

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Национальный исследовательский ядерный университет «МИФИ»

## **СХЕМОТЕХНИКА ЭВМ**

Сборник задач

Рекомендовано УМО «Ядерные физика и технологии» в качестве  
учебного пособия для студентов высших учебных заведений

Москва 2012

УДК 004.3  
ББК 32.973я7  
С92

**Схемотехника ЭВМ. Сборник задач:** учебное пособие. М.: НИЯУ МИФИ, 2012. – 240 с.

*Авторы:* Н.А. Дмитриев, М.Н. Ёхин, М.А. Иванов, Б.Н. Ковригин, Г.Г. Новиков, В.Г. Тышкевич, И.М. Ядыкин.

В пособии содержится описание типовых элементов и функциональных узлов цифровых схем, примеры их использования. Приведены схемотехнические решения для комбинационных и последовательностных схем, применяемые при разработке цифровых устройств. В пособии приведено большое количество задач по разработке функциональных цифровых устройств. Данное пособие предназначено для приобретения и закрепления практических навыков по курсу «Схемотехника ЭВМ» при разработке цифровых устройств.

Предназначено для студентов, специализирующихся в области информатики и вычислительной техники.

Пособие подготовлено в рамках  
Программы создания и развития НИЯУ МИФИ

Рецензент д-р техн. наук Кривин В.В.

ISBN 978-5-7262-1776-5

© Национальный исследовательский ядерный университет «МИФИ», 2012

# СОДЕРЖАНИЕ

|                                                         |    |
|---------------------------------------------------------|----|
| Введение .....                                          | 6  |
| 1. Типовые элементы и узлы цифровых схем.               |    |
| Примеры их использования .....                          | 7  |
| 1.1. Комбинационная логика .....                        | 7  |
| 1.1.1. Логические элементы .....                        | 7  |
| Простейшие логические элементы .....                    | 7  |
| Логические элементы «Исключающее ИЛИ» .....             | 10 |
| Контроль по чётности .....                              | 11 |
| Пороговые схемы .....                                   | 12 |
| Мажоритарные элементы .....                             | 13 |
| Сумматоры .....                                         | 15 |
| 1.1.2. Дешифраторы .....                                | 16 |
| Увеличение разрядности дешифраторов .....               | 18 |
| Реализация функций на дешифраторах .....                | 22 |
| 1.1.3. Мультиплексоры .....                             | 23 |
| Увеличение разрядности мультиплексоров .....            | 26 |
| Реализация функций на мультиплексорах .....             | 30 |
| Сдвоенный десятивходовой мультиплексор .....            | 32 |
| Детектор $M$ из $N$ .....                               | 33 |
| Схема формирования сигнала при делении числа на 7 ..... | 36 |
| Схема быстрого сдвига на $K$ разрядов .....             | 37 |
| Демультимплексоры и шины .....                          | 41 |
| 1.1.4. Компараторы .....                                | 42 |
| Схемы равенства .....                                   | 43 |
| Схемы сравнения .....                                   | 45 |
| Увеличение разрядности компараторов .....               | 46 |
| Компаратор на основе сумматора .....                    | 49 |
| 1.1.5. Преобразователи кодов .....                      | 50 |
| Двоичные шифраторы .....                                | 51 |
| Приоритетные шифраторы .....                            | 52 |
| Увеличение разрядности приоритетного шифратора .....    | 54 |
| Указатели старшей единицы .....                         | 55 |

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| 1.2. Последовательностные элементы и узлы .....                              | 57  |
| 1.2.1. Триггеры .....                                                        | 57  |
| Классификация триггерных схем .....                                          | 57  |
| Таблицы переходов и обозначения триггеров .....                              | 59  |
| Динамические параметры триггеров .....                                       | 63  |
| Проектирование триггера с заданной таблицей перехода ....                    | 63  |
| Реализация триггера с мультиплексором на входе.....                          | 67  |
| Делители частоты на триггерах.....                                           | 67  |
| Триггеры на мультиплексорах.....                                             | 69  |
| 1.2.2. Счётчики .....                                                        | 70  |
| Классификация счётчиков .....                                                | 71  |
| Счётчик с произвольным кодированием .....                                    | 72  |
| Синхронный четырехразрядный двоичный счётчик.....                            | 74  |
| Увеличение разрядности счётчиков.....                                        | 75  |
| Счётчик с заданным модулем.....                                              | 77  |
| Счётчик с переменным модулем.....                                            | 79  |
| Делители частоты .....                                                       | 81  |
| Формирователи импульсов .....                                                | 84  |
| Преобразователи код–частота .....                                            | 86  |
| 1.2.3. Регистры .....                                                        | 87  |
| Классификация регистров.....                                                 | 88  |
| Операции сдвигов.....                                                        | 89  |
| Универсальные регистры .....                                                 | 91  |
| Увеличение разрядности регистров .....                                       | 93  |
| Счётчик на регистре сдвига .....                                             | 94  |
| Кольцевые счётчики. Распределители тактов.....                               | 96  |
| Счётчик Джонсона.....                                                        | 99  |
| Генераторы псевдослучайных последовательностей.....                          | 102 |
| Регистр последовательного приближения.....                                   | 104 |
| Счётчики на регистрах сдвига с линейными<br>обратными связями.....           | 107 |
| 2. Задачи на проектирование цифровых схем.....                               | 111 |
| 2.1. Проектирование и анализ комбинационных схем.....                        | 111 |
| 2.1.1. Синтез комбинационных схем .....                                      | 111 |
| 2.1.2. Синтез комбинационных схем при словесном<br>описании их функций ..... | 113 |

|                                                                                                              |            |
|--------------------------------------------------------------------------------------------------------------|------------|
| 2.1.3. Построение временной диаграммы работы комбинационной схемы с учётом задержек элементов .....          | 120        |
| 2.1.4. Синтез комбинационных схем на основе дешифраторов.....                                                | 128        |
| 2.1.5. Синтез комбинационных схем на основе мультиплексоров.....                                             | 132        |
| 2.1.6. Синтез комбинационных схем на основе мультиплексоров при словесном задании функции ...                | 137        |
| 2.1.7. Синтез преобразователей кодов.....                                                                    | 139        |
| 2.1.8. Синтез шифраторов и дешифраторов.....                                                                 | 140        |
| 2.2. Проектирование и анализ последовательностных схем ...                                                   | 141        |
| 2.2.1. Определение типа триггера по временной диаграмме .....                                                | 141        |
| 2.2.2. Синтез триггера по заданной таблице переходов .....                                                   | 145        |
| 2.2.3. Проектирование схем на основе триггеров по графу переходов.....                                       | 147        |
| 2.2.4. Построение временных диаграмм схем на основе триггеров.....                                           | 151        |
| 2.2.5. Синтез схем на основе регистров .....                                                                 | 159        |
| 2.2.6. Построение синхронных счётчиков на регистрах сдвига с линейными и нелинейными обратными связями ..... | 163        |
| 2.2.7. Проектирование счётчиков с заданным модулем счета.....                                                | 163        |
| 2.2.8. Проектирование пересчётных схем.....                                                                  | 165        |
| 2.2.9. Построение временных диаграмм работы последовательностных схем на основе счётчиков и регистров.....   | 167        |
| 2.2.10. Синтез генераторов последовательностей сигналов по заданной временной диаграмме .....                | 171        |
| 2.2.11. Синтез генераторов последовательностей сигналов .....                                                | 179        |
| 2.2.12. Синтез устройств анализа цифровых последовательностей .....                                          | 187        |
| <i>Приложение. Справочные данные по элементам .....</i>                                                      | <i>201</i> |
| <i>Список литературы .....</i>                                                                               | <i>238</i> |

## Введение

Схемотехническая элементная база является основой построения разнообразных средств цифровой техники. Знание элементной базы и схемотехнических решений являются основой в овладении современными средствами цифровой вычислительной техники и их грамотном применении, а также при усовершенствовании, повышении быстродействия и разработке новых средств.

В пособии приведены схемотехнические основы и рассмотрен широкий круг вопросов, связанных с изучением, проектированием, анализом и применением современной элементной базы цифровой схемотехники ЭВМ.

В первой части пособия рассмотрены основные схемотехнические элементы цифровых устройств, описаны законы их функционирования, классификация, внутренняя структура и характеристики. Рассмотрены принципы построения и инженерные методы проектирования типовых функциональных узлов. Приведены примеры применения комбинационных и последовательностных элементов и узлов цифровых устройств широкого применения.

Во второй части приведены задачи на проектирование цифровых схем. Задачи ориентированы на овладение принципами проектирования и методами анализа цифровых схем, изучение возможностей элементов и приобретение опыта исследования и тестирования схем.

