2

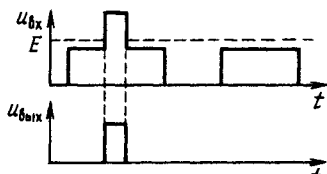


Рис. 4.3. Временная диаграмма к задаче 4.3

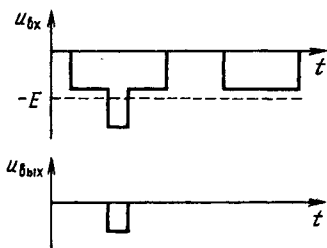


Рис. 4.4. Временные диаграммы к задаче 4.4

4.2. Построить статические характеристики передачи ограничителей $u_{\text{вых}} = f(u_{\text{вх}})$, схемы которых приведены на рис. 4.2.

4.3. Синтезировать схему последовательного диодного ограничителя для осуществления ограничения, показанного на рис. 4.3.

4.4. Синтезировать схему последовательного диодного ограничителя для осуществления ограничения, показанного на рис. 4.4.

4.5. На рис. 4.5 приведена схема ограничителя, включенного после разделительной цепи, и временная диаграмма входного периодического напряжения:

а) построить временную диаграмму выходного напряжения при $R_p \ll R$ без учета влияния паразитных емкостей;

б) построить временную диаграмму выходного напряжения при $R_p = \infty$.

4.6. На рис. 4.6 приведены схемы фиксаторов уровня и временная диаграмма входного периодического напряжения. По-

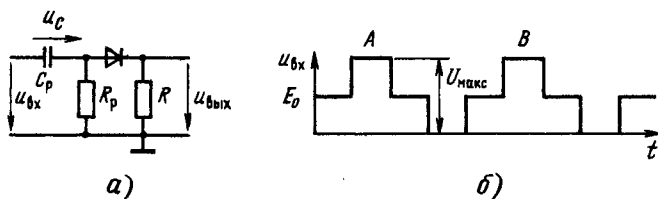


Рис. 4.5. Схема ограничителя и временная диаграмма входного напряжения к задаче 4.5

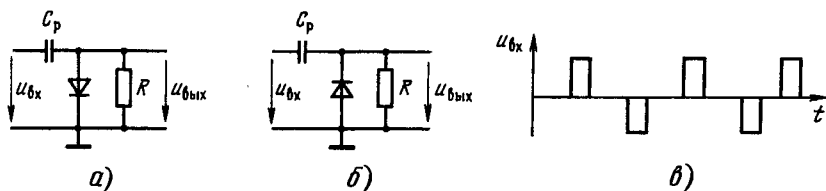


Рис. 4.6. Схемы фиксаторов уровня и временная диаграмма входного напряжения к задаче 4.6

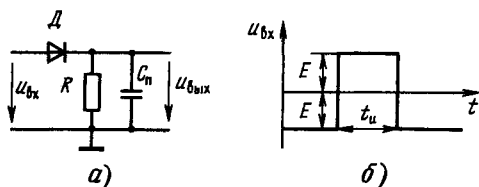


Рис. 4.7. Схема ограничителя и временная диаграмма входного напряжения к задаче 4.7

строить временные диаграммы выходного напряжения без учета влияния паразитных емкостей.

4.7. На вход ограничителя (рис. 4.7,а) поступают импульсы напряжения прямоугольной формы (рис. 4.7,б). Построить временную диаграмму выходного напряжения и определить длительности фронта и среза выходных импульсов. Параметры элементов схемы: $R = 10 \text{ кОм}$, $C_n = 100 \text{ пФ}$, сопротивление диода $R_d = 20 \text{ Ом}$, $t_n = 1 \text{ мкс}$.

4.8. На вход ограничителя (рис. 4.8,а) поступают импульсы напряжения прямоугольной формы (рис. 4.8,б). Построить временную диаграмму выходного напряжения и определить длительности фронта и среза выходных импульсов. Параметры элементов схемы: $R = 10 \text{ кОм}$, $C_n = 100 \text{ пФ}$, сопротивление диода $R_d = 20 \text{ Ом}$, $t_n = 3 \text{ мкс}$.

4.9. На рис. 4.9 приведена схема устройства для формирования напряжения прямоугольной формы из синусоидального напряжения, в которой ТТЛ-элементы используются в качестве ограничителей:

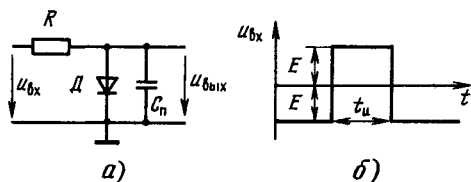


Рис. 4.8. Схема ограничителя и временная диаграмма входного напряжения к задаче 4.8

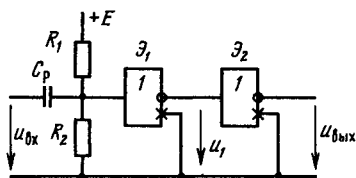


Рис. 4.9. Схема формирователя к задаче 4.9

а) построить временную диаграмму напряжения на выходе первого элемента $u_1(t)$ по временной диаграмме входного напряжения и передаточной характеристике ТТЛ-элемента без учета инерционности элементов схемы;

б) для какой цели используется элемент $\mathcal{E}_2$?

в) пояснить процесс нормализации фронтов сигнала при его прохождении через цепочку логических элементов.

4.2. ФОРМИРОВАТЕЛИ ИМПУЛЬСОВ НАПЯЖЕНИЯ ПРЯМОУГОЛЬНОЙ ФОРМЫ

Данный раздел содержит задачи и упражнения по формирователям импульсов напряжения прямоугольной формы и вспомогательным устройствам на интегральных ТТЛ-элементах [2—4].

В формирователе прямоугольных импульсов из перепадов напряжения (рис. 4.10) длительность формируемых импульсов определяется соотношением

$$t_n \cong t_{\text{з.ср}}(m+1), \quad (4.1)$$

где $t_{\text{з.ср}}$ — задержка в одном элементе; m — число элементов в цепи задержки.

При использовании элементов НЕ (И-НЕ; ИЛИ-НЕ) число элементов m должно быть четным.

При использовании в формирователе импульсов интегрирующей цепи задержки (рис. 4.11, а), основные соотношения следующие:

длительность формируемых импульсов при $R \ll R_{\text{кр}}$

$$t_n \cong RC \ln \frac{U^1}{U_{\text{пор}}}, \quad (4.2)$$

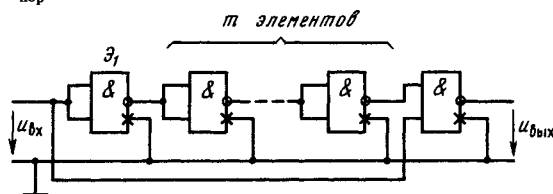
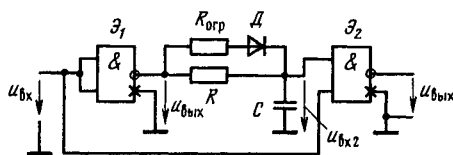
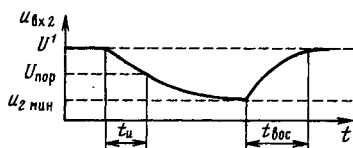


Рис. 4.10. Схема формирователя прямоугольных импульсов из перепадов напряжения



а)



б)

Рис. 4.11. Схема формирователя прямоугольных импульсов с интегрирующей цепью задержки

где U^1 — уровень, соответствующий логической единице; $U_{\text{пор}}$ — пороговое значение входного напряжения логического элемента (напряжение, при котором элемент переходит из одного состояния в другое); время восстановления исходного состояния

$$t_{\text{вос}} \cong 3R_{\text{огр}} C; \quad (4.3)$$

минимальное значение напряжения на входе элемента $\mathcal{E}_2$

$$u_{\text{вх } 2 \text{ мин}} = \frac{R}{R + R_6} (E_{\text{п}} - u_{\text{бэм}}), \quad (4.4)$$

где R_6 — сопротивление в цепи базы многоэмиттерного транзистора элемента $\mathcal{E}_2$; $E_{\text{п}}$ — напряжение источника питания; $u_{\text{бэм}}$ — напряжение между базой и эмиттером многоэмиттерного транзистора при открытом эмиттерном переходе ($u_{\text{бэм}} \cong 0,8 \text{ В}$);

минимально допустимое значение сопротивления

$$R_{\text{мин}} = \frac{U^1}{nI_{1 \text{ макс}}}, \quad (4.5)$$

где n — коэффициент разветвления по выходу элемента $\mathcal{E}_1$;

$$I_{1 \text{ макс}} \cong \frac{E_{\text{п}} - u_{\text{бэм}}}{R_6}.$$

Максимальное значение сопротивления

$$R_{\text{макс}} = R_{\text{кр}} = \frac{U_{\text{пор}}}{E_{\text{п}} - u_{\text{бэм}} - U_{\text{пор}}} R_6. \quad (4.6)$$

Формула (4.6) пригодна для схемы на элементах с характеристикой передачи, не имеющей участка с коэффициентом усиления $k \cong 1$ (рис. 4.12, а). При наличии участка $k \cong 1$ в формулу (4.6) вместо $U_{\text{пор}}$ нужно подставить U' (рис. 4.12, б).

В задачах используются элементы серий 155 и 134. При необходимости формирования импульсов с более короткими фронтами следует использовать ТТЛ-элементы с диодами Шотки (см. приложение 1).

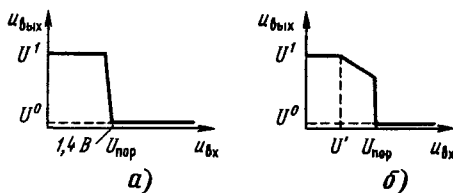


Рис. 4.12. Передаточные характеристики ТТЛ-элементов

ЗАДАЧИ И УПРАЖНЕНИЯ

4.10. Выбрать серию ТТЛ-элементов и определить их число в схеме, изображенной на рис. 4.10 и предназначенной для формирования импульсов отрицательной полярности из перепадов напряжения. Параметры выходных импульсов: $t_n = 0,9$ мкс; $t_f \leq 100$ нс. Для ориентировочного выбора элементов считать, что длительность фронтов, создаваемых логическим элементом t_f , не превышает среднего времени задержки элемента $t_{3\text{cp}}$.

4.11. Как следует изменить схему, приведенную на рис. 4.10, чтобы она была пригодна для формирования импульсов положительной полярности? Определить параметры схемы для формирования импульсов длительностью $t_n = 0,9$ мкс и $t_f = 20$ нс.

4.12. В формирователе прямоугольных импульсов из перепадов напряжения (рис. 4.11, а), собранном на элементах серии 134, увеличили в два раза сопротивление резистора R . При этом $2R < R_{кр}$. Как изменятся временные диаграммы $u_{вх2}(t)$, $u_{вых}(t)$ и длительность формируемых импульсов t_n (качественно)?

4.13. Решить задачу 4.12 при условии $R < R_{кр} < 2R$.

4.14. В формирователе прямоугольных импульсов из перепадов напряжения (рис. 4.11) увеличили в два раза емкость конденсатора C . Как изменятся временные диаграммы $u_{вх2}(t)$, $u_{вых}(t)$ и длительность формируемых импульсов t_n ?

4.15. В формирователе прямоугольных импульсов (см. рис. 4.11) произошло короткое замыкание диода. Как при этом изменится длительность формируемых импульсов?

4.16. В формирователе прямоугольных импульсов из перепадов напряжения (рис. 4.13) произошло короткое замыкание конденсатора C . Какой вид при этом будет иметь временная диаграмма $u_{вых}(t)$. Считать $t_3^{10} = t_3^{01} = t_{3\text{cp}}$.

4.17. В формирователе прямоугольных импульсов (рис. 4.13), собранном на элементах серии 133, увеличили в два раза сопротивление резистора R . При этом $2R < R_{кр}$. Как изменятся временные диаграммы $u_{вх2}(t)$, $u_{вых}(t)$ и длительность формируемых импульсов t_n (качественно)?

4.18. Как качественно изменится временная диаграмма $u_{вх2}(t)$ в задаче 4.17, если в качестве элемента Э₂ использовать элемент серии 134, а сопротивление резистора R оставить прежним? Сопротивление в цепи базы многоэмиттерного

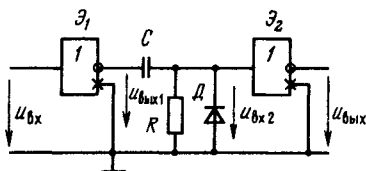


Рис. 4.13. Схема формирователя прямоугольных импульсов из перепадов напряжения

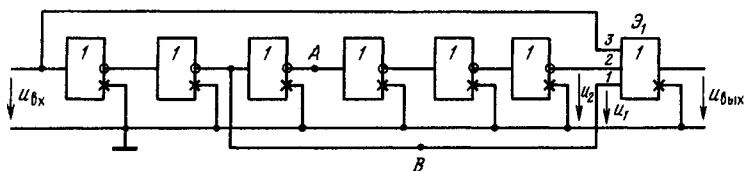


Рис. 4.14. Схема формирователя к задачам 4.20—4.22

транзистора у элементов серии 134 приблизительно в 10 раз больше, чем у элементов серии 133.

4.19. На вход формирователей импульсов из перепадов напряжения (см. рис. 4.10, 4.11, 4.13) поступил импульс, длительность которого $t_{ивх}$ меньше длительности импульса $t_{ивых}$, формируемого устройствами в обычном режиме работы, когда $t_{ивх} > t_{ивых}$. Какова при этом будет длительность выходного импульса? Для этого случая построить временные диаграммы напряжений во всех точках схемы. Считать, что у элементов $t_3^{01} = t_3^{10} = t_{3cp}$.

4.20. На вход схемы, приведенной на рис. 4.14, подаются импульсы положительной полярности длительностью $t_n = 0,5$ мкс. Построить временную диаграмму выходного напряжения для случая, когда в схеме используются ТТЛ-элементы с $t_{3cp} = 100$ нс. Указать область применения рассматриваемой схемы.

4.21. Как изменится временная диаграмма выходного напряжения в задаче 4.20 при обрыве цепи А (рис. 4.14).

4.22. Как изменится временная диаграмма выходного напряжения в задаче 4.20, если вход 1 элемента Э₁ заземлить, а цепь В оборвать?

4.23. Разработать схему устройства, служащего для формирования коротких прямоугольных импульсов отрицательной полярности по фронту и срезу входного импульса большой длительности. Привести временные диаграммы, поясняющие работу устройства.

4.24. Логическое устройство должно управляться при помощи ключа. Приведенная на рис. 4.15 схема обладает тем недостатком, что во время переключения возможен дребезг

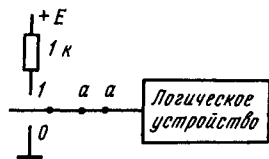


Рис. 4.15. Схема устройства к задаче 4.24

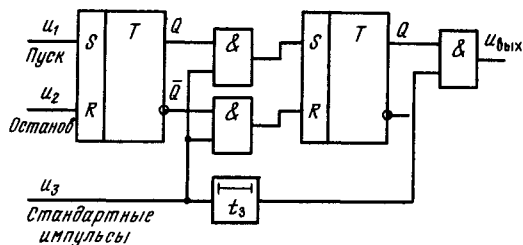


Рис. 4.16. Схема устройства для запуска и остановки серии стандартных импульсов

контактов, приводящий к ложному срабатыванию логического устройства. Предложить схему, устраняющую ложное срабатывание при этом. Схема должна включаться между ключом и логическим устройством (точки $a-a$).

4.25. На рис. 4.16 приведена схема устройства, служащая для запуска и остановки серии стандартных импульсов с помощью импульсов, которые могут приходить в произвольные моменты времени. Пояснить работу устройства временными диаграммами.

4.3. ГЕНЕРАТОРЫ ИМПУЛЬСОВ

Данный раздел содержит задачи и упражнения по генераторам импульсов прямоугольной формы на интегральных ТТЛ-элементах и по генераторам напряжения пилообразной формы [2, 4].

В ждущем мультивибраторе (рис. 4.17) длительность генерируемых импульсов определяется соотношением

$$t_n \cong RC \ln \frac{U^1}{U_{\text{пор}}}, \quad (4.7)$$

где U^1 — уровень, соответствующий логической единице; $U_{\text{пор}}$ — пороговое значение входного напряжения логического элемента; время восстановления исходного состояния

$$t_{\text{вос}} = 3R_{\text{огр}}C; \quad (4.8)$$

минимальное значение напряжения на входе элемента $\mathcal{E}_2$

$$u_{\text{вх } 2 \text{ мин}} = \frac{R}{R + R_6} (E_{\text{п}} - u_{\text{бэм}}), \quad (4.9)$$

где R_6 — сопротивление в цепи базы многоэмиттерного транзистора элемента $\mathcal{E}_2$; $E_{\text{п}}$ — напряжение источника питания; $u_{\text{бэм}}$ — напряжение между базой и эмиттером многоэмиттерного транзистора при открытом эмиттерном переходе ($u_{\text{бэм}} \cong 0,8 \text{ В}$); минимально допустимое сопротивление

$$R_{\text{мин}} = \frac{U^1}{nI_{1 \text{ макс}}}, \quad (4.10)$$

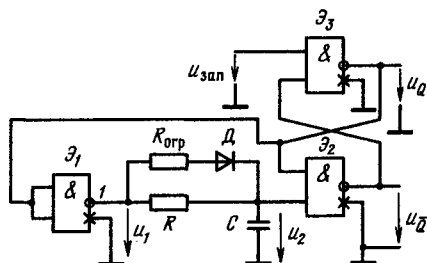


Рис. 4.17. Ждущий мультивибратор

где n — коэффициент разветвления по выходу элемента $\mathcal{E}_1$;

$$I_{1 \text{ макс}} \cong \frac{E_{\text{п}} - u_{\text{бэм}}}{R_6};$$

Максимальное сопротивление

$$R_{\text{макс}} = R_{\text{кр}} \cong \frac{U_{\text{пор}}}{E_{\text{п}} - u_{\text{бэм}} - U_{\text{пор}}} R_6. \quad (4.11)$$

В случае использования цепи задержки на m логических элементах вместо интегрирующей цепи длительность генерируемых импульсов определится формулой

$$t_{\text{н}} \cong m t_{\text{з ср}}. \quad (4.12)$$

Основные соотношения для генераторов напряжения пилообразной формы (ГПН):

коэффициент нелинейности

$$\gamma = \frac{\left| \frac{du_C}{dt} \right|_{\text{нач}} - \left| \frac{du_C}{dt} \right|_{\text{кон}}}{\left| \frac{du_C}{dt} \right|_{\text{нач}}} = \frac{|i_{C \text{ нач}}| - |i_{C \text{ кон}}|}{|i_{C \text{ нач}}|} = \frac{\Delta i_C}{i_{C \text{ нач}}}; \quad (4.13)$$

амплитуда пилообразного напряжения

$$U_m \cong (i_{C \text{ нач}} T_p) / C, \quad (4.14)$$

где i_C — ток заряда или разряда конденсатора; u_C — напряжение на конденсаторе; T_p — длительность рабочего хода.

Для ГПН, построенного на операционном усилителе (ОУ), γ будет порядка $1/k_0$, где k_0 — коэффициент усиления ОУ.

ЗАДАЧИ И УПРАЖНЕНИЯ

4.26. В ждущем мультивибраторе (рис. 4.17), собранном на ТТЛ-элементах, увеличили в два раза сопротивление резистора R . При этом $2R < R_{\text{кр}}$. Как качественно изменятся временные диаграммы напряжений $u_2(t)$, $u_Q(t)$ и длительность генерируемых импульсов $t_{\text{н}}$?

4.27. В ждущем мультивибраторе (рис. 4.17), собранном на ТТЛ-элементах, увеличили в два раза емкость конденсатора C . Как изменятся временные диаграммы $u_2(t)$, $u_Q(t)$ и длительность генерируемых импульсов $t_{\text{н}}$?

4.28. В ждущем мультивибраторе (рис. 4.17), собранном на ТТЛ-элементах, произошло короткое замыкание диода. Как при этом изменится длительность генерируемых импульсов $t_{\text{н}}$?

4.29. Во сколько раз можно изменять длительность генерируемых импульсов в ждущем мультивибраторе (рис. 4.17), изменяя сопротивление резистора R . В схеме используются

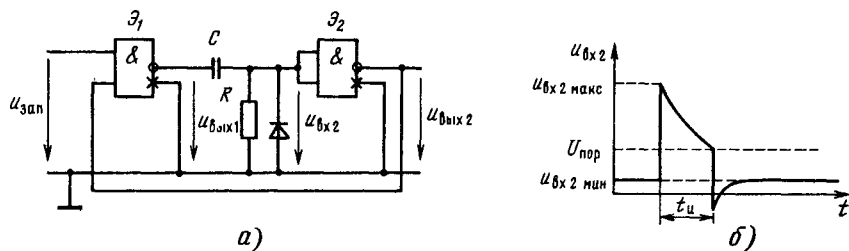


Рис. 4.18. Ждущий мультивибратор к задаче 4.34

элементы серии 155, для которых $R_6 = 4 \text{ кОм}$, $u_{бэм} = 0,8 \text{ В}$, $U_{пор} = 1,4 \text{ В}$, $n = 10$. Считать, что у используемых элементов $U^1 = 3,5 \text{ В}$.

4.30. Решить задачу 4.29 при условии, что в качестве элемента $\mathcal{E}_1$ используется элемент серии 155, а в качестве элемента $\mathcal{E}_2$ — элемент серии 134, у которого $R_6 = 40 \text{ кОм}$. Остальные параметры, необходимые для решения задачи, не отличаются от указанных в 4.29.

4.31. В ждущем мультивибраторе (рис. 4.17) отключили диод. Может ли при этом измениться длительность генерируемых импульсов?

4.32. Можно ли в схеме мультивибратора (рис. 4.17), собранного на ТТЛ-элементах, исключить инвертор $\mathcal{E}_1$, присоединив вход RC-цепи к выходу $\bar{Q}$?

4.33. На вход мультивибратора (рис. 4.17) поступил импульс, длительность которого $t_{и\text{ зап}}$ больше длительности импульса $t_{и}$, генерируемого мультивибратором в обычном режиме, когда $t_{и\text{ зап}} < t_{и}$. Какова будет при этом временная диаграмма $u_Q(t)$ и $u_{\bar{Q}}(t)$?

4.34. В ждущем мультивибраторе (рис. 4.18) увеличили в два раза сопротивление резистора R . При этом $2R < R_{кр}$. Как изменятся временные диаграммы $u_{вх2}(t)$, $u_{вых2}(t)$ и длительность генерируемых импульсов $t_{и}$ (качественно)?

4.35. В ждущем мультивибраторе (см. рис. 4.18) во время запуска произошло короткое замыкание конденсатора C . Какой вид будут иметь временные диаграммы $u_{вх2}(t)$ и $u_{вых2}(t)$?

4.36. На рис. 4.19 приведена схема ждущего мультивибратора на ТТЛ-элементах:

а) построить временные диаграммы сигналов, снимаемых с выходов Q и $\bar{Q}$, для нечетного числа элементов НЕ. Учитывать только задержку, создаваемую элементами НЕ;

б) может ли мультивибратор (см. рис. 4.19) генерировать импульсы, длительность которых меньше длительности запускающих?

в) будет ли работать мультивибратор при четном числе элементов НЕ?

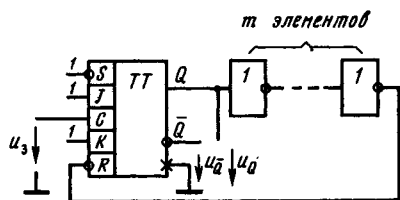


Рис. 4.19. Ждущий мультивибратор к задаче 4.36

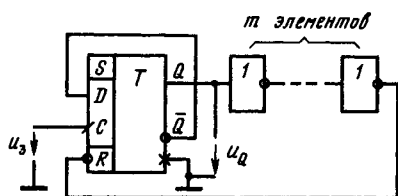


Рис. 4.20. Ждущий мультивибратор к задаче 4.37

4.37. На рис. 4.20 приведена схема ждущего мультивибратора на ТТЛ-элементах:

а) построить временные диаграммы сигналов, снимаемых с выходов Q и $\bar{Q}$, для нечетного числа элементов НЕ. Учитывать только задержку, создаваемую элементами НЕ;

б) может ли мультивибратор (рис. 4.20) генерировать импульсы, длительность которых меньше длительности запускающих?

в) будет ли работать мультивибратор при четном числе элементов НЕ?

4.38. На рис. 4.21 приведена схема простейшего генератора пилообразного напряжения (ГПН):

а) пояснить работу ГПН временными диаграммами. Указать назначение всех элементов схемы;

б) определить амплитуду пилообразного напряжения U_m и коэффициент нелинейности γ при следующих параметрах схемы: $E_k = 100$ В, $E_\phi = 20$ В, $R_k = 10$ кОм, $C = 10^4$ пФ, $t_{н\text{ вх}} = 10$ мкс;

в) как влияют значения C , R_k и $t_{н\text{ вх}}$ на временную диаграмму выходного напряжения?

г) каким образом (схемно) можно уменьшить коэффициент нелинейности γ ?

д) каким образом (схемно) можно увеличить коэффициент использования напряжения источника питания E_k ?

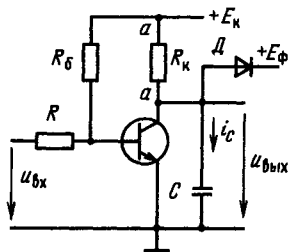


Рис. 4.21. Простейший генератор пилообразного напряжения

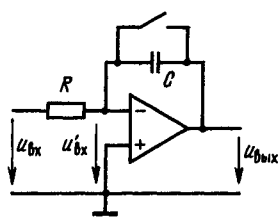


Рис. 4.22. Генератор пилообразного напряжения на операционном усилителе

4.39. На рис. 4.22 приведена схема ГПН на операционном усилителе (ОУ), где электронный ключ, предназначенный для разряда конденсатора C , представлен механическим ключом. На вход ГПН подаются отрицательные импульсы напряжения длительностью $t_{н\text{ вх}}$:

а) пояснить принцип работы ГПН и назначение всех элементов схемы;

б) построить временные диаграммы, поясняющие работу устройства;

в) пояснить, за счет чего в данной схеме обеспечивается высокая линейность (малое значение γ);

г) определить амплитуду выходного напряжения U_m при следующих параметрах элементов схемы: $U_{н\text{ вх}} = 5\text{ В}$, $R = 20\text{ кОм}$, $C = 5 \cdot 10^3\text{ пФ}$, $t_{н\text{ вх}} = T_{\text{раб}} = 100\text{ мкс}$;

д) оценить коэффициент нелинейности γ при отсутствии шунтирующих сопротивлений. В схеме используется ОУ с $k_o = 2 \cdot 10^3$;

е) к чему приводит наличие в реальной схеме сопротивлений, шунтирующих элементы схемы?

4.4. СЕЛЕКТОРЫ ИМПУЛЬСОВ

Данный раздел содержит задачи и упражнения по составлению принципиальных и функциональных схем селекторов импульсов различного типа [4]. При построении селекторов импульсов использовать устройства, рассмотренные в предыдущих параграфах данной главы.

ЗАДАЧИ И УПРАЖНЕНИЯ

4.40. Составить схему диодного амплитудного селектора импульсов, выделяющего импульсы, амплитуда которых больше E_1 (рис. 4.23).

4.41. Составить функциональную схему амплитудного селектора импульсов, выделяющего импульсы максимальной амплитуды. Устройство должно нормально функционировать

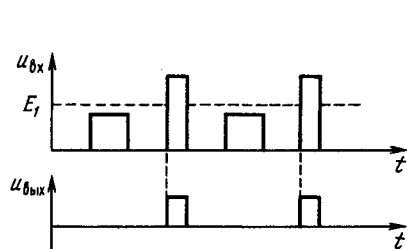


Рис. 4.23. Временные диаграммы к задаче 4.40

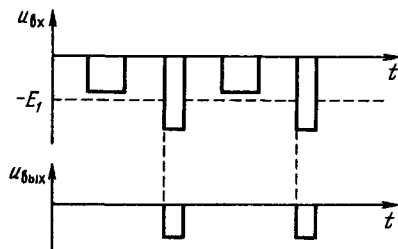


Рис. 4.24. Временные диаграммы к задаче 4.42



Рис. 4.25. Временная диаграмма к задаче 4.43

при изменении в определенных пределах амплитуды входных импульсов. Пояснить работу устройства временными диаграммами.

4.42. Составить схему диодного амплитудного селектора импульсов, выделяющего импульсы, амплитуда которых больше $|E_1|$ (рис. 4.24).

4.43. Составить функциональную схему селектора импульсов, выделяющего импульсы, амплитуда которых не превышает уровень E_1 (рис. 4.25). Работу селектора пояснить временными диаграммами.

4.44. Составить функциональную схему селектора импульсов максимальной длительности. Устройство должно формировать выходной импульс, если длительность входного больше $t_{\text{н мин}}$.

4.45. Составить функциональную схему селектора импульсов минимальной длительности. Селектор должен формировать выходной импульс, если длительность входного меньше $t_{\text{н мин}}$.

4.46. Составить функциональную схему устройства, формирующего короткие импульсы, задержанные относительно входных на время, пропорциональное значению регулируемого опорного напряжения u_p . Устройство такого типа можно использовать при построении селекторов импульсов различных типов.

ОТВЕТЫ И РЕШЕНИЯ

Р4.1. Статические характеристики передачи ограничителей приведены на рис. Р4.1.

Р4.2. Статические характеристики передачи ограничителей приведены на рис. Р4.2.

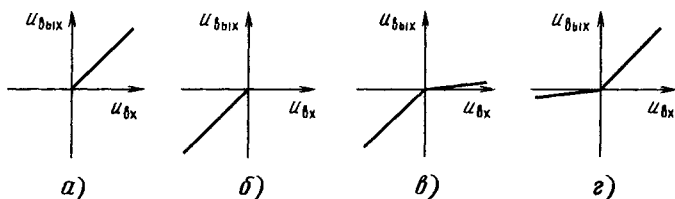


Рис. Р4.1. Статические характеристики передачи ограничителей к задаче 4.1

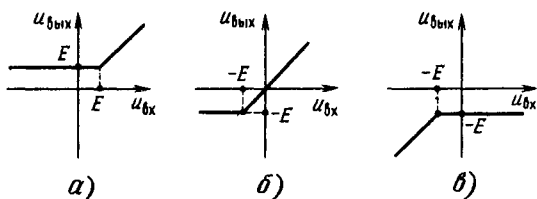


Рис. P4.2. Статические характеристики передачи ограничителей к задаче 4.2

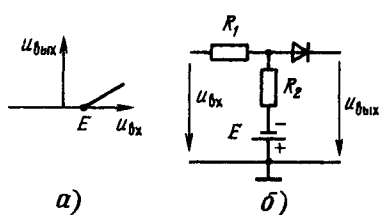


Рис. P4.3. Схема ограничителя к задаче 4.3

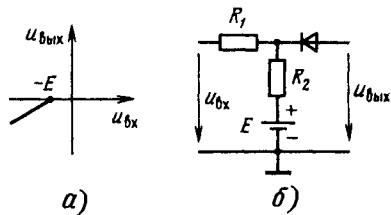


Рис. P4.4. Схема ограничителя к задаче 4.4

P4.3. Характеристика передачи должна иметь вид, приведенный на рис. P4.3,а. Схема ограничителя показана на рис. P4.3,б.

P4.4. Характеристика передачи должна иметь вид, показанный на рис. P4.4,а. Схема ограничителя приведена на рис. P4.4,б.

P4.5. а) При $R_p \ll R$ динамическое смещение практически отсутствует и напряжение на разделительном конденсаторе $u_c \cong E_0$. На выход будут поступать только положительные импульсы (рис. P4.5);

б) при $R_p = \infty$ конденсатор заряжается до напряжения $U_{\max}$ и диод не открывается, поэтому $u_{\text{вых}} = 0$.

P4.6. В фиксаторе уровня, схема которого приведена на рис. 4.6,а, осуществляется фиксация уровня сверху на нулевом уровне (рис. P4.6,а), в другой схеме (рис. 4.6,б)—снизу на нулевом уровне (рис. P4.6,б).

P4.7. Временная диаграмма выходного напряжения приведена на рис. P4.7:

$$\text{а) } U_m = \frac{E}{R_d + R} R \cong E; \quad \text{б) } t_{\phi} = 3 \frac{R_d R}{R_d + R} C \cong 3 R_d C, \quad t_{\phi} = 0,006 \text{ мкс};$$

$$\text{в) } t_{\text{ср}} = 3RC, \quad t_{\text{ср}} = 3 \text{ мкс}.$$

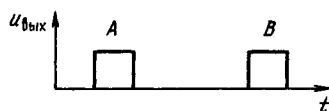


Рис. P4.5. Временная диаграмма к задаче 4.5

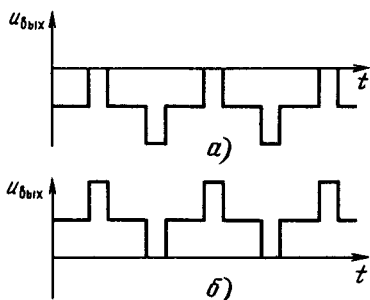


Рис. Р4.6. Временные диаграммы к задаче 4.6

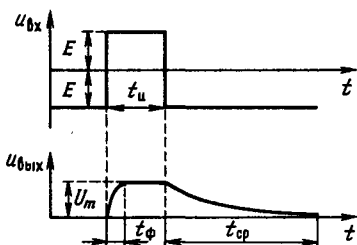


Рис. Р4.7. Временные диаграммы к задаче 4.7

Р4.8. Временная диаграмма выходного напряжения приведена на рис. Р4.8:

$$t_{\phi} = RC \ln \frac{u_c(\infty) - u_c(t_1)}{u_c(\infty) - u_c(t_2)} = RC \ln \frac{E + E}{E} = RC \ln 2;$$

$$t_{\phi} = 0,69 \text{ мкс}; \quad t_{\text{ср}} = 3RC; \quad t_{\text{ср}} = 3 \text{ мкс}.$$

Р4.9. а) Временная диаграмма $u_1(t)$ показана на рис. Р4.9; б) элемент $\mathcal{E}_2$ используется для повторного ограничения и усиления, что приводит к сокращению длительности фронтов сигнала;

в) после прохождения двух-трех элементов фронты сигнала не сокращаются. Их длительность определяется только инерционными свойствами элементов, т. е. происходит нормализация фронтов.

Р4.10. Серия элементов выбирается исходя из длительности фронтов выходных импульсов и минимальной потребляемой мощности. Выбираем ТТЛ-элементы серии 134—134ЛБ1 (в одном корпусе четыре элемента 2И-НЕ) (см. приложение 1).

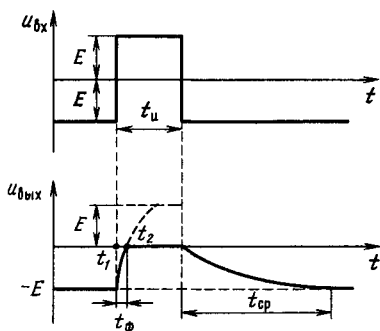


Рис. Р4.8. Временные диаграммы к задаче 4.8

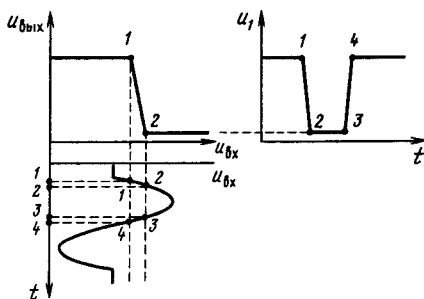


Рис. Р4.9. Временные диаграммы к задаче 4.9

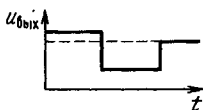


Рис. P4.10. Временная диаграмма к задаче 4.13

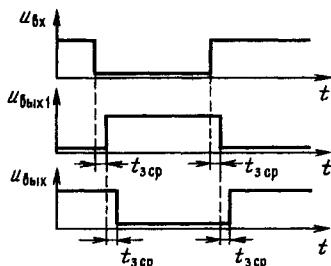


Рис. P4.11. Временные диаграммы к задаче 4.16

Число элементов m в цепи задержки определяется из формулы (4.1): $m=8$. Необходимое число элементов $N=m+2$, $N=10$.

P4.11. Необходимо подключить к выходу элемента $\mathcal{E}_2$ инвертор, в качестве которого использовать элемент серии 155 (ЛАЗ или ЛН1), у которого $t_{зср}=20$ нс и $t_{\phi} \leq 20$ нс. Остальные данные такие же, как и в задаче 4.10.

P4.12. $u_{вх\ 2\ мин}$ увеличится, так как

$$u_{вх\ 2\ мин} = \frac{R}{R+R_6} (E_{\pi} - u_{бэм}).$$

Длительность формируемых импульсов t_{π} увеличится, так как

$$t_{\pi} \cong RC \ln \frac{U^1}{U_{пор}}.$$

Форма выходных импульсов не изменится.

P4.13. $u_{вх\ 2\ мин}$ и t_{π} изменятся так же, как и в задаче 4.12. Форма выходного импульса при увеличении R в два раза изменится. Уровни выходного напряжения до и после импульса будут различными (рис. P4.10), так как $2R > R_{\kappa p}$.

P4.14. Длительность формируемых импульсов t_{π} увеличится в два раза, так как $t_{\pi} \cong RC \ln (U^1/U_{пор})$. Время восстановления $t_{вос}$ увеличится в два раза, так как $t_{вос} \cong 3R_{огр}C$, $u_{вх\ 2\ мин}$ не изменится. Форма выходного импульса останется прежней.

P4.15. Длительность формируемых импульсов уменьшится. Длительность импульсов до короткого замыкания диода $t_{\pi 1} \cong RC \ln (U^1/U_{пор})$. Длительность импульсов после короткого замыкания диода $t_{\pi 2} \cong R_{эв}C \ln (U^1/U_{пор})$, где $R_{эв} \cong R_{огр}R/(R_{огр}+R)$. Изменение длительности формируемых импульсов определится соотношением $t_{\pi 1}/t_{\pi 2} = R/R_{эв}$.

P4.16. Временная диаграмма $u_{вых}(t)$ приведена на рис. P4.11.

P4.17. $u_{вх\ 2\ мин}$ увеличится, так как $u_{вх\ 2\ мин} = \frac{E_{\pi} - u_{бэм}}{R+R_6} R$.

Напряжение $u_{вх\ 2\ макс}$ (рис. P4.12) увеличится за счет увеличения $u_{вх\ 2\ мин}$. Длительность формируемых импульсов t_{π} увеличится за счет увеличения постоянной времени цепи заряда конденсатора C и за счет увеличения $u_{вх\ 2\ макс}$.

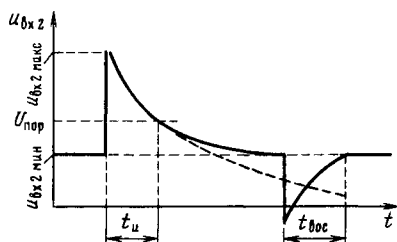


Рис. P4.12. Временная диаграмма к задаче 4.17

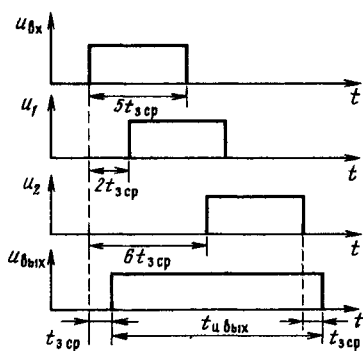


Рис. P4.13. Временные диаграммы к задаче 4.20

P4.18. Напряжения $u_{bx2 \text{ мин}}$, $u_{bx2 \text{ макс}}$ и t_n будут изменяться по тем же причинам, что и в задаче 4.17 (см. P4.17). Так как R_6 у элемента серии 134 в 10 раз больше, чем у элементов серии 133, то абсолютное изменение $u_{bx2 \text{ мин}}$ резко уменьшится. По этой причине резко уменьшится изменение $u_{bx2 \text{ макс}}$. Длительность формируемых импульсов t_n будет увеличиваться в основном за счет увеличения постоянной времени цепи заряда конденсатора C .

P4.19. $t_{н \text{ вых}} = t_{н \text{ вх}}$.

P4.20. Построение временной диаграммы $u_{вых}(t)$ приведено на рис. P4.13. Устройство используется для удлинения входных импульсов.

P4.21. Временные диаграммы при обрыве цепи A приведены на рис. P4.14.

P4.22. Временные диаграммы при обрыве цепи B и заземлении входа 1 элемента $\mathcal{E}_1$ приведены на рис. P4.15.

P4.23. На рис. P4.16 приведена схема формирователя импульсов по фронту и срезу входного импульса.

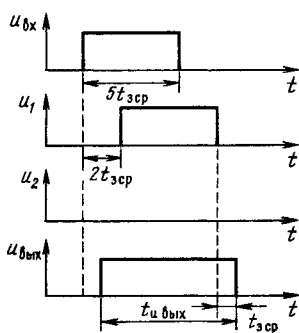


Рис. P4.14. Временные диаграммы к задаче 4.21

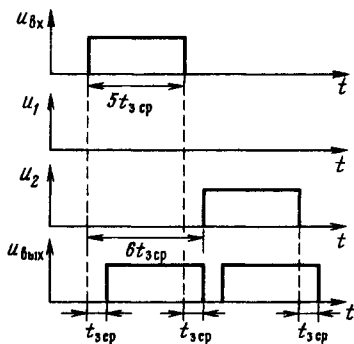


Рис. P4.15. Временные диаграммы к задаче 4.22

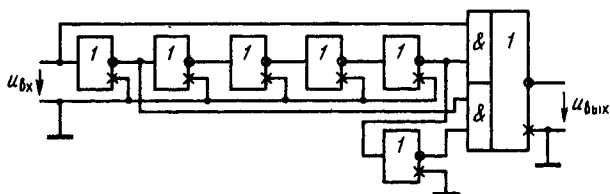


Рис. Р4.16. Схема формирователя импульсов по фронту и срезу входного импульса

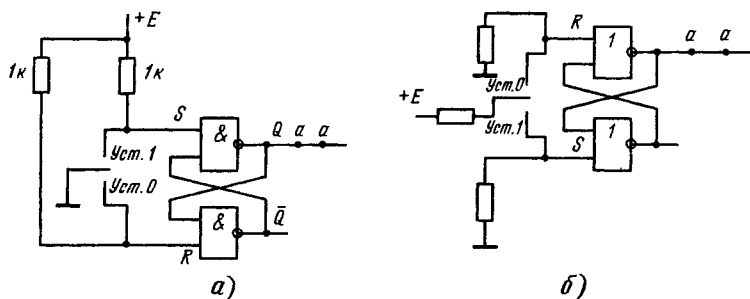


Рис. Р4.17. Схемы для устранения ложного срабатывания при дребезге контактов

Р4.24. На рис. Р4.17 приведены схемы на элементах И-НЕ (а) и на элементах ИЛИ-НЕ (б), устраняющие ложное срабатывание при дребезге контактов.

Р4.25. Временные диаграммы, поясняющие работу устройства при $t_{3\text{ср}}=0$, приведены на рис. Р4.18. В рассматриваемом устройстве всегда $t_{н\text{вых}}=t_{н\text{вх}}$.

Р4.26. При увеличении сопротивления резистора R в два раза $u_{2\text{мин}}$ увеличится, так как $u_{2\text{мин}} \cong \frac{R}{R+R_6} (E_{п} - u_{бэм})$. Длитель-

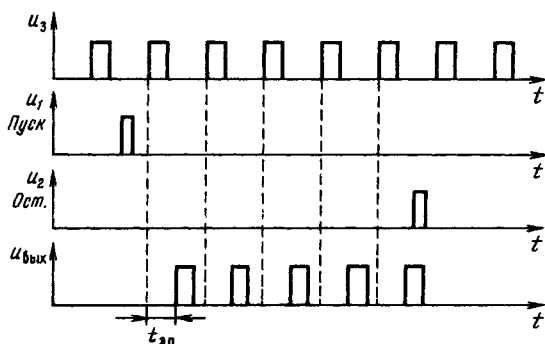


Рис. Р4.18. Временные диаграммы к задаче 4.25

ность генерируемых импульсов $t_{и}$ увеличится в два раза, так как $t_{и} \cong RC \ln \frac{U^1}{U_{пор}}$. Время восстановления $t_{вос}$ останется прежним, так как $t_{вос} \cong 3R_{огр} C$.

P4.27. Длительность генерируемых импульсов $t_{и}$ увеличится в два раза, так как $t_{и} \cong RC \ln \frac{U^1}{U_{пор}}$. Время восстановления $t_{вос}$ увеличится в два раза, так как $t_{вос} \cong 3R_{огр} C$, $u_{2\text{ мин}}$ останется неизменным.

P4.28. Длительность генерируемых импульсов до пробоя диода определяется выражением $t_{и1} \cong RC \ln \frac{U^1}{U_{пор}}$. Длительность генерируемых импульсов после пробоя диода

$$t_{и2} = R_{экв} C \ln \frac{U^1}{U_{пор}},$$

где $R_{экв} \cong R_{огр} R / (R_{огр} + R)$, следовательно, $t_{и1} / t_{и2} = R / R_{экв}$.

$$\begin{aligned} \text{P4.29. } R_{мин} &= \frac{U^1}{nI_{1\text{ макс}}}; I_{1\text{ макс}} \cong \frac{E_{п} - u_{бэм}}{R_6}; R_{макс} = \frac{U_{пор} R_6}{E_{п} - u_{бэм} - U_{пор}}; \frac{t_{и\text{ макс}}}{t_{и\text{ мин}}} = \\ &= \frac{R_{макс}}{R_{мин}} = \frac{U_{пор} n (E_{п} - u_{бэм})}{(E_{п} - u_{бэм} - U_{пор}) U^1}, \quad \frac{t_{и\text{ макс}}}{t_{и\text{ мин}}} = 6. \end{aligned}$$

$$\text{P4.30. } R_{мин} = \frac{U^1}{nI_{1\text{ макс}}}; I_{1\text{ макс}} \cong \frac{E_{п} - u_{бэм}}{R_{61}},$$

где R_{61} — сопротивление в цепи базы многоэмиттерного транзистора элемента серии 155;

$$R_{макс} \cong \frac{U_{пор}}{E_{п} - u_{бэм} - U_{пор}} R_{62},$$

где R_{62} — сопротивление в цепи базы элемента серии 134 ($\mathcal{E}_2$);

$$\frac{t_{и\text{ макс}}}{t_{и\text{ мин}}} = \frac{R_{макс}}{R_{мин}} = \frac{U_{пор} n (E_{п} - u_{бэм})}{(E_{п} - u_{бэм} - U_{пор}) U^1} \frac{R_{6,2}}{R_{61}},$$

$$\frac{t_{и\text{ макс}}}{t_{и\text{ мин}}} = 60.$$

P4.31. Если при отключении диода конденсатор C не будет успевать заряжаться до значения U^1 к моменту прихода очередного запускающего импульса, то длительность генерируемых импульсов уменьшится.

P4.32. Если исключить инвертор $\mathcal{E}_1$, то полученная схема (рис. P4.19,а) при $u_{зап} = U^1$ будет обладать двумя устойчивыми состояниями равновесия, т. е. не будет работать как ждущий мультивибратор.

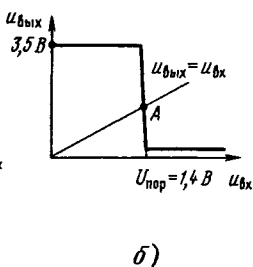
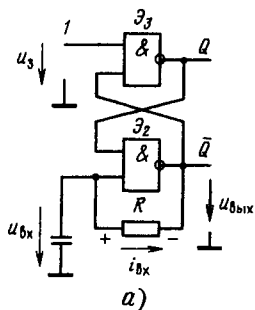


Рис. Р4.19. Схема и определение режима элемента $\mathcal{E}_2$ к задаче 4.32

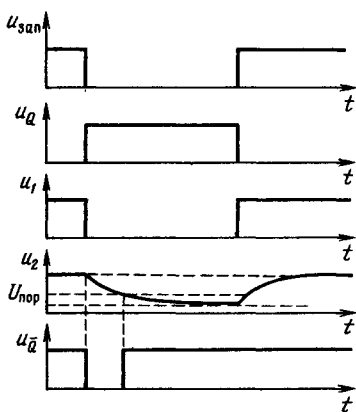


Рис. Р4.20. Временные диаграммы к задаче 4.33

Состояние $Q=0$, $\bar{Q}=1$ создается так же, как в исходной схеме. Но возможно такое исходное состояние, при котором $Q=1$. При $Q=1$ и $R=0$ $u_{вх}=u_{вых}\cong U_{пор}$ (рис. Р4.19,б). При наличии резистора R $u_{вых}=u_{вх}-i_{вх}R\cong U_{пор}-i_{вх}R < U_{пор}$. Этим напряжением элемент $\mathcal{E}_3$ оказывается открытым и $Q=1$. При подаче отрицательного запускающего импульса схема не опрокидывается.

Р4.33. Временные диаграммы без учета задержек, создаваемых логическими элементами, приведены на рис. Р4.20.

Р4.34. $u_{вх\ 2\ мин}$ увеличится, так как $u_{вх\ 2\ мин} = \frac{E_{п}-u_{бэм}}{R+R_6} R$; $u_{вх\ 2\ мин}$ увеличится за счет увеличения напряжения на конденсаторе C в исходном состоянии ($u_{вх\ мин}$). Длительность генерируемых импульсов увеличится за счет увеличения постоянной времени цепи заряда конденсатора C и за счет увеличения $u_{вх\ 2\ мин}$.

Р4.35. Временные диаграммы приведены на рис. Р4.21.

Р4.36. а) При нечетном числе элементов НЕ в исходном состоянии $Q=0$, так как при $Q=1$ на входе R будет сигнал 0 и триггер установится в состояние 0 ($Q=0$). При подаче запускающего импульса триггер перейдет в новое состояние ($Q=1$) после окончания импульса на входе C , так как при $J=K=1$ триггер работает, как T -триггер. Временные диаграммы приведены на рис. Р4.22;

б) может, так как формирование импульса происходит после окончания запускающего;

в) не будет.

Р4.37. а) При нечетном числе элементов НЕ в исходном состоянии $Q=0$, так как при $Q=1$ на входе R будет сигнал

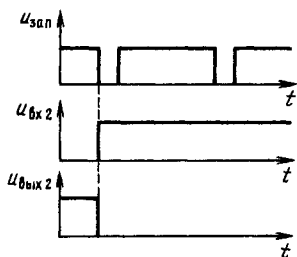


Рис. P4.21. Временные диаграммы к задаче 4.35

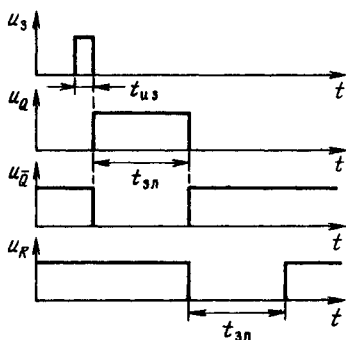


Рис. P4.22. Временные диаграммы к задаче 4.36

0 и триггер установится в состояние 0 ($Q=0$). При подаче запускающего импульса триггер перейдет в новое состояние ($Q=1$), так как он работает как T -триггер (выход Q соединен со входом D). Временные диаграммы приведены на рис. P4.23;

б) может, так как D -триггер имеет динамический вход C ;

в) не будет.

P4.38. а) Временные диаграммы, поясняющие работу ГПН, приведены на рис. P4.24;

$$\text{б) } U_m \cong \frac{i_{C \text{ нач}} T_p}{C} \cong \frac{E_k t_{мвх}}{R_k C} = \frac{100 \cdot 10 \cdot 10^{-6}}{10^4 \cdot 10^4 \cdot 10^{-12}} = 10 \text{ В.}$$

Определим γ . Подставляя в (4.13) значения $i_{C \text{ нач}} \cong E_k / R_k$, $i_{C \text{ кон}} = (E_k - U_m) / R_k$, получаем $\gamma = U_m / E_k = 0,1$, $\gamma = 10\%$;

в) увеличение емкости конденсатора C приводит к уменьшению скорости заряда конденсатора, следовательно, к уменьшению U_m . При этом уменьшается значение γ . Увеличение сопротивления R_k приводит к уменьшению U_m , так как

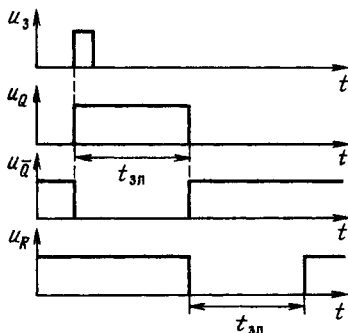


Рис. P4.23. Временные диаграммы к задаче 4.37

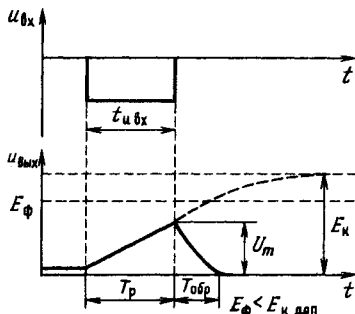


Рис. P4.24. Временные диаграммы к задаче 4.38

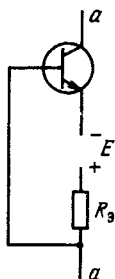


Рис. P4.25. Стабилизатор тока

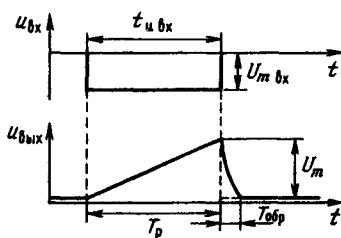


Рис. P4.26. Временные диаграммы к задаче 4.39

уменьшается ток заряда конденсатора. При этом уменьшается значение γ . Увеличение $t_{н.вх}$ приводит к увеличению U_m , так как увеличивается время заряда конденсатора ($T_p = t_{н.вх}$). При этом увеличивается значение γ ;

г) включить вместо R_x между точками $a—a$ стабилизатор тока (рис. P4.25);

д) включить вместо R_x стабилизатор тока и уменьшить E_x . Источник E_ϕ при этом не нужен.

P4.39. б) Временные диаграммы, поясняющие работу ГПН, приведены на рис. P4.26;

в) высокая линейность обеспечивается за счет большого усиления операционного усилителя, приводящего к высокой стабилизации тока заряда конденсатора;

г) $U_m = U_{m.вх} T_p / (RC) = 5 \cdot 100 \cdot 10^{-6} / (20 \cdot 10^3 \cdot 5 \cdot 10^3 \cdot 10^{-12}) = 5 \text{ В}$;

д) при отсутствии шунтирующих сопротивлений γ порядка $1/k_0 = 1/2 \cdot 10^3$, т. е. $\gamma = 0,05\%$;

е) шунтирующее сопротивление ухудшает линейность (увеличивает γ). Особенно сильно влияет на линейность сопротивление, подключаемое параллельно конденсатору C (сопротивление замкнутого ключа). По этой причине целесообразно ключ выполнять на полевых транзисторах.

P4.40. На рис. P4.27 приведена одна из возможных схем амплитудного селектора импульсов.

P4.41. На рис. P4.28 приведены один из возможных вариантов функциональной схемы устройства (а) и временные диаграммы, поясняющие его работу (б).

P4.42. На рис. P4.29 приведена одна из возможных схем амплитудного селектора импульсов.

