

Е.Г. Трофимов

ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СЕТИ

**ЛЕКЦИОННЫЙ КУРС
ПРАКТИЧЕСКИЕ ЗАНЯТИЯ
ТЕСТОВЫЕ ЗАДАНИЯ**

Учебно-методическое пособие



• ФЛИНТА •

УДК 004.7
ББК 32.971.3
Т76

Трофимов Е.Г.

Т76 Вычислительные машины, сети. Лекционный курс. Практические занятия. Тестовые задания. [Электронный ресурс] : учебно-методическое пособие / Е.Г. Трофимов. — 2-е изд., стер. — М. : ФЛИНТА, 2017. — 385 с.

ISBN 978-5-9765-3734-7

Учебно-методическое пособие посвящено организации и проведению лабораторных работ по курсу «Вычислительные машины, сети» студентами гуманитарных направлений. Может быть использовано для заочной формы обучения, поскольку включает теоретический материал и рекомендации по работе с лабораторной установкой, тестовые задания. Учебный материал, предоставленный в издании, соответствует стандартам.

УДК 004.7
ББК 32.971.3

ISBN 978-5-9765-3734-7

© Трофимов Е.Г., 2017
© Издательство «ФЛИНТА», 2017

СОДЕРЖАНИЕ

ЛЕКЦИОННЫЙ КУРС «ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СЕТИ»	6
Раздел I. ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ	6
Тема 1. Основные функциональные элементы ЭВМ, часть 1	6
Дешифратор	6
Шифратор	7
Тема 2. Основные функциональные элементы ЭВМ, часть 2	15
Счетчики	15
Регистры	16
Тема 3. Арифметико-логическое устройство	20
Тема 4. Устройство управления	24
Схемное устройство управления	24
Структурная схема микропрограммного устройства управления	27
Тема 5. Запоминающие устройства	29
Классификация запоминающих устройств	29
Тема 6. Режимы адресации и форматы команд 16-разрядного процессора	32
Тема 7. Кодирование команд (часть 1)	40
Тема 8. Кодирование команд переходов (часть 2)	41
Тема 9. Оценка влияния структуры программы на время ее выполнения	48
Тема 10. Взаимодействие основных узлов и устройств персонального компьютера при автоматическом выполнении команды. Архитектура 32-разрядного микропроцессора	50
Архитектура 32-разрядного микропроцессора	52
Тема 11. Конвейерная организация работы процессора	54
Оценка производительности идеального конвейера	54
Конфликты в конвейере и способы минимизации их влияния на производительность процессора	55
Тема 12. Организация работы мультипрограммных ЭВМ	60
Основные характеристики мультипрограммного режима работы ЭВМ	60
Тема 13. Дисциплины распределения ресурсов мультипрограммной ЭВМ	65
Основные режимы работы мультипрограммной ЭВМ	67
Тема 14. Система прерываний	70
Организация обработки прерываний в ЭВМ	70
Обработка прерываний в персональной ЭВМ	73
Тема 15. Система управления памятью	75
Организация распределения памяти в ЭВМ	75
Тема 16. Система управления памятью в персональной ЭВМ	80
Тема 17. Защита памяти в мультипрограммных ЭВМ	84
Средства защиты памяти в персональной ЭВМ	86
Тема 18. Ввод-вывод информации	89
Раздел II. ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ	94
Тема 1. Определение локальных сетей и их топология	94
История компьютерной связи	94
Определение локальной сети	96
Топология локальных сетей	99
Тема 2. Типы линий связи локальных сетей	109
Кабели на основе витых пар	110
Коаксиальные кабели	114
Оптоволоконные кабели	116
Бескабельные каналы связи	118
Тема 3. Подключение линий связи и коды передачи информации	121
Согласование, экранирование и гальваническая развязка линий связи	121
Кодирование информации в локальных сетях	125

Тема 4. Пакеты, протоколы и методы управления обменом.....	134
Назначение пакетов и их структура.....	134
Адресация пакетов.....	139
Методы управления обменом.....	140
Тема 5. Модель OSI. Нижние уровни	150
Эталонная модель OSI	150
Аппаратура локальных сетей	154
Тема 6. Модель OSI. Верхние уровни	160
Стандартные сетевые протоколы.....	160
Стандартные сетевые программные средства	166
Тема 7. Старейшие стандартные сети	172
Сети Ethernet и Fast Ethernet	173
Сеть Token-Ring.....	176
Сеть Arcnet	184
Тема 8. Скоростные и беспроводные сети	188
Сеть FDDI.....	188
Сеть 100VG-AnyLAN.....	194
Сверхвысокоскоростные сети	199
Беспроводные сети	203
Тема 9. Защита информации в локальных сетях	206
Классификация средств защиты информации.....	207
Классические алгоритмы шифрования данных.....	209
Стандартные методы шифрования и криптографические системы	211
Программные средства защиты информации.....	213
Тема 10. Алгоритмы сети Ethernet/Fast Ethernet.....	215
Метод управления обменом CSMA/CD	215
Использование помехоустойчивых кодов для обнаружения ошибок в сети.....	220
Тема 11. Стандартные сегменты Ethernet.....	226
Аппаратура 10BASE5	226
Аппаратура 10BASE2.....	229
Аппаратура 10BASE-T.....	232
Аппаратура 10BASE-FL.....	235
Тема 12. Стандартные сегменты Fast Ethernet.....	239
Аппаратура 100BASE-TX	239
Аппаратура 100BASE-T4.....	241
Аппаратура 100BASE-FX	243
Автоматическое определение типа сети (Auto-Negotiation).....	244
Тема 13. Оборудование Ethernet и Fast Ethernet	248
Адаптеры Ethernet и Fast Ethernet	248
Репитеры и концентраторы Ethernet и Fast Ethernet.....	252
Коммутаторы Ethernet и Fast Ethernet.....	257
Мосты и маршрутизаторы Ethernet и Fast Ethernet	261
Тема 14. Выбор конфигурации сетей Ethernet и Fast Ethernet	268
Выбор конфигурации Ethernet.....	268
Выбор конфигурации Fast Ethernet.....	271
Тема 15. Методика и начальные этапы проектирования сети	274
Исходные данные построения сети	275
Выбор размера и структуры сети.....	276
Выбор оборудования	279
Выбор программных средств	283
Тема 16. Выбор локальной сети с учетом ее стоимости, проектирование кабельной системы, оптимизация и отладка сети.....	286

Выбор локальной сети с учетом ее стоимости	286
Проектирование кабельной системы	288
Тема 17. Формулы Шеннона и типы линий передачи, в которых используются модемы.	
Структура модема, методы модуляции, стандарты и программные средства для модемов.....	297
Формулы Шеннона для непрерывного и дискретного каналов	297
Типы линий передачи, в которых используются модемы (варианты решения проблемы "последней мили").....	300
Методы модуляции, используемые в высокоскоростных модемах	307
Особенности стандартов V.34, V.90 и V.92	311
Классификация модемов.....	313
Программные средства для модемов.....	315
Тема 18. Глобальные вычислительные сети. Сеть Internet.....	317
III. ПРАКТИЧЕСКИЕ ЗАНЯТИЯ ПО КУРСУ «ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ».....	341
1. Построение ЗУ с заданной организацией	341
2. Кодирование линейных команд	345
3. Кодирование команд переходов.....	351
4. Оценка влияния структуры программы на время ее выполнения	357
5. Взаимодействие основных узлов и устройств персонального компьютера при автоматическом выполнении команды	360
6. Оценка производительности конвейера	362
7. Оценка конфигурации Ethernet, расчет временных характеристик выбранной конфигурации сети. Расчет по модели 2	364
8. Оценка максимального размера сети Ethernet. Расчет по модели 2	367
9. Выбор конфигурации Fast Ethernet. Правила модели 2	371
IV. ВОПРОСЫ К ТЕСТАМ ПО КУРСУ «ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ И СЕТИ» .	374
ОБЩИЕ МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ТЕСТОВЫМ ЗАДАНИЯМ	374
1. Вопросы к тесту по теме «Вычислительные машины»	374
2. Вопросы к тесту по теме «Вычислительные сети»	379

ЛЕКЦИОННЫЙ КУРС «ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СЕТИ»

Раздел I. ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ

Тема 1. Основные функциональные элементы ЭВМ, часть 1

Рассмотрим некоторые схемы, составляющие основу элементной базы любого компьютера.

Дешифратор

Дешифратором называется комбинационная схема, имеющая n входов и 2^n выходов и преобразующая двоичный код на своих входах в унитарный код на выходах. *Унитарным* называется двоичный код, содержащий одну и только одну единицу, например, 00100000. Условно-графическое обозначение (УГО) *дешифратора* на три входа приведено на рис. 1.1.1.