Набор задач позволяет совместить изучение методов проектирования с освоением практических схем, закрепить навыки анализа и синтеза как комбинационных, так и последовательностных схем на примерах разнообразных цифровых схем.

Проектирование и анализ схем проводятся с применением систем автоматизации проектирования, ориентированных на ПЛИС фирмы Xilinx.

Содержание пособия предназначено для проведения практических занятий и основано на материале курса лекций, читаемого на кафедре «Компьютерные системы и технологии» НИЯУ МИФИ.

В сборнике использованы задачи, приведенные в литературных источниках, а также задачи, предлагавшиеся на олимпиадах МИФИ и на московских городских олимпиадах по электронике и вычислительной технике в 1976 – 1985 гг.

# 1. ТИПОВЫЕ ЭЛЕМЕНТЫ И УЗЛЫ ЦИФРОВЫХ СХЕМ. ПРИМЕРЫ ИХ ИСПОЛЬЗОВАНИЯ

## 1.1. КОМБИНАЦИОННАЯ ЛОГИКА

### 1.1.1. Логические элементы

#### *Простейшие логические элементы*

**Логическая схема**, выходные сигналы которой зависят только от значений входных сигналов в данный момент времени, называются комбинационной схемой. Функции, выполняемые такой схемой, могут быть заданы системой логических выражений или описаны таблицей истинности, в которой перечислены все комбинации входных сигналов и соответствующие им значения сигналов на выходах. В комбинационных схемах отсутствуют цепи (петли) обратной связи.

Простейшие логические элементы реализуют стандартные логические функции от одной или нескольких переменных. К одновходовым логическим элементам относятся буферный элемент BUF или повторитель и инвертор INV, который реализует функцию отрицания НЕ. Буфер – это схема, которая просто преобразует «слабые» логические сигналы в «сильные» сигналы.

Многовходовые логические схемы реализуют функции:

- AND (логическое умножение, И, конъюнкция,  $\wedge$ ) – схема вырабатывает на выходе логический ноль, если хотя бы одним входе присутствует ноль;
- OR (логическое сложение, ИЛИ, дизъюнкция,  $\vee$ ) – схема вырабатывает на выходе логическую единицу, если хотя бы на одном входе присутствует единица.

Элементный базис, в котором можно реализовать логическую функцию от любого количества переменных, называется функционально полным. Набор трёх логических функций: НЕ, И, ИЛИ называется булевым базисом и является функционально полным.

Широко применяются одноэлементные функционально полные базисы:

- NAND (И-НЕ, штрих Шеффера,  $\downarrow$ ) – схема вырабатывает на выходе логическую единицу, если хотя бы одним входе присутствует ноль;

- NOR (ИЛИ-НЕ, стрелка Пирса,  $\uparrow$ ) – схема вырабатывает на выходе логический ноль, если хотя бы одном входе присутствует единица.

**Примечание:** Ниже будут употребляться следующие символы логических операций: символ « $\times$ » – операция И, символ « $+$ » – операция ИЛИ, символ « $\neg$ » – инверсия, символ « $\oplus$ » – сложение по модулю 2.

На основании закона двойственности (теорема де Моргана) для любого логического элемента можно использовать одно из двух условных графических обозначений основанных на операциях конъюнкции или дизъюнкции с применением инверсии на входах и/или выходах. На рис. 1 приведены эквивалентные условные обозначения элементов для простейших логических функций.

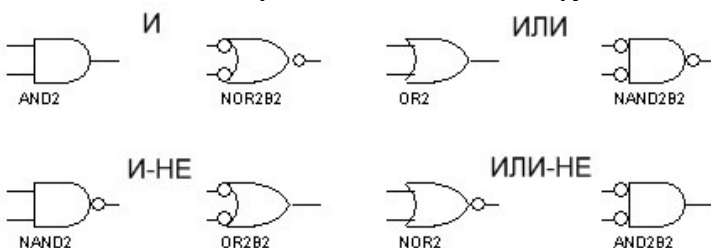


Рис. 1. Эквивалентные условные обозначения элементов

Правильный выбор названий сигналов и условных обозначений элементов может существенно облегчить понимание и использование логических схем.

**Построение многовходовых схем.** Для элементов И и ИЛИ это не представляет трудностей: для получения нужного числа входов берется несколько элементов, выходы которых объединяются далее элементом того же типа. Нарращивание числа входов для элементов И-НЕ и ИЛИ-НЕ, проводится аналогично, но в схемах могут появляться дополнительные инверторы (рис. 2).

В другом варианте построения многовходовых схем функция разбивается с помощью закона двойственности (теоремы де Моргана) на группы элементов, из заданного элементного базиса, размер которых соответствует числу входов имеющихся элементов (рис. 3).

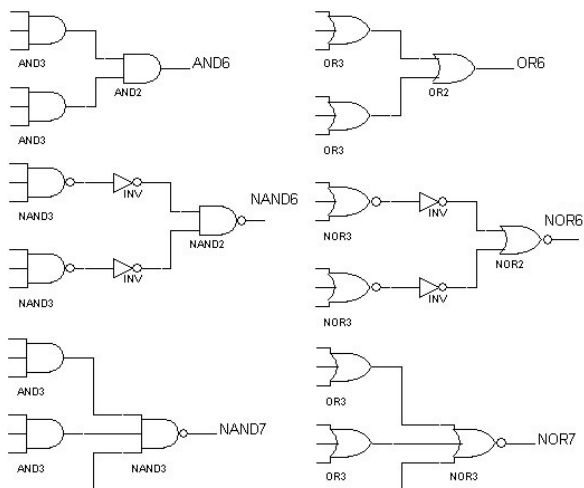


Рис. 2. Многовходовые схемы

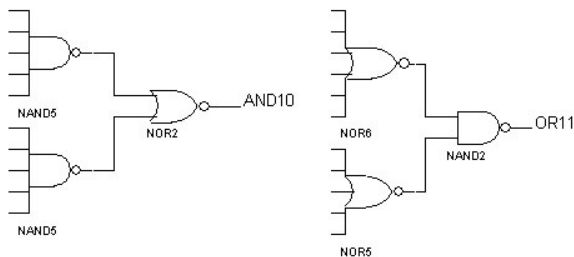


Рис. 3. Многовходовые схемы 10И и 11ИЛИ

**Неиспользуемые входы.** Иногда используются не все входы логического элемента. Если в схеме с  $N+1$  входом объединить два входа, то схема будет работать в качестве схемы с  $N$  входами. Можно также подать на неиспользуемые входы постоянные логические значения: на неиспользуемый вход схем И и И-НЕ следует подать логическую единицу, а на неиспользуемые входы схем ИЛИ и ИЛИ-НЕ необходимо подать логический ноль.

*Следует помнить, что в схемах неиспользуемые входы не должны быть разомкнутыми, так как на них могут наводиться паразитные потенциалы, которые могут изменить работу, а также изменить быстродействие схемы.*

## Логические элементы «Исключающее ИЛИ»

Логические элементы «Исключающее ИЛИ» реализуют логическую операцию неравнозначность (сумма по модулю два, Exclusive-OR, XOR) и обозначаются символами  $M2$ ,  $\oplus$ ,  $=1$ . Двухвходовой элемент сложения по модулю два вырабатывает на выходе единичный сигнал лишь в том случае, когда единица присутствует только на одном его входе или другой вариант определения – когда сигналы на его входах различны. Элемент «Исключающее ИЛИ» можно рассматривать как управляемый инвертор – если один из входов использовать как управляющий и подавать на него значение «0», то информация, поступающая по второму входу, будет передаваться на выход без изменения, а при подаче значения «1» – информация по второму входу будет инвертироваться.

Операция сумма по модулю два определяется соотношением

$$A \oplus B = \neg A \cdot B + A \cdot \neg B.$$

На рис. 4 приведены два варианта реализации двухвходовой схемы «Исключающее ИЛИ» и её условное графическое обозначение (УГО).

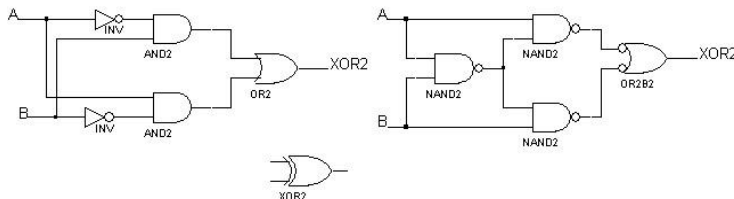


Рис. 4. Двухвходовые схемы «Исключающее ИЛИ» и УГО

Инверсия суммы по модулю два – элемент «Исключающее ИЛИ-НЕ» реализует операцию равнозначность (эквивалентность, Exclusive-NOR, XNOR) и имеет значение прямо противоположное – он равен единице, если сигналы на входах элемента одинаковы.