P4.43. На рис. P4.30 приведены одна из возможных функциональных схем селектора импульсов (а) и временные диаграммы, поясняющие работу устройства (б). Расширитель и элемент задержки необходимы для устранения ложных импульсов, вызыва-

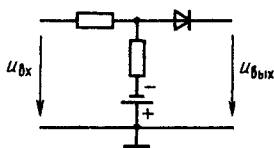


Рис. Р4.27. Амплитудный селектор импульсов к задаче 4.40

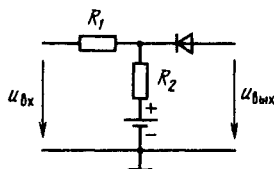


Рис. Р4.29. Амплитудный селектор импульсов к задаче 4.42

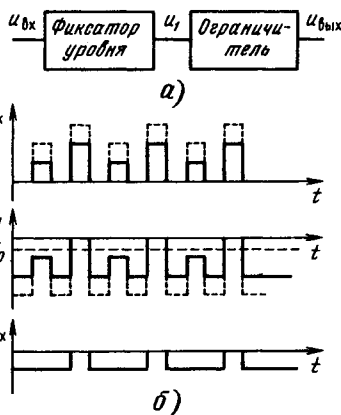


Рис. Р4.28. Функциональная схема амплитудного селектора импульсов к задаче 4.41

емых конечной длительностью фронтов входных импульсов и запаздыванием расширенного относительно входного импульса.

Р4.44. На рис. Р4.31 приведены две из возможных функциональных схем селектора импульсов и временные диаграммы, поясняющие работу устройств.

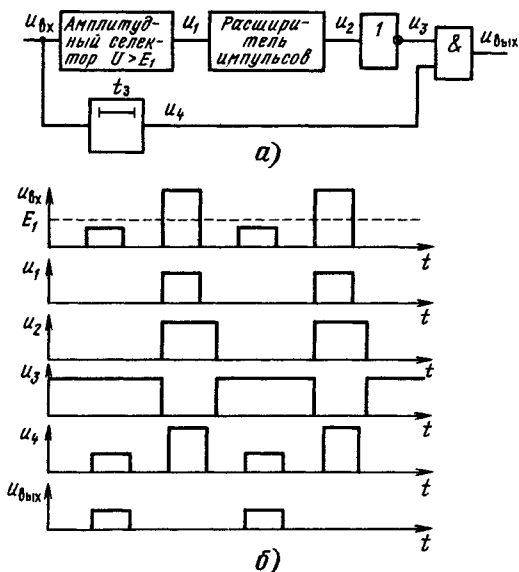


Рис. Р4.30. Функциональная схема амплитудного селектора импульсов к задаче 4.43

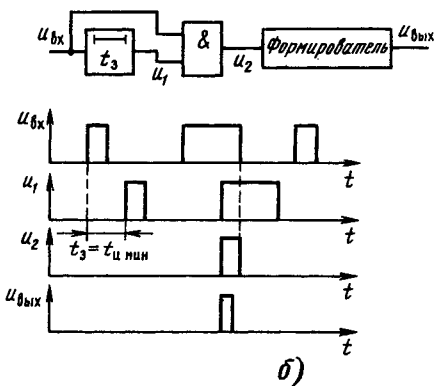
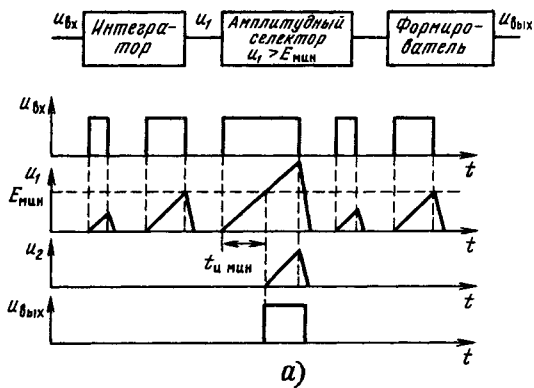


Рис. Р4.31. Функциональные схемы селекторов импульсов к задаче 4.44

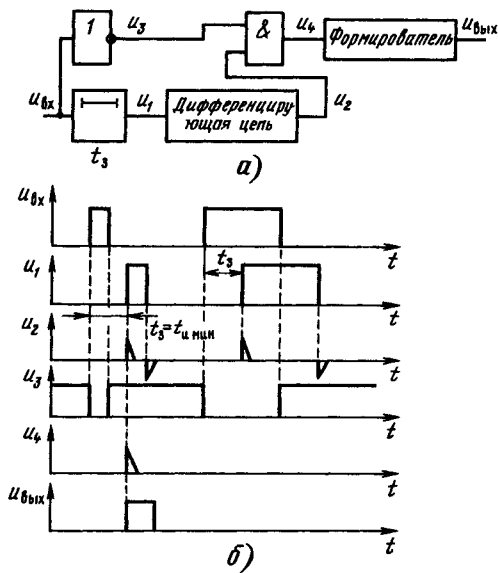


Рис. Р4.32. Функциональная схема селектора импульсов к задаче 4.45

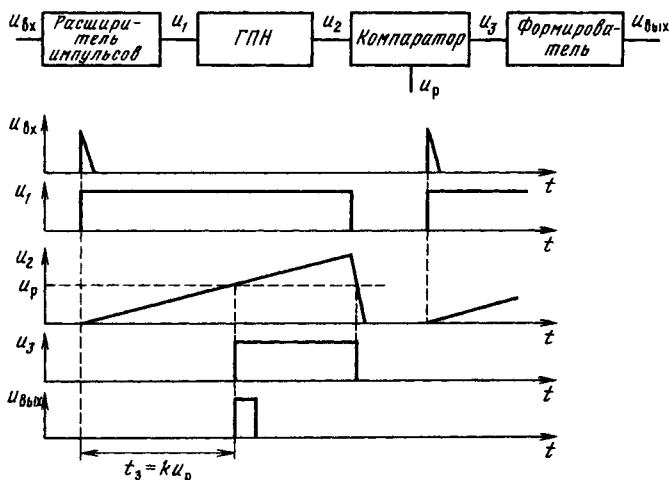


Рис. Р4.33. Функциональная схема устройства формирования задержанных импульсов к задаче 4.46

Р4.45. На рис. Р4.32 приведены одна из возможных функциональных схем селектора импульсов (а) и временные диаграммы, поясняющие работу устройства (б).

Р4.46. На рис. Р4.33 приведены одна из возможных функциональных схем устройства формирования задержанных импульсов (а) и временные диаграммы, поясняющие работу устройства (б).

Глава 5. ПОЛУПРОВОДНИКОВЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА

В данный раздел включены задачи и упражнения, связанные с организацией запоминающих устройств (ЗУ) на полупроводниковых микросхемах статического типа. Большинство задач связано с организацией ЗУ на микросхемах ОЗУ и ППЗУ, условные графические обозначения которых приведены на рис. 5.1. Необходимый справочный материал содержится в табл. 5.1—5.5, приложении 2 и в [20].

Таблица 5.1. Назначение выводов микросхем ОЗУ КМ132РУ8А, КМ132РУ8Б

Обозначение	Назначение
$A_0 \dots A_9$	Адресные входы
$DIO_0 \dots DIO_3$	Вход-выход данных
$\overline{CS}$	Выбор микросхемы
$\overline{WR}/RD$	Сигнал «Запись-считывание»

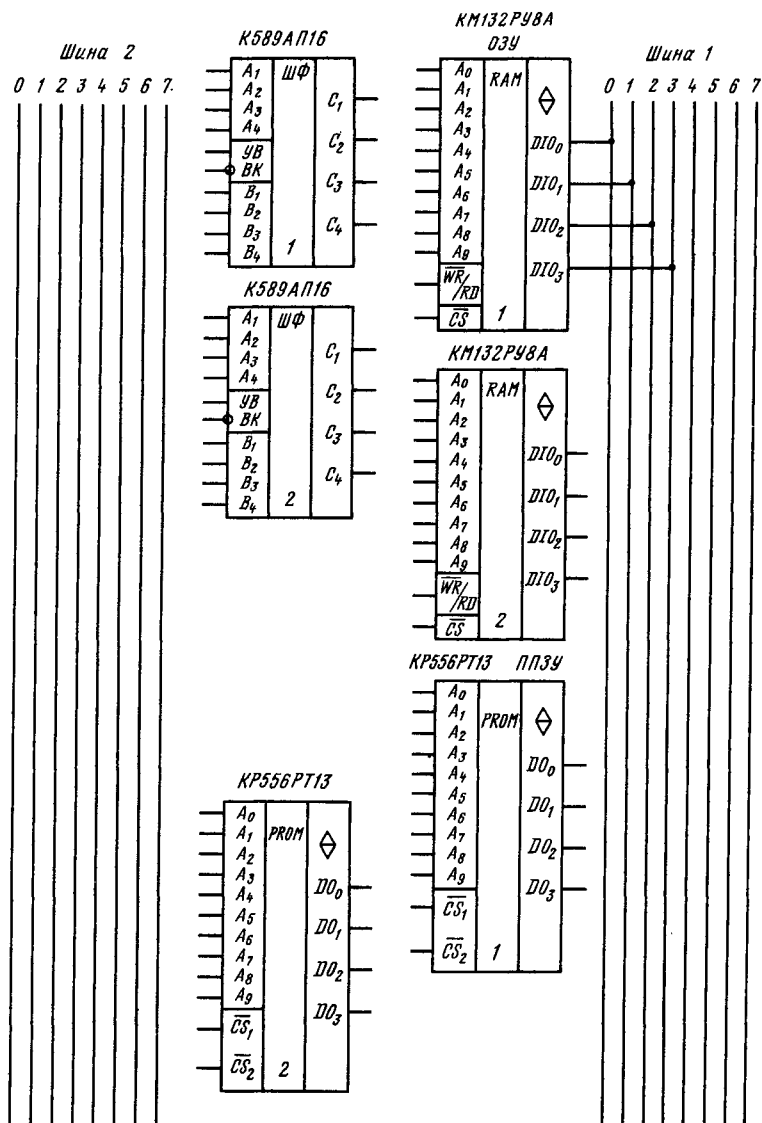


Рис. 5.1. Микросхемы ОЗУ, ППЗУ и шинных формирователей для организации запоминающего устройства

Таблица 5.2. Таблица истинности микросхем ОЗУ КМ132РУ8А, КМ132РУ8Б

$\overline{CS}$	$\overline{WR}/RD$	$A_0 \dots A_9$	$DIO_0 \dots DIO_3$	Режим работы
1	Φ	Φ	R_{off}	Хранение
0	0	A	0	Запись 0
0	0	A	1	Запись 1
0	1	A	Данные в прямом коде	Считывание

Примечание. Φ — безразличное значение сигнала; R_{off} — выходное сопротивление, равное ∞ .

Таблица 5.3. Назначение выводов микросхемы ППЗУ КР556РТ13

Обозначение	Назначение
$A_0 \dots A_9$	Адресные входы
$DO_0 \dots DO_3$	Выход данных
$\overline{CS}_1, \overline{CS}_2$	Выбор микросхемы

Таблица 5.4. Таблица истинности микросхемы ППЗУ КР556РТ13

$\overline{CS}_1$	$\overline{CS}_2$	$A_0 \dots A_9$	$DO_0 \dots DO_3$	Режим работы
M	M	Φ	R_{off}	Хранение
0	0	A	Данные в прямом коде	Считывание

Примечание. MM — любая комбинация, кроме 00; R_{off} — выходное сопротивление, равное ∞ .

Таблица 5.5. Таблица истинности микросхемы шинного формирователя К589АП16

$\overline{BK}$	УВ	Направление передачи	Значение A , B и C	Выходы в выключенном состоянии
0	0	От входа A_i на выход B_i	$B_i = A_i$	C_i
0	1	От входа B_i на выход C_i	$C_i = B_i$	—
1	0	Передача отсутствует	$B_i \text{ — } R_{off}$ $C_i \text{ — } R_{off}$	B_i, C_i
1	1	Передача отсутствует	$B_i \text{ — } R_{off}$ $C_i \text{ — } R_{off}$	B_i, C_i

Примечание. R_{off} — выходное сопротивление, равное ∞ .

ЗАДАЧИ И УПРАЖНЕНИЯ

5.1. а) Для каких целей используются оперативные запоминающие устройства (ОЗУ) и программируемые постоянные запоминающие устройства (ППЗУ)?

б) Какие типы ОЗУ вы знаете?

в) Какие выходные каскады используются в микросхемах статических ОЗУ и ППЗУ для параллельного соединения выводов?

г) Какие выходные каскады используются в шинных формирователях для присоединения к двунаправленной шине?

д) Разрушается ли информация в микросхемах статических ОЗУ и ППЗУ при выключении источника питания, при считывании информации?

е) Как присоединяются микросхемы ОЗУ и ППЗУ с открытым коллекторным выходом к двунаправленной шине?

ж) Что такое время выборки адреса в ОЗУ и ППЗУ?

з) Можно ли к двунаправленной шине присоединить одновременно статическое ОЗУ и ППЗУ?

и) Можно ли использовать ППЗУ для выполнения логических операций?

к) Можно ли использовать ППЗУ для выполнения арифметических операций?

л) В запоминающем устройстве используются микросхемы ОЗУ КМ132РУ8А (см. рис. 5.1). Как определить информационную емкость каждой микросхемы?

5.2. В ЗУ КМ132РУ8А (см. рис. 5.1) необходимо хранить только 512 8-разрядных слов.

а) Как должны быть соединены выводы микросхем ОЗУ?

б) Какое число адресных входов необходимо использовать?

5.3. Как должны быть соединены микросхемы ОЗУ КМ132РУ8А (см. рис. 5.1), каждая из которых может хранить 1024 слова, чтобы можно было хранить 2048 4-разрядных слов? Сколько адресных шин должно иметь ОЗУ?

5.4. Как построить ОЗУ на 1024 12-разрядных слов, используя микросхемы КМ132РУ8А (см. рис. 5.1)?

5.5. Как необходимо соединить микросхемы ОЗУ КМ132РУ8А (см. рис. 5.1), чтобы получить два ОЗУ, каждое емкостью 1024 4-разрядных слова, между которыми можно было бы производить обмен информацией?

5.6. Как должны быть соединены микросхемы (рис. 5.1), чтобы можно было считывать информацию как с ППЗУ 1, так и с ОЗУ 1 на общую двунаправленную шину, а также осуществлять запись информации в ОЗУ?

5.7. Можно ли в устройстве, созданном в результате решения задачи 5.6 (см. Р5.6), осуществлять передачу информации из ППЗУ в ОЗУ и из ОЗУ в ППЗУ?

5.8. Как необходимо соединить между собой две микросхемы ППЗУ (см. рис. 5.1), чтобы можно было хранить 1024 8-разрядных слова? Каждая микросхема КР556РТ13 может хранить 1024 слова.

5.9. Как необходимо соединить между собой две микросхемы ППЗУ (см. рис. 5.1), чтобы можно было хранить 2048

4-разрядных слова? Каждая микросхема может хранить 1024 слова.

5.10. Как необходимо соединить между собой ОЗУ 1 и шинный формирователь (см. рис. 5.1), чтобы можно было производить обмен информацией между ОЗУ и устройствами, присоединенными к двунаправленной шине 2?

5.11. Решить задачу 5.10 для случая хранения 8-разрядных слов.

5.12. В распоряжении проектировщика имеются следующие микросхемы ОЗУ и ППЗУ: К541РУ2А, КМ132РУ8А, К541РУ1А, К541РУ31А, К541РУ3А, КР556РТ12, КР556РТ13, КР556РТ14, КР556РТ18 и КР556РТ16. Все перечисленные микросхемы согласуются с ТТЛ схемами (параметры микросхемы см. в приложении 2).

а) Выбрать микросхемы ОЗУ и ППЗУ для построения ЗУ, содержащего ОЗУ на 1024 8-разрядных слова и ППЗУ на 2048 8-разрядных слова. ОЗУ и ППЗУ должны быть присоединены к общей двунаправленной шине.

б) То же, что и в п. а), но ОЗУ и ППЗУ должны обладать емкостью по 4096 8-разрядных слова.

в) То же, что и в п. а), но ОЗУ и ППЗУ должны обладать емкостью по 8192 8-разрядных слова.

ОТВЕТЫ И РЕШЕНИЯ

Р5.1. л) Количество 4-разрядных слов, хранимых в каждой микросхеме ОЗУ, $N=2^n=2^{10}=1024$, где n — число адресных входов микросхемы. Информационная емкость микросхемы $1024 \cdot 4 = 4096$ бит.

Р5.2. а) Все адресные входы и входы управления микросхемы 1 должны быть соединены с одноименными входами микросхемы 2. Выводы $DIO_0 \dots DIO_3$ микросхемы 1 ОЗУ должны быть соединены с соответствующими двунаправленными шинами с номера 0 по номер 3. Микросхема 2 ОЗУ присоединяется к шинам 4...7.

б) Девять входов ($N=2^9=512$).

Р5.3. Все выводы $DIO_0 \dots DIO_3$ микросхемы 1 должны быть соединены с одноименными выводами микросхемы 2 и присоединены к двунаправленной шине. Адресные входы одной микросхемы $A_0 \dots A_9$ должны быть соединены с одноименными адресными входами другой микросхемы.

Всего адресных шин должно быть 11 ($2^{11}=2048$). Сигнал старшего разряда адреса A_{10} используется совместно с сигналом управления $\overline{CS}$ для формирования сигналов управления, подаваемых на микросхемы 1 и 2 ($\overline{CS}_1, \overline{CS}_2$).

При записи информации в микросхему 1 и чтении информации из нее ($\overline{CS}_1=A_{10}=0$) на вход $\overline{CS}$ микросхемы 2 ($\overline{CS}_2$)

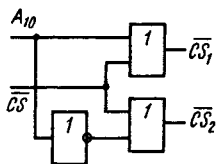


Рис. P5.1. Схема устройства к задаче 5.3

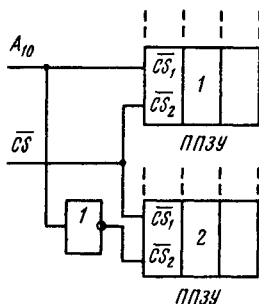


Рис. P5.2. Схема устройства к задаче 5.9

должен подаваться сигнал 1, чтобы она была в режиме хранения (см. табл. 5.2), а при сигнале $A_{10}=1$ наоборот. При сигнале $\overline{CS}=1$, поступающем из устройства управления, оба ОЗУ должны работать в режиме хранения, для чего $\overline{CS}_1=\overline{CS}_2=1$ (рис. P5.1).

P5.4. Необходимо использовать три микросхемы. Все одноименные адресные входы и входы управления микросхем соединить параллельно. Выводы $DIO_0...DIO_3$ присоединить к двунаправленным шинам с номера 0 по номер 11, так, чтобы каждая микросхема была присоединена к четырем шинам, т. е. чтобы каждая микросхема использовалась для хранения четырех из двенадцати разрядов слова.

P5.5. Выводы $DIO_0...DIO_3$ одной микросхемы соединить с одноименными выводами другой и присоединить их к 4-разрядной двунаправленной шине. Для обмена информацией необходимо подавать сигналы управления так, чтобы при считывании информации с одной микросхемы производилась запись в другую. Адресные шины должны быть разными для возможности записи по адресу, отличному от адреса, по которому осуществляется считывание.

P5.6. Выводы $DIO_0...DIO_3$ микросхемы ОЗУ должны быть соединены с соответствующими выводами ППЗУ и присоединены к общей двунаправленной шине. Адресные входы микросхемы ОЗУ должны быть соединены с соответствующими адресными входами микросхемы ППЗУ. При считывании информации из одного ЗУ выходное сопротивление другого должно быть равно бесконечности. Для этого должны подаваться сигналы управления, приведенные в табл. 5.2 и 5.4.

P5.7. Передать информацию из ППЗУ в ОЗУ можно. Для этого необходимо подать соответствующие сигналы на управляющие входы микросхем (см. табл. 5.2, 5.4). При этом информация будет записываться по тому же адресу, что и считывать-

ся. Передать информацию из ОЗУ в ППЗУ нельзя. В ППЗУ информацию после записи изменить нельзя.

Р5.8. Для хранения 1024 8-разрядных слова необходимо одноименные адресные входы и входы управления двух микросхем соединить параллельно. Одну микросхему использовать для хранения младших четырех разрядов слова, а другую — для старших. Выходы микросхем присоединить к 8-разрядной шине.

Р5.9. Все выводы $DO_0 \dots DO_3$ микросхемы 1 должны быть соединены с одноименными выводами микросхемы 2 и присоединены к 4-разрядной шине. Адресные входы одной микросхемы $A_0 \dots A_9$ должны быть соединены с одноименными адресными входами другой микросхемы. Всего адресных шин должно быть 11 ($2^{11} = 2048$). Сигнал старшего разряда адреса A_{10} совместно с сигналом управления $\overline{CS}$, поступающим из устройства управления, служат для формирования сигналов управления $\overline{CS}_1$ и $\overline{CS}_2$ микросхем ПЗУ. При считывании информации из одной микросхемы другая должна находиться в режиме хранения.

Для формирования сигналов управления используется схема, приведенная на рис. Р5.2. При $A_{10} = 0$ и $\overline{CS} = 0$ осуществляется считывание информации с микросхемы 1. При $A_{10} = 1$ и $\overline{CS} = 0$ осуществляется считывание информации с микросхемы 2 (см. табл. 5.4).

Р5.10. К двунаправленной шине 1, к которой присоединено ОЗУ 1, необходимо присоединить выводы шинного формирователя $B_1 \dots B_4$. Выводы $A_1 \dots A_4$ и $C_1 \dots C_4$ присоединить к двунаправленной шине 2 (см. табл. 5.5).

Р5.11. Для хранения 8-разрядных слов необходимо использовать две микросхемы ОЗУ и две микросхемы шинных формирователей. Двунаправленные шины 1 и 2 должны содержать восемь проводников по числу разрядов. Соединение ОЗУ и шинных формирователей осуществляется аналогично тому, как это делалось в задаче 5.10 (см. Р5.10).

Одноименные управляющие и адресные входы микросхем ОЗУ соединяются параллельно. Также параллельно соединяются одноименные управляющие входы шинных формирователей.

Р5.12. а) К общей шине должны быть присоединены две микросхемы ОЗУ КМ132РУ8А либо К541РУ2А и одна микросхема ППЗУ КР556РТ18.

б) К общей шине должны быть присоединены восемь микросхем ОЗУ К541РУ1А и две микросхемы ППЗУ КР556РТ18.

в) К общей шине должны быть присоединены восемь микросхем ОЗУ К541РУ31А и одна микросхема ППЗУ КР556РТ16.

6.1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

В данном разделе кратко разъясняются основные понятия и определения, которые используются в литературе по микропроцессорам. Более подробно с этим материалом можно ознакомиться в [1, 2, 5, 8, 22].

ЗАДАЧИ И УПРАЖНЕНИЯ

6.1. Дать определения понятий «микропроцессор» (МП), «микропроцессорная система» (МПС), «микропроцессорный комплект» (МПК), «микроЭВМ».

6.2. Объяснить понятия «микрооперация», «микрокоманда» и «микропрограмма».

6.3. Каковы отличия однокристальных микропроцессоров от однокристальных микроЭВМ?

6.4. Каковы преимущества и недостатки МП с фиксированной системой команд по сравнению с микропрограммируемым МП?

6.5. Какие характеристики микропроцессора определяют в первую очередь его эффективность при использовании в вычислительных, управляющих и коммутационных системах?

6.6. Объяснить термины «аппаратное», «программное» и «программно-аппаратное» обеспечение микроЭВМ или МПС.

6.7. Что такое «центральный процессорный элемент»?

6.8. Дать определение понятия «шина». Какие шины имеются в микропроцессорах?

6.9. Дать определение понятия «порт».

6.10. Дать определение понятия «интерфейс».

6.11. Назвать и охарактеризовать основные режимы работы МП.

6.12. Указать основные отличия естественной и принудительной адресации команд и области их применения.

6.2. МИКРОПРОЦЕССОР KP580BM80

Центральный процессорный элемент KP580BM80 (сокращенно KP580) представляет собой однокристальный 8-разрядный МП с фиксированной системой команд, в котором совмещены операционное, управляющее устройство и сверхоперативная память [1, 2, 5, 8]. Шина данных — 8-разрядная двунаправленная, адресная шина — 16-разрядная однонаправленная, обеспечивает возможность обращения к памяти, содержащей 64К ячеек.

Сигналы, необходимые для управления МПС, снимаются частично с управляющих выходов МП, частично — с шины данных. МП допускает использование 256 портов ввода и 256 портов вывода.

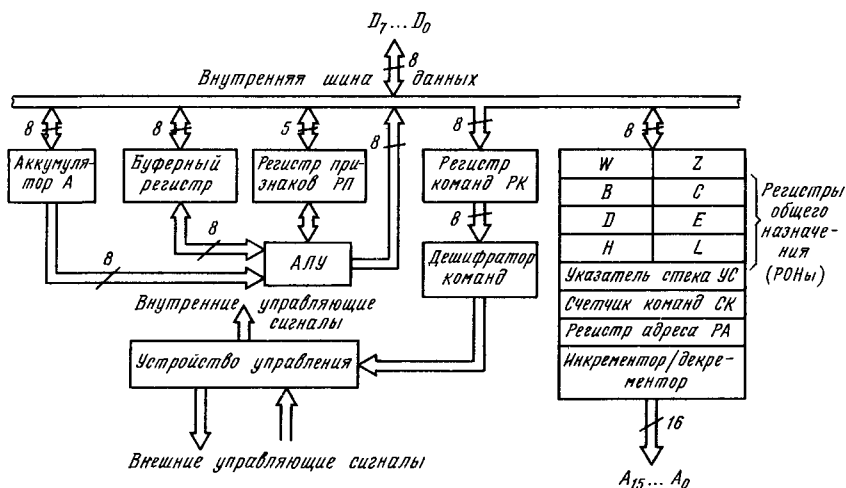


Рис. 6.1. Структурная схема МП КР580ВМ80

Основные характеристики МП: число команд 78 (с учетом модификаций — 244¹), максимальная тактовая частота 2 МГц, время выполнения команд 2...9 мкс, напряжения питания +5, -5, +12 В, число вводов 40. Система команд МП приведена в приложении 3.

ЗАДАЧИ И УПРАЖНЕНИЯ

6.13. На рис. 6.1 показана структурная схема микропроцессора (МП, МР)² КР580ВМ80. Объяснить назначение основных узлов МП: арифметическо-логического устройства (АЛУ, ALU), аккумулятора (А, ACC), регистров общего назначения (РОН, RX), регистра команд (РК, RI), регистра признаков (РП, RF), регистра адресов (РА, MAR), счетчика команд (СК, PC), указателя стека (УС, SP) и др.

6.14. Охарактеризовать режимы (состояния) захвата, прерывания и ожидания в МПС, построенной на базе МП КР580.

6.15. а) Что такое командный цикл, машинный цикл и такт МП КР580. Какова их длительность?

б) Какие машинные циклы реализуются в МП при выполнении команды ввода? Какова длительность этой команды?

¹ Модификацией называется изменение команды без изменения кода операции; например, двумя модификациями одной команды являются MOV A, B и MOV C, E.

² В скобках указаны часто используемые русские и английские обозначения узлов и устройств.

6.16. Каково назначение «слова состояния» МП КР580, когда оно формируется в МП и где фиксируется?

6.17. Расшифровать нижеуказанные байты состояния МП D₇...D₀. Каким типам машинных циклов они соответствуют и какие сигналы на выходах системного контроллера (см. § 8.1) в этих циклах устанавливаются:

а) 10000010; б) 00000000; в) 01000010; г) 00010000; д) 00100011?

6.18. Можно ли утверждать, что длительность выполнения команды в МП КР580 тем больше, чем больше ее формат (число байтов)? Сравнить по числу байтов и длительности команды LXI и XTHL.

6.19. Объяснить термин «модификация команды». Привести примеры модифицированных команд.

6.20. Объяснить значение термина «адресация операнда». В чем существо различных типов адресации: прямой, непосредственной, неявной и косвенной? Привести примеры для МП КР580.

6.21. а) Указать разновидности прямой адресации данных, используемые в МП КР580. Привести примеры команд с прямой адресацией.

б) Какие адреса имеют при прямой адресации регистры МП, порты МПС, ячейки памяти?

6.22. Указать основные разновидности непосредственной адресации данных, используемые в МП КР580. Привести примеры команд с непосредственной адресацией.

6.23. В чем заключается регистровая косвенная адресация данных в МП КР580? Привести примеры команд.

6.24. Какие виды адресации данных используются в следующих командах МП КР580:

- | | | |
|---------------|---------------|--------------|
| а) MOV D, A; | б) ADI CC; | в) SUB M; |
| г) MOV C,M; | д) ANA H; | е) JMP 00BC; |
| ж) CALL FF00; | з) MVI A, FF; | и) XCHG; |
| к) RET, | л) OUT 00; | м) CMP M? |

6.25. Расшифровать коды состояния регистра признаков МП КР580: а) 00000010; б) 10010111; в) 10000011.

Указание. См. [6, с. 29].

6.26. В МП КР580 в исходном состоянии содержимое регистров А, В и регистра признаков (РП): (А)=01100101; (В)=00001110; (РП)=00000010. Каково будет содержимое этих регистров после выполнения каждой из следующих команд:

- | | | |
|------------|--------------|---------------|
| а) ADD B; | б) SUB B; | в) RLC; |
| г) INR B; | д) STA FF00; | е) SUI FA; |
| ж) CPI 63; | з) JMP 0ABC; | и) CALL 012D; |
| к) NOP? | | |

6.27. Назвать и охарактеризовать три способа обмена данными между МПС и внешними устройствами (ВУ).

6.28¹. Составить фрагменты программ в мнемокодах и машинных кодах для следующих операций:

- а) записать константу A0 в ячейку памяти с адресом B02C;
- б) записать содержимое ячейки с адресом 020F в регистр D;
- в) записать содержимое регистра E в ячейку с адресом 1456;
- г) поменять местами содержимое ячеек с адресами 0102 и F1 F2;
- д) поменять местами содержимое регистра B и ячейки с адресом, хранящимся в регистровой паре DE;
- е) записать содержимое регистровой пары BC в ячейки с адресами FF00, FF01;
- ж) записать содержимое ячеек с адресами EE00, EE01 в регистровую пару DE;
- з) обнулить регистры A, B, C, D, E.

6.29. Определить содержимое указанных регистров после выполнения следующих фрагментов программ:

а) MVI A	б) MVI B
2D	54
ADI	MOV A,B
OF	SUI
	22

Регистр A	Регистр A
в) MVI A	г) LXI B
14	B0
SUI	FF
2D	MOV A,B
	ADD C

Регистр A	Регистр A и триггер переполнения CY
д) LXI B	е) LXI H
FF	F0
01	01
MOV A,C	LXI B
ADI	0F
01	88
MOV L,A	DAD B
ADD B	

Регистр L	Регистры H, L
-----------	---------------

6.30. Определить содержимое указанных регистров после выполнения фрагментов программ, если первоначально было: (A)=2F; (B)=D0; (C)=4A; (D)=0A; (E)=F2; (M)=0B; (H)=CE; (L)=E1.

¹ В задачах 6.28—6.39, рассчитанных на первоначальное ознакомление с программированием МП, команды для наглядности записаны побайтно. Все константы записаны в 16-ричном коде.

а) ADD B SUB C ADD D	б) ADD M MOV C, A MVI B 00 DAD B
Регистр A	Регистры H,L
в) ADD E MOV E,A MOV A,D ADI 01 MOV D,A	г) DAD B MOV A,E ADD D
Регистры D, E	Регистры A, H, L
д) MOV A,C ADI FF INR B MOV C,A	е) ADD C MOV L,A INX H
Регистры B, C	Регистры H, L

6.31. Проанализировать фрагменты программ, определить содержимое аккумулятора и время выполнения программ при следующих исходных данных: (A)=00; (C)=A2; (D)=26; ОЗУ [1112]=70; (H)=BC; (L)=EC; тактовая частота $f_T=2$ МГц:

а) 0005 MVI A 06 OE 07 ADD L 08 JNZ 09 OC 0A 00 0B MOV A,C 0C INR A 0D MOV C,A	б) 0013 MOV A,C 14 ADI 15 00 16 JZ 17 1C 18 00 19 LDA 1A 12 1B 11 1C DCR A
в) 0101 JMP 02 09 03 01 04 MOV A,C 05 LDA 06 12 07 11 08 SUB A 09 HLT	г) MVI A AA ANA D ORA C ANI 82 RRC

д) MVI M	е) MOV M,A
01	INR M
LDA	INR L
12	MVI M
11	10
MOV B,M	MOV B,M
XRA B	LDA
RLC	ED
	BC
	ORA B

6.32. Проанализировать фрагменты программ, определить содержимое аккумулятора или порта 05, рассчитать время выполнения программы при тактовой частоте $f_T = 2$ МГц; в исходном состоянии (порт 06)=01:

а) 0000 MVI A	б) 0000 MVI A
01 05	01 0A
02 MVI B	02 MOV C,A
03 00	03 NOP
04 INR B	04 NOP
05 SUB B	05 DCR A
06 JNZ	06 JNZ
07 04	07 03
08 00	08 00
09 MOV A,B	09 MOV A,C
0A OUT	
0B 05	
в) 0000 ANI	г) 0000 MVI A
01 00	01 EE
02 INR A	02 SUI
03 CPI	03 11
04 FF	04 JM
05 JNZ	05 02
06 02	06 00
07 00	07 NOP
08 HLT	08 NOP
	09 NOP
	0A OUT
	0B 05
д) 0103 IN	
04 06	
05 MVI B	
06 64	
07 MVI C	
08 FA	
09 DCR C	

0A JNZ
 0B 09
 0C 01
 0D DCR B
 0E JNZ
 0F 07
 10 01
 11 OUT
 12 05

6.33. Просуммировать операнды, находящиеся в ячейках ОЗУ с адресами FDC1 и FDC2, и результат записать в ячейку ОЗУ с адресом FDC1. Составить программу.

6.34. В ячейке ОЗУ 1F01 хранится число x_1 , в ячейке 1F02 — число x_2 . Необходимо вывести через порт FF большее из чисел. Составить программу.

6.35. Составить программы для вычисления следующих величин¹:

а) $y = (37 - 30/2) : 2 + 11$; б) $y = (61 + 134/2) : 8 - 6$;

в) $y = (24 \cdot 4 - 8 \cdot 2) : 4$; г) $y = (200 : 8 - 1) : 4$.

6.36. Составить программы для вычисления величин z ¹:

а) $z = 2x + 1$; г) $z = (4x - 8y) : 2$;

б) $z = 6x - 5$;

в) $z = 3x + 2$; д) $z = \begin{cases} x - y & \text{при } x \geq 8, \\ x + y & \text{при } x < 8; \end{cases}$

е) $z = \begin{cases} 3x + y & \text{при } x \geq 10, \\ 5x - y & \text{при } x < 10; \end{cases}$

ж) $z = \begin{cases} x + 2y + 1 & \text{при } x > 5, \\ 2x - 3y + 3 & \text{при } x \leq 5; \end{cases}$

з) $z = \begin{cases} x + 1 & \text{при } x \leq 15, \\ x + y & \text{при } 15 < x \leq 30. \end{cases}$

Число x поступает через порт 05, число y — через порт 08, число z должно храниться в ячейке ОЗУ 2233. Считать, что x , y , z , а также результаты промежуточных и окончательных вычислений — это целые числа, находящиеся в пределах $-127 \dots +127$.

6.37. Определить содержимое порта 06 МПС в результате выполнения следующей программы:

0000 IN
 01 05
 02 ADD A
 03 DCR A

¹ Для умножения и деления использовать операции сдвига.

04 CALL
05 00
06 10
07 INR A
08 OUT
09 06
0A HLT

Число a поступает через порт 05. В области памяти, начиная с ячейки 1000, записана подпрограмма вычисления выражения $p = \log_2 x$, где x — число, находящееся в аккумуляторе к моменту вызова подпрограммы.

6.38. Составить программу вычисления величины $z = 1 + \cos a + \cos 2a$, пользуясь подпрограммой вычисления выражения $p = \cos x$, которая записана в области памяти, начиная с ячейки 1100. Число a вводится через порт 05, результат следует вывести через порт 06.

6.39. В памяти МПС, начиная с ячейки 2000, записана подпрограмма перемножения двух чисел, которые берутся из РОНов D и E МП, причем произведение запоминается в E. Пользуясь этой подпрограммой, составить циклическую программу для вычисления выражения

$$z = x + x^2 + x^3 + x^4 + x^5 + x^6,$$

где x — число, вводимое в МПС через порт 11. Результат следует хранить в ячейке памяти 3000.

6.40¹. Составить фрагмент циклической программы, содержащий следующие операции:

- а) непосредственную загрузку константы в регистр МП;
- б) декремент регистра;
- в) организацию цикла при ненулевом результате.

Константа имеет следующие значения: 1) 100; 2) 1000.

Указание. Для варианта 1 использовать регистр В МП, для варианта 2 — регистровую пару ВС.

6.41. Составить для МП КР580 алгоритм и программу умножения двух 8-разрядных положительных двоичных чисел. Оценить время выполнения программы.

6.3. МИКРОПРОЦЕССОР K1810BM86

Микропроцессор K1810BM86 (сокращенно K1810) по сравнению с МП КР580BM80 обладает более высокой производительностью (примерно на порядок), расширенной системой команд (включая команды умножения и деления), возможностью использования большей памяти (до 1М слов) и т. д.

¹ Начиная с задачи 6.40, программы записываются, как правило, на языке ассемблера.

Микропроцессор выполняет операции над 8, 16, 32 и 64-разрядными кодами, над последовательностями (цепочками) байтов и слов, а также над двоично-кодированными десятичными числами. Тактовая частота 5 МГц, напряжение питания +5 В, число выводов корпуса 40.

Терминология в задачах данного раздела заимствована в основном из [5, 7]. Система команд приведена в [5].

ЗАДАЧИ И УПРАЖНЕНИЯ

6.42. а) Указать основные технические характеристики МП K1810BM86 и сравнить их с характеристиками МП KP580BM80.

б) Объяснить, как формируется исполнительный адрес ячейки памяти в МП K1810. Каковы преимущества и недостатки сегментирования памяти?

6.43. Каково адресное пространство ввода-вывода в МП K1810? Как адресуются порты ввода-вывода?

6.44. Указать основное назначение регистров AX, BX, CX, DX МП K1810.

6.45. Указать основное назначение регистров CS, SS, DS и ES МП K1810.

6.46. Указать основное назначение регистров SP, BP, SI и DI МП K1810.

6.47. Указать назначение и формат регистра F МП K1810.

6.48. Можно ли в МП K1810 одновременно (в одном такте) выставить адрес ячейки памяти и данные?

6.49. Каково назначение регистра команд МП K1810?

6.50. Что такое минимальный и максимальный режимы работы МП K1810? Как осуществляется выбор одного из этих режимов?

6.51. Объяснить назначение полей команды МП K1810 (рис. 6.2).

6.52. Определить эффективные адреса EA операндов при $(BX)=1234H$, $(DS)=3100H$, $disp\ H, L=1A33H$ для следующих режимов адресации: а) регистрового; б) регистрового косвенного; в) регистрового относительного. (Считается, что соответствующими командами программы выбирается базовый регистр BX.) Определить также исполнительные адреса, если в качестве сегментного регистра используется регистр DS, причем $(DS)=3100H$.

6.53. Расшифровать следующие команды K1810, заданные машинными кодами, и записать их мнемокоды:

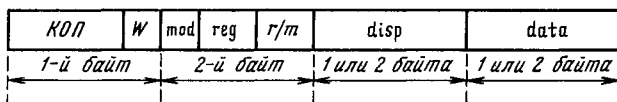


Рис. 6.2. Формат команды МП K1810

- а) 10001001 11000001; б) 10001000 00000000;
 в) 10001011 10011100 10000000 11010001.

6.54. Расшифровать следующие команды K1810:

- а) 00000001 11000001; б) 00000010 11011101;
 в) 00000011 11000111.

6.55. Расшифровать команду K1810, заданную 16-ричным кодом 81C70F30. Определить результат выполнения этой команды при (DI)=11F0H.

6.56. Определить эффективные адреса переходов в командах безусловных переходов при следующих исходных данных: (IP)=1000H; disp H, L=01F2H; (BX)=DE01H, если в командах предусмотрена:

- а) внутрисегментная прямая адресация;
 б) внутрисегментная косвенная адресация с использованием регистра BX и без смещения;
 в) внутрисегментная косвенная адресация с использованием регистра BX и смещения. Составить машинные коды этих команд.

6.57. Определить эффективный и исполнительный адреса в команде межсегментного прямого перехода, если необходимо перейти к сегментному адресу (CS)=B200H и disp H, L=010EH. Составить машинный код этой команды.

6.58. Составить машинные коды команд, обеспечивающих выполнение следующих операций:

- а) (AX)←(BX); г) (CL)←[(SI)+disp H, L];
 б) (BX)←(AX); д) (AX)←00F1H.
 в) (AL)←[(BP)+(DI)];

6.59. Составить машинные коды команд, обеспечивающих выполнение следующих операций:

- а) (AX)←(AX)+(BX); г) (CX)←(CX)+2468H;
 б) (AL)←(AL)+[(BX)+(SI)]; д) [(BX)]←0000.
 в) (BX)←(BX)+[(SI)+disp L];

6.60. Составить для МП K1810 фрагмент программы в мнемокодах для обмена содержимого двух ячеек памяти M1 и M2, находящихся в одном сегменте. Определить время выполнения фрагмента при тактовой частоте синхронизации $f_T = 5$ МГц.

6.61. Составить для МП K1810 фрагмент программы в мнемокодах, обеспечивающий сложение N 16-битовых слов, хранящихся в парах ячеек памяти, начиная с ячейки с меткой BEG. Результат необходимо поместить в ячейку с меткой RES. Рассчитайте время выполнения при тактовой частоте синхронизации $f_T = 5$ МГц и N=100.

6.62. Составить для МП K1810 фрагмент программы в мнемокодах для вычисления восьмиместной булевой функции, заданной в ДСНФ (т. е. суммой минтермов): $y = x_7 \bar{x}_6 x_5 \bar{x}_4 \bar{x}_3 x_2 x_1 x_0 \vee x_7 \bar{x}_6 \bar{x}_5 x_4 \bar{x}_3 x_2 \bar{x}_1 x_0$. Набор булевых пере-

менных периодически вводится в МПС через порт 07 в виде байта D, в котором биты соответствуют переменным $D_7=x_7$, $D_6=x_6$, ..., $D_0=x_0$. Значение функции выводится через порт 05 в виде младшего разряда байта.

Указание. Каждый минтерм булевой функции может быть задан байтом, в котором i -й бит ($i=0, \dots, 7$) равен 0, если x_i входит в минтерм с инверсией, и равен 1 — если без инверсии. Таким образом, первый минтерм задается константой $K_1=10100111=A7H$, а второй — константой $K_2=10010101=95H$.

6.63. Составить для МП K1810 фрагмент программы в мнемокодах для вычисления выражения

$$z=2(XY+T),$$

где X, Y, T — беззнаковые 8-битовые операнды, хранящиеся в ячейках памяти с адресами, находящимися соответственно в регистрах BX, CX и DX. Необходимо организовать также следующую процедуру:

при $z < 350$ — переход по метке PROC1,

при $z \geq 350$ — вывод результата через порт 02.

Указание. Умножение XY следует организовать с помощью команды MUL, умножение на 2 — удвоением результата.

6.4. МИКРОКОНТРОЛЛЕР KM1816BE48

Однокристалльный 8-разрядный микроконтроллер KM1816BE48 (сокращенно МК1816) содержит в своем составе все основные узлы микроЭВМ: АЛУ, УУ, ППЗУ, ОЗУ и интерфейсные схемы. Организация, система команд и средства ввода-вывода МК1816 приспособлены для решения задач управления несложными объектами. МК может работать с тактовой частотой до 6 МГц, выполнен в корпусе БИС с 40 внешними выводами, требует одного источника питания напряжением 5 В (при программировании ППЗУ необходим также источник напряжения 25 В).

Подробное описание МК1816 имеется в [6, 30]. В командах МК1816 используется ряд специфических обозначений, в частности знак # — это префикс (указатель) непосредственного операнда, знак @ — префикс косвенной адресации. Остальные обозначения поясняются в задачах.

ЗАДАЧИ И УПРАЖНЕНИЯ

6.64. Указать основные программно-доступные узлы микроконтроллера МК1816.

6.65. Какова адресуемая емкость памяти МК1816?

6.66. Охарактеризовать внутреннее ОЗУ МК1816: какие ячейки могут использоваться в качестве регистров общего назначения, какие доступны командам с прямой и косвенной адресацией.

6.67. В чем суть, каковы достоинства и недостатки страничной адресации ППЗУ, используемой в МК1816?

6.68. Каково назначение внутреннего таймера МК1816?

6.69. Расшифровать и объяснить следующие команды МК1816, указать виды адресации, используемые в этих командах, составить машинные коды команд:

- | | |
|------------------|----------------|
| а) MOV A, R5; | е) MOVP A, @A; |
| б) MOV A, #0A1H; | ж) MOVX A@R0; |
| в) MOV A, @R1; | з) XCH A, R4; |
| г) MOV A, T; | и) XCH A, @R1; |
| д) MOV @R0,A; | к) MOVP3 A, @A |

6.70. Расшифровать следующие команды МК1816, определить результат их выполнения — содержимое аккумулятора или указанного в команде регистра, а также триггера переноса (признака переноса (C)) при заданных исходных данных:

- а) ADD A,R5 при (A)=B3H, (R5)=16H;
- б) ADD A, #0BN при (A)=AAH;
- в) ADD A, @R0 при (A)=11H; [(R0)]=FBH;
- г) INC R4 при (R4)=4FH;
- д) INC @R1 при [(R1)]=2AH;
- е) DEC A при (A)=4CH;
- ж) RL A при (A)=32H, (C)=0;
- з) RR A при (A)=B3H, (C)=0;
- и) RRC A при (A)=B3H, (C)=0.

6.71. Расшифровать следующие команды МК1816 и определить результат их выполнения при указанных исходных данных:

- а) ANL A, R1 при (A)=2BH, (R1)=3CH;
- б) ANL A, # F0 при (A)=ABH;
- в) ORL A, @R0 при (A)=15H, [(R0)]=4CH;
- г) XRL A, # 61H при (A)=B3H;
- д) CLR A;
- е) CPL A при (A)=B3H.

6.72. Расшифровать следующие команды МК1816:

- а) IN A,P1; г) ANL P2, #0FH;
- б) INS A,BUS; д) OUTL BUS, A.
- в) OUTL P1,A;

6.73. Расшифровать и объяснить следующие команды МК1816 (для вариантов а), б), в), и) составить коды команд):

- а) JMP a при $a_{8...10}=010$, $a_{0...7}=01010100$;
- б) JMPP @ A при (A)=41H;
- в) DINZ R4, a при $a_{0...7}=11100010$;
- г) JC a;
- д) JNC a;
- е) JZ a;
- ж) JTO a;
- з) JNI a;
- и) JB3 a при $a_{0...7}=01101000$.

6.85. Составить программу для МК1816, обеспечивающую обмен содержимым двух ячеек ОЗУ R4 и R33.

6.86. Составить программу для МК1816, обеспечивающую очистку (обнуление) и последующий запуск внутреннего счетчика (таймера), а также переход по метке MET, если счетчик достиг состояния 32.

6.87. На вход T1 МК1816 поступают с интервалами от 20 мкс и более импульсы. Организовать их счет и инкрементировать аккумулятор каждый раз, как содержимое счетчика (таймера) достигнет 255; при этом вывести содержимое аккумулятора через порт P1. Когда содержимое аккумулятора достигнет 128, необходимо его очистить и начать счет сначала.

6.88. В МК1816 должна периодически с интервалом 8 мс выполняться подпрограмма с меткой PROG1. Составить программу, используя внутренний таймер.

6.89. Составить схему МПС на базе МК1816 с внешней оперативной памятью емкостью 256 байт. Какие команды позволяют адресоваться к внешнему ОЗУ? Какие управляющие сигналы при этом используются?

6.90. Составить схему микроконтроллерной системы на базе МК1816 с дополнительной внешней памятью программ емкостью 3 Кбайт, используя три микросхемы ПЗУ емкостью 1 Кбайт каждая. Какие команды позволяют адресоваться к внешней памяти программ? Какие управляющие сигналы при этом используются?

6.91. Составить на базе МК1816 схему микроконтроллерной системы с тремя дополнительными портами ввода-вывода. Какие команды требуются для управления этой схемой?

6.5. МИКРОПРОЦЕССОРНЫЙ КОМПЛЕКТ СЕРИИ K1804

Микропроцессорный комплект K1804 предназначен для построения быстродействующих микропроцессорных систем и ЭВМ с разрядно-модульной организацией и микропрограммным управлением. МПК содержит 4-разрядную

Таблица 6.1. Формат микрокоманды

МК	АМК	BR	P	MS ₂	КПП	MS ₁
		BR ₃ BR ₂ BR ₁ BR ₀	P ₃ P ₂ P ₁ P ₀		I ₈ I ₇ I ₆	
Условное обозначение микрокоманды	Адрес микрокоманды в ЗУ	Адрес следующей микрокоманды	Код управления переходом	Модификатор сдвига	Код приемника результата	Модификатор сдвига
Микро						

Код		Сдвиг вправо	Сдвиг влево
MS_2	MS_1		
0	0		
0	1		
1	0		
1	1		

Рис. 6.4. Коды операций сдвига МП К1804

рядных слов. Если в МПС необходимо вводить информацию из внешних устройств, в микрокоманде должно быть еще одно поле, содержимое которого управляет специальным мультиплексором, коммутирующим источники информации на шину данных. Основная память МПС, в которой хранятся программы решения задач на языке команд, на рис. 6.3 не показана.

Микрокоманда содержит 32 двоичных разряда, причем для поля адреса микрокоманды ВР отведено 4 разряда. Считается, что память микропрограмм содержит 16 ячеек. (Для N ячеек поле BR должно иметь $\lceil \log_2 N \rceil$ разрядов.) Расшифровка полей микрокоманды приведена на рис. 6.4 и в табл. 6.2—6.5.

Таблица 6.2. Коды операций АЛУ

КОП			Операция АЛУ
I_5	I_4	I_3	
0	0	0	$R + S + C_0$
0	0	1	$S - R - 1 + C_0$
0	1	0	$R - S - 1 + C_0$
0	1	1	$R \vee S$
1	0	0	$R \wedge S$
1	0	1	$\bar{R} \wedge S$
1	1	0	$R \oplus S$
1	1	1	$\bar{R} \oplus \bar{S}$

Примечание. Источники операндов R и S определяются по табл. 6.3, C_0 — перенос в младший разряд секции.

Таблица 6.3. Коды источников операндов АЛУ

КИС			Источники	
I_2	I_1	I_0	R	S
0	0	0	A	Q
0	0	1	A	B
0	1	0	0	Q

КИС			Источники	
I ₂	I ₁	I ₀	R	S
0	1	1	0	B
1	0	0	0	A
1	0	1	D	A
1	1	0	D	Q
1	1	1	D	0

Примечание. А, В—поля микрокоманды, которые определяют РОНЫ, участвующие в выполнении данной микрокоманды; Q—внутренний рабочий регистр ЦПЭ; D—поле данных (констант) микрокоманды.

Таблица 6.4. Коды управления переходами

Код				Вид перехода
P ₃	P ₂	P ₁	P ₀	
0	0	0	0	Переход на адрес из поля BR, если F ≠ 0
0	0	0	1	Безусловный переход на адрес из поля BR
0	0	1	0	Переход к следующему адресу
0	0	1	1	Переход на адрес из регистра команд
0	1	0	0	Переход к подпрограмме по адресу из поля BR, если F ≠ 0
0	1	0	1	Безусловный переход к подпрограмме
0	1	1	0	Возврат из подпрограммы
0	1	1	1	Переход по стеку
1	0	0	0	Окончить цикл и вытолкнуть из стека, если F = 0
1	0	0	1	Загрузить стек и продолжить
1	0	1	0	Вытолкнуть из стека и продолжить
1	0	1	1	Окончить цикл и вытолкнуть из стека, если C ₄ = 1
1	1	0	0	Переход на адрес из поля BR, если F = 0
1	1	0	1	Переход на адрес из поля BR, если F ₃ = 1
1	1	1	0	Переход на адрес из поля BR, если OVR = 1
1	1	1	1	Переход на адрес из поля BR, если C ₄ = 1

Примечание. F—результат выполнения предыдущей операции, F₃—значение знака (старшего разряда) результата; OVR—признак переполнения разрядной сетки, C₄—значение переноса из старшего разряда результата.

Таблица 6.5. Коды приемников результата

КПР			Приемник и его содержимое	Выходная шина данных ЦПЭ Y
I ₈	I ₇	I ₆		
0	0	0	Q: = F	F
0	0	1	Нет загрузки	F
0	1	0	B: = F	A

КПР			Приемник и его содержимое	Выходная шина данных ЦПЭ Y
I ₈	I ₇	I ₆		
0	1	1	B: = F	F
1	0	0	B: F/2; Q: = F/2	F
1	0	1	B: = F/2	F
1	1	0	B: = 2F; Q: = 2Q	F
1	1	1	B: = 2F	F

Примечание. F — результат выполнения операции.

ЗАДАЧИ И УПРАЖНЕНИЯ

6.92. Объяснить по схеме на рис. 6.3 принцип действия МПС с микропрограммным управлением. Каково назначение блоков МПС: центрального процессорного элемента (ЦПЭ), блока микропрограммного управления (БМУ), блока памяти микропрограмм (БПМ), регистра микрокоманд (РгМК), регистра признаков (РгП), мультиплексора MS?

6.93. а) Что такое модульная архитектура МПС? Каковы основные преимущества и недостатки секционированных МП по сравнению с однокристалльными?

б) Какие блоки осуществляют функцию управления МПС (рис. 6.3)?

в) В чем заключается принцип микропрограммирования?

г) Каковы основные отличия структурных схем МПС, построенных на секционированных и однокристалльных МП?

д) Почему для хранения микропрограмм целесообразно использовать ПЗУ, а не ОЗУ?

е) Какие элементы схемы на рис. 6.3 выполняют функцию сверхоперативной памяти (СОЗУ)? Каковы ее емкость и назначение?

ж) Как формируется адрес следующей микрокоманды? Какую роль играет при этом регистр признаков РгП?

6.94. а) Как осуществляется адресация приемника результата?

б) Перечислить признаки результата операции (флаги), которые фиксируются в ЦПЭ K1804.

в) Как обеспечиваются последовательный, безусловный и условный переходы в МПС, схема которой приведена на рис. 6.3? Рассмотреть случаи, когда в поле управления переходами Р микрокоманды указаны коды 0010; 0001; 0000; 0101 и др. Как при условных переходах используются флаги?

г) На основе МПК K1804 построить МПС для обработки 16-разрядных двоичных чисел. Нарисовать схему операционного устройства такой системы, используя в ней соответствующее число ЦПЭ типа K1804BC1 и схему ускоренного переноса СУП

типа K1804BP1. Оценить и сравнить задержку распространения сигнала в схемах без ускоренного переноса и с ускоренным переносом.

д) Как организуется и какие функции выполняет в МПС стековая память?

6.95. а) Как обеспечиваются переход к подмикропрограмме и возврат из нее?

б) Как осуществляется ввод внешней информации в МПС?

в) Как осуществляется вывод информации из МПС?

6.96. В чем принципиальное отличие управляющего устройства МПС, построенной на базе секционированных МП, от внутреннего управляющего устройства (первичного управляющего автомата) однокристалльного МП?

6.97. а) Каково назначение полей микрокоманды D, B, A, КОП, C₀, КИС, MS₁, КПП, MS₂, P, BR (см. табл. 6.1)?

б) Из каких элементов пользователь МПС с микропрограммным управлением составляет микрокоманды, микропрограммы, команды, программы?

в) Оценить число различных микрокоманд (без учета и с учетом модификаций), которыми может пользоваться программист при составлении микропрограмм для МПС (см. рис. 6.3). Сравнить его с числом команд однокристалльного МП КР580ВМ80.

