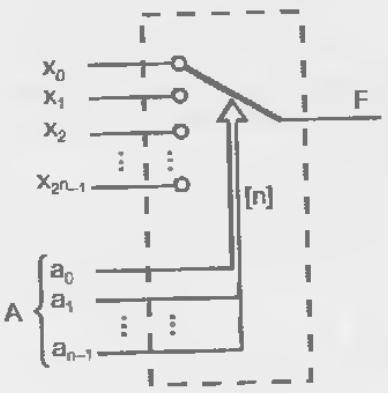
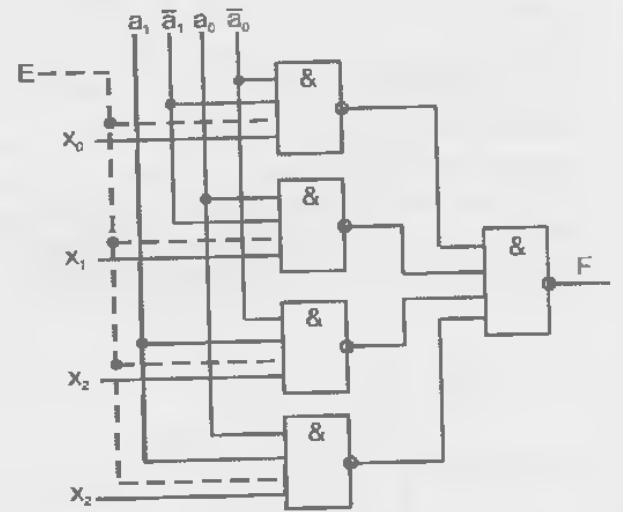
**17 Цифровые устройства комбинационного типа**

### **17.1 Мультиплексоры**

*Мультиплексоры осуществляют подключение одного из входных каналов к выходному под управлением управляющего (адресующего) слова.* Разрядности каналов могут быть различными, мультиплексоры для коммутации многоразрядных слов составляются из одноразрядных.

а) б)

Рисунок 17.1 – Упрощенное представление мультиплексора многопозиционным ключом (а) и реализация мультиплексора на элементах И-НЕ (б)

Входы мультиплексора делятся на две группы: информационные и адресую­щие. Работу мультиплексора можно упрощенно представить с помощью многопозиционного ключа. Для одноразрядного мультиплексора, представленного на рисунке 17.1*а,* адресующий код А задает переключателю определен­ное положение, соединяя с выходом F один из информационных входов хi . При нулевом адресующем коде переключатель занимает верхнее положение хо, с увеличением кода на единицу переходит в соседнее положение x1 и т. д.

Работа мультиплексора описывается соотношением

(17.1)

которое иногда называется *мультиплексной формулой.* При любом значении адресующего кода все слагаемые, кроме одного, равны нулю. Ненулевое слагаемое равно хi , где I — значение текущего адресного кода.

Схемотехнически мультиплексор реализует электронную версию показан­ного переключателя, имея, в отличие от него, только одностороннюю пере­дачу данных. На рисунке 17.1*6* показан мультиплексор с четырьмя информаци­онными входами, двумя адресными входами и входом разрешения работы При отсутствии разрешения работы (Е = 0) выход F становится нулевым независимо от информационных и адресных сигналов.

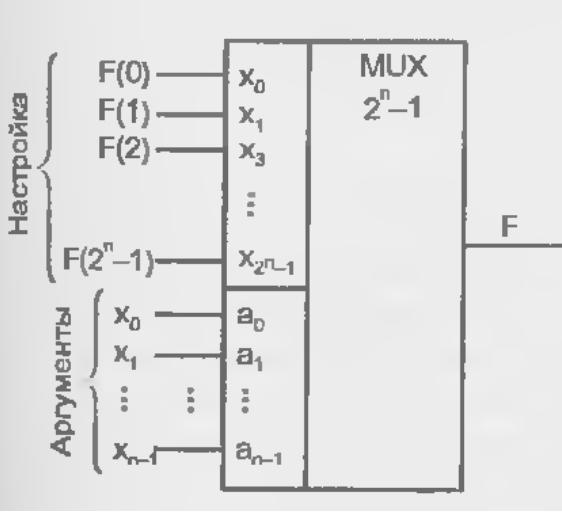
В стандартных сериях размерность мультиплексоров не более 16x1.

### **17.1.1 Универсальные логические модули на основе мультиплексоров**

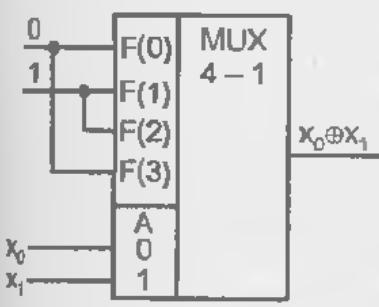
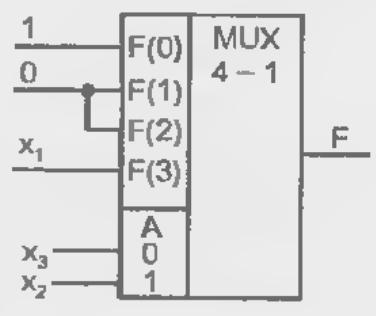
Универсальные логические модули (УЛМ) на основе мультиплексоров отно­сятся к устройствам, настраиваемым на решение той или иной задачи Уни­версальность их состоит в том, что для заданного числа аргументов можно настроить УЛМ на любую функцию. Известно, что общее число функций n аргументов выражается как 2n . С ростом n число функций растет чрезвы­чайно быстро. Хотя практический интерес представляют не все существую­щие функции, возможность получить любую из огромного числа функций свидетельствует о больших перспективах применения УЛМ.