**Построение многовходовых сумматоров по модулю два.** Если последовательно включить  $N$  двухвходовых схем «Исключающее ИЛИ», как показано на рис. 5,а, то получится схема с  $N+1$  входом и одним выходом. Сигнал на её выходе будет равен единице, если на нечётное число входов поданы единицы. На рис. 5,б приведена древовидная (пирамидальная) структура многовходовой схемы,

которая является более быstroдействующей, так как в этой схеме одновременно обрабатываются все входы, т.е. выполнение операции распараллелено.

Для многовходовых функций «Исключающее ИЛИ-НЕ», схемы строятся аналогично. Сначала суммируются по модулю два все разряды, а затем их значение инвертируется.

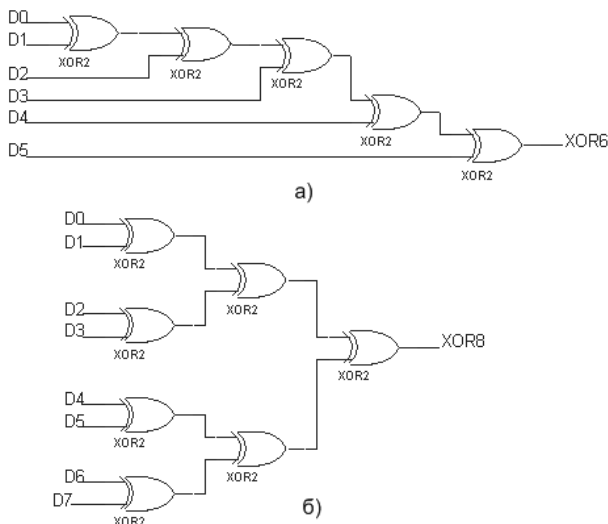


Рис. 5. Каскадное включение схем «Исключающее ИЛИ»

### ***Контроль по чётности***

Схемы контроля решают задачу выявления ошибок при работе устройств работе устройств функционирования процессора, а также применяются при диагностировании устройств. Следует отметить, что добавление функций контроля всегда связано с избыточностью – дополнительные аппаратные и временные затраты.

Контроль по чётности (контроль по модулю два) широко применяется для контроля правильности передач и хранения данных. При этом способе контроля кодовое слово данных дополняется контрольным разрядом (битом чётности), значение которого устанавливается так, чтобы сумма по модулю два всех разрядов, включая контрольный разряд, была чётной (нечётной). При одиночной

или нечётной ошибке изменится значение контрольного разряда, что обнаруживается схемой сравнения. При двойной ошибке или чётном количестве ошибок значение контрольного разряда не изменяется. Таким образом, контроль по чётности (по модулю два) обнаруживает не только одиночные ошибки, но и нечётной кратности.

Контроль по модулю два реализуется с помощью схем свертки. Пример схемы свертки пирамидального типа на элементах «Исключающее ИЛИ» приведён на рис. 5,б. Схема формирует контрольный разряд чётности для байта.

### ***Пороговые схемы***

Пороговой схемой  $k$  из  $N$  называется комбинационная схема, имеющая  $N$  входов и один выход и формирующая на выходе единичное значение, если на  $k$  и более входах установлена единица, и значение логического нуля, если количество единиц меньше  $k$  или если количество нулей больше  $k$ . При значении  $k=1$  пороговая схема вырождается в  $N$ -входовой логический элемент ИЛИ, а при значении  $k=N$  – в  $N$ -входовой элемент И.

Пороговые элементы применяются в анализаторах сигналов, работающих по принципу  $k$  из  $N$ , в устройствах шифрования при решении задачи «распределения» секрета и др.

На рис. 6 приведена таблица истинности для пятивходового порогового элемента при  $k=3$ .

Функционирование порогового элемента 3 из 5 описывается выражением:

$$F_{\geq 3} = X_5 \cdot X_4 \cdot X_3 + X_5 \cdot X_4 \cdot X_2 + X_5 \cdot X_4 \cdot X_1 + X_5 \cdot X_3 \cdot X_2 + X_5 \cdot X_3 \cdot X_1 + \\ + X_5 \cdot X_2 \cdot X_1 + X_4 \cdot X_3 \cdot X_2 + X_4 \cdot X_3 \cdot X_1 + X_4 \cdot X_2 \cdot X_1 + X_3 \cdot X_2 \cdot X_1.$$

При больших значениях  $k$  и  $N$  для реализации пороговых элементов в базисах И-НЕ или ИЛИ-НЕ требуется большое количество элементов для проверки большого числа сочетаний входных сигналов. Наиболее просто пороговые элементы могут быть реализованы с помощью одноразрядных двоичных сумматоров, так как все входные переменные имеют одинаковый вес, равный единице.

| $i$ | $x_5x_4x_3x_2x_1$ | $a_i$ | $i$ | $x_5x_4x_3x_2x_1$ | $a_i$ |
|-----|-------------------|-------|-----|-------------------|-------|
| 0   | 0 0 0 0 0         | 0     | 16  | 1 0 0 0 0         | 0     |
| 1   | 0 0 0 0 1         | 0     | 17  | 1 0 0 0 1         | 0     |
| 2   | 0 0 0 1 0         | 0     | 18  | 1 0 0 1 0         | 0     |
| 3   | 0 0 0 1 1         | 0     | 19  | 1 0 0 1 1         | 1     |
| 4   | 0 0 1 0 0         | 0     | 20  | 1 0 1 0 0         | 0     |
| 5   | 0 0 1 0 1         | 0     | 21  | 1 0 1 0 1         | 1     |
| 6   | 0 0 1 1 0         | 0     | 22  | 1 0 1 1 0         | 1     |
| 7   | 0 0 1 1 1         | 1     | 23  | 1 0 1 1 1         | 1     |
| 8   | 0 1 0 0 0         | 0     | 24  | 1 1 0 0 0         | 0     |
| 9   | 0 1 0 0 1         | 0     | 25  | 1 1 0 0 1         | 1     |
| 10  | 0 1 0 1 0         | 0     | 26  | 1 1 0 1 0         | 1     |
| 11  | 0 1 0 1 1         | 1     | 27  | 1 1 0 1 1         | 1     |
| 12  | 0 1 1 0 0         | 0     | 28  | 1 1 1 0 0         | 1     |
| 13  | 0 1 1 0 1         | 1     | 29  | 1 1 1 0 1         | 1     |
| 14  | 0 1 1 1 0         | 1     | 30  | 1 1 1 1 0         | 1     |
| 15  | 0 1 1 1 1         | 1     | 31  | 1 1 1 1 1         | 1     |

Рис. 6. Таблица истинности порогового элемента «3 из 5»

### **Мажоритарные элементы**

Мажоритарный элемент проводит «голосование» и передает на выход значение, соответствующее большинству входных сигналов. Мажоритарный элемент имеет нечётное число входов. Мажоритарный элемент, это пороговая схема  $k$  из  $N$ , которая формирует на выходе единичное значение, если на ее входах не менее  $k=(N+1)/2$  единичных сигналов.

Практическое применение в системах контроля нашли мажоритарные элементы, имеющие 3, 5 и 7 входов. На рис. 7 приведены таблица истинности и схема мажоритарного элемента 2 из 3. Кроме выхода  $F$ , в таблице даны функции двухразрядного кода  $A1, A0$ , указывающие номер отказавшего канала. Нулевому коду соответствует правильное функционирование всех каналов. Выход мажоритарного элемента совпадает со значениями входов, если они все одинаковые, и с двумя одинаковыми, если третий отличается от двух других.

Из таблицы истинности и после минимизации, можно легко получить функции выходов:

$$F = D3 \cdot D2 + D3 \cdot D1 + D2 \cdot D1, \quad A1 = D3 \oplus D2, \quad A0 = D3 \oplus D1.$$

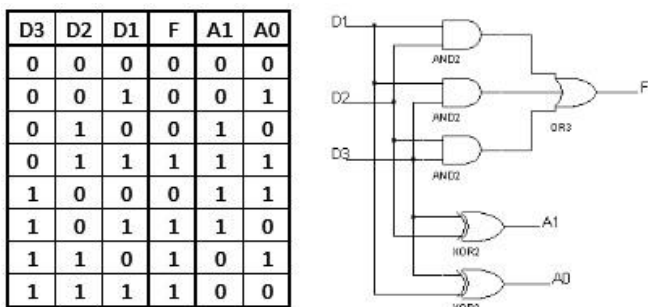


Рис. 7. Таблица истинности и мажоритарный элемент на три входа

Из анализа функции трёхразрядного мажоритарного элемента видно, что она совпадает с функцией сигнала переноса одноразрядного полного сумматора.

