

6.1. Что дают обратные связи? 6.2. Синхронизируемые триггеры. 6.3. «Делай, как я!» или о разных триггерах. 6.4. «Прыгай — держись!» 6.5. Регистры, счетчики. 6.6. Шифраторы, дешифраторы, мультиплексоры. 6.7. Как считает ЭВМ? 6.8. Электронная память, или СОЗУ, ОЗУ, ПЗУ, ППЗУ и т. п. 6.9. Магистраль. 6.10. Окно в аналоговый мир, или ЦАП и АЦП. 6.11. Еще раз о кодах.

ЭЛЕКТРОННЫЕ УСТРОЙСТВА

Гораздо быстрее, чем из кирпичей, возводят здания из крупных панелей, блоков. Благодаря искусству архитекторов даже домам из таких типовых строительных деталей свойственна определенная индивидуальность.

Что-то похожее наблюдается и в вычислительной технике. В принципе, можно построить электронные устройства и непосредственно из «кирпичей» типа элементов И-НЕ, но значительно удобнее использовать «детали» более крупные. Каждая такая деталь в целом выполняет вполне определенные функции, что и закреплено за ними название электронных функциональных элементов и устройства. Системы из таких автономных функциональных блоков проще конструировать, налаживать, обслуживать, ремонтировать.

6.1. Что дают обратные связи?

Обратная связь, или воздействие выходного сигнала на вход управления, играет важную роль в радиотехнике, электронике, системах автоматического управления и регулирования (см. 1.4). Устройства с обратной связью лежат в основе построения высококачественных усилителей, измерительных приборов, автоматов, радиолокаторов и т. п.

Так как сигналы, вырабатываемые цифровыми схемами, однотипны с сигналами управления этими схемами, то цепи обратной связи в цифровой электронике предельно просты. Они представляют собой соединения выходов со входами.

В некоторых случаях цифровая обратная связь позволяет как бы «удержать» сигнал на выходе после окончания воздействия

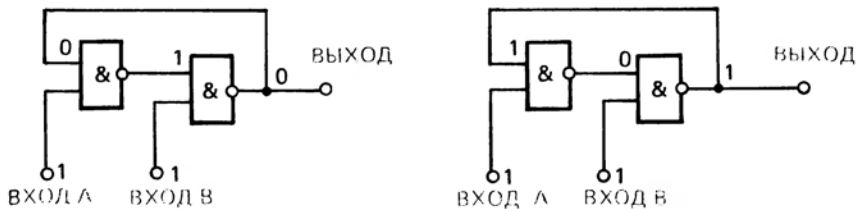


Рис. 6.1. Два последовательно соединенных элемента И-НЕ образуют триггер. В режиме хранения на оба входа *A* и *B* поданы логические 1, но в первом случае триггер хранит 0, во втором — 1. Оба состояния устойчивы.

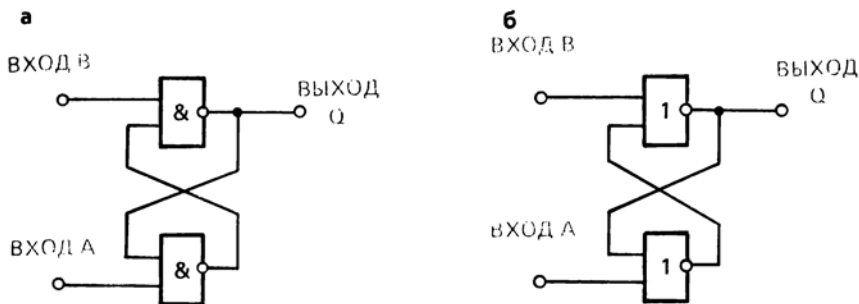


Рис. 6.2. Собственно триггер на схемах И-НЕ (а) и на схемах ИЛИ-НЕ (б). В первом случае для хранения информации на входы *A* и *B* подают логические 1, во втором — логические 0.

входного сигнала, т. е. придает схеме способность сохранять состояние 0 или 1. Схема, способная хранить («запоминать») 0 или 1, называется *триггером*. Триггер — простейший цифровой автомат с памятью (см. 2.3, 2.4).

На рисунке 6.1 показано, как построить триггер из двух схем И-НЕ. Обычно же триггеры изображают так, как показано на рисунке 6.2, и называют собственно триггерами. Собственно триггер — обязательная составная часть любых более сложных триггеров.

Далее будет использоваться только триггер на элементах И-НЕ (рис. 6.2, а). Его свойства поясняет таблица 6.1, в которой исходное состояние триггера обозначено Q_t , а новое — Q_{t+1} .

Таблица 6.1

<i>B</i>	<i>A</i>	Q_{t+1}
0	0	неопред.
0	1	1
1	0	0
1	1	Q_t

<i>B</i> \ <i>A</i>	0	1
0	?	1
1	0	Q_t

Предполагается, что после каждого воздействия набора входных сигналов B и A триггер переводится в режим хранения, т. е. на входы B и A одновременно подаются логические 1.

6.2. Синхронизируемые триггеры

Различают два типа входов триггеров — *информационные* и *синхронизирующие*. Сигналы на информационных входах и исходное состояние Q_i определяют, каким будет новое состояние триггера. Сигналы на синхронизирующих входах определяют время переключения.

Простейший синхронизируемый триггер показан на рисунке 6.3. Он называется *RS-триггером* (от англ. *set, reset* — установить, переустановить). R - и S -входы — информационные. C -вход — синхронизирующий (от англ. *clock* — часы, засекают время). На него постоянно подан 0, поэтому на выходах 1-й и 2-й схем И-НЕ — логические 1, и собственно триггер находится в режиме хранения информации. Если же на вход C на короткое время подать логическую 1 (C -импульс), то собственно триггер установится в состояние, зависящее от сигналов на входах R и S . Все возможные варианты переходов отражены в таблице 6.2.

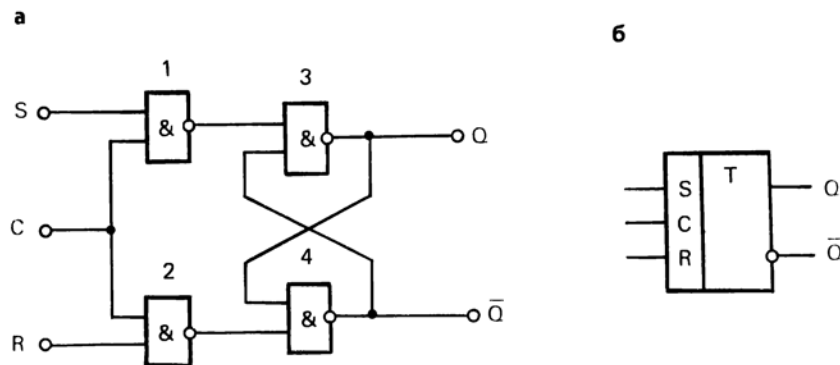
Таблица 6.2

S	R	Q_{i+1}
0	0	Q_i
0	1	0
1	0	1
1	1	неопред.

		R	
		0	1
S	0	Q_i	0
	1	1	?

Состояние триггера при наличии логической единицы на обоих входах не определено, потому что бессмысленно требовать установки и сброса триггера одновременно. Состояние реального триггера при таких входных сигналах зависит от его устройства.

Рис. 6.3. Синхронизируемый RS-триггер: а — структура, б — обозначение.



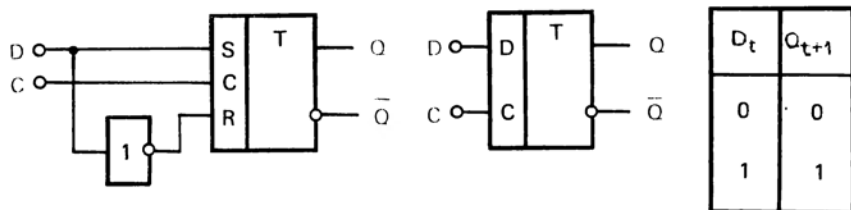


Рис. 6.4. D-триггер: а — структура, б — обозначение, в — таблица состояний.