г) Перечислить микрооперации, которые могут выполняться МПС (см. рис. 6.3).

д) Объяснить, почему безусловный переход по заданному адресу можно совместить в одной микрокоманде с выполнением какой-либо операции в АЛУ, а условный переход — нельзя.

е) Можно ли одной микрокомандой обеспечить выполнение нескольких микроопераций, таких, например, как сложение содержимого двух РОНов, удвоение полученного результата, проверка выполнения некоторого условия, переход по заданному адресу и т. д. Привести примеры совместимых и несовместимых в одной микрокоманде микроопераций.

ж) Как реализуется возможность приема на шину данных МПС информации от нескольких источников, одним из которых является константа, записанная в поле микрокоманды, а другими — внешние источники данных?

з) Как объединяются микропроцессорные секции, если требуемая разрядность операционного устройства больше четырех и составляет, например, 16? Какие меры принимаются для уменьшения времени задержки переноса в такой схеме?

6.98. Указать основные отличия программирования МПС, построенных на однокристалльных МП КР580ВМ80 и секционированных МП K1804.

6.99. Какие из нижеперечисленных операций могут быть реализованы в МПС на рис. 6.3 с помощью одной микрокоманды:

а) сложение содержимого двух РОНов и переход к подпрограмме при выполнении условия $F \neq 0$;

б) сложение содержимого РОНа с константой, загрузка удвоенной суммы в рабочий регистр ЦПЭ Q и безусловный переход по заданному адресу;

в) удвоение содержимого РОНа, загрузка результата в этот же РОН и в регистр результата RгР,

г) сдвиг на один разряд влево содержимого РОНа и регистра Q, безусловный переход к заданному адресу;

д) переход к заданному адресу, если результат операции, выполненной в предыдущей микрокоманде, отрицательный;

е) переход к заданному адресу, если результат операции положительный;

ж) поразрядное умножение содержимого двух РОНов, выдача произведения в выходную шину данных ЦПЭ Y, деление произведения пополам, загрузка результата в РОН, где находился один из сомножителей;

з) загрузка РОНа константой из поля D микрокоманды, считывание содержимого РОНа в регистр результата RгР;

и) загрузка двух РОНов одной и той же константой из поля микрокоманды;

к) переход к заданному адресу при выполнении определенного условия (например, $F \neq 0$) и переход к следующему адресу при невыполнении этого условия;

л) переход к заданному адресу при выполнении определенного условия и переход к другому заданному адресу при выполнении другого условия;

м) поразрядное сложение по модулю 2 содержимого двух РОНов, загрузка результата в один из этих РОНов и выдача содержимого второго РОНа в выходную шину данных ЦПЭ Y;

н) сложение содержимого трех РОНов с загрузкой суммы в один из них;

о) сложение содержимого двух РОНов с загрузкой суммы в третий РОН.

6.100. Расшифровать следующие микрокоманды МПС (см. рис. 6.3), выполненной на базе МПК K1804:

Вариант	BR	P	MS ₂ КПП	MS ₁ КИС	C ₀ КОП	A	B	D
а	0000	0010	0 011	0 111	0 000	0000	1000	0110
б	0000	0010	0 000	0 001	1 001	0101	1010	0000
в	0000	0010	0 111	0 111	0 000	0111	0111	0111
г	0000	0010	0 011	0 001	1 110	1100	0011	0000
д	0101	0001	0 011	1 111	0 011	0010	0101	1001
е	1010	0101	0 011	0 001	0 000	0110	1001	0000
ж	0000	0010	0 110	0 001	1 010	0000	1111	0110
з	0000	0110	0 000	0 100	0 011	1001	0100	0000

Указание. См. формат микрокоманды в табл. 6.1, а также в табл. 6.2—6.5 и рис. 6.4. При расшифровке микрокоманд следует иметь в виду, что некоторые символы или поля в данной микрокоманде являются избыточными, т. е. не несут информации о ее содержании.

6.101. Составить два-три варианта микрокоманд загрузки одного из РОНов МПС рис. 6.3 (например, $РОН_0$) константой (например, $D=5$) с выдачей содержимого этого РОНа в выходную шину данных ЦПЭ Y .

Указание. Воспользоваться различными кодами операции КОП АЛУ.

6.102. Составить два-три варианта микрокоманд для МПС (рис. 6.3), обеспечивающих «очистку» (т. е. установку в нулевое состояние) рабочего регистра ЦПЭ Q .

6.103. Составить микрокоманду для МПС (см. рис. 6.3), обеспечивающую выполнение операции суммирования $F=(РОН_4)+(РОН_5)$, загрузку результата в $РОН_5$ и безусловный переход к адресу 0011.

6.104. Составить микропрограмму для МПС (см. рис. 6.3), обеспечивающую выполнение перехода к заданному адресу (например, 1110), если в заданном РОНе (например, $РОН_6$) при выполнении предыдущей микрокоманды образовался нечетный результат.

6.105. Составить микропрограмму для МПС (см. рис. 6.3), обеспечивающую сравнение содержимого двух РОНов (например, $РОН_0$ и $РОН_{15}$) и переход к МК с заданным адресом (например, 0011), если содержимое РОНов одинаково.

6.106. Составить микропрограмму для МПС (см. рис. 6.3), обеспечивающую сравнение содержимого РОНа (например, $РОН_0$) с константой k и переход к МК с заданным адресом (например, 1101), если $(РОН_0)=k=0101$.

6.107. Составить микропрограмму для МПС (см. рис. 6.3), обеспечивающую сложение содержимого двух РОНов (например, $РОН_0$ и $РОН_1$), загрузку суммы в один из них (например, $РОН_1$) и в регистр Q .

6.108. Составить фрагмент микропрограммы для МПС (см. рис. 6.3), обеспечивающий циклическую работу МПС при $i \neq k$ и выход из цикла при $i=k$, где i —номер цикла, k —константа (схема алгоритма показана на рис. 6.5).

6.109. Составить микропрограмму для МПС (см. рис. 6.3), имитирующую непрерывную работу двоичного суммирующего счетчика с модулем счета $k=16$. Текущие состояния счетчика ...0, 1, ..., 15, 0, 1 ... должны выдаваться в выходную шину данных ЦПЭ.

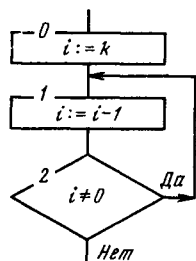


Рис. 6.5. Схема алгоритма формирования цикла

6.110. Составить микропрограмму для МПС (см. рис. 6.3), имитирующую непрерывную работу двоичного вычитающего счетчика с модулем счета $k=16$. Текущие состояния счетчика ... 15, 14, ..., 1, 0, 15 ... должны выводиться в выходную шину данных ЦПЭ Y.

6.111. Составить микропрограмму для МПС (см. рис. 6.3), имитирующую непрерывную работу двоичного суммирующего счетчика с модулем счета k , который равен: а) 256; б) 4096. Необходимо предусмотреть начальную установку счетчика на 0. Коды текущих состояний счетчика должны выводиться в регистр Q полубайтами (при $k=256$: 0000, 0000, 0000, 0001 и т. д.).

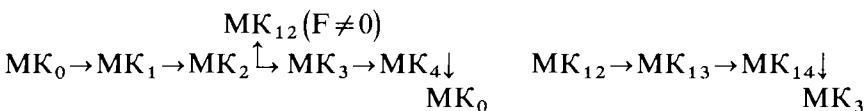
6.112. Составить циклические микропрограммы, обеспечивающие формирование и вывод в выходную шину данных ЦПЭ Y бесконечных последовательностей чисел:

- а) 0, 3, 6, 9, 12, 15, 2, 5, 8 и т. д.
- б) 0, 5, 10, 15, 4, 9, 14, 3, 8 и т. д.
- в) 1, 2, 4, 8, 1, 2, 4, 8 и т. д.
- г) 1, 2, 4, 8, 4, 2, 1, 2 и т. д.
- д) 0, 1, 4, 13, 8, 9, 12, 5, 0 и т. д.
- е) 0, 1, 2, 5, 12, 13, 6, 9, 8, 9, 10, 13 и т. д.
- ж) 0, 1, 2, 2, 3, 5, 7, 10, 15, 6, 0, 15 и т. д.
- з) 0, 8, 12, 14, 15, 7, 3, 1, 0, 8 и т. д.

Указание. Прежде чем приступить к составлению микропрограммы, необходимо выявить закономерность, связывающую числа в данной последовательности. Эту закономерность следует представить в виде рекуррентного соотношения, в котором должны использоваться элементарные операции МП КР1804 (арифметические, логические, сдвиговые).

6.113. В одной из микрокоманд микропрограммы МПС (см. рис. 6.3) в поле управления переходами Р вместо кода 0100 ошибочно указан код 0000. Допустимо ли это?

6.114. В области памяти ЗУ 0000 ... 0100 записана основная микропрограмма МПС (рис. 6.3) (микрокоманды МК₀ ... МК₄), в области 1100 ... 1110 — подмикропрограмма (микрокоманды МК₁₂ ... МК₁₄). В МК₂ имеется условный переход к подмикропрограмме при ненулевом результате, в МК₄ — безусловный переход к МК₀, в МК₁₄ — возврат из подмикропрограммы. Все эти переходы могут быть условно изображены в виде схемы:



(угловая стрелка обозначает разветвление и переход к подмикропрограмме по указанному адресу и условию; стрелка вправо обозначает последовательный переход; стрелка вниз или вверх — безусловный переход по указанному адресу).

Составить микропрограмму, обеспечивающую выполнение указанных переходов. (В микрокомандах указать только поля Р и BR; считать для простоты, что микрокоманда МК_{*i*} располагается в ячейке памяти с адресом *i*.)

6.115. Составить микропрограммы, в которых управление переходами осуществляется в соответствии со следующими схемами (см. задачу 6.114):

$$\text{МК}_{10}(F_3 = 1)$$

$$\text{а) } \text{МК}_0 \rightarrow \text{МК}_1 \xrightarrow{\uparrow} \text{МК}_2 \rightarrow \text{МК}_3 \downarrow \quad \text{МК}_{10} \rightarrow \text{МК}_{11} \downarrow$$

МК_1 МК_3

$$\text{МК}_4(C_4 = 1) \quad \text{МК}_4(C_4 = 1) \quad \text{МК}_{10}(F = 0)$$

$$\text{б) } \text{МК}_0 \xrightarrow{\uparrow} \text{МК}_1 \xrightarrow{\uparrow} \text{МК}_2 \rightarrow \text{МК}_3 \xrightarrow{\uparrow} \text{МК}_4 \downarrow \quad \text{МК}_{10} \rightarrow \text{МК}_{11} \downarrow$$

МК_1 МК_4

$$\text{МК}_{10}(F \neq 0) \quad \text{МК}_{12}(C_4 = 1)$$

$$\text{в) } \text{МК}_0 \rightarrow \text{МК}_1 \xrightarrow{\uparrow} \text{МК}_2 \rightarrow \text{МК}_3 \xrightarrow{\uparrow} \text{МК}_4 \downarrow$$

МК_0

$$\text{МК}_{10} \rightarrow \text{МК}_{11} \downarrow \quad \text{МК}_{12} \rightarrow \text{МК}_{13} \downarrow$$

МК_2 МК_4

$$\text{МК}_{10}(F \neq 0) \quad \text{МК}_{12}$$

$$\text{г) } \text{МК}_0 \rightarrow \text{МК}_1 \xrightarrow{\uparrow} \text{МК}_2 \rightarrow \text{МК}_3 \rightarrow \text{МК}_4 \uparrow$$

$$\text{МК}_{10} \rightarrow \text{МК}_{11} \rightarrow \text{МК}_{12} \downarrow$$

МК_2

$$\text{МК}_6(F \neq 0)$$

$$\text{МК}_{12}(F = 0)$$

$$\text{д) } \text{МК}_0 \rightarrow \text{МК}_1 \xrightarrow{\uparrow} \text{МК}_2 \rightarrow \text{МК}_3 \downarrow \quad \text{МК}_6 \rightarrow \text{МК}_7 \xrightarrow{\uparrow} \text{МК}_8 \downarrow$$

МК_0 МК_2

$$\text{МК}_{12} \rightarrow \text{МК}_{13} \downarrow$$

МК_8

$$\text{МК}_{12}(F = 0) \quad \text{МК}_{12}(F_3 = 1)$$

$$\text{е) } \text{МК}_0 \xrightarrow{\uparrow} \text{МК}_1 \rightarrow \text{МК}_2 \xrightarrow{\uparrow} \text{МК}_3 \downarrow$$

МК_1

$$\text{МК}_2(C_4 = 1)$$

$$\text{МК}_{12} \xrightarrow{\uparrow} \text{МК}_{13} \downarrow$$

МК_0

6.116. Составить микропрограмму для МПС КР1804 (см. рис. 6.3), обеспечивающую выполнение следующих операций:

в поле данных записано 4-разрядное положительное число N ; если оно нечетное, то в регистр Q выводится число $N-1$, если четное — выводится число $N/2$.

6.117. Определить длительность выполнения микропрограмм из задач 6.114 и 6.115 при тактовой частоте $f_T = 5$ МГц.

6.118. Составить микропрограммы для МПС (см. рис. 6.3), обеспечивающие вычисление следующих выражений:

- а) $F = 2\varphi_1 + \varphi_2$;
- б) $F = \varphi_1/2 + \varphi_2 + 1$;
- в) $F = 2(\varphi_1 - 2\varphi_2)$;
- г) $F = \frac{1}{2}(\varphi_1 - \varphi_2 + \varphi_3/4)$;
- д) $F = 4(\varphi_1 - 2\varphi_2 + \varphi_3/2)$;
- е) $F = 3\varphi_1 - 1 - \varphi_2/2$;
- ж) $F = 7(\varphi_1/2 + 3\varphi_2 - \varphi_3)$,

где $\varphi_1, \varphi_2, \varphi_3$ — целые неотрицательные числа, которые вводятся через поле D микрокоманды. Результат необходимо вывести в регистр Q .

Определить длительность выполнения микропрограмм при тактовой частоте $f_T = 5$ МГц.

6.119. Составить микропрограммы для МПС (см. рис. 6.3), обеспечивающие вычисление следующих выражений:

- а) $F = \begin{cases} \varphi_1 & \text{при } \varphi_1 < 2, \\ \varphi_1 + 2\varphi_2 & \text{при } \varphi_1 \geq 2; \end{cases}$
- б) $F = \begin{cases} \varphi_1 + \varphi_2 & \text{при } \varphi_1 < \varphi_2, \\ \varphi_1 - \varphi_2 & \text{при } \varphi_1 \geq \varphi_2; \end{cases}$
- в) $F = \begin{cases} \varphi_1 + 3\varphi_2 & \text{при } \varphi_1 - \varphi_2 > 2, \\ 3\varphi_1 + \varphi_2 & \text{при } \varphi_1 - \varphi_2 \leq 2; \end{cases}$
- г) $F = \begin{cases} 4\varphi_1 & \text{при } \varphi_1 < 4, \\ 2\varphi_1 & \text{при } \varphi_1 = 4, \\ \varphi_1/2 & \text{при } \varphi_1 > 4; \end{cases}$
- д) $F = \begin{cases} \varphi_1\varphi_2 & \text{при } \varphi_1, \varphi_2 \leq 3, \varphi_1 \leq \varphi_2, \\ \varphi_1 + \varphi_2 & \text{при } \varphi_1, \varphi_2 > 3, \varphi_1 \leq \varphi_2, \\ \varphi_1 - \varphi_2 & \text{при } \varphi_1 > \varphi_2; \end{cases}$
- е) $F = \begin{cases} \varphi_1^2 & \text{при } \varphi_1 > \varphi_2, \\ \varphi_2^2 & \text{при } \varphi_1 \leq \varphi_2; \end{cases}$
- ж) $F = \begin{cases} \lfloor \log_2 \varphi_1 \rfloor & \text{при } 15 \geq \varphi_1 \geq 1, \\ 0 & \text{при } \varphi_1 = 0, \end{cases}$

где ϕ_1, ϕ_2 — целые неотрицательные числа. Результат должен быть выведен в шину данных У.

6.120. Составить микропрограмму для МПС (см. рис. 6.3), имитирующую работу 4-разрядного кольцевого регистра — распределителя импульсов. В регистр Q должны последовательно выводиться следующие коды: 0001, 0010, 0100, 1000, 0001 и т. д.

6.121. Составить схему алгоритма и циклическую микропрограмму для МПС (см. рис. 6.3), имитирующую непрерывную работу 8-разрядного кольцевого регистра, в котором циркулирует одна единица. Вывод информации в регистр Q должен производиться полубайтами: 0000, 0001, 0000, 0010, ..., 1000, 0000, 0000, 0001 и т. д. (запятые указаны условно).

6.122. Задание то же, что в задаче 6.121, но кольцевой регистр должен быть: а) 12-разрядным; б) 16-разрядным; в) 48-разрядным; г) 10-разрядным.

6.6. МИКРОПРОЦЕССОР TMS 32010¹

Современный этап развития радиоэлектронной аппаратуры характеризуется широким применением методов цифровой обработки сигналов (ЦОС), реализуемых с помощью микропроцессорных средств. Появился целый класс так называемых сигнальных процессоров, характерным примером которых является семейство TMS 320 [24—27, 32]. Приборы этого семейства применяются в тех случаях, когда требуется высокое быстродействие и неизбежны большие вычислительные затраты. Процессор TMS 32010 является первым из семейства приборов серии TMS 320. Он способен выполнять 5 млн. опер/с. Такая высокая производительность является результатом разработки эффективной системы команд и высокоразвитой конвейерной архитектуры.

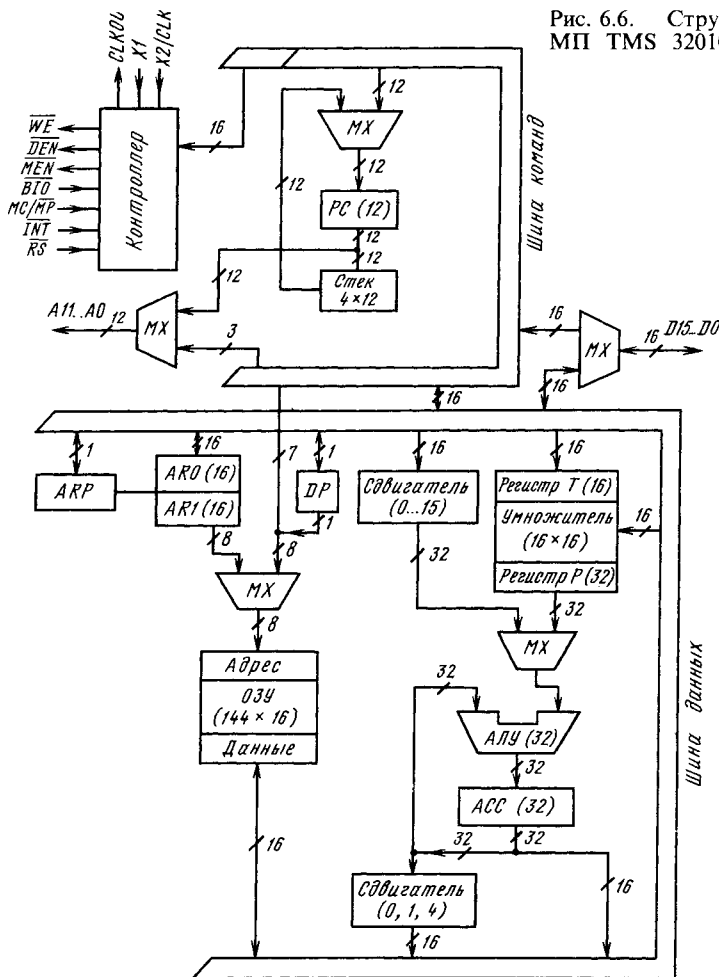
Основные характеристики МП TMS 32010

Длительность командного цикла	200 нс
Емкость оперативной памяти (ОЗУ) данных, содержащейся на одном кристалле с процессором	288 байт
Разрядность слов команд и данных	16 двоичных разрядов
Разрядность АЛУ, аккумулятора и регистра результата умножения	32 двоичных разряда
Длительность выполнения операции умножения двух 16-разрядных двоичных чисел	200 нс
Двухнаправленная 16-разрядная шина данных обеспечивает передачу данных со скоростью	40 Мбит/с
Напряжение питания	5 В

В процессоре реализована арифметика с фиксированной точкой в двоичном дополнительном коде. Процессор адресует восемь портов ввода и восемь портов вывода данных. Существует модификация процессора с ПЗУ емкостью

¹ При написании § 6.6 использован материал, представленный А. А. Игнатьевым.

Рис. 6.6. Структурная схема
МП TMS 32010



3 Кбайт на одном кристалле с процессором (TMS 320M10); имеется возможность расширения внешней программной памяти до общего объема 8 Кбайт при сохранении максимального быстродействия.

Для достижения высокого быстродействия и гибкости в основу процессора TMS 32010 положена модифицированная гарвардская архитектура (рис. 6.6), в которой программная память и память данных расположены в двух различных областях. Это позволяет полностью совместить во времени процессы выборки и выполнения команд. В отличие от традиционной гарвардской архитектуры версия, использованная при реализации TMS 32010, позволяет осуществлять передачи между областями программной памяти и памяти данных, увеличивая тем самым гибкость вычислительного устройства. Мак-

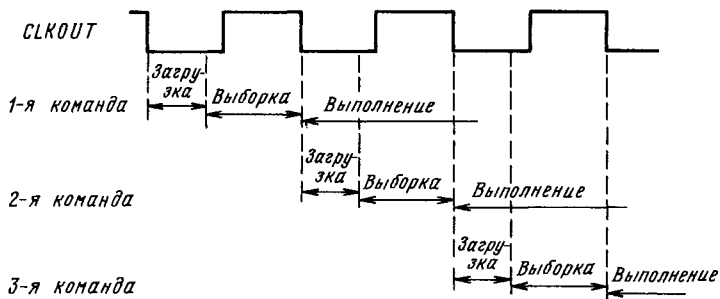


Рис. 6.7. Схема выполнения команд в МП TMS 32010

симальная емкость прямо адресуемой программной памяти составляет 4К 16-разрядных слов. При этом в качестве внешней необходимо использовать память со временем доступа менее 100 нс. Память данных представляет собой ОЗУ емкостью 144 16-разрядных слов, расположенное на кристалле. Из этого ОЗУ считываются операнды команд; из памяти, расположенной вне кристалла, нельзя непосредственно считывать операнды команд.

На рис. 6.7 приведены временные диаграммы, иллюстрирующие процедуру совмещения операций предварительной выборки и выполнения команды.

Процессор содержит четыре основных арифметических узла: АЛУ, аккумулятор (АСС), умножитель и устройство сдвига. АЛУ предназначено для выполнения операций сложения, вычитания и логических операций над 32-разрядными словами. Один из операндов берется из аккумулятора, второй поступает либо из регистра (Р) произведения умножителя, либо с внутренней шины данных с возможностью сдвига влево на 0...15 разрядов. Результат операции заносится в аккумулятор. Имеется возможность работы аккумулятора с насыщением, когда при переполнении АЛУ аккумулятор загружается самым большим по абсолютной величине положительным или отрицательным числом, в зависимости от знака переполнения.

Параллельный умножитель 16-разрядных чисел состоит из трех узлов: входного регистра Т, регистра произведения Р и матричного умножителя. Чтобы использовать умножитель, сначала необходимо загрузить множимое в регистр Т из памяти данных, а затем выполнить команду умножения, при этом в качестве множителя может быть число из ОЗУ данных или константа, извлекаемая из команды непосредственного умножения.

Процессор содержит два устройства сдвига: быстродействующее устройство последовательного сдвига для пересылки слов из ОЗУ данных в АЛУ, позволяющее получить сдвиг влево на 0...15 разрядов, и параллельное сдвигающее устройство для пересылки данных из аккумулятора в ОЗУ данных со сдвигом влево на 0, 1, 4 разряда.

Используются два способа адресации памяти данных: прямой и косвенный. Для формирования адреса ячейки памяти при прямой адресации семь младших разрядов слова команды объединяются с содержимым указателя страницы (DP) памяти данных, т. е. реализуется следующая схема деления памяти на страницы:

DP	Номера ячеек памяти
0	0...127
1	128...144

В режиме косвенной адресации в качестве адреса памяти данных используются восемь младших разрядов одного из вспомогательных регистров (AR0 или AR1), при этом нет необходимости разделения памяти данных на страницы. Текущий вспомогательный регистр выбирается с помощью указателя вспомогательного регистра ARP. При выполнении любой команды с косвенной адресацией вспомогательные регистры можно установить в режим автоинкремента/автодекремента (увеличить/уменьшить значение вспомогательного регистра на 1), а также изменить содержимое указателя вспомогательного регистра (ARP). Содержимое вспомогательного регистра и ARP изменяются после окончания выполнения текущей команды.

Вспомогательные регистры могут быть использованы для выполнения трех функций: временного хранения данных, косвенной адресации памяти данных и для управления циклами, для чего существует команда перехода по ненулевому состоянию вспомогательного регистра.

Программный счетчик (PC) и стек предоставляют возможность пользователю организовать переходы, вызовы подпрограмм, прерывания. Программный счетчик представляет собой 12-разрядный регистр, содержащий текущий адрес ячейки программной памяти, в которой записана команда, выполняемая следующей. После подачи сигнала «Сброс» на вход $\overline{RS}$ программный счетчик обнуляется. Стек представляет собой четыре 12-разрядных регистра, что определяет глубину вложений подпрограмм не более четырех. Система команд позволяет при необходимости расширить емкость стека за счет ОЗУ данных.

Система команд

Основные команды TMS 32010 приведены в приложении 4.

Формат команд с прямой адресацией:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Код операции 0 Прямой адрес памяти

Седьмой разряд ($P7=0$) определяет режим прямой адресации

Формат команд с косвенной адресацией:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Код операции 1 0 INC DEC ARP 0 0 ARP

Седьмой разряд ($P7=1$) определяет режим косвенной адресации. В разрядах 6—0 содержатся биты управления косвенной адресации.

Если третий разряд нулевой ($P3=0$), то содержимое нулевого разряда ($P0$) загружается в ARP. Если в третьем разряде содержится единица ($P3=1$), то содержимое ARP остается прежним. Если в пятом разряде содержится единица ($P5=1$), то содержимое текущего вспомогательного регистра увеличивается на единицу после выполнения текущей команды. Если $P4=1$, то содержимое текущего вспомогательного регистра уменьшается на единицу. Если в четвертом

и пятом разрядах нули, то операции инкремента/декремента не выполняются. Разряды 6, 2, 1 являются резервными и должны всегда содержать нули. Прямая и косвенная адресации могут быть использованы со всеми командами, использующими операнды данных, за исключением команд, обрабатывающих непосредственные операнды. Существуют четыре команды, использующие непосредственные операнды (LDPK, LARK, MРУК и LACK). Эти команды извлекают операнд непосредственно из кода команды.

Примеры команд¹

Прямая адресация

1. ADD 9, 5—сложить содержимое аккумулятора с содержимым ячейки памяти 9, которое предварительно сдвигается влево на 5 разрядов:

$(ACC) \leftarrow (ACC) + [9] \times 2^5$.

2. SUB 15, 3—из содержимого аккумулятора вычесть содержимое ячейки памяти 15, которое предварительно сдвигается влево на три разряда:

$(ACC) \leftarrow (ACC) - [15] \times 2^3$.

Косвенная адресация

3. ADD *, 8—сложить содержимое аккумулятора с содержимым ячейки памяти, адрес которой определяется содержимым текущего вспомогательного регистра. Перед сложением данные из ячейки памяти сдвигаются влево на восемь разрядов.

4. ADD *+, 8—те же действия, что в примере 3; содержимое текущего вспомогательного регистра после выполнения команды увеличивается на единицу.

5. ADD *- , 8—те же действия, что и в примере 3, но содержимое текущего вспомогательного регистра уменьшается на единицу.

6. ADD *+, 8, 1—те же действия, что и в примере 4, но в указатель вспомогательного регистра после выполнения команды дополнительно загружается единица, т. е. для следующей команды текущим вспомогательным регистром будет AR1.

7. ADD *+, 8, 0—те же действия, что и в примере 4, но в указатель вспомогательного регистра загружается нуль.

ЗАДАЧИ И УПРАЖНЕНИЯ

6.123. Какие способы повышения быстродействия использованы при построении МП TMS 32010?

6.124. Какова разрядность шины данных МП TMS 32010?

6.125. Какова разрядность внешней шины адреса МП TMS 32010?

6.126. Сколько разрядов требуется для адресации портов ввода-вывода?

6.127. Какова разрядность АЛУ и аккумулятора?

6.128. Какие функции могут выполнять вспомогательные регистры?

¹ Все адреса и операнды в примерах команд и задачах приведены в 16-ричной системе счисления.

6.129. В чем специфика гарвардской архитектуры и в чем заключается ее модификация в МП TMS 32010?

6.130. Какие существуют режимы адресации памяти данных?

6.131. Каково назначение указателя страниц памяти данных? При каком режиме адресации он используется?

6.132. Какие режимы работы программного счетчика допускаются в TMS 32010?

6.133. Что произойдет в МП, если на вход $\overline{RS}$ подать напряжение низкого уровня?

6.134. Что такое текущий вспомогательный регистр? Чем определяется, какой вспомогательный регистр — текущий?

6.135. Сколько команд требуется для умножения двух 16-разрядных чисел?

6.136. Чем определяется глубина вложений подпрограмм? Как можно увеличить глубину вложений подпрограмм и как это сказывается на быстродействии микропроцессорной системы?

6.137. Каким будет значение AR0, AR1 и ARP после выполнения следующего фрагмента программы:

- | | |
|--|--|
| а) LARP 0
LARK AR0,5
LARK AR1,9
ADD *+,0,1
ADD * - | б) LACK 3
SACL 8
LAR AR0,8
LARP 0
LAR AR1,8
ADD *+,5,1
ADD * - |
|--|--|

в)

Адрес програм- мной памяти	Команда
100	LARK AR0, 5
101	LARP 0
102	ZAC
103	ADD *
104	BANZ
105	103

6.138. Каким будет содержимое аккумулятора после выполнения фрагментов программы? Определить время их выполнения:

- | | | |
|---|---|--|
| а) LACK 3
SACL 8
ADD 8,1
ADD 8,2 | б) LACK 7
SACL 2
LT 2
MPYK 5
APAC | в) LACK 1
SACL 5
LT 5
MPYK 2
LTA 5
MPYK 3
APAC |
|---|---|--|

Г)

Адрес	Команда
200	ZAC
201	LARP, 0
202	LARK AP0, 5
203	SAR AR0, *
204	ADD *
205	BANZ
206	203

Д)

Адрес	Команда
200	ZAC
201	LARK AR1, 3
202	LARP 1
203	SAR AR1, *
204	LT *
205	MPY *
206	APAC
207	BANZ
208	203

6.139. Составить фрагмент программы вычисления величины $y = x^2$ (x — целое число в диапазоне $-128 \leq x < 128$), если x вводится через порт 3, а y выводится в порт 5.

6.140. Составить фрагмент программы вычисления величины $z = 9x + 7y$ (x и y — целые числа в диапазоне $-1024 \leq x < 1024$, $-1024 \leq y < 1024$), если x вводится через порт 3, y вводится через порт 4, а z выводится в порт 1.

6.141. Составить фрагмент программы для записи в ячейки памяти данных с адресами 0...9 чисел 0...9 соответственно.

6.142. Составить фрагмент программы, которая по сигналу «Сброс» обнуляет аккумулятор, оба вспомогательных регистра, устанавливает в нуль указатель вспомогательного регистра и передает управление команде, расположенной в ячейке программной памяти по адресу 100H.

6.143. Составить фрагмент программы, которая по сигналу «Сброс» обнуляет все ячейки памяти данных и передает управление команде, расположенной в ячейке программной памяти по адресу 200H.

6.144. Составить фрагмент программы для вычисления величины $y = \sum_{i=0}^9 a_i$, если величины a_i ($|a_i| < 1024$) хранятся в ОЗУ данных в ячейках с адресами 0...9. Величину y вывести в порт 2.

6.145. Составить фрагмент программы для вычисления $y = \sum_{i=0}^9 i \cdot a_i$ ($|a_i| < 512$), если a_i хранятся в ОЗУ данных в ячейках с адресами 10...19H. Величину y вывести в порт 5.

6.146. Составить программу для реализации нерекурсивного цифрового фильтра, описываемого разностным уравнением

$y(n) = \sum_{i=0}^8 a_i \cdot x(n-i)$, где $|a_i| < 1$ и $|x(n-i)| < 1$, коэффициенты a_i хранятся в ОЗУ данных в ячейках с адресами 0...8; $x(n)$ вводится через порт 0, величину $y(n)$ вывести в порт 2.

6.147. Составить программу и определить структуру для реализации ³⁹нерекурсивного цифрового фильтра с разностным уравнением $y(n) = \sum_{i=0}^{39} a_i x(n-i)$ для обработки сигнала звукового диапазона с частотой дискретизации 48 кГц [9]. Входные и выходные отсчеты представлены 16-разрядным двоичным числом с фиксированной точкой. Работа фильтра синхронизируется внешним сигналом частотой 48 кГц (одновременно с приходом нового входного отсчета на входе синхронизации появляется единичный импульс длительностью не менее 1 мкс).

ОТВЕТЫ И РЕШЕНИЯ

Р6.1. Микропроцессор — программно-управляемое устройство, осуществляющее обработку информации и управление ею и построенное на одной или нескольких БИС.

Микропроцессорная система — специализированная система, построенная на базе микропроцессорных средств, полупроводниковой памяти, средств сопряжения с обслуживаемыми объектами и других вспомогательных устройств.

Микропроцессорный комплект — набор совместимых БИС для построения микропроцессорных систем и микроЭВМ.

МикроЭВМ — универсальная ЭВМ, состоящая из микропроцессоров, полупроводниковой памяти, средств связи с периферийными устройствами, пульта управления, источников питания и других устройств, объединенных общей конструкцией.

Р6.2. Микрооперация — элементарная функциональная операция, выполняемая за один такт под действием одного управляющего сигнала. Микрокоманда — совокупность микроопераций, выполняемых за один такт под действием нескольких управляющих сигналов. Микропрограмма — последовательность микрокоманд, обеспечивающая выполнение данной операции.

Р6.3. Однокристалльная микроЭВМ помимо арифметическо-логического устройства, устройства управления и сверхоперативной памяти (регистров общего назначения), которые входят и в однокристалльный микропроцессор, содержит также оперативную и постоянную память, устройство ввода-вывода и некоторые другие устройства. Однако адресуемая емкость памяти у однокристалльной микроЭВМ существенно меньше, чем у однокристалльного микропроцессора. Поэтому однокристалльные микроЭВМ применяются лишь в узкоспециализированных системах с несложными алгоритмами обработки информации.

Р6.4. МП с фиксированной системой команд проще программировать, чем микропрограммируемый. Однако при решении той или иной конкретной задачи фиксированная система команд может оказаться менее эффективной, чем набор специально сконструированных микрокоманд, и ее использование потребует

большого числа команд и большого времени выполнения. При использовании микропрограммируемого МП разработчик может для каждой конкретной задачи создать адекватный набор команд, но поскольку программирование здесь ведется на языке микрокоманд, ему придется для каждой команды составлять микропрограмму, что существенно усложнит процесс программирования. Следует, однако, отметить, что расширенная система команд, возможность их разнообразной модификации, использование многочисленных режимов адресации, работа с очередями и другие усовершенствования современных 16-разрядных однокристальных микропроцессоров практически полностью ликвидируют указанный недостаток и делают их более перспективными для использования как в микроЭВМ, так и в специализированных МПС.

Р6.5. В вычислительных системах, особенно рассчитанных на сложные алгоритмы обработки информации с высокой точностью, требуется большая емкость памяти и большая разрядность операндов.

В управляющих системах, которые часто работают в реальном масштабе времени, и в коммутационных системах к МП предъявляются высокие требования по быстродействию и возможности подключения большого числа внешних устройств.

Наряду с указанными характеристиками весьма важны, особенно во встроенных МПС, малые габариты, малая потребляемая мощность, высокая надежность и низкая стоимость.

Р6.6. Аппаратное (схемное) обеспечение — это совокупность электронных и механических средств. Программное (математическое) обеспечение — это набор программ, используемых в данной системе и хранящихся обычно на магнитных носителях (диски, ленты) или в ППЗУ. Программно-аппаратное обеспечение — это набор программ, реализованных («зашитых») в ПЗУ.

Р6.7. Центральным процессорным элементом (ЦПЭ) МПС или микроЭВМ называется устройство, реализующее арифметические и логические операции, функции управления. В качестве ЦПЭ может использоваться однокристальный МП.

Р6.8. Шина — группа линий, по которым передается однотипная информация. В МП обычно имеются: шина данных (ШД), шина адресов (ША) и шина управления (ШУ). В некоторых МП (например, в K1810BM86) используется совмещенная шина, по которой передаются и данные, и адреса. Достоинством такой организации является уменьшение числа внешних выводов БИС, недостатком — невозможность одновременной передачи той и другой информации. Благодаря совмещенной шине 16-разрядный МП K1810BM86 имеет столько же внешних выводов (40), сколько 8-разрядный KP580BM80.

Шина данных в МП обычно двунаправленная.

Р6.9. Порт — устройство, с помощью которого МП связывается с внешними устройствами системы. Как правило, порт

содержит внутреннюю память и может настраиваться внешними сигналами на различные режимы работы.

Р6.10. Интерфейс — полная система сопряжения всех устройств МПС или микроЭВМ — совокупность аппаратных средств (интерфейсных БИС), шин (линий связи) и так называемого протокола (алгоритмов обмена, управляющих сигналов и т. д.). В узком смысле слова интерфейс — это устройства ввода-вывода, которые управляют потоком и форматами данных между МП и внешними (периферийными) устройствами.

Р6.11. Основные режимы работы МП — работа по программе, записанной в памяти, обслуживание прерываний по командам от внешних устройств и режим прямого доступа к памяти, когда внешние устройства обмениваются информацией с памятью, минуя МП.

Р6.12. При естественной адресации команд адрес очередной команды определяется прибавлением к адресу исполняемой команды константы (например, единицы). При принудительной адресации адрес очередной команды содержится в одном из полей предыдущей команды. При первом способе команда получается короче, однако второй способ проще реализовать схемно, если программы содержат много переходов. В однокристалльных МП с фиксированной системой команд основной вид адресации — естественный, но в некоторых командах, например в командах перехода или обращения к подпрограммам, обойтись без принудительной адресации невозможно. Поэтому команды с естественной и принудительной адресацией в однокристалльных МП имеют разный формат (в КР580ВМ80 — 1, 2, 3 байта, в К1810ВМ86 — 2...6 байтов). В секционированных МП с микропрограммным управлением используется принудительная адресация, т. е. в каждой микрокоманде указывается адрес следующей микрокоманды или по крайней мере способ его определения.

Р6.13. См. [5, с. 76; 6, с. 19].

Р6.14. Состояние захвата характеризуется тем, что в МП буферы ШД и ША переходят в состояние высокого выходного сопротивления, так что МП отключается от других устройств МПС и прекращает работу. Это состояние наступает после подачи на МП сигнала запроса захвата $3Zx=1$ и используется в режиме прямого доступа к памяти (ПДП).

Состояние прерывания характеризуется тем, что по запросу от внешнего устройства $3Pr=1$ после выполнения текущей команды и подачи от внешнего устройства команды RST МП прекращает выполнение основной программы и переходит к выполнению программы обслуживания прерываний. (Адрес первой ячейки этой программы содержится в команде RST.) Возврат к основной программе происходит автоматически после выполнения программы обслуживания прерываний.

В состоянии ожидания МП переходит по команде останова HLT; в этом состоянии буферы ШД и ША отключены от других устройств системы.

Р6.15. а) Командный цикл—это цикл выполнения одной команды МП. Каждый командный цикл состоит из 1...5 машинных циклов. Имеется 10 типов машинных циклов, и в различных командных циклах они присутствуют в том или ином наборе. Каждый машинный цикл состоит из 3...5 тактов; каждый такт длится фиксированное время (в МП КР580ВМ80 при тактовой частоте 2 МГц длительность такта 0,5 мкс); в течение такта выполняется одна микрокоманда. Длительность самого короткого командного цикла, состоящего из одного четырехтактного машинного цикла (команда суммирования ADD r и ряд других), 2 мкс, самого длинного, состоящего из 5 машинных циклов, 18 тактов (команда обмена между вершиной стека и регистровой парой HL XTHL)—9 мкс.

б) Командный цикл ввода содержит три машинных цикла: M_1 —выборка 1-го байта команды из ячейки памяти—4 такта; M_2 —чтение из следующей по порядку ячейки памяти 2-го байта команды—адреса устройства ввода (порта)—3 такта; M_3 —чтение из устройства ввода байта данных и загрузка его в аккумулятор—3 такта. Общее число тактов 10, длительность выполнения всего командного цикла (при тактовой частоте 2 МГц) 5 мкс.

Р6.16. МП в первом такте каждого машинного цикла генерирует на ШД слово состояния, которое содержит информацию о типе цикла. Для фиксации слова состояния на время всего цикла в МПС используется либо специальный регистр, либо системный контроллер, в состав которого входит такой регистр. Поскольку всего существует 10 типов машинных циклов, возможны 10 различных слов состояния: выборка первого байта команды, чтение из памяти, запись в память, ввод из устройства ввода-вывода, вывод в устройство ввода-вывода и т. д. В системном контроллере из разрядов слова состояния и выходных сигналов МП формируются сигналы, которые используются для управления устройствами МПС (см. § 8.1).

Р6.17

Вариант	Тип машинного цикла	$\overline{CлП}$	$\overline{ЗлП}$	$\overline{Вв}$	$\overline{Выв}$	$\overline{ППр}$
а)	Чтение из памяти	0	1	1	1	1
б)	Запись в память	1	0	1	1	1
в)	Ввод из устройства ввода-вывода	1	1	0	1	1
г)	Вывод в устройство ввода-вывода	1	1	1	0	1
д)	Подтверждение прерывания	1	1	1	1	0

Р6.18. Длительность выполнения команды в МП КР580 определяется числом и типами машинных циклов, которые требуются для ее выполнения, а не числом байтов, которые она содержит. Поэтому команда с большим форматом не обязательно выполняется за большее время. Например, трехбайтовая команда LXI (загрузка пары регистров) содержит три машинных цикла и 10 тактов, т. е. при тактовой частоте 2 МГц выполняется за 5 мкс, тогда как однобайтовая команда XTHL (обмен между вершиной стека и парой регистров HL) содержит 5 машинных циклов и 18 тактов, т. е. выполняется за 9 мкс.

Р6.19. Модификация команды—это ее изменение без изменения кода операции. Например, команды MOV A, B; MOV A, C; MOV B, M и т. д. имеют одинаковый код операции 01 (два старших бита), а остальные 6 битов—это модифицируемые (т. е. различные) коды приемника и источника информации.

Р6.20. Для того чтобы МП мог выполнить команду, ему необходимо сообщить код операции, операнды или их адреса (РОНЫ МП, ячейки памяти), а также адрес, куда следует поместить результат. Адресацией называется способ указания адресов операндов и результата. При прямой адресации физический (исполнительный) адрес операнда указывается прямо в коде команды. (Пример: команда INR B—инкремент регистра B; в данной команде и операнд, и результат операции находятся в РОНе B микропроцессора.) При непосредственной адресации в команде указывается сам операнд. (Пример: команда MVI B, 00—загрузка в регистр B константы 00; здесь операнд указан непосредственно, а адрес результата—регистр B—прямо.)

При неявной адресации адрес операнда или результата подразумевается самим кодом операции; таким способом чаще всего адресуются аккумулятор. (Пример: команда CMA—инверсия разрядов аккумулятора.)

При косвенной адресации исполнительный адрес определяется микропроцессором по косвенным признакам. (Пример: команда INR M—инкремент ячейки памяти, адрес которой находится в регистровой паре HL.) Существуют многочисленные разновидности косвенной адресации (относительная, индексная и др.). В МП КР580 они не применяются, но применяются, например, в МП K1810BM86.

Использование различных видов адресации позволяет уменьшить формат команд, сэкономить память и в конечном счете повысить производительность МП.

Р6.21. а) Прямой адрес РОНа в однобайтовых командах. Пример: команда ADD B, в коде которой 10000000 последние три разряда 000—код (адрес) РОНа B:

прямые адреса двух РОНов в однобайтовых командах. Пример: команда MOV C, E, в коде которой 01001011 разряды 001 и 011—коды РОНов C и E;

прямой адрес регистровой пары в однобайтовых командах (команда DAD D, в коде которой 00001001 5-й и 4-й разряды (считая справа) 01—код регистровой пары DE);

прямой адрес порта ввода или вывода (второй байт в двухбайтовой команде IN 05, в которой 05—это код одного из портов ввода);

прямой адрес ячейки памяти, содержащийся во 2-м и 3-м байтах трехбайтовых команд (LDA 0000).

б) Диапазон адресов регистров МП КР580 $0 \dots 7 = 111_{(2)}$ (см. табл. ПЗ.1); диапазон адресов портов $0 \dots 255 = 11111111_{(2)} = FF_{(16)}$; диапазон адресов ячеек памяти $0 \dots 65535 = 1111111111111111_{(2)} = FFFF_{(16)}$.

Р6.22. Восьмибитовый операнд во вторых байтах двухбайтовых команд (пример ADI F1H);

16-битовый операнд во вторых и третьих байтах команд. Пример: LXI B, D011H; здесь используется прямая адресация регистровой пары BC и непосредственная адресация операнда D011.

Р6.23. Регистровая косвенная адресация—это двухступенчатая прямая адресация: первый байт команды содержит адрес регистровой пары, в которой находится адрес операнда. Пример: команда LDAX B. В коде этой команды 00001010 5-й и 6-й (считая справа) разряды 00—это код регистровой пары BC, в которой находится адрес ячейки ЗУ, содержащей операнд, загружаемый в аккумулятор. Другой пример—команда ADD M, в коде которой 10000110 разряды 110—это код, указывающий местонахождение адреса операнда—регистровую пару HL.

Р6.24. а), д), е), ж) прямая адресация; б) непосредственная адресация; в) косвенная адресация; з) прямая адресация регистра A и непосредственная адресация операнда FF.

Р6.25. а) Результат выполнения операции положительный (первый разряд справа), отличный от нуля (второй разряд), перенос между полубайтами отсутствует (4-й разряд), число единиц в байте результата нечетное (6-й разряд), перенос из старшего разряда отсутствует (8-й разряд); 3, 5 и 7-й разряды полезной информации не несут;

б) результат выполнения операции отрицательный, отличный от нуля, перенос между полубайтами имеется, число единиц в байте результата четное, перенос из старшего разряда имеется.

Р6.26. а) (A)=01110011; (B)=00001110; (РП)=00010010;

б) (A)=01010111; (B)=00001110; (РП)=00010010;

в) (A)=11001010; (B)=00001110; (РП)=10000110;

г) (A)=01100101; (B)=00001111; (РП)=00000110.

Р6.27. Первый способ обмена—это программный обмен, который осуществляется под управлением программы, для чего в соответствующие места программы вводятся специальные команды ввода и вывода. Типовая команда ввода или вывода содержит код операции, адрес (код) ВУ (например, порта) и сопровождается управляющими сигналами. В некоторых случаях по команде ввода или вывода сначала осуществляется проверка готовности ВУ на обмен информации и лишь затем осуществляется обмен.

Второй способ обмена—обмен с прерыванием программы, инициируемый ВУ, которое посылает в МП сигнал «Запрос прерывания». После выполнения текущей команды МП прекращает выполнение основной программы, выдает в ВУ сигнал «Подтверждение прерывания» и после получения от ВУ сигнала «Рестарт», содержащего информацию о том, какая подпрограмма обработки прерывания должна выполняться, приступает к ее выполнению. После окончания подпрограммы происходит автоматический возврат к основной программе.

Третий способ обмена—это непосредственный обмен ВУ с ОЗУ без использования аккумулятора и регистров МП (режим прямого доступа к памяти или режим захвата). При поступлении сигнала «Запрос захвата» МП заканчивает выполнение очередной команды и выдает сигнал «Разрешение захвата», по которому шины МПС переводятся в состояние высокого выходного сопротивления, в результате чего возникает возможность непосредственного подключения ВУ к ОЗУ.

Р6.28.¹

а) MVI A	00111110	б) LDA	00111010
A0	10100000	OF	00001111
STA	00110010	02	00000010
2C	00101100	MOV D,A	01010111
B0	10110000		
в) MOV A,E	г) LXI H	д) MOV C,B	
STA	02	LDAX D	
56	01	MOV B,A	
14	MOV B,M	MOV A,C	
	LDA	STAX D	
	F2		
	F1		
	MOV M,A		
	MOV A,B		
	STA		
	F2		
	F1		

¹ См. примечание к задаче 6.28.

е) MOV H,B	з) XRA A
MOV L,C	MOV B,A
SHLD	MOV C,A
00	MOV D,A
FF	MOV E,A

Р6.29. а) По команде MVI A,2D происходит загрузка константы 2D в регистр A. По команде ADI 0F происходит сложение 2D и 0F. В двоичном коде

$$\begin{array}{r} 00101101 \\ + 00001111 \\ \hline 00111100 \end{array}$$

В результате сложения $(A) = 3C_{(16)}$.

б) По команде MVI B,54 происходит загрузка константы 54 в регистр B, а по команде MOV A,B—ее пересылка в регистр A. По команде SUI 22 происходит вычитание из константы 54 константы 22.

$$\begin{array}{r} \text{В двоичном коде } 01010100 \\ - 00100010 \\ \hline 00110010 \end{array}$$

В результате вычитания $(A) = 32_{(16)}$.

в) По команде MVI A,14 происходит загрузка константы 14 в регистр A. По команде SUI 2D происходит вычитание из 14 константы 2D.

$$\begin{array}{r} \text{В двоичном коде } 00010100 \\ - 00101101 \\ \hline -00011001 \end{array}$$

Так как число -00011001 в дополнительном коде 11100111, в результате вычитания получим $(A) = E7_{(16)}$.

г) $(A) = AF$, $(CY) = 1$.

д) $(L) = 00$.

е) $(H) = 89$, $(L) = FF$.

Р6.30. а) $(A) = BF$; б) $(H) = CE$; $(L) = 1B$;

в) $(D) = 0B$; $(E) = 21$; г) $(A) = FC$; $(H) = 9F$; $(L) = 2B$;

д) $(B) = D1$, $(C) = 49$; е) $(H) = CE$, $(L) = 7A$.

Р6.31. а) Решение приведено в табл. Р6.1.

Таблица Р6.1

Адрес ячейки	Содержимое (мнемо-коды и операнды)	Выполняемая операция	Результат операции	Число тактов
0005 06	MVI A 0E	$(A) \leftarrow < B_2 >$	$(A) = 00001110$	7
07	ADD L	$(A) \leftarrow (A) + (L)$	$(A) = 11111010$	4
08	JNZ	Переход к 000C, если $(A) \neq 0$	Переход к 000C	10

Адрес ячейки	Содержимое (мнемокоды и операнды)	Выполняемая операция	Результат операции	Число тактов
09	0C			
0A	00			
0B	MOV A,C	$(A) \leftarrow (C)$	Пропуск	5
0C	INR A	$(A) \leftarrow (A) + 1$	$(A) = 11111011$	5
0D	MOV C,A	$(C) \leftarrow (A)$	$(C) = (A) = 11111011$	5

Всего тактов $N=31$ (команда MOV A,C не выполняется), время выполнения $t=15,5$ мкс, содержимое аккумулятора после выполнения программы $(A)=FB$.

б) $(A)=6F$, $t=20$ мкс; в) $(A)=00$, $t=8,5$ мкс;

г) $(A)=41$, $t=13$ мкс; д) $(A)=E2$, $t=19$ мкс;

е) $(A)=10$, $t=28$ мкс.

Р6.32. а) Программа содержит цикл (ячейки 0004...0008), который выполняется 5 раз;

(Порт 05)=05; $t=0,5(7+7+(5+4+10) \cdot 5+5+10)=62$ мкс;

б) $(A)=0A$, $t=123,5$ мкс;

в) $(A)=FF$, $t=2,812$ мс;

г) (Порт 05)=77, $t=74$ мкс;

д) Программа содержит два цикла: внешний (ячейки 0107...0110) и внутренний (ячейки 0109...010C); первый выполняется $64_{(16)}=100$ раз, второй $FA_{(16)}=250$ раз;

(Порт 05)=01; $t=0,5(10+7+(7+(5+10) \cdot 250+5+10) \cdot 100+10)=0,5 \cdot 377227 \approx 0,1886$ с.

Р6.33. LDA
C1
FD
MOV B,A
LDA
C2
FD
ADD B
STA
C1
FD

Р6.34. 0000 LDA
01 01
02 1F
03 MOV B,A
04 LDA
05 02
06 1F
07 MOV C,A
08 SUB B
09 JP
0A 10
0B 00
0C MOV A,B
0D OUT
0E FF
0F HLT
10 MOV A,C
11 OUT
12 FF
13 HLT

P6.35. а) Все операнды переводим в 16-ричную систему счисления, для умножения и деления на 2, 4, 8 используем операции сдвига;

а) MVI A
25
MOV B,A
MVI A
1E
RAR
MOV C,A
MOV A,B
SUB C
RAR
ADI
0B
HLT

P6.36.

а) IN	д) 0000 IN	
05	01 08	0B MOV A,C
ADD A	02 MOV B,A	0C ADD B
INR A	03 IN	0D STA
STA	04 05	0E 33
33	05 MOV C,A	0F 22
22	06 SUI	10 HLT
HLT	07 08	11 MOV A,C
	08 JP	12 SUB B
	09 11	13 STA
	0A 00	14 33
		15 22
		16 HLT

P6.37. С помощью команды CALL осуществляется вызов подпрограммы вычисления величины $p = \log_2 x$; во время выполнения подпрограммы адрес возврата 0007 запоминается в стеке; подпрограмма заканчивается командой RET, после которой выполняется очередная команда программы INR A: (порт 06) = $1 + \log_2 (2a - 1)$.

P6.38. 0000 IN
01 05
02 MOV B,A
03 CALL
04 00
05 11
06 INR A
07 MOV C,A
08 MOV A,B
09 ADD A
0A CALL

P6.39. 0000 IN
01 11
02 MOV D,A
03 MOV E,A
04 MOV C,A
05 MVI B
06 00
07 CALL
08 00
09 20
0A ADD E

```

0B 00
0C 11
0D ADD C
0E OUT
0F 06
10 HLT

```

```

0B MOV C,A
0C INR B
0D MOV A,B
0E SUI
0F 05
10 JM
11 07
12 00
13 MOV A,C
14 STA
15 00
16 30
17 HLT

```

Вызов подпрограммы $p = \cos x$ осуществляется с помощью команды CALL дважды.

Вызов подпрограммы, с помощью которой вычисляются степени x , осуществляется циклически с помощью команды CALL.

Промежуточные результаты суммирования хранятся в регистре C. Счет циклов проводится в регистре B.

Р6.40. а) Используем для загрузки константы $100 = 64_{(16)}$ регистр B.

```

MVI    B, 64H;
MET: DCR    B ;
      JNZ    MET ;

```

б) Поскольку константу $1000 = 3E8_{(16)}$ нельзя разместить в одном регистре, используем регистровую пару BC. Один из вариантов программы следующий:

```

LXI    B, 03E8H ;Загрузка регистровой пары BC кон-
                ;стантой
MT: DCX    B    ;Декремент регистровой пары BC
      MOV    A,C ;Пересылка (A)←(C)
      ORA    B   ;Логическое сложение (A)←(A)∨(B)
      JNZ    MT  ;Условный переход по ненулевому
                ;результату

```

При выполнении операции DCX признаки результата не формируются, поэтому для проверки содержимого регистровой пары BC приходится предварительно выполнять логическое сложение содержимого B и C. Очевидно, что $(B) \vee (C) = 0$ только при $(B) = (C) = 0$.