На рис. 8 приведена схема пятивходового мажоритарного элемента реализованного на трёхвходовых мажоритарных элементах.

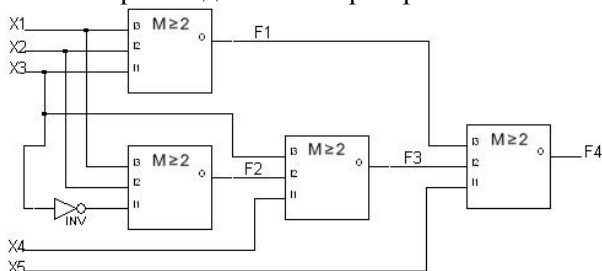


Рис. 8. Схема пятивходового мажоритарного элемента

Функции сигналов на выходах элементов могут быть представлены в следующем виде:

$$\begin{aligned}
 F1 &= X3 \cdot X2 + X3 \cdot X1 + X2 \cdot X1, & F2 &= \neg X3 \cdot X2 + \neg X3 \cdot X1 + X2 \cdot X1, \\
 F3 &= X4 \cdot X3 + X4 \cdot F2 + X3 \cdot F2 = X4 \cdot X3 + X4 \cdot X2 + X4 \cdot X1 + X3 \cdot X2 \cdot X1, \\
 F4 &= X5 \cdot F3 + X5 \cdot F1 + F3 \cdot F1 = X5 \cdot X4 \cdot X3 + X5 \cdot X4 \cdot X2 + X5 \cdot X4 \cdot X1 + \\
 &+ X5 \cdot X3 \cdot X2 + X5 \cdot X3 \cdot X1 + X5 \cdot X2 \cdot X1 + X4 \cdot X3 \cdot X2 + X4 \cdot X3 \cdot X1 + X4 \cdot X2 \cdot X1 + X3 \cdot X2 \cdot X1
 \end{aligned}$$

В схемах контроля от мажоритарного элемента требуется особенно высокая надёжность. В реальных системах мажоритарного контроля обычно применяют три узла мажоритарного контроля. Своими входами они подключены к трём входным каналам параллельно, а их выходы образуют три выходных канала. Такое включение защищает систему от одиночных ошибок не только во входном канале, но и в самом узле мажорирования.

## Сумматоры

Сумматоры выполняют операцию арифметического сложения и вычитания двоичных чисел, которые являются самыми распространенными арифметическими действиями. Особенностью операции арифметического сложения для многоразрядных схем, в отличие от логического сложения и сложения по модулю два, является организация цепей межразрядного переноса в сторону старших разрядов. Правила арифметического сложения для числовых разрядов справедливы как для чисел без знака, так и для чисел, представленных в машинных кодах (прямом, дополнительном, обратном).

**Полусумматор** (half adder) складывает два одноразрядных операнда  $A$  и  $B$  и формирует 2-разрядную сумму. Младший бит суммы называется полусуммой  $HS$ , а старший бит – переносом  $CO$  (carry output) в старший разряд, который означает переполнение разрядной сетки (рис. 9,а).

**Одноразрядный полный сумматор** (full adder) помимо входов слагаемых есть имеет вход бита входного переноса  $CI$ . В этом случае сигнал на выходе суммы  $S=1$ , если на нечётном числе входов присутствуют единицы, а сигнал на выходе переноса  $CO=1$ , если единицы имеются на двух и более входов (рис. 9,б).

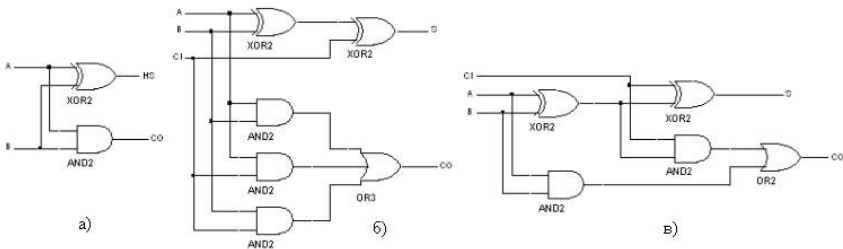


Рис. 9. Схемы полусумматора и одноразрядных сумматоров

Полный сумматор можно реализовать на основе полусумматоров и схемы ИЛИ (рис. 9,в).

Аппаратная сложность и быстродействие сумматоров являются важными параметрами, поэтому разработано множество вариантов сумматоров, которые имеют разветвленную классификацию.

### 1.1.2. Дешифраторы

Дешифратором (или декодером (decoder)) называют кодирующее устройство, преобразующее двоичный код в унитарный код логических единиц. Унитарным кодом называется двоичный код, содержащий только одну единицу, например, 01000, или его называют кодом «1 из  $N$ ».

Дешифратор – это комбинационная схема, имеющая  $K$  адресных (селективных) входов и  $N$  выходов. Двоичный код, поданный на адресные входы  $K$ , определяет номер активного выхода, на котором устанавливается единица. На всех остальных выходах устанавливаются нули. Кроме того, для расширения функциональных возможностей дешифраторы могут иметь входы разрешения  $E$ . Данные входы используются для наращивания дешифраторов, синхронизации их работы с работой других узлов. Если между числом выходов  $N$  и числом адресных входов  $K$  действует соотношение  $N=2^K$ , то такой дешифратор называют полным. Если  $N < 2^K$ , то дешифратор называют неполным.

Дешифраторы широко применяются в различных устройствах компьютеров, когда необходимо выбрать блок или разрешить работать одному из блоков устройств. Прежде всего, они применяются в структурах выборки адреса в системах памяти. Кроме того, они применяются для дешифрации выполняемой операции, маршрутизации данных, построения распределителей импульсов, а также для формирования элементарных конъюнктивных форм в логических схемах управления и функциях.

Условное графическое обозначение дешифратора и таблица истинности для двухвходового дешифратора D2\_4E приведены на рис. 10 (при  $K=2$  и  $N=4$ ).



Рис. 10. УГО и таблица истинности дешифратора D2\_4E

В обозначении дешифратора D2\_4E первая цифра указывает число входов, а вторая выходов и читается как дешифратор «два в четыре», или для дешифратора D3\_8E – «три в восемь».

Когда разрешающий вход  $E=1$ , то дешифратор D2\_4E устанавливает единицу только на одном выходе в зависимости от состояния управляющих (адресных) входов A1,A0. При  $E=0$  состояние всех выходов равно «0».

Из таблицы истинности дешифратора D2\_4E видно, что сигнал на каждом выходе формируется только при одном входном наборе и может быть описан следующими функциями:

$$\begin{aligned} D0 &= E \cdot \neg A1 \cdot \neg A0; & D1 &= E \cdot \neg A1 \cdot A0; \\ D2 &= E \cdot A1 \cdot \neg A0; & D3 &= E \cdot A1 \cdot A0. \end{aligned}$$

Реализация дешифратора D2\_4E приведена на рис. 11. Данный дешифратор содержит четыре трёхвходовых элемента И. Структура данного дешифратора является наиболее простой и называется **линейной**.

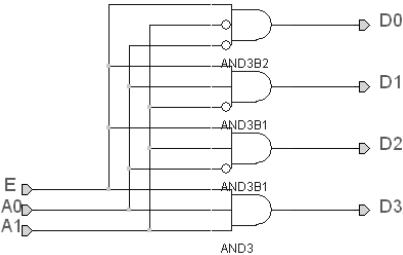


Рис. 11. Логическая схема дешифратора D2\_4E

На рис. 12 приведены условное графическое обозначение и таблица истинности для двухвходового дешифратора D2\_4E с инверсными выходами (в библиотеке элементов X74\_139).



Рис. 12. УГО и таблица истинности дешифратора D2\_4E с инверсными выходами

На выходах данного дешифратора образуется унитарный код, содержащий только один ноль, а на остальных входах сигнал единицы, например, 110111. При  $E=1$  состояние всех выходов равно «1».

### ***Увеличение разрядности дешифраторов***

Разрядность дешифраторов можно увеличивать построением следующих структур:

- линейной;
- пирамидальной;
- матричной.
- каскадной.

В линейной структуре объем аппаратуры составляют  $N$   $(K+1)$ -входовых элементов И и  $K$  инверторов входных переменных. При этом с ростом выходов  $N$  растёт количество входов  $(K+1)$  элементов И. Кроме того, при большой разрядности дешифраторов также необходимо учитывать коэффициенты разветвления логических элементов и коэффициенты нагрузки (количество входов, подключаемых к линиям управляющих сигналов). Данная структура обладает наивысшим быстродействием и задержка равна сумме задержек инвертора и элемента И.