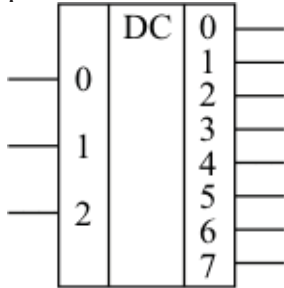


Рис. 1.1.1. Условно-графическое обозначение трехвходового дешифратора

Номер разряда, в котором устанавливается "1" на выходе *дешифратора*, определяется кодом на его входах. Ниже приведена таблица истинности трехвходового *дешифратора* (таблица 1.1.1).

Таблица 1.1.1.

Входы		Выходы							
2	1	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0
0	1	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1

Реализация *дешифратора* в одноэлементном базисе "Штрих Шеффера" достаточно проста, так как таблица истинности для любого выхода имеет только одну единицу. На рис. 1.1.2 представлена схема формирования сигнала на одном из выходов *дешифратора* (сигнал f_5 на выходе 5):

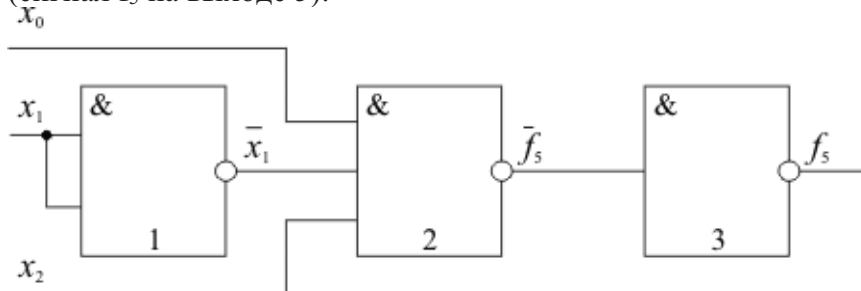


Рис. 1.1.2. Схема формирования сигнала на выходе 5 трехвходового дешифратора

Из представленной схемы видно, что фактически логику преобразования выполняет лишь элемент 2, в то время как элемент 1 служит для получения инверсии сигнала x_1 , а элемент 3 преобразует полученное на элементе 2 инверсное значение функции в прямое. Многие элементы хранения, например, триггерные схемы, позволяют получать сигнал в парафазном коде, то есть имеют два выхода, на одном из которых сигнал имеет прямое, а на другом — инверсное значение. Это позволяет избавиться от элемента 1 в схеме. Если предположить, что значения выходных сигналов имеют инверсный вид по отношению к представленному в табл. 1.1.1, то отпадает необходимость в элементе 3. В большинстве реальных интегральных микросхем реализованы именно *дешифраторы* с инверсными выходами. Обозначение такого *дешифратора* показано на рис. 1.1.3.

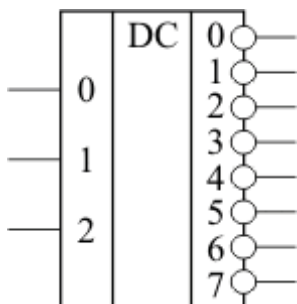


Рис. 1.1.3. Условно-графическое обозначение дешифратора с инверсными выходами

На выходах такого *дешифратора* образуется унитарный код, содержащий один и только один ноль. Например, если входные сигналы имеют значение $110_2 = 6_{10}$, то выходы *дешифратора*, представленного на рис. 1.1.3, будут находиться в состоянии 10111111, то есть выход 6 будет иметь значение, отличное от остальных выходов.

Дешифраторы широко применяются в различных устройствах компьютеров. Прежде всего, они используются для выбора ячейки запоминающего устройства, к которой производится обращение для записи или считывания информации. При этом часть разрядов адресного кода может дешифроваться *дешифраторами*, выполненными в виде отдельных интегральных схем, а другая часть разрядов (обычно младшая) дешифруется с помощью *дешифраторов*, встроенных непосредственно в БИС запоминающего устройства. Кроме того, *дешифраторы* находят применение в устройстве управления для определения выполняемой операции, построения распределителей импульсов и в других блоках.

Шифратор

Шифратор — схема, имеющая 2^n входов и n выходов, функции которой во многом противоположны функции *дешифратора* (рис. 1.1.4). Эта комбинационная схема в соответствии с унитарным кодом на своих входах формирует позиционный код на выходе (таблица 1.1.2).

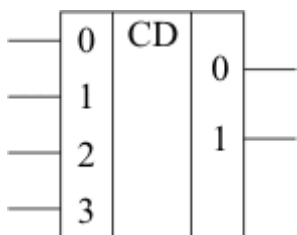


Рис.1. 1.4. Условно-графическое обозначение шифратора на 4 входа

Входы		Выходы	
3	2	1	0
0	0	1	0
0	0	x	1
0	1	x	1
1	x	x	1

Триггер

Триггер — электронная схема, обладающая двумя устойчивыми состояниями. Переход из одного устойчивого состояния в другое происходит скачкообразно под воздействием управляющих сигналов. При этом также скачкообразно изменяется уровень напряжения на выходе *триггера*.

Триггеры служат основой для построения регистров, *счетчиков* и других элементов, обладающих функцией хранения.

Главной частью любого *триггера* является запоминающая ячейка (ЗЯ). Схема запоминающей ячейки на элементах "И-НЕ" представлена на рис. 1.1.5.

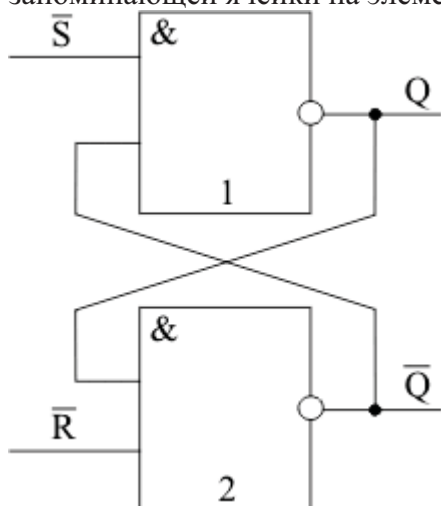


Рис. 1.1.5. Запоминающая ячейка на элементах "И-НЕ"

Входной сигнал S (Set) служит для установки ЗЯ в состояние "1" ($Q=1$, $\bar{Q}=0$). Сигнал R (Reset) устанавливает ЗЯ в состояние "0" ($Q=0$, $\bar{Q}=1$). Активными значениями для них являются сигналы низкого уровня.

Пусть на входы ЗЯ поданы сигналы: $\bar{S}=0$, $\bar{R}=1$. Тогда при любом исходном состоянии ЗЯ на выходе элемента 1 установится высокий уровень напряжения. Так как на входы элемента 2 поступают значения Q и \bar{R} , то на его выходе будет сигнал низкого уровня. Таким образом, ЗЯ перейдет в состояние "1".

Аналогично при $\bar{S}=1$, $\bar{R}=0$ запоминающая ячейка перейдет в состояние $Q=0$, $\bar{Q}=1$, то есть в "0".

Если $\bar{S}=1$, $\bar{R}=1$, то состояние ЗЯ будет определяться ее предыдущим состоянием. Если ЗЯ находилась в состоянии "1", то сигнал $\bar{Q}=0$, поступая на вход элемента 1, подтвердит состояние его выхода $Q=1$. На входы элемента 2 поступят сигналы только высокого уровня. Поэтому его выход будет находиться в состоянии $\bar{Q}=0$, то есть — не изменится. Если ЗЯ находилась в состоянии "0", то сигнал $Q=0$, поступая на вход элемента 2,

подтвердит состояние его выхода $Q=1$. В свою очередь, выход элемента 1 также останется без изменения. Таким образом, эта комбинация входных сигналов соответствует режиму хранения.

Если на входы S и R поданы сигналы низкого уровня ($S = R = 0$), то сигнал на выходах элементов 1 и 2 будет высоким ($Q = \bar{Q} = 1$). При переводе ЗЯ в режим хранения ($S = R = 1$), выходы элементов 1 и 2 могут установиться в произвольное состояние. Поэтому комбинация сигналов $S = R = 0$ на управляющих входах не используется.

Работа триггерной схемы определяется не таблицей истинности, как для комбинационной логической схемы, а таблицей переходов. Таблица переходов показывает изменение состояния *триггера* при изменении состояния входных сигналов в зависимости от его текущего состояния. Таблица переходов запоминающей ячейки, показанной на рис. 1.1.5, представлена в табл. 1.1.3.

Таблица 1.1.3.

S	R	$Q(t+1)$	Функция
0	0	x	Запрещенная комбинация
0	1	1	Установка в "1"
1	0	0	Установка в "0"
1	1	$Q(t)$	Хранение

Аналогичная запоминающая ячейка может быть построена на элементах "ИЛИ-НЕ".

