

ЛЕКЦИЯ №4

Регистры

Регистром (английское Register) называется устройство, предназначенное для ввода (записи), хранения и выдачи (считывания) цифрового кода, выраженного в виде двоичного числа (слова).

На схемах регистры обозначаются буквами RG. В отечественных сериях микросхем регистрам соответствуют буквы ИР. Все регистры делятся на две большие группы:

- параллельные регистры;
- регистры сдвига (или сдвиговые регистры).

В параллельных регистрах каждый из триггеров имеет свой независимый информационный вход (D) и свой независимый информационный выход. Тактовые входы (C) всех триггеров соединены между собой. В результате параллельный регистр представляет собой многоразрядный, многоходовый триггер.

В сдвиговых регистрах все триггеры соединены в последовательную цепочку (выход каждого предыдущего триггера соединен со входом D следующего триггера). Тактовые входы всех триггеров (C) объединены между собой. В результате такой триггер может рассматриваться как линия задержки, входной сигнал которой последовательно перезаписывается из триггера в триггер по фронту тактового сигнала C. Информационные входы и выходы триггеров могут быть выведены наружу, а могут и не выводиться в зависимости от функции, выполняемой регистром.

Параллельные регистры в свою очередь делятся на две группы:

- регистры, срабатывающие по фронту управляющего сигнала C (или тактируемые регистры);
- регистры, срабатывающие по уровню управляющего сигнала C (или стробируемые регистры).

Чаще всего в цифровых схемах используются регистры, управляемые фронтом (то есть тактируемые), однако и стробируемые регистры имеют свой круг задач, в которых их ничто не может заменить.

Основными элементами регистра являются двоичные ячейки памяти, роль которых играют триггеры различных типов.

Количество двоичных ячеек (триггеров) определяется количеством разрядов двоичного числа, предназначенного для ввода в регистр. Если разряды двоичного числа записываются в ячейки регистра одновременно, то его называют *параллельным* или *регистром памяти*. На рис. 1, а, б приведена функциональная схема простейшего двухразрядного параллельного регистра.

В исходном состоянии на шине «Ввод» имеем сигнал «О». Тогда на входах RS-триггеров устанавливается «1» независимо от комбинации сигналов (нулей или единиц) на входах x_1 и x_2 логических элементов И-НЕ, характеризующих информацию, предназначенную для записи.

Пусть на входе x_2 имеем логический нуль, а на входе x_1 — логическую единицу. Тогда при поступлении «1» на шину «Ввод» переключается элемент Э₁ и на S входе RS-триггера T₁ устанавливается «О». Триггер переключается в состояние $Q_1 = 1$. На входе RS-триггера T₂ остается «1» и соответственно на выходе T₂ $Q_2 = 0$.

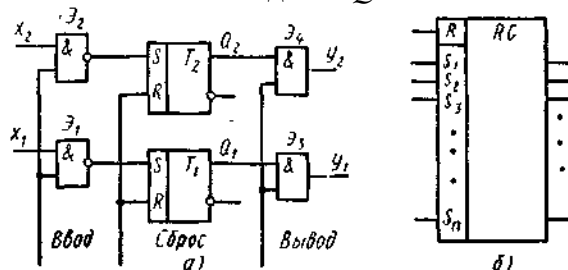


Рис.1. Параллельный регистр

Таким образом, в триггерах записывается и дополнительная двоичная информация, поступающая на входы регистра. На выходах регистра, работающего в режиме хранения, имеются логические нули. При подаче «1» на шину «Вывод» переключится элемент Э₃ и на выходе y₁ появится «1», а на выходе y₂ останется «О». Следовательно, происходит считывание записанной информации.

Параллельный регистр на D-триггерах

В параллельных регистрах информация записывается параллельно на все входы.

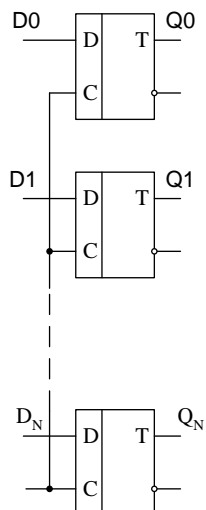


Рис. 2 – Принципиальная схема параллельного регистра

Каждый из триггеров записывает информацию, подаваемую на вход D, и сохраняет ее до прихода следующего синхрои импульса. Ниже показаны временные диаграммы параллельного регистра.

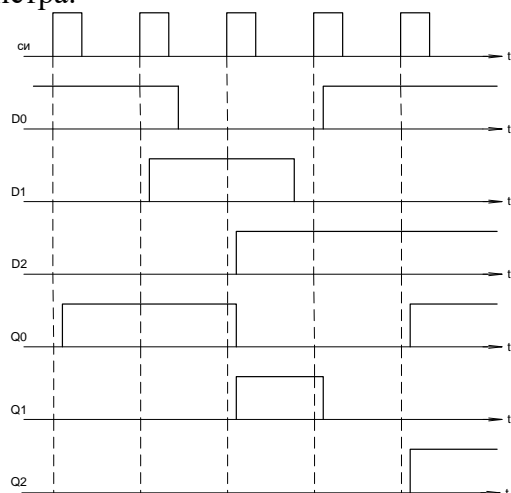


Рис. 3 – Временные диаграммы параллельного регистра

В стандартные серии входит несколько типов параллельных регистров, срабатывающих по фронту (рис. 9.4). Различаются они количеством разрядов, наличием или отсутствием инверсных выходов, наличием или отсутствием входа сброса (-R) или разрешения записи (-WE), а также типом выходных каскадов (2C или 3C) и соответственно наличием или отсутствием входа разрешения -EZ. Иногда на схемах тактовый вход C обозначается WR — сигнал записи в регистр.

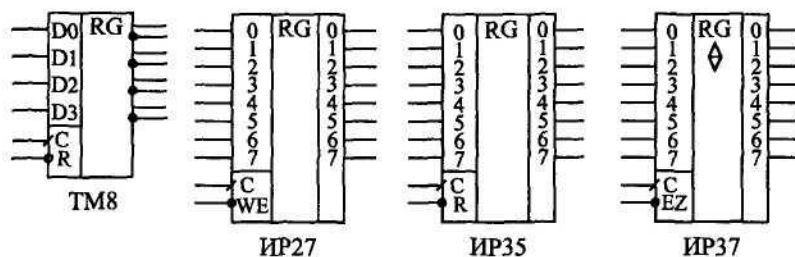


Рис. 4. Параллельные регистры стандартных серий, срабатывающие по фронту.

Большинство регистров имеют восемь разрядов, то есть запоминают один байт информации. Регистр TM8 в справочниках обычно называется счетверенным D-триггером (он и в наименовании несет буквы TM), хотя он вполне может рассматриваться и как регистр, так как тактовый вход C и вход сброса -R у всех четырех триггеров объединены между собой.