Р6.41. Вычисление произведения двух 8-разрядных положительных двоичных чисел $P = M \cdot N$ с помощью МП КР580, в системе команд которого команды умножения нет, может быть организовано в виде циклической процедуры, каждый цикл которой содержит команды проверки очередного разряда

множителя, суммирования и сдвига. Для вычисления произведения Π к накапливающейся сумме частичных произведений Π_i ($i=8, \dots, 1$) добавляется множитель M , если очередной разряд множителя $N_i=1$, после чего осуществляется сдвиг; при $N_i=0$ для единообразия циклов к Π_i прибавляется 0 и также осуществляется сдвиг. Число циклов определяется числом разрядов множителя N , т. е. равно 8. Анализ N_i можно начинать со старших разрядов ($i=8$) и сдвигать накапливающуюся сумму влево, либо с младших разрядов ($i=1$) и сдвигать ее вправо. Схема алгоритма, соответствующая первому варианту, представлена на рис. Р6.1. Отметим, что поскольку произведение двух 8-разрядных двоичных чисел содержит 16 разрядов, для хранения накапливающейся суммы частичных произведений Π_i и результата Π требуется пара РОНов МП.

Используем РОНЫ МП следующим образом: число M храним в регистре C , число N — в аккумуляторе, накапливающуюся сумму частичных произведений и результат — в регистровой паре HL , счетчик циклов организуем в регистре D . В программе используется команда суммирования содержимого пар регистров $DAD B$, т. е. $(H)(L) \leftarrow (H)(L) + (B)(C)$, поэтому задействован также регистр B .

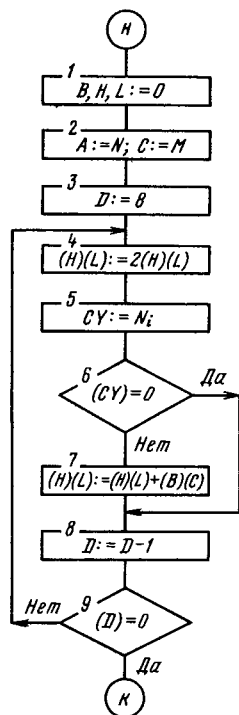


Рис. Р6.1. Схема алгоритма умножения

Программа умножения

MULT:	LXI	H,00	;Обнуление пары HL	10
	MOV	B,H	;Обнуление регистра B	5
	MVI	D,08	;Начальная установка счетчика циклов	7
M1:	DAD	H	;Удвоение содержимого пары HL ;(сдвиг Π_i влево на 1 разряд)	10
	RLC		;Сдвиг множителя N влево на ;1 разряд (ввод N_i в триггер переноса CY)	4
	JNC	M2	;Переход к M2, если $N_i=0$	10
	DAD	B	;Прибавление к Π_i множителя N	10
M2:	DCR	D	;(D) $\leftarrow$ (D) - 1 — организация счета циклов	5
	JNZ	M1	;Переход к M1 при (D) $\neq$ 0	10
Суммарное число тактов (при $N=N_{\text{макс}}=255$)				

$$n = 10 + 5 + 7 + 8(10 + 4 + 10 + 10 + 5 + 10) = 414.$$

Время выполнения программы при $f_T = 2$ МГц $t = 207$ мкс.
Р6.42. а)

Характеристика	КР580ВМ80	К1810ВМ86
Разрядность шины данных	8	16
Адресуемая емкость памяти	64 Кбайт	1 Мбайт
Тактовая частота	2 МГц	5 МГц
Арифметические операции	Сложение, вычитание	Сложение, вычитание, умножение, деление
Возможность работы с цепочками слов	Нет	Есть
Напряжение питания, В	+5, -5, +12	+5

б) Адресное пространство памяти 1 Мбайт разбито на 16 сегментов по 64 Кбайт в каждом. Четыре сегмента специализированы: сегмент программы, сегмент стека, сегмент данных и сегмент дополнительных данных. Начальный адрес каждого сегмента (сегментный адрес) содержит 20 бит, но 4 младших бита — всегда нули. Адрес ячейки в пределах сегмента называется смещением и содержит 16 бит. Исполнительный адрес получается при суммировании сегментного адреса и смещения (рис. Р6.2).

Сегментирование памяти позволяет хранить все адреса в 16-разрядных (а не в 20-разрядных) регистрах, но при этом приходится тратить дополнительное время на получение исполнительных адресов. Сегментные адреса хранятся в четырех сегментных регистрах CS, SS, DS, и ES, которые выбираются по программе.

Р6.43. Адресное пространство ввода-вывода — 65 536 портов. Исполнительный адрес порта имеет формат 20 бит, но старшие 4 бита всегда нулевые. Поэтому при адресации портов сегментные регистры не используются. Первые 256 портов могут иметь прямую адресацию; все порты могут адресоваться косвенно (адрес порта — в регистре, определяемом командой).

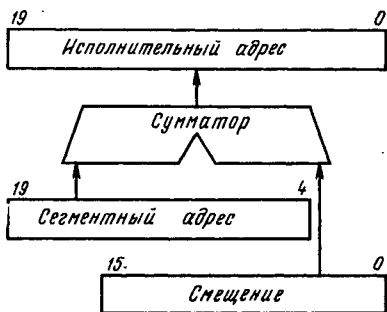


Рис. Р6.2. Схема вычисления исполнительного адреса в МП К1810

Р6.44. Регистр AX обычно выполняет функцию аккумулятора, BX используется как источник базового адреса (соответствует регистровой паре HL МП КР580ВМ80); CX часто используется в качестве счетчика циклов; DX используется в командах ум-

ножения и деления, а также при косвенной адресации портов ввода-вывода.

Все эти регистры—16-разрядные, но допускают независимую адресацию старшего и младшего байтов (AH-AL, BH-BL и т. д.).

Р6.45. 16-разрядные сегментные регистры содержат начальные адреса сегментов памяти: CS—сегмент программы, SS—сегмент стека, DS—сегмент данных, ES—сегмент дополнительных данных.

При формировании исполнительных адресов эти начальные адреса суммируются со смещением, указанным в команде, в соответствии с формулой

(исполнительный адрес) = $2^4 \cdot (\text{сегментный адрес}) + (\text{смещение})$.

Р6.46. 16-разрядные указательные и индексные регистры (SP, BP, SI и DI) предназначены для формирования эффективных адресов операндов и используются для хранения внутрисегментных смещений, но могут участвовать и в выполнении арифметических и логических операций.

Регистры SP и BP обеспечивают доступ к ячейкам памяти в сегменте стека, а SI и DI—в сегменте данных. В некоторых командах регистр SI относится к источнику данных, а регистр DI—к приемнику.

Р6.47. Регистр F—16-разрядный регистр признаков (флагов)—имеет следующий формат:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ф	Ф	Ф	Ф	OF	DF	IF	TF	SF	ZF	Ф	AF	Ф	PF	Ф	CF

CF—перенос (при CF=1—есть перенос); PF—паритет (при PF=1 число единиц в результате—четное); AF—межтетрадный перенос в байте (при AF=1 есть перенос из младшей тетрады); ZF—нулевой результат (при ZF=1—нуль); SF—знак (при SF=1—минус); TF—признак прослеживания (при TF=1 обеспечивается шаговый режим исполнения программы); IF—признак прерывания (при IF=1 прерывания разрешены); DF—признак направления (при DF=0 выполняется инкрементирование регистра указателя массива, при DF=1—декрементирование); OF—признак переполнения (при OF=1 есть переполнение). Биты 1, 3, 5, 12, 13, 14 и 15 не используются. Младшая половина регистра F МП K1810 аналогична регистру признаков МП KP580.

Р6.48. В МП K1810 используется совмещенная шина адресов и данных, и поэтому выставить одновременно адрес ячейки памяти и данные нельзя. Обычно адрес выставляется в первом такте машинного цикла, а данные—в последующих.

Р6.49. Регистр команд МП K1810, называемый очередью команд, состоит из шести однобайтовых регистров. Команды

извлекаются из него побайтно, но в процессе исполнения последовательных участков программы, как только два регистра освобождаются, параллельно с работой операционного устройства и независимо от него из памяти программы выбираются следующие 2 байта. Это существенно увеличивает производительность МП.

Р6.50. В минимальном режиме МП К1810 сам вырабатывает весь набор сигналов для управления микропроцессорной системой. В максимальном режиме, рассчитанном на более сложные системы, МП вырабатывает сигналы для управления специальным контроллером, который, в свою очередь, вырабатывает сигналы для управления системой. Выбор режима осуществляется подачей сигнала на вход МП МИН/МАКС.

Р6.51. Команда МП К1810 (см. рис. 6.2) имеет формат 2...6 байт: 1-й байт содержит код операции и поле w ; 2-й, так называемый постбайт,—поля mod , reg и r/m ; 3-й и 4-й байты, присутствующие не во всех командах,—это так называемое смещение, которое позволяет вычислить исполнительный адрес; 5-й и 6-й байты представляют в командах с непосредственной адресацией второй операнд. Поле w задает формат операнда ($w=0$ —один байт, $w=1$ —двухбайтовое слово), поле reg —код регистра, содержащего второй операнд. (В командах с непосредственной адресацией поле reg используется для расширения поля кода операции.) Поля mod и r/m определяют способ формирования исполнительного адреса в сегментах данных и стека, причем поле mod задает используемый режим адресации и позволяет определить первый операнд, а поле r/m кодирует способ вычисления адреса (смещения) первого операнда.

При значении $mod=11$ поля r/m и reg определяют адрес РОНа.

Р6.52. а) Эффективный адрес отсутствует, операнд находится в регистре ВХ.

б) $EA=(1234H)$: операнд находится в ячейке памяти с эффективным адресом 1234H, исполнительный адрес $(BX)+(DS) \cdot 16=32234H$.

в) $EA=(BX)+disp\ H$, $L=2C67H$; исполнительный адрес $EA+(DS) \cdot 16=33C67H$.

Р6.53. а) $\underbrace{1\ 0\ 0\ 0\ 1\ 0\ 0\ 1}_{d\ w}\quad \underbrace{1\ 1\ 0\ 0\ 0\ 0\ 1}_{mod\ reg\ r/m}$

КОП $d\ w\ mod\ reg\ r/m$

КОП=100010—это код команды пересылки; бит $w=1$ означает, что обрабатываются 16-битовые слова; $mod=11$ свидетельствует о том, что оба адреса относятся к регистрам, причем поскольку $d=0$, источником информации является регистр $reg=000$, т. е. АХ, а получателем—регистр $r/m=001$, т. е. СХ. Сокращенно $(CX) \leftarrow (AX)$.

Мнемокод команды MOV CX,AX;
 б) 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟

КОП d w mod reg r/m

КОП=100010—команда пересылки; бит w=0—обработка 8-битовых слов; постбайт (mod=00, reg=000, r/m=000) при d=0 означает, что источник информации—регистр AL, получатель—ячейка памяти с адресом (BX)+(SI).

Мнемокод команды MOV [(BX)+(SI)], AL;

в) MOV BX, [(SI)+disp H,L]—пересылка из ячейки памяти с адресом, являющимся суммой содержимого регистра SI и 16-разрядного смещения, указанного в 3-м и 4-м байтах команды (disp H,L=D180H).

P6.54. а) 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1

⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟

КОП d w mod reg r/m

КОП=000000—код операции сложения; w=1—обработка 2-байтовых слов; mod=11-регистрая адресация; reg=000—код регистра AX; r/m=001—код регистра CX. При d=0 регистр AX—источник, CX—приемник; поэтому результат операции—сумма содержимого регистров AX и CX—помещается в регистр CX.

Мнемокод ADD CX, AX; комментарий (CX)←(AX)+(CX);

б) ADD BL, CH; (BL)←(CH)+(BL);

в) ADD AX,DI; (AX)←(DI)+(AX).

P6.55. Двоичные коды 1-го и 2-го байтов:

б) 1 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1

⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟

КОП d w mod КОП r/m

Состоящий из двух частей код операции 100000 000—это код команды сложения непосредственного операнда (4-й или 3-й байты) с содержимым регистра или ячейки памяти. При mod=11 и r/m=111 2-й операнд содержится в регистре DI; в него же помещается результат суммирования.

Мнемокод команды ADD DI,300FH. В результате выполнения команды содержимое регистра DI будет (DI)=41FFH.

P6.56. а) Эффективный адрес перехода

EA=(IP)+disp H,L=11F2H;

машинный код команды

11101001 11110010 00000001

⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟ ⏟

Е 9 F 2 0 1

б) EA=(BX)=DE01H;

машинный код команды

11111111 00100111

(1-й байт—КОП, 2-й байт—постбайт, у которого $\text{mod}=00$, $r/m=111$);

в) $EA = (BX) + \text{disp } H, L = DFF3H$;

машинный код команды

11111111 10100111 11110010 00000001

$\underbrace{\quad\quad\quad}_F \quad \underbrace{\quad\quad\quad}_F \quad \underbrace{\quad\quad\quad}_A \quad \underbrace{\quad\quad\quad}_7 \quad \underbrace{\quad\quad\quad}_F \quad \underbrace{\quad\quad\quad}_2 \quad \underbrace{\quad\quad\quad}_0 \quad \underbrace{\quad\quad\quad}_1$

(1-й байт—КОП, 2-й байт—постбайт адресации, у которого $\text{mod}=10$ и $r/m=111$; 4-й и 3-й байты—смещение).

P6.57. При межсегментном прямом переходе в счетчик команд загружается смещение $(IP) \leftarrow \text{disp } H, L$, и эффективный адрес перехода $EA = \text{disp } H, L = 010EH$; исполнительный адрес $EA + (CS) \cdot 16 = B210EH$.

Машинный код команды

11101010 00001110 00000001 00000000 10110010

$\underbrace{\quad\quad\quad}_E \quad \underbrace{\quad\quad\quad}_A \quad \underbrace{\quad\quad\quad}_0 \quad \underbrace{\quad\quad\quad}_E \quad \underbrace{\quad\quad\quad}_0 \quad \underbrace{\quad\quad\quad}_1 \quad \underbrace{\quad\quad\quad}_0 \quad \underbrace{\quad\quad\quad}_0 \quad \underbrace{\quad\quad\quad}_B \quad \underbrace{\quad\quad\quad}_2$

(1-й байт—КОП, 3-й и 2-й байты—смещение, 5-й и 4-й байты—сегментный адрес).

P6.58. См. решение задач 6.53—6.55.

P6.59. См. решение задач 6.53—6.55.

P6.60. Фрагмент программы:

MOV AL, M1 ;Загрузка регистра AL из ячейки памяти 10 ;M1

XCHG AL, M2 ;Обмен содержимого ячейки памяти 17 ;M2 и регистра AL

MOV M1, AL ;Загрузка ячейки памяти M1 из регистра 10 ;AL

(справа указано число тактов).

Число тактов, необходимое для выполнения команды XCHG, равно $11 + n_{EA}$, где n_{EA} —число тактов, необходимое для вычисления эффективного адреса [7, с. 53, 64, 500]. В данном случае используется прямая адресация ячейки памяти, поэтому $n_{EA} = 6$. Суммарное время выполнения фрагмента программы при $f_T = 5 \text{ МГц}$

$$t = (10 + 17 + 10) \cdot \frac{1}{5 \cdot 10^6} = 7,4 \text{ мкс.}$$

P6.61. Фрагмент программы:

SUMN: MOV CX, N	;Загрузка регистра CX кон-	
	;стантой N	4
MOV AX, 0	;Обнуление регистра AX	4
MOV SI, AX	;Обнуление регистра SI	2

ROT: ADD AX,BEG [SI]	;Прибавление очередного 18
	;слова из памяти
ADD SI,2	;Прибавление к (SI) числа 4
	;2
LOOP ROT	;Декремент CX и повторе- 17/5
	;ние цикла, если (CX) $\neq 0$
MOV RES,AX	;Загрузка результата в 15
	;ячейку памяти RES

В программе использована команда зацикливания LOOP, заменяющая пару команд — декремент и условный переход по ненулевому результату. Обозначение BEG [SI] в команде суммирования ADD указывает на то, что эффективный адрес очередного слова определяется суммированием начального адреса BEG и содержимого индексного регистра SI.

Число тактов для команды ADD AX, BEG [SI] определяется как $n' = 9 + n_{EA}$, где n_{EA} — число тактов для вычисления эффективного адреса. В данном случае $EA = (SI) + \text{disp H,L}$, где disp H,L — смещение, непосредственно задаваемое в команде как адрес ячейки BEG [SI] (относительная регистровая адресация). При этом $n_{EA} = 9$ и $n' = 18$.

Число тактов для команды LOOP ROT при выполнении условия $(CX) \neq 0$ равно 17, при невыполнении — 5. Число тактов для команды MOV RES,AX равно $n'' = 9 + n_{EA}$, и, поскольку имеет место прямая адресация памяти, $n_{EA} = 6$ и $n'' = 15$.

Цикл выполняется N раз, поэтому суммарное число тактов

$$n = (4 + 4 + 2) + (18 + 4 + 17) \cdot N + (18 + 4 + 5 + 15) = 52 + 39N.$$

При $f_T = 5$ МГц и $N = 100$ общее время выполнения фрагмента программы

$$t = (52 + 39 \cdot 100) \cdot \frac{1}{5 \cdot 10^6} = 790,4 \text{ мкс.}$$

Р6.62. Фрагмент программы:

MOV BL,0A7H	;Загрузка в регистр BL констан-
	;ты k_1
MOV CL,95H	;Загрузка в регистр CL констан-
	;ты k_2
INPUT: IN 07	;Ввод в регистр AL набора
	;переменных D из порта 07
XOR BL,AL	;Сравнение байтов k_1 и D
JZ OUT1	;Переход к OUT1 при $k_1 = D$
XOR CL,AL	;Сравнение байтов k_2 и D
JZ OUT1	;Переход от OUT1 при $k_2 = D$
JMP OUT0	;Переход от OUT0
OUT1: MOV AL,01H	;Загрузка регистра AL единицей
OUT 05	;Вывод единичного результата

		; через порт 05
	JMP INPUT	; Возврат к INPUT
OUT0:	MOV AL, 00H	; Загрузка регистра AL нулем
	OUT 05	; Вывод нулевого результата че-
		; рез порт 05
	JMP INPUT	; Возврат к INPUT

Команды XOR — исключающее ИЛИ (сумма по mod 2) — позволяют сравнить содержимое регистра AL, т. е. входной набор переменных, с содержимым регистров BL и CL, т. е. с минтермами. При совпадении (BL) или (CL) с (AL) $(BL) \oplus (AL) = 0$ или $(CL) \oplus (AL) = 0$, и через порт 05 выводится единица, в противном случае — нуль.

P6.63. Фрагмент программы:

MOV AX, [BX]	; Загрузка операнда X в регистр AX
MUL AX, [CX]	; Умножение X на Y
ADD AX, [DX]	; Прибавление T
ADD AX, AX	; Удвоение результата
CMP AX, 015EH	; Сравнение с константой 350
JS PROC1	; Переход по адресу PROC1 при $z < 350$
OUT 02	; Вывод результата

В программе используется операция беззнакового умножения MUL. Поскольку операнды — 8-битовые, произведение может иметь 16 бит, и для результата используется регистр AX. Для перехода по метке PROC1 используется команда сравнения CMP и условный переход по отрицательному результату JS.

P6.64. Программно-доступными узлами МК1816 являются: 8-разрядное АЛУ, стираемое перепрограммируемое ППЗУ емкостью 1 Кбайт, регистровое ОЗУ емкостью 64 байта, 8-разрядный программируемый таймер-счетчик, 8-уровневый указатель стека, программно-управляемые схемы ввода-вывода (27 линий), 12-разрядный счетчик команд.

P6.65. Поскольку счетчик команд МК1816 содержит 12 разрядов, адресуемая емкость памяти $2^{12} = 4\text{К}$ ячеек.

P6.66. Внутреннее ОЗУ МК1816 имеет общую емкость 64 байта (рис. P6.3, а). Два банка 8-битовых РОНов RB0 и RB1 — по 8 регистров в каждом банке, т. е. всего 16 регистров, — доступны командам с прямой адресацией. Остальные 48 регистров доступны командам с косвенной адресацией, причем адреса размещаются в аккумуляторе или РОНах — R0 и R1 — каждого банка. Банк выбирается с помощью команды SEL RB0 или SEL RB1. При выполнении каждой из этих команд номер выбранного банка фиксируется, и в дальнейшем, вплоть до подачи другой команды, используются РОНЫ только этого банка. Ячейки ОЗУ с адресами 8...23 адресуются также указателем стека и могут быть использованы в качестве 8-уровневого стека.

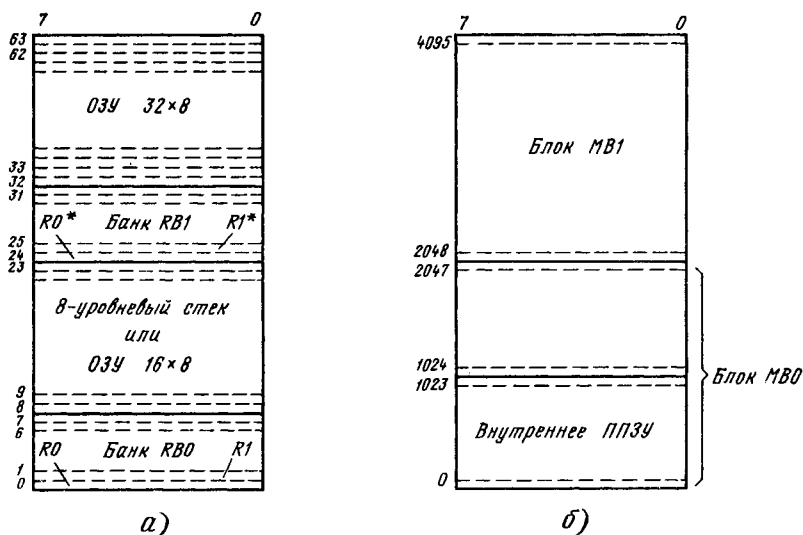


Рис. Р6.3. Память микроконтроллера КР1816:

а) внутреннее ОЗУ (память данных); б) внутреннее и внешнее ППЗУ (память программ)

Р6.67. Каждый из двух блоков ППЗУ МК1816 (рис. Р6.3, б), имеющий емкость 2 Кбайта, делится на 8 страниц по 256 байтов в каждом. Поэтому для указания адреса ячейки в пределах страницы достаточно 1 байта. В большинстве команд МК1816, в том числе в командах условных переходов, адресация осуществляется в пределах страницы, что накладывает определенные ограничения при составлении программ, но позволяет обходиться 2-байтовыми командами (а не 3-байтовыми, как в МП КР580ВМ80). Для перехода к другой странице данного блока ППЗУ используются команды вызова подпрограммы CALL или безусловного перехода JMP, которые также содержат по два байта, но в них три старших бита первого байта указывают код выбираемой страницы.

Р6.68. Таймер микроконтроллера МК1816 представляет собой 8-разрядный суммирующий счетчик, который может быть использован для организации временных задержек и подсчета внешних импульсов. В первом случае посредством начальной установки счетчика, счета импульсов, следующих с периодом 80 мкс, и анализа признака переполнения может быть реализована задержка от 80 мкс до 20 мс. Во втором случае внешние импульсы подаются на вход МК Т1; разрешающее время счетчика 7,5 мкс. Для управления таймером в системе команд МК1816 имеются специальные команды.

Р6.69. а) (A)←(R5); прямая адресация приемника и источника информации: в аккумулятор пересылается содержимое регистра R5 ОЗУ; код команды 11111101=FDH;

б) $(A) \leftarrow A1H$; прямая адресация приемника и непосредственная адресация источника: в аккумулятор загружается число $A1H$; код команды (2 байта) $00100011\ 10100001 = 23A1H$;

в) $(A) \leftarrow [(R1)]$; прямая адресация приемника и косвенная адресация источника; в аккумулятор пересылается содержимое ячейки внутреннего ОЗУ, адрес которой хранится в разрядах $0...5$ регистра $R1$ ОЗУ; код команды $11110001 = F1H$;

г) $(A) \leftarrow (T)$; прямая адресация приемника и источника: в аккумулятор пересылается содержимое счетчика таймера; код команды $01000010 = 42H$;

д) $[(R0)] \leftarrow (A)$; прямая адресация источника и косвенная адресация приемника; в ячейку ОЗУ, адрес которой хранится в разрядах $0...5$ регистра $R0$ ОЗУ, пересылается содержимое аккумулятора; код команды $10010000 = 90H$;

е) прямая адресация приемника информации — аккумулятора; косвенная адресация источника информации — ячейки из текущей страницы блока памяти; код команды $10100011 = A3H$;

ж) $(A) \leftarrow [(R0)]$; прямая адресация приемника и косвенная адресация источника; в аккумулятор пересылается содержимое ячейки памяти внешнего ЗУ, адрес которой указан в регистре $R0$ ОЗУ; код команды $10000000 = 80H$;

з) $(A) \leftrightarrow (R4)$; прямая адресация источника и приемника; аккумулятор и регистр $R4$ обмениваются содержимым; код команды $00101100 = 2CH$;

и) $(A) \leftrightarrow [(R1)]$; аккумулятор и ячейка ОЗУ с адресом, указанным в регистре $R1$ ОЗУ, обмениваются содержимым; код команды $00100001 = 21H$.

к) прямая адресация приемника информации — аккумулятора; косвенная адресация источника информации — ячейки из непосредственно указанной 3-й страницы блока памяти; код команды $11100011 = E3H$.

P6.70. а) $(A) \leftarrow (A) + (R5)$; $(A) = C9H$; $(C) = 0$;

б) $(A) \leftarrow (A) + OBH$; $(A) = B5H$; $(C) = 0$;

в) $(A) \leftarrow (A) + [(R0)]$; $(A) = 0CH$; $(C) = 1$;

г) $(R4) \leftarrow (R4) + 1$; $(R4) = 50H$; $(C) = 0$;

д) $[(R1)] \leftarrow [(R1)] + 1$; $[(R1)] = 2BH$; $(C) = 0$;

е) $(A) \leftarrow (A) - 1$; $(A) = 4BH$; $(C) = 0$;

ж) Циклический сдвиг содержимого аккумулятора влево: $(A) = 64H$, $(C) = 0$;

з) Циклический сдвиг содержимого аккумулятора вправо: $(A) = D9H$; $(C) = 1$;

и) Циклический сдвиг содержимого аккумулятора вправо через перенос: $(A) = 59H$; $(C) = 1$.

P6.71. а) $(A) \leftarrow (A) \wedge (R1)$; $(A) = 28H$;

б) $(A) \leftarrow (A) \wedge F0H$; $(A) = A0H$ (с помощью непосредственно заданного операнда $F0H$ выделяется старшая тетрада содержимого аккумулятора);

в) $(A) \leftarrow (A) \vee [(R0)] = (A) \vee 4CH$; $(A) = 5DH$;
 г) $(A) \leftarrow (A) \oplus 61H$; $(A) = D2H$;
 д) $(A) \leftarrow 0$; $(A) = 0$; очищение (обнуление) аккумулятора;
 е) $(A) \leftarrow (\bar{A})$; $(A) = 4CH$; инвертирование содержимого аккумулятора.

Р6.72. а) $(A) \leftarrow (P1)$; ввод данных из порта P1 в аккумулятор;
 б) $(A) \leftarrow (BUS)$; стробируемый ввод данных из порта BUS в аккумулятор;

в) $(P1) \leftarrow (A)$; вывод данных из аккумулятора в порт P1;
 г) $(P2) \leftarrow (P2) \wedge 0FH$; логическое умножение содержимого порта P2 и непосредственно заданного операнда 0FH — выделение младшей тетрады содержимого порта P2;

д) $(BUS) \leftarrow (A)$; вывод из аккумулятора в порт BUS. Байт данных из аккумулятора передается и фиксируется в буферном регистре порта BUS до тех пор, пока он не изменится любой последующей командой, требующей использования порта BUS. Исключение составляет только команда стробируемого ввода через порт BUS INS, при которой ранее выведенный командой OUTL байт все же сохраняется в регистре порта BUS; тем самым обеспечивается его двунаправленность.

Р6.73. Все команды предусматривают переход в пределах ранее выбранного блока памяти, а команды б) ... и) к тому же — в пределах ранее выбранной страницы.

а) безусловный переход по адресу $a_{0...10}$, содержащему 11 бит: $a_{8...10}$ — код страницы, $a_{0...7}$ — код ячейки памяти в пределах страницы. Машинный код команды (2 байта) 01000100 01010100 = 4454H;

б) безусловный переход по косвенному адресу $a = 01000001$, содержащемуся в аккумуляторе. Код команды 10110011 = V3H;

в) комбинированная команда: декремент регистра R4 и переход по адресу a , если в регистре после декрементирования содержится не нуль. Код команды 11101100 11100010 = ECE2H;

г) условный переход по адресу a , если $(C) = 1$, т. е., если есть перенос;

д) то же, если $(C) = 0$, т. е. нет переноса;

е) то же, если $(A) = 0$.

ж) то же, если на тестовом входе МК T0 = 1;

з) то же, если на входе запроса прерывания МК ЗПр = 0;

и) то же, если бит A_3 содержимого аккумулятора $(A_3) = 1$; код команды 01110010 01101000 = 7268H.

В вариантах в) ... и) при выполнении соответствующего условия счетчик команд загружается адресом a , указанным во втором байте команды, а при невыполнении содержимое счетчика увеличивается на две единицы.

Р6.74. а) Безусловный вызов подпрограммы по адресу $a_{0...10}$, содержащему 11 бит; $a_{8...10}$ — код страницы, $a_{0...7}$ — код ячейки памяти в пределах страницы. Переход осуществляется

в пределах ранее выбранного блока памяти. Отличие от команды JMP *a* в том, что при выполнении команды CALL в стеке запоминается содержимое счетчика команд, т. е. адрес возврата.

Код команды 01010100 11010100 = 54D4H;

б) возврат в основную программу по адресу, запомненному в стеке. Командой RET обычно заканчивается подпрограмма.

P6.75. а) Запуск таймера от внутренних сигналов синхронизации МК1816; содержимое таймера инкрементируется каждые 32 машинных цикла (при $f_1 = 6$ МГц — через 80 мкс); диапазон реализуемых задержек от 80 мкс до $256 \cdot 80 = 20,48$ мс;

б) запуск таймера на работу в качестве счетчика внешних импульсов, поступающих по входу МК T1; разрешающее время 7,5 мкс, длительность импульсов не менее 0,5 мкс;

в) останов таймера;

г) разрешение прерывания от таймера; сигнал переполнения таймера вызывает переход к подпрограмме обслуживания прерывания.

P6.76. а) Выбор банка регистров RB0; далее в программе могут использоваться восемь РОНов R0 ... R7 только из этого банка;

б) выбор блока ППЗУ MB1; далее после команд JMP или CALL в программе могут использоваться 2048 ячеек памяти из этого блока;

в) разрешение прерываний; появление на входе МК1816 $\overline{3Pr}$ сигнала логического нуля, инициирующего процедуру обработки запроса прерывания;

г) нет операции; команда используется при организации временной задержки (выполняется за один машинный цикл, при $f_1 = 2$ МГц — за 2,5 мкс), а также в качестве вставки, которая в процессе отладки программы может быть заменена на любую другую команду.

P6.77. Фрагмент программы:

```
RECR:  MOV R0, #23H      ;Загрузка в R0 числа 35
        MOV @R0, #2DH   ;Загрузка в ячейку 35 числа
                        ;2DH
        INC R0           ; $(R0) \leftarrow (R0) + 1$ 
        MOV @R0, #3FH   ;Загрузка в ячейку 36 числа
                        ;3FH
```

P6.78. Поскольку команда декремента в МК1816 может относиться только к аккумулятору, требуется предварительно переслать информацию из ячейки ОЗУ в аккумулятор.

Фрагмент программы:

```
DECM:  MOV R0, #17H     ;Загрузка в R0 числа 23
        XCH A, @(R) 0, A ;Обмен содержимого аккумуля-
                        ;лятора и ячейки 23
```


DEC A
XCH @R0,A

;(A) ← (A) − 1
;Обмен содержимого ячейки
;23 и аккумулятора

P6.79. Фрагмент программы:

SUMMR:	MOV A, #2FH	;Загрузка в аккумулятор ;числа 2FH
	SEL MB0	;Выбор банка памяти MB0
	JMP M1	;Переход к программе M1
M1:	MOVP3 A, @A	;Загрузка в аккумулятор ;содержимого ячейки памяти ;2FH со страницы ;3 блока памяти
	SEL RB0	;Выбор банка регистров ;RB0
	ADD A, R5	;(A) ← (A) + (R5)
	SEL RB1	;Выбор банка регистров ;RB1
	MOV R7, A	;(R7) ← (A)

P6.80. Фрагмент программы:

MASK:	SEL RB0	;Выбор банка регистров RB0
	MOV R0, #29H	;Загрузка в R0 числа 41
	ANL A, @R0	;Логическое умножение $Q \wedge N$
	MOV R0, A	;(R0) ← (A)

Результат вычисления: $(A) = 111011101 \wedge 00000111 = 00000101 = 05H$.

P6.81. Фрагменты программ:

a) OUTR:	SEL RB0	;Выбор банка регистров RB0
	MOV A, R6	;(A) ← (R6)
	OUTL P1, A	;Вывод в порт P1
б) OUTM:	SEL MB0	;Выбор блока памяти MB0
	JMP M1	;Переход к программе M1
M1:	MOV A, #0F1H	;Загрузка аккумулятора чис- ;лом F1H
	MOVP3 A, @A	;Загрузка в аккумулятор со- ;держимого ячейки памяти ;F1H страницы 3
	OUTL P1, A	;Вывод в порт P1

P6.82. Фрагмент программы:

THRESQ:	IN A, P1	;Ввод числа N из порта P1
	XRL A, #Q	;(A) ← $N \oplus Q$
	JZ L1	;Переход к L1 при (A) = 0
	MOV A, #01H	;(A) ← 1
L1:	OUTL P2, A	;Вывод в порт P2
	JMP THRESQ	;Переход к началу програм- ;мы

Команды XRL A, #Q (сложение по mod 2) и JZ позволяют сравнить числа N и Q, так как при $N=Q$ $N \oplus Q = 0$.

P6.83. В системе команд МК1816 нет операции вычитания, но есть операция инвертирования содержимого аккумулятора CPL A. Поскольку $\bar{d} = 2^8 - 1 - d$, получим

$y = d1 - d2 = -2^8 + 1 + d1 + \bar{d2}$. Первое слагаемое образует перенос, который в данном случае учитывать не следует. Программа вычитания:

SUBRA:	CPL A	; (A) $\leftarrow (\bar{A})$
	ADD A, R3	; (A) $\leftarrow (A) + (R3)$
	INC A	; (A) $\leftarrow (A) + 1$
	CLP C	; Очистка триггера переноса

P6.84. В МК1816 нет команды сравнения. Поэтому используем соотношение $N1 + \bar{N2} = 2^8 - 1 + N1 - N2$, где $\bar{N2}$ — инверсия N2.

При выполнении в аккумуляторе операции $N1 + \bar{N2}$:

если $N1 > N2$, то образуется перенос (C) = 1;

если $N1 \leq N2$, то (C) = 0.

Таким образом, программа должна содержать операции инвертирования N2, суммирования N1 и $\bar{N2}$ и проверки признака переноса.

COMP:	IN A, P1	; Ввод N1
	MOV R1, A	; Хранение N1 в R1 ОЗУ
	IN A, P2	; Ввод N2
	MOV R2, A	; Хранение N2 в R2 ОЗУ
	CPL A	; Инвертирование N2
	ADD A, R1	; (A) $\leftarrow N1 + \bar{N2}$
	JC K1	; Переход по метке K1 при
		; $N1 > N2$
	MOV A, R2	; (A) $\leftarrow N2$
	OUTL P2, A	; Вывод N2
	JMP COMP	; Повторение ввода N1 и N2
K1:	MOV A, R1	; (A) $\leftarrow N1$
	OUTL P1, A	; Вывод N1
	JMP COMP	; Повторение ввода N1 и N2.

P6.85. В системе команд МК1816 имеются команды обмена типа XCH, которые позволяют обменивать содержимое аккумулятора и регистров ОЗУ.

P6.86. Фрагмент программы:

START:	CLR A	; Очистка аккумулятора
	MOV T, A	; Очистка счетчика
	STRT CNT	; Запуск счетчика
MT:	MOV A, T	; Загрузка аккумулятора из
		; счетчика

JB5 MET	;Переход к метке MET при ;A ₅ =1, т.е. (A)=32
JMP MT	;Переход к метке MT

P6.87. Фрагмент программы:

COUNT:	STRT CNT	;Запуск счетчика
BEG:	CLR A	;Очистка аккумулятора
	MOV T,A	;Очистка счетчика
L1:	JTF L2	;Условный переход к метке ;L2 по признаку переполне- ;ния счетчика
	JMP L1	;Возврат к L1
L2:	INC A	;Инкремент аккумулятора
	OUTL P1,A	;Вывод (A) в порт P1
	JB8 BEG	;Возврат к метке BEG при ;A ₈ =1, т. е. при (A)=128
	JMP L2	;Возврат к L2

P6.88. При тактовой частоте МК1816 6 МГц во внутреннем счетчике (таймере) состояние инкрементируется каждые 80 мкс. Таким образом, заданный интервал 8 мс соответствует состоянию таймера 100=64H. Вариант программы приведен ниже. Подпрограмма PROG1 выполняется в начале каждого интервала (предполагается, что время ее выполнения меньше 8 мс). Проверка состояния таймера производится с помощью команд XPL и JZ.

INT:	CLR A	;Очистка аккумулятора
	MOV T, A	;Очистка таймера
	STRT T	;Запуск таймера
	CALL PROG1	;Вызов подпрограммы ;PROG1
L1:	MOV A, T	;Пересылка из таймера в ;аккумулятор
	XRL A, #64H	; (A) ← (A) ⊕ 01100100
	JZ INT	;Переход к метке INT, если ; (A)=0
	JMP L1	;Возврат к метке L1
PROG1:		} Подпрограмма
		
	RET	

P6.89. Схема МПС на базе МК1816 с дополнительным внешним ОЗУ емкостью 256 байт приведена на рис. P6.4. ОЗУ показано условно: оно может состоять и из нескольких микросхем. Управление ОЗУ производится через многорежимный буферный регистр (МБР) с помощью сигналов «строб адреса внешней памяти» (САВП), Чт и Зп, снимаемых с выходов МК. Адреса выбираемых ячеек ОЗУ и данные передаются через двухнаправленный внутренний порт BUS.

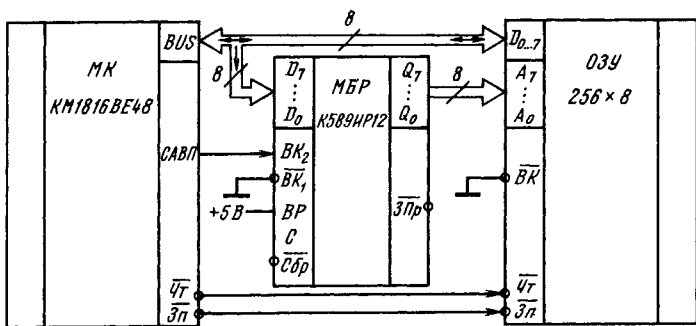


Рис. Р6.4. Схема подключения дополнительного (внешнего) ОЗУ к микроконтроллеру KM1816

При $ВР=1$ информационные выходы МБР постоянно открыты. В исходном состоянии $САВП=ВК_2=0$ и информационные входы МБР от шины BUS отключены. Обращение к внешнему ОЗУ осуществляется с помощью команд пересылки $MOVX A, @Rr$ и $MOVX @Rr, A$ с косвенной адресацией выбираемых ячеек ОЗУ в регистрах $R0$ и $R1$ внутреннего ОЗУ МК. При выполнении этих команд в МК образуется сигнал $САВП=1$, который открывает информационные входы МБР, в результате чего байт адреса ячейки ОЗУ выводится через порт BUS на адресные входы внешнего ОЗУ $A_0...7$. После этого МК выдает сигналы $Зп$, $ЧТ$ и обменивается информацией с ОЗУ опять же через порт BUS , но теперь по входам $D_0...7$ ОЗУ: при $Зп=0$, $ЧТ=1$ информация выводится из МК в ОЗУ, а при $Зп=1$, $ЧТ=0$ — вводится в МК из ОЗУ.

Р6.90. Схема МПС с расширенной памятью программ представлена на рис. Р6.5. При обращении к внутренней памяти программ емкостью 1 Кбайт (адреса $0 \dots 1023$) МК

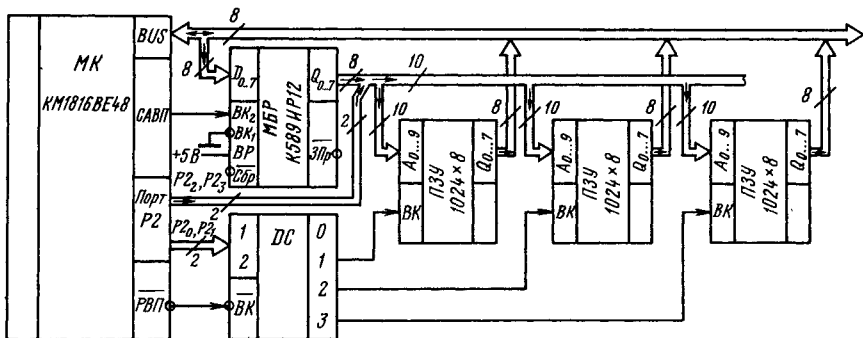


Рис. Р6.5. Схема подключения дополнительного (внешнего) ПЗУ к микроконтроллеру KM1816

не генерирует внешних управляющих сигналов (кроме сигнала САВП, сопровождающего каждый машинный цикл). Начиная с адреса 1024, МК при выполнении команд JMP и CALL автоматически выводит содержимое счетчика команд (12 бит) через порт BUS (биты $A_0 \dots A_7$) и младшую тетраду порта P2 (биты $A_8 \dots A_{11}$), а также формирует сигнал «Разрешение внешней памяти» $\overline{РВП}=0$. Последний используется для фиксации байта адреса $A_0 \dots A_7$ в регистре МБР (биты $A_8 \dots A_{11}$ фиксируются в порте P2). Поскольку при $\overline{ВР}=1$ выходы МБР всегда открыты, адрес выбираемой ячейки $A_0 \dots A_9$ по сигналу $\overline{РВП}=0$ подается параллельно на все БИС ПЗУ.

По окончании сигнала $\overline{РВП}=0$ шина BUS переходит в режим ввода. Поскольку для расширения памяти согласно заданию используются три микросхемы ПЗУ по 1 Кбайт каждая, для выбора микросхемы используется дешифратор.

При использовании внутренней и внешней памяти программы команды безусловного перехода JMP или вызова подпрограммы CALL позволяют перейти к любой ячейке памяти в пределах блока памяти MB0 (адреса 0 ... 2047). Для перехода к блоку памяти MB1 (адреса 2048 ... 4095) необходимо выполнить вначале команду SEL MB1 и уже затем JMP или CALL с указанием соответствующего адреса.

Р6.91. В схеме на рис. Р6.6 для организации трех дополнительных портов ввода-вывода используется параллельный периферийный адаптер (ППА) KP580BB55, информационные входы $D_0 \dots D_7$ которого подсоединены к порту BUS МК, а адресные входы $A_1 \dots A_0$ — к двум линиям порта P2 МК. ППА адресуется как внешняя память, т. е. с помощью команды MOVX. Для настройки ППА необходимо предварительно вывести через порт BUS командой OUTL BUS, A управляющее слово.

Р6.92 — Р6.95. Принцип действия МПС на базе МК К1804 описан в [5, 17, 23].

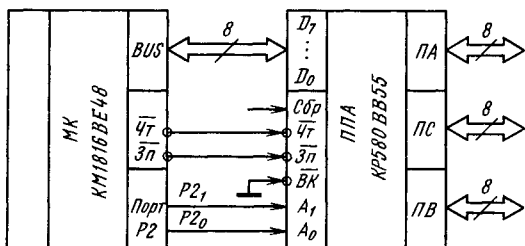


Рис. Р6.6. Схема подключения ППА KP580BB55 к микроконтроллеру KM1816

Р6.96. Управляющее устройство (УУ) МПС, построенной на базе секционированных МП, обеспечивает при однократном «запуске» выполнение одной микрокоманды, т. е. одной или нескольких микроопераций, осуществляемых в течение одного такта. УУ однокристалльного МП обеспечивает выполнение командного цикла, содержащего несколько последовательно выполняемых машинных циклов, каждый из которых, в свою очередь, состоит из нескольких тактов.

Р6.97. См. [5, 17].

Р6.98. При программировании МПС, построенных на базе однокристалльного МП КР580ВМ80, программист имеет в своем распоряжении систему команд, из которых строит программы. При программировании МПС, построенных на базе секционированных МП, программист пользуется набором микрокоманд, из которых составляет микропрограммы; эти последние используются далее при составлении программ.

Р6.99. С помощью одной микрокоманды могут быть реализованы команды из п. б), г), д), ж), к), м).

Р6.100. Расшифровку микрокоманды (см. табл. 6.1) целесообразно начинать с поля кода операции КОП АЛУ ($I_5 I_4 I_3$), затем расшифровать коды источников операндов КИС ($I_2 I_1 I_0$) и приемника результата КПР ($I_8 I_7 I_6$), модификаторы сдвига MS_1 , MS_2 и, наконец, поле управления переходами Р и поле адреса следующей микрокоманды BR (см. табл. 6.2—6.5 и рис. 6.4).

а) Загрузка $РОН_8$ числом 6; вывод этого числа в выходную шину данных ЦПЭ Y; переход к следующей по порядку микрокоманде.

б) Вычитание из содержимого $РОН_{10}$ содержимого $РОН_5$ с загрузкой результата в $РОН_{10}$ и переход к следующей по порядку микрокоманде.

в) Загрузка $РОН_7$ числом $2 \cdot 7 = 14$ с выводом в выходную шину данных Y числа 7; переход к следующей по порядку микрокоманде.

г) Сложение по модулю 2 содержимого $РОН_{12}$ и $РОН_3$ с загрузкой результата в $РОН_3$ и переход к следующей по порядку микрокоманде.

д) Загрузка $РОН_5$ числом 9; вывод этого числа в выходную шину данных Y и безусловный переход к микрокоманде по адресу 0101.

е) Сложение содержимого $РОН_6$ и $РОН_9$ с загрузкой суммы в $РОН_9$; вывод суммы в выходную шину данных Y; безусловный переход к подмикропрограмме по адресу 1010.

ж) Вычитание из содержимого $РОН_0$ содержимого $РОН_{15}$ с загрузкой удвоенной разности в $РОН_{15}$ и в регистр Q; переход к следующей по порядку микрокоманде.

3) Логическое поразрядное сложение содержимого POH_3 и POH_4 ; загрузка суммы в регистр Q; возврат из подмикропрограммы.

Р6.101. Два варианта микрокоманд загрузки POH_0 в МПС (см. рис. 6.3) константой 5 с одновременным выводом содержимого POH_0 в выходную шину данных ЦПЭ Y представлены в табл. Р6.2 (коды, не указанные в табл. Р6.2, могут иметь любые значения).

Таблица Р6.2

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₁	0001				011		111	0	000		0000	0101	(POH_0) ← 5
МК ₂	0010				011		111		011		0000	0101	(POH_0) ← 5

В МК₁ КОП АЛУ 000 обеспечивает выполнение операции $F = (R) + (S) + C_0$. При $C_0 = 0$ и коде источника КИС = 111 (R) = D, (S) = 0, получаем результат $F = D = 0101 = 5$.

Код приемника КПР = 011 обеспечивает загрузку результата F в регистр B = 0000, т. е. в POH_0 , и вывод этого результата в выходную шину данных ЦПЭ Y.

В МК₂ КОП АЛУ 011 обеспечивают выполнение операции $F = R \vee S$. При коде источника КИС = 111 (R) = D, (S) = 0 и получаем $F = D = 5$. Далее аналогично МК₁.

Р6.102. Два варианта микрокоманд «очистки» регистра Q ЦПЭ представлены в табл. Р6.3.

Таблица Р6.3

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₁	0001				000		010		100				(Q) ← 0
МК ₂	0010				000		111		011			0000	(Q) ← 0

В МК₁ код операции КОП = 100 и код источников операндов КИС = 010 обеспечивают выполнение операции $F = R \wedge S = 0 \wedge 0 = 0$.

В МК₂ код операции КОП = 011, код источников операндов КИС = 111 и код в поле данных D = 0000 обеспечивают выполнение операции $F = R \vee S = D \vee 0 = 0$. Код приемника КПР = 000 обеспечивает в обеих микрокомандах загрузку $F = 0$ в регистр Q.

Р6.103. Микрокоманда представлена в табл. Р6.4.

Таблица Р6.4

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000	0011	0001		011		001	0	000	0100	0101		(РОН ₅) ← (РОН ₄) + + (РОН ₅); пере- ход к 0011

Р6.104. Признак (флаг) АЛУ — «нечетный результат», определяемый значением младшего разряда результата $F_0=1$, в МП К1804 не фиксируется. Поэтому в приведенной в табл. Р6.5 микропрограмме этот признак формируется искусственно за счет операции сдвига.

Таблица Р6.5

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	1000		0010	0	101	1	011		011		0110		(РОН ₆) ← F ₀ 000
МК ₁	1001	1110	1101		001								К 1110 при F ₀ =1

МК₀ при КОП=101 и модификаторах сдвига MS₂=0 и MS₁=1 обеспечивает в РОН₆ сдвиг символа F₀ в позицию старшего разряда результата F₃. МК₁ за счет кода P=1101 — условный переход к заданному в поле BR адресу 1110 при F₃=1.

Р6.105. Микропрограмма сравнения содержимого двух РО-Нов с переходом к заданному адресу при их равенстве представлена в табл. Р6.6. МК₀ при КОП АЛУ 001 обеспечивает выполнение операции $F=(S)-(R)-1+C_0$. При C₀=1, коде источника КИС=001, A=0000 и B=1111 получим S=B, R=A, $F=(B)-(A)=(РОН_{15})-(РОН_0)$.

Таблица Р6.6

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000		0010		001		001	1	001	0000	1111		F=(РОН ₀)-(РОН ₁₅)
МК ₁	0001	0011	1100		001								К 0011 при F=0

МК₁ при коде P=1100 обеспечивает переход к адресу BR=0011 при F=0, т. е. при (РОН₀)=(РОН₁₅).

Р6.106. Микропрограмма сравнения содержимого POH_0 с константой $k=0101$ с переходом к заданному адресу при их равенстве представлена в табл. Р6.7.

Таблица Р6.7

МК	Адрес МК	BR	P	MS ₂	КПП	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₁	1000		0010		011		101	1	001	0000	0001	0101	(POH ₁)←(POH ₀)-5 K 1101 при F=0
МК ₂	1001	1101	1100		001								

В МК₁ используется операция вычитания с загрузкой разности в POH₁, в МК₂ обеспечивается переход к адресу 1101 при условии (POH₁)=0, т. е. (POH₀)-5=0.

Р6.107. Один из возможных вариантов микропрограммы представлен в табл. Р6.8.

Микрокоманда МК₀ обеспечивает выполнение операции $F=(R)+(S)+C_0=(POH_0)+(POH_1)$, загрузку результата в POH₁ и выдачу его на выходную шину данных ЦПЭ Y. Код P=0010 обеспечивает последовательный переход к микрокоманде МК₁, результатом выполнения которой является загрузка полученной суммы в регистр Q.

Таблица Р6.8

МК	Адрес МК	BR	P	MS ₂	КПП	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000		0010		011		001	0	000	0000	0001		(POH ₁)←(POH ₀)+ +(POH ₁) (Q)←(POH ₁)
МК ₁	0001		0010		000		011	0	000		0001		

Р6.108. Микропрограмма для схемы алгоритма рис. 6.5 представлена в табл. Р6.9. Для организации счета используется POH₁₅. Принято k=10.

Таблица Р6.9

МК	Адрес МК	BR	P	MS ₂	КПП	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000		0010		011		111	0	000		1111	1010	(POH ₁₅)←10 (POH ₁₅)←F= =(POH ₁₅)-1 K МК ₁ при F≠0
МК ₁	0001		0010		010		011	0	001		1111		
МК ₂	0010	0001	0100		001								

Р6.109. Микропрограмма двоичного суммирующего счетчика с модулем счета $k=16$ представлена в табл. Р6.10. Поскольку счетчик должен иметь 16 состояний, а РОНЫ МП К1804 имеют по четыре двоичных разряда, для фиксации состояний счетчика достаточно использовать один РОН, например РОН₀.

Таблица Р6.10

МК	Адрес МК	ВР	Р	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	А	В	D	Примечание
МК ₀	0000	0000	0001		011		011	1	000		0000		(РОН ₀)←(РОН ₀)+1; повтор МК ₀

МК₀ обеспечивает инкрементирование (прибавление единицы) с помощью операции $F=(R)+(S)+C_0$ (КОП АЛУ 000). Принято $C_0=1$, источники операндов: $R=0$, $S=B$ (код источника КИС=011); при коде приемника КПП=011 и $B=0000$ результат $F=(B)+1$ поступает в РОН₀ и в выходную шину данных ЦПЭ.

Код $P=0001$ обеспечивает безусловный переход к адресу $ВР=0000$, т. е. повторение МК₀. Таким образом, микропрограмма состоит из одной микрокоманды, которая циклически повторяется.

Р6.110. Решение аналогично решению задачи 6.109. (Можно использовать КОП АЛУ 001 при $C_0=0$.)

Р6.111. а) Поскольку $k=256=2^8=(2^4)^2$, для фиксации состояний счетчика необходимо использовать два РОНа МП, например РОН₀ и РОН₁. Схема алгоритма показана на рис. Р6.7. Младшие разряды кода состояний счетчика фиксируются в РОН₀, старшие — в РОН₁. Пока $(РОН_0) \leq 15$, новое состояние формируется инкрементированием РОН₀. При $(РОН_0)=16$, или, что то же самое, $(РОН_0)=0$, инкрементируется РОН₁. После заполнения РОН₁ и РОН₀ цикл повторяется.

Микропрограмма представлена в табл. Р6.11.

Микрокоманды МК₀ и МК₁ обеспечивают начальную установку счетчика в 0; МК₂ и МК₃ — выдачу текущего состояния счетчика в регистр Q. МК₄ обеспечивает инкрементирование РОН₀, МК₅ — условный переход к МК₂ при $(РОН_0) \neq 0$,

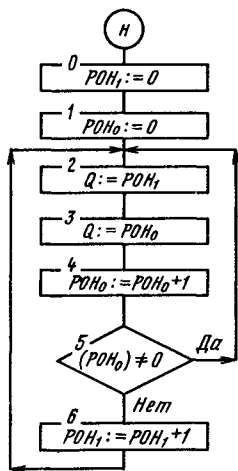


Рис. Р6.7. Схема алгоритма счетчика с модулем счета $k=256$

МК₆ — инкрементирование РОН₁ и безусловный переход к МК₂, т. е. повторение цикла счета.

Таблица Р6.11

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000		0010		011		111	0	000		0001	0000	(РОН ₁)←0
МК ₁	0001		0010		011		111	0	000		0000	0000	(РОН ₀)←0
МК ₂	0010		0010		000		011	0	000		0001		(Q)←(РОН ₁)
МК ₃	0011		0010		000		011	0	000		0000		(Q)←(РОН ₀)
МК ₄	0100		0010		001		011	1	000		0000		(РОН ₀)←(РОН ₀)+1
МК ₅	0101	0010	0000		001								к МК ₂ при (РОН ₀)≠0
МК ₆	0110	0010	0001		001		011	1	000		0001		(РОН ₁)←(РОН ₁)+1; к МК ₂

Р6.112. а) Числа z_i последовательности связаны рекуррентным соотношением $z_i = z_{i-1} + 3 \pmod{16}$. Микропрограмма представлена в табл. Р6.12.