Такие запоминающие ячейки можно рассматривать как простейшие асинхронные *триггеры*, на базе которых строятся синхронные триггерные схемы.

Триггеры можно классифицировать по различным признакам, например так, как это показано на рис. 1.1.6.



Рис. 1.1.6. Классификация триггерных схем

Триггер называется синхронным, если его таблица переходов хотя бы по одному управляющему входу реализуется под воздействием синхронизирующего сигнала.

Рассмотрим организацию синхронного одноступенчатого *триггера* (рис. 1.1.7).

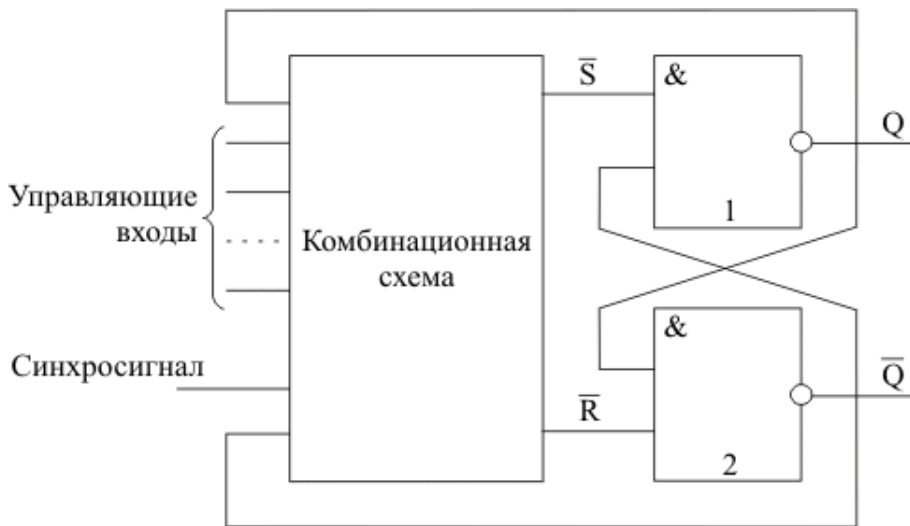


Рис. 1.1.7. Обобщенная схема синхронного одноступенчатого триггера

Основу синхронного одноступенчатого *триггера* составляет рассмотренная выше запоминающая ячейка (элементы 1, 2).

Комбинационная схема преобразует управляющие сигналы *триггера*, а также, для некоторых типов *триггеров*, сигналы Q и \bar{Q} с выходов ЗЯ в сигналы \bar{S} и \bar{R} на входах запоминающей ячейки. Синхросигнал C разрешает передачу на входы элементов 1 и 2 таких значений сигналов \bar{S} и \bar{R} , которые устанавливают ЗЯ в то или иное состояние. Неактивное значение синхросигнала обеспечивает на входах запоминающей ячейки состояние управляющих сигналов $\bar{S} = \bar{R} = 1$, что соответствует для нее режиму хранения. Схема синхронного одноступенчатого *RS-триггера* приведена на рис. 1.1.8. Его таблица переходов представлена в табл. 1.1.4.

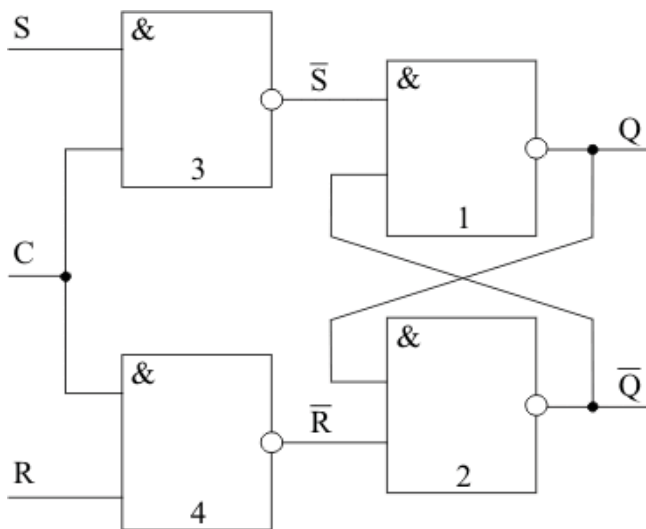


Рис. 1.1.8. Синхронный одноступенчатый RS-триггер

Таблица 1.1.4.

R	S	Q(t+1)	Функция
0	0	Q(t)	Хранение
0	1	1	Установка в "1"
1	0	0	Установка в "0"
1	1	x	Запрещенная комбинация

Еще раз подчеркнем, что данная таблица переходов будет реализовываться лишь при активном уровне синхросигнала C (для данной организации это $C = 1$). При $C = 0$ выходы элементов 3 и 4 (см. рис. 1.1.8) будут в состоянии "1", что соответствует режиму хранения запоминающей ячейки, реализованной на элементах 1 и 2.

Таблицы переходов JK — и D — *триггеров* приведены в таблицах 1.1.5 и 1.1.6 соответственно.

Таблица 1.1.5.

J	K	Q(t+1)	Функция
0	0	Q(t)	Хранение
0	1	0	Установка в "0"
1	0	1	Установка в "1"
1	1	Q(t)	Инвертирование предыдущего состояния

Таблица 1.1.6.

D	Q(t+1)	Функция
0	0	Установка в "0"
1	1	Установка в "1"

Представленный на рис. 1.1.8 *триггер* имеет статическую синхронизацию, при которой управляющие сигналы активизируют

входы S и R запоминающей ячейки во время высокого уровня сигнала на входе синхронизации. Его условно-графическое обозначение приведено на рис. 1.1.9,а. Условно-графические обозначения *триггеров*, использующих другие типы синхронизации, приведены на рис. 1.1.9, б, в, г (на примере RS -триггера).

На рис. 1.1.9,б представлено УГО *триггера* со статической синхронизацией в случае, если активный уровень синхросигнала низкий. Условно-графические обозначения *триггеров* с динамической синхронизацией показаны на рис. 1.1.9, в и 1.1.9, г. В первом случае изменение состояния *триггера* под воздействием поступивших управляющих сигналов происходит только в момент переключения синхронизирующего сигнала с низкого уровня на высокий, а во втором — при переключении с высокого на низкий уровень. При постоянном значении уровня синхросигнала состояние выхода *триггера* с динамической синхронизацией не меняется при любых изменениях управляющих сигналов на его входах.

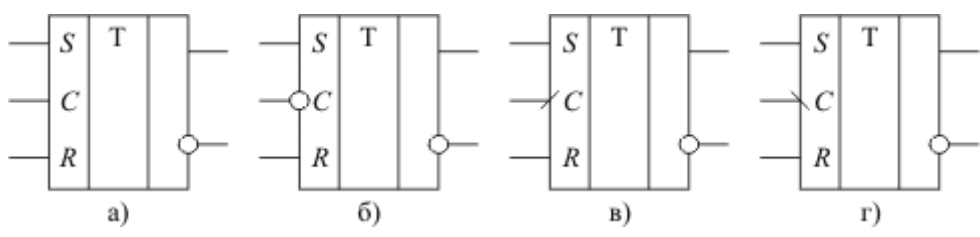


Рис. 1.1.9. Условно-графические обозначения RS -триггера с различной синхронизацией: а — статическая синхронизация; б — статическая инверсная синхронизация; в — динамическая синхронизация передним фронтом синхросигнала; г — динамическая синхронизация задним фронтом синхросигнала

Идеализированная (без учета задержек) временная диаграмма работы RS -триггеров с различными типами синхронизации приведена на рис. 1.1.10.

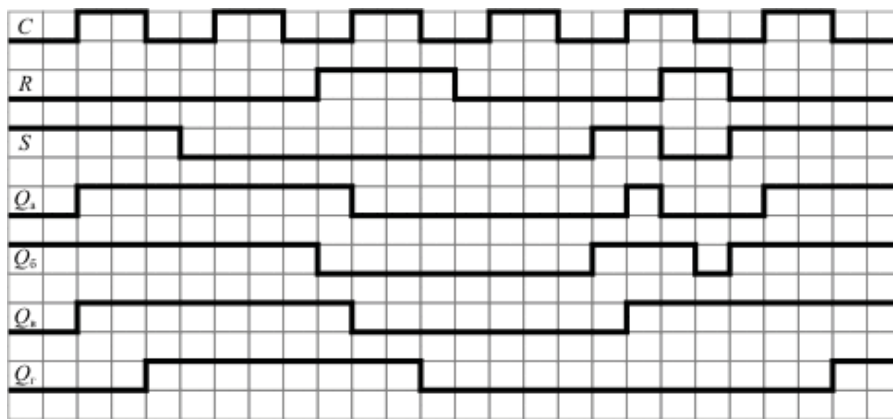


Рис. 1.1.10. Временная диаграмма работы RS-триггера с различными типами синхронизации: Q_a — статическая синхронизация; Q_b — статическая инверсная синхронизация; Q_c — динамическая синхронизация передним фронтом синхросигнала; Q_d — динамическая синхронизация задним фронтом синхросигнала

Как отмечалось выше, синхронный *триггер*, помимо управляющих входов, воздействующих на его состояние при подаче сигнала синхронизации, может иметь входы, которые воздействуют на его состояние непосредственно. Обычно они используются для установки *триггера* в то или иное начальное состояние перед подачей последовательности синхросигналов. Схема синхронного *RS-триггера* с асинхронными входами установки в "0" и в "1" приведена на рис. 1.1.11, а его условно-графическое обозначение — на рис. 1.1.12.