### **17.1.1.1 Первый способ настройки УЛМ**

Первым способом настройки, используемым в УЛМ, является фиксация не­которых входов. Для этого способа справедливо следующее соотношение между числом аргументов и числом настроечных входов. Пусть число аргу­ментов n и требуется настройка на любую из функций. Тогда число комби­наций для кода настройки, равное числу функций, есть 2n . Для двоичного кода число комбинаций связано с разрядностью кода выражением 2m, где m— разрядность кода. Приравнивая число воспроизводимых функций к числу комбинаций кода настройки, имеем для числа настроечных входов соотношение m = 2n.



а)

б) в)

Рисунок 17.2 - Схема использования мультиплек­сора в качестве УЛМ (а), примеры воспроизведения функций при настройке константами (б) и при переносе одного аргумента в число сигналов настройки (в)

Полученному выражению отвечает соотношение между числом входов раз­ного типа для мультиплексора. При этом на адресные входы следует подавать аргументы функции, а на информационные входы — сигналы настрой­ки (рисунок 17.2*а).* Таким образом, *для использования мультиплексора в качест­ве УЛМ следует изменить назначение его входов.*

Рисунок 17.2*а -* иллюстрирует возможность воспроизведения с помощью мультиплексора любой функции n аргументов. Действительно, каждому на­бору аргументов соответствует передача на выход одного из сигналов на­стройки. Если этот сигнал есть значение функции на данном наборе аргументов, то задача решена. Разным функциям будут соответствовать разные коды настройки. Алфавитом настройки будет {0,1} — настройка осуществля­ется константами 0 и 1. На рисунке 17.2*б* показан пример воспроизведения функции неравнозначности x1  х2 с помощью мультиплексора "4—1".

Большое число настроечных входов затрудняет реализацию УЛМ. Для УЛМ, расположенных внутри кристалла, можно вводить код настройки последова­тельно в сдвигающий регистр, к разрядам которого подключены входы на­стройки. Тогда внешним входом настройки будет всего один, но настройка будет занимать не один такт, а 2n тактов. Возможны и промежуточные по­следовательно-параллельные варианты ввода кода настройки.

### **17.1.1.2 Второй способ настройки УЛМ**

Большее число входов настройки наталкивает на поиск возможностей их уменьшения. Такие возможности существуют и заключаются в расширении алфавита настроечных сигналов. Если от алфавита {0,1} перейти к алфавиту {0,1, i}, где - литерал одного из аргументов, то число входов аргументов сократится на единицу, а число настроечных входов - вдвое. Напомним, что под литералом переменной принимается либо сама переменная, либо се инверсия. Перенос одного из аргументов в число сигналов настройки не влечет за собою каких-либо схемных изменений. На том же оборудовании будут реализованы функции с числом аргументов на единицу больше, чем при настройке константами.

Для нового алфавита код настройки находится следующим образом. Аргу­менты за исключением полаются на адресующие входы, что соответствует их фиксации в выражении для искомой функции, которая становится функцией единственного аргумента . Эту функцию, которую назовем остаточ­ной, и нужно подавать на настроечные входы.

Если искомая функция зависит от и аргументов и в число сигналов на­стройки будет перенесен один из аргументов, то возникает *n* вариантов ре­шения задачи, т. к. в сигналы настройки может быть перенесен любой аргу­мент. Спрашивается, какой именно аргумент целесообразно переносить в сигналы настройки? Здесь можно опираться на рекомендацию: в настроеч­ные сигналы следует переводить аргумент, который имеет минимальное число вхождений в термы функции. В этом случае будут максимально ис­пользованы как бы внутренние логические ресурсы мультиплексора, а среди сигналов настройки увеличится число констант, что и считается благопри­ятным для схемной реализации УЛМ.

Проиллюстрируем сказанное примером воспроизведения функции трех ар­гументов F=х1х2х3 \/ . Минимальное число вхождений в выражение функции имеет переменная х1 которую и перенесем в число сигналов на­стройки. Остаточная функция определится по таблице 17.1, *а.*

Таблица 17.1 (a) - Остаточная функция для функции трех ар­гументов F=х1х2х3 \/

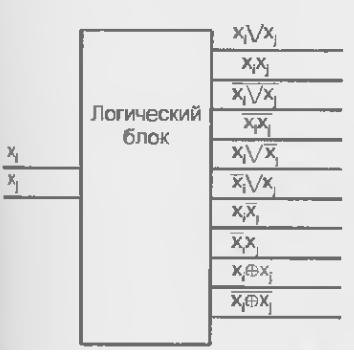
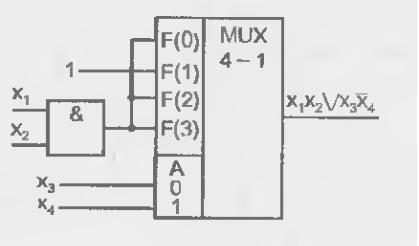
|  |  |  |
| --- | --- | --- |
| х2 | х3 | Fост |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | х1 |

Таблица 17.1 (б) - Остаточная функция для функции двух ар­гументов F=х1х2 \/ х3

|  |  |  |
| --- | --- | --- |
| х4 | х3 | Fост |
| 0 | 0 | х1х2 |
| 0 | 1 | 1 |
| 1 | 0 | х1х2 |
| 1 | 1 | х1х2 |

Схема УЛМ приведена на рисунке 17.2, *в.*

По пути расширения алфавита сигналов настройки можно идти и дальше, но при этом понадобятся дополнительные логические схемы, воспроизво­дящие остаточные функции, которые будут уже зависеть более чем от од­ного аргумента.