Таблица Р6.12

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000		0010		011		111	0	000		0000	0000	(РОН ₀)←0
МК ₁	0001	0001	0001		011		110	0	000		0000	0011	(РОН ₀)←(РОН ₀)+3; переход к МК ₁

Суммирование по модулю 16 при КОП=000 обеспечивается за счет того, что РОН₀ содержит 4 разряда, а перенос при переполнении не учитывается.

Р6.113. Замена кода Р=0100 на код Р=0000, вообще говоря, недопустима. Хотя оба кода обеспечивают переход на адрес из поля BR при выполнении одинакового условия F≠0, но код Р=0100 обеспечивает также запоминание в стеке адреса возврата из подмикропрограммы, тогда как при Р=0000 возврат будет обеспечиваться только в том случае, если в последней команде подмикропрограммы указан адрес возврата.

Р6.114. Микропрограмма представлена в табл. Р6.13. Переход в МК₂ к подмикропрограмме обеспечивается кодом Р=0100 и указанием адреса перехода BR=1100. Возврат в МК₁₄ из подмикропрограммы обеспечивается кодом Р=0110; адрес возврата МК₃ здесь указывать не нужно, так как при

переходе в МК₂ к подмикропрограмме он автоматически запомнился в стеке.

Таблица Р6.13

МК	Адрес МК	ВР	Р	Примечание
МК ₀	0000	1100	0010	К МК ₁
МК ₁	0001		0010	К МК ₂
МК ₂	0010		0100	К МК ₁₂ при F ≠ 0
МК ₃	0011		0010	К МК ₄
МК ₄	0100	0000	0001	К МК ₀
МК ₁₂	1100		0010	К МК ₁₃
МК ₁₃	1101		0010	К МК ₁₄
МК ₁₄	1110		0110	Возврат к МК ₃

Р6.115. См. решение задачи 6.114.

Р6.116. Для определения четности или нечетности числа N можно выполнить логическое поразрядное умножение числа N на 0001Н (выделение младшего разряда) и условный переход по нулевому результату.

Р6.117. Длительность выполнения микропрограммы из задачи 6.114 (см. табл. Р6.13) $t = N/f_T$, где N — число микрокоманд в микропрограмме; $f_T = 5$ МГц — тактовая частота. Если результат выполнения МК₂ ненулевой ($F \neq 0$), то $N = 8$ и $t = 1,6$ мкс; если же в МК₂ $F = 0$, то подмикропрограмма не вызывается, $N = 5$ и $t = 1,0$ мкс.

Р6.118. а) Микропрограмма для вычисления выражения $F = 2\phi_1 + \phi_2$ приведена в табл. Р6.14: МК₀ — загрузка $2\phi_1$ в РОН₀ (код приемника КПП = 111 при модификаторах MS₁ = MS₂ = 0 обеспечивает сдвиг влево на один разряд, т. е. удвоение результата); МК₁ — загрузка ϕ_2 в РОН₁; МК₂ — вычисление $F = 2\phi_1 + \phi_2$; МК₃ — вывод F в регистр Q.

Таблица Р6.14

МК	Адрес МК	ВР	Р	MS ₂	КПП	MS ₁	КИС	C ₀	КОП	А	В	D	Примечание
МК ₀	0000		0010	0	111	0	111	0	000		0000	ϕ_1	(РОН ₀) ← $2\phi_1$
МК ₁	0001		0010		011		111	0	000		0001	ϕ_2	(РОН ₁) ← ϕ_2
МК ₂	0010		0010		011		001	0	000	0001	0000		(РОН ₀) ← F
МК ₃	0011		0010		000		011		011		0000		(Q) ← (РОН ₀)

Длительность выполнения микропрограммы

$$t = N/f_T = 4/(5 \cdot 10^6) = 0,8 \text{ мкс},$$

где N — число микрокоманд в микропрограмме; $f_T = 5$ МГц — тактовая частота.

Р6.119. а) Микропрограмма приведена в табл. Р6.15. В МК₂ производится проверка $\phi_1 - 2 < 0$; если неравенство выполняется, результат отрицательный, т. е. старший разряд F_3 (в дополнительном коде) равен 1.

Таблица Р6.15

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000		0010		011		111		011		0000	ϕ_1	(РОН ₀) ← ϕ_1
МК ₁	0001		0010		001		101	1	001	0000		0010	$F = \phi_1 - 2$
МК ₂	0010	0110	1101		001								К МК ₆ при $F_3 = 1$
МК ₃	0011		0010	0	111	0	111		011		0001	ϕ_2	(РОН ₁) ← $2\phi_2$
МК ₄	0100		0010		011		001	0	000	0000	0001		(РОН ₁) ← $\phi_1 + 2\phi_2$
МК ₅	0101	0000	0001		000		011		011		0001		(Y) ← $\phi_1 + 2\phi_2$; к МК ₀
МК ₆	0110	0000	0001		000		011		011		0000		(Y) ← ϕ_1 ; к МК ₀

д), е) Умножение и возведение в степень следует выполнить посредством суммирования.

ж) Следует записать в РОНах массив чисел $\lfloor \log_2 \phi_1 \rfloor$ и затем по ϕ_1 выбирать нужное число.

Р6.120. Один из возможных вариантов микропрограммы для 4-разрядного кольцевого регистра представлен в табл. Р6.16.

В микрокомандах МК₂, МК₄, МК₆ используются операции сдвига влево на один разряд; МК₇ обеспечивает возврат к МК₀.

Таблица Р6.16

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000		0010		011		111	0	000		0000	0001	(РОН ₀) ← 0001
МК ₁	0001		0010		000		011	0	000		0000		(Q) ← (РОН ₀)
МК ₂	0010		0010	0	110	0	011	0	000		0000		(РОН ₀) ← 0010
МК ₃	0011		0010		000		011	0	000		0000		(Q) ← (РОН ₀)
МК ₄	0100		0010	0	110	0	011	0	000		0000		(РОН ₀) ← 0100
МК ₅	0101		0010		000		011		000		0000		(Q) ← (РОН ₀)
МК ₆	0110		0010	0	110	0	011	0	000		0000		(РОН ₀) ← 1000
МК ₇	0111	0000	0001		000		011		000		0000		(Q) ← (РОН ₀); к МК ₀

Р6.121. Для формирования 8-разрядного кода используются два РОНа. Схема алгоритма представлена на рис. Р6.8. Старшие разряды кода хранятся в РОН₁, младшие — в РОН₀. Вложенный цикл (операции 5...10) обеспечивает сдвиг кодов в РОН₁ и РОН₀ влево, вывод результата, декремент РОН₁₅ и проверку выполнения условия РОН₁₅ ≠ 0. Поскольку единица имеется в данном цикле только в одном из РОНов (в другом

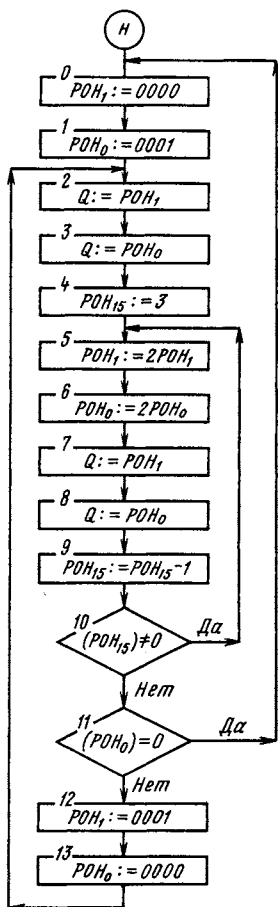


Рис. Р6.8. Схема алгоритма кольцевого регистра

0000), то сдвиг фактически происходит только в этом РОНе, причем повторяется он три раза. Затем проверяется условие $(РОН_0)=0$. Если оно не выполняется (это означает, что в данный момент $(РОН_1)=0000$, $(РОН_0)=1000$), то $РОН_1$ загружается единицей, $РОН_0$ — нулем и далее сдвиги осуществляются в $РОН_1$. Если условие $(РОН_0)=0$ выполняется (т. е. в данный момент $(РОН_0)=0000$, $(РОН_1)=1000$), то единицей загружается $РОН_0$, нулем — $РОН_1$ и сдвиги осуществляются в $РОН_0$.

Микропрограмма представлена в табл. Р6.17. Для организации счетчика циклов используется $РОН_{15}$. Микрокоманды $МК_5$ и $МК_6$ обеспечивают сдвиги кодов влево на один разряд. Информация выводится в регистр Q в микрокомандах $МК_2$, $МК_3$, $МК_7$ и $МК_8$. Микрокоманда $МК_{14}$ обеспечивает помимо загрузки $РОН_0$ нулем безусловный переход к $МК_2$.

Р6.122. См. решение задачи 6.121.

Р6.123. Конвейерная обработка сигналов, гарвардская архитектура, аппаратная реализация функций (умножение, сдвиг влево на 0...15 разрядов, инкремент/декремент вспомогательных регистров), которые обычно реализуются программно.

Р6.124. 16 двоичных разрядов.

Р6.125. 12 двоичных разрядов.

Р6.126. 3 двоичных разряда.

Р6.127. 32 двоичных разряда.

Р6.128. Функции счетчика циклов, регистров для временного хранения данных, косвенной адресации памяти данных.

Р6.129. Гарвардская архитектура предполагает наличие двух независимых ЗУ программ и данных для совмещения во времени операций выборки и выполнения команд. Модификация заключается в возможности обмена информацией между ЗУ программ и данных.

Р6.130. Прямой и косвенный режимы адресации.

Р6.131. При прямой адресации памяти данных поле адреса в команде содержит 7 разрядов, что не позволяет использовать полностью внутреннее ЗУ данных (емкость 144 слова), поэтому внутреннее ЗУ разбито на две страницы. При косвенной адресации указатель страниц не используется.

Таблица Р6.17

МК	Адрес МК	BR	P	MS ₂	КПР	MS ₁	КИС	C ₀	КОП	A	B	D	Примечание
МК ₀	0000		0010		011		111		011		0001	0000	(POH ₁) ← 0
МК ₁	0001		0010		011		111		011		0000	0001	(POH ₀) ← 1
МК ₂	0010		0010		000		011	0	000		0001		(Q) ← (POH ₁)
МК ₃	0011		0010		000		011	0	000		0000		(Q) ← (POH ₀)
МК ₄	0100		0010		011		111		011		1111	0011	(POH ₁₅) ← 3
МК ₅	0101		0010	0	111	0	011	0	000		0001		(POH ₁) ← 2(POH ₁)
МК ₆	0110		0010	0	111	0	011	0	000		0000		(POH ₀) ← 2(POH ₀)
МК ₇	0111		0010		000		011	0	000		0001		(Q) ← (POH ₁)
МК ₈	1000		0010		000		011	0	000		0000		(Q) ← (POH ₀)
МК ₉	1001		0010		011		011	0	001		1111		(POH ₁₅) ← ← (POH ₁₅) - 1
МК ₁₀	1010	0101	0000		001								К МК ₅ при (POH ₁₅) ≠ 0
МК ₁₁	1011		0010		011		011		011		0000		(POH ₀) ← ← (POH ₀) + 0
МК ₁₂	1100	0000	1100		001								К МК ₀ при (POH ₀) = 0
МК ₁₃	1101		0010		011		111		011		0001	0001	(POH ₁) ← 1
МК ₁₄	1110	0010	0001		011		111		011		0000	0000	(POH ₀) ← 0; к МК ₂

Р6.132. Режим увеличения состояния программного счетчика РС на единицу, режим загрузки РС из стека, режим загрузки РС 12-разрядным адресом, извлекаемым из команды перехода. Программная память всегда адресуется содержимым РС.

Р6.133. Подача на вход $\overline{RS}$ напряжения низкого уровня соответствует функции СБРОС, при этом управляющие сигналы на выходах $\overline{DEN}$, $\overline{WE}$ и $\overline{MEN}$ устанавливаются на высоком уровне, шина данных (D15...D0) переходит в «третье» состояние, программный счетчик и адресная шина (A11...A0) устанавливаются в нуль, запрещаются прерывания. Процессор TMS 32010 может удерживаться в состоянии «Сброс» неограниченное время.

Р6.134. Один из двух вспомогательных регистров является текущим, т. е. доступным для использования. Текущий вспомогательный регистр определяется состоянием указателя вспомогательных регистров ARP.

Р6.135. Две команды: первая загружает один из сомножителей в регистр T, вторая — команда умножения.

Р6.136. Глубина вложений подпрограмм и обработки прерываний — не больше четырех. При этом не произойдет переполнения стека, если не выполняются команды TBLR и TBLW, которые используют один уровень стека. Существует программная возможность переслать содержимое вершины стека в аккумулятор, а затем и в ОЗУ данных. Это позволяет

расширить емкость стека за счет ОЗУ данных, но существенно ухудшает быстродействие.

P6.137. а) $(ARP)=1$, $(AR0)=6$, $(AR1)=8$; б) $(ARP)=1$, $(AR0)=4$, $(AR1)=2$; в) $(ARP)=0$, $(AR0)=1FF$.

P6.138. а) $15_{(16)}$; б) $2A$; в) 6 ; г) $0F$; д) $0E$.

P6.139.

Команда	Комментарий
IN 40,3	$[40] \leftarrow (\text{порт } 3)$
LT 40	$T \leftarrow [40]$
MPY 40	$P \leftarrow (T) \times [40]$
PAC	$ACC \leftarrow (P)$
SACL 41	$[41] \leftarrow (ACC)$ (16 младших разрядов)
OUT 41,5	Порт $5 \leftarrow [41]$

P6.140.

Команда	Комментарий
IN 10,3	$[10] \leftarrow (\text{порт } 3)$
IN 11,4	$[11] \leftarrow (\text{порт } 4)$
ZAC	$ACC \leftarrow 0$
LT 10	$T \leftarrow [10]$
MPYK 9	$P \leftarrow (T) \times 9$
LTA 11	$ACC \leftarrow (P) + (ACC); T \leftarrow [11]$
MPYK 7	$P \leftarrow (T) \times 7$
APAC	$ACC \leftarrow (P) + (ACC)$
SACL 12	$[12] \leftarrow (ACC)$ (16 младших разрядов)
OUT 12,1	Порт $1 \leftarrow [12]$

P6.141.

Адрес	Команда	Комментарий
100	LARP 0	$ARP \leftarrow 0$
101	LARK AR0,9	$AR0 \leftarrow 9$
102	SAR AR0, *	$[(AR0)] \leftarrow (AR0)$
103	BANZ	Если $(AR0) \neq 0$, то $AR0 \leftarrow (AR0) - 1$ и
104	102	$PC \leftarrow 102$, иначе $PC \leftarrow (PC) + 1$

P6.142.

Адрес	Команда	Комментарий
0	B	PC ← 50 (переход к команде, записанной в ячейке 50 программной памяти)
1	50	
⋮	⋮	
50	ZAC	ACC ← 0
51	LARK AR0,0	ARO ← 0
52	LARK AR1,0	AR1 ← 0
53	LARP 0	ARP ← 0
54	B	PC ← 100
55	100	

P6.143.

Адрес	Команда	Комментарий
0	B	
1	60	PC ← 60
⋮	⋮	
60	LARK AR0,0	AR0 ← 0
61	LARK AR1,8F	AR1 ← 8F
62	LARP 1	ARP ← 1
63	SAR AR0,*	[(AR1)] ← (AR0)
64	BANZ	Если (AR1) ≠ 0, то AR1 ← (AR1) - 1 и
65	63	PC ← 63, иначе PC ← (PC) + 1
66	B	} PC ← 200
67	200	

P6.144.

Адрес	Команда	Комментарий
100	LARK, AR0,9	AR0 ← 9
101	LARP 0	ARP ← 0
102	ZAC	ACC ← 0
103	ADD*	ACC ← (ACC) + [(AR0)]
104	BANZ	Если (AR0) ≠ 0, то AR0 ← (AR0) - 1 и
105	103	PC ← 103, иначе PC ← (PC) + 1
106	SACL 10	[10] ← (ACC) (16 младших разрядов)
107	OUT 10,2	Порт 2 ← [10]

Р6.145.

Адрес	Команда	Комментарий
100	LARK AR0,9	$AR0 \leftarrow 9$
101	LARK AR1,19	$AR1 \leftarrow 19$
102	LARP 1	$ARP \leftarrow 1$
103	ZAC	$ACC \leftarrow 0$
104	LT $\ast-,0$	$T \leftarrow [(AR1)]; ARP \leftarrow 0$
105	SAR AR0, $\ast$	$[(AR0)] \leftarrow (AR0)$
106	MPY $\ast,1$	$P \leftarrow (T) \times [(AR0)]; ARP \leftarrow 1$
107	LTA $\ast-,0$	$T \leftarrow [(AR1)]; ACC \leftarrow (P) + (ACC); ARP \leftarrow 0$
108	BANZ	Если $(AR0) \neq 0$, то $AR0 \leftarrow (AR0) - 1$ и
109	105	$PC \leftarrow 105$, иначе $PC \leftarrow (PC) + 1$
10A	SACL 20	$[20] \leftarrow (ACC)$ (16 младших разрядов)
10B	OUT 20,5	Порт $5 \leftarrow [20]$

Р6.146.

Адрес	Команда	Комментарий
100	IN 10, 0	$[10] \leftarrow (\text{Порт } 0)$
101	LARK AR0,8	$AR0 \leftarrow 8$
102	LARK AR1,18	$AR1 \leftarrow 18$
103	ZAC	$ACC \leftarrow 0$
104	LARP 1	$ARP \leftarrow 1$
105	LT $\ast-,0$	$T \leftarrow [(AR1)]; AR1 \leftarrow (AR1) - 1; ARP \leftarrow 0$
106	MPY $\ast,1$	$P \leftarrow (T) \times [(AR0)]; ARP \leftarrow 1$
107	LTD $\ast-,0$	$T \leftarrow [(AR1)]; [(AR1) + 1] \leftarrow [(AR1)];$ $ACC \leftarrow (ACC) + (P); AR1 \leftarrow (AR1) - 1;$ $ARP \leftarrow 0$
108	BANZ	$\left\{ \begin{array}{l} \text{Если } (AR0) \neq 0, \text{ то } AR0 \leftarrow (AR0) - 1 \text{ и} \\ PC \leftarrow 106, \text{ иначе } PC \leftarrow (PC) + 1 \end{array} \right.$
109	106	
10A	SACH 20,1	$[20] \leftarrow (ACC) \times 2^1$ (старшие разряды)
10B	OUT 20,2	Порт $2 \leftarrow [20]$
10C	B	$\left\{ \begin{array}{l} PC \leftarrow 100 \end{array} \right.$
10D	100	

Р6.147. Нерекursивный цифровой фильтр можно реализовать циклическим повторением фрагмента программы (см. Р6.146):

```

MET: MPY  $\ast,1$ 
      LTD  $\ast-,0$ 
      BANZ
      MET
  
```

Время выполнения этого фрагмента — 4 такта по 200 нс, т. е. 800 нс. Для реализации заданного фильтра 40-го порядка требуется выполнить приведенный фрагмент в цикле 40 раз, при этом время выполнения 32 мкс. При частоте дискретизации 48 кГц каждый отсчет поступает на вход цифрового фильтра через 20,8 мкс, следовательно, данное решение неприемлемо по быстродействию. Это объясняется тем, что половина времени уходит на выполнение команды условного перехода BAZN MET. Поэтому в данном случае целесообразно отказаться от циклического построения программы и построить линейную программу, состоящую из многократно повторяющихся команд MPY и LTD. Это позволяет исключить команду условного перехода и улучшить быстродействие, но требует увеличения программы и объема программной памяти. Для составления программы необходимо распределить ячейки внутренней памяти данных TMS 32010. Коэффициенты фильтра $a_0 \dots a_{39}$ будут храниться в ячейках памяти с адресами 0...27 (в шестнадцатеричной системе счисления), отсчеты $x(n) \dots x(n-39)$ в ячейках 30...57. Программа работы фильтра записывается в ПЗУ программ с адреса 0A, так как в начальных адресах записывается программа инициализации работы устройства (см. ниже). Программа работы фильтра может иметь следующий вид:

Адрес	Команда	Комментарий
0A	IN 30, 0	[30] ← (Порт 0) (ввод отсчета входного сигнала $x(n)$)
0B	LARK AR0,27	AR0 ← 27 (адрес a_{39})
0C	LARK AR1,57	AR1 ← 57 (адрес $x(n-39)$)
0D	ZAC	ACC ← 0
0E	LAR P1	ARP ← 1
0F	LT * —, 0	T ← [(AR1)]; AR1 ← (AR1) - 1; ARP ← 0
10	MPY * —, 1	P ← (T) × [(AR0)]; AR0 ← (AR0) - 1; ARP ← 1
11	LTD * —, 0	T ← [(AR1)]; [(AR1) + 1] ← [(AR1)]; ACC ← (ACC) + P; AR1 ← (AR1) - 1; ARP ← 0
		Адреса с 12 по 5D занимают повторяющиеся команды MPY * —, 1 и LTD * —, 0
5E	MPY *, 1	P ← (T) × [(AR0)]; ARP ← 1
5F	APAC	ACC ← (ACC) + P
60	SACH 7F, 1	[7F] ← (ACC) × 2 ¹ (старшие разряды)
61	OUT 7F, 1	(Порт 1) ← [7F] (вывод $y(n)$)
62	NOP	

Адрес	Команда	Комментарий
63	BIOZ	Если $\overline{\text{BIO}}=0$, то $\text{PC} \leftarrow 62$ (ожидание нового отсчета $x(n)$), иначе $\text{PC} \leftarrow 65$
64	62	$\text{PC} \leftarrow 0A$ (безусловный переход к адресу 0A)
65	B	
66	0A	

Коэффициенты фильтра a_i могут также храниться в ПЗУ по адресам 67...8E и в начале работы устройства (после включения питания) переписываться во внутреннее ОЗУ данных процессора TMS 32010. Для этой цели служит следующая программа:

Адрес	Команда	Комментарий
0	LARK AR0,8E	$\text{AR0} \leftarrow 8E$ (адрес a_{39} в ПЗУ)
1	LARK AR1,27	$\text{AR1} \leftarrow 27$ (адрес a_{39} в ОЗУ)
2	LARP 1	$\text{ARP} \leftarrow 1$
3	LDPK 0	$\text{DP} \leftarrow 0$
4	SAR AR0,7F	$[7F] \leftarrow (\text{AR0})$
5	LAC 7F,0	$\text{ACC} \leftarrow [7F]$
6	TBLR *,0	Перепись коэффициентов a_i из ПЗУ (адрес определяется содержимым ACC) в ОЗУ (адрес определяется содержимым AR1), $\text{ARP} \leftarrow 0$
7	MAR *,—,1	$\text{AR0} \leftarrow (\text{AR0}) - 1$; $\text{ARP} \leftarrow 1$
8	BANZ	} Если $(\text{AR1}) \neq 0$, то $\text{AR1} \leftarrow (\text{AR1}) - 1$ } и $\text{PC} \leftarrow 4$, иначе $\text{PC} \leftarrow 0A$
9	4	

При этом требуется ПЗУ объемом не менее 143 слов.

Схема устройства приведена на рис. P6.9. Регистр RG 1 служит для хранения входных отсчетов $x(n)$ и может быть выполнен, например, на четырех корпусах ИС K1804ИР1 или на любых других ИС регистров с ТТЛ-уровнями и тремя состояниями на выходе. ПЗУ (ROM) может состоять из четырех корпусов K556PT11 или других ИС ПЗУ, обеспечивающих требуемый объем памяти и имеющих 3 состояния на выходе. RG 2 — выходной регистр для хранения $y(n)$, который также может быть реализован, например, на четырех ИС K1804ИР1. Входом фильтра является вход регистра RG 1. Двухнаправленная шина данных процессора TMS 32010 (D0...D15) соединена

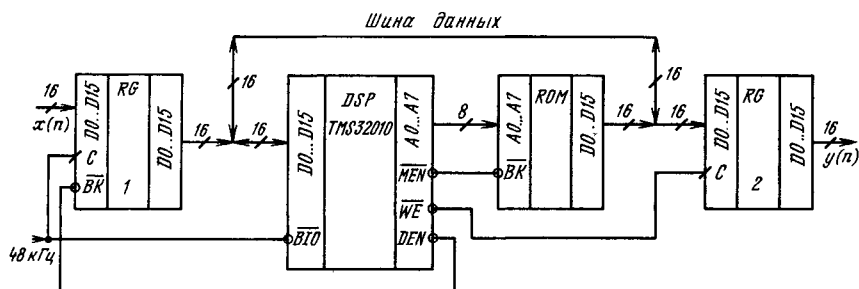


Рис. Р6.9. Схема цифрового фильтра на основе МП TMS 32010

с выходом регистра RG 1, выходом ПЗУ и входом регистра RG 2, выход которого является выходом цифрового фильтра. Восемь разрядов шины адреса процессора ($A0...A7$) подключены к адресным входам ПЗУ.

Глава 7. УЗЛЫ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

7.1. ШИННЫЙ ФОРМИРОВАТЕЛЬ KP580BA86

Микросхема KP580BA86 (рис. 7.1)—двунаправленный 8-разрядный шинный формирователь (ШФ), предназначенный для обмена информацией между МП и системной шиной, обладает повышенной нагрузочной способностью и тремя состояниями. Управление ШФ осуществляется по входам $\overline{BK}$ и T в соответствии с табл. 7.1 [5].

Таблица 7.1. Управление операциями в ШФ

Сигналы управления		Направление передачи информации
$\overline{BK}$	T	
0	0	$A \leftarrow B$
0	1	$A \rightarrow B$
1	Φ	Выходы A и B в состоянии с высоким выходным сопротивлением (в третьем состоянии)

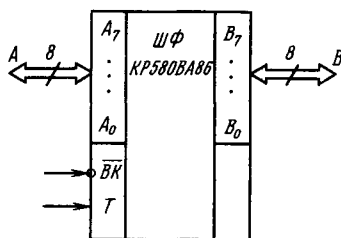


Рис. 7.1. Условное обозначение шинного формирователя (драйвера) KP580BA86

Выходы B имеют большую нагрузочную способность, чем выходы A, и допускают включение нагрузки, потребляющей ток 32 мА при низком уровне сигнала и -5 мА—при высоком. Поэтому ШФ обычно включается стороной A к выводам МП, а стороной B—к шинам микропроцессорной системы.

ЗАДАЧИ И УПРАЖНЕНИЯ

7.1. Каково назначение ШФ КР580ВА86? Можно ли использовать ШФ в качестве порта ввода или вывода?

7.2. ШФ КР580ВА86 используется в качестве буфера 8-разрядной шины данных МП КР580ВМ80. Представить соответствующую схему и указать, какие сигналы МП следует использовать для управления шинным формирователем.

7.3. Два ШФ КР580ВА86 необходимо использовать в качестве буфера шины адреса МП КР580ВМ80. Нарисовать схему подключения ШФ к МП.

7.4. Три ШФ типа КР580ВА86 подключены шинными выводами V^1 , V^2 , V^3 к системной 8-разрядной шине. Нарисовать эту схему и указать, какие управляющие сигналы на ШФ нужно подавать, чтобы обеспечить выполнение следующих операций:

а) $A^1 \rightarrow A^2$; ШФ₃ отключен; б) $A^2 \rightarrow A^3$; ШФ₁ отключен; в) $A^3 \rightarrow A^1$, A^2 ; г) A^1 , $A^2 \rightarrow A^3$.

7.5. Четыре ШФ КР580ВА86 подключены к 8-разрядной системной шине следующим образом: ШФ₁ и ШФ₂ — выводами V^1 и V^2 ; ШФ₃ и ШФ₄ — выводами A^3 и A^4 . Нарисовать эту схему и указать, какие управляющие сигналы на ШФ следует подавать, чтобы обеспечить выполнение следующих операций:

а) $A^1 \rightarrow V^3$, V^4 ; ШФ₂ отключен; б) $A^2 \rightarrow A^1$, V^3 ; ШФ₄ отключен;

в) $V^3 \rightarrow A^1$, A^2 , V^4 ; г) $V^4 \rightarrow A^1$; ШФ₂ и ШФ₃ отключены;

д) A^1 , $A^2 \rightarrow V^3$; ШФ₄ отключен.

7.2. МНОГОРЕЖИМНЫЙ БУФЕРНЫЙ РЕГИСТР К589ИР12

Многорежимный буферный регистр (МБР) предназначен для подключения к МПС периферийных устройств. В МБР (см. рис. 7.2) возможны режимы работы, при которых входы и выходы независимо друг от друга отключаются от соответствующих шин. Управление работой МБР осуществляется в соответствии с табл. 7.2, где используются следующие обозначения:

ВР — вход выбора режима (английское обозначение — MD), С — вход стробирования (EW), $\overline{BK}_1$ и BK_2 ($\overline{CS}_1$ и CS_2) — входы выбора кристалла. Обозначение «Откл» в табл. 7.2 означает для входов МБР отключение от входной шины данных, для выходов МБР «третье» состояние с высоким выходным сопротивлением и отключение регистра от выходной шины данных.

Сигнал 0 на выходе $\overline{ЗПр}$ (запрос прерывания) формируется при выборе МБР, т. е. при $\overline{BK}_1 \cdot BK_2 = 1$, либо при подаче строба на вход С. Сигнал $\overline{Сбр} = 0$ устанавливает все триггеры МБР в нулевое состояние. Подробное описание работы МБР приведено в [5, 6, 11].

Таблица 7.2. Управление операциями в МБР

Сигналы управления			Состояние	
С	ВР	$\overline{ВК}_1 \cdot ВК_2$	Входы	Выходы
0	0	0	Откл	Откл
0	0	1	Откл	Вкл
0	1	0	Откл	Вкл
0	1	1	Вкл	Вкл
1	0	0	Вкл	Откл
1	0	1	Вкл	Вкл
1	1	0	Откл	Вкл
1	1	1	Вкл	Вкл

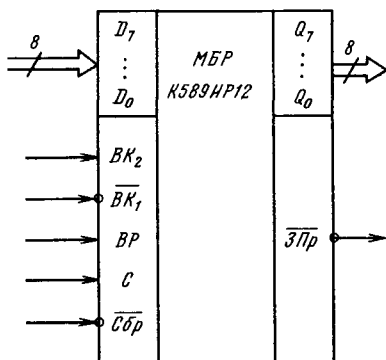


Рис. 7.2. Условное обозначение многорежимного буферного регистра К589ИР12

ЗАДАЧИ И УПРАЖНЕНИЯ

7.6. Каково назначение МБР и какие режимы в нем реализуются?

7.7. При каких управляющих сигналах в МБР реализуются следующие режимы:

а) стробируемый по входам $\overline{ВК}_1$, $ВК_2$ ввод информации в МБР при постоянно включенных входах D ;

б) стробируемый по входам $\overline{ВК}_1$, $ВК_2$ вывод информации из МБР при постоянно включенных выходах Q ;

в) стробируемый по входу $С$ вывод информации из МБР при постоянно включенных выходах Q ?

7.8. Используя МБР К589ИР12, составить принципиальную схему буфера шины адреса КР580ВМ80. Указать, какие сигналы нужно использовать для управления буфером.

7.9. Используя многорежимные буферные регистры К589ИР12, составить принципиальную схему устройства двусторонней передачи информации, включаемого между 8-разрядными шинами данных ШД¹ и ШД². Укажите необходимые сигналы управления.

7.10. Сравните возможности МБР К589ИР12 и ШФ КР580ВА86 при их использовании в качестве буферных устройств.

7.11. Нарисовать схему включения МБР в качестве порта ввода. Данные из внешнего устройства ВУ должны вводиться в МБР и фиксироваться в нем по сигналу от ВУ, а вводиться из МБР в ШД МПС по сигналу ввода $\overline{ВВ}$, поступающему из ШУ МПС (см. рис. 8.2). (Предполагается, что МПС имеет и другие порты.)

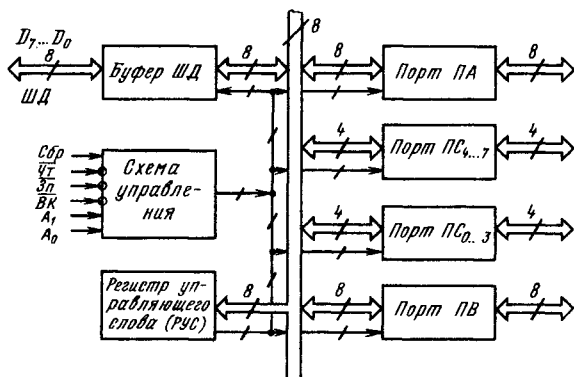


Рис. 7.3. Схема программируемого периферийного адаптера (интерфейса) KP580BV55

7.12. Составить схему включения МБР в качестве порта вывода. Выходы МБР должны быть постоянно подключены к ВУ; информация из ШД МПС должна фиксироваться в МБР и поступать в ВУ по сигналу вывода $\overline{\text{Выв}}$, формируемому в МПС (см. рис. 8.2). (Предполагается, что МПС имеет и другие порты.)

7.13. К МПС (см. рис. 8.2) необходимо подключить 16 внешних устройств: 10 должны работать на ввод информации в МП, 6 — на вывод из МП. Подключение необходимо выполнить с помощью МБР K589IP12, причем порты ввода имеют номера 0, 1, ..., 9, а порты вывода — номера 10, 11, ..., 15. Составить соответствующую схему.

7.3. ПРОГРАММИРУЕМЫЙ ПЕРИФЕРИЙНЫЙ АДАПТЕР KP580BV55

Программируемый периферийный (параллельный) адаптер (ППА) (рис. 7.3) при использовании в качестве интерфейсной БИС позволяет подключать к МПС различное периферийное оборудование без дополнительных логических схем. Инициализация и управление работой ППА осуществляются с использованием шести управляющих и адресующих сигналов, подаваемых из МПС по линиям управления, 8-разрядного управляющего слова режима, подаваемого от МП через шину данных (рис. 7.4), а также ряда сигналов, поступающих из внешних устройств (табл. 7.3)¹.

Описание работы ППА приведено в [5, 6, 11].

¹ Здесь приводятся не все возможные режимы работы ППА. Английские обозначения выводов ППА: $\overline{\text{Чт}}$ — $\overline{\text{RD}}$, $\overline{\text{Зп}}$ — $\overline{\text{WR}}$, $\overline{\text{ВК}}$ — $\overline{\text{CS}}$, Сбр — SR .

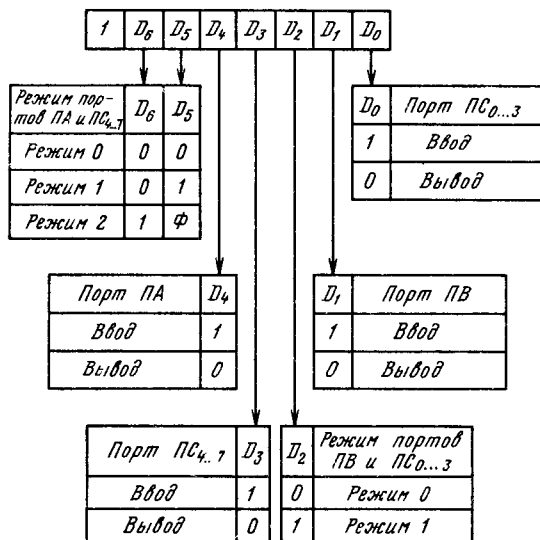


Рис. 7.4. Формат управляющего слова ППА

Таблица 7.3. Управление операциями и направлением передачи информации в ППА

Операция	Сигналы управления и адресации					Направление передачи информации
	$\overline{Чт}$	$\overline{Зп}$	A_1	A_0	$\overline{ВК}$	
ШД						
Чтение (ввод) данных	0	1	0	0	0	ШД←ПА
	0	1	0	1	0	ШД←ПВ
	0	1	1	0	0	ШД←ПС
Запись (вывод) данных	1	0	0	0	0	ШД→ПА
	1	0	0	1	0	ШД→ПВ
	1	0	1	0	0	ШД→ПС
Запись управляющего слова	1	0	1	1	0	ШД→РУС ППА
Отключение	Ф	Ф	Ф	Ф	1	Буфер ШД и порты в третьем состоянии

ЗАДАЧИ И УПРАЖНЕНИЯ

7.14. Какие режимы работы реализуются в ППА КР580ВВ55?

7.15. Каково назначение управляющего слова в ППА?

7.16. Каковы преимущества ППА по сравнению с ШФ КР580ВА86 и МБР К589ИР12 при их использовании в качестве буферных устройств?

7.17. Каково назначение входов A_1 и A_0 в ППА?

7.18. При каких управляющих сигналах на входах ППА $\overline{CT}$, $\overline{3П}$: A_1 , A_0 , $\overline{ВК}$ обеспечиваются:

- а) запись в РУС ППА управляющего слова,
- б) запись в порт ПА данных,
- в) чтение данных из порта ПВ?

7.19. При каком управляющем слове $D_7...D_0$ обеспечивается работа ППА в режиме простого ввода-вывода (режим 0), причем:

- а) все порты работают на ввод данных,
- б) все порты работают на вывод данных,
- в) порты ПА и $ПС_{4...7}$ работают на вывод данных, а порты ПВ и $ПС_{0...3}$ — на ввод данных?

7.20. Сколько вариантов режима 0 имеется в ППА?

7.21. МПС должна содержать: а) 18 8-разрядных портов; б) 48 8-разрядных портов. Используя ППА, составьте схему их подключения к МП КР580ВМ80. (Считать, что других интерфейсных устройств ввода-вывода в МПС нет.) Выбрать также адреса ППА и портов.

7.22. а) Запрограммировать ППА для работы в режиме простого ввода-вывода (режим 0) так, чтобы порты ПА и ПВ работали на ввод данных из внешних устройств, а порт ПС — на вывод;

б) Определить сигналы управления и адресации, обеспечивающие выполнение следующих операций: ввод данных в МП из порта ПА, вывод данных из МП в порт ПС, запись управляющего слова в регистр управляющего слова (РУС) ППА.

7.23. Составить программу для МПС (см. рис. 8.1 и 8.2), к которой подключен ППА (см. рис. 7.3), обеспечивающую передачу данных из порта ПА ППА через МП в порт ПС ППА (см. задачу 7.21).

7.24. МПС на базе МП КР580ВМ80 содержит четыре ППА КР580ВВ55. Нарисовать схему подключения ППА к МПС, используя для управления ППА по входам $\overline{ВК}$ линии $A_4...A_7$ шины адреса МПС. Запрограммировать ППА согласно табл. 7.4.

Таблица 7.4

ППА	Режим	Порты			
		ПА	$ПС_{4...7}$	$ПС_{0...3}$	ПВ
ППА ₁	0	Ввод	Вывод	Ввод	Вывод
ППА ₂	0	Ввод	Вывод	Ввод	Ввод
ППА ₃	0	Вывод	Ввод	Вывод	Ввод
ППА ₄	0	Вывод	Вывод	Ввод	Ввод

7.25. К МПС (см. рис. 8.2) необходимо подключить 16 ППА, причем ППА № 14 должен быть запрограммирован так, чтобы порт $PC_{0...3}$ работал на ввод, а порты ПА, ПВ и $PC_{4...7}$ — на вывод. Нарисовать схему подключения. Указать, какими командами МП обеспечивается программирование указанного ППА и его работа.

7.26. Запрограммировать интерфейс и составить программы для МПС на базе МП КР580ВМ80 (см. рис. 8.2), обеспечивающие решение следующих задач¹:

а) число N поступает через порт ПС ППА в МП, где сравнивается с константой k . При $N \neq k$ в порт ПА ППА выводится сигнал 0, а при $N = k$ — сигнал 1;

б) число N поступает через порт ПА ППА в МП, где обрабатывается по подпрограмме с меткой MARK. Результат выводится через порт ПВ ППА.

в) через порты ПВ и ПС ППА последовательно вводятся в МП числа N_1 и N_2 ; большее из них выводится через порт ПА ППА;

г) периодически один раз за 128 тактов МП производится опрос порта ПС ППА и имеющаяся в нем информация суммируется с накоплением. Когда сумма достигает заданного значения k в пределах $2^5 \dots 2^{15}$, через порт $PC_{4...7}$ ППА выводится сигнал 1 и суммирование вновь начинается от 0;

д) через порт ПА ППА в МП поступает двоичный код числа N , старший разряд которого равен 0. Если код содержит четное число единиц, то он без изменений выводится через порт ПВ ППА; если число единиц нечетное, то в старший разряд ставится единица и результат также выводится через порт ПВ ППА;

е) через порт ПА ППА в МП поступает двоичный код числа N . Если N является четным числом, то через порт ПВ ППА выводится сигнал 0; в противном случае выводится сигнал 1;

ж) в порт ПА ППА из внешнего устройства в заранее неизвестный момент времени поступает 8-разрядное двоичное число N , которое должно обрабатываться в МП по заданной программе PROG. Сигнал готовности ВУ к выдаче информации «1» поступает по линии C_3 порта ПС ППА. Результат вычисления из аккумулятора МП выводится в ВУ в синхронном режиме через порт ПВ ППА;

з) задача аналогична задаче ж), но вывод результата также осуществляется в асинхронном режиме, причем сигнал готовности ВУ «1» к приему информации поступает по линии C_4 порта ПС ППА.

¹ В задачах а) — е) в МПС реализуется синхронный ввод и вывод информации, в задаче ж) — асинхронный ввод и синхронный вывод, в задаче з) — асинхронный ввод и вывод.

7.4. ПРОГРАММИРУЕМЫЙ СВЯЗНОЙ АДАПТЕР KP580BB51

Программируемый связной адаптер (ПСА) представляет собой универсальное приемно-передающее устройство, которое преобразует снимаемую с ШД МПС информацию из параллельной формы в последовательную, пригодную для передачи в линию связи, а также осуществляет обратное преобразование (рис. 7.5).

Для инициализации и управления ПСА необходимы, во-первых, сигналы, определяющие направление передачи и вид передаваемой информации (табл. 7.5), во-вторых, два управляющих слова, вводимые в ПСА по ШД: инструкция режима (рис. 7.6) и инструкция команды; в-третьих, для реализации синхронного режима необходимо вывести в ПСА кодовые комбинации,

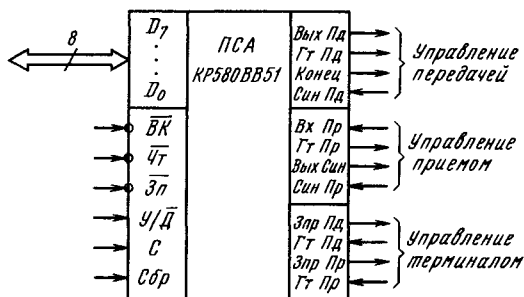


Рис. 7.5. Условное обозначение программируемого связного адаптера KP580BB51



Рис. 7.6. Формат инструкции режима ПСА

выбранные для синхронизирующих слов; для асинхронного режима необходимо перед выдачей в ПСА очередного байта данных ввести в МП из ПСА слово состояния [5, 6, 8].

Таблица 7.5. Управление операциями в ПСА

Сигналы управления				Направление передачи информации	Вид передаваемой информации
У/Д	Чт	Зп	ВК		
0	0	1	0	ШД←ПСА	Данные
0	1	0	0	ШД→ПСА	Данные
1	0	1	0	ШД←ПСА	Слово состояния ПСА
1	1	0	0	ШД→ПСА	Управляющие слова

Примечание. Английские обозначения выводов У/Д—CO/D, Чт—RD, Зп—WR, ВК—CS

ЗАДАЧИ И УПРАЖНЕНИЯ

7.27. Описать для ПСА асинхронный и синхронный режимы вывода. При каком режиме обеспечивается большая скорость?

7.28. Какие характеристики вводимой и выводимой информации определяет инструкция режима ПСА?

7.29. Составить инструкцию режима ПСА для асинхронного режима ввода-вывода со скоростью, равной 1/64 от тактовой частоты сигнала синхронизации, длиной слова данных 6 бит, контролем на четность и длительностью стоповой посылки 2 бита. Указать также формат слова для данного примера.

7.30. МПС на базе МП КР580ВМ80 работает на линию связи через ПСА КР580ВВ51. Управляющие входы Чт, Зп, У/Д и ВК ПСА подключены соответственно к линиям Вв, Выв шины управления и линиям А₀, А₁ шины адреса (см. рис. 8.2). Составить программу, обеспечивающую асинхронный вывод информации из МПС с характеристиками, указанными в задаче 7.29. Информация для вывода представляет собой блок из 16 байт, хранящихся в ячейках памяти МПС, начиная с ячейки ADR1.

Указание. После инструкции режима в ПСА необходимо послать инструкцию команды, задающую операцию вывода. В данном примере принять для нее код D₇...D₀=00100001₍₂₎=21₍₁₆₎ (D₀=1—вывод, D₂=0—нет ввода, D₅=1—запрос готовности модема линии связи на прием информации). Для инструкции команды окончания вывода использовать код 00001000₍₂₎=08₍₁₆₎ (D₀=0—нет вывода, D₃=1—конец передачи).

7.31. Для МПС из задачи 7.30 составить программу, обеспечивающую асинхронный ввод и обработку блока информации с характеристиками, указанными в задачах 7.29, 7.30. Каждое принятое слово необходимо обработать в МПС по подпрограмме PROG, хранящейся в ПЗУ, после чего результат, имеющий формат одного байта, записать в области ОЗУ, начиная с ячейки ADR1.

Указание. В данной задаче принять для инструкции команды ввода $D_7...D_0$ код $00000110_{(2)} = 06_{(16)}$ ($D_0=0$ — нет вывода, $D_1=1$ — запрос готовности модема линии связи на ввод данных, $D_2=1$ — ввод), а для инструкции команды окончания ввода — код $00001000_{(2)} = 08_{(16)}$ ($D_2=0$ — нет ввода, $D_3=1$ — конец передачи). Инструкцию режима см. в задаче 7.29.

7.5. ПРОГРАММИРУЕМЫЙ ИНТЕРВАЛЬНЫЙ ТАЙМЕР KP580BI53

Программируемый интервальный таймер (ПИТ) предназначен для реализации различных времязадающих функций. Инициализация и управление работой ПИТ осуществляются с использованием пяти управляющих сигналов, подаваемых от МП по линиям управления (рис. 7.7, табл. 7.6), трех 8-разрядных управляющих слов, подаваемых от МП через шину данных (рис. 7.8), а также трех входных и трех разрешающих сигналов, поступающих от внешних устройств. Подробное описание работы ПИТ имеется в [5, 6, 11].

Таблица 7.6. Управление операциями и направлением передачи данных в ПИТ

Операция	Сигналы управления и адресации					Направление передачи данных
	$\overline{ЧТ}$	$\overline{Зп}$	A_1	A_0	$\overline{ВК}$	
Чтение данных	0	1	0	0	0	ШД ← Сч 0
	0	1	0	1	0	ШД ← Сч 1
	0	1	1	0	0	ШД ← Сч 2
Запись данных	1	0	0	0	0	ШД → Сч 0
	1	0	0	1	0	ШД → Сч 1
	1	0	1	0	0	ШД → Сч 2

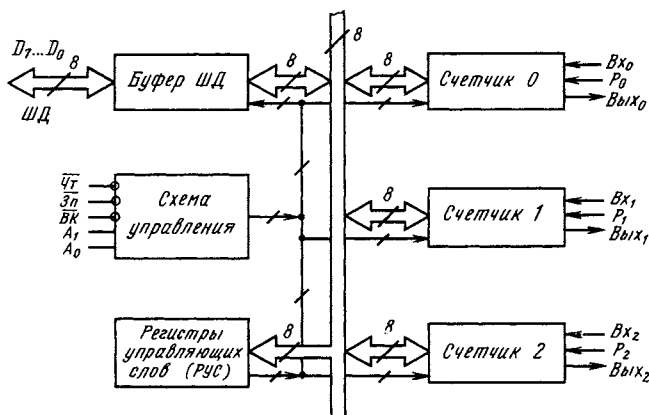


Рис. 7.7. Схема программируемого интервального таймера KP580BI53

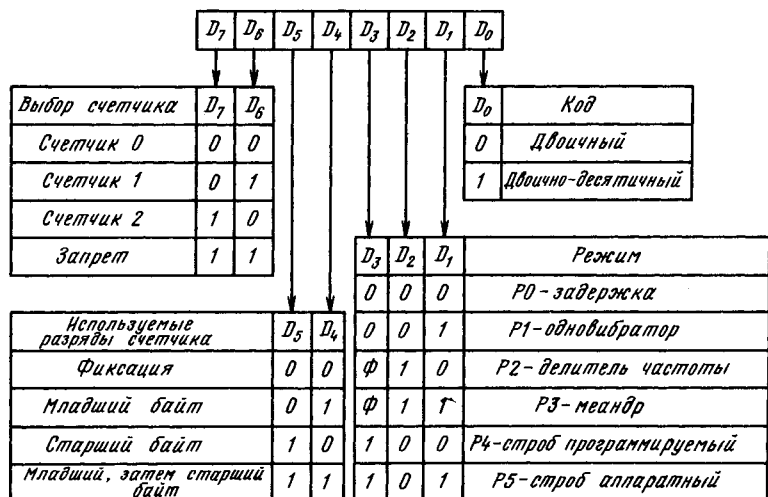


Рис. 7.8. Формат управляющего слова ПИТ

Окончание табл. 7.6

Операция	Сигналы управления и адресации					Направление передачи данных
	$\overline{ЧТ}$	$\overline{Зп}$	A_1	A_0	$\overline{ВК}$	
Запись управляющего слова	1	0	1	1	0	ШД→РУС
Отключение	Ф	Ф	Ф	Ф	1	Буфер ШД в третьем состоянии

Примечание. Английские обозначения выводов: $\overline{ЧТ}$ —RD, $\overline{Зп}$ —WR, $\overline{ВК}$ —CS.

ЗАДАЧИ И УПРАЖНЕНИЯ

7.32. Перечислить основные функции ПИТ КР580ВИ53.

7.33. Каково назначение буфера шины данных и других узлов ПИТ (см. рис. 7.7)?

7.34. Каково назначение управляющих слов в ПИТ?

7.35. Каково назначение управляющих сигналов $\overline{Зп}$, $\overline{ЧТ}$, A_1 , A_0 , $\overline{ВК}$ в ПИТ?

7.36. Каково назначение сигналов P_0 , P_1 , P_2 в ПИТ?

7.37. При каких управляющих сигналах $\overline{ЧТ}$, $\overline{Зп}$, A_1 , A_0 , $\overline{ВК}$ обеспечиваются:

а) загрузка в ПИТ управляющего слова из ШД;

б) загрузка в счетчик 2 ПИТ информационного байта (начального состояния) из ШД;

в) считывание показаний счетчика 2 ПИТ в ШД?

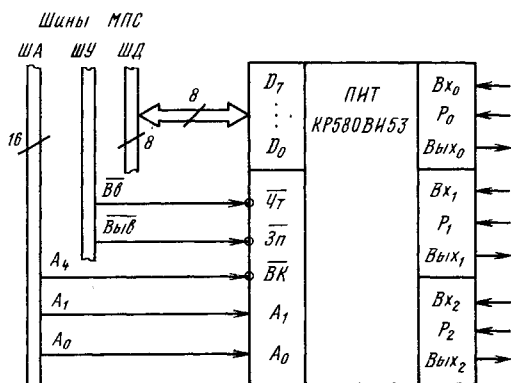


Рис. 7.9. Схема подключения ПИТ КР580ВН53 к МПС

7.38. Составить управляющие слова $D_7 \dots D_0$ для следующих режимов работы ПИТ:

а) счетчик 0 работает в режиме делителя частоты; используется двоичный код; коэффициент деления частоты $N < 255$;

б) счетчик 1 работает в качестве формирователя задержки; используется двоичный код; длительность задержки $t_3 = 1000 T_0$, где T_0 — период тактовых импульсов;

в) счетчик 2 работает в режиме одновибратора; используется двоичный код; длительность формируемого импульса $t_n = 100 T_0$.

7.39. Составить управляющее слово $D_7 \dots D_0$ для ПИТ, обеспечивающее работу счетчика 0 в двоичном коде и в режиме деления частоты (режим P2), причем коэффициент деления частоты N может задаваться программно в пределах 1..255. Построить временные диаграммы для $N=4$.

7.40. Схема подключения ПИТ к МПС (см. рис. 8.2) показана на рис. 7.9. Запрограммировать счетчик 2 ПИТ для работы в режиме одновибратора (режим P1). Длительность формируемого импульса $t_n = N T_0$, где T_0 — период тактовых импульсов.

а) Составить необходимое управляющее слово $D_7 \dots D_0$.

б) Определить сигналы управления и адресации $\overline{C\tau}$, $\overline{3\pi}$, A_1 , A_0 , $\overline{B\kappa}$, которые должны подаваться на ПИТ при загрузке в него управляющего слова $D_7 \dots D_0$.

в) Определить сигналы управления и адресации, которые должны подаваться на ПИТ при загрузке в него константы $N=6$.

г) Составить программу для МПС, обеспечивающую решение поставленной задачи.

д) Построить временные диаграммы, иллюстрирующие работу МПС.

7.41. МПС на базе МП КР580ВМ80 содержит системный контроллер СК КР580ВК28, ППА КР580ВВ55

и ПИТ К580ВИ53 (см. рис. 8.2). Нарисовать схему подключения ППА и ПИТ к МПС и запрограммировать интерфейс для выполнения следующих операций:

ППА должен работать в режиме 0, порты ПА и ПВ работают на ввод, порт ПС—на вывод;

в ПИТ счетчик 0 работает в режиме одновибратора и формирует импульс длительностью $t_{\text{и}}=64 T_0$, а счетчики 1 и 2 работают в режиме делителей частоты с коэффициентами деления частоты соответственно $N_1=16$ и $N_2=32$.

Указание. Для управления ППА и ПИТ по входам $\overline{\text{ВК}}$ использовать разряды A_4, A_5 шины адреса МПС, по входам $\overline{\text{ЧТ}}$ и $\overline{\text{Зп}}$ —выходные сигналы $\overline{\text{Вв}}$ и $\overline{\text{Выв}}$ системного контроллера СК (см. рис. 8.2).

ОТВЕТЫ И РЕШЕНИЯ

Р7.1. Шинные формирователи используются чаще всего для увеличения нагрузочной способности микросхем. Предусмотрена возможность двунаправленной передачи информации и установки входов и выходов ШФ в третье состояние. Благодаря этому к одной шине можно подсоединить несколько ШФ. Однако ШФ не имеет внутренней памяти, поэтому использовать его в качестве порта ввода или вывода (если информация в порту должна фиксироваться) нельзя.

Р7.2. На вход ШФ ВК следует подать сигнал с выхода МП «Подтверждение захвата» (ПЗх), а на вход Т—через инвертор—сигнал с выхода «Прием» (Пм). Если режим захвата (прямого доступа к памяти) в МПС не используется, на вход ШФ ВК следует подать постоянный уровень 0.

Р7.3. На входы ШФ ВК следует подать сигнал с выхода МП ПЗх. На входы Т следует подать постоянный уровень 1 (схема показана на рис. 8.2). Если режим захвата в МПС не используется, на входы ШФ ВК следует подать постоянный уровень 0.

Р7.4.

Таблица Р7.1

Операция	Управляющие сигналы					
	$\overline{\text{ВК}}_1$	T_1	$\overline{\text{ВК}}_2$	T_2	$\overline{\text{ВК}}_3$	T_3
а	0	1	0	0	1	Ф
б	1	Ф	0	1	0	0
в	0	0	0	0	0	1

Операция г) невозможна, так как запараллеливание включенных ШФ по выходам недопустимо.