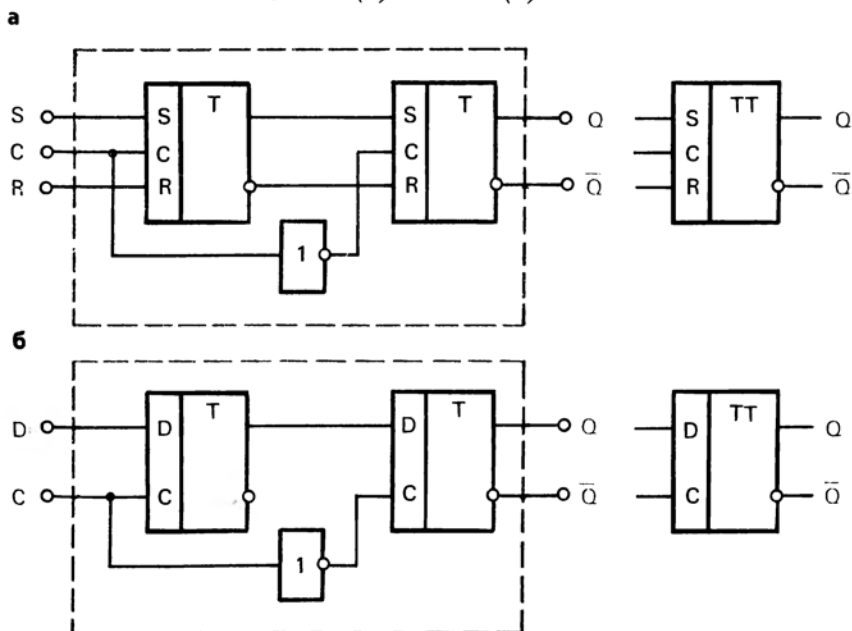
Триггер задержки или *D-триггер* (от англ. *delay* — задержка) имеет только один информационный вход. Его схема приведена на рисунке 6.4. Легко убедиться, что после *C*-импульса он повторяет сигнал на входе *D*.

6.3. «Делай, как я!» или о разных триггерах

Обратная связь помогла превратить комбинационную схему в триггер. А не способна ли она расширить возможности самого триггера?

Попытка ввести обратную связь в рассмотренные синхрони-

Рис. 6.5. Структура и обозначение триггеров с двухступенчатым запоминанием RS-типа (а) и D-типа (б).



зируемые триггеры наталкивается на определенные трудности, так как сигнал обратной связи может измениться до окончания импульса синхронизации и привести к повторному, т. е. незапланированному, переключению триггера.

Чтобы исключить подобного рода нежелательные явления, применяют триггеры с двухступенчатым запоминанием. Их называют также *MS-триггерами* по первым буквам английских слов *master-slave* — хозяин-раб, подчеркивая тем самым, что триггер состоит из двух частей, одна из которых (*master*) как бы заставляет другую (*slave*) повторить свои действия. Более четко суть происходящих процессов отражают термины ведущий-ведомый: ведомый поступает так, как ведущий.

На рисунке 6.5 показана структура *RS*- и *D*-триггеров с двухступенчатым запоминанием. Каждый из них содержит по два обычных синхронизируемых триггера и описывается той же таблицей состояний, что и обыкновенный триггер эквивалентного типа. В отличие от обычных *RS*- и *D*-триггеров сигнал переключения появляется на выходе триггера с двухступенчатым запоминанием после окончания импульса синхронизации: в начале *C*-импульса переключается ведущий триггер, в момент окончания — ведомый.

6.4. «Прыгай-держись!»

Аббревиатура *JK*, входящая в название наиболее универсального триггера, обязана своим происхождением английским словам *Jump-Keep* — прыгай-держись.

Изучение *JK-триггера* начнем с анализа некоторых способов включения уже известных нам *RS*- и *D*-триггеров с двухступенчатым запоминанием. В этих триггерах, как и во всех остальных, наряду с основным обычно предусматривается и инверсный выход \bar{Q} . Он не несет никакой дополнительной информации, но

Рис. 6.6. Организация счетного входа у триггеров с двухступенчатым запоминанием.

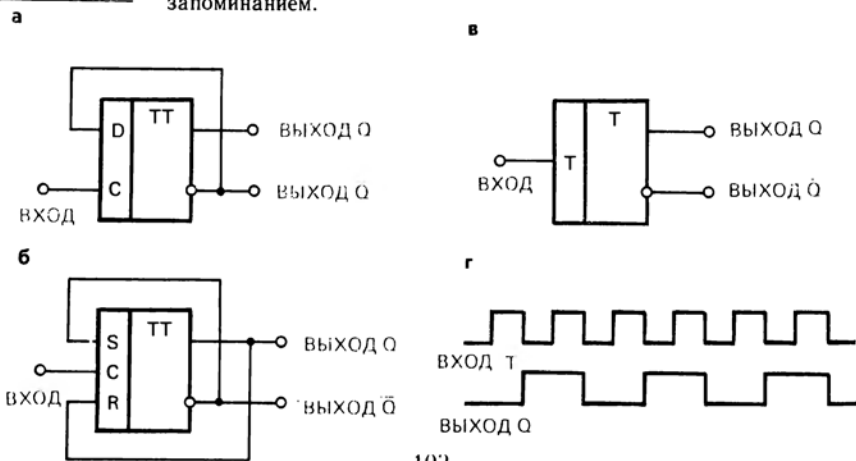
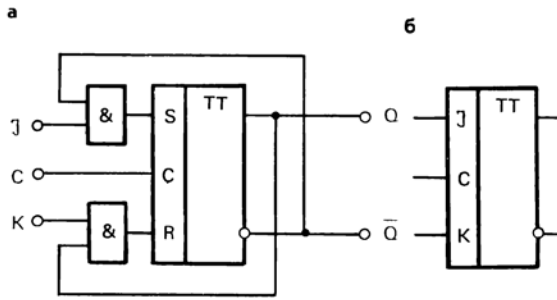


Рис. 6.7. Структура JK-триггера (а) с двухступенчатым запоминанием и его обозначение (б).



в ряде случаев удобен как источник сигнала, противоположного (инверсного) тому, который наблюдается на основном выходе.

В схеме на рисунке 6.6, а инверсный выход D -триггера с двухступенчатым запоминанием соединен с его же информационным входом, и поэтому после C -импульса D -триггер переходит из состояния Q в противоположное состояние \bar{Q} . После второго C -импульса триггер возвращается в исходное состояние. Иными словами, на выходе такого триггера в ответ на два импульса на входе появляется один импульс на выходе (рис. 6.6, з).

Такой режим работы триггера как и образовавшийся при этом T -вход называют *счетным*. На рисунке 6.6, б показано, как можно получить счетный вход в RS -триггере, на рисунке 6.6, в дано обозначение таких триггеров.

Структура JK -триггера (рис. 6.7) напоминает двухступенчатый RS -триггер с обратными связями для получения счетного входа. Однако обратные связи заведены не непосредственно на R - и S -входы, а через схемы И. На другие входы этих схем И подаются соответственно информационные сигналы J и K . Если $J=1$ и $K=1$, то триггер полностью эквивалентен схеме, показанной на рисунке 6.6, б, и работает как счетный. Свойства JK -триггера пояснены таблицей 6.3.

Таблица 6.3

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	\bar{Q}_t

$J \backslash K$	0	1
0	Q_t	0
1	1	\bar{Q}_t

6.5. Регистры, счетчики

Регистр — это цепочка триггеров для запоминания одного двоичного числа. Общее количество триггеров равно наибольшей разрядности хранимого числа. Каждому триггеру может быть поставлен в соответствие весовой коэффициент или просто вес (см. 3.1). Если все триггеры регистра находятся в состоянии 0,

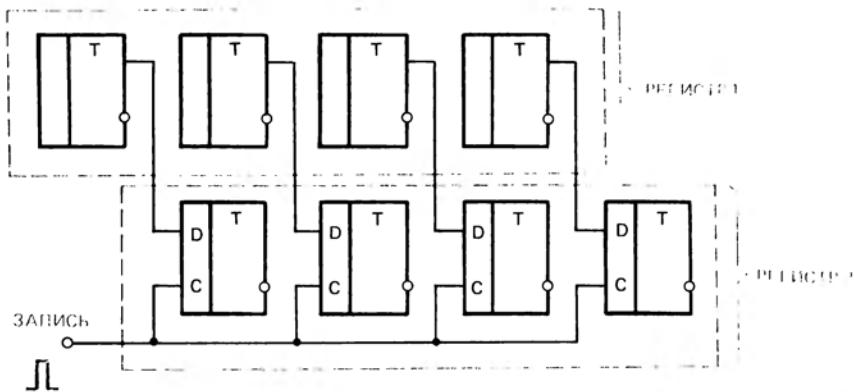


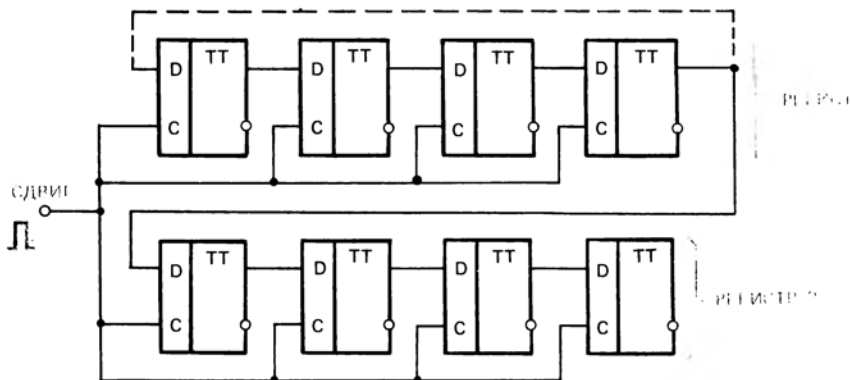
Рис. 6.8. Параллельный способ передачи кода регистра 1 в регистр 2.