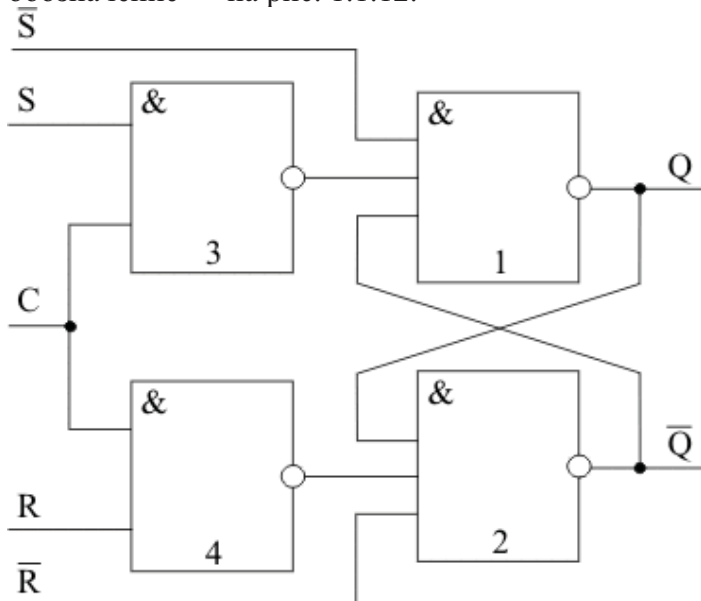


Рис. 1.1.11. Синхронный одноступенчатый RS-триггер с асинхронными установочными входами

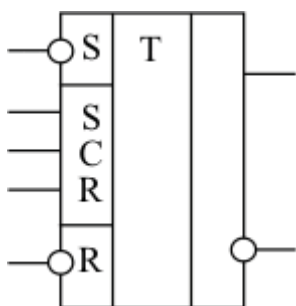


Рис. 1.1.12. Условно-графическое обозначение синхронного одноступенчатого RS-триггера с асинхронными установочными входами

Сигналы, поступающие по асинхронным входам S и R , подаются непосредственно на входы запоминающей ячейки, образованной элементами 1 и 2, минуя цепь, управляемую синхросигналом (элементы 1 и 2), и вызывают переключение запоминающей ячейки согласно табл. 1.1.3.

Триггеры некоторых типов используют значения выходного сигнала для формирования управляющих сигналов на входах запоминающей ячейки (см. рис. 1.1.7). Это может привести к непредсказуемой последовательности его переключений. При построении отдельных схем на основе триггеров, например, регистров сдвига, необходимо, чтобы значения выходных сигналов триггера не изменялись на то время, пока производится их запись и значения его выходных сигналов в другой триггер, но сам этот триггер должен воспринимать значения с выхода некоторой другой триггерной схемы. Эти, а также некоторые другие ситуации требуют особых подходов к организации триггера, основным из которых является создание двухступенчатых триггеров.

Двухступенчатый RS-триггер (рис. 1.1.13 и рис. 1.1.14) строится на основе двух одноступенчатых триггеров с прямой статической синхронизацией. Информация в первую ступень триггера (элемент 1) заносится во время действия высокого уровня синхросигнала. После того как синхросигнал на входе принимает низкое значение, элемент 1 переходит в режим хранения, а значение высокого сигнала на выходе инвертора 3 обеспечивает запись состояния триггера 1 в триггер 2. Идеализированная временная диаграмма работы двухступенчатого RS-триггера приведена на рис. 1.1.15.

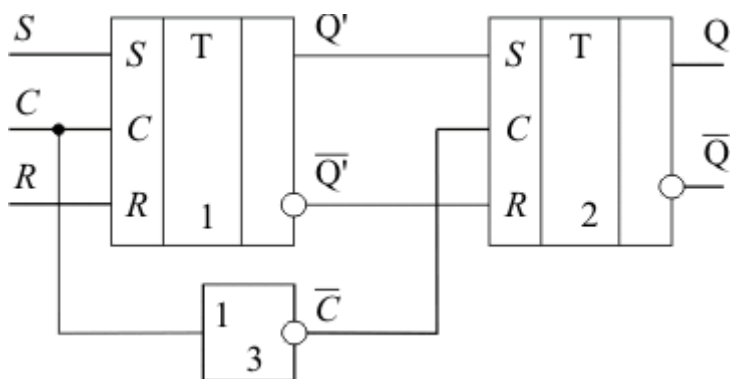


Рис. 1.1.13. Схема двухступенчатого RS-триггера

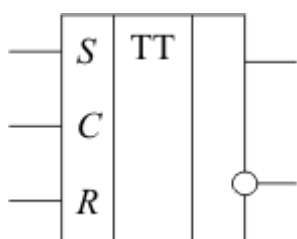


Рис. 1.1.14. Условно-графическое обозначение двухступенчатого RS-триггера

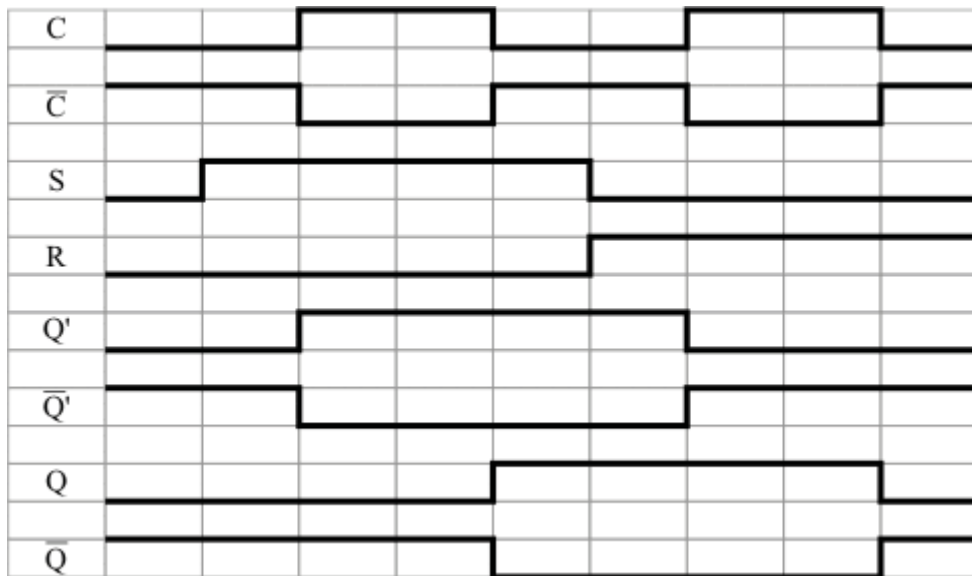


Рис. 1.1.15. Временная диаграмма работы двухступенчатого RS-триггера

Следует отметить, что первая ступень представляет собой одноступенчатый *триггер*, реализующий заданную таблицу переходов, в то время как вторая ступень — это всегда одноступенчатый синхронный *RS-триггер*. Например, на рис. 1.1.16 показана схема двухступенчатого *JK-триггера*.

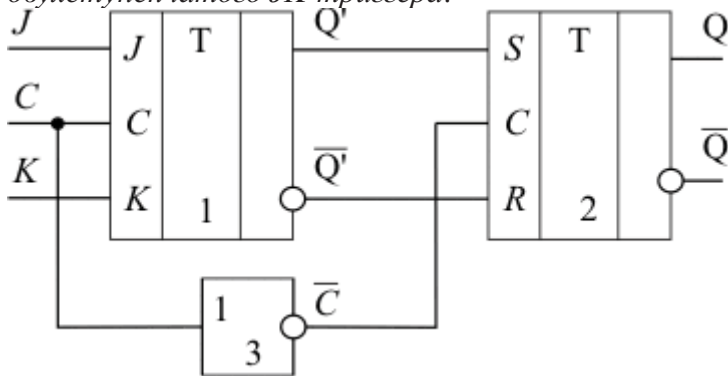


Рис. 1.1.16. Двухступенчатый JK-триггер

Тема 2. Основные функциональные элементы ЭВМ, часть 2

Счетчики

Счетчиком называется электронная схема, предназначенная для подсчета числа сигналов, поступающих на его счетный вход. *Счетчики* используются в устройстве управления компьютера при построении распределителей импульсов и организации циклов, в *счетчиках* команд для формирования адреса выполняемой команды при естественном порядке выполнения и в некоторых других устройствах ЭВМ. Также *счетчики* широко применяются в качестве самостоятельных узлов в различных системах цифровой автоматики.