Таблицы истинности регистров очень просты и не отличаются принципиально от таблицы истинности D-триггеров. Отличие от триггеров появляется только в случае наличия у регистра дополнительных управляющих входов разрешения записи -WE и разрешения выхода -EZ. В

качестве примеров ниже приведены таблицы истинности регистра ИР27 и регистра ИР37 (табл. 1. и 2 соответственно). По переходу тактового сигнала С из 0 в 1 (положительный фронт) оба регистра записывают в себя входную информацию.

Таблица 1. Таблица истинности регистра ИР27

Входы			Выходы
-WE	C	D	Q
0	0→1	0	0
0	0→1	1	1
0	0	X	Не меняется
0	1	X	Не меняется
1	X	X	Не меняется

Таблица 2. Таблица истинности регистра ИР37

Входы			Выходы
-EZ	C	D	Q
0	0→1	0	0
0	0→1	1	1
0	0	X	Не меняется
0	1	X	Не меняется
1	X	X	Z

Все регистры, имеющие выход с тремя состояниями, обеспечивают повышенную нагрузочную способность. Задержка переключения регистров примерно соответствует задержке переключения триггеров. Все временные ограничения, накладываемые на входные сигналы в случае триггеров, справедливы и для входных сигналов регистров. Например, не должна быть слишком малой длительность сигнала С, а также не должна быть слишком малой задержка между установлением сигнала D и приходом положительного фронта сигнала С. Иначе работа регистра может быть нестабильной или даже неправильной.

Параллельные регистры, срабатывающие по уровню стробирующего сигнала (или, как их еще называют, регистры-защелки, английское Latch), можно рассматривать как некий гибрид между буфером и регистром. Когда сигнал на стробирующем входе единичный, такой регистр пропускает через себя входные информационные сигналы, а когда стробирующий сигнал становится равен нулю, регистр переходит в режим хранения последнего из пропущенных значений входных сигналов.

Применение таких регистров сильно ограничено, хотя иногда они довольно удобны. В некоторых схемах они могут успешно заменять регистры, срабатывающие по фронту, а в других схемах их применение вместо регистров, срабатывающих по фронту, недопустимо.

В стандартных сериях микросхем регистры, срабатывающие по уровню, представлены гораздо меньше, чем регистры, срабатывающие по фронту. На рис. 5 в качестве примеров показаны две микросхемы этого типа: 4-разрядного регистра ТМ7 и 8-разрядного регистра ИР22. Стробирующие входы С нередко на схемах обозначают Е (от английского Enable — разрешение) для того, чтобы не путать их с тактовыми входами D-триггеров.

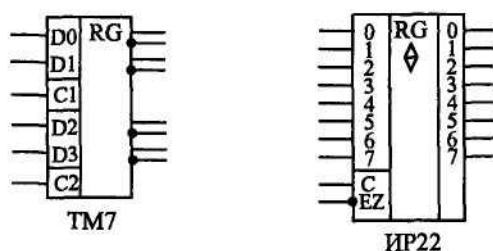


Рис. 5. Регистры, срабатывающие по уровню.

Микросхему ТМ7 (и близкую к ней ТМ5) часто называют набором триггеров, но ее можно рассматривать и как регистр. Микросхема состоит из четырех триггеров, стробирующие входы которых С соединены попарно, то есть можно говорить о двух двухразрядных регистрах-защелках. Входы С1 и С2 микросхемы управляют каждым двумя разрядами данных. Все триггеры имеют как прямые, так и инверсные выходы, что иногда очень удобно. Таблица истинности микросхемы ТМ7 приведена ниже (табл. 3).

Таблица 3. Таблица истинности регистра ТМ7

Входы		Выходы	
D	C	Q	-Q
0	1	0	1
1	1	1	0
0	0	Не меняется	
1	0	Не меняется	

При единице на входе С выходные сигналы повторяют входные, то есть регистр работает как обычный буфер с прямыми и инверсными выходами. При нуле на входе С на выходе регистра постоянно хранится та входная информация, которая была в момент прихода отрицательного фронта сигнала С. Однако говорить, что регистр ТМ7 срабатывает по отрицательному фронту сигнала С, неверно, так как информация на выходе меняется не только по этому фронту, но и в момент изменения входных сигналов при С = 1.

Регистр ИР22 отличается от ТМ7 тем, что имеет выходы с тремя состояниями (и соответственно вход разрешения всех выходов -EZ) и тем, что всеми восемью разрядами управляет один стробирующий сигнал С. Суть работы от этого не изменяется. При единице на входе С регистр работает как буфер-повторитель, а при нуле на входе С — хранит ту информацию, которая была на входе в момент отрицательного фронта сигнала С. Выходы у регистра ИР22 только прямые. Как и все регистры с тремя состояниями выхода, регистр ИР22 имеет повышенную нагрузочную способность. Таблица истинности регистра ИР22 приведена ниже (табл. 4).

Таблица 4. Таблица истинности регистра ИР22

Входы			Выход
-EZ	C	D	Q
0	1	1	1
0	1	0	0
0	0	X	Не меняется
1	X	X	Z

Величины задержек триггеров, срабатывающих по уровню, в 1,5—2 раза превышают задержки D-триггеров. Для правильной работы микросхем положительный импульс на входе С не должен быть слишком коротким, а задержка между изменением информации на входе D и отрицательным фронтом сигнала С не должна быть слишком малой. Информация на входе D не должна слишком быстро сниматься после отрицательного фронта сигнала С.

Последовательный регистр

Более экономичным является *последовательный* регистр (*регистр сдвига*, английское Shift Register), в котором разряды двоичного числа записываются и считываются последовательно во времени—разряд за разрядом.

На рис. 6 приведена схема трехразрядного последовательного регистра на D-триггерах. В крайнем слева триггере хранится информация, соответствующая старшему разряду числа, а в крайнем справа—младшему. В отличие от параллельного регистра для ввода многоразрядной информации в последовательный регистр используется только один вход D. На рисунке 9.6 введены следующие обозначения: DI – data input – вход данных, DO – data output – выход данных.

Поступление разрядов двоичного числа в регистр чередуется с подачей тактовых импульсов(синхроимпульсов или импульсов синхронизации) на тактовый вход С. Временные диаграммы, иллюстрирующие работу последовательного регистра, приведены на рис. 7.

Пусть в исходном состоянии на входе DI действует логическая единица, соответствующая младшему разряду числа. С поступлением тактового импульса его фронтом переключается триггер T_1 ($Q_0 = 1$), а состояния триггеров T_2 и T_3 остаются без изменения ($Q_1=0, DO=0$). Таким образом происходит запись информации в память триггера T_1 . После окончания действия логической единицы на входе состояния триггеров регистра остаются без изменения (информация хранится в первом триггере T_1) до прихода следующего тактового импульса. С приходом второго тактового импульса его фронтом переключаются триггеры T_1 и T_2 ($Q_0 = 0, Q_1 = 1$), а состояние триггера T_3 не изменяется ($DO = 0$). Информация, записанная в триггере T_1 , сдвигается вправо и записывается в триггере T_2 , а T_1 оказывается подготовленным к приходу следующего разряда двоичного числа. С приходом третьего тактового импульса переключаются триггеры T_2 и T_3 ($Q_2 = 0, Q_3 = 1$) и информация младшего разряда считывается на выходе триггера T_3 .