а данный триггер — в состоянии 1, то хранимое в регистре число равно весу этого триггера. В чисто двоичном n -разрядном регистре веса триггеров равны $2^0, 2^1, \dots, 2^{n-1}$, т. е. 1, 2, 4, 8, ...

Код числа можно передать из одного регистра в другой параллельным или последовательным способом. Первый способ поясняется рисунком 6.8. Кратковременное появление логической 1 на объединенном C -входе второго регистра заставляет каждый триггер этого регистра устанавливаться в состояние, соответствующее сигналу на D -входе, т. е. в состояние связанного с ним триггера первого регистра.

При последовательном способе передачи (рис. 6.9) на объединенный C -вход первого и второго регистров подают n импульсов. Каждый C -импульс устанавливает данный триггер в состояние соседа слева, поэтому после n импульсов (4 для случая, представленного на рис. 6.9) код первого регистра будет передан во

Рис. 6.9. Последовательный способ передачи кода из регистра 1 в регистр 2.



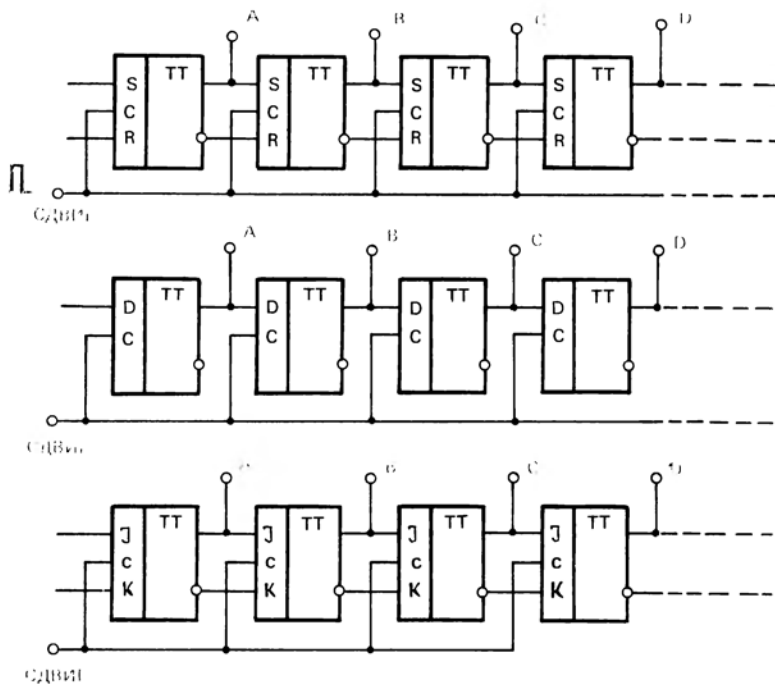


Рис. 6.10 Сдвигающие регистры на RS-, D- и JK-триггерах с двухступенчатым запоминанием.

второй. Когда при этом необходимо сохранить содержимое в первом регистре, то делается соединение, показанное на рисунке 6.9 пунктиром.

В *сдвигающем регистре* (рис. 6.10) при каждом импульсе управления весь код смещается по отношению к цепочке триггеров на одну позицию. Если веса триггеров фиксированы, то сдвиг в одну сторону эквивалентен умножению числа на 2, в другую — делению на 2.

Одна из разновидностей регистров — пересчетные схемы или просто *счетчики*. Каждый импульс на входе суммирующего счетчика (рис. 6.11, а) увеличивает код хранимого в нем числа на 1. В вычитающем счетчике импульсы вычитаются из содержимого счетчика (рис. 6.11, б). *Реверсивный счетчик* (рис. 6.11, в) объединяет в себе свойства и того и другого, имея вход суммирования «+» и вход вычитания «-» (сравни рис. 3.10. и рис. 3.12).

Счетчик из n триггеров имеет 2^n устойчивых состояний и может хранить числа от 0 до $2^n - 1$. Величину $m = 2^n$ называют *коэффициентом пересчета*. Различными способами (например, введением линий обратной связи) некоторые из устойчивых состояний можно запретить. Тогда итоговый коэффициент пересчета станет меньше, чем 2^n . Счетчик, в котором реализуется десять

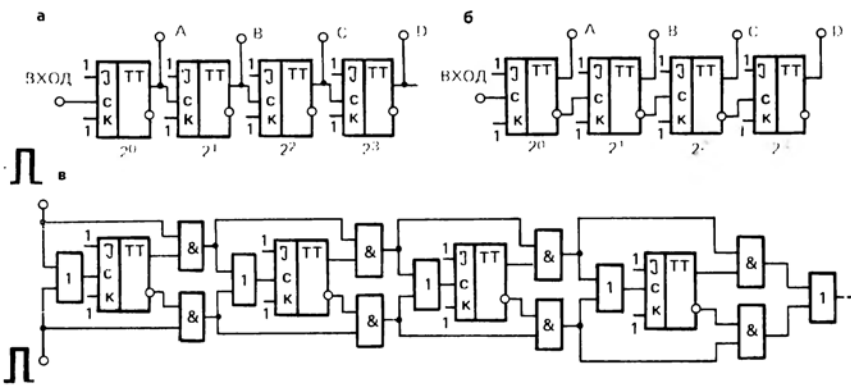


Рис. 6.11.

В простейших счетчиках при поступлении на вход очередного импульса все подлежащие переключению триггеры изменяют свое состояние не одновременно, а последовательно друг за другом, поэтому такие пересчетные схемы называют асинхронными.

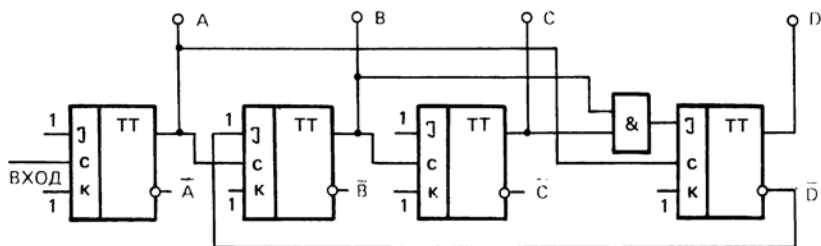


Рис. 6.12.

Суммирующий счетчик с коэффициентом пересчета 10 («декада»).

устойчивых состояний, называют десятичным или *декадой* (рис. 6.12). Несколько включенных друг за другом декад образуют многоразрядный счетчик, работающий в привычной десятичной системе счисления.

Заметим, что сами по себе электронные счетчики широко применяются в физических исследованиях, особенно в ядерной физике, где скорость счета событий нередко исчисляется сотнями миллионов и даже миллиардами в секунду.

6.6. Шифраторы, дешифраторы, мультиплексоры

Установить, в каком из 2^n возможных состояний находится n -разрядный регистр, позволяет *дешифратор* (от фр. *dechiffre* — расшифровывать). Он имеет n входов и 2^n выходов. При любой комбинации сигналов на входах сигнал 1 наблюдается только на одном из выходов, т. е. каждое из 2^n возможных состояний регистра обуславливает появление 1 на «своем» выходе. Выходы принято обозначать таким образом, что индекс при букве y является десятичным аналогом распознаваемого двоичного кода.

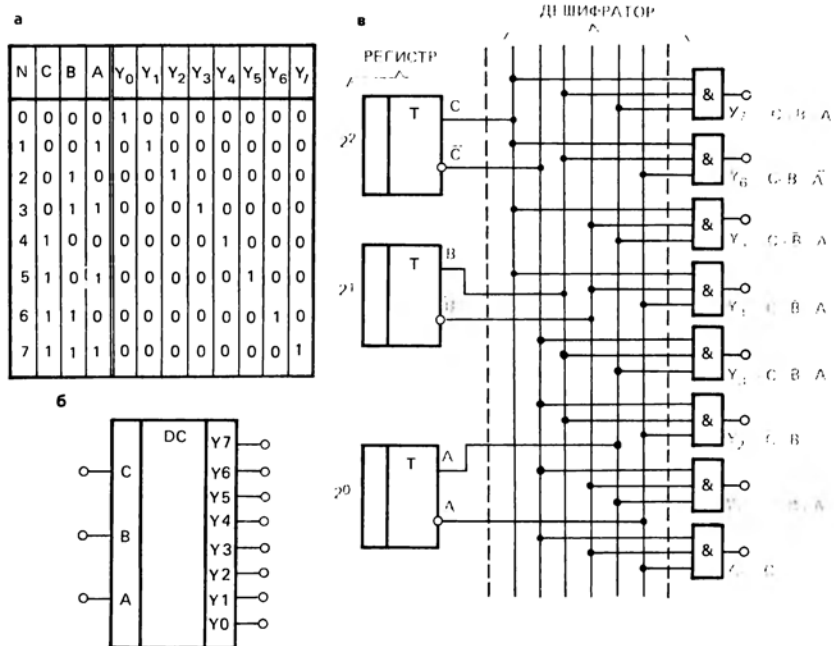


Рис. 6.13. Трехразрядный двоичный дешифратор: а — таблица состояний, б — обозначение, в — структура и принцип подключения к регистру.