Суть работы *счетчика* заключается в изменении на единицу зафиксированного в нем значения с приходом каждого счетного сигнала. *Счетчики* принято подразделять на суммирующие, вычитающие и реверсивные. Суммирующие *счетчики* увеличивают свое значение, вычитающие — уменьшают, а реверсивные могут работать как на прибавление, так и на вычитание в зависимости от сигналов управления. Параметром, определяющим информационную емкость *счетчика*, является **модуль пересчета**, равный числу внутренних состояний. Это значение проставляется на УГО после аббревиатуры СТ.

Пример асинхронного трехразрядного двоичного суммирующего *счетчика* приведен на рис. 1.2.1, а его условно-графическое обозначение — на рис. 1.2.2. Для построения этого *счетчика* использованы *JK-триггеры* с динамической синхронизацией по спаду синхросигнала. Каждый *JK-триггер* в *счетчике* включен в режим инвертирования своего состояния при переключении синхросигнала с высокого уровня на низкий (см. табл. 1.1.5). Идеализированная временная диаграмма работы этого *счетчика* показана на рис. 1.2.3.

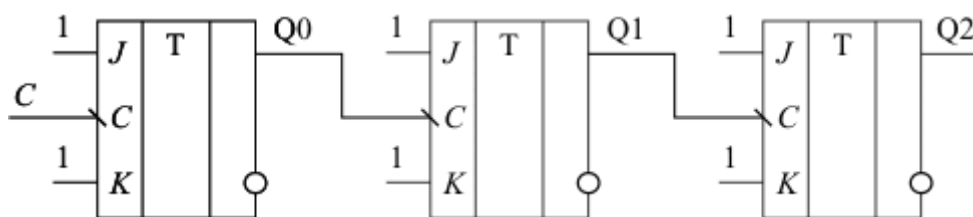


Рис. 1.2.1. Схема асинхронного трехразрядного счетчика

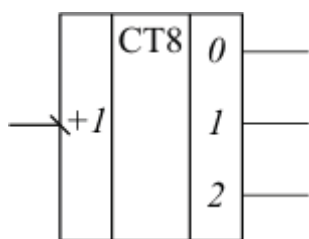


Рис. 1.2.2. Условно-графическое обозначение трехразрядного суммирующего счетчика

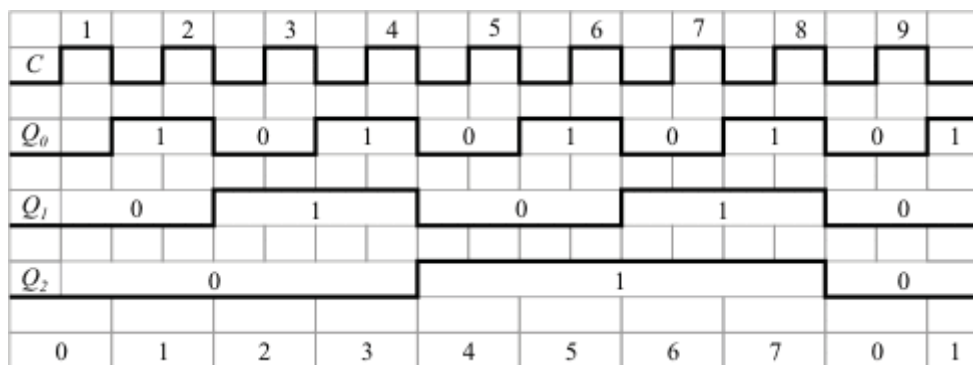


Рис. 1.2.3. Временная диаграмма работы счетчика

Быстродействие асинхронного *счетчика* определяется максимальной задержкой от изменения сигнала на его счетном входе до полного установления состояния всех его выходов. Проведем оценку быстродействия на примере переключения выходов *счетчика* после поступления восьмого синхросигнала на его вход (рис. 1.2.4), так как именно в этом такте время переключения выходов *счетчика* будет максимальным.

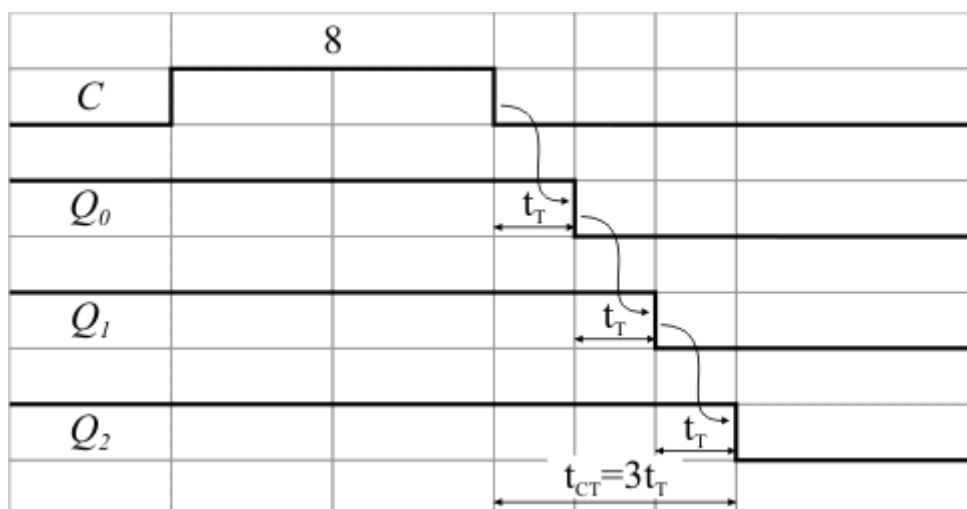


Рис. 2.4. Оценка быстродействия асинхронного счетчика

По фронту 1-0 сигнала *C*(8) с задержкой сигнала, равной t_T , на *триггере* *Q₀* (см. рис. 1.2.1) происходит изменение сигнала на выходе *Q₀*. Это изменение, в свою очередь, приведет к переключению сигнала *Q₁* с соответствующей задержкой относительно переключения *Q₀*. Вслед за этим с задержкой сигнала на следующем *триггере* переключится сигнал *Q₂*. То есть общее время задержки переключения сигналов на выходе трехразрядного *счетчика* составит $3t_T$. Очевидно, что для *n* разрядного *счетчика* **время задержки** составит $t_{СТ} = n \cdot t_T$

Таким образом, с увеличением разрядности асинхронного *счетчика* увеличивается его задержка и, следовательно, уменьшается быстродействие.

Этот недостаток устраняется в *счетчиках*, работающих по синхронной схеме. В них за счет дополнительных управляющих комбинационных схем обеспечивается одновременное переключение всех разрядов при поступлении сигнала на счетный вход (с задержкой, равной задержке одного *триггера* вне зависимости от разрядности *счетчика*). Обычно *счетчик* имеет вход установки в нулевое состояние (асинхронный сброс составляющих его *триггеров* в "0"). Некоторые *счетчики* имеют цепи установки в произвольное начальное состояние, с которого уже будет начинаться операция счета.

Регистры

Регистр — внутреннее запоминающее устройство процессора или внешнего устройства, предназначенное для временного хранения обрабатываемой или управляющей информации. *Регистры* представляют собой совокупность *триггеров*, количество которых равняется разрядности *регистра*, и вспомогательных схем, обеспечивающих выполнение некоторых элементарных операций. Набор этих операций, в зависимости от функционального назначения *регистра*, может включать в себя одновременную установку всех разрядов *регистра* в "0", параллельную или последовательную загрузку *регистра*, сдвиг содержимого *регистра* влево или вправо на требуемое число разрядов, управляемую выдачу информации из *регистра* (обычно используется при работе нескольких схем на общую шину данных) и т.д.

Регистры хранения используются для приема, хранения и выдачи многоразрядного кода. Они представляют собой совокупность одноступенчатых *триггеров* (как правило, *D*-типа) с общим входом синхронизации. Иногда в *регистре* имеется также и общий вход

асинхронной установки всех *триггеров* в "0". Схема четырехразрядного *регистра хранения* приведена на рис. 1.2.5, а его условно-графическое обозначение — на рис. 1.2.6.