Во многих случаях при построении арифметических устройств необходимо осуществить сдвиг информации влево. Последовательные регистры, обладающие способностью осуществлять сдвиг информации в обоих направлениях, называются *реверсивными*.

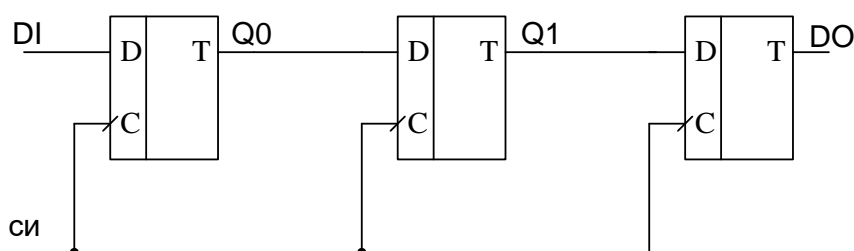


Рис. 6 – Принципиальная схема последовательного регистра

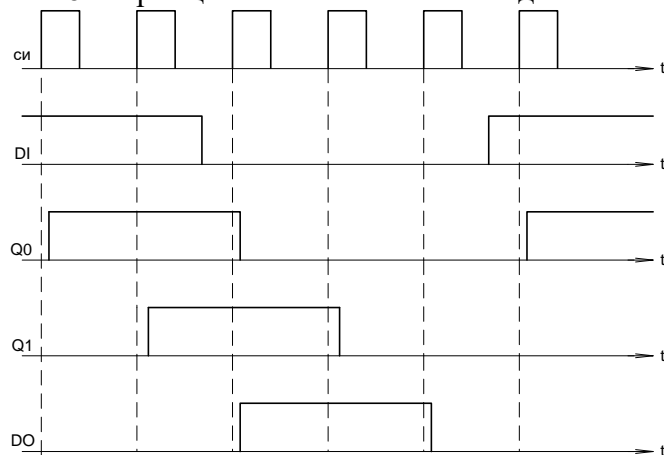


Рис. 7 – Временные диаграммы последовательного регистра

Реверсивный сдвиговый регистр позволяет сдвигать информацию вправо или влево, в зависимости от значения управляющих сигналов на дополнительном входе.

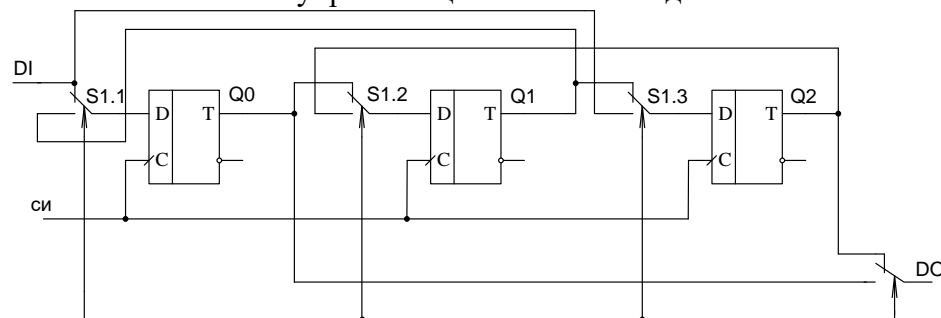


Рис. 8 – Схема реверсивного сдвигового регистра

Если $V = 0$, то $S1$ – в верхнем положении.

Если $V = 1$, то $S1$ – в нижнем положении.

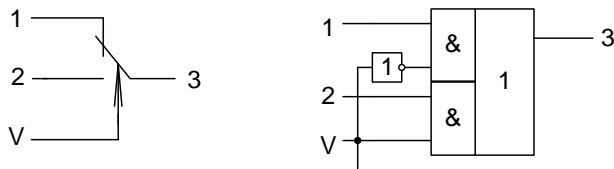


Рис. 9 – Схема ключа и его реализация на логических элементах

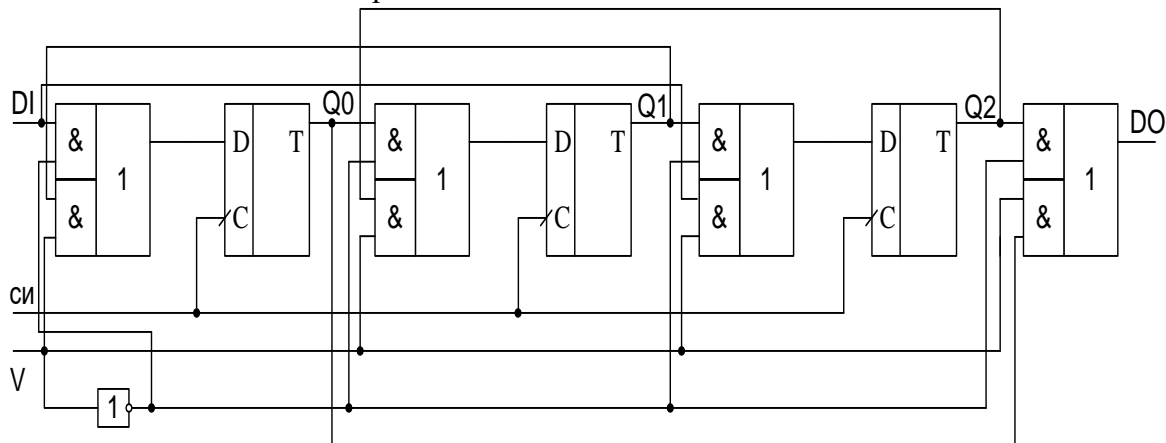


Рис. 10 – Принципиальная схема реверсивного сдвигового регистра

С названиями направлений сдвига в сдвиговых регистрах часто возникает путаница. Сдвиг бывает двух видов: вправо (основной режим, который есть у всех сдвиговых регистров) и влево (этот режим есть только у некоторых, реверсивных сдвиговых регистров). Названия эти отражают внутреннюю структуру регистров сдвига (рис. 6) и перезапись сигналов последовательно по цепочке триггеров. При этом триггеры, вполне естественно, нумеруются слева направо, например, от 0 до 7 (или от 1 до 8) для 8-разрядных регистров. В результате сдвиг информации регистром вправо представляет собой сдвиг в сторону разрядов, имеющих большие номера, а сдвиг информации регистром влево — это сдвиг в сторону разрядов, имеющих меньшие номера.

Однако, как известно, в любом двоичном числе слева расположены старшие разряды, а справа — старшие разряды. Поэтому сдвиг двоичного числа вправо будет сдвигом в сторону младших разрядов, а сдвиг влево — сдвигом в сторону старших разрядов. Это противоречие не чей-то злой умысел, просто так сложилось исторически, и об этом надо помнить разработчику цифровой аппаратуры.

В стандартные серии цифровых микросхем входит несколько типов сдвиговых регистров, отличающихся возможными режимами работы, режимами записи, чтения и сдвига, а также типом выходных каскадов (2C или 3C). Большинство регистров сдвига имеет восемь разрядов. На рис. 11 представлены для примера четыре типа микросхем регистров сдвига.