В простейшем случае дешифратор представляет собой набор из 2^n n -входовых схем И. На входы каждой из них поступают прямые или инверсные сигналы с выходов всех триггеров регистра (рис. 6.13).

Эта схема может рассматриваться как преобразователь двоичного натурального кода в код «1 из n » (см. 3.1). Обратную функцию производит *шифратор*. Логическая 1 на любой из вертикальных шин (рис. 6.14) преобразуется в параллельный двоичный код. В этом легко убедиться, вспомнив принцип действия схемы ИЛИ. Аналогично осуществляется преобразование и в коды других типов, а комбинация устройств дешифратор-шифратор позволяет преобразовывать код одного типа в другой код. Рисунок 6.15 поясняет, как преобразуется двоичный код в код управления сегментным индикатором.

Слово *мультиплексор* заимствовано из английского и обозначает молоточки для выстукивания пациентов. Звук от ударов по различным точкам тела воспринимается врачом на слух. Таким образом, различные источники информации (точки тела) передаются для анализа через один и тот же канал (ухо врача).

По аналогии использование одних и тех же шин для передачи информации от различных источников называют мультиплексиро-

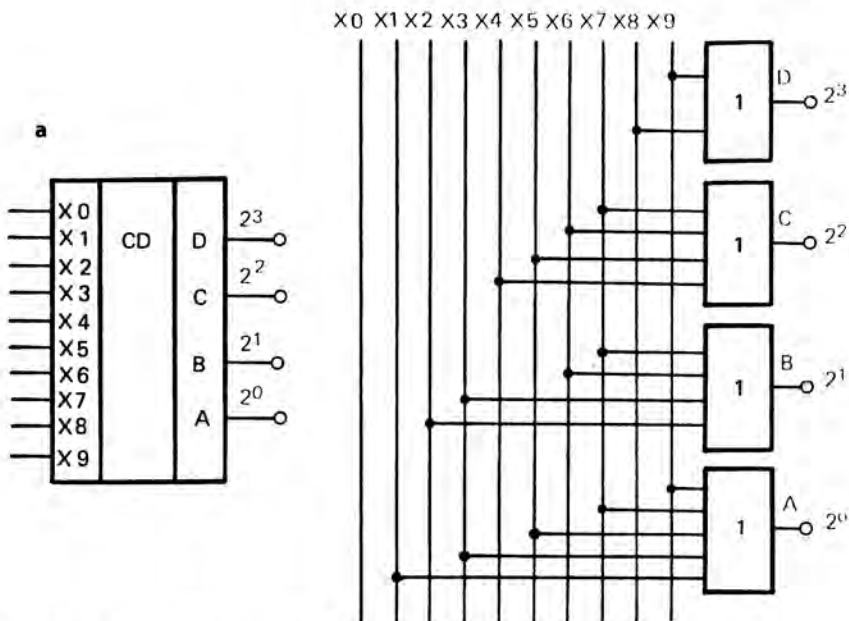
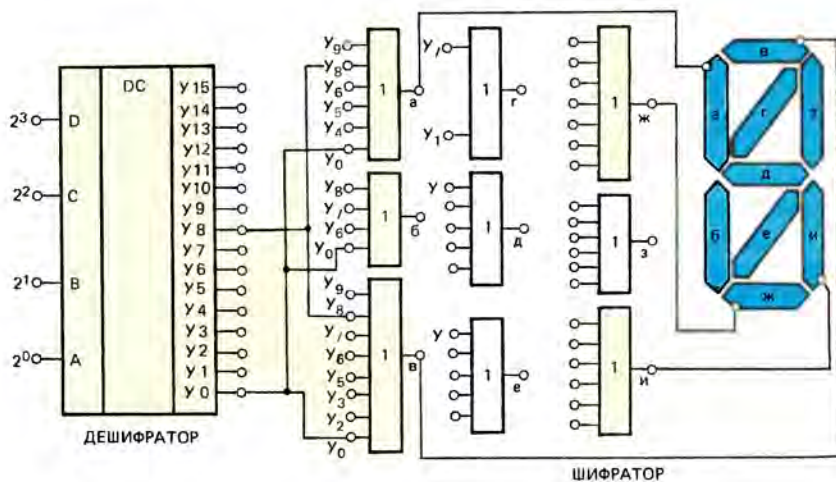


Рис. 6.14. Обозначение (а) и структура (б) двоичного шифратора.

ванием, а устройства сведения информации в одну шину — мультиплексорами. Обратная операция — разделение информации по адресам назначения производится при помощи *демультим*

Рис. 6.15. Принцип преобразования параллельного двоичного кода *DCBA* в код управления сегментным индикатором абвгдежзи (часть соединительных проводов на рисунке не показана).



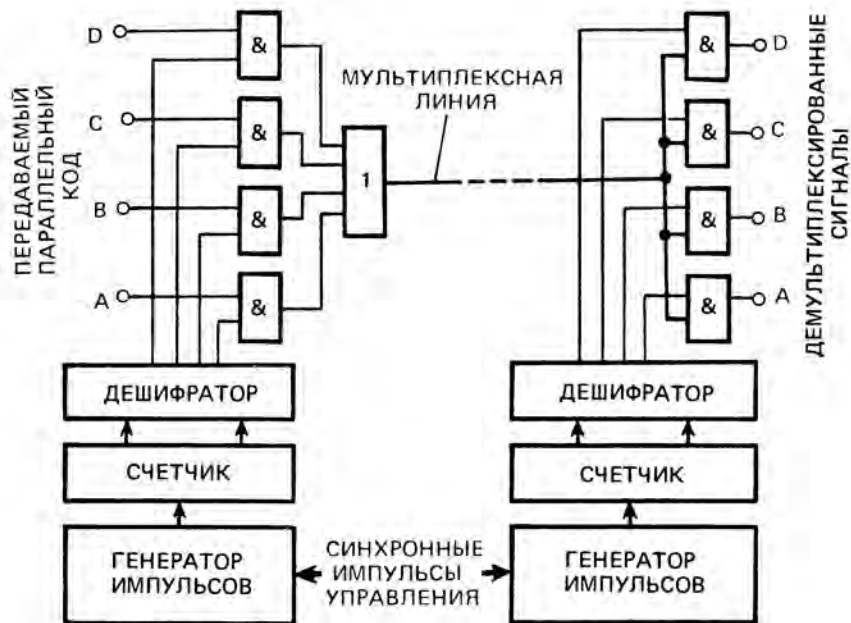


Рис. 6.16. Мультиплексор-демультиплексор.

плексора. Принцип действия мультиплексора поясняется рисунком 6.16. Предполагается, что исходные состояния счетчиков одинаковы, а импульсы управления строго синхронны. В вычислительных устройствах такие импульсы вырабатываются единым для всей системы тактовым генератором.

6.7. Как считает ЭВМ?

Обработка информации на ЭВМ сводится к процедурам, которые умеет выполнять процессор — к логическим и арифметическим операциям. Используя, например, представление чисел в дополнительном коде (см. 3.10), вычитание, умножение и деление можно свести к действиям одного типа — суммированию. Сочета-



Рис. 6.17. Сигналы арифметико-логического устройства.