1. б)

Рисунок 17.3 - Логический блок выработки сигналов настройки УЛМ с переносом двух аргументов в сигналы настройки (а) и пример схемы воспроизведения функции четырех аргументов на мультиплексоре "4—1" (б)

Если в сигналы настройки перевести два аргумента, то дополнительные ло­гические схемы будут двухвходовыми вентилями, что мало усложняет УЛМ и может оказаться приемлемым решением. В этом случае для сохранения универсальности УЛМ мультиплексору нужно предпослать блок выработки остаточных функций, в котором формируются все функции 2-х переменных (за исключением констант 0 и 1 и литералов самих переменных, которые не требуется вырабатывать). Такой блок показан на рисунке 17.3а. Пример реали­зации функции F=х1х2 \/ х3.при алфавите настройки {0, 1, , }показан на рисунке 17.3*б.* Таблица остаточной функции для этого примера приведена в таблице 17.1б*.*

### **17.2 Демультиплексоры**

*Демультиплексорами называются устройства, которые позволяют подключать один вход к нескольким выходам.* Демультиплексор можно построить на основе точно таких же схем логического "И", как и при построении мультиплексора. Существенным отличием от мультиплексора является возможность объединения нескольких входов в один без дополнительных схем. Однако для увеличения нагрузочной способности микросхемы, на входе демультиплексора для усиления входного сигнала лучше поставить инвертор.

Схема демультиплексора приведена на рисунке 17.4. В этой схеме для выбора конкретного выхода демультиплексора, как и в мультиплексоре, используется двоичный дешифратор.

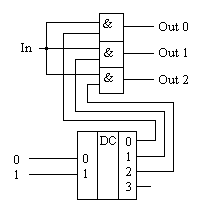


Рисунок 17.4 - Принципиальная схема демультиплексора, управляемого двоичным кодом.

Однако, если рассмотреть принципиальную схему самого [дешифратора](http://www.sibsutis.ru/~mavr/digital/DC.htm), то можно значительно упростить демультиплексор. Достаточно просто к каждому логическому элементу 'И', входящему в состав дешифратора просто добавить ещё один вход – In. Такую схему часто называют дешифратором с входом разрешения работы. Условно-графическое изображение демультиплексора приведено на рисунке 17.5.

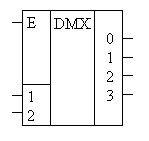


Рисунок 17.5 - Условно графическое обозначение демультиплексора с четырьмя выходами.

В этом обозначении вход In обозначен как вход E, а выходы не названы никак, оставлены только их номера.

В МОП микросхемах не существует отдельных микросхем демультиплексоров, так как МОП мультиплексоры, описанные ранее по информационным сигналам не различают вход и выход, т.е. направление распространения информационных сигналов, точно также как и в механических ключах, может быть произвольным. Если поменять входы и выход местами, то КМОП мультиплексоры будут работать в качестве демультиплексоров. Поэтому их часто называют просто коммутаторами.

### **17.3 Компараторы**

*Компараторы (устройства сравнения) определяют отношения между двумя словами.* Основными отношениями, через которые можно выразить осталь­ные, можно считать два — "равно" и "больше".

Определим функции, вырабатываемые компараторами, следующим образом: они принимают единичное значение (истинны), если соблюдается условие, указанное в индексе обозначения функции. Например, функция РА=В =1, если А = В и принимает нулевое значение при АВ.

Приняв в качестве основных отношения "равно" и "больше", для остальных можно записать:

FAB=FA=B; FA<B=FB>A; FA≥B=FB>A; FA≤B= FA>B (17.2)

Эти отношения используются как логические условия в микропрограммах, в устройствах контроля и диагностики ЭВМ и т. д.

В сериях цифровых элементов обычно имеются компараторы с тремя выхо­дами: "равно", "больше" и "меньше" (рисунок 17.6) Для краткости записей в ин­дексе выходных функций указывается только слово А.

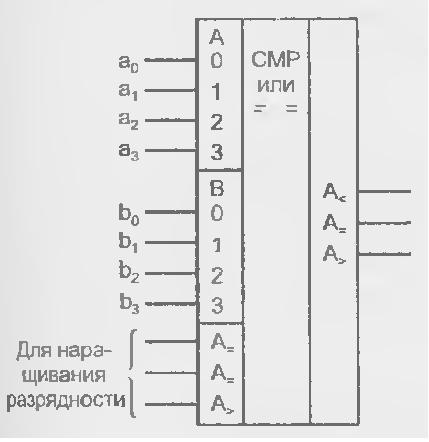


Рисунок 17.6 - Условное обозначение компаратора с тремя выходами

Устройства сравнения на равенство строятся на основе поразрядных опера­ций над одноименными разрядами обоих слов. Слова равны, если равны все одноименные их разряды, т. е. если в обоих нули или единицы. Признак равенства разрядов

(17.3)

Признак неравенства разрядов

(17.4)

Признак равенства слов R = rn-1  rn-2 … r0.

Схема компаратора на равенство в базисе И-НЕ показана на рисунке 17.7, *а.*

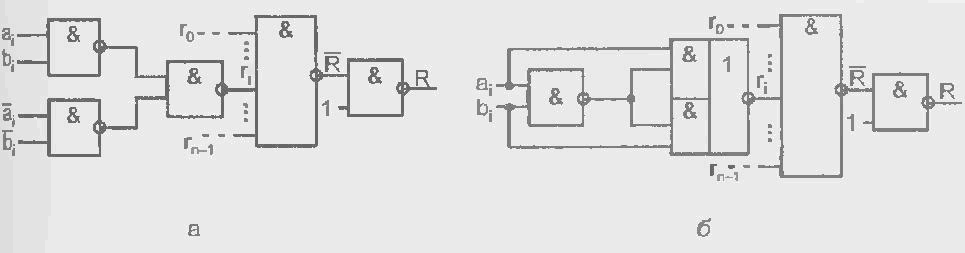
**

Рисунок 17.7 – Схемы компараторов на равенство

Схема без парафазных входов (рисунок 17.7, *6)* основана на выражениях для гр преобразованных следующим образом:

(17.5)

Таблица 17.2 - Построение компаратора на "больше" для одноразряд­ных слов

|  |  |  |
| --- | --- | --- |
| a | b | FA>B |
| 0  0  1  1 | 0  1  0  1 | 0  0  1  0 |

Построение компаратора на "больше" для одноразряд­ных слов (таблица 17.2) требует реализации функции .