В пирамидальной структуре каждая ступень дешифратора строится только на двухвходовых элементах 2И, количество которых увеличивается вдвое на каждой ступени. Пример пирамидальной структуры дешифратора D3\_8 приведён на рис. 13. В данной структуре задержка в каждой ступени равна двум элементам.

В матричной структуре дешифратора выходные сигналы формируются на выходах матрицы из двухвходовых элементов 2И, на входы которых поступают функции от двух вспомогательных дешифраторов. В матрице выбирается выход, находящийся на пересечении выбранных строки и столбца, которые поступают с выходов вспомогательных дешифраторов. Пример матричной структуры дешифратора D4\_16 приведён на рис. 14. Задержка выходного сигнала в данной структуре составит три элемента, при этом уменьшается нагрузка на входы. В целом данная структура является оптимальной.

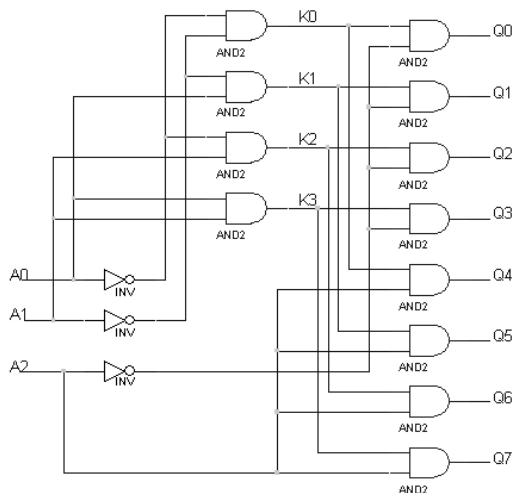


Рис. 13. Логическая схема дешифратора D3\_8 пирамидальной структуры

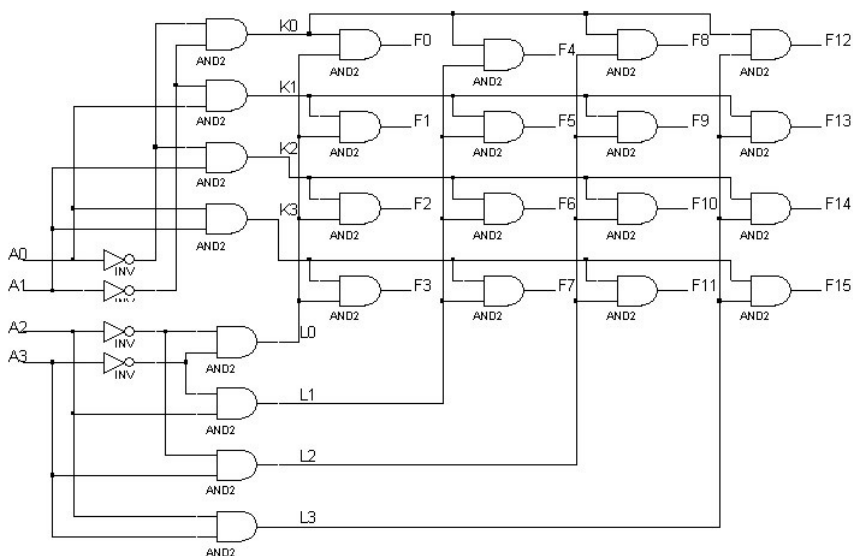


Рис. 14. Логическая схема дешифратора D4\_16 матричной структуры

Сигнал разрешения  $E$  в пирамидальной структуре можно подать только на первую ступень, а в матричной структуре дешифраторов достаточно подать только на один из входных вспомогательных

дешифраторов. При этом запираются все выходы первой ступени, или все строки, или все столбцы.

Для уменьшения аппаратных затрат применяется способ управления по одной из входных переменных. Для этого одну из входных адресных переменных в прямом и инверсном виде передают через два элемента 2И и заводят на них сигналы  $E$  (рис. 15). При  $E=0$  конъюнкторы будут заперты и следовательно не будет выбран ни один выход дешифратора. В этом способе задержка дешифратора увеличивается на задержку элемента 2И.

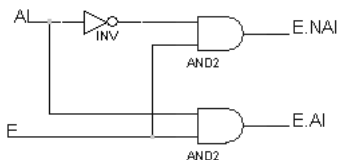


Рис. 15. Схема управления по одной переменной

В каскадной структуре для построения дешифраторов большой разрядности используются функционально законченные дешифраторы меньшей разрядности и с входом разрешения  $E$ . Количество дешифраторов в каждом каскаде равно количеству выходов предыдущего каскада. В каждом каскаде активным будет только один выход, который разрешает работу одному дешифратору следующего каскада. В каждом каскаде одноимённые адресные входы объединены между собой. Старшие разряды адреса поступают на первый каскад, а младшие разряды на последний выходной каскад. На рис. 16 приведена двухкаскадная структура для дешифратора D4\_16, реализованный на пяти дешифраторах D2\_4E.

Часто дешифраторы имеют несколько разрешающих входов, часть из которых может быть инверсными, а разрешающей комбинацией является их конъюнкция. Данные входы кроме функции разрешения могут использоваться для дешифрации адреса.

В выходном каскаде дешифратора возможно использование матричной структуры на элементах 2И. Данная организация дешифраторов, как правило, применяется при внутренней дешифрации адресов в микросхемах памяти, обладает наименьшими аппаратными затратами

и высоким быстродействием. На 17 приведена каскадная структура для дешифратора D4\_16 и матричным выходом.

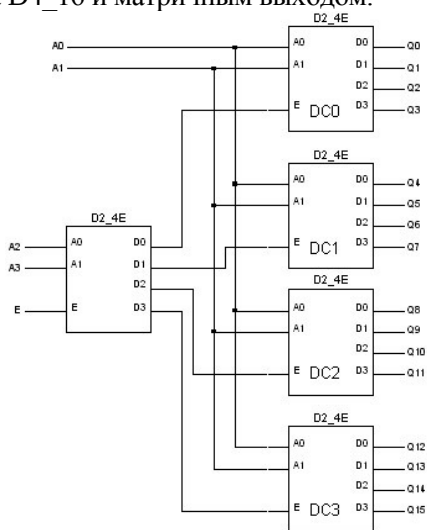


Рис. 16. Логическая схема дешифратора D4\_16 каскадной структуры

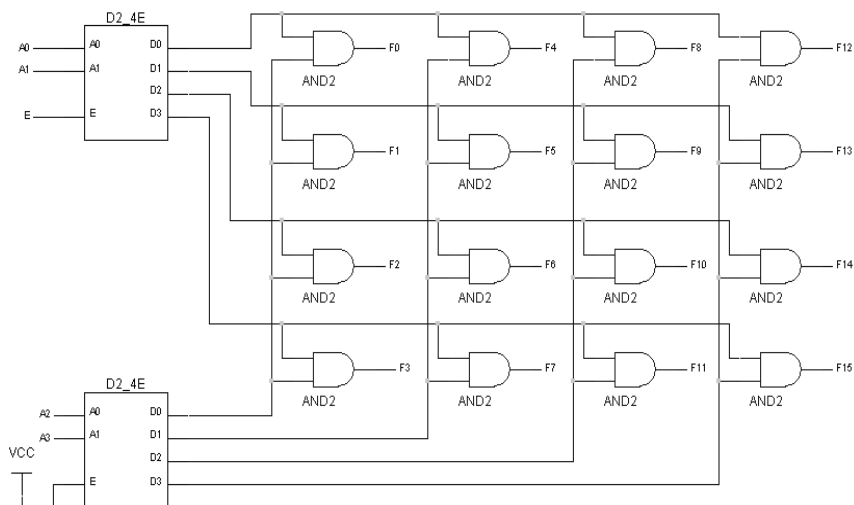


Рис. 17. Логическая схема дешифратора D4\_16 каскадной структуры и матричным выходом

## Реализация функций на дешифраторах

Дешифраторы можно использовать для генерации произвольных логических функций. На выходах дешифратора вырабатываются все  $N$  элементарных конъюнктивных термов, которые можно составить из данного числа аргументов  $K$ , поступающих на адресные входы. Логическая функция в СДНФ может быть реализована как дизъюнкция некоторого числа таких термов. Подключая к входам схемы ИЛИ нужные термы, можно получить любую функцию для данного числа аргументов.