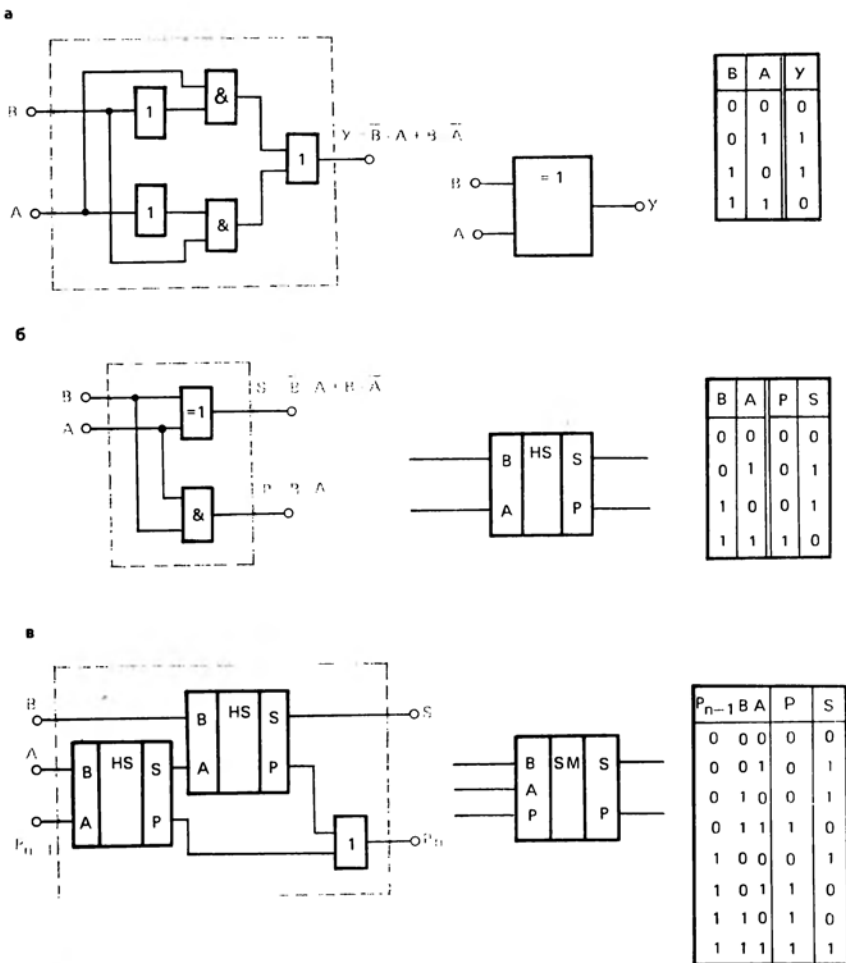


Рис. 6.18. Схему ИСКЛЮЧАЮЩЕЕ ИЛИ (а) нередко называют СУММАТОРОМ ПО МОДУЛЮ ДВА, или СХЕМОЙ НЕРАВНОЗНАЧНОСТИ. Она является составным элементом полусумматора (б). Сумматор (в) традиционно представляют состоящим из двух полусумматоров.

ние суммирования со сдвигом (см. 6.5) позволяет осуществить умножение n -разрядных двоичных чисел за n операций суммирования и $(n - 1)$ операций сдвига. Деление опять же сводится к суммированию и сдвигу, а число этих операций также сравнительно невелико и зависит от требуемой точности.

Часть процессора, в которой выполняются операции над кодами чисел, называют *арифметико-логическим устройством* (АЛУ). Остальные его составляющие относятся к *устройству управления* (УУ). Одна из простейших задач УУ — подавать в АЛУ оба операнда и шифр операции, выполняемой над ними.

В простейшем случае АЛУ представляет собой систему комби-

национных схем, входами которых являются коды чисел A и B и сигнал переноса C . На выходе АЛУ получают код результата и сигнал переноса P . Вид операции, выполняемой над числами A и B , определяет код управления F (рис. 6.17).

В выпускаемых промышленностью микросхемах АЛУ коды A , B и F — четырехразрядные. Несколько таких АЛУ могут быть соединены для параллельной обработки чисел большей длины.

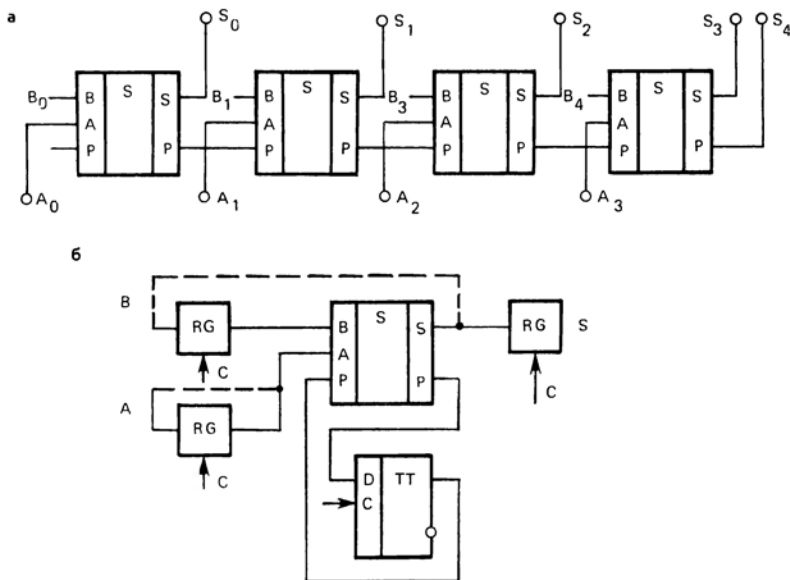
В режиме арифметического суммирования АЛУ функционирует как четырехразрядный *сумматор*, который как и любой параллельный многоразрядный *сумматор* вообще составлен из одноразрядных сумматоров. Каждая из его составляющих должна уметь складывать три одноразрядных числа, так как к суммируемым цифрам данного разряда может прибавляться еще и цифра переноса из младшего разряда.

Рисунок 6.18 поясняет действие одноразрядного сумматора и его составляющих, а рисунок 6.19 — методы построения параллельного и последовательного сумматоров многоразрядных чисел.

В современных ЭВМ умножение производится в одно действие при помощи так называемых матричных умножителей.

Рис. 6.19.

Устройства для суммирования многоразрядных двоичных чисел: параллельный сумматор (а) и сумматор последовательного типа (б). Во втором случае C -входы всех m -разрядных сдвигающих регистров и D -триггера соединены между собой. После поступления на этот общий C -вход m импульсов в регистре будет сумма чисел $B + A$. Пунктиром показаны соединения, позволяющие занести сумму в регистр A (т. е. превратить его в аккумулятор и сохранить слагаемое в регистре B (сравните с рис. 6.9).



6.8. Электронная память, или СОЗУ, ОЗУ, ПЗУ, ППЗУ и т. п.

В любой ЭВМ имеется *запоминающее устройство* (ЗУ). Емкость, или объем ЗУ — это количество информации, которое может храниться в нем в одно и то же время. Поэтому объем ЗУ измеряется в тех же единицах, что и количество информации — в битах (1 или 0), байтах (1 байт = 8 бит) или единицах К (сокращение от «кило», но $1К = 2^{10} = 1024$, а не 1000) и М ($1М = 2^{20}$).

Большинство ЗУ организовано таким образом, что в них хранят двоичные коды определенной разрядности. Такой код размещают во вполне определенном месте ЗУ. Каждому месту присваивается номер, или *адрес*. Адрес, выраженный в двоичной системе счисления, тоже является двоичным кодом, поэтому адреса также могут храниться в ЗУ (см. 2.10).

ЗУ в целом выполняет две основные операции: заносит слово по указанному адресу (**запись**) и сообщает, какое слово хранится по заданному адресу (**считывание**). Длительности этих процедур — время записи и время считывания — определяют быстроедействие ЗУ, а большее из них называют *временем обращения к ЗУ*. В полупроводниковых ЗУ время обращения, как правило, меньше одной миллионной доли секунды (1 мкс).

Наибольшим быстрыедействием характеризуется СОЗУ. За ним следует ОЗУ. В СОЗУ и ОЗУ информация может как записываться, так и считываться в процессе вычислений и обработки данных.

Для хранения данных, которые в процессе работы ЭВМ не изменяются, служат ПЗУ и ППЗУ — *постоянные и программируемые постоянные ЗУ*. Из ПЗУ и ППЗУ информация только считывается, причем в ПЗУ она может быть занесена лишь один раз (при изготовлении), а в ППЗУ записывается, стирается и перезаписывается на специальном стенде. ППЗУ называют иногда *репрограммируемыми ЗУ*.

Электронная память типа ОЗУ, СОЗУ, ПЗУ, ППЗУ организована таким образом, что в произвольный момент времени можно обратиться к любой ячейке, т. е. считать или записать код по произвольному адресу. Поэтому их называют ЗУПВ.

В отличие от ЗУПВ в ЗУ на *магнитных дисках*, магнитных лентах, ультразвуковых линиях задержки и др. информация записана последовательно код за кодом, и произвольный доступ к адресам таких ЗУ невозможен. Электронная память также может иметь последовательную структуру. Различают два способа ее организации: «первым вошел — последним вышел», или *стек*, и «первым вошел — первым вышел», или *очередь* (FIFO).

В основу работы современных ЗУ положены различные физические явления: магнитные, электрические, ультразвуковые, оптические, включая голографические и многие другие. Однако в настоящее время в мини-ЭВМ и микропроцессорных системах преобладают полупроводниковые СОЗУ, ОЗУ, ПЗУ, ППЗУ, а в качестве ЗУ большой емкости — устройства хранения информации на магнитных дисках и лентах.

Полупроводниковые ОЗУ, ПЗУ и ППЗУ — это БИС с наиболее высокой степенью интеграции. Если еще недавно к ординарным относили ЗУ в виде отдельной микросхемы емкостью 256 бит, то сейчас ординарной стала ИС ЗУ на 256 Кбит и ожидается появление микросхем ОЗУ емкостью 1, 4 и 16 Мбит.

6.9. Магистраль

На определенном этапе развития вычислительной техники добавление в структуру ЭВМ новых блоков порождало все увеличивающееся количество проводов для связи этих блоков с остальными устройствами ЭВМ. Но в начале 60-х годов был обоснован принцип и построены первые ЭВМ, в которых дополнительные устройства требовали не введения новых линий связи, а лишь удлинения уже существующих, причем такое удлинение конструкторы, как правило, предусматривали при разработке ЭВМ, устанавливая в них «лишние» разъемы для дополнительных блоков.