Функцию FA>B для многоразрядных слов проще всего получить на основе рассуждений. Пусть нужно сравнить двухразрядные слова. Если старшие разряды а1 и b1 не равны, то результат известен независимо от младших раз­рядов: при а1 = 1 и b1 = 0 имеем А>В, а при а1 = 0 и b1 = 1 имеем A<В. Ес­ли же а1=b1, результат еще неизвестен, и требуется анализ следующего разряда по тому же алгоритму. Поэтому для двухразрядных слов можно за­писать

(17.6)

Подобный же подход справедлив и для слои любой разрядности — к ана­лизу следующего разряда нужно переходить только при равенстве преды­дущих. Таким образом, для общего сличая n-разрядных слов имеем

(17.7)

### **17.4 Шифраторы, дешифраторы и преобразователи кодов**

*Шифратор* — это комбинационное устройство, преоб­разующее десятичные числа в двоичную систему счисле­ния, причем каждому входу может быть поставлено в соот­ветствие десятичное число, а набор выходных логических сигналов соответствует определенному двоичному коду. Шифратор иногда называют «кодером» (от англ. coder) и используют, например, для перевода десятичных чисел, набранных на клавиатуре кнопочного пульта управления, в двоичные числа. Если количество входов настолько ве­лико, что в шифраторе используются все возможные ком­бинации сигналов на выходе, то такой шифратор называ­ется полным, если не все, то неполным. Число входов и выходов в полном шифраторе связано соотношением n = 2m, где n — число входов, m — число выходов. Так, для пре­образования кода кнопочного пульта в четырехразрядное двоичное число достаточно использовать лишь 10 входов, в то время как полное число возможных входов будет рав­но 16 (n = 24 = 16), поэтому шифратор 10x4 (из 10 в 4) будет неполным.

Рассмотрим пример построения шифратора для преоб­разования десятиразрядного единичного кода (десятичных чисел от 0 до 9) в двоичный код. При этом предполагает­ся, что сигнал, соответствующий логической единице, в каждый момент времени подается только на один вход. Условное обозначение такого шифратора и таблица соот­ветствия кода приведены на рисунке 17.8. Используя данную таблицу соответствия, запишем логические выражения, включая в логическую сумму те входные переменные, ко­торые соответствуют единице некоторой выходной пере­менной. Так, на выходе у, будет логическая «1»тогда, когда логическая «1» будет или на входе X,, или Х3, или Х5, или Х7, или X,, т. е.

У1 = X1 + Х3 + Х5 + Х7 + X9. (17.8)

Аналогично получаем

У2 = X2 + X3 + X6 + X7 (17.9)

У3 = X4 + X5 + X6 + Х7, у4= Х8+ Х9. (17.10)

Представим на рисунке 17.9 схему такого шифратора, ис­пользуя элементы ИЛИ.

На практике часто используют шифратор с приорите­том. В таких шифраторах код двоичного числа соответ­ствует наивысшему номеру входа, на который подан сигнал «1», т. е. на приоритетный шифратор допускается по­давать сигналы на несколько входов, а он выставляет на выходе код числа, соответствующего старшему входу.

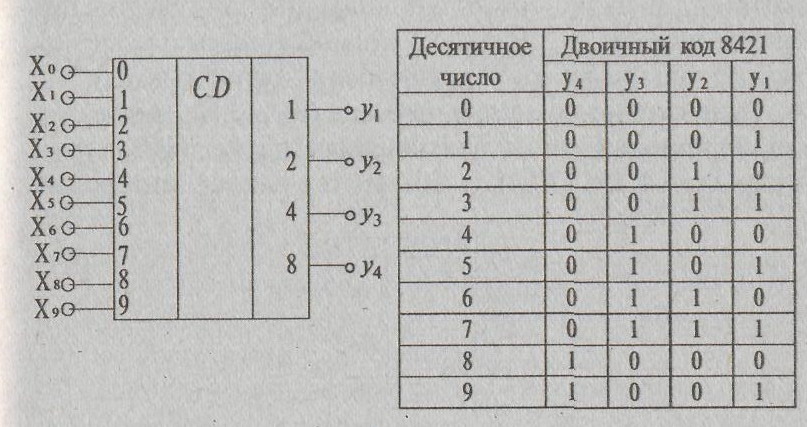


Рисунок 17.8 - Условное обозначение шифратора и таблица соот­ветствия кода

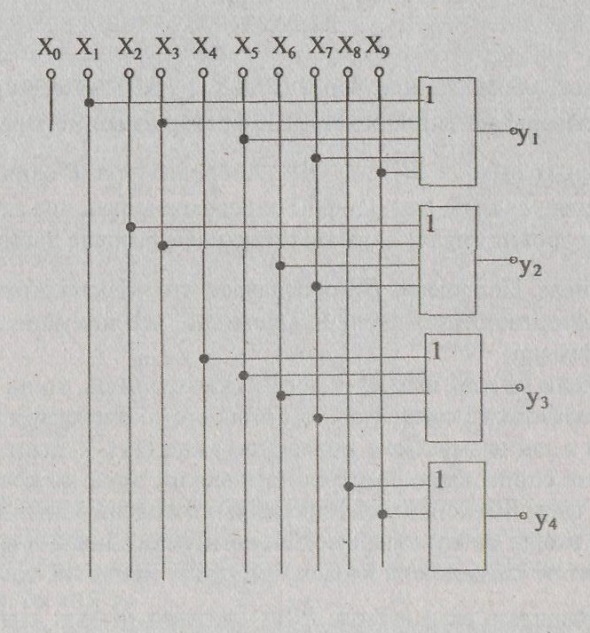


Рисунок 17.9 – Схема шифратора, с ис­пользованием элементов ИЛИ

Рассмотрим в качестве примера (рисунок 17.10) шифратор с приоритетом (приоритетный шифратор) К555ИВЗ серии микросхем К555 (ТТЛШ). Шифратор имеет 9 инверсных входов, обозначенных через *PR1, ... , РР9.* Аббревиатура *PR* обозначает «приоритет». Шифратор имеет четыре ин­версных выхода *В1, ... , В8*. Аббревиатура *В* означает «шина» (от англ. *bus).* Цифры определяют значение актив­ного уровня (нуля) в соответствующем разряде двоично­го числа. Например, В8 обозначает, что ноль на этом вы­ходе соответствует числу 8. Очевидно, что это неполный шифратор.