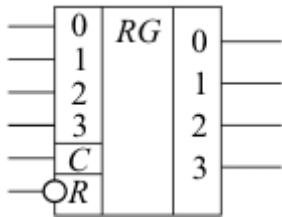


Рис. 1.2.6. Условно-графическое обозначение четырехразрядного регистра хранения с асинхронным входом установки в "0"

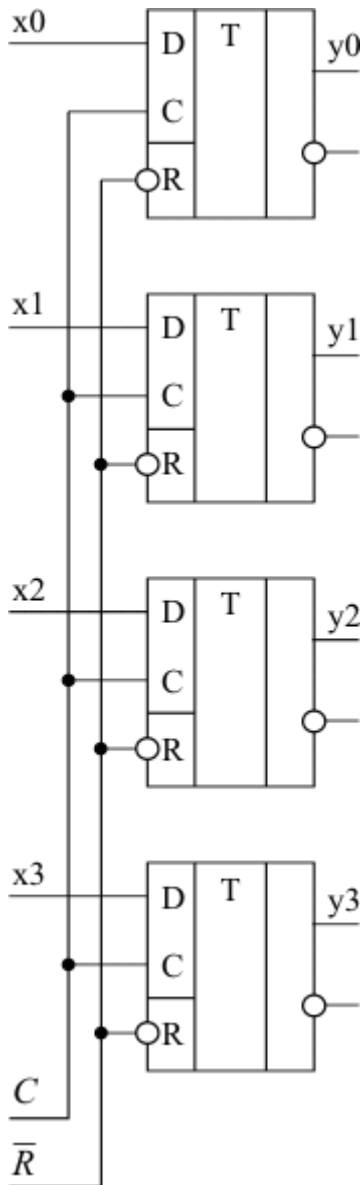


Рис. 1.2.5. Структура четырехразрядного регистра хранения с асинхронным входом установки в "0"

Регистр сдвига — *регистр*, обеспечивающий помимо хранения информации, сдвиг влево или вправо всех разрядов одновременно на одинаковое число позиций. При этом выдвигаемые за пределы *регистра* разряды теряются, а в освобождающиеся разряды заносится информация, поступающая по отдельному внешнему входу *регистра сдвига*. Обычно эти *регистры* обеспечивают сдвиг кода на одну позицию влево или вправо. Но

существуют и универсальные *регистры сдвига*, которые выполняют сдвиг как влево, так и вправо в зависимости от значения сигнала на специальном управляющем входе или при подаче синхросигналов на разные входы *регистра*. *Регистр сдвига* может быть спроектирован и таким образом, чтобы выполнять сдвиг одновременно не на одну, а на несколько позиций.

Регистры сдвига строятся на *двухступенчатых триггерах*. Схема четырехразрядного *регистра*, выполняющего сдвиг на один разряд от разряда 0 к разряду 3, показана на рис. 1.2.7, а его условно-графическое обозначение — на рис. 1.2.8. Ввод информации в данный *регистр* — последовательный через внешний вход D0. *Регистр* имеет вход асинхронной установки всех разрядов в "0". Для наглядности каждый двухступенчатый *регистр* представлен двумя одноступенчатыми с соответствующей организацией синхронизации первой и второй ступеней. Пунктиром обозначен реальный *двухступенчатый триггер*.

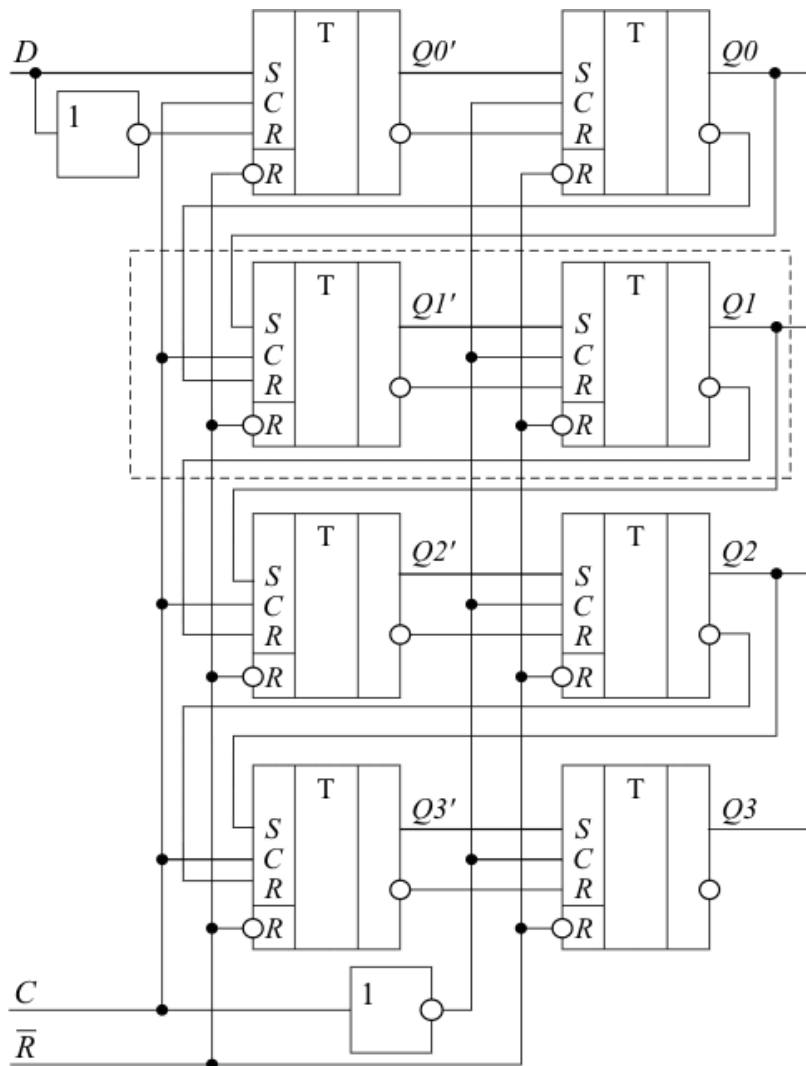


Рис. 1.2.7. Структура регистра сдвига

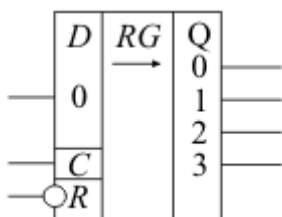


Рис. 1.2.8. Условно-графическое обозначение четырехразрядного регистра сдвига с асинхронным входом установки в "0"

Идеализированная временная диаграмма работы *регистра сдвига*, структура которого представлена на рис. 1.2.7, показана на рис. 1.2.9. Предполагаем, что начальное состояние *регистра* следующее: $Q_0=0, Q_1=1, Q_2=1, Q_3=0$.

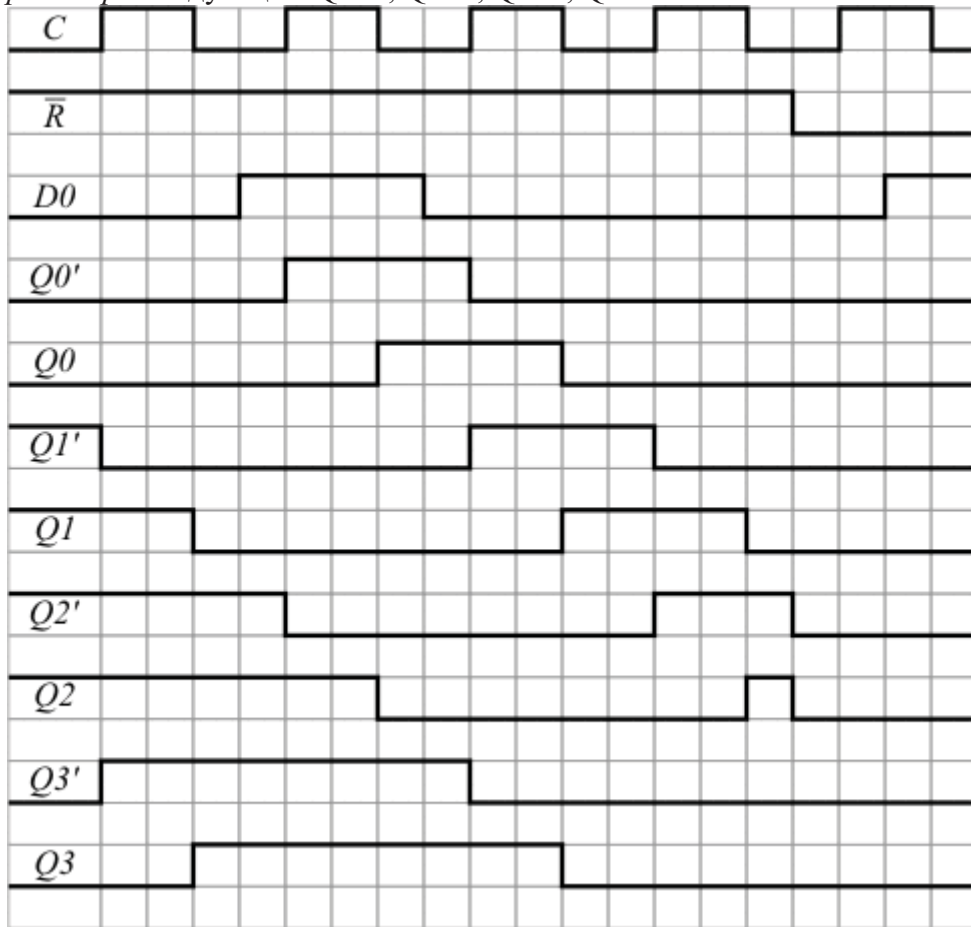


Рис. 1.2.9. Временная диаграмма работы регистра сдвига

Работа *регистра сдвига* в каждом периоде сигнала синхронизации разбивается на две фазы: при высоком и при низком значении синхросигнала:

1. При высоком уровне синхросигнала проводится запись значения выхода ($i - 1$)-го разряда *регистра* в первую ступень i -го разряда. Вторая ступень каждого разряда сохраняет свое прежнее значение. В этой фазе состояние первой ступени i -го *триггера* повторяет состояние второй ступени ($i - 1$)-го *триггера*. Вторые ступени каждого *триггера*, а следовательно, и выходы *регистра* в целом, остаются неизменными.
2. При низком уровне синхросигнала значение, записанное в первой ступени каждого *триггера*, перезаписывается в его вторую ступень. Запись в первую ступень *триггера* запрещена. В этой фазе состояния первой и второй ступеней каждого *триггера* становятся одинаковыми.

Поступление сигнала $\overline{R} = 0$ вне зависимости от значения сигнала на входе синхронизации C и сигнала на входе D_0 устанавливает все разряды *регистра* в нулевое состояние.

Тема 3. Арифметико-логическое устройство

Классическая ЭВМ состоит из трех основных устройств: *арифметико-логического устройства*, устройства управления и запоминающего устройства. Рассмотрим особенности организации этих устройств. Прежде всего, рассмотрим структуру *арифметико-логического устройства*.

В современных ЭВМ *арифметико-логическое устройство* не является самостоятельным схмотехническим блоком. Оно входит в состав микропроцессора, на котором строится компьютер. Однако знание структуры и принципов работы *АЛУ* весьма важно для понимания работы компьютера в целом. Для лучшего понимания этих вопросов проведем синтез арифметического устройства, предназначенного для выполнения только одной операции — *умножения* чисел с фиксированной запятой, заданных в прямом коде, со старших разрядов множителя. В ходе этого процесса также обратим внимание на особенности использования рассмотренных выше основных схмотехнических элементов ЭВМ.

Синтез *АЛУ* проходит в несколько этапов. Сначала необходимо выбрать метод, по которому предполагается выполнение операции, и составить алгоритм соответствующих действий. Исходя из алгоритма и формата исходных данных, следует определить набор составляющих *АЛУ* элементов. Затем требуется определить связи между элементами, установить порядок функционирования устройства и временную диаграмму управляющих сигналов, которые должны быть поданы на *АЛУ* от устройства управления.

Пусть операнды имеют вид:

$$[X]_{\text{пк}} = x_0x_1x_2\dots x_n, \text{ и } [Y]_{\text{пк}} = y_0y_1y_2\dots y_n$$

где x_0, y_0 — знаковые разряды.

Операция *умножения* чисел с фиксированной запятой, заданных в прямом коде, со старших разрядов множителя выполняется по следующей формуле:

$$\text{Sign } Z = \text{Sign } X \oplus \text{Sign } Y$$

$$|Z| = y_1 \cdot |X| \cdot 2^{-1} + y_2 \cdot |X| \cdot 2^{-2} + \dots + y_n \cdot |X| \cdot 2^{-n}$$

$$[X]_{\text{пк}} = 0.1101 ; \text{Sign } X = 0$$

$$[Y]_{\text{пк}} = 1.1011 ; \text{Sign } Y = 1$$

$$\text{Sign } Z = 0 \oplus 1 = 1$$

$$|X| = 0.1101$$

$$|Y| = 0.1011$$

$$y_1y_2y_3y_4$$

$$+0.00000000 \quad |Z| = 0$$

$$y_1 = 1 \quad 0.01101000 \quad 1 \cdot |X| \cdot 2^{-1}$$

$$+ \boxed{0.01101000} \quad |Z| = |Z| + |X| \cdot 2^{-1}$$

$$y_2 = 0 \quad 0.00000000 \quad 0 \cdot |X| \cdot 2^{-2}$$

$$+ \boxed{0.01101000} \quad |Z| = |Z| + 0$$

$$y_3 = 1 \quad 0.00011010 \quad 1 \cdot |X| \cdot 2^{-3}$$

$$+ \boxed{0.10000010} \quad |Z| = |Z| + |X| \cdot 2^{-3}$$

$$y_4 = 1 \quad 0.00001101 \quad 1 \cdot |X| \cdot 2^{-4}$$

$$\boxed{0.10001111} \quad |Z| = |Z| + |X| \cdot 2^{-4}$$

Алгоритм вычислений представлен на рис. 1.3.1

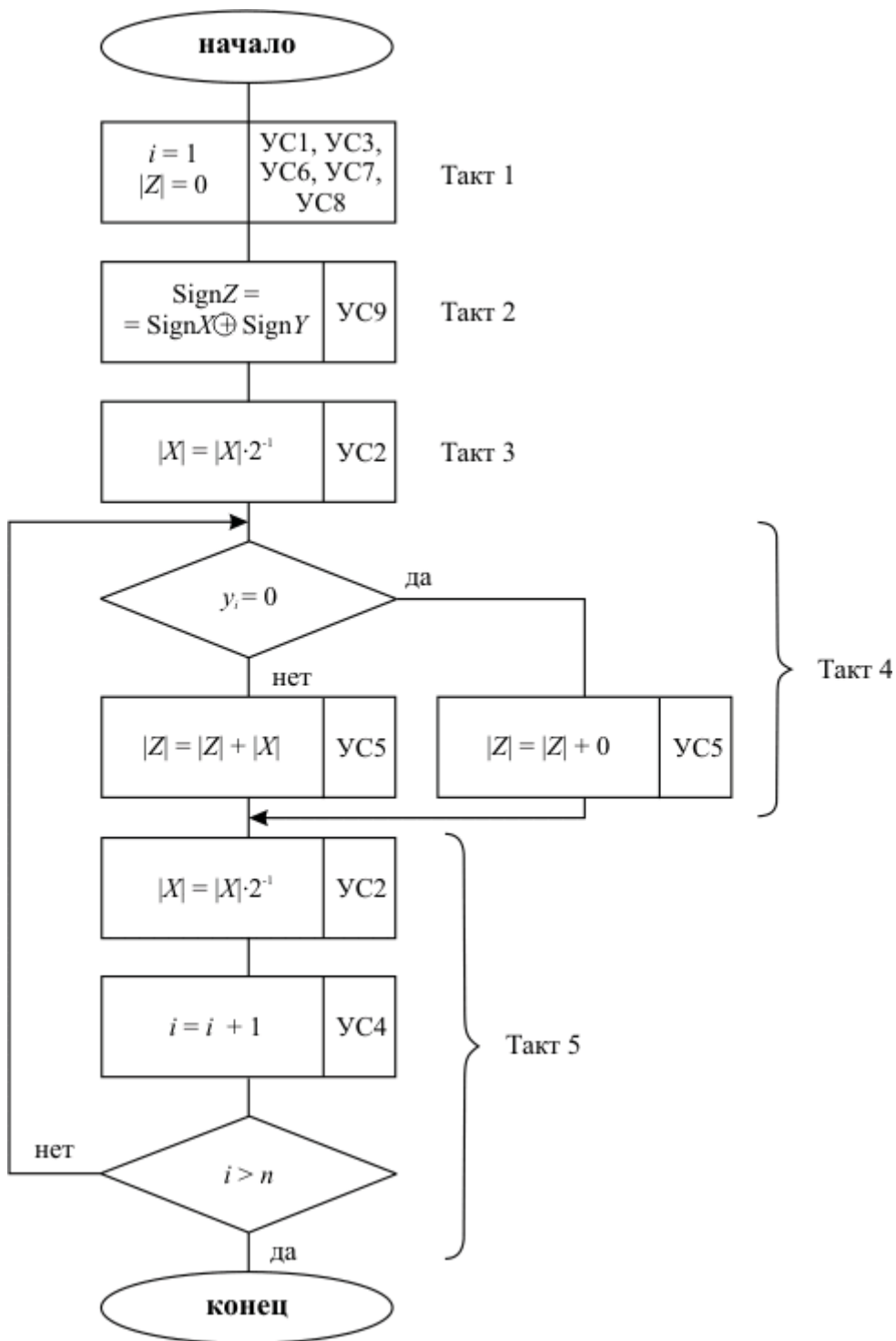


Рис. 1.3.1. Алгоритм операции умножения чисел с фиксированной запятой, заданных в прямом коде, со старших разрядов множителя

Каждой переменной, представленной в алгоритме, в схеме должен соответствовать элемент хранения. Разрядность модуля произведения равна сумме разрядностей сомножителей. Умножение двоичного числа на 2^{-i} обеспечивается сдвигом этого числа вправо на соответствующее количество разрядов. Переход к анализу очередного разряда множителя ($i = i + 1$) может быть обеспечен сдвигом регистра множителя на один разряд в сторону старших разрядов.