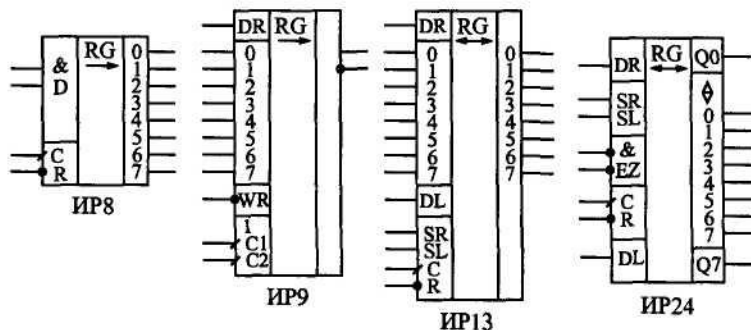


Рис. 11. Сдвиговые регистры.

Регистр IP8 — наиболее простой из регистров сдвига. Он представляет собой 8-разрядную линию задержки, то есть имеет только один информационный вход, на который подается последовательная сдвигаемая информация (точнее, два входа, объединенных по функции 2И), и восемь параллельных выходов. Сдвиг в сторону выходов со старшими номерами осуществляется по переднему фронту тактового сигнала C. Имеется также вход сброса -R, по нулевому сигналу на котором все выходы регистра сбрасываются в нуль. Таблица истинности регистра IP8 приведена ниже (табл. 5).

Таблица 5. Таблица истинности регистра сдвига IP8

Входы				Выходы			
-R	C	D1	D2	Q0	Q1	...	Q7
0	X	X	X	0	0	...	0
1	0	X	X	Не меняются			
1	1	X	X	Не меняются			
1	0→1	1	1	1	Q0	...	Q6
1	0→1	0	X	0	Q0	...	Q6
1	0→1	X	0	0	Q0	...	Q6

Регистр IP9 выполняет функцию, обратную регистру IP8. Если IP8 преобразует входную последовательную информацию в выходную параллельную, то регистр IP9 преобразует входную параллельную информацию в выходную последовательную. Однако суть сдвига не меняется, просто в регистре IP9 все внутренние триггеры имеют выведенные параллельные входы, и только один, последний триггер имеет выход (причем как прямой, так и инверсный). Запись входного кода в регистр производится по нулевому сигналу на входе -WR. Сдвиг осуществляется по положительному фронту на одном из двух тактовых входов C1 и C2, объединенных по функции 2ИЛИ. Имеется также вход расширения DR, сигнал с которого в режиме сдвига перезаписывается в младший разряд сдвигового регистра. Таблица истинности регистра IP9 приведена ниже (табл. 6).

Таблица 6. Таблица истинности регистра сдвига IP9

Входы			Функция
-WR	C1	C2	
0	X	X	Параллельная запись
1	1	X	Хранение
1	X	1	Хранение
1	0	0→1	Сдвиг
1	0→1	0	Сдвиг

Как и все остальные сдвиговые регистры, регистры IP8 и IP9 допускают каскадирование, то есть совместное включение для увеличения разрядности. На рис. 12 показано совместное включение трех регистров IP9. В результате объединения получается 24-разрядный сдвиговый регистр. При этом увеличение разрядности не приводит к увеличению задержки сдвига, так как тактовые входы всех используемых регистров объединяются параллельно. Входной 24-разрядный параллельный код преобразуется в выходной последовательный код.

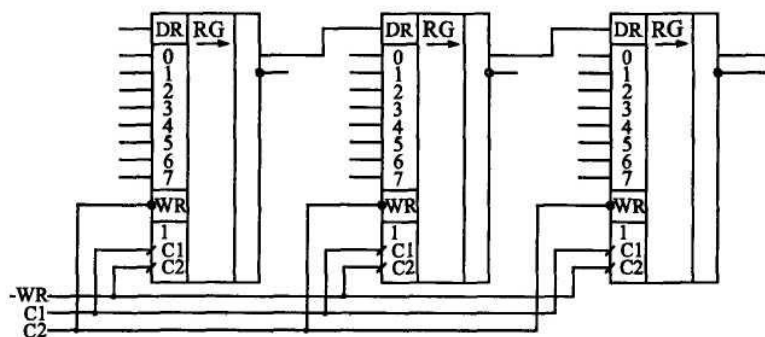


Рис. 12. Соединение регистров IP9 для увеличения разрядности.

Универсальный последовательно–параллельный регистр

Универсальный последовательно–параллельный регистр должен позволять:

- 1) Принимать информацию в последовательном или параллельном коде (в зависимости от сигнала управления).
- 2) Формировать на выходах параллельный код.
- 3) Формировать на выходе последовательность данных в виде последовательного кода, сдвиг которого возможен как в одном, так и в другом направлении (в зависимости от второго сигнала управления).

Для того чтобы построить универсальный регистр целесообразно использовать RS и D триггеры. Входы R и S используют для записи параллельного кода. Вход D – для построения сдвигового регистра.

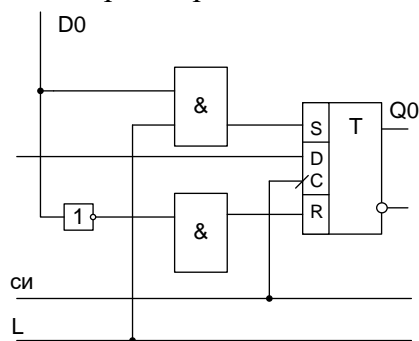


Рис.13 – Принципиальная схема универсального регистра

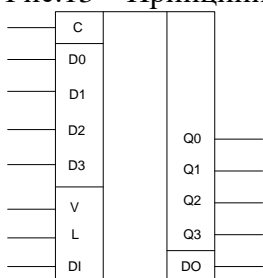


Рис. 14 – Условное обозначение универсального регистра

На рисунке 14 приняты следующие обозначения: C – вход синхронизации, D0, D1, D2 и D3 – входы параллельного кода, V – направление сдвига, L – разрешение записи параллельного кода, DI – вход последовательного кода, DO – выход последовательного кода; Q0, Q1, Q2 и Q3 – выходы параллельного кода.

Регистр ИР 13 соединяет в себе возможности регистров ИР8 и ИР9. Он имеет как восемь входов для параллельной записи, так и соответствующие им восемь выходов параллельной информации. Сдвиг осуществляется по положительному фронту тактового сигнала C, причем сдвиг возможен как в сторону старших разрядов (вправо), так и в сторону младших разрядов (влево). Для наращивания разрядности у регистра ИР 13 имеются последовательные информационные входы DR и DL, сигналы с которых вводятся соответственно в младший и в старший разряды. Предусмотрен сброс всех выходов регистра в нуль по нулевому сигналу на входе -R.