P7.5

Таблица P7.2

Операция	Управляющие сигналы							
	$\overline{BK}_1$	T_1	$\overline{BK}_2$	T_2	$\overline{BK}_3$	T_3	$\overline{BK}_4$	T_4
а	0	1	1	Ф	0	1	0	1
б	0	0	0	1	0	1	1	Ф
в	0	0	0	0	0	0	0	1
г	0	0	1	Ф	1	Ф	0	0

Операция д) невозможна

P7.6. В МБР реализуются режимы ввода, вывода и хранения информации, причем входы и выходы могут отключаться (переводиться в третье состояние) независимо друг от друга. Основные применения МБР—в качестве буферной памяти и портов ввода-вывода.

P7.7. а) $C=1$, $BP=0$, $\overline{BK}_1 \cdot \overline{BK}_2 = 0 \rightarrow 1 \rightarrow 0$; б) $C=1$, $BP=1$, $\overline{BK}_1 \cdot \overline{BK}_2 = 0 \rightarrow 1 \rightarrow 0$; в) $BP=0$, $\overline{BK}_1 \cdot \overline{BK}_2 = 0$, $C=0 \rightarrow 1 \rightarrow 0$.

P7.8. Поскольку шина адреса ША содержит 16 линий, а МБР K589IP12—8-разрядный, необходимо использовать два МБР. Выходы $A_0 \dots A_{15}$ ША МП подключаются к входам $D_0 \dots D_7$ двух МБР, выходы $Q_0 \dots Q_7$ обоих МБР подключаются к ША МПС. Для управления МБР используются входы $\overline{BK}_1$, на которые подается сигнал «Подтверждение захвата» (ПЗх) с выхода МП. На остальных входах МБР $\overline{BK}_2 = C=1$, $BP=0$. В нормальном режиме работы МП сигнал $ПЗх=0$, и усиленные по мощности сигналы с адресной шины МП поступают в адресную шину МПС. В режиме захвата (прямого доступа к памяти) $ПЗх=1$, выходы МБР отключаются и МП отсоединяется от адресной шины.

P7.9. Необходимо использовать два МБР, причем шину $ШД^1$ подключить ко входам МБР¹ и выходам МБР², а шину $ШД^2$ —ко входам МБР² и выходам МБР¹. Для управления можно использовать, например, входы $\overline{BK}_1^1$ и $\overline{BK}_2^2$, включив их параллельно. При $BP^1 = BP^2 = 0$, $C^1 = C^2 = 1$, $\overline{BK}_2^1 = 1$, $\overline{BK}_1^2 = 0$ выходы обоих МБР постоянно открыты. При $\overline{BK}_1^1 = \overline{BK}_2^2 = 0$ открываются входы МБР¹ и информация передается через МБР¹ из $ШД^1$ в $ШД^2$; при $\overline{BK}_1^1 = \overline{BK}_2^2 = 1$ открываются входы МБР², и информация передается через МБР² из $ШД^2$ в $ШД^1$.

P7.10. МБР имеет внутреннюю память и может использоваться в качестве порта ввода или вывода, но не позволяет менять направление передачи информации, тогда как ШФ внутренней памяти не имеет, в качестве порта не

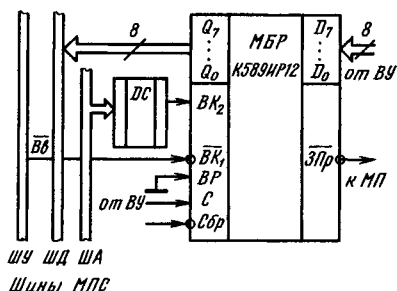


Рис. Р7.1. Схема подключения МБР к МПС в качестве порта ввода

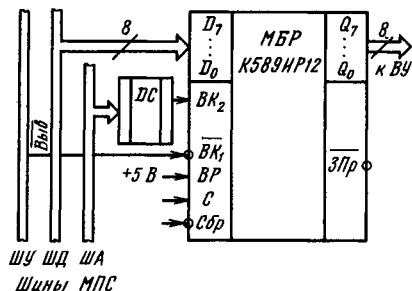


Рис. Р7.2. Схема подключения МБР к МПС в качестве порта вывода

используется, но зато может использоваться в качестве двунаправленного буфера.

Р7.11. Схема подключения МБР в качестве порта ввода показана на рис. Р7.1. При поступлении от ВУ сигнала $С$ входы МБР открываются, в нем фиксируется информация из ВУ и формируется сигнал $\underline{3Пр}$, который подается на МП. После получения сигналов $Вв = ВК_1 = 0$ и $ВК_2 = 1$, которые формируются в МПС при выполнении команды IN (см. рис. 8.2), открываются выходы МБР, информация из МБР поступает в ШД МПС. (Сигнал $ВК_2 = 1$ формируется дешифратором $ДС$ из кода адреса (номера) порта ввода, указанного во втором байте команды IN .)

Р7.12. Схема подключения МБР к МПС в качестве порта вывода показана на рис. Р7.2. При $ВР = 1$ выходы МБР постоянно открыты. При выполнении команды OUT в МПС формируются сигналы $Выв = 0$, $ВК_2 = 1$ и открываются входы МБР; информация из ШД поступает в МБР, фиксируется в нем и передается в ВУ. (Сигнал $ВК_2 = 1$ формируется дешифратором $ДС$ из кода адреса (номера) порта вывода, указанного во втором байте команды OUT .)

Р7.13. За основу можно взять схемы на рис. Р7.1 и Р7.2 использовав для выбора каждого из 16 МБР соответствующие выходные сигналы дешифратора $K155ИД3$.

Р7.14. ППА предназначен для организации обмена информацией между МП и периферийными устройствами в параллельном коде. В ППА реализуются режимы асинхронного и синхронного ввода-вывода, обмена в режиме прерываний и т. д.

Р7.15. Управляющее слово $D_7 \dots D_0$ поступает по шине данных МПС, запоминается в специальном регистре ППА и обеспечивает его настройку на различные режимы работы.

Р7.16. Основное преимущество ППА по сравнению с ШФ — наличие внутренней памяти, по сравнению с МБР — возможность

двунаправленной передачи информации. Кроме того, ППА допускает программную настройку на различные режимы ввода-вывода и различные варианты подключения внешних устройств.

Р7.17. Сигналы на входах A_1 , A_0 ППА определяют порт ПА, ПВ или ПС, который обменивается информацией с МП, или адресуют регистр управляющего слова, который получает информацию из МП (см. рис. 7.4). В качестве сигналов A_1 , A_0 часто используют младшие разряды шины адреса МПС.

Р7.18. а) 10110; б) 10000; в) 01010.

Р7.19. а) 10011011; б) 10000000; в) 10000011.

Р7.20. Поскольку в режиме 0 порты ПА, ПВ, $ПС_{0...3}$ и $ПС_{4...7}$ настраиваются на ввод-вывод независимо, всего имеется $2^4 = 16$ вариантов реализации режима 0.

Р7.21. а) Для адресации портов в МП КР580ВМ80 используются команды IN и OUT, в которых адреса задаются 8-разрядным двоичным кодом. Имея в виду, что каждый ППА содержит 3 порта ПА, ПВ и ПС, адреса которых можно задать двумя младшими разрядами шины адреса A_1 и A_0 , поставим в соответствие каждому из оставшихся шести разрядов свой ППА. Тогда адреса ППА будут следующие: 111110 A_1A_0 , 111101 A_1A_0 , ..., 011111 A_1A_0 . Для обеспечения такой организации ввода-вывода необходимо к входам ВК ППА₁...ППА₆ подсоединить соответствующие линии $A_2...A_7$ шины адреса МПС. Для схемы, показанной на рис. Р7.3а, адреса РУС и портов ППА будут следующие: РУС—11011111=DF, ПА—DC, ПС—DE, ПВ—DD;

б) для подключения к МПС 16 ППА необходимо использовать дешифратор на 4 входа с 16 инверсными выходами, например типа К155ИДЗ. Входы дешифратора следует подсоединить к линиям ША МПС, например $A_2...A_5$, а выходы—к входам ВК ППА. При этом адреса ППА будут следующие: ФФ0000 A_1A_0 , ФФ0001 A_1A_0 , ..., ФФ1111 A_1A_0 , а адреса портов в ППА задаются, как обычно, разрядами A_1A_0 (см. рис. Р7.3,б).

Р7.22. а) Управляющее слово 10010010=92H.

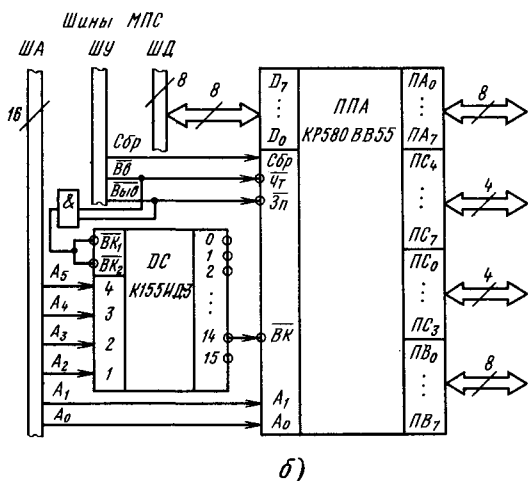
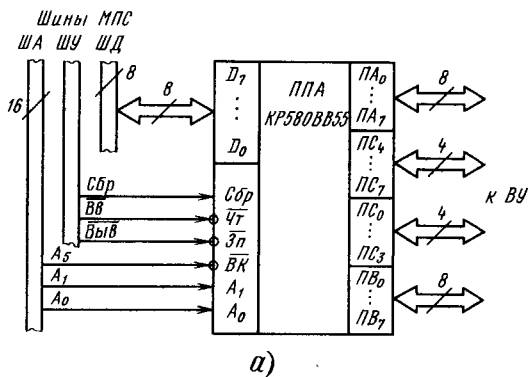
б)

$\overline{CT}$	$\overline{3n}$	$\overline{BK}$	A_1	A_0	Направление передачи информации
0	1	0	0	0	ШД←ПА
1	0	0	1	0	ШД→ПС
1	0	0	1	1	ШД→РУС ППА

Р7.23. Программа для схемы на рис. Р7.3, а:

Рис. Р7.3. Схема подключения ППА к МПС:

а) подключен один из шести ППА; б) подключен один из 16 ППА (№ 14)



TRANS: MVI A,92H

OUT 0DFH

IN 0DCH

;Загрузка в аккумулятор МП
;управляющего слова, обеспе-
;чивающего работу порта ПА
;на ввод и порта ПС на
;вывод
;Вывод из МП и запись
;в РУС ППА управляющего
;слова. Управляющие сиг-
;налы $\overline{ЧТ}=1$, $\overline{ЗП}=0$ и адресу-
;ющие сигналы $A_5=\overline{ВК}=0$,
; $A_1=A_2=1$ формируются в
;системном контроллере СК
;Ввод данных в МП из порта
;ПА ППА. Управляющие сиг-

OUT 0DEN

;налы $\overline{CT}=0$, $\overline{3п}=1$ и адресующие сигналы $A_5=\overline{BK}=0$,
 $A_1=A_0=0$ формируются
 ;в системном контроллере СК
 ;Вывод данных из МП в порт
 ;ПС ППА. Управляющие сигналы $\overline{CT}=1$, $\overline{3п}=0$ и адресующие сигналы $A_5=\overline{BK}=0$,
 $A_1=1$, $A_0=0$ формируются
 ;в системном контроллере СК

P7.24. См. решение задач 7.21 и 7.23.

P7.25. Вариант подключения ППА № 14 к схеме на рис. 8.2 показан на рис. P7.3, б (остальные ППА подключаются к соответствующим выводам дешифратора). Необходимые режимы работы портов ППА № 14 обеспечиваются управляющим словом $1000\,0001_{(2)}=81H$. Адресация ППА № 14 обеспечивается подаваемым по ША кодом $A_5A_4A_3A_2=1110_{(2)}=14$; адресация регистра управляющего слова — кодом $A_1A_0=11_{(2)}$. Управляющее слово выводится в РУС ППА № 14 по команде OUT FB ($FB=A_7A_6A_5A_4A_3A_2A_1A_0=1111\,1011$; два старших разряда A_7 и A_6 безразличны). Ввод информации через порт ПС_{0...3} осуществляется по команде IN FA ($FA=1111\,1010$; адресация РУС ППА та же, порта ПС — $A_1A_0=10$). Вывод информации через порт ПВ осуществляется по команде OUT F9 ($F9=1111\,1001$; адресация РУС ППА — та же, порта ПВ — $A_1A_0=01$). Итак, для программирования ППА № 14 на ввод через порт ПС_{0...3} и вывод через порт ПВ в схеме рис. P7.3, б необходимы следующие команды:

MVI A, 81; OUT FB; IN FA; OUT F9.

P7.26. а) — е). См. решение задачи P7.23.

ж), з). В программы, аналогичные P7.23, необходимо включить циклы из команд ввода (IN), маскирования (ANI) и проверки признаков готовности (JZ).

P7.27. В асинхронном режиме вывода передатчик ПСА сопровождает каждое передаваемое в модем линии связи слово стартовыми и стоповыми битами. Выдача этого слова на выход передатчика производится последовательно разряд за разрядом с тактовой частотой, кратной частоте сигнала синхронизации, поступающего от отдельного генератора. Очередное слово выдается из ПСА при поступлении из модема сигнала готовности к приему информации. Максимальная скорость передачи данных в асинхронном режиме 9 Кбод.

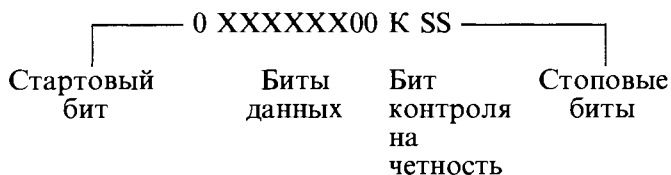
В синхронном режиме вывода передатчик ПСА сопровождает блок данных одним или двумя синхронизирующими словами, причем скорость передачи равна частоте синхронизации. Под-

тверждения готовности модема к приему информации при синхронном выводе не требуется. Максимальная скорость передачи данных в синхронном режиме 56 Кбод, т. е. гораздо выше, чем в асинхронном режиме.

P7.28. Инструкция режима ПСА, являясь одним из управляющих слов, засылаемых в ПСА из ШД МПС, определяет следующие характеристики выбранного режима: тип режима (синхронный или асинхронный), длину информационной части слова, наличие контроля на четность или нечетность. Кроме того, для асинхронного режима указывается скорость выдачи информации из ПСА (коэффициент деления частоты сигнала синхронизации) и длина стоповой посылки, а для синхронного режима — число синхрослов и вид синхронизации (внешняя или внутренняя).

P7.29. Инструкция режима $D_7 \dots D_0$ имеет вид 11110111. В 16-ричном коде — F7.

Формат слова:



Поскольку длина информационной части слова по заданию 6 бит, а в формате она должна быть равна 8 битам, добавлены два нулевых бита. Контрольный бит К определяется как сумма по модулю 2 всех битов данных.

P7.30. Программа асинхронного вывода:

OUTPUT:	MVI A, 0F7H	;Непосредственная загрузка ;в аккумулятор инструкции ;режима асинхронного вво- ;да-вывода
	OUT 01	;Засылка инструкции режи- ;ма в ПСА
	MVI A, 21H	;Загрузка в аккумулятор ин- ;струкции команды вывода
	OUT 01	;Засылка инструкции коман- ;ды вывода в ПСА
	LXI H, ADR1	;Загрузка в регистровую па- ;ру HL начального адреса ;блока данных
	MVI B, 10H	;Начальная установка счет- ;чика байтов, организо- ;ванного в PОНе В.

L1:	IN	01	;Чтение слова состояния ;ПСА
	ANI	01	;Выделение бита D ₀ слова ;состояния ПСА—сигнала ;готовности передатчика ;к выводу очередного байта ;данных
	JZ	L1	;Возврат к L1 при D ₀ =0 ;(передатчик не готов)
	MOV	A, M	;Пересылка в аккумулятор ;из косвенно адресуемой ;ячейки памяти очередного ;байта данных
	OUT	00	;Вывод байта данных ;в ПСА
	INX	H	;Вычисление адреса следу- ;ющей ячейки памяти
	DCR	B	;Декремент счетчика выво- ;димых байтов
	JNZ	L1	;Возврат к L1, если выведен ;не весь блок данных
	MVI	A, 08	;Загрузка в аккумулятор ин- ;струкции команды конца ;передачи
	OUT	01	;Засылка инструкции коман- ;ды в ПСА
	HLT		;Конец программы

Перед исполнением программы в ПСА подается из МПС сигнал сброса Сбр=1.

Р7.31. Программа асинхронного ввода и обработки:

INPUT:	MVI	A, 0F7H	;Непосредственная загрузка ;в аккумулятор инструкции ;режима асинхронного вво- ;да-вывода
	OUT	01	;Засылка инструкции режи- ;ма в ПСА
	MVI	A, 06	;Непосредственная загрузка ;в аккумулятор инструкции ;команды ввода
	OUT	01	;Засылка инструкции коман- ;ды ввода в ПСА
	LXI H,	ADR1	;Загрузка в регистровую па- ;ру HL начального адреса ;ячейки памяти
	MVI	B, 10H	;Начальная установка счет- ;чика вводимых байтов, ор-

K1:	IN	01	;ганизованного в регистре В
			;Чтение слова состояния
			;ПСА
	ANI	02	;Выделение бита D_1 слова
			;состояния — сигнала готов-
			;ности приемника ПСА
			;к вводу очередного байта
			;данных
	JZ	K1	;Возврат к K1 при $D_1 = 0$
			;(приемник не готов)
	IN	00	;Ввод слова данных из ПСА
			;в МП
	CALL	PROC	;Обработка принятого сло-
			;ва
	MOV	M, A	;Пересылка результата об-
			;работки в косвенно-адресу-
			;емую ячейку памяти
	INX	H	;Вычисление адреса следу-
			;ющей ячейки памяти
	DCR	B	;Декремент счетчика байтов
	JNZ	K1	;Возврат к K1, если введен
			;не весь блок информации
	MVI	A, 08	;Непосредственная загрузка
			;в аккумулятор инструкции
			;команды конца приема
	OUT	01	;Засылка инструкции коман-
			;ды в ПСА
	HLT		;Конец программы
PROC:			;Подпрограмма обработки
			;введенной информации
			;

Перед исполнением программы в ПСА из МПС подается сигнал сброса $Sbr = 1$.

Р7.32. ПИТ может использоваться для генерирования периодических импульсных сигналов с различной частотой следования, формирования сигналов различной длительности и задержанных во времени, для деления частоты, подсчета числа импульсов и т. д. Установка режима и характеристик ПИТ производится программным путем.

Р7.33. Буфер ШД ПИТ предназначен для приема управляющих слов и входных данных, поступающих из МП в ПИТ по шине данных МПС, а также передачи показаний счетчиков, поступающих из ПИТ в МП по той же шине.

Р7.34. Управляющее слово, поступающее по ШД МПС и запоминаемое в специальном регистре ПИТ, служит для программного управления режимом работы соответствующего счет-

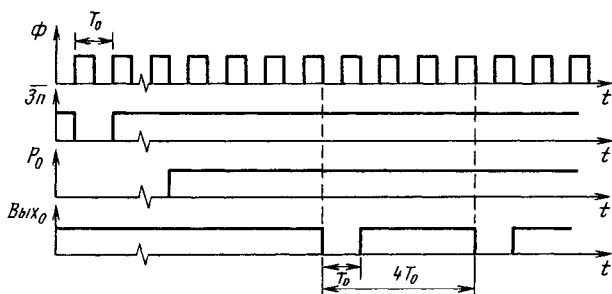


Рис. Р7.4. Временные диаграммы для таймера, используемого в качестве делителя частоты

чика ПИТ, т. е. позволяет настраивать его на работу в качестве формирователя задержек, одновибратора, делителя частоты и т. д. Управление осуществляется независимо по каждому счетчику.

Р7.35. Сигналы на входах $\bar{3}n$ и $\bar{Чт}$ ПИТ определяют направление передачи информации между буфером ШД и счетчиками (01—запись в счетчики, 10—чтение из счетчиков). Сигналы на входах A_1 и A_0 определяют счетчик или регистр управляющего слова, которые обмениваются информацией с МП по ШД (00—счетчик 0, 01—счетчик 1, 10—счетчик 2, 11—регистр управляющего слова). Сигнал $\bar{ВК}$ —сигнал выбора ПИТ (на счетчики сигнал $\bar{ВК}$ не действует). В качестве сигналов A_1, A_0 часто используются младшие разряды шины адреса МПС, а сигнал $\bar{ВК}$ формируется посредством дешифрации нескольких разрядов шины адреса МПС.

Р7.36. Сигналы на входах P_0, P_1, P_2 ПИТ разрешают работу счетчиков; они могут использоваться также для синхронизации счетчиков от внешних устройств ($f_{вх} \leq 2$ МГц).

Р7.37. а) 10110; б) 01100; в) 10100.

Р7.38. а) 00010100; б) 01110000; в) 10010010.

Р7.39. Управляющее слово $D_7 \dots D_0$ 00010100. Временные диаграммы показаны на рис. Р7.4.

По сигналу $\bar{3}n=0$ (при $\bar{Чт}=1, A_1=A_0=0, \bar{ВК}=0$) в счетчик 0 записывается число $N=4$. Импульсы на выводе ПИТ $Вых_0$ появляются после подачи сигнала $P_0=1$, имеют длительность $t_n=T_0$ и период следования $T_{вых}=4T_0$.

Р7.40. а) Используем в ПИТ счетчик 2; режим одновибратора обеспечивается управляющим словом $D_7 \dots D_0$ 10010010=92Н. (Поскольку по условию задачи $N=6 < 255$, в счетчике используется только младший байт.)

б) Для загрузки в ПИТ управляющего слова $D_7 \dots D_0$ необходимы сигналы $\bar{Чт}=1, \bar{3}n=0, A_1=1, A_0=1, \bar{ВК}=0$.

в) Для загрузки в Сч2 ПИТ начального состояния необходимы сигналы $\bar{Чт}=1, \bar{3}n=0, A_1=1, A_0=0, \bar{ВК}=0$.

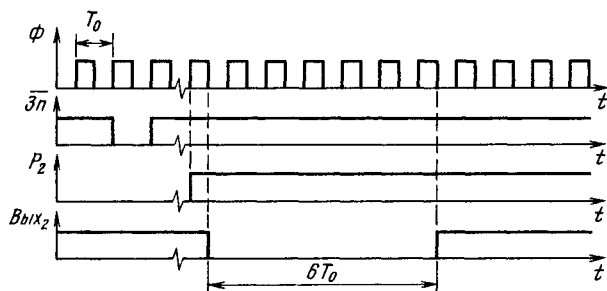


Рис. Р7.5. Временные диаграммы для таймера, используемого в качестве одновибратора

г) Программа для МПС:

PULSE: MVI A,92H

;Непосредственная загрузка
;в аккумулятор управляюще-
;го слова 10010010=92 H

OUT OEFH

;Вывод из МП и запись
;в РУС ПИТ управляющего
;слова 92H. Управляющие си-
;гналы $\overline{C_T}=1$, $3n=0$ форми-
;руются в системном кон-
;троллере СК при выполне-
;нии в МП команды OUT

MVI A,06H

;Непосредственная загрузка
;в аккумулятор числа 6

OUT 0EEN

;Вывод из МП и запись в Сч2
;ПИТ начального состояния
;N=6

д) Временные диаграммы представлены на рис. Р7.5. По сигналу $3n=0$ (при выполнении команды OUT 0EEN) в счетчик 2 ПИТ записывается число 6. После подачи сигнала $P_2=1$ на выходе ПИТ Вых₂ появляется импульс длительностью $t_n=6 T_0$.

Р7.41. См. решение задач 7.25 и 7.40.

Глава 8. МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА И СИСТЕМЫ НА БАЗЕ МП КР580ВМ80

Проектирование специализированного микропроцессорного устройства (МПУ) или системы (МПС) содержит обычно три этапа [6, 8, 17, 13]:

1) выбор элементов и составление структурной, функциональной и принципиальной схем;

2) распределение памяти и присвоение элементам, к которым в процессе выполнения программы необходимо обращаться, соответствующих кодов и адресов;

3) составление и отладка программы.

Основными исходными данными для проектирования являются: описание или схема алгоритма решаемой задачи, характеристики входных и выходных сигналов, требования к точности, быстродействию, потребляемой мощности, надежности. Часто в задании содержатся также рекомендации по применению определенной элементной базы. Во многих случаях указанные требования закладываются в схему устройства вначале лишь в общем виде (посредством выбора подходящих элементов), затем в процессе проектирования проверяется их выполнение и вносятся необходимые коррективы.

На первом этапе проектирования выбирается элементная база, и в первую очередь микропроцессорный комплект. Выбор определяет прежде всего специфика реализуемых в МПС алгоритмов.

Если МПС предназначается для обработки сигналов, определяющим обычно является быстродействие МП и его разрядность. До последнего времени преимущество в этом смысле имели разрядно-модульные МП с микропрограммным управлением (например, типа КР1804), но в настоящее время разработаны весьма совершенные специализированные однокристалльные МП для цифровой обработки сигналов типа ТМС 320, которым во многих случаях и следует отдать предпочтение.

Если МПС предназначена для реализации алгоритмов управления, важнейшим фактором при выборе МПК является чаще всего необходимость «удобного» сопряжения МПС с объектами управления. Поскольку в системах связи таковыми часто являются электромеханические устройства, обладающие большой инерционностью, к быстродействию МП здесь предъявляются сравнительно невысокие требования, и в таких системах во многих случаях можно использовать однокристалльные МП типа КР580, К1810 и другие или микроконтроллеры, например типа КР1816, которые позволяют строить достаточно простые и экономичные схемы.

Если МПС предназначена для построения информационно-справочной системы, определяющим требованием обычно является возможность подключения к МП ЗУ большой емкости.

Разумеется, гарантировать оптимальный выбор МПК на основании указанных весьма общих соображений нельзя и может потребоваться проработка нескольких вариантов с последующим выбором наилучшего исходя из таких требований, как экономичность, минимальная потребляемая мощность, тип корпуса и т. д. В некоторых случаях решающим при выборе МПК является возможность использования имеющегося программного обеспечения, поскольку трудоемкость разработки новых программ сравнима или даже выше трудоемкости разработки аппаратной части.

На первом этапе проектирования выбираются также элементы памяти ОЗУ и ПЗУ, имеющие требуемую емкость, быстродействие и потребляемую мощность, а также интерфейсные модули, обеспечивающие связь МП с внешними устройствами (ВУ). Если система «компактна», связь обеспечивается, как правило, с помощью параллельных периферийных адаптеров или многорежимных буферных регистров; если же ВУ разнесены на большие расстояния, связь

иногда организуется с помощью последовательных адаптеров; при этом линии связи упрощаются, но производительность системы падает. Для частичной разгрузки центрального процессорного элемента в систему иногда включают таймер, который берет на себя функции «вреязадающего» элемента, а также другие устройства.

Далее проводится проверка электрического сопряжения элементов схемы и при необходимости вводятся усилители, формирователи, преобразователи уровней и т. д. Затем составляются исходные варианты структурной, функциональной и принципиальной схем.

Выполнение второго этапа начинается с распределения адресного поля между ОЗУ, ПЗУ и ВУ. Если ЗУ имеет емкость 2^k слов (байтов), для адресации его ячеек используются k линий шины адреса (ША) МПС, обычно линии $A_0 \dots A_{k-1}$ (в МПС на базе КР580ВМ80 $k=16$). Выбор (селекция) ОЗУ или ПЗУ осуществляется с помощью одной из неиспользованных линий ША. (Например, $A_{15}=0$ может соответствовать ОЗУ, а $A_{15}=1$ — ПЗУ.)

Для адресации ВУ, подключаемых к МПС через порты, обычно используются младшие разряды ША. (В МПС на базе КР580ВМ80 — восемь разрядов.) Применение одних и тех же адресов и линий ША для ЗУ и ВУ возможно, поскольку обращение к этим устройствам осуществляется с помощью разных команд МП, т. е. для селекции ЗУ или ВУ используются генерируемые в МПС управляющие сигналы.

Если в качестве портов применяются многорежимные буферные регистры и их не более восьми, то каждому порту ставится в соответствие одна линия ША, по которой передается сигнал, выбирающий этот порт (иными словами, каждый порт кодируется унитарным кодом). Если портов в МПС более восьми (в МПС на базе МП КР580ВМ80 — 9 ... 256), они кодируются различными двоичными кодами и для выбора используется дешифратор.

Если в качестве портов применяются программируемые периферийные адаптеры (параллельные или последовательные), то возникает возможность программного изменения конфигурации системы, но максимальное число портов оказывается меньше 256, так как некоторая часть адресного поля отводится для адресации служебных регистров самих адаптеров (см. соответствующие задачи гл. 7).

Далее на втором этапе выделяются области ПЗУ для запоминания основной программы, подпрограмм (в том числе подпрограмм обработки прерываний), констант и т. д. В ОЗУ выделяются области для хранения поступающих из ВУ данных и промежуточных результатов, а также для организации стека; начало (вершина) стека часто располагается в последней ячейке ОЗУ. Затем специфицируются РОНЫ МП.

На третьем этапе составляется программа работы устройства. При этом, если решаемая задача простая (программа содержит до 100 команд), используется язык ассемблера выбранного МП, который позволяет непосредственно получить двоичные («объектные») коды команд, записываемые далее в ПЗУ. Если же задача достаточно сложная, то используется тот или иной язык высокого уровня; выбор языка определяется тем, какие имеются в распоряжении разработчика средства отладки программ и их трансляции в машинные коды выбранного МП. Система команд МП позволяет, как правило, выполнить ту или иную операцию алгоритма

множеством способов. Поэтому составленную программу, даже если она правильно решает поставленную задачу, необходимо тщательно проанализировать и по возможности оптимизировать с целью уменьшения емкости используемой памяти и времени выполнения.

В данном разделе при программировании используется язык ассемблера МП КР580ВМ80, но без псевдокоманд.

При отладке программ на ЭВМ, имеющих трансляторы с ассемблера, программы необходимо дополнить соответствующими псевдокомандами.

Следует отметить, что широкие возможности использования типовых решений на первом и втором этапах проектирования и ограниченные возможности такого рода на третьем этапе приводят к тому, что 60 ... 90% трудоемкости проектирования специализированного МПУ или МПС приходится на третий этап.

В данном разделе приводятся задачи на построение и использование различных устройств и систем, элементами которых являются микропроцессор КР580ВМ80, микросхемы памяти, шинные формирователи КР580ВА86, многорежимные буферные регистры К589ИР12, программируемые периферийные адаптеры КР580ВВ55, системный контроллер КР580ВК28, генератор тактовых импульсов КР580ГФ24, аналого-цифровые и цифро-аналоговые преобразователи и т. д. (Более простые задачи на построение и использование устройств и систем из подобных, а также иных микропроцессорных элементов приведены в гл. 6 и 7.)

Многие задачи данного раздела могут быть непосредственно использованы при составлении и отладке программ на микротренажере «Микролаб КР580ИК80». Процедура программирования интерфейса в задачах соответствует принятой в этой микроЭВМ.

ЗАДАЧИ И УПРАЖНЕНИЯ

8.1. На рис. 8.1 показана структурная схема МПС на базе микропроцессорного комплекса (МПК) КР580. Объяснить назначение основных устройств МПС: микропроцессора (МП),

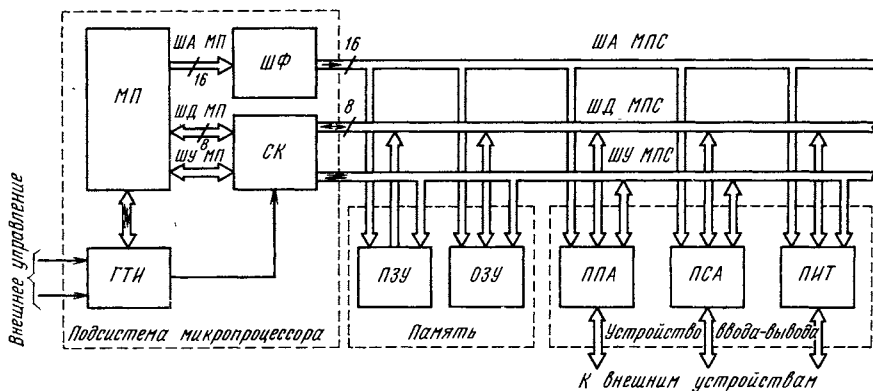


Рис. 8.1. Схема микропроцессорной системы на основе МП КР580

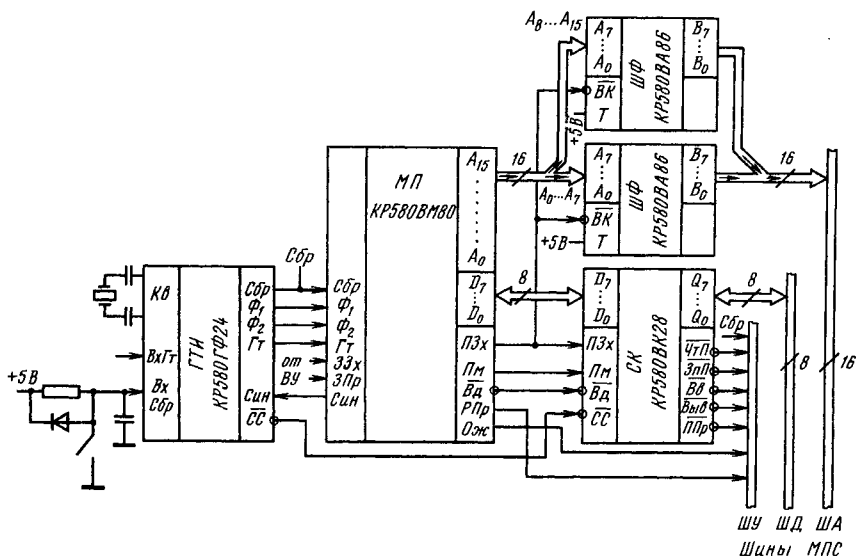


Рис. 8.2. Схема подсистемы МП КР580ВМ80

генератора тактовых импульсов (ГТИ), системного контроллера (СК), оперативного и постоянного запоминающих устройств (ОЗУ и ПЗУ), шин данных, адресов и управления (ШД, ША, ШУ), а также интерфейсных модулей — параллельного и связанного адаптеров (ППА и ПСА), программно-интервального таймера (ПИТ).

8.2. а) На рис. 8.2 показана схема подключения к МП КР580ВМ80 генератора тактовых импульсов КР580ГФ24, шинных формирователей КР580ВА86 и системного контроллера КР580ВК28. Объяснить назначение ГТИ и СК, а также всех внешних выводов МП и СК.

б) На рис. 8.3 показаны временные диаграммы для подсистемы МП КР580ВМ80 (рис. 8.2), поясняющие процесс формирования сигналов управления микропроцессорной системой при выполнении команды ввода IN. Используются следующие обозначения: M^1 , M^2 , M^3 — машинные циклы; A^1 , A^2 — адреса ячеек памяти, в которых размещается двухбайтовая команда ввода; V^1 и V^2 — первый и второй байты команды ввода; D — байт данных, вводимых из порта; W^1 , W^2 , W^3 — слова состояния МП; Y^1 , Y^2 , Y^3 — сигналы управления, формируемые в соответствующих машинных циклах. Заштрихованы многоразрядные сигналы.

Объяснить назначение всех машинных циклов и всех сигналов, показанных на временных диаграммах (см. рис. 8.3). Определить сигналы управления ЧтП, ЗпП, Вв, Выв, ПП в каждом машинном цикле [5, 6, 16, 29].

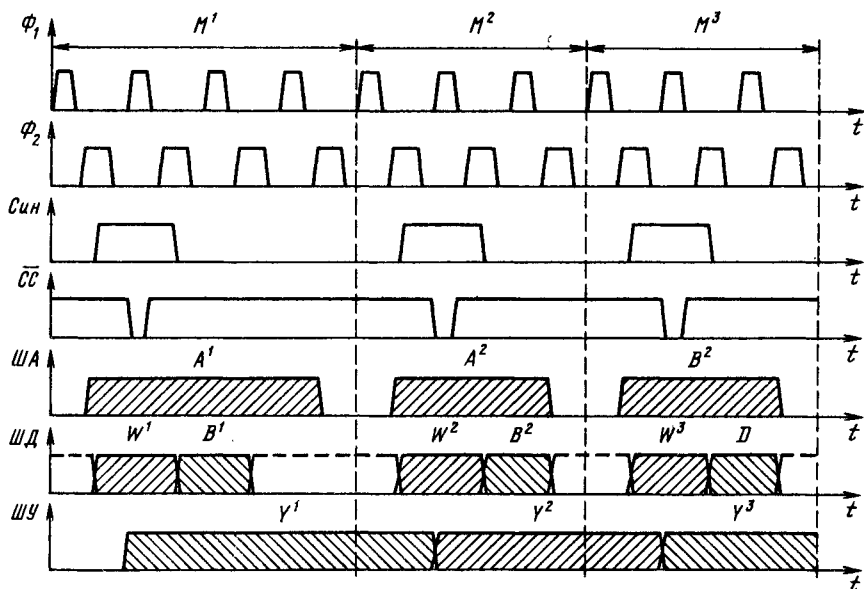


Рис. 8.3. Временные диаграммы при выполнении команды ввода

в) Составить временные диаграммы, аналогичные приведенным на рис. 8.3, для команды загрузки аккумулятора из ячейки памяти LDA. Эта команда выполняется за четыре машинных цикла: M^1 —выборка кода команды, M^2 и M^3 —чтение из памяти второго и третьей байтов команды (адреса ячейки памяти), M^4 —чтение из памяти байта данных. Определить сигналы управления в каждом машинном цикле.

8.3. Проверить, обеспечивается ли в схеме на рис. 8.2 «стыковка» микросхем по электрическим параметрам. Необходимые для расчета электрические параметры микросхем серии КР580 приведены в табл. 8.1 [16, 5].

Таблица 8.1.

Параметр	МП КР580ВМ80	ШФ КР580ВА86	СК КР580ВК28	ППА КР580ВВ55
Входной ток низкого уровня I_{IL} , мА		−0,2	−0,2	−0,2
Входной ток высокого уровня I_{IH} , мА		0,05	0,05	0,05
Выходной ток низкого уровня I_{OL} , мА	1,9	32	Выходы Q 32	
Выходной ток высокого уровня I_{OH} , мА	−0,15	−5	Выходы Q 6	

8.4. К подсистеме МП (см. рис. 8.2) необходимо подключить 64 программируемых периферийных адаптера KP580BV55. Проверить, достаточна ли для этого нагрузочная способность системного контроллера KP580BK28 (см. табл. 8.1).

8.5. Составить для МП KP580 алгоритм и программу преобразования унитарного 8-разрядного кода в двоичный 3-разрядный 4-2-1 (имитация шифратора). Унитарный код $y_7 y_6 \dots y_0$ вводится через порт 07, двоичный код $x_3 x_2 x_1$ выводится через порт 05 (три младших разряда).

8.6. Составить алгоритм и программу преобразования 3-разрядного двоичного кода 4-2-1 в унитарный (имитация полного дешифратора на три входа). Двоичный код $x_3 x_2 x_1$ вводится через порт 07 (три младших разряда), унитарный код $y_7 y_6 \dots y_0$ выводится через порт 05.

8.7. В чем отличие синхронного и асинхронного способов обмена данными между МП и внешним устройством (ВУ)?

8.8. МПС, схема которой приведена на рис. 8.1, содержит ППА KP580BV55, адресуемый кодом FB. К ППА подключены следующие внешние устройства: к порту ПС (адрес FA)—наборное поле из восьми тумблеров, к порту ПВ (адрес F9)—восемь светодиодных индикаторов.

Составить циклическую программу, обеспечивающую ввод данных из порта ПС, обработку их в МП и вывод результата в порт ПВ. Обработка заключается в выполнении одной из следующих операций:

а) маскирования (установки в нулевое состояние) нечетных разрядов;

б) маскирования четырех старших разрядов;

в) инвертирования четных разрядов;

г) циклического сдвига влево на один разряд;

д) инвертирования всех разрядов.

8.9. Составить для МПС алгоритм и программу формирования прямоугольного напряжения с частотой f , уровнями $a=0$ и 1 и скважностью $q=2$ (меандр). Напряжение должно сниматься с выхода младшего разряда порта ПВ (адрес F9) ППА (адрес РУС ППА FB). Вывести соотношение для параметров программы. Оценить погрешность установки частоты. Произвести расчет для $f=1$ кГц.

8.10. Условия те же, что в задаче 8.9, но скважность импульсов $q \neq 2$. Произвести расчет и составить программу для $f=1$ кГц и $q=1,67$.

8.11. МПС (см. рис. 8.1) используется в качестве синтезатора звуков. К МПС подключены через ППА KP580BV55 клавиатура из восьми клавиш и динамик. При нажатии на одну из клавиш синтезатор должен выдавать звуковой сигнал с соответствующей частотой, указанной в табл. 8.2.

Таблица 8.2

Клавиша I	0	1	2	3	4	5	6	7
Тон	До	Ре	Ми	Фа	Соль	Ля	Си	До
Частота f_i , Гц	261,6	293,7	329,6	349,2	392	440	493,8	523,2

Составить программу, определить ее параметры.

Указания. Константы, определяющие частоты сигнала, необходимо хранить в памяти в виде массива. Для ввода данных с клавиатуры используйте порт ПС ППА, для вывода — младший разряд порта ПВ ППА.

8.12. МПС (см. рис. 8.1) используется в качестве генератора меандра с дискретно регулируемым периодом колебаний. Диапазон регулировки 100...2000 мкс с дискретностью 20 мкс. Установка необходимого периода должна производиться с помощью наборного поля из восьми тумблеров в двоичном коде, причем единица младшего разряда должна соответствовать 10 мкс. (Например, код $k=01010010_{(2)}=82$ означает установку периода 820 мкс.) Напряжение должно сниматься с младшего разряда выходного порта.

8.13. МПС (см. рис. 8.1) используется в качестве частотного модулятора, который работает по следующему правилу: через порт FA ППА в МПС поступает число W (интервал времени между различными значениями числа W не менее 100 мс); МП периодически опрашивает порт FA и выдает через младший разряд порта F9 меандровое напряжение с частотой $f=300+30W$ (Гц).

Составить алгоритмы и программы для следующих случаев:

а) число W имеет два значения $W \in \{0, 1\}$, причем вводится через младший разряд порта;

б) число W имеет три значения $W \in \{-1, 0, 1\}$ и вводится в прямом коде;

в) число W имеет восемь значений $W \in \{0, 1, 2, \dots, 7\}$ и вводится в унитарном коде.

Оценить точность установки частоты для каждого случая при тактовой частоте МП $f_T=2$ МГц.

8.14. Составить алгоритм и программу для МП КР580, имитирующие работу непрерывно работающего управляемого счетчика с модулем счета k и начальным состоянием Q^0 . Коды текущих состояний счетчика Q должны выводиться через порт 01. Управление счетчиком производится с помощью внешних сигналов x_2 и x_1 (табл. 8.3), которые должны вводиться через порт 02 двумя младшими разрядами байта (формат

000000 x_2x_1). Проверка значений x_2 и x_1 должна осуществляться каждый раз после вывода очередного кода Q .

Т а б л и ц а 8.3

x_1	x_2	Режим счетчика	Вывод
0	Ф	Установка начального состояния	Q^0
1	0	Прекращение счета	Последнее Q
1	1	Счет	Текущее Q

Рассмотреть два варианта счетчика: а) суммирующий (принять $Q^0=0$, $k=100$); б) вычитающий (принять $Q^0=9$, $k=10$).

8.15. К МПС (см. рис. 8.1) через порт ПВ (адрес F9) ППА (адрес РУС ППА FB) подключен 8-разрядный цифро-аналоговый преобразователь (ЦАП), через порт ПС (адрес FA)—наборное поле из восьми тумблеров. МПС используется для формирования ступенчатого пилообразного напряжения (рис. 8.4) с начальным уровнем $U^0=0$, высотой ступеньки $\Delta U_{ст}=0,01$ В, фиксированной крутизной рабочего хода $S_p=U_m/T_p$ и амплитудой U_m , изменяемой в диапазоне 0,1...2,55 В с дискретностью $\Delta U_m=0,01$ В. Установка амплитуды производится в наборном поле в двоичном коде. Составить алгоритм и программу. Произвести конкретный расчет при $U_m=1$ В и $S_p=0,1$ В/мкс.

Указание. Считать, что зависимость уровня выходного напряжения от входного кода пропорциональная, причем минимальный и максимальный коды $0=00000000_{(2)}$ и $255=11111111_{(2)}$ соответствуют уровням 0 и 2,55 В.

8.16. Условия и задание те же, что в задаче 8.15, но ступеньки должны иметь высоту $\Delta U_{ст}=0,08$ В.

8.17. Условия и задание те же, что в задаче 8.15, но МПС используется для формирования импульсов ступенчатой треугольной формы с одинаковой длительностью прямого и обратного хода $T_{пр}=T_{обр}=1$ мс.

8.18. Условия и задания те же, что в задаче 8.15, но импульсы должны иметь сложную форму, заданную в табл. 8.4, в которой U_i и T_i —соответственно уровень и длительность i -й ступеньки.

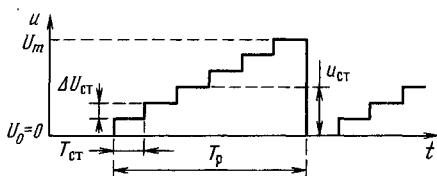


Рис. 8.4. Ступенчатое пилообразное напряжение

Таблица 8.4

i	0	1	2	3	4	5	6	7
U_i , В	0	0,1	0,5	1,5	1,0	1,8	0,6	0,2
T_i , мкс	200	100	100	300	100	100	200	100

8.19. МПС для цифровой обработки аналоговых сигналов включает в себя подсистему МП (рис. 8.2), запоминающие устройства, а также подключенные через порт ввода 01 6-разрядный аналого-цифровой преобразователь (АЦП) и через порт вывода 02 8-разрядный цифро-аналоговый преобразователь (ЦАП) (рис. 8.5). Выбрать элементы МПС, объяснить их назначение.

Указание. В АЦП должна обеспечиваться пропорциональная зависимость между входным напряжением и цифровым сигналом, причем диапазону входных напряжений $0...0,63$ В должен соответствовать диапазон цифровых сигналов $0...63$, а в ЦАП должна обеспечиваться пропорциональная зависимость между цифровым сигналом и выходным напряжением, причем диапазону цифровых сигналов $0...255$ должен соответствовать диапазон выходных напряжений $0...2,55$ В.

8.20. На вход МПС (см. рис. 8.5) подаются положительные трапецеидальные импульсы с амплитудой $0,2...0,6$ В и длительностью не менее 100 мкс. Составьте алгоритм и программу, обеспечивающие работу МПС в качестве ограничителя сверху на заданном уровне $U_{\pi}=0,5$ В (см. задачу 8.19).

8.21. Условия те же, что в задаче 8.20, но необходимо обеспечить двустороннее ограничение снизу на уровне $U^0=0,3$ В и сверху — на уровне $U^1=0,5$ В.

8.22. а) МПС, схема которой представлена на рис. 8.5 (см. также задачу 8.19), работает в качестве селектора импульсов по

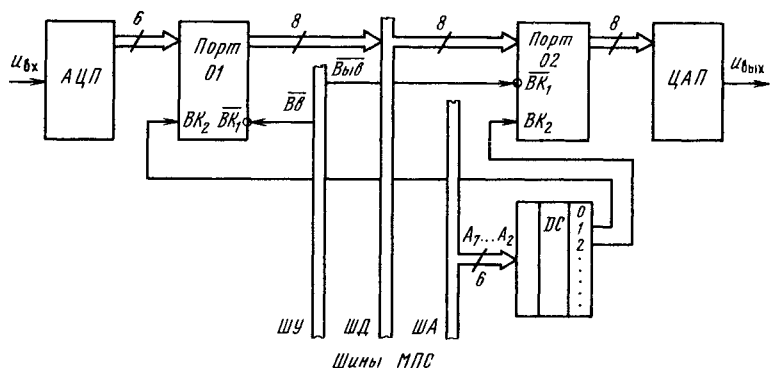


Рис. 8.5. МПС для цифровой обработки аналоговых сигналов

длительности. На вход поступают положительные прямоугольные импульсы с амплитудой 0,2 В...0,6 В, длительностью 10...500 мкс и интервалом между импульсами не менее 2 мс. Обеспечить формирование импульсов с амплитудой 1 В и длительностью 1 мс каждый раз, как на входе появляется импульс с длительностью больше 300 мкс.

б) На вход АЦП МПС (см. рис. 8.5) поступает плавно меняющееся напряжение. Необходимо обеспечить формирование на выходе ЦАП прямоугольных импульсов с длительностью $t_n = 30$ мкс, начальным уровнем $U^0 = 0$ и амплитудой $U_m = 2$ В каждый раз, когда входное напряжение, возрастая, достигает уровня $U_n = 0,5$ В.

в) На вход АЦП МПС (см. рис. 8.5) поступают положительные импульсы произвольной формы с амплитудой не более 0,5 В. Составьте алгоритм и программу, обеспечивающие задержку этих импульсов на заданное время.

г) МПС (см. рис. 8.5) работает в качестве селектора серий импульсов (см. задачу 8.19). Входные импульсы имеют амплитуду 0,2...0,5 В и длительность 50 мкс. На выходе должен формироваться прямоугольный импульс с амплитудой 2 В и длительностью 100 мкс в том случае, если в течение интервала времени 1 мс на вход поступило не менее пяти импульсов.

8.23. МПС (см. рис. 8.5) используется в качестве однородного цифрового фильтра — дециматора, в котором обработка сигнала производится в соответствии с формулой

$$y_n = \sum_{i=0}^{N-1} x_{n-i},$$

где x_{n-i} — цифровой отсчет входного сигнала в $(n-i)$ -м интервале дискретизации; $n = N, 2N, 3N, \dots$; N — число суммируемых отсчетов; y_n — цифровой отсчет выходного сигнала в n -м интервале дискретизации [9]. В фильтре-дециматоре выходные отсчеты следуют с частотой, в N раз меньшей частоты следования входных отсчетов.

Составить алгоритм и программу для этого фильтра в предположении, что $y_n \leq 255$. Определить максимальные частоты дискретизации входного и выходного сигналов при $N = 4$.

Указание. Предполагается, что к моменту подачи команды ввода в МП IN 01 в АЦП уже сформирован очередной отсчет входного сигнала, а к моменту подачи команды вывода из МП OUT 02 ЦАП готов к приему выходного сигнала, т. е. реализуется синхронный режим ввода-вывода.

8.24. МПС с АЦП и ЦАП (см. рис. 8.5) используется в качестве однородного цифрового фильтра, в котором обработка сигнала производится согласно формуле

$$y_n = \sum_{i=0}^{N-1} x_{n-i},$$

где $n = N, N+1, N+2, \dots$, причем входные и выходные отсчеты следуют с одинаковой частотой. Остальные условия и задание — те же, что в задаче 8.23.

8.25. Условия и задания те же, что в задачах 8.23 и 8.24, но выходной отсчет y_n может превышать 255.

8.26. Что общего и чем отличаются команды вызова подпрограммы обработки прерываний RST и обращения к подпрограмме CALL?

8.27. Какова последовательность действий МП КР580 после получения команды RST (рис. 8.6)?

8.28. Объяснить назначение команд EI, DI, PUSH и POP при работе с прерываниями (см. рис. 8.6).

8.29. Составить подпрограмму обработки прерываний, которая решает следующую задачу. После запроса прерывания и получения команды RST МП опрашивает устройство ввода-вывода, определяет знак вводимого через порт 05 числа, представленного в дополнительном коде, и выводит это число через порт 07 в том случае, если оно положительное, после чего возвращается к выполнению основной программы. Предполагается, что других подпрограмм обработки прерываний нет.

8.30. Составить подпрограмму обработки прерываний, решающую следующую задачу. В МП определяется полусумма числа, находящегося в аккумуляторе, и числа, находящегося в ячейке памяти с адресом, записанным в регистровой паре HL; результат загружается в ту же ячейку памяти.

8.31. Составить подпрограмму обработки прерываний, по которой в МП должна вычисляться и запоминаться в ячейке памяти с меткой M1 величина

$$z = \cos(X+5) - \cos(X+3),$$

где X — число, вводимое через порт 05 ($0 \leq X \leq 255$).

Указание. В ПЗУ имеется подпрограмма вычисления косинуса с меткой COSN, где N — число, находящееся в аккумуляторе в момент обращения к подпрограмме.

8.32. Разработать МПС, предназначенную для проверки в условиях серийного производства модулей оперативной памяти емкостью 1 Кбайт (1024×8). Каждый модуль выполнен в виде платы, на которой смонтированы восемь БИС КР565РУ2А (1024×1) с соответствующим обрамлением. Предусмотреть индикацию дефектной БИС и адреса дефектной ячейки памяти.

Рекомендуется использовать следующие тестовые сигналы: 00000000, 01010101, 10101010 и 11111111. Модуль считать неработоспособным, если хотя бы в одной ячейке памяти записанный и считанный тестовые сигналы не совпадают. Изобразите структурную и принципиальную схемы МПС, составьте алгоритм и программу.

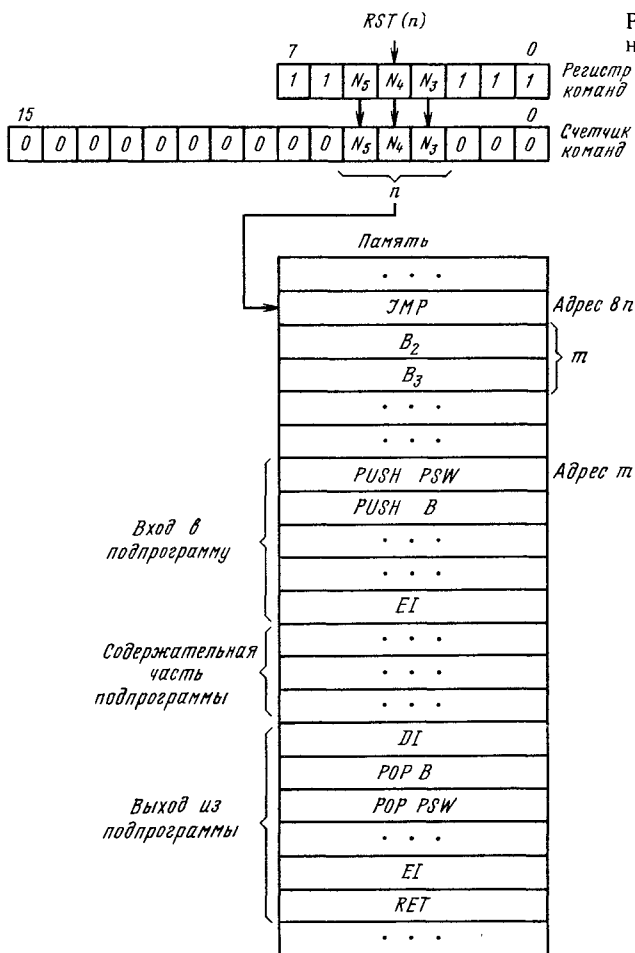


Рис. 8.6. Схема выполнения прерываний в МПС

Указание. В МПС рекомендуется использовать МП КР580ВМ80, индикацию осуществить на светодиодах и звуковыми сигналами высокого и низкого тона (соответствующие подпрограммы ВЕЕРН и ВЕЕРЛ считать известными и записанными в ПЗУ).

8.33. Обосновать известный способ вычисления значений логической функции $F(x_1, x_2, \dots, x_n)$, заданной в дизъюнктивной форме [8], в основе которого лежит формирование и использование тестового n -разрядного кода, который обращается в 0 только в том случае, если данный минтерм (импликант) функции на данном наборе значений аргументов обращается в 1. Составьте алгоритм и программу, если набор значений аргументов вводится в МПС через порт X (предполагается, что

$n \leq 8$); результат вычисления ($F=0$ или $F=1$) выводится через порт F .