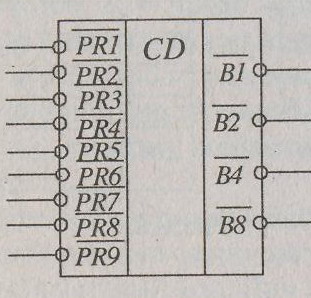


Рисунок 17.10 - Шифратор с приоритетом (приоритетный шифратор) К555ИВЗ серии микросхем К555.

Если на всех входах — логическая единица, то на всех выходах также логическая единица, что соответствует чис­лу 0 в так называемом инверсном коде (1111). Если хотя бы на одном входе имеется логический ноль, то состоя­ние выходных сигналов определяется наибольшим номе­ром входа, на котором имеется логический ноль, и не за­висит от сигналов на входах, имеющих меньший номер.

Например, если на входе *PR1 —* логический ноль, а на всех остальных входах — логическая единица, то на выходах имеются следующие сигналы: *В1* - 0, В2 -1, В4 -1, В8 -1, что соответствует числу 1 в инверсном коде (1110).

Если на входе *PR9* логический ноль, то независимо от других входных сигналов на выходах имеются следующие сигналы: *В1-0, В2-1, В4-1, В8-0*, что соответствует числу 9 в инверсном коде (0110).

Основное назначение шифратора — преобразование номера источника сигнала в код (например, номера нажа­той кнопки некоторой клавиатуры).

Для получения шифраторов с большим числом входов, т. е. наращивания размерности шифратора, объединяют микросхемы шифраторов с дополнительными выводами.

Так микросхема К555ИВ1 (рисунок 17.11) представляет со­бой приоритетный шифратор 8 х 3, т. е. имеет 8 инверсных входов и 3 инверсных выхода. Помимо этого она имеет вход разрешения EI, выход переноса Е0 и выход G, опре­деляющий признак входного информационного сигнала.

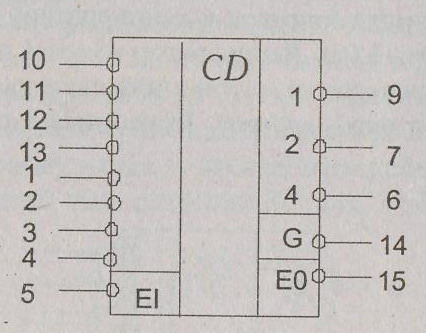


Рисунок 17.11 - Микросхема К555ИВ1

Если на всех информационных входах логическая 1, то при подаче на вход EI логического 0, на выходах 1-2-4 и G будут такие логические 1, а на выходе переноса Е0 - логический 0.

Если активизировать один из информационных входов (подать на него логический 0), то на выходах 1-2-4 появится инверсный код, соответствующий номеру активи­зированного входа, на входе G - логический 0, являю­щийся признаком подачи входного сигнала, а на выходе Е0 - логическая 1.

Если же микросхема не активизирована, т. е. на вход разрешения Е1 подана логическая 1, то на всех выходах микросхемы также будет логическая 1 независимо от того, что будет подано на информационные входы.

*Дешифратором* называется комбинационное устрой­ство, преобразующее n-разрядный двоичный код в логи­ческий сигнал, появляющийся на том выходе, десятичный номер которого соответствует двоичному коду. Число вхо­дов и выходов в так называемом полном дешифраторе свя­зано соотношением

*т* = , где *п -* число входов, а *т -* число выходов. Если в работе дешифратора используется неполное число выходов, то такой дешифратор называет­ся неполным. Так, например, дешифратор, имеющий 4 входа и 16 выходов, будет полным, а если бы выходов было только 10, то он являлся бы неполным.

Обратимся для примера к дешифратору К555ИД6 се­рии К555 (рисунок 17.12). Дешифратор имеет 4 прямых входа, обозначенных через *А1..,А*8. Аббревиатура А обознача­ет «адрес» (от англ. *address).* Указанные входы называют адресными. Цифры определяют значения активного уров­ня (единицы) в соответствующем разряде двоичного числа.

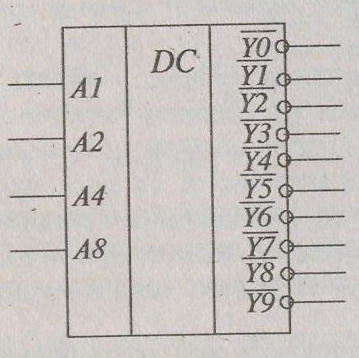


Рисунок 17.12 - Микросхема К555ИД6

Дешифратор имеет 10 инверсных выходов Y0, ... , Y9. Цифры определяют десятичное число, соответствующее заданному двоичному числу на входах. Очевидно, что этот дешифратор неполный.

Значение активного уровня (нуля) имеет тот выход, номер которого равен десятичному числу, определяемому двоичным числом на входе. Например, если на всех вхо­дах - логические нули, то на выходе *Y0 -* логический ноль, а на остальных выходах - логическая единица. Если на входе *А2 -* логическая единица, а на остальных вхо­дах - логический ноль, то на выходе *Y2 -* логический ноль, а на остальных выходах - логическая единица. Если на входе - двоичное число, превышающее 9 (например, на всех входах единицы, что соответствует двоичному чис­лу 1111 и десятичному числу 15), то на всех выходах - ло­гическая единица.