Режим работы регистра ИР 13 определяется двумя управляющими входами SR и SL. При единице на входе SR и нуле на входе SL по фронту сигнала C происходит сдвиг в сторону старших разрядов. При нуле на входе SR и единице на входе SL по фронту сигнала C происходит сдвиг в сторону младших разрядов. При обоих единичных сигналах на входах SR и SL по фронту сигнала C происходит параллельная загрузка информации в регистр. Все это видно из таблицы истинности регистра ИР13(табл.9.7).

Отметим, что регистр ИР 13 применяется заметно реже, чем более простые регистры ИР8 и ИР9, так как задач, в которых были бы нужны все возможности регистра ИР 13 не так уж много, а управление работой регистра ИР 13 довольно сложное.

Таблица 7. Таблица истинности регистра ИР 13

Входы				Функция
C	-R	SR	SL	
X	0	X	X	Сброс
0→1	1	1	0	Сдвиг вправо
0→1	1	0	1	Сдвиг влево
0→1	1	0	0	Хранение
0→1	1	1	1	Параллельная запись

Наконец, последний сдвиговый регистр, который мы рассмотрим подробнее, это регистр ИР24. По своим возможностям он близок к ИР 13, однако его главной особенностью является двунаправленная параллельная шина данных. То есть одни и те же выводы микросхемы используются как для параллельной записи информации в регистр, так и для параллельного чтения информации из регистра. При этом двунаправленные выводы данных имеют повышенную нагрузочную способность. Это позволяет легко сопрягать регистр ИР24 с многоразрядными микросхемами памяти и с двунаправленными буферами. Поэтому применяется данный регистр чаще, чем ИР 13.

Регистр ИР24 обеспечивает сдвиг информации в обоих направлениях. Имеются входы расширения DR и DL, а также выходы расширения Q0 и Q7, что позволяет легко наращивать разрядность. Отличие выходов Q0 и Q7 от нулевого и седьмого разрядов данных состоит в том, что Q0 и Q7 — однонаправленные, то есть в любом режиме работы выдают информацию с выходов внутренних триггеров младшего и старшего разрядов. Тактируется регистр положительным фронтом сигнала С. Предусмотрен сброс регистра нулевым сигналом на входе -R.

Режим работы микросхемы определяется сигналами на управляющих входах SR и SL.

При единичном сигнале на SR и нулевом сигнале на SL по положительному фронту сигнала С происходит сдвиг информации вправо (в сторону разрядов с большими номерами). Запись в разряд 0 производится при этом со входа расширения DR.

При единичном сигнале на SL и нулевом сигнале на SR по положительному фронту сигнала С происходит сдвиг информации влево (в сторону разрядов с меньшими номерами). Запись в разряд 7 производится при этом со входа расширения DL.

При обоих нулях на входах SR и SL регистр переходит в режим хранения. Во всех этих случаях разряды данных работают как вход или как выход в зависимости от сигналов -EZ.

При обеих единицах на входах SR и SL по положительному фронту С в регистр записывается параллельный код, причем разряды данных переходят в состояние приема независимо от сигналов -EZ. Таблица истинности регистра ИР24 приведена ниже (табл. 8).

Таблица 8. Таблица истинности регистра ИР24

Входы				Функция
-R	C	SR	SL	
0	X	X	X	Сброс
1	0→1	1	0	Сдвиг вправо
1	0→1	0	1	Сдвиг влево
1	0→1	1	1	Параллельная запись
1	X	0	0	Хранение

Объединяя два регистра ИР24, легко получить 16-разрядный сдвиговый регистр с сохранением всех возможностей одной микросхемы (рис. 15). Точно так же можно объединять и большее количество микросхем.

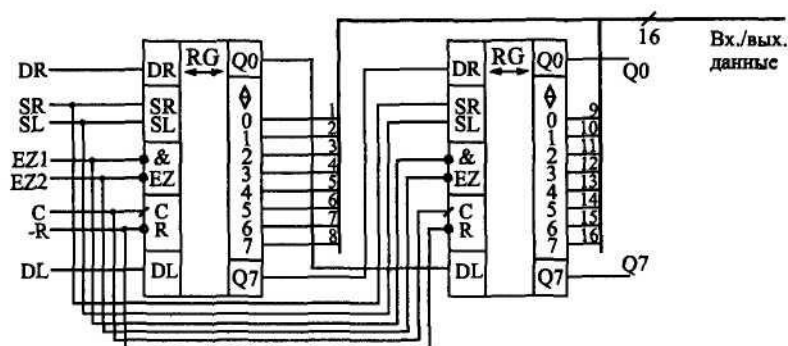


Рис. 15. Объединение регистров ИР24 для увеличения разрядности.

Практическое применение регистров

Одно из основных применений регистров состоит в хранении требуемого кода в течение нужного времени. Если для работы остальной части схемы необходимо иметь входной код, который можно легко изменять, то для этого как раз подходит регистр.

На рис. 16 показана типичная схема включения регистра для хранения кода и временная диаграмма его работы. Код на входе регистра может изменяться произвольным образом, но в тот момент, когда этот код принимает необходимое значение, на вход С триггера подается синхросигнал (строб), который записывает код в регистр. Этот код будет храниться в регистре до прихода следующего строба. Причем важно и то, что все разряды выходного кода регистра будут переключаться одновременно даже в том случае, когда разряды входного кода переключаются не одновременно. Главное, чтобы к приходу положительного фронта строба (сигнала С) все разряды входного кода приняли нужное, устойчивое значение.

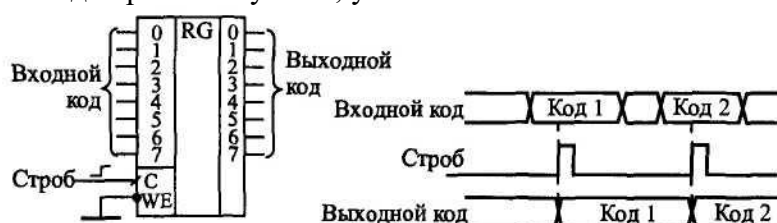


Рис. 16. Хранение кода в параллельном регистре.

Еще одно важнейшее применение регистров связано с запоминанием нескольких последовательных значений изменяющегося входного кода. Это позволяет, например, сравнивать предыдущее значение кода с последующим значением этого же кода или производить арифметические операции над несколькими последовательными значениями одного и того же кода. То есть регистр в данном случае выступает как элемент линии задержки, хранящей в себе историю поведения входного кода.

Для примера на рис. 17 показана схема вычисления разности двух последовательных значений входного кода. Такая задача возникает в частности при цифровой обработке аналоговых сигналов. Последовательные значения входного 4-разрядного кода сопровождаются тактовым сигналом, по положительному фронту которого производится запись в два последовательно включенных регистра. Когда на выходе регистра RG1 присутствует N-е значение входного кода, на выходе регистра RG2 будет (N-1)-е значение этого же кода.