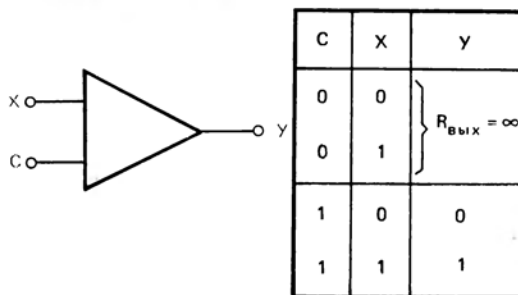
Новый способ обмена информацией сильно повлиял на структуру самих ЭВМ и в значительной мере способствовал появлению микропроцессоров. Обмен между многими источниками и приемниками стали осуществлять по группе одних и тех же проводов (*шина*), названных *магистралью*.

Подача сигнала на любой из проводов магистрали осуществляется единообразно через так называемые *магистральные усилители* («трестабильные схемы»).

Магистральный усилитель имеет два входа — информационный X и вход управления C . Если на входе C находится логическая 1, то на выходе Y повторяется сигнал X . Если же на C ноль, то независимо от сигнала на входе X выход Y изолирован и не оказывает никакого влияния на шину, к которой он присоединен (рис. 6.20).

Источниками кодов, подаваемых в магистраль, являются регистры, размещенные в различных блоках. Выходы триггеров каждого такого регистра выводятся на общую шину через группу магистральных усилителей с объединенными C -входами (рис. 6.21). Сигналы разрешения на такие общие C -входы различных блоков вырабатываются схемой типа дешифратора,

Рис. 6.20. Обозначение и описание работы магистрального усилителя.



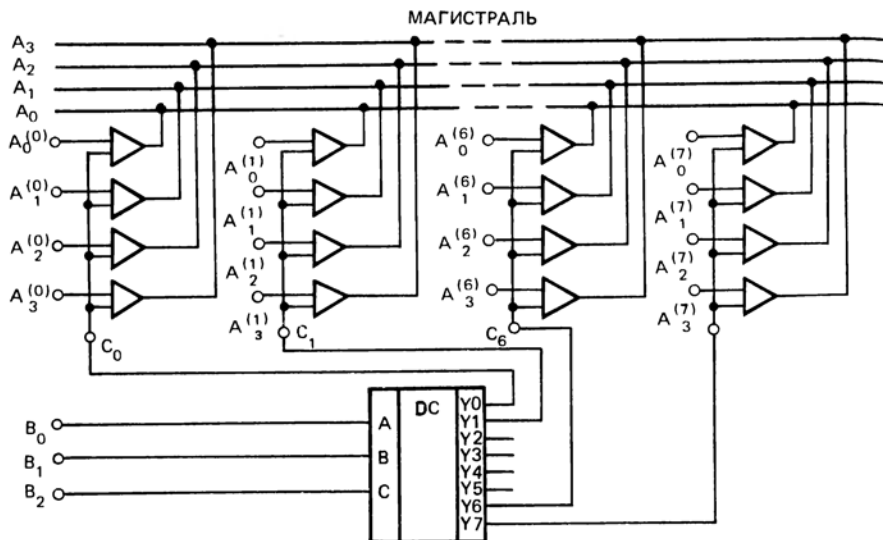


Рис. 6.21. В каждый момент времени логическая единица наблюдается лишь на одном из выходов дешифратора, поэтому информация на шины магистрали поступает только из одного источника.

поэтому одновременное подключение к магистрали двух и более источников исключено. Однако показанный на рисунке 6.21 дешифратор обычно не существует в виде автономного устройства, а как бы расчленен на части, находящиеся в блоках. Каждая такая часть дешифрирует одно значение кода на шинах *B* (на рис. 6.21 — *B*₀, *B*₁, *B*₂), которые разводятся таким же образом, как и остальные провода магистрали, но в их функции не входит прием информации от различных блоков. По ним во все блоки передаются коды управления, которые в самих блоках преобразуются в управляющие сигналы. Подобным образом в блоки, подключенные к магистрали, могут поступать и другие коды управления.

6.10. Окно в аналоговый мир или ЦАП и АЦП

Для имитации сложных сигналов управления исполнительными механизмами, наглядного представления итогов обработки данных и многих других целей широко используют *цифро-аналоговые преобразователи* (ЦАП) — устройства, превращающие код в пропорциональное ему напряжение или ток. Чаще всего преобразованию подвергается параллельный двоичный код связанного с ЦАП триггерного регистра, а действие ЦАП основано на суммировании токов (или напряжений), пропорциональных весам триггеров, находящихся в состоянии 1 (рис. 6.22). Разрядность современных ЦАП достигает 16, что эквивалентно возможности вырабатывать до $2^{16} = 65\,536$ градаций напряжения.

Цифровые значения различных физических величин получают при помощи *аналого-цифровых преобразователей* (АЦП). Под цифрой понимают обычно параллельный двоичный (двоично-десятичный) код, т. е. сигнал, пригодный для непосредственной обработки устройствами вычислительной техники.

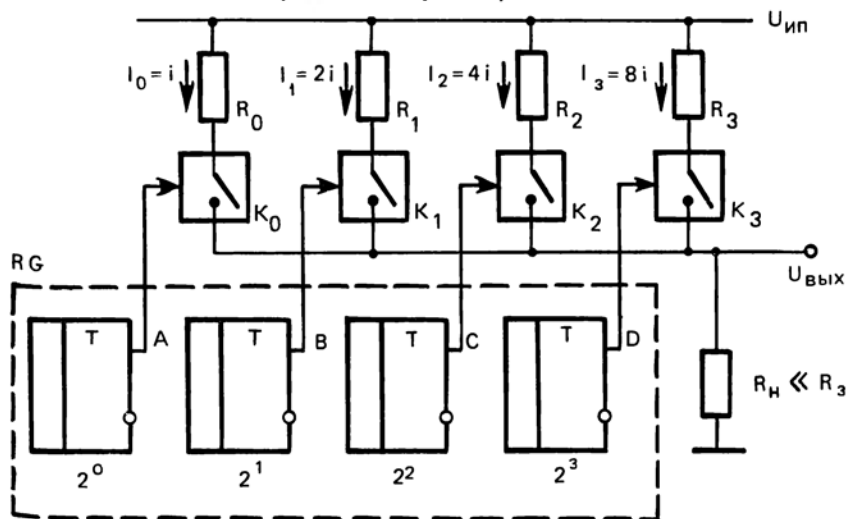
На вход АЦП поступают электрические сигналы, чаще всего — напряжение. Нередко их источниками служат преобразователи различных физических величин в их электрические аналоги. Сочетание преобразователей неэлектрических величин в их электрические аналоги и АЦП позволяет придавать любым физическим величинам форму, необходимую для обработки на ЭВМ.

В общем случае АЦП, преобразующий в код электрическое напряжение, состоит из двух частей — *устройства выборки и хранения* (УВХ) и собственно АЦП. УВХ запоминает напряжение в нужный момент времени и удерживает его неизменным до тех пор, пока собственно АЦП не осуществит преобразование в код. Запоминающим элементом УВХ служит конденсатор. При таком подходе можно считать, что собственно АЦП преобразует в код постоянное напряжение. Сравнение этого постоянного напряжения с напряжением, цифровое значение которого известно, — один из способов аналого-цифрового преобразования.

Источником напряжения с известным цифровым эквивалентом может быть ЦАП типа изображенного на рисунке 6.22 с высокостабильным источником питания. Если удается подобрать код на

Рис. 6.22.

ЦАП. Управляя замыканием ключей, триггеры включают в общую нагрузку R_H токи, пропорциональные весу разрядов, поэтому падение напряжения на резисторе R_H пропорционально коду ДСВА. В многоразрядных ЦАП токи задаются не резисторами, а электронными стабилизаторами тока и преобразуются в величину, пропорциональную весу разряда, при помощи делителей на прецизионных резисторах.



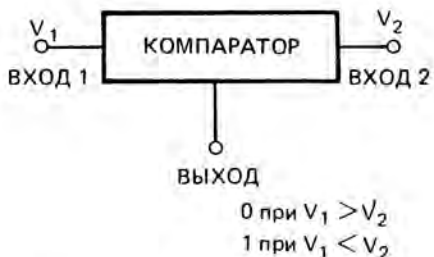


Рис. 6.23. Сигналы компаратора.

входе ЦАП, при котором его выходное напряжение равно измеряемому, этот код и есть цифровой эквивалент измеряемого напряжения.