8.34. В МПС необходимо моделировать комбинационное цифровое устройство, реализующее логическую функцию $F(x_1, \dots, x_8) = x_1 x_2 \bar{x}_3 x_7 x_8 \vee \bar{x}_2 x_3 x_4 \bar{x}_5 \bar{x}_8$, по алгоритму и программе, приведенным в Р8.33. Составить коды, представляющие указанную функцию, и проверить данный алгоритм на наборах значений аргументов $V=11010011, 11111111, 00000000$.

8.35. Описать алгоритм сбора и обработки одномерного массива данных [31].

8.36. Составить фрагмент формирования массива длины $N=57H$ однобайтовых целых чисел, вводимых через порт ввода с адресом $05H$; массив разместить в области памяти МПС, начиная с ячейки с адресом $BASE=025DH$.

8.37. Составить фрагмент программы формирования массива длины $N=25H$ из двухбайтовых элементов, вводимых через порт ввода с адресом $05H$. Массив разместить в области памяти МПС, начиная с ячейки с адресом $025DH$.

8.38. Составить программу нахождения максимального числа в массиве целых однобайтовых чисел, размещенных в области памяти МПС, начиная с ячейки с адресом $025DH$. Длина массива $25H$ размещена в ячейке памяти с адресом $8060H$.

8.39. В 3У МПС размещены два массива X, Y однобайтовых элементов, содержащих одинаковое число положительных целых чисел. Каждый из массивов X, Y размещается на одной странице памяти, т. е. длина каждого из массивов $N < 256 = 2^8$. Например, массив X размещен на странице памяти XX (старший байт адреса); первый элемент массива располагается в ячейке ОЗУ с адресом, младший байт которого равен $01H$, т. е. адрес этого элемента будет $XX01H$, а адрес любого элемента массива X — в диапазоне $(XX00, XX00+00FF)$ (рис. 8.7). Например, $XX=20$, первый элемент массива X имеет адрес $2001H$, а конечный элемент — адрес $20FFH$.

Сформировать массив Z однобайтовых чисел: а) $Z = X \oplus Y$; б) $Z = X \wedge Y$; в) $Z = \bar{X} \wedge Y$.

Элементы Z_i массива Z являются результатом выполнения логических операций над соответствующими элементами X_i, Y_i массивов X и Y .

8.40. В 3У МПС размещен массив из $N < 256$ однобайтовых элементов, каждый из которых неотрицательное целое число. Символический адрес начального элемента массива — $BASE$. Вывести в порт вывода с символическим адресом $OPORT$ номер (индекс) первого элемента

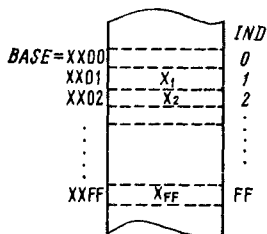


Рис. 8.7. Размещение массива X на одной странице памяти

(ближнего к началу массива), значение которого равно нулю, т. е. 00H.

8.41. В ЗУ МПС размещен массив из $N < 256$ однобайтовых элементов, каждый из которых неотрицательное целое число. Символический адрес начального элемента массива — BASE. Необходимо вывести в порт вывода с символическим адресом OPORT самый близкий к концу массива номер (индекс) элемента, значение которого равно FFH.

8.42. Указать основные особенности формирования в ОЗУ МПС структуры (массива) данных в виде *очереди*.

8.43. Составить программу включения по запросу элемента в очередь и исключения по запросу элемента из очереди. Очередь длиной $N < 256$ размещается на одной странице памяти, начиная с ячейки с символическим адресом BASE в направлении увеличения адресов. Элементы поступают в очередь из порта с символическим адресом PORT1 и исключаются в порт с адресом PORT2. Признак x_1 поступления запроса на включение в очередь — код 01H — размещен в ячейке ОЗУ с символическим адресом ADRPR1; признак x_2 поступления запроса на исключение элемента из очереди — код 01H — в ячейке ОЗУ с символическим адресом ADRPR2. Обращение к очереди асинхронное; признаки включения и исключения из очереди опрашиваются постоянно.

ОТВЕТЫ И РЕШЕНИЯ

Р8.1. См. [5, с. 90; 6, с. 11].

Р8.2. а) ГТИ предназначен для создания последовательностей тактовых импульсов Φ_1 и Φ_2 , а также сигналов готовности Гт, сброса Сбр и строба состояния СС.

Системный контроллер СК предназначен для формирования сигналов управления МПС: чтения данных в МП из памяти ЧтП, записи данных из МП в память ЗпП, ввода данных из устройства ввода-вывода в МП Вв, вывода данных из МП в устройство ввода-вывода Выв, подтверждения прерывания ППр. В СК имеется регистр-фиксатор слова состояния МП, информация в котором обновляется в начале каждого машинного цикла; поэтому сигналы на выходах СК имеют длительность соответствующего машинного цикла.

Выводы МП имеют следующее назначение (приведены русские и английские обозначения):

$A_{15}...A_0$ — выходы с тремя состояниями — адресная шина, обеспечивающая адресацию ячеек памяти и устройства ввода-вывода;

$D_7...D_0$ — входы-выходы с тремя состояниями — шина данных, обеспечивающая двунаправленную передачу данных и команд между МП, памятью и устройством ввода-вывода;

Син (SIN)—выход сигнала синхронизации, указывающего момент начала каждого машинного цикла;

Φ_1 и Φ_2 (C1 и C2)—входы тактовых импульсов;

ГТ (RDY)—вход сигнала готовности, информирующего МП о том, что на ШД имеются данные, считанные из памяти или устройства ввода-вывода;

Пм (RC)—выход сигнала, разрешающего прием информации из памяти или устройства ввода-вывода;

Вд (WR)—выход сигнала готовности к выдаче информации в память или устройство ввода-вывода;

ЗЗх (HLD)—вход сигнала запроса на переход МП в состояние захвата;

ПЗх (HLDA)—выход сигнала, подтверждающего состояние захвата;

ЗПр (INT)—вход сигнала запроса прерывания;

РПр (INTE)—выход сигнала, разрешающего прерывание;

Ож (WI)—выход сигнала, подтверждающего состояние ожидания;

Сбр (SR)—вход сигнала, по которому происходит обнуление счетчика команд.

Кроме того, МП имеет три вывода питания (+12, +5, -5 В) и вывод корпуса—всего 40 выводов.

Выводы СК:

ЧтП и ЗпП—чтение из памяти и запись в память;

Вв и Выв—ввод и вывод из/в устройство ввода-вывода;

ППр (INTA)—подтверждение прерывания.

Остальные выводы аналогичны выводам МП.

Двухбайтовая команда ввода выполняется в МП КР580 за три машинных цикла: M^1 —чтение 1-го байта команды B^1 (кода операции ввода) из ячейки памяти с адресом A^1 ; M^2 —чтение 2-го байта команды B^2 (адреса порта ввода) из ячейки памяти с адресом $A^2 = A^1 + 1$; M^3 —ввод в МП байта данных D из порта. На рис. 8.3 Φ_1 и Φ_2 —последовательности тактовых импульсов, снимаемых с выходов ГТИ (см. рис. 8.2). Как видно из рис. 8.3, команда ввода выполняется за 10 тактов.

Сигнал синхронизации Син генерируется микропроцессором в начале каждого машинного цикла; он подается в ГТИ, в результате чего в последнем формируется импульс СС (строб состояния МП), совпадающий по времени со вторым тактовым импульсом последовательности Φ_1 . В начале машинных циклов M^1 и M^2 на шине адресов ША формируются 16-разрядные адреса ячеек памяти A^1 и A^2 , а в начале машинного цикла M^3 —8-разрядный адрес порта ввода B^2 (в последнем случае восемь старших разрядов ША не несут полезной информации).

Во время сигнала синхронизации Син на ШД появляется слово состояния МП W, в котором содержится информация

о типе машинного цикла. Затем по ШД в каждом машинном цикле передается либо соответствующий байт команды V^1 или V^2 , либо байт входных данных D, после чего до начала следующего машинного цикла ШД отключается (переходит в третье состояние, показанное на рис. 8.3 штриховой линией).

Слово состояния МП W по сигналу $\overline{CS}$ записывается в СК и сохраняется в нем до начала следующего машинного цикла. Кроме того, на СК поступают из МП сигналы управления ПЗх, Пм и Вд, не показанные на временных диаграммах рис. 8.3. В результате на выходах СК, т. е. в шине управления ШУ, формируется набор сигналов Y, необходимых для управления микропроцессорной системой, в частности сигналы $\overline{ЧтП}$, $\overline{ЗпП}$, $\overline{Вв}$, $\overline{Выв}$, $\overline{ППр}$.

В машинных циклах M^1 и M^2 $\overline{ЧтП}=0$, остальные сигналы имеют единичные значения, в M^3 $\overline{Вв}=0$, остальные сигналы — 1.

в) Во всех четырех машинных циклах

$$\overline{ЧтП}=0, \overline{ЗпП}=\overline{Вв}=\overline{Выв}=\overline{ППр}=1.$$

Р8.3. Наихудшие условия стыковки имеют место на выходе МП ПЗх, который нагружен на входы трех микросхем. При низком уровне сигнала на выходе ПЗх микропроцессор обеспечивает ток нагрузки $I_{OL}=1,9$ мА, а суммарный входной ток трех микросхем

$$|I_{1L\Sigma}| \leq 3 \cdot 0,2 = 0,6 \text{ мА} < I_{OL}.$$

При высоком уровне сигнала на выходе ПЗх обеспечивается ток нагрузки $|I_{OH}|=0,15$ мА, а суммарный входной ток микросхем

$$|I_{1H}| \leq 3 \cdot 0,05 = 0,15 \text{ мА} \leq |I_{OH}|.$$

Таким образом, электрическая стыковка в схеме на рис. 8.2 обеспечивается, но без запаса.

Р8.4. Каждая линия шины данных МПС рис. 8.2 (т. е. выходной шины данных Q СК КР580ВК28) нагружена на 64 входа ППА КР580ВВ55. При низком уровне сигнала на выходе СК обеспечивается ток нагрузки (см. табл. 8.1) $I_{OL}=32$ мА, а суммарный входной ток 64 ППА:

$$|I_{1L\Sigma}| \leq 64 \cdot 0,2 = 12,8 \text{ мА} < I_{OL}.$$

При высоком уровне сигнала на выходе СК обеспечивается ток нагрузки $|I_{OH}|=5$ мА, а суммарный входной ток 64 ППА:

$$|I_{1H\Sigma}| = 64 \cdot 0,05 = 3,2 \text{ мА} < |I_{OH}|.$$

Таким образом, нагрузочная способность СК по выходам достаточна для подключения 64 ППА.

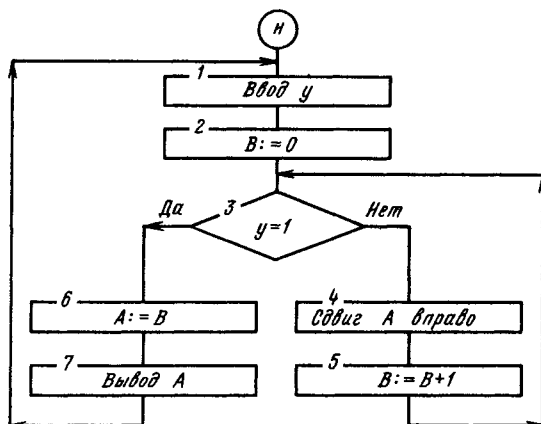


Рис. P8.1. Алгоритм шифратора

P8.5. В алгоритм, имитирующий работу шифратора, можно заложить следующую идею:

если введенное число $y=1$, то вывести 0;

если $y \neq 1$, то подсчитать, сколько раз нужно сдвинуть y вправо, чтобы получить единицу, и это число вывести (рис. P8.1). Программа будет следующей:

```

CODER: IN      07
        MV1    B,0
M1:     CP1    01
        JZ     M2
        RAR
        INR    B
        JMP    M1
M2:     MOV    A,B
        OUT    05
        JMP    CODER
  
```

P8.6. В алгоритм, имитирующий работу дешифратора, можно заложить следующую идею:

если введенное число $x=0$, то вывести унитарный код нуля 00000001;

если введенное число $x \neq 0$, то сдвинуть код 00 000 001 влево x раз и вывести результат сдвига.

Программа будет следующей:

```

DECOD: IN      07
        ADI    00
        JZ     M2
        MOV    B,A
        MVI    A,01
  
```

M1:	RAL	
	DCR	B
	JNZ	M1
	JMP	M3
M2:	MVI	A,01
M3:	OUT	05
	JMP	DECOD

P8.7. Обмен данными между МП и ВУ осуществляется по командам «Ввод» (мнемокод IN) и «Вывод» (мнемокод OUT). При синхронном обмене готовность ВУ к приему данных из МП или выдаче данных в МП обеспечивается за счет жесткого временного согласования между МП и ВУ. При асинхронном обмене такого согласования нет; МП, прежде чем производить обмен данными, запрашивает готовность ВУ к обмену и производит этот обмен только после подтверждения готовности.

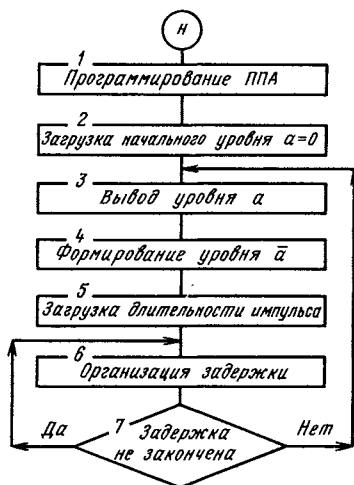
P8.8. а) Программа имеет следующий вид:

PROC:	MVI	A,81H	;Программирование ППА (ре-
			жим 0,
	OUT	0FBH	;Порт ПС—ввод, порт ПВ—
			;вывод)
LP:	IN	0FAH	;Ввод данных из наборного
			;поля
	ANI	0AAH	;Маскирование нечетных раз-
			;рядов
	OUT	0F9H	;Вывод результата на индика-
			;торы
	JMP	LP	;Организация цикла

P8.9. Алгоритм формирования меандра представлен на рис. P8.2. В аккумулятор загружается уровень $a \in \{0, 1\}$, в регистре В организуется счетчик. Программа имеет следующий вид (в столбце справа указано число тактов):

MEANDR:	MVI	A,81H	;Программирование ППА на 7
	OUT	0FBH	;вывод из порта ПВ 10
	MVI	A,0	;Загрузка в аккумулятор 7
			;уровня 0
M1:	OUT	0F9H	;Вывод уровня в порт F9 10
	CMA		;(A)←(A). Загрузка в акку- 4
			;мулятор нового уровня
	MVI	B,K	;Загрузка в регистр В 7
			;числа K, определяющего
			;частоту колебаний
M2:	DCR	B	;Декремент регистра В 5
	JNZ	M2	;Организация внутреннего 10
			;цикла
	JMP	M1	;Организация внешнего 10
			;цикла

Рис. Р8.2. Алгоритм формирования меандрового напряжения



Длительность каждой полуволны формируемого напряжения определяется длительностью выполнения программы, которая, в свою очередь, зависит от числа внутренних циклов K , вводимого командой MVI B, K. Общее число тактов во внешнем цикле

$$N = 10 + 7 + (5 + 10)K + 4 + 10 = 31 + 15K.$$

При тактовой частоте МП $f_T = 2$ МГц период и частота колебаний на выходе будут соответственно

$$T = 2 \frac{N}{f_T} = 31 + 15K \text{ (мкс)}; \quad f = \frac{10^3}{31 + 15K} \text{ (кГц)}.$$

Поскольку число K может находиться в пределах $1 \leq K \leq 256$, предельные значения

$$T_{\max} = 3871 \text{ мкс}; \quad T_{\min} = 46 \text{ мкс}; \quad f_{\min} \approx 260 \text{ Гц}; \quad f_{\max} \approx 21,8 \text{ кГц}.$$

Для получения требуемых периодов и частоты колебаний T и f необходимо, чтобы K имело значение

$$K = \left[\frac{T - 31}{15} \right] = \left[\frac{66,7}{f} - 2,07 \right],$$

где T — в микросекундах; f — в килогерцах; $[x]$ означает ближайшее к x целое число.

Погрешности установки периода и частоты колебаний за счет дискретности K соответственно

$$\Delta T \leq 7,5 \text{ мкс};$$

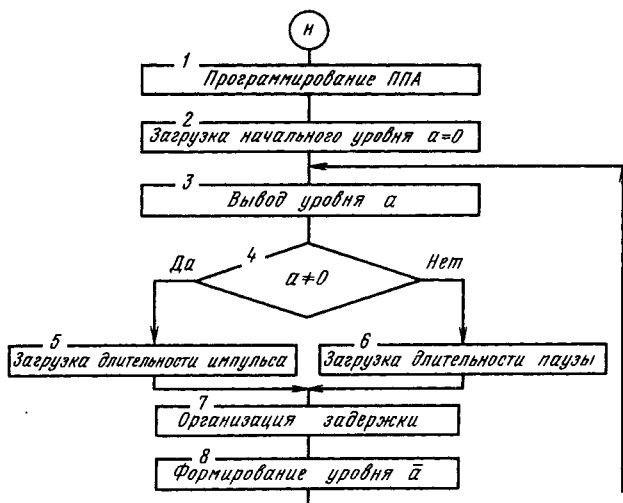


Рис. Р8.3. Алгоритм формирования прямоугольного напряжения с произвольной скважностью импульсов

$$\Delta f = \frac{\partial f}{\partial K} \Delta K \leq \frac{10^3 \cdot 15 \cdot 0,5}{(31 + 15K)^2} = 7,5 f^2$$

(f —в килгерцах; Δf —в герцах).

Для получения $f=1$ кГц необходимо выбрать $K=65$; при этом истинные период колебаний и частота будут соответственно $T_{\text{ист}}=31+15 \cdot 65=1006$ мкс; $f_{\text{ист}} \approx 994$ Гц.

Р8.10. Схема алгоритма на рис. Р8.3 отличается от приведенной на рис. Р8.2 тем, что вместо четвертой команды вначале проверяется выполнение неравенства $a \neq 0$ и затем при $a \neq 0$ регистр С загружается числом L, а при $a=0$ —числом K, где L и K определяют соответственно длительности импульса и паузы.

Программа формирования прямоугольного напряжения со скважностью импульса, не равной 2:

VOLT:	MVI	A,81H	;Программирование ППА	7
	OUT	0FBH	;на вывод через порт В	10
	MVI	A,0	;Загрузка в А уровня $a=0$	7
M1:	OUT	0F9H	;Вывод в порт уровня a	10
	ADI	00	;Пустая команда	7
	JNZ	M3	;Проверка $a \neq 0$	10
	MVI	C,K	;Загрузка в С числа K	7
M2:	DCR	C	;Декремент С	5
	JNZ	M2	;Организация внутреннего цикла	10
	CMA		;Загрузка в А нового	4

			;уровня <i>a</i>	
	JMP	M1	;Организация внешнего	10
			;цикла	
M3:	MVI	C,L	;Загрузка в C числа L	7
	JMP	M2	;Переход к M2	10

(справа приведено число тактов, за которое выполняется каждая команда).

Длительности импульса и паузы при $f_{\tau}=2$ МГц соответственно:

$$t_n = (1/f_{\tau})(10 + 7 + 10 + 7 + 10 + (5 + 10)L + 4 + 10) = \\ = 29 + 7,5L \text{ (мкс)},$$

$$t_n = (1/f_{\tau})(10 + 7 + 10 + 7 + 10 + (5 + 10)K + 4 + 10) = \\ = 29 + 7,5K \text{ (мкс)},$$

откуда

$$L = \left[\frac{t_n - 29}{7,5} \right] = \left[\frac{10^3/q - 29f}{7,5f} \right];$$

$$K = \left[\frac{t_n - 29}{7,5} \right] = \left[\frac{10^3 \frac{q-1}{q} - 29f}{7,5f} \right],$$

где t_n и t_n — в микросекундах; f — в кГц.

Для получения $f=1$ кГц при $q=T/t_n=1,67$ необходимо иметь $K=50=32_{(16)}$, $L=76=4C_{(16)}$.

Р8.11. Клавиатуру из восьми клавиш подключим к порту С (адрес FA) ППА (адрес регистра управляющего слова ППА FB), динамик — к младшему разряду порта В (адрес F9). Будем считать, что нажатие клавиши соответствует 1, отжатие 0. Особенностью задачи является то, что входная информация подается в виде унитарного кода. Поэтому в алгоритме и программе необходимо предусмотреть преобразование унитарного кода в двоичный 4-2-1 (см. задачу 8.5). Программа синтезатора звука представлена ниже. Число тактов во внешнем цикле при формировании l -го сигнала

$$N = 65 + 29l + 15K_l,$$

откуда при тактовой частоте МП 2 МГц получаем

$$K_l = \left[\frac{66,7}{f_l} - 4,3 - 1,9l \right],$$

где f_l — в кГц и $[x]$ обозначает ближайшее к x целое число.

Рассчитанные по последней формуле значения K_l приведены в табл. Р8.1.

Таблица P8.1

I	0	1	2	3	4	5	6	7
K_I	252	221	194	184	158	138	119	110
$K_{I(16)}$	FC	DD	C2	B5	9E	8A	77	6E

Программа синтезатора звуков:

SOUND:	MVI	A,81H	;Программирование ППА 7					
	OUT	0FBH	;(режим 0, порт С—ввод, 10 ;порт В—вывод)					
	LXI	B,00	;Обнуление регистров 10 ;B, C					
INP:	MOV	D,B	;Обнуление регистра D 5					
	LXI	H, 0DO1H	;Загрузка в HL началь- 10 ;ного адреса массива					
	IN	0FAH	;Ввод числа I в унитарном 10 ;коде					
	MOV	C,B	;Обнуление регистра C 5					
	ADD	00	;Формирование признака 7 ;Z в A					
	JZ	INP	;Повторный ввод, если 10 ;(A)=00					
M1:	RAR		;Сдвиг $I_{\text{унит}}$ вправо на один 4 ;разряд с переносом					
	JC	M2	;Переход к выводу при 10 ;(CY)=1					
	INR	C	;Инкремент C 5					
M2:	JMP	M1	;Организация цикла 10					
	MOV	A,D	;Пересылка в A уровня 5 ;сигнала					
	OUT	0F9H	;Вывод сигнала на 10 ;динамик					
	INR	D	;Инвертирование млад- 5 ;шего разряда D—изме- ;нение уровня сигнала ;на противоположный					
	DAD	B	;Вычисление и загрузка 10 ;в HL адреса константы ; K_I					
	MOV	B,M	;Пересылка в B констан- 5 ;ты K_I					
M3:	DCR	B	;Организация задержки 5					
	JNZ	M3	;Организация задержки 10					
	JMP	INP	;Организация повторного 10 ;опроса клавиатуры					

DO1:	0FCH	; Константы, определяющие частоты звуковых сигналов
RE:	0DDH	
MI:	0C2H	
FA:	0B5H	
SOL:	9EH	
LA:	8AH	
SI:	77H	
DO2:	6EH	

P8.12. Решение задачи аналогично решению задач 8.9—8.11. При $f_T = 2$ МГц период колебаний

$$T = 2N/f_T = 60 + 20l = 20(l + 3),$$

где N — суммарное число тактов; l — число повторений внутреннего цикла (метка M2).

Для получения требуемой дискретности 20 мкс во внутренний цикл добавлена пустая команда MOV A,A. Поскольку по условию задачи необходимо, чтобы длительность периода была связана с вводимой константой K соотношением $T = 10K$, должно быть

$$20(l + 3) = 10K, \text{ т. е. } l = K/2 - 3.$$

Поэтому во внешний цикл введены команды RAR (сдвиг вправо на один разряд, т. е. деление на 2), SUI 03 (вычитание константы 3), а также пустая команда MOV A,A.

Программа меандра с регулируемым периодом колебаний:

TIME:	MVI	A,81H	;Программирование ППА 7
	OUT	0FBH	;(режим 0, порт С—ввод, 10
			;порт В—вывод)
	MVI	A,0	;Загрузка в аккумулятор 7
			;уровня 0
M1:	OUT	0F9H	;Вывод уровня в порт F9 10
	CMA		;Получение в аккумулято- 4
			;ре нового уровня
	MOV	C,A	;Хранение уровня в C 5
	IN	0FAH	;Ввод числа K с клави- 10
			;атуры
	RAR		;Деление числа K пополам 4
	SUI	03	;Вычитание константы 3 7
	MOV	A,A	;Пустая команда 5
M2:	MOV	A,A	;Пустая команда 5
	DCR	A	;Декремент A 5
	JNZ	M2	;Организация внутреннего 10
			;цикла
	MOV	A,C	;Пересылка уровня в A 5
	JMP	M1	;Организация внешнего 10
			;цикла

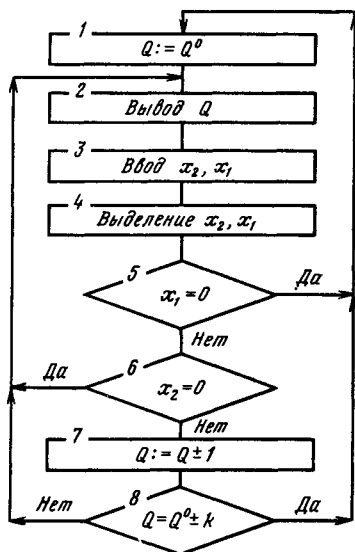


Рис. Р8.4. Алгоритм управляемого счетчика

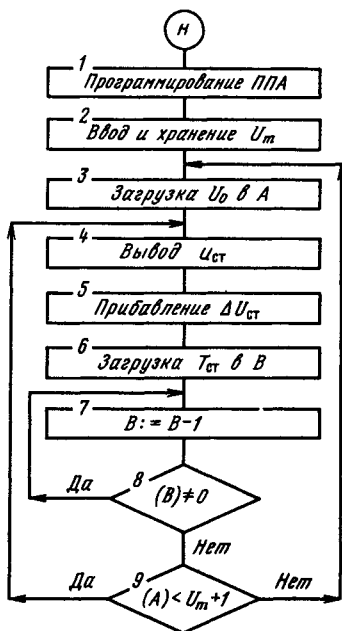


Рис. Р8.5. Алгоритм формирования ступенчатого пилообразного напряжения

Р8.13. См. решение задач 8.9—8.12

Р8.14. Схема алгоритма управляемого счетчика показана на рис. Р8.4. Знак «плюс» соответствует суммирующему счетчику, знак «минус» — вычитающему. Выделение управляющих сигналов x_2 и x_1 в программе можно произвести с помощью команд ANI 02 и ANI 01.

Р8.15. Программа формирования ступенчатого пилообразного напряжения, соответствующая алгоритму рис. Р8.5.:

SAW:	MVI	A, 81H	; Программирование ППА 7
	OUT	0FBH	; Программирование ППА 10
	IN	0FAH	; Ввод в А амплитуды им- 10
			; пульса K_{U_m}
	MOV	C, A	; Хранение в С амплитуды 5
			; импульса
LP1:	MVI	A, 0	; Загрузка в А начального 7
			; уровня $K_{U0} = 0$
LP2:	OUT	0F9H	; Вывод уровня K_U 10
	INR	A	; Приращение уровня ΔK_U 5
	MVI	B, K	; Загрузка в В длительно- 7
			; сти ступеньки $K_{ст}$

LP3:	DCR	B	; Организация в В счетчика 5 ; циклов
	JNZ	LP3	; Организация в В счетчика 10 ; циклов
	CMP	C	; Сравнение уровня K_U с ам- 4 ; амплитудой $K_{Um} + 1$
	JC	LP2	; Переход к LP2 при $K_U < 10$; $< K_{Um}$
	JMP	LP1	; Переход к LP1 при $K_U = 10$; $= K_{Um}$

Константу K_{Um} , задающую амплитуду импульса, определим по формуле

$$K_{Um} = \frac{K_{Um \max}}{U_{m \max}} U_m = \frac{255}{2,55} U_m = 100 U_m.$$

Константу $K_{ст}$, задающую длительность ступеньки, определим из соотношения

$$T_{ст} = \frac{N_{ст}}{f_t} = \frac{\Delta U_{ст}}{\Delta U_m S_p} = \frac{0,01}{0,01 \cdot 0,01} = 100 \text{ мкс}, \quad N_{ст} = T_{ст} f_t = 100 \cdot 2 = 200,$$

где $T_{ст}$ — длительность ступеньки; $N_{ст}$ — число тактов для получения ступеньки, т. е. в цикле LP2 (от команды MVI B, K до команды JC LP2).

Так как $N_{ст} = 36 + 15K_{ст}$, получим

$$K_{ст} = \left[\frac{N_{ст} - 36}{15} \right] = \lfloor 10,9 \rfloor = 11.$$

Р8.16. В программе вместо команды INR A (5 тактов) следует использовать команду ADI $\Delta K_{ст}$ (7 тактов), где $\Delta K_{ст}$ — приращение константы $K_{ст}$, зависящее от величины ступеньки $\Delta U_{ст}$. Длительность ступеньки можно определить по-прежнему по формуле

$$T_{ст} = \Delta U_{ст} / S_p.$$

Р8.17. Решение аналогично решению задачи 8.15.

Р8.18. Константы следует рассчитать исходя из заданных параметров импульса и записать в ЗУ в виде массива. В программе следует предусмотреть перед формированием каждой ступеньки загрузку в РОНЫ МП констант из соответствующих ячеек ЗУ.

Р8.19. Упрощенная схема МПС показана на рис. 8.5. В качестве порта ввода, который подключается к шине данных МПС только по команде МП «Ввод», а в остальное время должен быть отключен от нее, можно использовать многорежимный буферный регистр типа K589ИР12, имеющий три

состояния (см. рис. Р7.1). В качестве порта вывода, который может быть постоянно подключен к ШД МПС, можно использовать либо МБР (см. рис. Р7.2), либо более простой регистр типа K155ИР13. Для адресации портов можно использовать дешифратор типа K155ИД3.

Р8.20. Основной командой программы является сравнение кода входного сигнала K_U с кодом порога $K_{Uп}$; при $K_U < K_{Uп}$ выводится код K_U , а при $K_U \geq K_{Uп}$ — код $K_{Uп}$. Поскольку обработка в этих двух случаях занимает различное время, необходимо его выровнять за счет включения в программу пустых команд.

Р8.21. Решение аналогично Р8.20.

Р8.22. См. решение задач 8.15—8.20.

Р8.23. Алгоритм показан на рис. Р8.6. Программа однородного цифрового фильтра-дециматора:

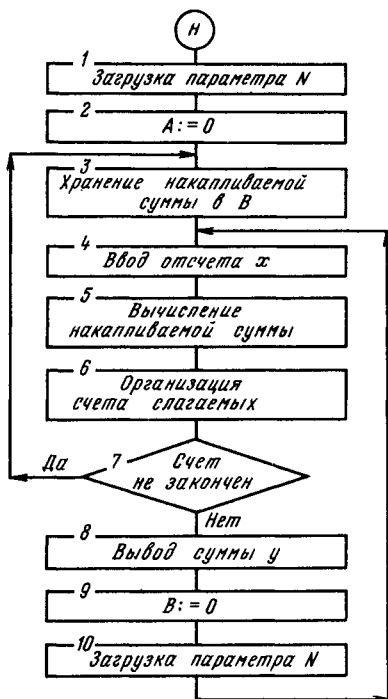


Рис. Р8.6. Алгоритм однородного цифрового фильтра-дециматора

FILTD:	MVI	C,N	;Загрузка константы N, 7
			;определяющей порядок
			;фильтра
	MVI	A,00	;Обнуление аккумулятора 7
LP1:	MOV	B,A	;Хранение накапливаю-5
			;щейся суммы
	MOV	A,A	; 5
	NOP		; Пустые команды для 4
	NOP		; выравнивания длитель- 4
	NOP		; ности циклов 4
	NOP		; 4
	NOP		; 4
	NOP		; 4
LP2:	IN	01	;Ввод очередного отсчета 10
			;x _n
	ADD	B	;Вычисление накапливаю- 4
			;щейся суммы отсчетов
	DCR	C	;Организация счета и за- 5

JNZ	LP1	;цикливание	10
OUT	02	;Вывод у	10
MVI	B,00	;Обнуление регистра В	7
MVI	C,N	;Загрузка константы N	7
JMP	LP2	;Повторение	10

Пустые команды MOV A,A и NOP введены для выравнивания длительностей цикла LP1, в котором рассчитывается частичная сумма, и цикла LP2, в котором рассчитывается окончательная сумма и выводится результат. Для этих длительностей при $f_T = 2$ МГц и $N = 4$ имеем

$$t_{LP1} = \frac{1}{f_T} (5 + 5 + 6 \cdot 4 + 10 + 4 + 5 + 10) = 31,5 \text{ мкс};$$

$$t_{LP2} = \frac{1}{f_T} (10 + 4 + 5 + 10 + 10 + 7 + 7 + 10) = 31,5 \text{ мкс}.$$

Время получения одного выходного отсчета, которое складывается из трех циклов LP1 и одного цикла LP2,

$$t_\Phi = 3t_{LP1} + t_{LP2} = 126 \text{ мкс}.$$

Минимальные интервалы дискретизации входного и выходного сигналов 31,5 и 126 мкс, соответственно максимальные частоты 32 и 8 кГц.

Р8.24. Отличие однородного фильтра от однородного фильтра-дециматора из задачи 8.23 состоит в том, что если в последнем необходимо накапливать сумму отсчетов (вплоть до получения суммы N отсчетов), но нет необходимости сохранять сами отсчеты после получения суммы, то в фильтре без децимации необходимо и накапливать сумму, и сохранять $N-1$ отсчетов. Программу естественно построить таким образом, чтобы в каждом цикле суммирования все хранимые в памяти отсчеты сдвигались и последняя ячейка освобождалась для нового отсчета. В нижеприведенной программе такая процедура организуется с помощью указателя стека SP и команды МП ХТНЛ, которая позволяет обменивать содержимое регистровой пары HL и двух соседних ячеек памяти, адреса которых содержатся в указателе стека. (В данной программе регистр Н не применяется и поэтому ячейки памяти используются «через одну».)

Программа однородного цифрового фильтра:

```

FILTER:  MVI    B,N      ;Загрузка в регистр В кон- 7
                                ;станты N, определяющей
                                ;порядок фильтра
          LXI    SP,ST    ;Загрузка в указатель стека 10
                                ;адреса вершины стека
          IN     01        ;Ввод в аккумулятор через 10

```

			;порт 01 очередного отсче-	
			;та x_n	
	MOV	L,A	;Пересылка x_n в регистр L	5
LP:	XTHL		;Обмен содержимым реги-	18
			;стра и ячейки памяти, ад-	
			;рес которой содержится в	
			;указателе стека	
	ADD	L	;Вычисление в аккумулято-	4
			;ре накапливающейся сум-	
			;мы отсчетов	
	INX	SP	;}(SP)←(SP)+2	5
	INX	SP	;}	5
	DCR	B	;Организация цикла сум-	5
	JNZ	LP	;мирования	10
	OUT	02	;Вывод через порт 02	10
			;выходного сигнала y_n	
	JMP	FILTER	;Переход к вычислению	10
			;следующего значения y_n	
ST:		x_n	;}	
ST+2:		x_{n-1}	;}	
ST+4:		x_{n-2}	;}	
.....			;}	
ST+2(N-1):		$x_{n-(N-1)}$	;}	

Отсчеты сигнала

Длительность вычисления выходного сигнала y_n

$$t_{\Phi} = \frac{1}{f_r}(7 + 10 + 10 + 5 + (18 + 4 + 5 + 5 + 5 + 10)N + 10 + 10) = \frac{1}{f_r}(52 + 47N).$$

При $f_r = 2$ МГц и $N = 4$ $t_{\Phi} = 120$ мкс.

Следовательно, максимальная частота дискретизации сигнала

$$f_d = 1/t_{\Phi} = 8,3 \text{ кГц.}$$

Р8.25. При суммировании отсчетов здесь может произойти переполнение регистров МП. Поэтому в программах необходимо предусмотреть использование регистровых пар и суммирование с помощью команды DAD.

Р8.26. В обоих случаях—как после команды RST, так и после команды CALL—нарушается естественный порядок выполнения команд и осуществляется переход к подпрограмме с адресом, указанным в командах RST и CALL, причем адрес возврата запоминается в стеке, но команда RST подается от внешнего устройства (например, от контроллера прерываний) «аппаратным» путем, а команда CALL содержится в программе, хранящейся в ЗУ.

Р8.27. Команда RST подается в МП по ШД от внешнего устройства (контроллера прерываний или блока приоритетных прерываний), обеспечивает прерывание и запоминание в стеке прерванного процесса, а также определение в МП начального адреса подпрограммы обслуживания прерываний. Схема работы МП в режиме прерываний после получения команды RST показана на рис. 8.6.

Команда RST содержит в третьем — пятом битах один из восьми возможных «уровней» прерывания $n = N_5 N_4 N_3$, соответствующий одной из восьми возможных подпрограмм обслуживания прерываний. В счетчике команд фиксируется адрес $8n$, который можно было бы использовать как начальный адрес подпрограммы, однако, поскольку интервал между адресами $8n$ и $8(n+1)$ составляет всего восемь ячеек памяти, а подпрограмма занимает, как правило, больший объем, по адресу $8n$ обычно располагается команда безусловного перехода $JMP B_2 B_3$, которая и содержит начальный адрес подпрограммы $\langle B_3 \rangle \langle B_2 \rangle = m$.

Возврат к основной программе после выполнения подпрограммы происходит по команде RET; при этом используется адрес, сохраненный в стеке при выполнении команды RST.

Р8.28. Команды EI и DI соответственно разрешают и запрещают прерывания; их включают в различные точки основной программы и подпрограммы обслуживания прерываний в соответствии с алгоритмом решаемой задачи (см. рис. 8.6).

В начале подпрограммы обслуживания прерываний команда DI не ставится, так как при переходе к обслуживанию прерываний новые прерывания автоматически запрещаются впредь до подачи команды EI. Команда разрешения прерываний EI перед началом содержательной части подпрограммы ставится в том случае, если имеются несколько подпрограмм с разными приоритетами. В этом случае во время выполнения подпрограммы контроллер прерываний может выдать новую команду RST, и МП перейдет к обслуживанию другой подпрограммы с большим приоритетом.

Команды PUSH позволяют сохранить в стеке содержимое аккумулятора, РОНов и регистра признаков при переходе к обслуживанию прерываний, а команды POP — восстановить их содержимое перед возвратом к основной программе. Каждая из этих команд позволяет оперировать сразу двумя байтами (содержимым регистровой пары). В подпрограмме обслуживания прерываний целесообразно сохранять и восстанавливать содержимое только тех регистров, которые используются и в ней, и в основной программе.

Во время исполнения команд PUSH и POP должен обеспечиваться запрет прерываний.

Р8.29. Подпрограмма обработки прерываний:

DET: PUSH PSW	;Сохранение в стеке содержимого ак-
	;кумулятора и регистра признаков
IN 05	;Ввод числа из порта 05
ANI 80	;Выделение знака (старшего бита)
	;числа
JNZ M1	;Проверка знака числа, обход вывода
	;при отрицательном знаке
OUT 07	;Вывод числа в порт 07
M1: POP PSW	;Восстановление содержимого аккумуля-
	;мулятора и регистра признаков
EI	;Разрешение прерываний
RET	;Возврат в основную программу

Р8.30. В подпрограмме необходимо предусмотреть сохранение в стеке и последующее восстановление содержимого аккумулятора и регистра признаков. Сохранять содержимое регистровой пары HL, а также других РОНов нет необходимости, так как в подпрограмме оно не меняется.

Р8.31. Обращение к подпрограмме COSN в процессе выполнения подпрограммы обработки прерываний реализуется с помощью команды CALL COSN. В остальном решение аналогично решению задачи 8.30.

Р8.32. При проектировании МПС, предназначенной для проверки работоспособности модулей ОЗУ, за основу можно взять схему на рис. 8.2, к которой необходимо подключить ПЗУ, ОЗУ и индикаторы. Можно использовать БИС ПЗУ типа КР556РТ5 с емкостью памяти 512×8 и 8 БИС ОЗУ типа КР565РУ2А с общей емкостью памяти 1024×8 . (Для решения поставленной задачи столь большая память не требуется; предполагается, что МПС выполняет и другие функции.)

Для подключения индикаторов (светодиодов, динамика) следует использовать регистр K589IP12.

Алгоритм представлен на рис. Р.8.7.

Проверка работоспособности модуля производится последовательно ячейка за ячейкой¹ путем записи и считывания тестового сигнала и сравнения считанной информации с тестовым сигналом. При совпадении этой информации для всех ячеек осуществляется переход к новому тестовому сигналу; после проверки на всех тестовых сигналах выдается звуковой сигнал низкого тона, свидетельствующий об исправности модуля. При несовпадении информации для какой-либо ячейки на каком-либо тестовом сигнале индицируются адрес дефектной ячейки и номер дефектной БИС и выдается звуковой сигнал высокого тона, свидетельствующий о неисправности модуля.

¹ Одна ячейка памяти (линейка) образуется восьмью разрядами восьми БИС ЗУ, имеющими одинаковые адреса.

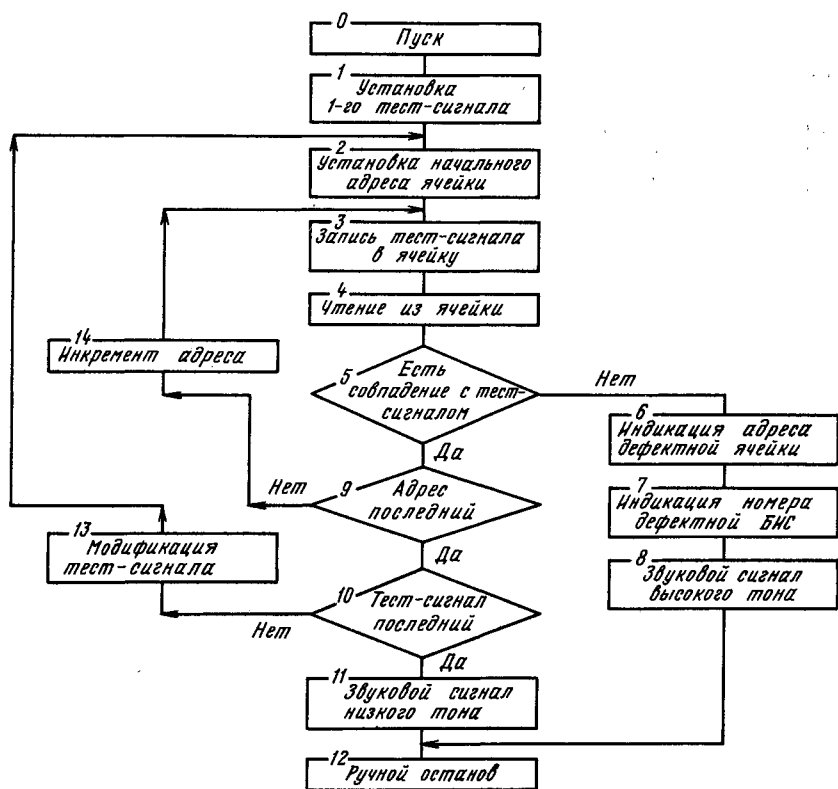


Рис. Р8.7. Алгоритм проверки работоспособности модулей ОЗУ

При составлении программы необходимо руководствоваться следующими соображениями:

для адресации ячеек проверяемого модуля необходимо выбрать область памяти емкостью 1 Кбайт, не используемую в самой МПС (например, от F000 до F3FF);

для перехода от одного тестового сигнала к другому можно использовать тот факт, что рекомендованные в задании тестовые сигналы отличаются один от другого на постоянную величину 01010101;

совпадение считанной из ячейки информации с записанной целесообразно проверять с помощью команды суммирования по mod2; при этом несовпадающий бит будет указывать на номер дефектной БИС.

Р8.33. Пусть $F(x_1, x_2, \dots, x_n) = \bigvee_{j=1}^m R^{(j)}$, где $R^{(j)}$ — j -й минтерм или импликант функции F :

$$R^{(j)} = \rho_1^{(j)} \rho_2^{(j)} \dots \rho_n^{(j)},$$

причем $\rho_i^{(j)} \in \{x_i, \bar{x}_i, 1\}$.

Для запоминания в МПС функции F закодируем каждый из минтермов $R^{(j)}$ двумя n -разрядными кодами $M^{(j)} = \mu_1 \mu_2 \dots \mu_n$ и $C^{(j)} = c_1 c_2 \dots c_n$, образуемыми по правилу, которое представлено в табл. P8.2.

Таблица P8.2

ρ_i	μ_i	c_i
x_i	1	1
$\bar{x}_i$	1	0
1	0	0

Таблица P8.3

b_i	μ_i	c_i	t_i
0	0	0	0
0	0	1	Ф
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	Ф
1	1	0	1
1	1	1	0

Например, $\mu_i = 1$ означает, что в $R^{(j)}$ входит x_i или $\bar{x}_i$, а $c_i = 1$ — что x_i входит в неинвертированном виде.

Пусть необходимо вычислить функцию F на наборе значений аргументов $x_1 = b_1, x_2 = b_2, \dots, x_n = b_n$, который задан кодом $B = b_1 b_2 \dots b_n$, введенным в МПС.

Составим тест $T^{(j)} = t_1 t_2 \dots t_n$ по правилу, представленному в табл. P8.3.

Сопоставляя табл. P8.2 и табл. P8.3, легко заключить, что данный импликант или минтерм на заданном наборе значений аргументов будет равен 1 в том, и только в том случае, если все t_i ($i = 1, \dots, n$) равны 0. Полагая в табл. P8.3 $\Phi = 1$, получаем

$$t_i = \bar{b}_i \bar{\mu}_i c_i \vee \bar{b}_i \mu_i c_i \vee b_i \bar{\mu}_i c_i \vee b_i \mu_i \bar{c}_i = b_i \mu_i \oplus c_i.$$

Полученный тест удобен для МП KP580BM80, поскольку в нем используются операции поразрядного умножения, суммирования по модулю 2 и проверки нулевого результата, входящие в систему команд этого МП.

Алгоритм вычисления логической функции, в котором используется вышеуказанный тест, представлен на рис. P8.8.

Программа вычисления логической функции:

```

FUNC:  IN      PORTX      ;Ввод набора значений аргу-
                                ментов В в аккумулятор
        MOV     B,A        ;Дублирование А в регистре В
        MVI     C,m        ;Организация в регистре С
                                ;счетчика минтермов и им-
```

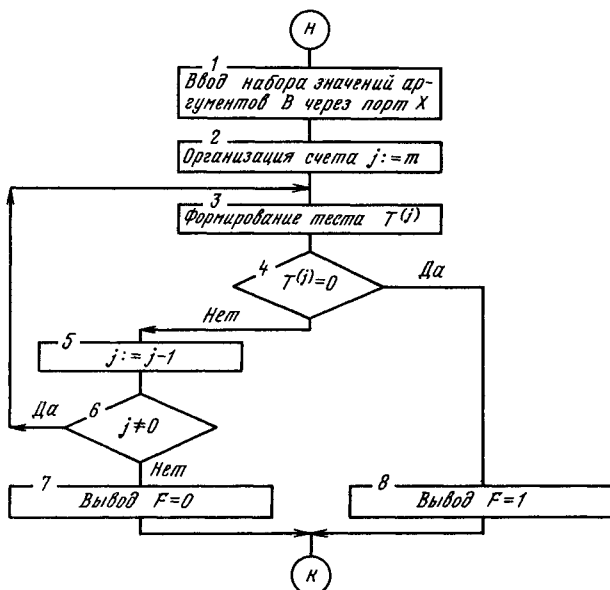


Рис. Р8.8. Алгоритм вычисления логической функции

			;пликантов
	LXI	H,M1	;Загрузка в регистровую пару
			;HL адреса $M^{(1)}$
MASC:	MOV	A,B	;Пересылка В в аккумулятор
	ANA	M	;Вычисление $M^{(j)} \wedge B$
	INX	H	;Инкремент адреса $M^{(j)}$
	BPA	M	;Вычисление теста $T^{(j)} =$
			$= M^{(j)} \wedge B \oplus C^{(j)}$
	JZ	F1	;Переход к выводу F=1 при
			$T^{(j)}=0$
	INX	H	;Инкремент адреса $C^{(j)}$
	DCR	C	;Декремент счетчика
	JNZ	MASC	;Переход к анализу сле-
			дующего минтерма (импли-
			канта) при $T^{(j)} \neq 0$
	MVI	A,00	;Засылка в аккумулятор «0»
	OUT	PORTF	;Вывод значения функции
			;F=0
F1:	MVI	A,01	;Засылка в аккумулятор «1»
	OUT	PORTF	;Вывод значения функции
			;F=1
	HLT		;Останов
			
			

M1:	M ¹	⋮	} Коды минтермов и импликантов
	C ¹	⋮	
	M ²	⋮	
	C ²	⋮	
		⋮	
	M ^m	⋮	
	C ^m	⋮	

Р8.34. Коды, представляющие импликанты функции $F(x_1, \dots, x_8) = x_1 x_2 \bar{x}_3 x_7 x_8 \vee \bar{x}_2 x_3 x_4 \bar{x}_5 \bar{x}_8$, находим по табл. Р8.2:

$M^{(1)} = 11100011$; $C^{(1)} = 11000011$; $M^{(2)} = 01111001$; $C^{(2)} = 00110000$.

Для первого набора значений аргументов по табл. Р8.3 сформируем тест, соответствующий первому импликанту:

$B = 11010011$; $M^{(1)} = 11100011$; $B \wedge M^{(1)} = 11000011$; $C^{(1)} = 11000011$; $T^{(1)} = B \wedge M^{(1)} \oplus C^{(1)} = 00000000$.

Поскольку тест равен 0, первый импликант равен 1 и, следовательно, $F = 1$.

На других наборах значений аргументов тесты для обоих импликантов не равны 0, поэтому $F = 0$.

Р8.35. Одномерный массив — набор элементов данных одинаковой длины, размещенный в определенной области смежных ячеек ЗУ. Пример элемента данных — однобайтовое число. Могут быть и многобайтовые элементы данных. Адрес любого элемента данных в массиве (рис. Р8.9,а) определяется суммой адреса начального элемента (символический адрес BASE) и порядкового номера — индекса IND данного элемента в массиве (если элемент данных многобайтовый, то следует индекс элемента умножить на число байтов). Алгоритм формирования и обработки массивов является циклическим и содержит следующие блоки (рис. Р8.9,б):

1) *инициализация*: во-первых, загрузка указателя адреса начальным адресом BASE массива (обычно в МП КР580 роль указателя выполняет регистровая пара HL); во-вторых, загрузка счетчика числом, равным длине (числу элементов) N массива (обычно роль счетчика выполняет один из РОНов микропроцессора или ячейка ЗУ, например ячейка с символическим адресом LENGTH). Часто реализуют суммирующий счетчик циклов, например, на одном из РОНов, а длину массива N загружают в некоторую ячейку ЗУ (с символическим адресом LENGTH);

2) *ввод данных*: введение элемента массива из порта ввода (например, из порта с символическим адресом IPORT) в ячейку ЗУ, адрес которой размещен в указателе адреса;

3) *обработка массива*: выполнение необходимых арифметических и логических операций над элементами массива, например выполнение операций суммирования элементов массива;

4) *модификация команд*:

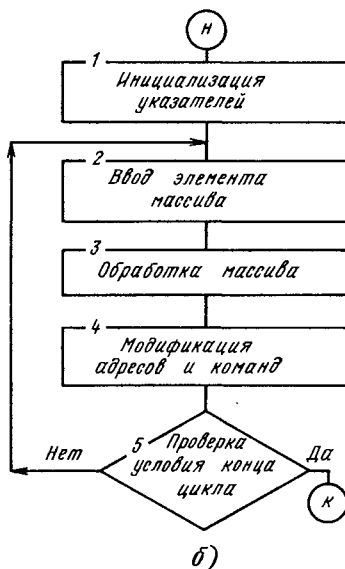
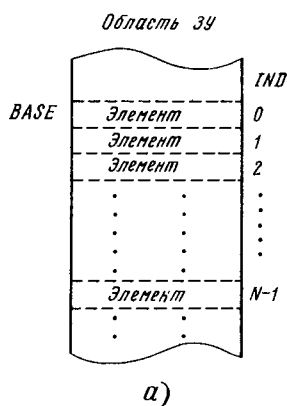


Рис. Р8.9. К решению задачи 8.35:

а) размещение массива в ЗУ; б) схема алгоритма формирования и обработки массива

(указатель) ← (указатель) + 1;

(счетчик) ← (счетчик) - 1;

5) проверка условия окончания цикла ввода элементов массива.

Р8.36. В качестве указателя адресов элементов массива используем регистровую пару (HL), а в качестве счетчика — РОН С. Пусть длина массива хранится в ячейке ЗУ с адресом LENGTH = 8060H.

Фрагмент программы:

LDA	8060H	; (A) = 57H
MOV	C, A	; (Счетчик) = 57H
LXI	H, 025DH	; Загрузка указателя
		; (HL) = 025DH
SAVE:	IN	05H
	MOV	M, A
		; Ввод элемента
		; Пересылка элемента массива
		; в ячейку ЗУ, адрес которой
		; в указателе
	INX	H
		; (HL) ← (HL) + 1
	DCR	C
		; (C) ← (C) - 1
	INZ	SAVE
		; Проверка условия (C) ≠ 0 и
		; продолжение формирования
		; массива

Р8.37. Выбираем в качестве указателя адреса регистровую пару (HL), а в качестве счетчика — РОН С.

Фрагмент программы:

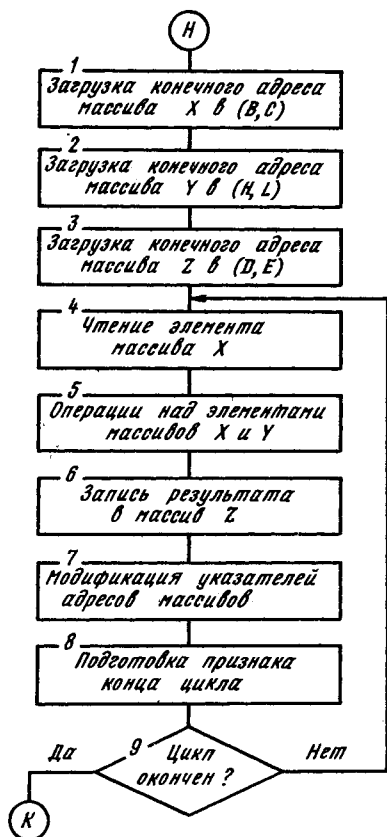
	MVI	C,25H	;Загрузка счетчика
	LXI	H,025DH	;Инициализация указателя ад-
			;реса
SAVE:	IN	05H	;Ввод младшего байта эле-
			;мента
	MOV	M,A	;Пересылка младшего байта
			;элемента в ячейку памяти
	INX	H	;Указатель на старший байт
	IN	05H	;Ввод старшего байта в А
	MOV	M,A	;Загрузка старшего байта в
			;ячейку памяти
	INX	H	;Модификация указателя
	DCR	C	;Счет
	INZ	SAVE	;Проверка окончания цикла
			;и формирования массива

Р8.38. Алгоритм решения задачи может быть таким: в качестве максимального принимается начальный элемент, т. е. элемент с индексом $IND=0$, размещенный в ячейке с базовым адресом 025DH; затем с этим элементом сравнивается следующий элемент с индексом $IND=1$, и если он больше, то принимается за максимум; далее каждый последующий элемент массива сравнивается с ранее зафиксированным максимумом, и если он его превышает, то принимается за новый максимум и фиксируется в регистре максимума. Согласно этому алгоритму составляем программу. Выбираем в качестве указателя адреса регистровую пару (HL), счетчика — регистр С и регистра максимума — регистр А.