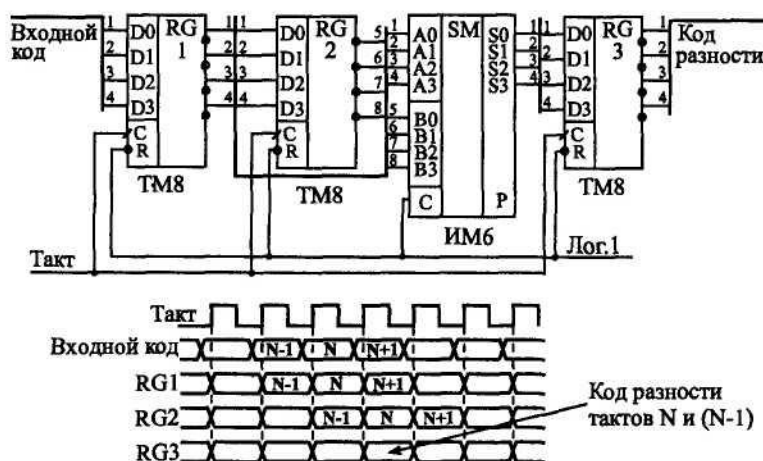


Рис. 17. Схема вычисления разности значений кодов в двух последовательных тактах.

Подавая эти два кода с выходов регистров на 4-разрядный сумматор, включенный в режиме вычитания, мы получаем на выходе сумматора код разности между N-м значением и (N-1)-М значением. В данном случае очень удобен регистр TM8, имеющий инверсные выходы. Для обеспечения строго одновременного изменения выходных сигналов сумматора можно включить дополнительный выходной регистр RG3, тактируемый тем же самым общим тактовым сигналом. Правда, код разности при этом будет задержан на один такт.

Регистры также широко используются для организации конвейерной обработки, позволяющей существенно повысить тактовую частоту работы схемы. Ускорение при этом достигается за счет распараллеливания работы нескольких последовательно включенных узлов схемы.

Пусть, например, последовательность входных кодов, следующих с периодом T , поступает на вход цепочки из двух узлов, производящих обработку или преобразование этих кодов (рис. 18). Узлы эти могут представлять собой комбинационные микросхемы (например сумматоры) или более сложные устройства, включающие в себя микросхемы счетчиков или микросхемы памяти. Главное состоит в том, что выходные сигналы этих узлов выставляются не мгновенно, а в течение какого-то конечного времени, величина которого определяется внутренним строением узла. Пусть задержка установления выходного кода первого узла равняется t_1 , а задержка установления выходного кода второго узла составляет t_2 . Очевидно, что период следования входных кодов T не должен быть меньше, чем сумма этих двух задержек:

$$T > t_1 + t_2$$

Иначе код на выходе цепочки может никогда не принять устойчивого значения, так как переходный процесс предыдущего такта будет сменяться переходным процессом следующего такта. То есть быстродействие узлов накладывает жесткое ограничение на тактовую частоту.

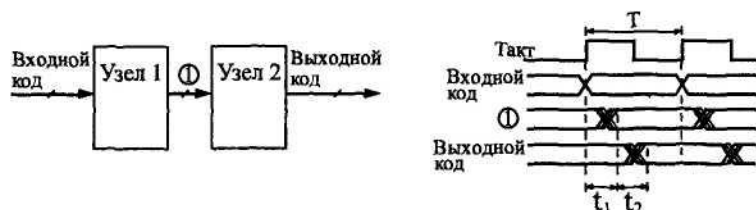


Рис. 18. Работа последовательной цепочки двух узлов.

Однако можно обойти это ограничение, если воспользоваться принципом конвейера, заставить узлы работать не последовательно, а параллельно. Это достигается включением между узлами регистра, тактируемого входным тактовым сигналом. Еще один регистр целесообразно включить на входе второго узла, что обеспечит длительность устойчивого кода на выходе всего устройства, равную длительности периода тактового сигнала T (рис. 19). В результате ограничение на период тактового сигнала становится более мягким: T не должно быть меньше максимальной из двух величин t_1 и t_2 с добавлением времени задержки регистра:

$$T > \max \{t_1, t_2\} + t_{RG}$$

То есть к следующему фронту тактового сигнала должен закончить свою работу самый медленный из узлов, и тогда его выходной код будет записан в регистр правильно.

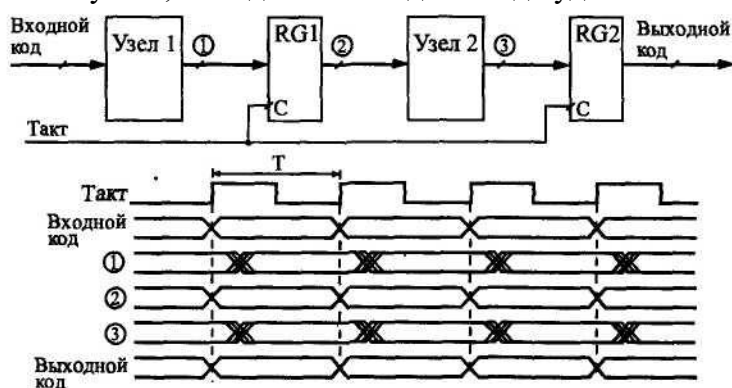


Рис. 19. Конвейерная обработка с помощью регистров.

Точно так же можно построить конвейер на любое количество последовательно включенных узлов. Конечно, в результате введения конвейера происходит задержка выполнения полной функции устройства на число тактов, равное числу введенных регистров. Однако в том случае, когда необходимо обрабатывать большие последовательности входных кодов, эта задержка наблюдается только один раз — в самом начале последовательности, а затем уже она не имеет значения.

Регистры могут также применяться в составе вычислителей, выполняя функцию накопителя результата вычислений. В данном случае мы уже имеем дело с более сложной обработкой информации, чем в случае чисто комбинационных схем. С каждым тактом в регистре обновляется

содержимое, являющееся результатом математической обработки входного кода и результата предыдущего вычисления.

Преобразование параллельного кода в последовательный

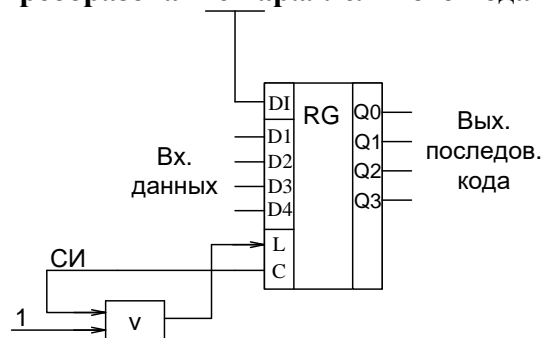


Рисунок 20. Преобразование параллельного кода в последовательный

D1 – вход последовательных данных,
D0...D3 – вход параллельного кода,
L – загрузка параллельного кода,
C – синхрои́мпульс для сдвига информации в регистре,
V – схема деления частоты и синхрои́мпульса,
1 – синхрои́мпульс параллельный данным от источника данных.