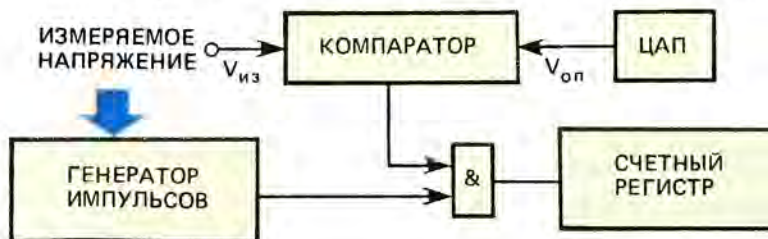
В зависимости от алгоритма нахождения искомого кода различают два основных типа АЦП — с *единичным приращением и поразрядным уравниванием*.

Работа первого из них напоминает процесс взвешивания, при котором на одну чашу весов кладут груз, «равный» постоянному напряжению на выходе УВХ, на вторую — одну за другой гири единичной массы. Если при очередном добавлении чаша с гирями перетянет, то число гирь на ней — искомый цифровой эквивалент измеряемого напряжения. Роль «электронных весов» выполняет весьма распространенная схема — *компаратор* (от лат. *comparator* — сравниваю). Она имеет два входа и один выход. Если напряжение на первом входе меньше, чем на втором, на выходе — логическая 1, если же наоборот — на выходе 0 (рис. 6.23). В качестве чаши, наполняемой гирями, служит счетчик импульсов на триггерах. Рисунок 6.24 поясняет действие АЦП с единичными приращениями.

При поразрядном уравнивании структура АЦП остается той же, но используются не единичные уравнивающие напряжения («гири»), а напряжения с весами 1, 2, 4, ..., 2^{n-1} . Сначала включается напряжение 2^{n-1} , затем к нему добавляется 2^{n-2}

Рис. 6.24.

АЦП При открытой схеме И каждый импульс генератора увеличивает содержимое счетчика на 1 и напряжение на выходе ЦАП получает единичное приращение. Когда это напряжение сравняется с измеряемым, компаратор запретит поступление импульсов в счетный регистр, а установившийся в нем код будет цифровым эквивалентом $V_{из}$. После считывания этого кода счетчик сбрасывается в 0 и АЦП готов к следующему измерению.



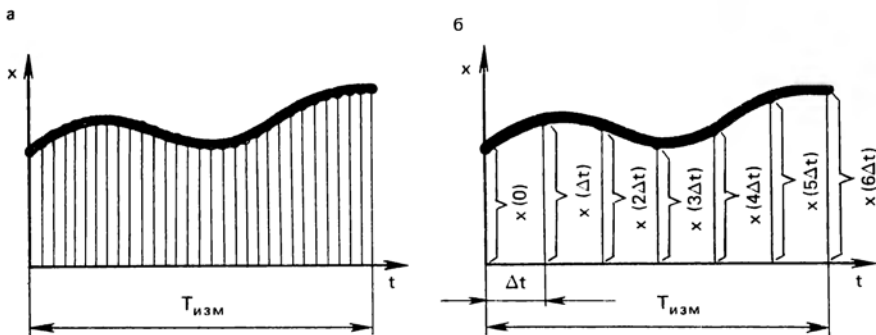


Рис. 6.25.

Чисто интуитивно для замены набором цифр непрерывной функции $x(t)$ отсчеты должны сниматься через как можно меньшие промежутки времени (а). Практически (б) достаточно измерить («оцифровать») значения функции через интервалы времени $\Delta t = 1/2f_c$, где f_c — предельная частота в спектре функции $x(t)$. Как правило, f_c известна или же может быть оценена в каждом конкретном случае.

и т. д. до 1. После каждого добавления анализируется сигнал компаратора. Если компенсирующее напряжение превысило измеряемое, последнюю добавку отключают, сбрасывая триггер соответствующего разряда, если же нет — оставляют и т. д. Искомый цифровой эквивалент находят за n процедур сравнения (в отличие от 2^n при методе единичных приращений), что и обуславливает более высокое быстродействие АЦП с поразрядным уравниванием.

Высокое быстродействие АЦП позволяет вводить в ЭВМ информацию о высокоскоростных процессах, так как заменить непрерывно изменяющийся сигнал набором цифр, например набором мгновенных значений этого сигнала, можно только в том случае, если набор цифр несет такую же информацию, как и сам сигнал. Ответ на вопрос, с какой минимальной частотой нужно преобразовывать мгновенные значения сигнала в цифру, дает теорема Котельникова, суть которой поясняется рисунком 6.25.

6.11. Еще раз о кодах

Предположим, что необходимо вводить в ЭВМ информацию о линейных перемещениях. Простейший способ — воспользоваться линейкой с делениями и преобразовывать соответствующие десятичные цифры в двоичный код. Если же линейка проградуирована в двоичной системе счисления, то отсчеты можно сразу получать в виде кода.

Чтобы автоматически следить за перемещениями, на линейку наносят проводящее покрытие, как показано на рисунке 6.26, а. Если к нему подвести электрическое напряжение, соответствующее логической 1, то со скользящих контактов, расположенных

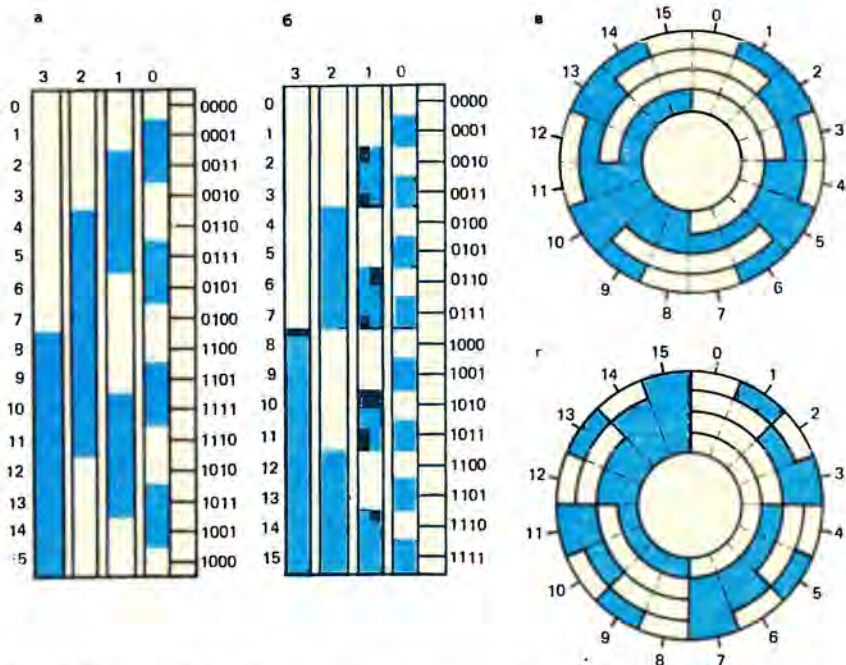


Рис. 6.26. В преобразователях перемещений или угла поворота в код (контактных и с фотосчитывателем) шкалы градуируются в коде Грея (а, в), и ошибка считывания не превышает одной наименьшей градации. В случае обыкновенного двоичного кода (б, г) она может достигать единицы в любом разряде.

на линии, перпендикулярной к длинной стороне линейки, будут сниматься коды, пропорциональные расстоянию от ее начала. Аналогичным образом в код может быть преобразован угол поворота (рис. 6.26, а, в). Однако и в том, и в другом случае применение прямого двоичного кода таит опасность получить большие ошибки считывания, обусловленные неточностью нанесения проводящего контакта и неточностью расположения скользящих контактов.

К сожалению, подобного рода неточности всегда имеют место и избежать их невозможно, поэтому если, например, при переходе от отсчета 7 к отсчету 8 контакт старшего разряда соприкоснется с проводящим слоем раньше, чем разомкнутся контакты младших разрядов, то вместо кода числа 8 будет считан код числа 15, и ошибка измерения длины составит почти 50 %.

Для исключения подобного рода грубых ошибок применяются специальные двоичные коды. Один из них — код Грея — характеризуется тем, что при переходе от любого числа к соседнему в коде изменяется только одна значащая цифра. Принцип построения преобразователей в коде Грея поясняет рисунок 6.26, б, г.