Программа:

	LXI	H,025DH	;Инициализация указателя
	LDA	8060H	;Инициализация счетчика в
	MOV	C,A	;РОН С
NMX:	MOV	A,M	;Загрузка максимума в ре-
			;гистр А
TIN:	DCR	C	;Счет в (С)
	JZ	FDD	;Проверка окончания цикла:
			;если (С)=0, то максимум
			;в (А)
	INX	H	;Модификация указателя:
			;(HL) $\leftarrow$ (HL)+1
	CMP	M	;Сравнение нового элемента
			;массива с содержимым (А):
			;если (А)<([M]), то
			;флаг CY=1

Рис. Р8.10. Алгоритм формирования массива



JS NMX

JMP TIN

FDD: XXX

;Если есть перенос из 8-го
;разряда, т. е. если очередной
;элемент больше содержи-
;мо (A), то переход по метке
;NMX; в (A) записывается
;новый максимум
;Очередной элемент массива
;меньше содержимого (A),
;безусловный переход к метке
;TIN
;Максимум определен, разме-
;щен в (A)

Р8.39. Алгоритм решения задачи приведен на рис. Р8.10. Для указателей адресов элементов массивов используем пары регистров: массив X→регистровая пара (BC), массив Y→регистровая пара (HL), массив Z→регистровая пара (DE).

Программа:

	LXI	B,XXFFH	;Загрузка указателей
	LXI	H,YYFFH	;конечных адресов
	LXI	D,ZZFFH	;массивов X, Y, Z
ADRESS:	LDAX	B	;Чтение элемента
			;массива Z
	ANA	M	;Выполнение опера-
			;ции $(A) \leftarrow X_i \wedge Y_i$;
			;для задачи $\bar{X} \wedge Y$
			следует выполнить
			;команды CMA и
			;ANA M, а для зада-
			;чи $X \oplus Y$ команду
			;XRA M
	STAX	D	;Пересылка резуль-
			;тата в ячейку ЗУ с
			;адресом указателя
			;(DE)
	DCX	D	;Изменение текущих
	DCX	H	;адресов элементов
	DCX	B	;массивов
	MOV	A,C	;Подготовка при-
	ADI	00H	;знака для проверки
			;окончания цикла
	INZ	ADRESS	;Проверка условия и
			;переход
	HLT		;Останов

P8.40. В качестве указателя адресов массивов используем регистровую пару (HL), а счетчик организуем в POH C. Алгоритм решения задачи приведен на рис. P8.11.

Программа:

	MVI	C,N	;(C)=N
	LXI	H, BASE	;Начальный адрес в паре (HL)
ADR1:	MOV	A,M	;Пересылка в A элемента из
			;ячейки ЗУ с адресом, запи-
			;санным в (HL)
	CPI	00H	;Установка признака, провер-
			;ка условия
	JZ	ADR2	;и переход по метке
	MOV	A,C	;Установка признака конца
	CPI	01H	;массива
	JZ	ADR3	;При (Z)=0 переход по метке
	DCR	C	;Счет в (C)
	INX	H	;Модификация указателя
	JMP	ADR1	;Безусловный переход по мет-

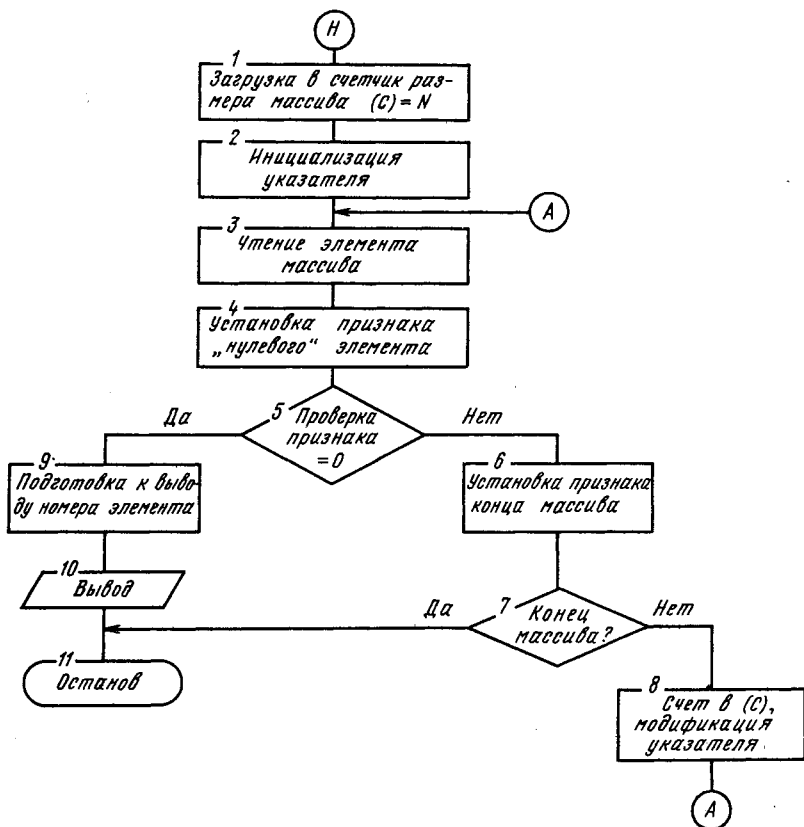


Рис. Р8.11. Алгоритм выделения номера первого нулевого элемента массива

ADR2: MOV	A,C	;ке
OUT	OPORT	; (A) ← (C)
		; Вывод номера элемента из
		; (A) в порт с символическим
		; адресом OPORT
ADR3: HLT		

Примечание. Для поиска номера самого первого элемента, содержащего все единицы, или любого другого кода следует в команде CPI записать соответствующий операнд.

Р8.41. Алгоритм решения задачи совпадает с алгоритмом на рис. Р8.11 из задачи 8.40 при внесении следующих изменений:

1) В блоке 2 необходимо занести не начальный, а конечный адрес массива, т. е. символический адрес последнего элемента массива $ADRN = BASE + N$;

2) в блоке 8 — модификация указателя адреса — следует не увеличивать указатель, а уменьшать указатель: $(HL) \leftarrow (HL) - 1$;

3) в блоке 4 сравнивать код в аккумуляторе не с 0, а с FFH или, возможно, с любым другим заданным в задаче кодом.

Программа:

	MVI	C,N
	LXI	H, ADRN
ADR1:	MOV	A, M
	CPI	FFH
	JZ	ADR2
	MOV	A,C
	CPI	01H
	JZ	ADR3
	DCR	C
	DCX	H
ADR2:	JMP	ADR1
	MOV	A,C
	OUT	OPORT
ADR3:	HLT	

Р8.42. Очередь представляет собой массив, в котором новые элементы можно включать только с конца очереди, а исключать только с начала очереди (в отличие от массива типа «стек», где реализуется принцип «первый приходит, последний уходит», в очереди реализуется принцип «первый приходит, первый уходит»).

Для организации очереди вводятся два указателя: указатель начала очереди, который адресует элемент очереди, подлежащий выходу (исключению), и указатель конца очереди, который адресует последний элемент очереди. Для выхода из очереди считывается элемент, адресуемый указателем, и затем производится увеличение на единицу (инкремент) указателя начала, т. е. очередь как бы растет «вниз» — в сторону больших адресов.

Для включения в очередь элемент массива записывается в ячейку, адресуемую указателем конца очереди, и затем этот указатель увеличивается на единицу.

Чтобы можно было просто обнаружить особые случаи в организации очереди — попытки включить элемент в очередь, в которой все выделенные для массива ячейки памяти уже заполнены, или исключить элемент из пустой очереди, часто область памяти, выделяемую для очереди, размещают на одной странице ЗУ, т. е. у всех элементов очереди адреса отличаются только младшим байтом, а старший байт адреса у всех элементов один и тот же; при этом, естественно, можно реализовать очередь всего на $N < 2^8 = 256$ элементов.

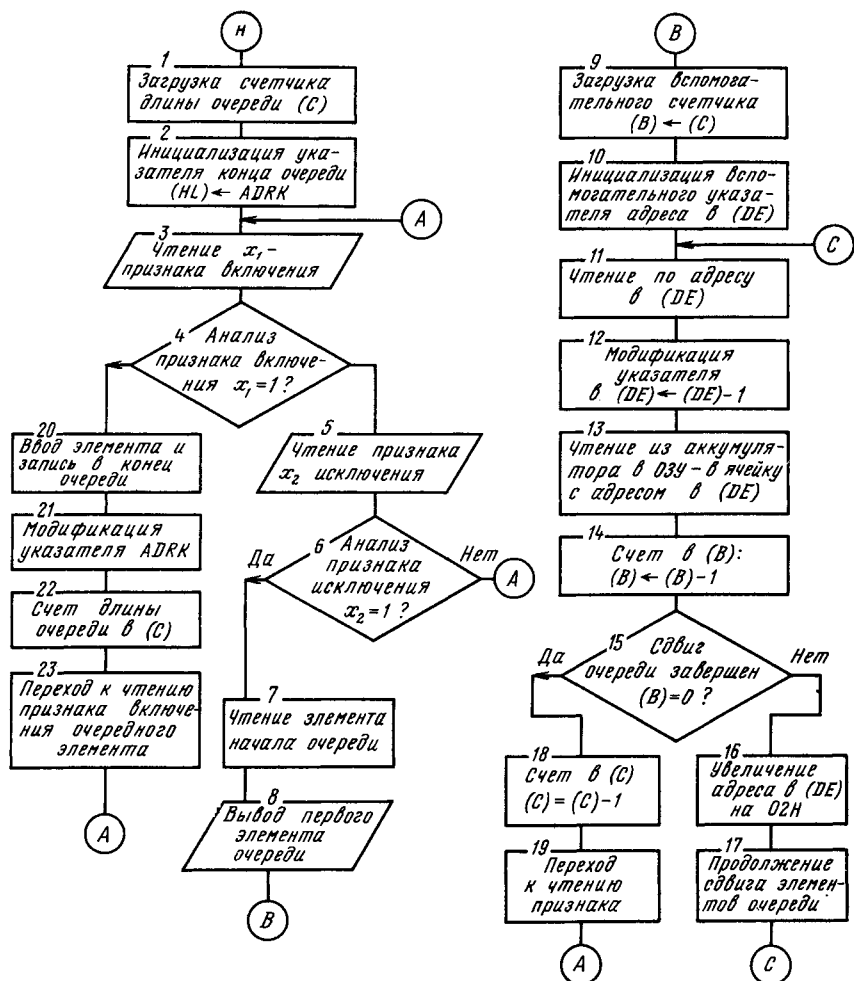


Рис. Р8.12. Алгоритм включения элементов в очередь и исключения из нее

Р8.43. Алгоритм решения задачи приведен на рис. Р8.12. В качестве указателя адреса конца очереди выберем регистровую пару (HL), а указателя адреса элемента очереди при сдвиге — пару (DE). Счетчик длины очереди реализуем на регистре С, а вспомогательный счетчик для сдвига очереди — на регистре В. Предполагается, что исключение из очереди производится только из ячейки с начальным адресом BASE, поэтому после выполнения операции исключения элемента из очереди производится «подъем» всех элементов очереди вверх (в направлении уменьшающихся адресов) на одну ступень.

Программа:

1	MVI	C, 00H	;Загрузка счетчика длины ;очереди
2	LXI	M, ADRK = BASE	;Инициализация указате- ;ля адреса пары (HL) те- ;кущего конца очереди ;ADRK = BASE
3	ADR1: LDA	ADRPR1	;Чтение признака включе- ;ния нового элемента ;в очередь: (A) ← x_1
4	CPI	01H	;Анализ признака x_1 — ;сравнение (A) — 01H и ;проверка условия (флаг ;Z): если (A) = x_1 = 01H, то ;переход к метке ADR5 ;(блок 20)
5	ADR2: LDA	ADRPR2	;Чтение признака исклю- ;чения элемента из очере- ;ди (A) ← x_2
6	CPI	01H	;Анализ признака x_2 — ;сравнение (A) — 01H и пе- ;реход по метке ADR1, ;если $x_2 \neq 01H$
7	JNZ	ADR1	
7	LDA	BASE	;Чтение из ячейки ОЗУ ;с начальным адресом ;(A) ← ([BASE])
8	OUT	PORT2	;Исключение первого эле- ;мента очереди в порт вы- ;вода (PORT2) ← (A)
9	MOV	B, C	;Подготовка к сдвигу эле- ;ментов очереди: Загрузка ;вспомогательного счетчи- ;ка (регистр B) значением ;длины очереди в (C)
10	LXI	D, BASE + 1	;Инициализация вспомо- ;гательного указателя те- ;кущих адресов элементов ;очереди
10	LXI	D, BASE + 1	
11	ADR3: LDAX	D	;Чтение из ячейки ОЗУ по ;адресу в паре (DE) ;(A) ← ([DE])
12	DCX	D	;Модификация вспомога- ;тельного указателя ;(DE) ← (DE) - 1
13	STAX	D	;Сдвиг элемента «вверх»: ;чтение из (A) в ячейку ;ОЗУ с адресом в (DE)
14	DCR	B	;Счет во вспомогательном ;счетчике
15	JZ	ADR4	;Проверка конца (завер- ;шения) сдвига всех эле- ;ментов очереди ;«вверх» — по (B) ≠ 0
16	INX	D	;Если B ≠ 0, то увеличение ;текущего адреса в (DE) ;на 02H и безусловный ;переход к метке ADR3
17	JMP	ADR3	
18	ADR4: DCR	C	;Если (B) = 0, то восстана-

19		JMP	ADR1	вливается значение длины ;очереди в (C) и безуслов- ;ный переход к метке ;ADR1
20	ADR5:	IN	PORT1	;Ввод нового элемента из ;порта ввода
		MOV	M,A	;Включение нового эле- ;мента в конец очереди ;в ячейку ОЗУ по адресу в ;паре (HL)
21		INX	H	;Модификация указателя ;адреса конца очереди ;ADRK:(HL)←(HL)+1
22		INR	C	;Счет в счетчике (C) дли- ;ны очереди
23		JMP	ADR1	;Переход к метке ADR1 ;и продолжение формиро- ;вания очереди.

Примечание. При необходимости в алгоритм и программу могут быть включены блоки и команды проверки отсутствия *переполнения* очереди или *пустой* очереди (сравнение содержимого счетчика в регистре C с заданной длиной N очереди или с 00H).

Приложение 1. СВЕДЕНИЯ О МИКРОСХЕМАХ ТТЛ **СЕРИЙ 133, 134, 155 и 531**

Таблица П1.1. Электрические параметры базовых схем ТТЛ-типа

Параметр	Серия		
	133, 155	134	531 с диодами Шоттки
Выходное напряжение «0» $U_{\text{вых}}^0$, В, не более	0,4	0,3	0,5
Выходное напряжение «1» $U_{\text{вых}}^1$, В, не менее	2,4	2,3	2,7
Коэффициент разветвления по выходу $k_{\text{раз}}$	10	10	10
Среднее время задержки $t_{\text{з, ср}}$, нс, не более	20 ($C_{\text{н}} = 15$ пФ)	100 ($C_{\text{н}} = 40$ пФ)	5 ($C_{\text{н}} = 15$ пФ)
Средняя статическая мощность потребления $P_{\text{пот}}$, мВт, не более	22	2	19
Частота переключений f , МГц, не более	10	3	50

Таблица П1.2. Функциональное назначение ИС ТТЛ

Функциональное назначение	Обозначение	Номер рисунка
Четыре логических элемента 2И-НЕ (133, 155)	ЛА3	П1.1, <i>a</i>
Четыре логических элемента 2И-НЕ (134)	ЛБ1	П1.1, <i>b</i>
Три логических элемента 3И-НЕ (133, 155)	ЛА4	П1.2
Два логических элемента 4И-НЕ (133, 155)	ЛА1	П1.3 (для 155)
Два логических элемента 4И-НЕ (134)	ЛБ2	—
Логический элемент 8И-НЕ (133, 134, 155)	ЛА2	П1.4
Два логических элемента 2И-2ИЛИ-НЕ (133, 134, 155)	ЛР1	П1.5
Шесть логических элементов НЕ (155)	ЛН1	П1.6
Четыре логических элемента 2ИЛИ-НЕ (133, 155)	ЛЕ1	П1.7
Четыре логических элемента 2И (133, 155)	ЛИ1	П1.8
Два логических элемента 4И (155)	ЛИ6	—
Четыре логических элемента 2ИЛИ (133, 155)	ЛЛ1	П1.9
Два триггера Шмита с логическим элементом на входе (133, 155)	ТЛ1	П1.10
Четыре D-триггера с прямыми и инверсными выходами (133, 155)	ТМ7	П1.11

Функциональное назначение	Обозначение	Номер рисунка
Четыре <i>D</i> -триггера (133, 155)	ТМ5	П1.12
Два <i>D</i> -триггера (133, 134, 155)	ТМ2	П1.13
<i>JK</i> -триггер с логикой на входе (133, 134, 155)	ТВ1	П1.14 (для 133, 155)
Два <i>JK</i> -триггера (134)	ТВ14	—
Четыре 2-входовых элемента «исключающее ИЛИ» (155)	ЛП5	—
Селектор-мультиплексор на восемь каналов со стро- бированием (133, 155)	КП7	П1.15
Дешифратор-демультиплексор четыре линии на 16 (133, 134, 155)	ИД3	П1.16
Два 4-входовых расширителя по ИЛИ (133, 155)	ЛД1	—
Одноразрядный полный сумматор (133, 155)	ИМ1	—
Двухразрядный сумматор (133, 155)	ИМ2	—
Четырехразрядный сумматор (133, 155)	ИМ3	—
Одновибратор с логическим элементом на входе (133, 155)	АГ1	—
Четырехразрядный универсальный сдвигающий ре- гистр (133, 134, 155)	ИР1	П1.17 (133, 155)
Реверсивный 8-разрядный регистр сдвига (133, 155)	ИР13	—
Четырехразрядный регистр с тремя состояниями выхода (155)	ИР15	—
Восьмиразрядная схема контроля четности и нечет- ности (134, 155)	ИП2	—
Схема быстрого переноса для арифметическо-логиче- ского узла (134, 155)	ИП4	—
Четырехразрядный двоичный реверсивный счетчик (133, 155)	ИЕ7	П1.18
Двоичный счетчик (133, 134, 155)	ИЕ5	П1.19 (133, 155)

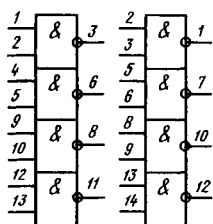


Рис. П1.1

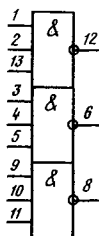


Рис. П1.2

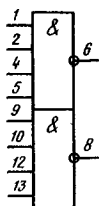


Рис. П1.3

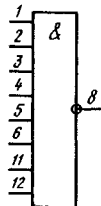


Рис. П1.4

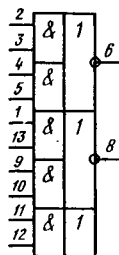


Рис. П1.5

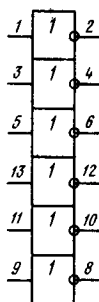


Рис. П1.6

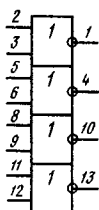


Рис. П1.7

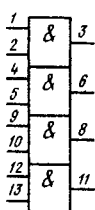


Рис. П1.8

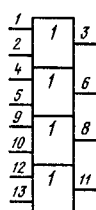


Рис. П1.9



Рис. П1.10

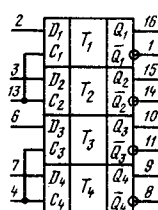


Рис. П1.11

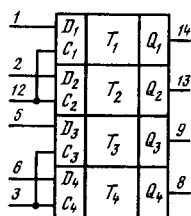


Рис. П1.12

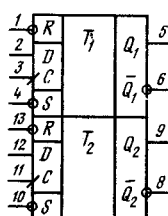


Рис. П1.13

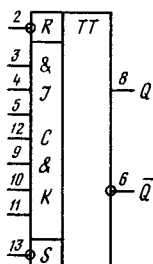


Рис. П1.14

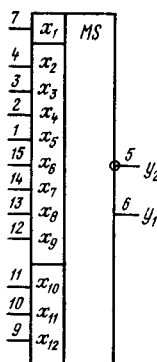


Рис. П1.15

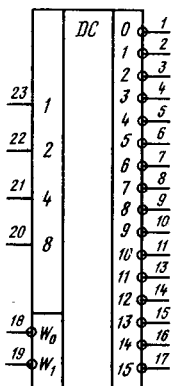


Рис. П1.16

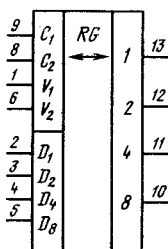


Рис. П1.17

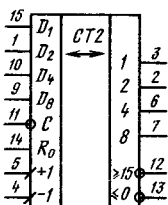


Рис. П1.18

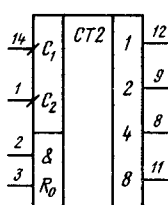


Рис. П1.19

Приложение 2. СВЕДЕНИЯ О БИС ЗАПОМИНАЮЩИХ УСТРОЙСТВ

Таблица П2.1. Статические оперативные запоминающие устройства, совместимые по входу и выходу с ТТЛ-схемами

Тип БИС	Технология	Организация слов × разрядов	Время выборки адреса, нс, не более	Выход
K132PY3A	n-МОП	1024 × 1	75	Три состояния
KP132PY4A	n-МОП	1024 × 1	33	Три состояния
K541PY2A	И ² Л	1024 × 4	90	Три состояния
KM132PY8A	n-МОП	1024 × 4	60	Три состояния
K54PY1A	И ² Л	4096 × 1	70	Три состояния
K541PY31A	И ² Л	8192 × 1	100	Три состояния
K541PY3A	И ² Л	16384 × 1	100	Три состояния

Таблица П2.2. Программируемые постоянные запоминающие устройства, совместимые по входу и выходу с ТТЛ-схемами

Тип БИС	Технология	Организация слов × разрядов	Время выборки адреса, нс, не более	Выход
KP556PT12	ТТЛШ	1024 × 4	60	Открытый коллектор
KP556PT13	ТТЛШ	1024 × 4	60	Три состояния
KP556PT14	ТТЛШ	2048 × 4	60	Открытый коллектор
KP556PT15	ТТЛШ	2048 × 4	60	Три состояния
KP556PT18	ТТЛШ	2048 × 8	60	То же
KP556PT16	ТТЛШ	8192 × 8	85	—>—

Приложение 3. СИСТЕМА КОМАНД МП KP580BM80

Система команд данного МП содержит команды всех основных групп — пересылки, арифметические, логические, команды управления, оперативные, связи с подпрограммами и команды ввода-вывода.

Для описания системы команд, приведенной в табл. П3.5—П3.10, используются следующие условные обозначения: DDD и SSS — код одного из рабочих регистров ЦПЭ в формате команды, причем D — регистр-приемник; S — регистр-источник (коды регистров приведены в табл. П3.1); M — ячейка памяти, адрес которой указан в регистровой паре HL; gr — регистровая пара (B, D, H и UC); RP — код регистровой пары в формате команды, имеющий значения, приведенные в табл. П3.2; gh — регистр пары, в котором содержатся старшие разряды; gi — регистр пары, в котором содержатся младшие разряды; <B₂> и <B₃> — второй и третий байты команды соответственно; [() ()] — содержимое ячейки памяти, адрес которой указан в скобках.

Таблица П3.1. Коды регистров

Регистр ЦПЭ	A	B	C	D	E	H	L	M
Код регистра	111	000	001	010	011	100	101	110

Таблица П3.2. Коды регистровых пар

Регистровая пара	BC	DE	HL	УС
RP	00	01	10	11

Для записи команд при программировании используются мнемокоды; операнды записываются, как правило, в 16-ричном коде. Обозначения цифр и чисел от 0 до 15 в различных системах счисления приведены в табл. П3.3.

Таблица П3.3. Системы счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Обозначения флагов и коды условий приведены в табл. П3.4.
Команды МП КР580ВМ80 приведены в табл. П3.5—П3.10.

Таблица П3.4. Коды условий и значения флагов

Условие	(CCC)—код условия в формате команды	Условие	(CCC)—код условия в формате команды
NZ—ненулевой результат текущей операции, (Z)=0	000	PO—нечетность числа единиц результата, (P)=0	100
Z—нулевой результат текущей операции, (Z)=1	001	PE—четность числа единиц результата, (P)=1	101
NC—отсутствие переноса, (CY)=0	010	P—результат положительный, (S)=0	110
C—наличие переноса, (CY)=1	011	M—результат отрицательный, (S)=1	111

Таблица П3.5. Команды пересылки, загрузки и хранения

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
MOV r_1, r_2	Пересылка из регистра в регистр	$(r_1) \leftarrow (r_2)$	01DDSSSS	5	Не формируются
MOV r, M	Пересылка из памяти в регистр	$(r) \leftarrow [(H)(L)]$	01DDD110	7	—»—
MOV M, r	Пересылка из регистра в память	$[(H)(L)] \leftarrow (r)$	01110SSS	7	—»—
MVI r	Загрузка регистра константой	$(r) \leftarrow \langle B_2 \rangle$	00DDD110 $\langle B_2 \rangle$	7	—»—
MVI M	Загрузка памяти константой	$[(H)(L)] \leftarrow \langle B_2 \rangle$	001100110 $\langle B_2 \rangle$	10	—»—
LXI gr	Загрузка пары регистров	$(rh) \leftarrow \langle B_3 \rangle,$ $(rl) \leftarrow \langle B_2 \rangle$	00RP0001 $\langle B_2 \rangle$ $\langle B_3 \rangle$	10	—»—
LDA	Загрузка A прямая	$(A) \leftarrow \langle B_3 \rangle \langle B_2 \rangle$	00111010 $\langle B_2 \rangle$ $\langle B_3 \rangle$	13	—»—
STA	Загрузка памяти прямая	$\langle B_3 \rangle \langle B_2 \rangle \leftarrow (A)$	00110010 $\langle B_2 \rangle$ $\langle B_3 \rangle$	13	—»—
LHLD	Загрузка регистров H и L содержимым двух ячеек памяти	$(L) \leftarrow \langle B_3 \rangle \langle B_2 \rangle,$ $(H) \leftarrow \langle B_3 \rangle \langle B_2 \rangle + 1$	00101010 $\langle B_2 \rangle$ $\langle B_3 \rangle$	16	—»—
SHLD	Запоминание содержимого регистров HL по адресу	$\langle B_3 \rangle \langle B_2 \rangle \leftarrow (L),$ $\langle B_3 \rangle \langle B_2 \rangle + 1 \leftarrow (H)$	00100010 $\langle B_2 \rangle$ $\langle B_3 \rangle$	16	—»—
LDAX gr (LDAX B, LDAX D)	Косвенная загрузка A	$(A) \leftarrow [(gr)]$ только для $gr = B, gr = D$	00RP1010	7	—»—
STAX gr (STAX B, STAX D)	Запоминание содержимого A по адресу (gr)	$[(gr)] \leftarrow (A)$ только для $gr = B, gr = D$	00RP0010	7	—»—
XCHG	Обмен между регистрами HL и DE	$(H) \leftrightarrow (D), (L) \leftrightarrow (E)$	11101011	4	—»—

Таблица ПЗ.6. Арифметические, инкрементные и декрементные команды

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
ADD r	Сложение содержимого регистра с A	$(A) \leftarrow (A) + r$	1000SSS	4	Формируются все флаги
ADD M	Сложение содержимого памяти с A	$(A) \leftarrow (A) + [(H)(L)]$	10000110	7	То же
ADI	Сложение константы с A	$(A) \leftarrow (A) + \langle B_2 \rangle$	11000110 $\langle B_2 \rangle$	7	—»—
ADC r	Сложение содержимого регистра с A с переносом	$(A) \leftarrow (A) + (r) + (CY)$	10001SSS	4	—»—
ADO M	Сложение содержимого памяти с A переносом	$(A) \leftarrow (A) + [(H)(L)] + (CY)$	10001110	7	—»—
INR r	Инкремент регистра	$(r) \leftarrow (r) + 1$	000DDD100	5	Z, S, P, CY ₁
INR M	Инкремент ячейки памяти	$[(H)(L)] \leftarrow [(H)(L)] + 1$	00110100	10	Z, S, P, CY ₁
DCR r	Декремент регистра	$(r) \leftarrow (r) - 1$	00DDD101	5	Z, S, P, CY ₁
DCR M	Декремент ячейки памяти	$[(H)(L)] \leftarrow [(H)(L)] - 1$	00110101	10	Z, S, P, CY ₁
INX rp	Инкремент пары регистров	$(rh)(rl) \leftarrow (rh)(rl) + 1$	00RP0011	5	Не формируются
DCX rp	Декремент пары регистров	$(rh)(rl) \leftarrow (rh)(rl) - 1$	00RP1011	5	—»—
DAD rp	Двойное сложение пары регистров	$(H)(L) \leftarrow (H)(L) + (rh)(rl)$	00RP1001	10	CY
DAA	Десятичная коррекция содержимого аккумулятора	8-разрядное число в аккумуляторе преобразуется в двоично-десятичную форму по следующему правилу:	00100111	4	Формируются все флаги

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
		1) если содержимое четырех младших разрядов аккумулятора больше 9 или если $(CY_1)=1$, то к аккумулятору добавляется число 6; 2) если содержимое четырех старших разрядов стало после этого больше 9 или если $(CY)=1$, то число 6 добавляется и к содержимому четырех старших разрядов аккумулятора			
ACI	Сложение константы с A с переносом	$(A) \leftarrow (A) + \langle B_2 \rangle + (CY)$	11001110 $\langle B_2 \rangle$	7	—»—
SUB r	Вычитание содержимого регистра из A	$(A) \leftarrow (A) - (r)$	10010SSS	4	—»—
SUB M	Вычитание содержимого памяти из A	$(A) \leftarrow (A) - [(H)(L)]$	10010110	7	—»—
SUI	Вычитание константы из A	$(A) \leftarrow (A) - \langle B_2 \rangle$	11010110 $\langle B_2 \rangle$	7	—»—
SBB r	Вычитание регистра с заемом	$(A) \leftarrow (A) - (r) - (CY)$	10011SSS	4	—»—
SBB M	Вычитание памяти с заемом	$(A) \leftarrow (A) - [(H)(L)] - (CY)$	10011110	7	»
SBI	Вычитание константы с заемом	$(A) \leftarrow (A) - \langle B_2 \rangle - (CY)$	11011110 $\langle B_2 \rangle$	7	»

Таблица ПЗ.7. Логические команды

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
ANA r	Логическое умножение регистра с A	$(A) \leftarrow (A) \wedge (r),$ $(CY) \leftarrow 0; (CY_1) \leftarrow 0$	10100SSS	4	Формируются все флаги То же
ANA M	Логическое умножение ячейки памяти с A	$(A) \leftarrow (A) \wedge [(H)(L)],$ $(CY) \leftarrow 0; (CY_1) \leftarrow 0$	10100110	7	
ANI	Логическое умножение константы с A	$(A) \leftarrow (A) \wedge \langle B_2 \rangle,$ $(CY) \leftarrow 0; (CY_1) \leftarrow 0$	11100110 $\langle B_2 \rangle$	7	
XRA r	Отрицание равнозначности с регистром	$(A) \leftarrow (A) \oplus (r),$ $(CY) \leftarrow 0; (CY_1) \leftarrow 0$	10101SSS	4	»
XRA M	Отрицание равнозначности с памятью	$(A) \leftarrow (A) \oplus [(H)(L)],$ $(CY) \leftarrow 0; (CY_1) \leftarrow 0$	10101110	7	»
XRI	Отрицание равнозначности с константой	$(A) \leftarrow (A) \oplus \langle B_2 \rangle,$ $(CY) \leftarrow 0; (CY_1) \leftarrow 0$	11101110 $\langle B_2 \rangle$	7	»
ORA r	Логическое сложение регистра с A	$(A) \leftarrow (A) \vee (r),$	10110SSS	4	»
ORA M	Логическое сложение ячейки памяти с A	$(A) \leftarrow (A) \vee [(H)(L)],$ $(CY) \leftarrow 0; (CY_1) \leftarrow 0$	10110110	7	То же
ORI	Логическое сложение константы с A	$(A) \leftarrow (A) \vee \langle B_2 \rangle;$ $(CY) \leftarrow 0; (CY_1) \leftarrow 0$	11110110 $\langle B_2 \rangle$	7	»
CMP r	Сравнение A с регистром	$(A) - (r), (A)$ не изменяется; $(Z) = 1$, если $(A) = (r);$ $(CY) = 1$, если $(A) < (r)$	10111SSS	4	»
CMP M	Сравнение A с ячейкой памяти	$(A) - [(H)(L)], (A)$ не изменяется; $(Z) = 1$, если $(A) = [(H)(L)];$ $(CY) = 1$, если $(A) < [(H)(L)]$	10111110	7	»
CPI	Сравнение A с константой	$(A) - \langle B_2 \rangle, (A)$ не изменяется; $(Z) = 1$, если $(A) = \langle B_2 \rangle;$ $(CY) = 1$, если $(A) < \langle B_2 \rangle$	11111110 $\langle B_2 \rangle$	7	»

Таблица П3.8. Оперативные команды

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
RLC	Сдвиг А влево циклический	$(A_{n+1}) \leftarrow (A_n),$ $(A_0) \leftarrow (A_7),$ $(CY) \leftarrow (A_7)$	00000111	4	CY
RRC	Сдвиг А вправо циклический	$(A_n) \leftarrow (A_{n+1}),$ $(A_7) \leftarrow (A_0),$ $(CY) \leftarrow (A_0)$	00001111	4	CY
RAL	Сдвиг А влево с переносом	$(A_{n+1}) \leftarrow (A_n),$ $(CY) \leftarrow (A_7),$ $(A_0) \leftarrow (CY)$	00010111	4	CY
RAR	Сдвиг А вправо с переносом	$(A_n) \leftarrow (A_{n+1}),$ $(CY) \leftarrow (A_0),$ $(A_7) \leftarrow (CY)$	00011111	4	CY
CMA	Инвертирование аккумулятора	$(A) \leftarrow (\bar{A})$	00101111	4	Не формируются
CMC	Инвертирование регистра переноса	$(CY) \leftarrow (\overline{CY})$	00111111	4	CY
STC	Установка регистра переноса	$(CY) \leftarrow 1$	00110111	4	CY

Таблица П3.9. Команды управления в связи с подпрограммами

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
JMP	Безусловный переход	$(CK) \leftarrow \langle B_3 \rangle \langle B_2 \rangle$	11000011 $\langle B_2 \rangle$ $\langle B_3 \rangle$	10	Не формируются
J (условие)	Условный переход	Если условие (CCC) ¹ выполняется, то $(CK) \leftarrow \langle B_3 \rangle \langle B_2 \rangle$	11CCCC010 $\langle B_2 \rangle$ $\langle B_3 \rangle$	10	То же
CALL	Вызов подпрограммы безусловный	$[(YC) - 1] \leftarrow (CK)_H,$ $[(YC) - 2] \leftarrow (CK)_L,$ $(YC) \leftarrow (YC) - 2,$ $(CK) \leftarrow \langle B_3 \rangle \langle B_2 \rangle$	11001101 $\langle B_2 \rangle$ $\langle B_3 \rangle$	17	Не формируются
C (условие)	Вызов подпрограммы условный	Если условие (CCC) ¹ выполняется, то $[(YC) - 1] \leftarrow (CK)_H,$ $[(YC) - 2] \leftarrow (CK)_L,$ $(YC) \leftarrow (YC) - 2,$ $(CK) \leftarrow \langle B_3 \rangle \langle B_2 \rangle$	11CCCC100 $\langle B_2 \rangle$ $\langle B_3 \rangle$	11/17	То же

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
RET	Возврат из подпрограммы безусловный	$(CK)_L \rightarrow [(YC)],$ $(CK)_H \leftarrow [(YC) + 1],$ $(YC) \leftarrow (YC) + 2$	11001001	10	Не формируются
R (условие)	Возврат из подпрограммы условный	Если условие (CCC) ¹ , выполняется, то $(CK)_L \leftarrow [(YC)],$ $(CK)_H \leftarrow [(YC) + 1],$ $(YC) \leftarrow (YC) + 2$	11CCCC000	5/11	»
RST N	Вызов подпрограммы прерываний	$[(YC) - 1] \leftarrow (CK)_H,$ $[(YC) - 2] \leftarrow (CK)_L,$ $(YC) \leftarrow (YC) - 2,$ $(CK) \leftarrow 8 \cdot (AAA)$	11AAA111 Состояние СК после выполнения RST N 0000000000AAA000 └─ номер N прерывающей программы	11	»
PCHL	Загрузка счетчика команд	$(CK)_H \leftarrow (H),$ $(CK)_L \leftarrow (L)$	11101001	5	»

¹ См. табл. ПЗ.4.

Таблица ПЗ.10. Команды управления, ввода-вывода и операции со стеком

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
PUSH гр	Проталкивание в стек содержимого регистровой пары (кроме гр = YC)	$[(YC) - 1] \leftarrow (rh),$ $[(YC) - 2] \leftarrow (rl),$ $(YC) \leftarrow (YC) - 2$	11RP0101	11	Не формируются
PUSH PSW	Проталкивание в стек слова состояния ЦПЭ	$[(YC) - 1] \leftarrow (A),$ $[(YC) - 2] \leftarrow (CY),$ $[(YC) - 2]_1 \leftarrow 1,$ $[(YC) - 2]_2 \leftarrow (P),$ $[(YC) - 2]_3 \leftarrow 0,$ $[(YC) - 2]_4 \leftarrow (CY_1),$ $[(YC) - 2]_5 \leftarrow 0,$ $[(YC) - 2]_6 \leftarrow (Z),$ $[(YC) - 2]_7 \leftarrow (S),$ $(YC) \leftarrow (YC) - 2$	11110101 Слово состояния ЦПЭ: $D_0 = CY_1; D_1 = 1;$ $D_2 = P; D_3 = 0;$ $D_4 = CY_1; D_5 = 0;$ $D_6 = Z; D_7 = S$	11	То же
POP гр	Вытаскивание	$(rl) \leftarrow [(YC)],$	11RP0001	10	Не

Мнемоническое обозначение	Название	Выполняемые операции	Формат команды	Число тактов	Формирование флагов
	ние из стека содержит регистровую пару (кроме $gr = UC$)	$(rh) \leftarrow [(YC) + 1]$, $(YC) \leftarrow (YC) + 2$			формируются
POP PSW	Выталкивание из стека слова состояния ЦПЭ	$(CY) \leftarrow [(YC)]_0$, $(P) \leftarrow [(YC)]_2$, $(CY_1) \leftarrow [(YC)]_4$, $(Z) \leftarrow [(YC)]_6$, $(S) \leftarrow [(YC)]_7$, $(A) \leftarrow [(YC) + 1]$ $(YC) \leftarrow (YC) + 2$	11110001	10	S, Z, CY, CY ₁ , P
XTHL	Обмен между вершиной стека и парой HL	$(L) \leftrightarrow (YC)$, $(H) \leftrightarrow [(YC) + 1]$	11100011	18	Не формируются
SPHL	Загрузка указателя стека	$(YC) \leftarrow (H)(L)$	11111001	5	То же
IN BY_i	Ввод	$(A) \leftarrow (BY_i)$	11011011 Адрес BY_i	10	»
OUT BY_i	Вывод	$(BY_i) \leftarrow (A)$	11010011 Адрес BY_i	10	»
EI	Разрешение прерываний	Разрешается прерывание после выполнения следующей команды	11111011	4	»
DI	Запрет прерываний	Прерывание запрещается после выполнения команды DI	11110011	4	»
HLT	Останов	Процессор останавливается	01110110	7	»
NOP	Пустая операция	Никакие операции не выполняются	00000000	4	»

Приложение 4. НЕКОТОРЫЕ КОМАНДЫ TMS 32010

Т а б л и ц а П4.1

Мнемо-код	Описание	Число слов	Число тактов	Пример	Примечание к примеру
ABS	Взятие абсолютной величины содержимого аккумулятора	1	1	ABS	—
ADD	Сложение с аккумулятором содержимого ячейки памяти данных со сдвигом	1	1	ADD 18,3 ADD *—,4,1	$ACC \leftarrow (ACC) + [18] \times 2^3$ $ACC \leftarrow (ACC) + [(AR)] \times \times 2^4$, косвенная адресация
APAC	Сложение содержимого регистра Р с содержимым аккумулятора	1	1	APAC	$ACC \leftarrow (P) + (ACC)$
B	Безусловный переход по адресу, указанному во втором слове команды	2	2	B 200	Переход к команде, записанной в ячейке 200 программной памяти ($PC \leftarrow 200$)
BANZ	Переход, если содержимое текущего вспомогательного регистра отлично от 0	2	2	BANZ 100	Если $(AR) \neq 0$, то $AR \leftarrow (AR) - 1$ и $PC \leftarrow 100$, иначе $PC \leftarrow (PC) + 1$ и $AR \leftarrow (AR) - 1$
BIOZ	Переход, если сигнал на входе BIO равен нулю	2	2	BIOZ 10	Если $BIO = 0$, то $PC \leftarrow 10$, иначе $PC \leftarrow (PC) + 1$
BGEZ	Переход, если содержимое аккумулятора больше или равно нулю	2	2	BGEZ 300	Если $(ACC) \geq 0$, то $PC \leftarrow 300$, иначе $PC \leftarrow (PC) + 1$
BLZ	Переход, если содержимое аккумулятора меньше нуля	2	2	BLZ 50	Если $(ACC) < 0$, то $PC \leftarrow 50$, иначе $PC \leftarrow (PC) + 1$
BZ	Переход, если содержимое аккумулятора равно нулю	2	2	BZ 100	Если $(ACC) = 0$, то $PC \leftarrow 100$, иначе $PC \leftarrow (PC) + 1$
CALA	Переход на подпрограмму по адресу из аккумулятора	1	2	CALA	Вершина стека $\leftarrow (PC) + 1$, $PC \leftarrow$ Разряды 11—0 ACC
CALL	Непосредственный вызов подпрограммы	2	2	CALL 400	Вершина стека $\leftarrow (PC) + 1$, $PC \leftarrow 400$

Мнемо-код	Описание	Число слов	Число тактов	Пример	Примечание к примеру
DINT	Запрещение прерываний	1	1	DINT	Бит режима прерываний $\leftarrow \langle 1 \rangle$
EINT	Разрешение прерываний	1	1	EINT	Бит режима прерываний $\leftarrow \langle 0 \rangle$
DMOV	Содержимое заданной ячейки памяти пересылается в следующую ячейку с большим адресом	1	1	DMOV 20 DMOV * +, 0	$[21] \leftarrow [20]$, $[(AR) + 1] \leftarrow [(AR)]$, косвенная адресация
IN	Ввод данных из порта	1	2	IN 30, 7 IN * -, 7, 0	$[30] \leftarrow (\text{Порт } 7)$, $[(AR)] \leftarrow (\text{Порт } 7)$, косвенная адресация
LAC	Содержимое ячейки памяти со сдвигом влево загружается в аккумулятор	1	1	LAC 21,5 LAC *, 3, 1	$ACC \leftarrow [21] \times 2^5$, $ACC \leftarrow [(AR)] \times 2^3$, косвенная адресация
LACK	Загрузка аккумулятора непосредственной восьмизначной положительной константой	1	1	LACK 95	Младшие разряды $ACC \leftarrow 95$, старшие 24 разряда ACC обнуляются
LAR	Содержимое ячейки памяти загружается в указанный вспомогательный регистр	1	1	LAR AR0, 22 LAR ARi * -, 0	$AR0 \leftarrow [22]$, $AR1 \leftarrow [(AR)]$, косвенная адресация
LARK	Загрузка указанного в команде вспомогательного регистра непосредственной 8-разрядной положительной константой	1	1	LARK AR0 77, LARK AR1,5	Младшие разряды $AR0 \leftarrow 77$, старшие разряды обнуляются, младшие разряды $AR1 \leftarrow 5$
LARP	Непосредственная загрузка указателя вспомогательного регистра	1	1	LARP 0 LARP 1	$ARP \leftarrow \langle 0 \rangle$, $ARP \leftarrow \langle 1 \rangle$
LDPK	Непосредственная загрузка указателя страниц	1	1	LDPK 1	$DP \leftarrow \langle 1 \rangle$

Мнемо-код	Описание	Число слов	Число тактов	Пример	Примечание к примеру
LT	Загрузка регистра Т	1	1	LT 12 LT *, 1	Регистр $T \leftarrow [12]$, регистр $T \leftarrow [(AR)]$, косвенная адресация
LTA	Команда объединяет в себе команды LT и APAC	1	1	LTA 25	Регистр $T \leftarrow [25]$; $ACC \leftarrow (ACC) + (P)$
LTD	Выполнение этой команды аналогично одновременному выполнению команд LT, APAC и DMOV	1	1	LTD 21 LTD * -, 1	$T \leftarrow [21]$; $ACC \leftarrow (ACC) + (P)$; $[22] \leftarrow [21]$, косвенная адресация
MAR	Модификация содержимого вспомогательного регистра	1	1	MAR * +, 1	В режиме прямой адресации равносильна команде NOP; $AR \leftarrow (AR) + 1$; $ARP \leftarrow \langle 1 \rangle$
MPY	Содержимое ячейки памяти умножается на содержимое регистра Т	1	1	MPY 10 MPY * +, 0	$P \leftarrow [10] \times (T)$, $P \leftarrow [(AR)] \times (T)$, косвенная адресация
MPYK	Содержимое регистра Т умножается на 13-разрядную константу со знаком, указанную в команде	1	1	MRYK -, 9	$P \leftarrow (T) \times (-9)$
NOP	Нет операции	1	1	NOP	
OUT	Вывод данных в порт	1	2	OUT 7, 3 OUT * -, 4, 0	Порт $3 \leftarrow [7]$, порт $4 \leftarrow [(AR)]$, косвенная адресация
PAC	Загрузка аккумулятора содержимым регистра Р	1	1	PAC	$ACC \leftarrow (P)$
RET	Возврат из подпрограммы	1	2	RET	$PC \leftarrow (\text{Вершина стека})$
SACH	Запись в память старших 16 разрядов аккумулятора со сдвигом влево на 0,1 или 4 разряда	1	1	SACH 12, 4	Сначала содержимое ACC сдвигается влево на 4 разряда, затем старшие 16 разрядов записываются в ячейку памяти 12
SACL	Запись в память младших 16 раз-	1	1	SACL 13 SACL * -, 1	$[13] \leftarrow (ACC, \text{разряды } 15...0)$, $[(AR)] \leftarrow (ACC,$

Мнемо-код	Описание	Число слов	Число тактов	Пример	Примечание к примеру
SAR	рядов аккумулятора Сохранение содержимого, указанного в команде вспомогательного регистра	1	1	SAR AR0, 15 SAR AR1, *, +, 0	разряды 15...0), косвенная адресация [15]←(AR0). [(AR)]←(AR1), косвенная адресация
SUB	Вычитание из аккумулятора содержимого ячейки памяти	1	1	SUB 25, 7 SUB *, 9, 0	$ACC \leftarrow (ACC) - [25] \times 2^7$, $ACC \leftarrow (ACC) - [(AR)] \times 2^9$, косвенная адресация
TBLR	Пересылка слова из программной памяти в память данных. Адрес программной памяти определяется содержимым аккумулятора	1	3	TBLR 28	1. Вершина стека←←(PC)+1. 2. PC←(ACC), [28]←←(шина данных). 3. PC←(Вершина стека)
TBLW	Пересылка слова из памяти данных в программную память	1	3	TBLW 17 TBLW *, -, 1	Косвенная адресация
ZAC	Обнуление аккумулятора	1	1	ZAC	$ACC \leftarrow 0$

СПИСОК ЛИТЕРАТУРЫ

1. **Микропроцессоры.** Ч. 1—8: Учеб. пособие для вузов/П. В. Нестеров, В. Ф. Шаньгин, А. Е. Костин и др.; Под ред. Л. Н. Преснухина.—М.: Высшая школа, 1984—1985.
2. **Вычислительная** и микропроцессорная техника: Учебник для вузов/Э. В. Евреинов, Ю. Т. Бутыльский, И. А. Мамзелев и др.; Под ред. Э. В. Евреинова.—М.: Радио и связь, 1991.—464 с.
3. **Гольденберг Л. М., Бутыльский Ю. Т., Поляк М. Н.** Цифровые устройства на интегральных схемах в технике связи.—М.: Радио и связь, 1979.—232 с.
4. **Гольденберг Л. М.** Импульсные устройства.—Учебник для вузов.—М.: Радио и связь, 1981.—224 с.
5. **Калабеков Б. А.** Микропроцессоры и их применение в системах передачи и обработки сигналов: Учеб. пособие для вузов.—М.: Радио и связь, 1988.—368 с.
6. **Каган Б. М., Сташин В. В.** Основы проектирования микропроцессорных устройств автоматики.—М.: Энергоатомиздат, 1987.—304 с.
7. **Лю Ю.—Чжэн, Гибсон Г.** Микропроцессоры семейства 8086/8088. Архитектура, программирование и проектирование микрокомпьютерных систем: Пер. с англ.—М.: Радио и связь, 1987.—512 с.
8. **Проектирование** импульсных и цифровых устройств радиотехнических систем: Учеб. пособие для вузов/Ю. П. Гришин, Ю. М. Казаринов, В. М. Катиков и др.; Под ред. Ю. М. Казаринова.—М.: Высшая школа, 1985.—320 с.
9. **Гольденберг Л. М., Матюшкин Б. Д., Поляк М. Н.** Цифровая обработка сигналов: Справочник.—М.: Радио и связь, 1985.—312 с.
10. **Клингман Э.** Проектирование микропроцессорных систем: Пер. с англ.—М.: Мир, 1985.
11. **Алексенко А. Г., Галицын А. А., Иванников А. Д.** Проектирование радиоэлектронной аппаратуры на микропроцессорах: Программирование, типовые решения, методы отладки.—М.: Радио и связь, 1984.—272 с.
12. **Применение** интегральных микросхем в электронной вычислительной технике: Справочник/Р. В. Данилов, И. Ф. Осипов, С. А. Ельцова и др.; Под ред. Б. Н. Файзулаева, Б. В. Тарабрина.—М.: Радио и связь, 1986.
13. **Фридман М., Ивэнс Л.** Проектирование систем с микрокомпьютерами: Пер. с англ.—М.: Мир, 1986.—408 с.
14. **Каган Б. М.** Электронные вычислительные машины и системы: Учеб. пособие для вузов.—М.: Энергоатомиздат, 1985.—542 с.
15. **Баранов С. И., Скляров В. А.** Цифровые устройства на программируемых БИС с матричной структурой.—М.: Радио и связь, 1986.—272 с.
16. **Микропроцессоры** и микропроцессорные комплекты интегральных микросхем: Справочник: В 2-х т. /В-Б. Б. Абрайтис, Н. Н. Аверьянов, А. И. Белоус и др.; Под ред. В. А. Шахнова.—М.: Радио и связь, 1988.—Т. 1 и 2.
17. **Проектирование** цифровых систем на комплектах микропрограммируемых БИС/С. С. Булгаков, В. И. Мещеряков, Л. А. Шумилов и др.—М.: Радио и связь, 1984.—240 с.
18. **Микропроцессоры.** Сборник примеров и задач/П. В. Нестеров, В. Ф. Шаньгин, А. Е. Костин и др.; Под ред. Л. Н. Преснухина.—М.: Высшая школа, 1985.—96 с.

19. **Расчет и проектирование импульсных устройств:** Учеб. пособие для вузов/В. А. Малев, М. Н. Поляк, Э. А. Крогиус и др.; Под ред. Л. М. Гольденберга.— М.: Связь, 1975.—296 с.
20. **Полупроводниковые БИС запоминающих устройств:** Справочник/В. В. Баранов, Н. В. Бекин, А. Ю. Гордонов и др.; Под ред. А. Ю. Гордонова и Ю. Н. Дьякова.— М.: Радио и связь, 1986.—360 с.
21. **Интегральные микросхемы:** Справочник/Б. В. Тарабрин, Л. Ф. Лунин, Ю. Н. Смирнов и др.; Под ред. Б. В. Тарабрина.— М.: Энергоатомиздат, 1985.—528 с.
22. **Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и мини-ЭВМ:** Учеб. пособие для вузов.— Л.: Энергоатомиздат, 1984.
23. **Соколов П. А. Микропроцессоры и основы построения микроЭВМ:** Учебное пособие.— М.: ВЗЭИС, 1986.
24. **Куньсянь Л., Дж. А. Франц, Р. Саймер, мл. Цифровые процессоры обработки сигналов серии TMS320//ТИИЭР.**—1987.—Т. 75, № 9.—С. 8—27.
25. **Макдоуноу К. и др. Микрокомпьютер с 32-разрядным АУ для вычисления с высокой точностью//Электроника.**—1982.—Т. 55, № 4.—С. 25—32.
26. **Lin K. Digital Signal Processing Applications with the TMS320 Family.**—N. J.: Prentice-Hall, 1987.
27. **First Generation TMS320 User's Guide.**—Houston, TX: Texas Instruments Inc., 1987.
28. **Соловьев Г. Н. Арифметические устройства ЭВМ.**—М.: Энергия, 1978.
29. **Микропроцессоры:** Справочник.—Л.: Судостроение, 1988.
30. **Сташин В. В., Урусов А. В., Мологоницева О. Ф. Проектирование цифровых устройств на однокристалльных микроконтроллерах.**— М.: Энергоатомиздат, 1990.—224 с.
31. **Гуртовцев А. Л., Гудыменко С. В. Программирование для микропроцессоров.**—Минск, Высшая школа, 1989.—352 с.
32. **Цифровой процессор обработки сигналов TMS 32010 и его применение/Под ред. А. А. Ланнэ.**—Л.: ВАС, 1990.—296 с.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава 1. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ И АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ В ЦИФРОВЫХ УСТРОЙСТВАХ	5
1.1. Позиционные системы счисления	5
1.2. Представление чисел с фиксированной и плавающей точкой	7
1.3. Кодирование чисел и арифметические операции	8
Глава 2. КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА	17
2.1. Комбинационные цифровые устройства на интегральных логических микросхемах	17
2.2. Цифровые устройства на программируемых логических матрицах	21
Глава 3. ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА	34
3.1. Триггеры	34
3.2. Функциональные узлы последовательностного типа	37
3.3. Анализ и синтез последовательностных устройств	42
Глава 4. ФОРМИРОВАТЕЛИ, ГЕНЕРАТОРЫ И СЕЛЕКТОРЫ ИМПУЛЬСОВ	68
4.1. Ограничители	68
4.2. Формирователи импульсов напряжения прямоугольной формы	71
4.3. Генераторы импульсов	75
4.4. Селекторы импульсов	79
Глава 5. ПОЛУПРОВОДНИКОВЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА	92
Глава 6. МИКРОПРОЦЕССОРЫ	99
6.1. Основные понятия и определения	99
6.2. Микропроцессор КР580ВМ80	99
6.3. Микропроцессор К1810ВМ86	106
6.4. Микроконтроллер КМ1816ВЕ48	109
6.5. Микропроцессорный комплект К1804	112
6.6. Микропроцессор TMS 32010	123
Глава 7. УЗЛЫ МИКРОПРОЦЕССОРНЫХ СИСТЕМ	173
7.1. Шинный формирователь КР580ВА86	173
7.2. Многорежимный буферный регистр К589ИР12	174
7.3. Программируемый периферийный адаптер КР580ВВ55	176
7.4. Программируемый связной адаптер КР580ВВ51	180
7.5. Программируемый интервальный таймер КР580ВИ53	182
Глава 8. МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА И СИСТЕМЫ НА БАЗЕ МП КР580ВМ80	195
Приложение 1. Сведения о микросхемах ТТЛ серий 133, 134, 155 и 531	239
Приложение 2. Сведения о БИС запоминающих устройств	242
Приложение 3. Система команд МП КР580ВМ80	242
Приложение 4. Некоторые команды TMS 32010	251
Список литературы	255