Преобразователь параллельного кода в последовательный с использованием мультиплексора

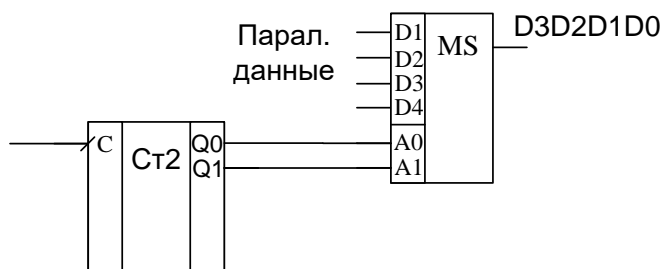


Рисунок 21 – Преобразователь параллельного кода в последовательный с использованием мультиплексора

На вход мультиплексора данные необходимо подавать на протяжении четырех периодов следующих синхрои́мпульсов, за это время на выходе мультиплексора по очереди появятся последовательности данных. После завершения считывания можно подать другую информацию. Параллельный код преобразуют в последовательный, чтобы уменьшить количество линий ghb передаче информации от одного устройства к другому.

Преобразование последовательного кода в параллельный

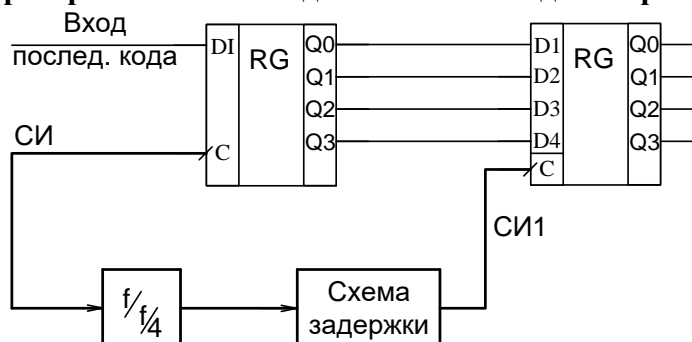


Рисунок 22 – Преобразование последовательного кода в параллельный

СИ1 формируется через каждые четыре импульса СИ. За четыре такта вся информационная последовательность кода вводится в регистр, как только она будет введена (сразу после окончания четвертого синхрои́мпульса) можно осуществить запись во второй регистр на выходе которого и будет сформирован параллельный код. Длина регистра определяется длиной кода.

Передача данных в последовательном коде

В качестве примера на рис. 23 показана простейшая схема передачи цифровой информации в последовательном коде по двум линиям: информационной и синхронизирующей. Такая передача позволяет сократить количество соединительных проводов, а также упростить защиту передаваемых данных от действия внешних электромагнитных помех, правда, ценой снижения скорости передачи.

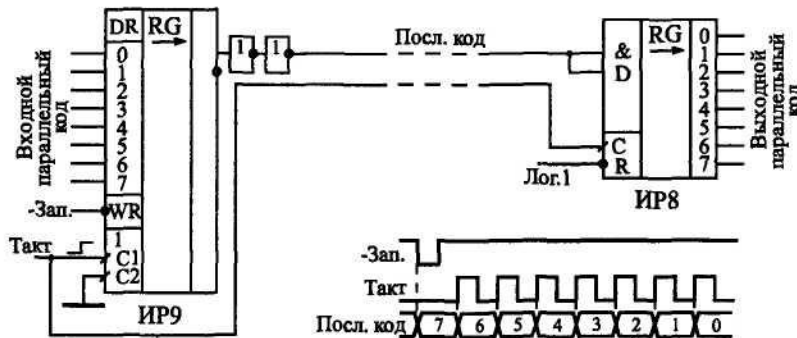


Рис. 23. Последовательная передача информации с помощью регистров сдвига.

На передающем конце (слева на рисунке) с помощью сдвигового регистра ИР9 входной параллельный 8-разрядный код преобразуется в последовательность разрядов данных, следующих с частотой тактового сигнала. На приемном конце (справа на рисунке) с помощью сдвигового регистра ИР8 эта последовательность разрядов данных снова преобразуется в параллельный код. Оба регистра тактируются одним и тем же тактовым сигналом, который передается по линии связи параллельно с последовательностью данных. Для увеличения надежности передачи информационный сигнал дополнительно задерживается относительно фронта тактового сигнала с помощью цепочки из двух инверторов.

Первый бит последовательного входа (со входа 7 регистра ИР9) начинает передаваться с началом сигнала записи -Зап. Следующие разряды передаются с каждым следующим положительным фронтом тактового сигнала С. Последним передается сигнал со входа 0. В регистр ИР8 разряды последовательного кода записываются в том же самом порядке, в каком они были в регистре ИР9. По окончании передачи первый переданный сигнал данных, окажется в разряде 7 шины данных регистра ИР 8, а последний переданный сигнал данных — в разряде 0.

Линии задержки на основе регистров сдвига

Следующее применение сдвиговых регистров состоит в организации всевозможных линий задержек, особенно имеющих значительное количество каскадов. С помощью сдвиговых регистров можно обеспечить задержку любого входного сигнала на целое число тактов. Правда, надо учитывать, что длительность входного сигнала (и любого его элемента) будет также передаваться по линии задержки с точностью до одного такта. Такие линии задержки могут применяться для сравнения нескольких последующих тактов входного сигнала, для выполнения арифметических операций с несколькими тактами входного сигнала, для других подобных целей. Работа линии задержки на регистре сдвига иллюстрируется рис. 24.

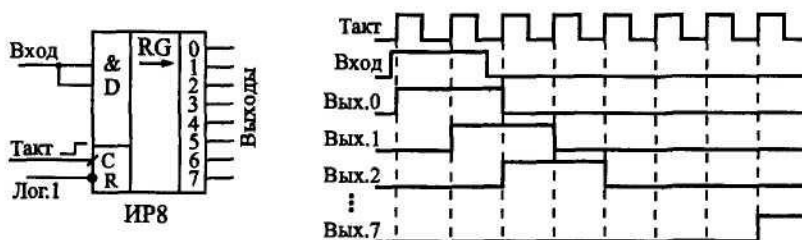


Рис. 24. Линия задержки входного сигнала на регистре сдвига.

Генератор импульсов программируемой длительности

Сдвиговые регистры могут также применяться для формирования импульсов заданной длительности, причем длительность импульса может задаваться управляющим кодом, то есть быть программно управляемой. На рис. 25 приведена возможная схема такого формирователя.

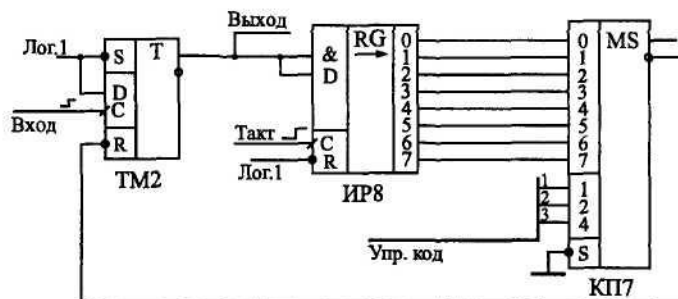


Рис. 25. Формирователь импульсов с длительностью, задаваемой управляющим кодом.