На рис. 18 приведена реализация одnorазрядного полного вычитателя на дешифраторе D3\_8E. На входы вычитателя поступают сигналы уменьшаемого  $A$ , вычитаемого  $C$  и сигнал входного заёма  $BI$ , поступающий из младшего разряда. На выходе формируются сигналы разности  $D$  и выходного заёма  $BO$ , поступающего в старший разряд.

| Входы |   |    | Выходы |   | Выходы |
|-------|---|----|--------|---|--------|
| A     | C | BI | BO     | D | DC     |
| 0     | 0 | 0  | 0      | 0 | Q0     |
| 0     | 0 | 1  | 1      | 1 | Q1     |
| 0     | 1 | 0  | 1      | 1 | Q2     |
| 0     | 1 | 1  | 1      | 0 | Q3     |
| 1     | 0 | 0  | 0      | 1 | Q4     |
| 1     | 0 | 1  | 0      | 0 | Q5     |
| 1     | 1 | 0  | 0      | 0 | Q6     |
| 1     | 1 | 1  | 1      | 1 | Q7     |

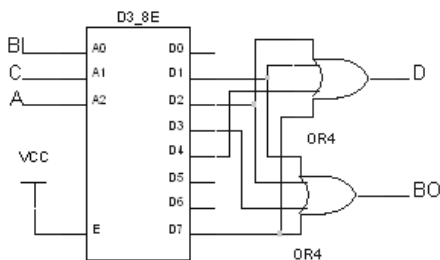


Рис. 18. Таблица истинности и логическая схема одnorазрядного вычитателя

На рис. 19 приведена схема реализации дополнения до 9. Сумма входного кода  $X_3, X_2, X_1, X_0$  и выходного кода  $Y_3, Y_2, Y_1, Y_0$  равна 9. Из таблицы видно, что значение младшего разряда  $Y_0$  инверсно входу  $X_0$ . Первые разряды  $X_1$  и  $Y_1$  равны. В таблице истинности в правом столбце показаны выходы дешифратора, если на адресные входы дешифратора поданы сигналы  $X_2, X_1$ . Чтобы получить значение сигнала  $Y_2$ , необходимо объединить по ИЛИ выходы дешифратора  $Q_1$  и  $Q_2$ . Сигнал  $Y_3$  можно получить с выхода  $Q_0$  дешифратора, если на разрешающий вход  $E$  подать входной сигнал  $X_3$ .



синхронизации их работы с работой других узлов. Сигналы на разрешающих входах могут разрешать, а могут и запрещать подключение определенного входа к выходу, т.е. могут блокировать действие всего устройства.

Условное графическое обозначение мультиплексора приведено на рис. 20. Количество адресных входов зависит от количества информационных входов и определяется как  $K \leq \log_2 N$ .



Рис. 20. Условное графическое обозначение мультиплексора

На рис. 21. приведены УГО, таблица истинности и логическая схема мультиплексора M2\_1E (при  $N=2$  и  $K=1$ ). Когда разрешающий вход  $E=1$ , то мультиплексор M2\_1E выбирает один бит данных из двух информационных входов  $D1$  или  $D0$  в зависимости от состояния управляющего (адресного) входа  $A0$  ( $S0$ ). При  $E=0$  состояние выхода равно «0». В соответствии с таблицей истинности мультиплексор M2\_1E с входом разрешения  $E$  реализует следующую функцию:

$$F_{M2\_1} = E \cdot \neg A0 \cdot D0 + E \cdot A0 \cdot D1.$$

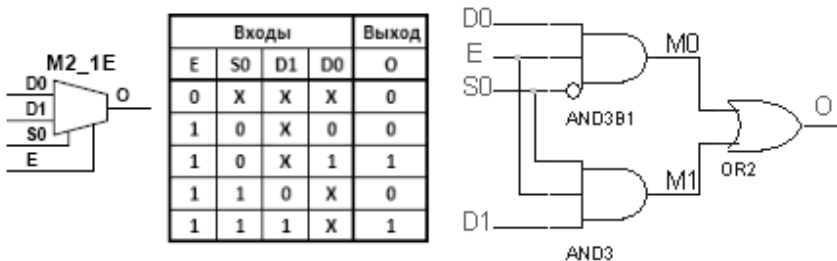


Рис. 21. УГО, таблица истинности и логическая схема мультиплексора M2\_1E

Таблица истинности для четырёхходового мультиплексора М4\_1Е приведена в таблице 1 (при  $N=4$  и  $K=2$ ). Когда разрешающий вход  $E=1$ , то мультиплексор М4\_1Е выбирает один бит данных из четырёх информационных входов  $D3$ ,  $D2$ ,  $D1$  или  $D0$  в зависимости от состояния управляющих (адресных) входов  $A1$ – $A0$  ( $S1$ – $S0$ ). При  $E=0$  состояние выхода равно «0».

Таблица 1

| Входы |                  |                  |                |      |      |      | Выход |      |
|-------|------------------|------------------|----------------|------|------|------|-------|------|
| $E$   | Адресные         |                  | Информационные |      |      |      | $O$   |      |
|       | $A1$<br>( $S1$ ) | $A0$<br>( $S0$ ) | $D3$           | $D2$ | $D1$ | $D0$ |       |      |
| 0     | 0                | 0                | X              | X    | X    | 0    | 0     | $D0$ |
|       |                  |                  |                |      |      | 1    | 1     |      |
| 0     | 0                | 1                | X              | X    | 0    | X    | 0     | $D1$ |
|       |                  |                  |                |      | 1    |      | 1     |      |
| 0     | 1                | 0                | X              | 0    | X    | X    | 0     | $D2$ |
|       |                  |                  |                | 1    |      |      | 1     |      |
| 0     | 1                | 1                | 0              | X    | X    | X    | 0     | $D3$ |
|       |                  |                  | 1              |      |      |      | 1     |      |
| 1     | X                | X                | X              | X    | X    | X    | 0     |      |

В соответствии с таблицей 1 мультиплексор М4\_1Е с входом разрешения  $E$  реализует следующую функцию:

$$F_{M4\_1} = E \cdot \neg A1 \cdot \neg A0 \cdot D0 + E \cdot \neg A1 \cdot A0 \cdot D1 + E \cdot A1 \cdot \neg A0 \cdot D0 + E \cdot A1 \cdot A0 \cdot D1.$$

Реализация мультиплексора М4\_1Е приведена на рис. 22.

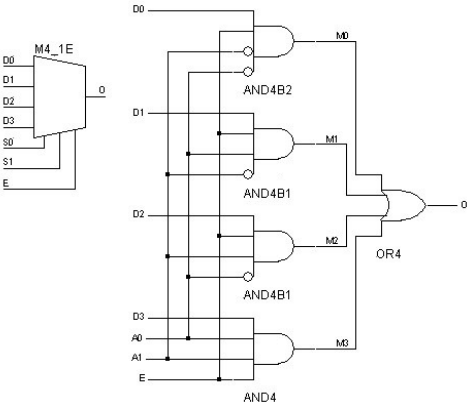


Рис. 22. УГО и логическая схема мультиплексора М4\_1Е

Мультиплексор M4\_1E может быть реализован на одном конфигурируемом логическом блоке (CLB) ПЛИС FPGA. Для этого в генераторы логических функций, реализованные на трех таблицах преобразования F-LUT, G-LUT и H-LUT, загружают значения в соответствии с таблицами истинности, приведенными на рис. 23.

| G-LUT |    |    |    |       | F-LUT |    |    |    |       | H-LUT |   |   |       |
|-------|----|----|----|-------|-------|----|----|----|-------|-------|---|---|-------|
| Входы |    |    |    | Выход | Входы |    |    |    | Выход | Входы |   |   | Выход |
| G4    | G3 | G2 | G1 |       | F4    | F3 | F2 | F1 |       | H1    | G | F | H     |
| E     | A0 | D1 | D0 | G     | E     | A0 | D3 | D2 | F     | A1    |   |   |       |
| 0     | 0  | 0  | 0  | 0     | 0     | 0  | 0  | 0  | 0     | 0     | 0 | 0 | 0     |
| 0     | 0  | 0  | 1  | 0     | 0     | 0  | 0  | 1  | 0     | 0     | 0 | 1 | 0     |
| 0     | 0  | 1  | 0  | 0     | 0     | 0  | 1  | 0  | 0     | 0     | 1 | 0 | 1     |
| 0     | 0  | 1  | 1  | 0     | 0     | 0  | 1  | 1  | 0     | 0     | 1 | 1 | 1     |
| 0     | 1  | 0  | 0  | 0     | 0     | 1  | 0  | 0  | 0     | 1     | 0 | 0 | 0     |
| 0     | 1  | 0  | 1  | 0     | 0     | 1  | 0  | 1  | 0     | 1     | 0 | 1 | 1     |
| 0     | 1  | 1  | 0  | 0     | 0     | 1  | 1  | 0  | 0     | 1     | 1 | 0 | 0     |
| 0     | 1  | 1  | 1  | 0     | 0     | 1  | 1  | 1  | 0     | 1     | 1 | 1 | 1     |
| 1     | 0  | 0  | 0  | 0     | 1     | 0  | 0  | 0  | 0     |       |   |   |       |
| 1     | 0  | 0  | 1  | 1     | 1     | 0  | 0  | 1  | 1     |       |   |   |       |
| 1     | 0  | 1  | 0  | 0     | 1     | 0  | 1  | 0  | 0     |       |   |   |       |
| 1     | 0  | 1  | 1  | 1     | 1     | 0  | 1  | 1  | 1     |       |   |   |       |
| 1     | 1  | 0  | 0  | 0     | 1     | 1  | 0  | 0  | 0     |       |   |   |       |
| 1     | 1  | 0  | 1  | 0     | 1     | 1  | 0  | 1  | 0     |       |   |   |       |
| 1     | 1  | 1  | 0  | 1     | 1     | 1  | 1  | 0  | 1     |       |   |   |       |
| 1     | 1  | 1  | 1  | 1     | 1     | 1  | 1  | 1  | 1     |       |   |   |       |

Рис. 23. Таблицы истинности мультиплексора M4-1E для генераторов логических функций

### *Увеличение разрядности мультиплексоров*

Разрядность мультиплексоров можно увеличивать линейно или каскадно (пирамидально).