Код Грея получается из прямого двоичного кода при помощи одной операции — суммирования по модулю 2. Число в прямом

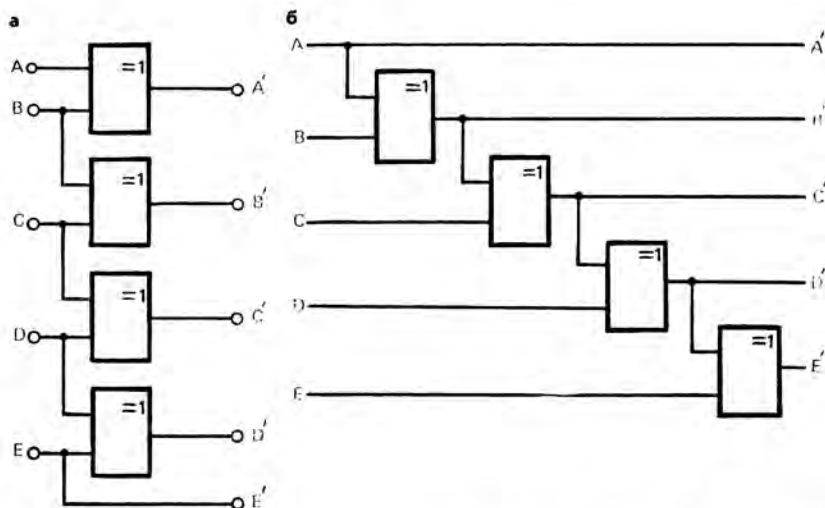


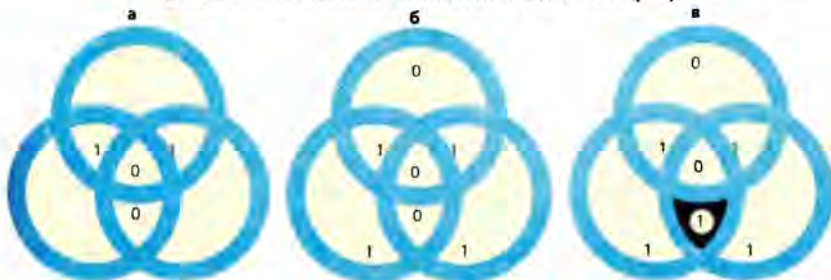
Рис. 6.27. Схемы преобразования кодов: прямого — в код Грея (а) и кода Грея — в прямой (б).

двоичном коде $EDCBA$ преобразуется в число в коде Грея $E'D'C'B'A'$ по следующим правилам:

$$\begin{aligned}
 A' &= A \oplus B, \\
 B' &= B \oplus C, \\
 C' &= C \oplus D, \\
 D' &= D \oplus E, \\
 E' &= E \oplus 0 = E,
 \end{aligned}
 \tag{6.1}$$

где знаком \oplus обозначена сумма по модулю 2 (см. гл. 4). На рисунке 6.27 пояснен принцип построения преобразователей двоичного кода в код Грея и наоборот.

Рис. 6.28. Диаграммы, поясняющие принцип формирования кода с исправлением ошибок. Подлежащее передаче слово 1010 (а) дополняется кодом 101 (б) таким образом, чтобы сумма цифр внутри каждого круга была четной. Если в процессе передачи-приема один из символов идентифицирован ошибочно (в), то это легко выявить и исправить, так как четность нарушится в двух кругах, а ошибочный символ находится в зоне их перекрытия.



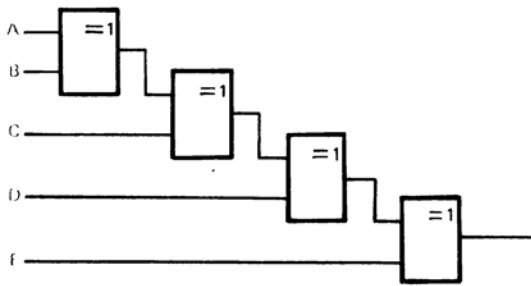
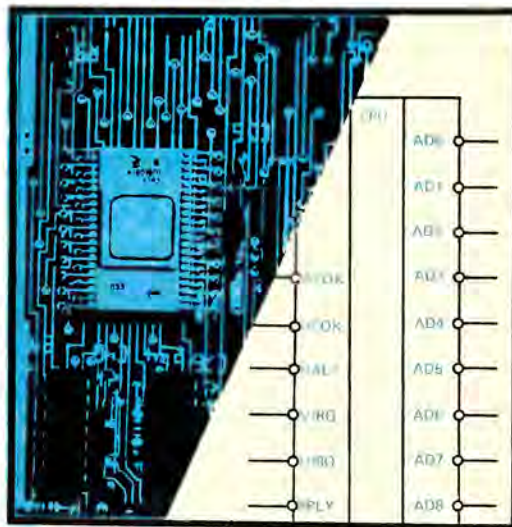


Рис. 6.29. Принцип построения многоходового устройства проверки на четность.

При передаче двоичных кодов по каналам связи на большие расстояния или же в условиях помех сигналы, соответствующие отдельным символам, могут исказиться, и приемник идентифицирует их неправильно. Подобного рода ошибки могут привести к крайне нежелательным последствиям. Избежать их позволяют коды с исправлением ошибок, часто называемые *кодами Хемминга* по имени инженера, предложившего их в конце 40-х годов.

Для этого по линии связи наряду с подлежащим передаче кодом передаются дополнительные символы. Их значения, т. е. 1 или 0, выбираются такими, чтобы сумма цифр определенных позиций всегда была четной. В приемнике производится проверка этих же сумм на четность. Сочетания позиций выбираются таким образом, чтобы по результатам проверки на четность можно было точно установить, какой именно бит принят ошибочно. При этом предполагается, что ошибка появляется не более, чем в одной позиции (рис. 6.28). Если же возможен ошибочный прием двух битов, то, в принципе, и это можно и установить, и исправить, но при помощи более сложных кодов.

Заметим, что проверка на четность производится при помощи схемы ИСКЛЮЧАЮЩЕЕ ИЛИ, причем число входов устройства проверки на четность увеличивается таким же способом, как и число входов схем И и ИЛИ (рис. 6.29).



7.1. Микропроцессор — еще не ЭВМ
 7.2. Схема микропроцессора — еще не ЭВМ
 7.3. Схема микропроцессора — еще не ЭВМ
 7.4. Схема микропроцессора — еще не ЭВМ
 7.5. Схема микропроцессора — еще не ЭВМ
 7.6. Схема микропроцессора — еще не ЭВМ
 7.7. Схема микропроцессора — еще не ЭВМ
 7.8. Схема микропроцессора — еще не ЭВМ
 7.9. Схема микропроцессора — еще не ЭВМ
 7.10. Схема микропроцессора — еще не ЭВМ

МИКРОПРОЦЕССОР КАК ИНТЕГРАЛЬНАЯ СХЕМА

Созданный в 1971 году первый микропроцессор (МП) был предназначен для работы в качестве процессора четырехразрядной ЭВМ и размещался в стандартном корпусе с шестнадцатью выводами. Собственно микросхема была выполнена на кремниевой пластинке размерами $2,8 \times 3,8$ мм и содержала 2250 транзисторов.

В вычислительной технике МП — «кирпичик» для построения компактных высокопроизводительных ЭВМ, в электронике — основной элемент **контроллеров** — устройств управления, непосредственно встраиваемых в объекты.

И в том, и в другом случае основная функция МП — выполнение написанных для него программ, что возможно только при постоянном взаимодействии МП с окружающим миром. Микропроцессор должен откуда-то получать данные и куда-то направлять результаты их обработки. Кроме того, само выполнение программы требует поступления в МП кодов команд из ОЗУ или ПЗУ, которые обычно не входят в его состав.

Использованные в данной и последующих главах названия сигналов, информационных линий, аббревиатуры команд имеют англоязычное происхождение.

7.1 Микропроцессор — еще не ЭВМ

Двигатель может выполнять различную работу, но требует для этого источника энергии, приспособлений, формирующих движения нужного характера, и объектов воздействия. Работа МП возможна при наличии источников электропитания и, по

крайней мере, двух типов устройств: запоминаящих и *ввода-вывода данных* (УВВ). Связь МП с УВВ и ЗУ осуществляется по *шинам* — системам проводов, несущих сигналы от МП к *внешним устройствам* (ВУ) и обратно. Выработка и прием этих сигналов как в МП, так и в ВУ осуществляется специальными схемами, называемыми интерфейсными. Вообще *интерфейс* (от англ. *interface* — стык, взаимодействие) — это совокупность средств, определяющих логический порядок взаимодействия систем (протокол) и вытекающие из протокола требования к аппаратуре и к программному обеспечению, если обмен данными осуществляется под управлением программы.

На практике под интерфейсом чаще всего понимают только электрическую схему, содержащуюся в устройстве для сопряжения его с ЭВМ, а интерфейсные средства, входящие в состав УУ процессора, носят название *системы ввода-вывода*. Чтобы упростить интерфейс внешних устройств, самые сложные его функции возлагают на систему ввода-вывода процессора. При наиболее распространенном (магистральном) принципе организации интерфейсы устройств единообразно подключаются к общей системе шин — *магистралам*, исходящей из процессора (рис. 7.1, а). При *радиальной организации интерфейса* к каждому устройству от процессора идет своя система шин (рис. 7.1, б).

Состав ВУ, подключаемых к магистралам МП, зависит от характера решаемых задач. Рассмотрим три основные конфигурации систем на базе МП (рис. 7.2).

Контроллер, или специализированное устройство обработки информации (рис. 7.2, а), помимо МП, содержит ПЗУ для хранения заранее отлаженной программы;

Рис. 7.1.

В вычислительных системах используются магистральный (а) и радиальный (б) принципы организации интерфейса. Физически магистраль выполняется в виде многопроводной шины с разъемами для подключения внешних устройств (ВУ), а в одноплатной микро-ЭВМ — с гнездами (панельками) для микро-схем.