Помимо информационных имеется один или более входов, называемых входами разрешения, или адресными входами. Так, микросхема КР531ИД14 представляет собой два дешифратора 2 х 4, т. е. каждый дешифратор имеет два информационных входа и четыре инверсных выхода, а также инверсный вход разрешения (рисунок 17.13).

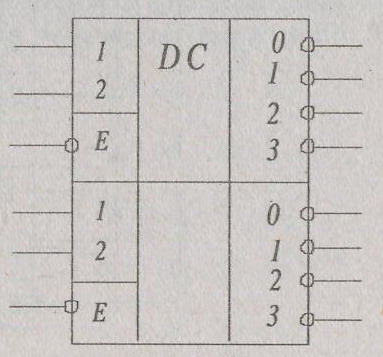


Рисунок 17.13 - Микросхема К555ИД14

### **17.5 Сумматоры и полусумматоры**

Полусумматор это логическая цепь, которая вырабатывает сигналы суммы (S) и переноса (С) при сложении двух двоичных чисел а и в.

Составим таблицу функционирования.

Таблица 17.4 - Функционирование полусумматора

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | S | C |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Из таблицы получим:

– сигнал суммы; (17.11)

– сигнал переноса. (17.12)

Эти выражения упрощению не поддаются.

Приведем к виду, удобному для реализации на элементах ИЛИ-НЕ.

(17.13)

(17.14)

Исходя из полученных формул составим схему полусумматора (рисунок 17.15а):

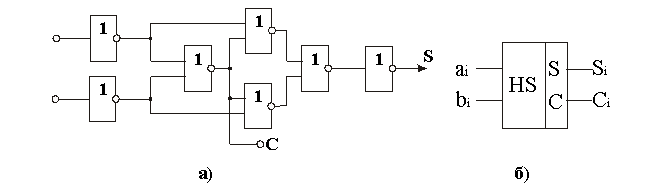


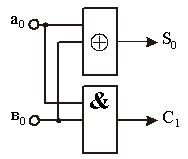
Рисунок 17.15 – Схема полусумматора

Поскольку полусумматор имеет широкое применение и его выпускают в виде отдельной микросхемы, он имеет собственное обозначение (рисунок 17.15б).

Составляя дизъюнктивную нормальную форму для полусумматора, мы получили следующие булевы функции:

и (17.15)

Следовательно, перенос происходит с помощью функции И, а выработка сигнала суммы (функции неравнозначности) производится элементом ИСКЛЮЧАЮЩЕЕ – ИЛИ. На рисунке 17.16 показана схема полусумматора, составленная из этих элементов.

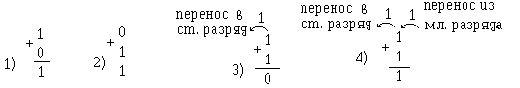


# Рисунок 17.16 – Схема полусумматора

Сумматор. В отличие от полусумматора должен воспринимать 3 входных сигнала: 2 слагаемых и сигнал переноса с предыдущего разряда.

Сумматором называется операционный узел ЭВМ, выполняющий операцию арифметического сложения двух чисел.

Чтобы понять сущность работы комбинационного сумматора, рассмотрим примеры суммирования двух одноразрядных двоичных чисел:



Из приведенных примеров (1 - 4) видно, что если отсутствует перенос из младшего разряда, то перенос в старший разряд может быть только в одном случае, когда оба числа равны единице. Если же имеется перенос из младшего разряда, то перенос в старший разряд будет всегда, кроме одного случая, когда оба слагаемых равны нулю.

Составим таблицу функционирования:

Таблица 17.5 - Функционирование сумматора

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ai | bi | Ci | Si | Ci+1 |
| 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 |

Схема сумматора может быть реализована на двух полусумматорах, соединенных как указано на схеме рисунка 17.17. В этой схеме выделим промежуточные сигналы Pi, gi, ri. Введем эти сигналы в новую таблицу функционирования. Соответствие работы этой схемы (рисунок 17.17) и таблицы фунционирования можно проверить перебором всех возможных вариантов.

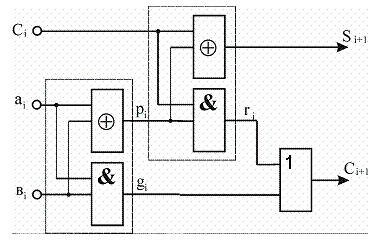


Рисунок 17.17 – Схема полного сумматора

# Многоразрядный сумматор с последовательным переносом. Таким образом, в общем случае для каждого разряда необходима логическая схема с тремя входами ai, bi, Ci и двумя выходами Si, Ci+1. Такая схема и есть полный сумматор. Ее можно реализовать с помощью двух полусумматоров.

Таблица 17.6 - Функционирование полного сумматора

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Входы | | | Промежуточные величины | | | Выходы | |
| ai | bi | Ci | pi | gi | ri | Si | Ci+1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |

Для сложения двух многоразрядных двоичных чисел на каждый разряд необходим один полный сумматор. Только в младшем разряде можно обойтись полусумматором. На рисунке 17.18 приведена схема, предназначенная для сложения двух четырехразрядных чисел *А* и *В*. Эта схема выпускается в интегральном исполнении. В ее младшем разряде также используется полный сумматор, чтобы иметь возможность наращивания разрядности схемы.

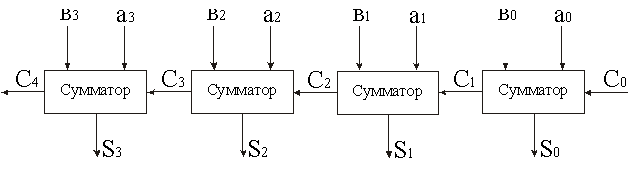


Рисунок 17.18 – Сумматор с последовательным переносом

# Сумматоры с параллельным переносом. Время выполнения операции в сумматоре с параллельным переносом намного больше времени сложения в одноразрядном сумматоре. Действительно, сигнал переноса С4 только тогда может принять истинное значение, когда будет установлено правильное значение С3. Такой порядок выполнения операций называется последовательным переносом (Ripple Carry).