При линейном методе с ростом количества информационных входов  $N$  пропорционально количеству адресных входов  $K$  увеличивается количество входов  $(K+2)$  у элементов И и пропорционально входам  $N$  количество входов у элемента ИЛИ. Количество входов элементов И и ИЛИ для мультиплексоров приведены в таблице 2.

Таблица 2

| MUX    | Элементы И        |                      | Элемент ИЛИ       |
|--------|-------------------|----------------------|-------------------|
|        | Количество входов | Количество элементов | Количество входов |
| M2_1E  | 3                 | 2                    | 2                 |
| M4_1E  | 4                 | 4                    | 4                 |
| M8_1E  | 5                 | 8                    | 8                 |
| M16_1E | 7                 | 16                   | 16                |

Другим вариантом увеличения разрядности мультиплексора является применение на входе мультиплексора дешифратора для выбора информационного входа, набора элементов И и элемента ИЛИ. Логическая схема мультиплексора M4\_1E с применением дешифратора приведена на рис. 24.

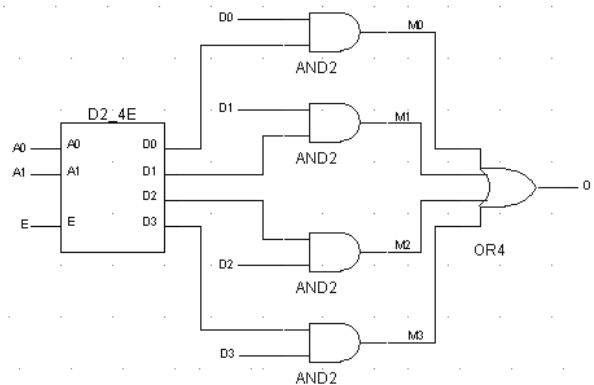


Рис. 24. Логическая схема мультиплексора M4\_1E

Пирамидальные мультиплексоры имеют древовидную структуру и выполняются объединением нескольких мультиплексоров по

ступенчатому (каскадному) принципу. При этом применяются две, три и более ступеней. Пирамидальный характер схемы состоит в том, что каждая ступень, начиная с первой, содержит больше мультиплексоров, чем последующая. Одноимённые адресные входы в каждой ступени объединяются между собой. Младшие разряды кода адреса мультиплексора подают на адресные входы первой ступени, а ступеням более высокого ранга соответствуют старшие разряды. Недостатком пирамидального увеличения разрядности является снижение быстродействия из-за суммирования задержек при последовательном прохождении сигналов по ступеням пирамиды.

Общее число информационных входов при одностипных мультиплексорах младшего ранга равно произведению числа входов отдельных мультиплексоров и числа самих мультиплексоров.

Логическая схема пирамидального мультиплексора М8\_1Е приведена на рис. 25. Для мультиплексора М8\_1Е применяется четыре мультиплексора М2-1 и три мультиплексора М2-1Е объединенные в три ступени.

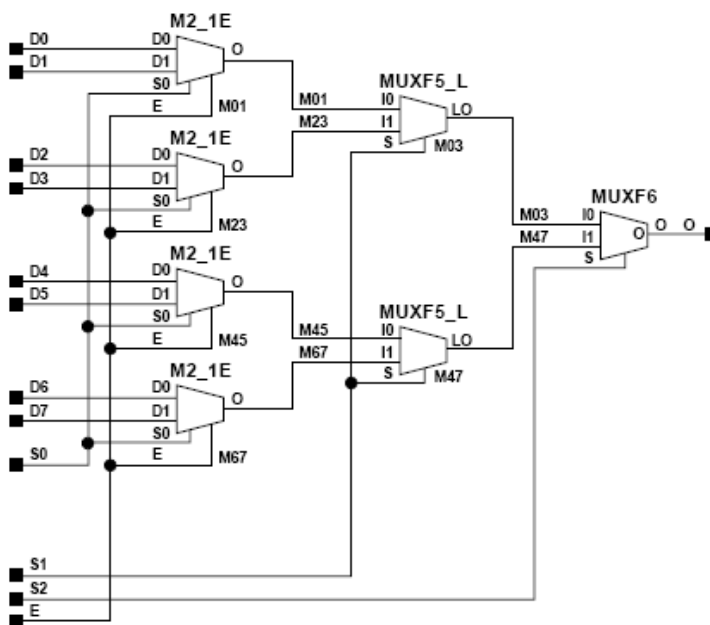


Рис. 25. Логическая схема мультиплексора М8\_1Е

Мультиплексор M16\_1E может быть построен аналогично на пятнадцати мультиплексорах M2\_1 и M2\_1E, но объединённых уже в четыре ступени. Другими вариантами мультиплексора M16\_1 может быть реализация на пяти мультиплексорах M4-1, объединённых в две ступени или на двух мультиплексорах M8\_1 и одном мультиплексоре M2-1, также объединённых в две ступени.

Еще одним вариантом увеличения разрядности мультиплексоров является применение в первой ступени дешифратора. Старшие разряды кода адреса декодируются дешифратором и его выходные сигналы служат сигналами разрешения для мультиплексоров. Так как в каждый момент времени разрешена работа только одного мультиплексора, то результирующий выходной сигнал можно получить объединением выходов всех мультиплексоров с помощью схемы ИЛИ. Логическая схема мультиплексора M16\_1E приведена на рис. 26.

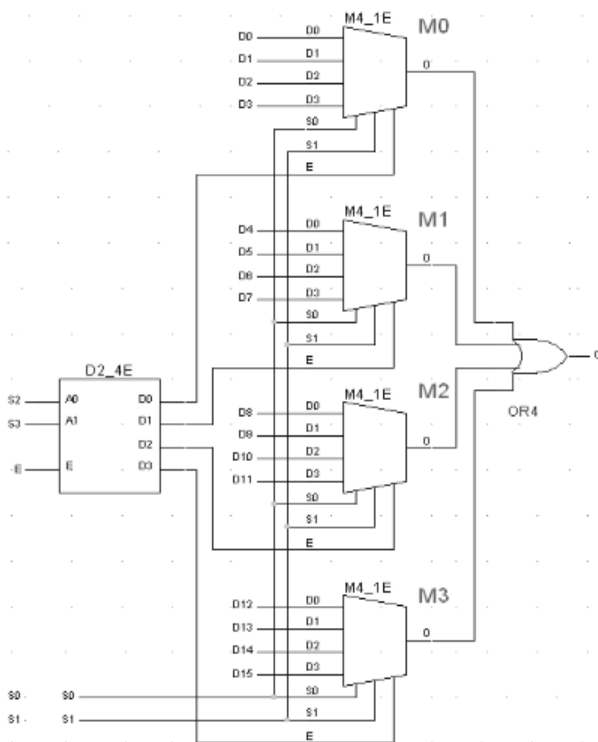


Рис. 26. Логическая схема мультиплексора M16\_1E

Выходная схема ИЛИ может быть исключена, а выходы мультиплексоров просто соединяются вместе «Монтажным ИЛИ», если мультиплексоры имеют выходы с третьим состоянием или с открытым коллектором. Дешифратор гарантирует, что в любой момент времени активирован только один мультиплексор.

### *Реализация функций на мультиплексорах*

На мультиплексорах достаточно просто можно реализовать любую функцию от заданного числа  $n$  аргументов (переменных). Известно, что общее число функций  $N$  от  $n$  переменных выражается как  $N = 2^{2^n}$ . Например, для двух переменных ( $n=2$ ) количество функций будет равно  $N = 2^{2^2} = 16$ . С ростом числа  $n$  количество функций растет чрезвычайно быстро, хотя практический интерес представляют не все существующие функции.