Исходя из этого, определим состав оборудования, необходимого для реализации АЛУ заданного типа для $n = 4$ (таблица 1.3.1).

Схема	Разрядность	Функции	Управляющий сигнал
Регистр модуля множимого RGX	8	Загрузка. Сдвиг в сторону младших разрядов.	УС1 УС2
Регистр модуля множителя RGY	4	Загрузка. Сдвиг в сторону старших разрядов.	УС3 УС4
Регистр модуля результата RGZ	8	Загрузка. Установка в "0".	УС5 УС6
Триггер знака множимого TX		Загрузка	УС7
Триггер знака множителя TY		Загрузка	УС8
Триггер знака результата TZ		Загрузка	УС9
АЛУ	8	Комбинационный сумматор	–
Комбинационные схемы		Получение на входе АЛУ сигналов "0" или RGX в зависимости от значения y_i	–

Структурная схема устройства представлена на рис. 1.3.2.

Временная диаграмма управляющих сигналов, поступающих на *арифметико-логическое устройство*, показана на рис. 1.3.3.

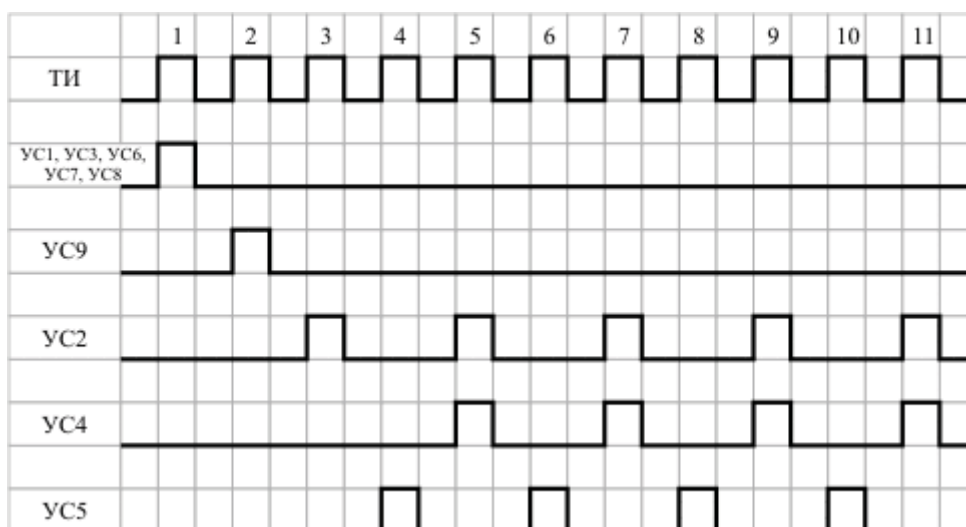


Рис. 1.3.3. Временная диаграмма управляющих сигналов

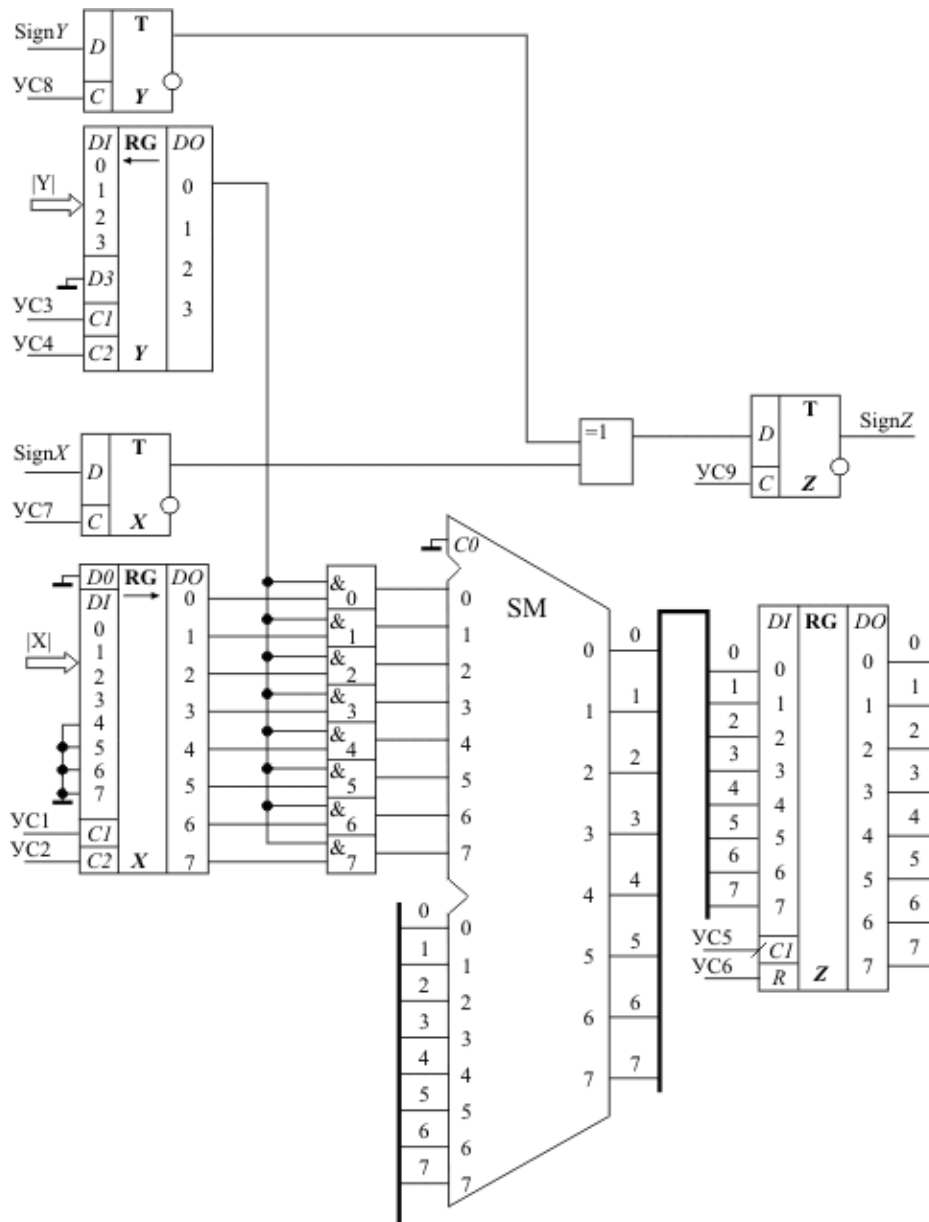


Рис. 1.3.2. Структурная схема арифметического устройства для выполнения операции умножения со старших разрядов множителя чисел, заданных в прямом коде

Работа схемы

Такт 1. Загрузка модулей операндов в регистры RGX, RGY, а их знаков — в триггеры TX и TY. Сброс в "0" регистра результата RGZ.

Такт 2. Запись знака результата в триггер TZ.

Такт 3. Сдвиг регистра RGX на один разряд вправо. Через время, равное задержке на переключение регистров и комбинационных схем, на выходе комбинационного сумматора и, следовательно, на входе регистра RGZ устанавливается результат $0+y_1 \cdot |X| \cdot 2^{-1}$.

Такт 4. Загрузка RGZ: $|Z|=|Z|+y_1 \cdot |X| \cdot 2^{-1}$.

Такт 5. Сдвиг RGX на 1 разряд вправо: $|X|=|X| \cdot 2^{-1}$.

Сдвиг RGY на 1 разряд влево: $i=i+1$.

Устройство управления проверяет условие окончания операции: $i > n$.

Такты (6,7), (8,9), (10,11) ... Повтор действий тактов (4,5) с анализом других значений y_i . В такте 10 в регистре RGZ формируется модуль произведения. Такт 11 используется лишь для определения условия окончания операции умножения.