Чтобы уменьшить время операции сложения многоразрядных чисел можно использовать схемы параллельного переноса (Carry look-ahead). При этом все сигналы переноса вычисляются непосредственно по значениям входных переменных.

Согласно таблице переключений, в общем случае для сигнала переноса любого i-го разряда справедливо соотношение:

(17.16)

Величины gi, ri вычисляются в качестве промежуточных результатов и в полном сумматоре. Следовательно, их получение не требует дополнительных затрат. Смысл этих величин объясняется совсем просто. Сигнал gi вырабатывается тогда, когда в данном разряде перенос происходит из-за комбинации входных переменных ai, bi. Поэтому его называют функцией генерации переноса. Сигнал Pi показывает, передается ли полученный в младшем разряде сигнал переноса Ci дальше. Поэтому он называется функцией распространения переноса.

Пользуясь выражением (17.16), можно вывести следующие формулы для вычисления сигналов переноса:

(17.17)

 Очевидно, что хотя полученные выражения достаточно сложные, время формирования сигнала переноса в любой разрад с помощью вспомогательных функций определяется только времением здержки распространения сигнала на двух элементах. Эти функции реализуются специальным комбинационным устройством – схемой ускоренного переноса.

Схема сумматора с параллельным переносом приведена на рисунке 17.19. На рисунке 17.20 изображена схема устройства параллельного переноса в группе из четырех разрядов. Эта схема реализует систему уравнений (17.2).

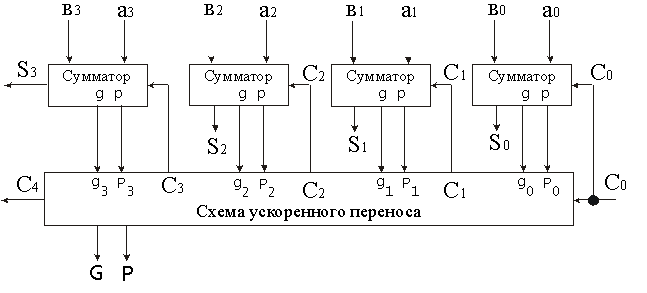
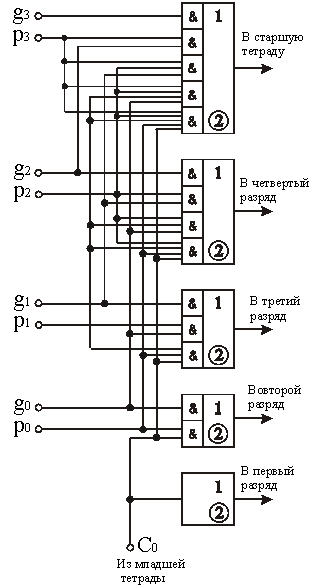


Рисунок 17.19 –Схема сумматора с параллельным переносом



# Рисунок 17.20 – Схема ускоренного переноса

# 

Схема выпускается в интегральном исполнении.

Сложение чисел, содержащих более четырех разрядов, можно реализовать подключением нескольких четырехразярадных сумматоров.

### **17.6 Арифметико-логические устройства и блоки ускоренного переноса**

Арифметико-логические устройства АЛУ (ALU, Arithmetic-Logic Unit) вы­полняют над словами ряд действий. Основой АЛУ служит сумматор, схема которого дополнена логикой, расширяющей функциональные возможности АЛУ и обеспечивающей его перестройку с одной операции на другую.

Обычно АЛУ четырехразрядные и для наращивания разрядности объединя­ются с формированием последовательных или параллельных переносов. Логические возможности АЛУ разных технологий (ТТЛШ, КМОП, ЭСЛ) сходны. В силу самодвойственности выполняемых операций условное обо­значение и таблица истинности АЛУ встречаются в двух вариантах, отличающихся взаимно инверсными значениями переменных.

АЛУ (рисунок 17.21) имеет входы операндов А и В, входы выбора операций S, вход переноса Ci и вход М (Моdе), сигнал которого задает тип выполняе­мых операций: логические (М = 1) или арифметико-логические (М = 0). Ре­зультат операции вырабатывается на выходах F, выходы G и Н дают функции генерации и прозрачности, используемые для организаций параллельных пе­реносов при наращивании размерности АЛУ. Сигнал Со — выходной пере­нос, а выход А = В есть выход сравнения на равенство с открытым коллек­тором.

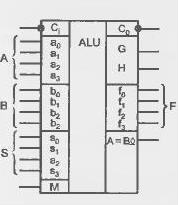


Рисунок 17.21 - Условнее обозначение АЛУ

Перечень выполняемых АЛУ операций дан в таблице 17.7. Для краткости дво­ичные числа s3s2s1s0 представлены их десятичными эквивалентами. Под утолщенными обозначениями 1 и 0 следует понимать наборы 1111 и 0000, входной перенос поступает в младший разряд слова, т. е. равен 000Сi. При арифметических операциях учитываются межразрядные переносы.

Таблица 17.7 - Перечень выполняемых АЛУ операций

|  |  |  |
| --- | --- | --- |
| S | Логические операции  (M=1) | Арифметико-логические функции (M=0) |
| 0 |  |  |
| 1 |  |  |
| 2 |  |  |
| 3 | *0* |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 | *A* |  |
| 11 | *AB* |  |
| 12 | 1 |  |
| 13 |  |  |
| 14 |  |  |
| 15 | *A* |  |

Шестнадцать логических операций позволяют воспроизводить все функции двух переменных. В логико-арифметических операциях встречаются и логи­ческие и арифметические операции одновременно.