В исходном состоянии (до прихода положительного фронта входного сигнала) триггер сброшен в нуль, на всех выходах регистра сдвига нули, на инверсном выходе мультиплексора единица. На мультиплексор подан управляющий код, определяющий длительность выходного сигнала. При поступлении положительного фронта входного сигнала триггер перебрасывается в единицу (начало действия выходного импульса), и этот единичный сигнал начинает последовательно сдвигаться регистром сдвига по каждому фронту тактового сигнала.

Пусть управляющий код равен 5. Тогда в тот момент, когда на выходе 5 сдвигового регистра появится единица, она будет передана на выход мультиплексора КП7 с инверсией. При этом нулевой сигнал на входе \bar{R} триггера сбросит триггер в нуль, что соответствует окончанию действия выходного импульса.

Таким образом, длительность выходного сигнала будет определяться управляющим кодом. Погрешность установки этой длительности равна одному периоду тактового сигнала и зависит от временного сдвига между фронтом входного сигнала и фронтом ближайшего к нему тактового импульса. Чем больше длительность выходного сигнала, тем меньше относительная погрешность установки его точности. Например, при управляющем коде 0 длительность выходного сигнала может быть от 0 до T, где T — период тактового сигнала. А при управляющем коде 7 длительность выходного сигнала будет от 7T до 8T. При этом мы не учитываем задержек триггера, сдвигового регистра и мультиплексора.

Сдвиговые регистры могут также использоваться для умножения и деления двоичных чисел на 2^n , где n — целое число, большее нуля. Сдвиг двоичного числа вправо (в сторону младших разрядов) на один разряд равносильен делению на 2. Сдвиг двоичного числа влево (в сторону старших разрядов) на один разряд равносильен умножению на 2. Для того чтобы сдвиговый регистр умножал и делил двоичный код, надо всего лишь записать этот код в регистр и сдвинуть его нужное количество раз вправо или влево. Наиболее удобен для этого регистр ИР13. При этом необходимо, чтобы в освободившиеся разряды вводились нули, то есть на входы расширения DR и DL регистра нужно подать нулевые сигналы.

Генераторы случайных последовательностей

Наконец, последнее применение сдвигового регистра, которое мы рассмотрим, — это генератор случайной последовательности сигналов или случайной последовательности кодов. Строго говоря, последовательности будут не полностью случайные, а квазислучайные, то есть будут периодически повторяться, но период этот довольно большой. Случайные последовательности сигналов и кодов широко применяются в тестирующей аппаратуре, в генераторах шума, в логических игровых устройствах.

Задача состоит в том, чтобы выходной сигнал или код менял свое состояние случайно (или почти случайно). Сигнал должен случайно переключаться из 0 в 1 и из 1 в 0, а код должен случайно принимать значения из диапазона от 0 до $(2^N - 1)$, где N — число разрядов кода (например, от 0 до 255 при 8-разрядном коде). Псевдослучайные последовательности имеют то преимущество перед истинно случайными, что они предсказуемые и периодические, но в этом же и их недостаток.

Структура генератора квазислучайной последовательности на сдвиговом регистре очень проста (рис. 26). Она представляет собой регистр сдвига с параллельными выходами (например, ИР8), несколько (минимум два) выходных сигналов которого, объединены с помощью элемента Иключающее ИЛИ, с выхода которого сигнал подается на вход регистра, замыкая схему в кольцо. Схема тактируется сигналом с частотой f_T .

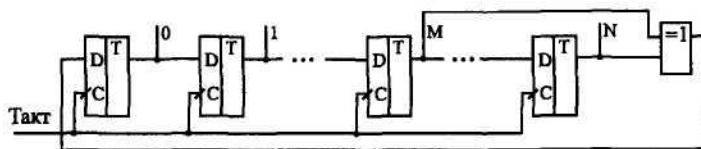


Рис. 26. Структура генератора псевдослучайной последовательности.

Выбор номеров разрядов для подключения обратной связи представляет собой непростую задачу, но существуют справочные таблицы, в которых они приведены. В любом случае одна из точек подключения — выход старшего разряда. В табл. 9 приведены точки подключения обратной связи для регистров сдвига с разным количеством разрядов N (номера разрядов считаются от нуля).

Таблица 9. Точки подключения обратной связи

N	7	8	15	16	24	31
Выходы	6, 5	7, 6, 4, 2	14, 13	15, 13, 12, 10	23, 22, 21, 16	30, 17

Из таблицы видно, что выгоднее брать число разрядов не кратное 8, например, 7, 15 или 31. В этом случае для обратной связи используются всего лишь два выхода, то есть достаточно одного двухвходового элемента Иключающее ИЛИ.

Период выходной последовательности генератора составляет $(2^N - 1)$ тактов, где N — количество разрядов регистра сдвига. За это время каждое из возможных значений выходного кода (кроме одного) встречается один раз. Количество единиц в выходном сигнале больше количества нулей на единицу.

Выходной код 000...0 представляет собой запрещенное состояние, так как он блокирует работу генератора, воспроизводя сам себя снова и снова. Но в то же время получить такой нулевой код может только сам из себя, поэтому достаточно обеспечить, чтобы его не было при включении питания схемы.

Частоты в спектре выходного сигнала будут следовать с интервалом $f_T / (2^N - 1)$, а огибающая спектра будет практически постоянной до частоты $0,25f_T$, то есть шум до этой частоты можно считать белым (спад в 3 дБ происходит на частоте $0,45 f_T$).

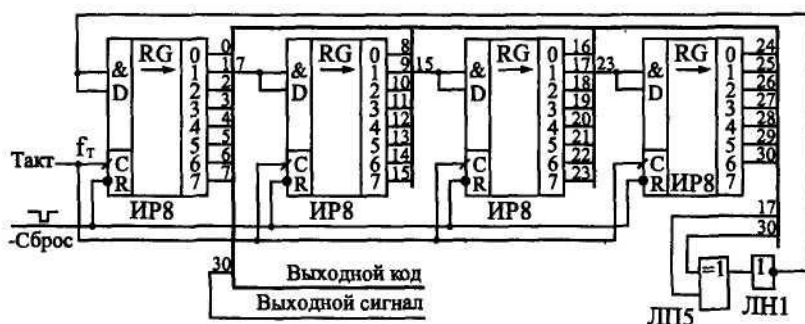


Рис. 27. 31-разрядный генератор псевдослучайной последовательности на регистрах сдвига.

На рис. 27 показана практическая схема генератора псевдослучайной последовательности на 31-разрядном сдвиговом регистре. Обратная связь осуществляется с выходов 30 и 17 регистра через двухвходовой элемент Иключающее ИЛИ с инвертором. Из-за применения инвертора запрещенным состоянием генератора является код 1111...1 (а не код 000...0), который в данном случае исключается очень просто — начальным сбросом регистров в нуль при включении питания по сигналу -Сброс. Генератор выдает квазислучайную последовательность 31-разрядных кодов со всех выходов регистра, а также квазислучайную последовательность нулей и единиц на любом из выходов регистра. Такой генератор использовала известная фирма Hewlett-Packard в своем генераторе шума.