Запись типа А\/В + АВ следует понимать так: вначале поразрядно выпол­няются операции инвертирования (В), логического сложения (А\/В) и ум­ножения (АВ), а затем полученные указанным образом два четырехразряд­ных числа складываются арифметически.

При операциях над словами большой размерности АЛУ соединяются друг с другом с организацией последовательных (рисунок 17.22а) или параллельных (рисунок 17.22б) переносов. В последнем случае совместно с АЛУ применяют микросхемы — блоки ускоренного переноса (СRU, Саrrу Unit), получающие от отдельных АЛУ функции генерации и прозрачности, а также входной перенос и вырабатывающие сигналы переноса

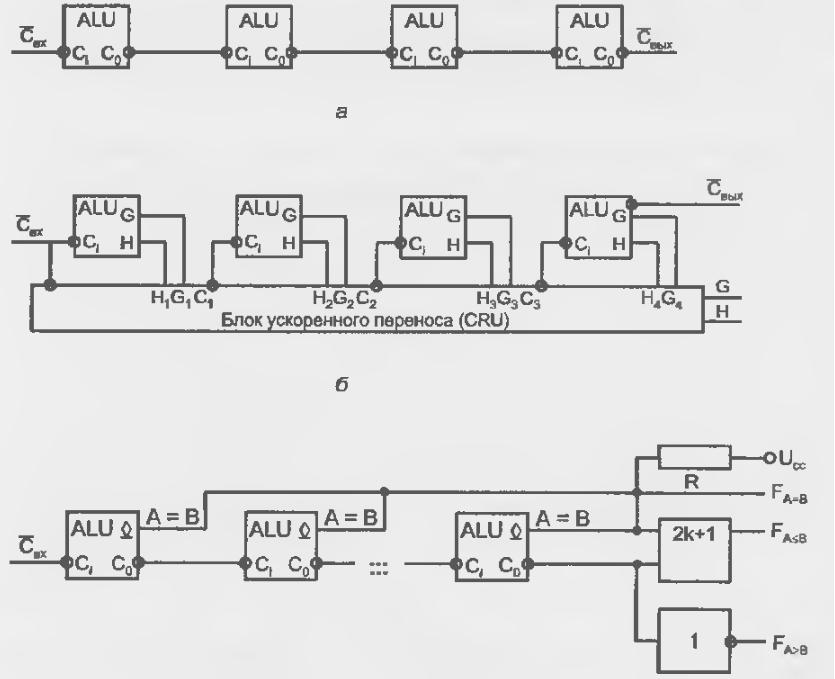


Рисунок 17.22 - Схемы наращивания АЛУ при последовательном (в) и параллельном (б) переносах и реализация функций компаратора для группы АЛУ (в)

Блок СКГ вырабатывает также функции генерации и прозрачности для всей группы обслуживаемых им АЛУ, что при необходимости позволяет органи­зовать параллельный перенос на следующем уровне (между несколькими группами из четырех АЛУ).

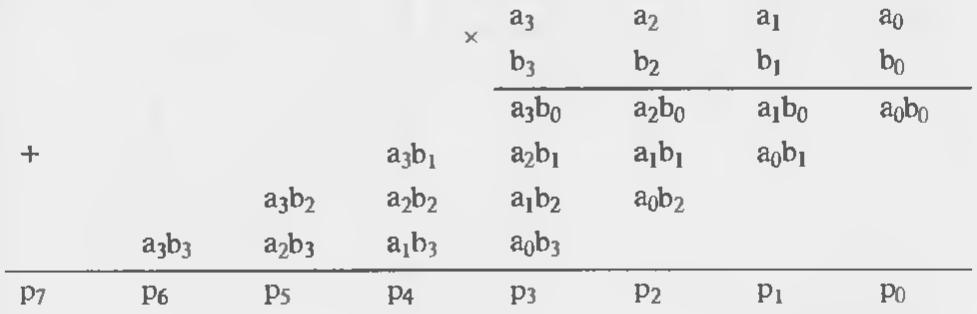
На рисунке 17.22, в показаны способы выработки сигналов сравнения слов для группы АЛУ. Выход сравнения на равенство выполняется по схеме монтаж­ной логики для выходов типа ОК. Комбинируя сигнал равенства слов с сиг­налом переноса на выходе группы при работе АЛУ в режиме вычитания, легко получить функции FA≥B и FA≤B. Если А < В, то при вычитании возника­ет заем из старшего разряда и FA≤B = 1. Если заем отсутствует (А>В), то по­дучим FA≥B = 1.

### **17.7 Матричные умножители**

Микросхемы множительных устройств появились в 1980-х годах, когда дос­тигнутый уровень интеграции позволил разместить на одном кристалле дос­таточно большое количество логических элементов.

Структура матричных умножителей тесно связана со структурой математических выражений, описывающих операцию умножения.

Пусть имеются два целых двоичных числа без знаков Аm = аm-1...ао и Вn = bn-1...bо, Их перемножение выполняется по известной схеме "умножения столбиком". Если числа четырехразрядные, т. е. m = n = 4, то



Произведение выражается числом Рm+n = Рm+n-1 Pm+n-2 … Ро.

Члены вида аibj, где i = 0... (m-1) и j = 0... (n-1) вырабатываются парал­лельно во времени конъюнкторами. Их сложение в столбцах, которое мож­но выполнять разными способами, составляет основную операцию для ум­ножителя и определяет почти целиком время перемножения.

Матричные перемножители могут быть просто множительными блоками (МБ) или множительно-суммирующими (МСБ), последние обеспечивают удобство наращивания размерности умножителя.

МСБ реализует операцию Р = Аm х Вn + Сm + Dn, т. е. добавляет к произ­ведению два слагаемых: одно разрядности m, совпадающей с разрядностью множимого, другое разрядности n, совпадающей с разрядностью множителя.