При реализации (генерации) функций на мультиплексорах не требуется ни записи функции в СДНФ, ни минимизации функции. Для реализации функции применяются таблицы истинности. Применяется три способа настройки мультиплексоров:

1. Число адресных входов  $m$  равно числу переменных  $n$ .
2. Число адресных входов  $m$  равно  $(n - 1)$  для  $n$  переменных.
3. Применение дополнительных схем на входе мультиплексора.

В первом способе на адресные входы мультиплексора подают аргументы функции  $X_n, \dots, X_1$ , а на информационные входы коды настройки – константы 0 и 1, в такой последовательности, которая полностью копирует последовательность единиц и нулей таблицы истинности. На рис. 27 приведён пример воспроизведения функции «голосования» – мажоритарного элемента 2 из 3 на мультиплексоре M8\_1E.

Во втором способе от алфавита  $\{0, 1\}$  переходят к алфавиту  $\{0, 1, \bar{X}_i\}$ , где  $\bar{X}_i$  – литерал одного из аргументов. Под литералом переменной понимается либо сама переменная  $X_i$ , либо ее инверсия  $\bar{X}_i$ . При этом число адресных входов мультиплексора уменьшится на единицу, а число информационных входов уменьшится вдвое. В качестве литерала  $\bar{X}_i$  целесообразно использовать аргумент, который имеет минимальное число вхождений в термы

функции. В этом случае будут максимально использованы внутренние логические ресурсы мультиплексора, а среди сигналов настройки увеличится число констант «0» и «1», что считается благоприятным для схемной реализации.



Рис. 27. Схема мажоритарного элемента на мультиплексоре M8\_1E

В таблице истинности целесообразно расположить аргумент  $\overline{Xi}$ , как младший разряд входных сигналов (справа среди входов). Далее таблица истинности разбивается на блоки по две строки. При этом старшие аргументы в каждом блоке постоянны и они подаются на соответствующие адресные входы мультиплексора. В каждом блоке заданная функция реализуется как функция единственного аргумента  $\overline{Xi}$  или констант «0» и «1», данные значения подаются на соответствующие информационные входы мультиплексора. Рассмотрим реализацию одноразрядного полного сумматора на двойном мультиплексоре M4\_1E (рис. 28).

В качестве аргумента  $\overline{Xi}$  применяется сигнал второго слагаемого  $B$ . Поэтому выходы сумматора реализуются как функции от аргумента  $B$  – суммы  $S(B)$  и переноса  $CO(B)$ . При этом для инверсии аргумента введен инвертор INV.

Мультиплексор X74\_153 является двухразрядным четырёхходовым мультиплексором и состоит из двух мультиплексоров. Первый мультиплексор M4\_1E передает на выход Y1 значения входов I1C0–I1C3 при разрешающем сигнале G1=0, а второй мультиплексор M4\_1E передает на выход Y2 значения входов I2C0–I2C3 при разрешающем сигнале G2=0. Выбор информационного входа определяется адресными входами  $A$  (младший разряд) и  $B$  (старший

разряд). На выходе Y1 формируется значение сигнала переноса  $CO$ , а на выходе Y2 – значение суммы, которые реализуются как функции от второго слагаемого  $B$  (см. таблицу истинности).

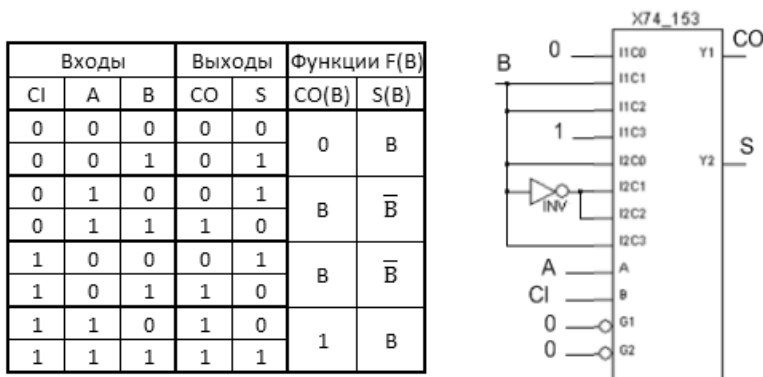


Рис. 28. Схема полного сумматора на мультиплексорах M4\_1E

Таким образом на мультиплексоре M4\_1 можно реализовать любую из 256 функций от трёх переменных, на мультиплексоре M8\_1 любую из 65536 функций от четырёх переменных, а на мультиплексоре M16\_1 любую из более четырёх миллиардов функций от пяти переменных.

При третьем способе реализации функций на мультиплексоре на входах мультиплексора устанавливается блок предварительной реализации сигналов настройки информационных входов, которые реализуются как функции от двух, трёх и более числа аргументов. В этом случае на входах мультиплексора вводятся дополнительные логические элементы И, ИЛИ, «Исключающее ИЛИ», дешифратор, сумматор и др. Применение данного способа будет показано в приведённых далее примерах.

### ***Сдвоенный десятивходовый мультиплексор***

При коммутации многоразрядных слов в каждом разряде используется свой мультиплексор. Рассмотрим реализацию сдвоенного мультиплексора M10\_1E на двух мультиплексорах M8\_1E и четырёхразрядном двухвходовом мультиплексоре X74\_157 (рис. 29). Данная схема реализует параллельную выборку двух раз-

рядов из двух 10-разрядных каналов информации  $DA9-DA0$  и  $DB9-DB0$ . Мультиплексор  $X74\_157$  выбирает две пары либо младших разрядов  $DA1, DA0$  и  $DB1, DB0$ , либо старших разрядов  $DA9, DA8$  и  $DB9, DB8$  под управлением старшего разряда адреса  $A3$ . Далее два мультиплексора  $M8\_1E$  одновременно выбирают один из восьми сигналов по двум каналам под управлением адресных входов  $A2, A1, A0$  и разрешающем сигнале  $E=1$ .

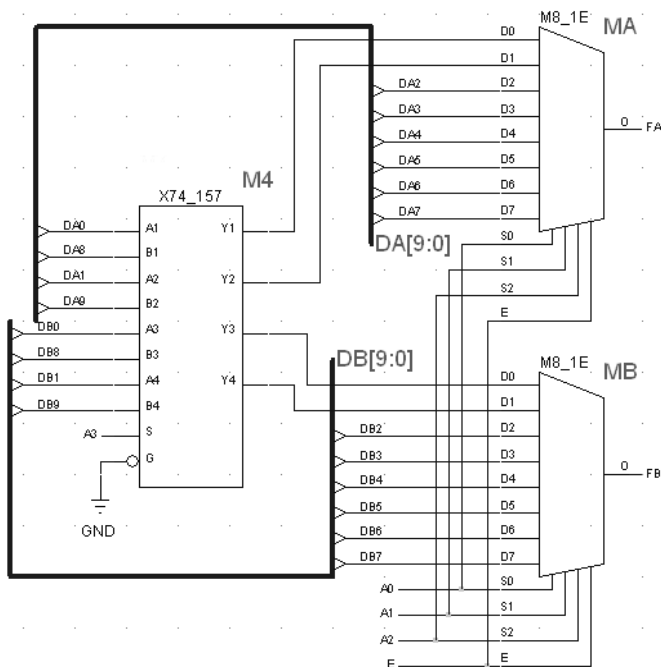


Рис. 29. Схема двояного мультиплексора  $M10\_1E$

### *Детектор $M$ из $N$*

При обработке параллельных данных, например, в кодах с исправлением ошибок, возникает проблема выделения заданного числа (или заданного набора) из входных чисел. Разработка такого детектора путем прямой минимизации логических элементов весьма сложна и обычно не эффективна. Решение задачи может упро-

ститься при применении мультиплексоров или сумматоров или их комбинации.

Рассмотрим реализацию детектора при  $M=3$  и  $N=6$ . Детектор 3 из 6 формирует сигнал «1», если во входном шестиразрядном двоичном числе  $I5-I0$  присутствует только три единицы. В схеме детектора 3 из 6 (рис. 30) на входе использованы полные сумматоры ADD1, которые уменьшают число входов до четырех переменных и затем применён мультиплексор M8\_1 для формирования заданной функции от четырёх переменных.

| SM1 |    | SM0 |    | Сумма<br>единиц | Выход<br>F<br>3 из 6 | Выход<br>F(S0) |
|-----|----|-----|----|-----------------|----------------------|----------------|
| CO  | S0 | CO  | S0 |                 |                      |                |
| 2   | 1  | 2   | 1  |                 |                      |                |
| 0   | 0  | 0   | 0  | 0               | 0                    | 0              |
| 0   | 0  | 0   | 1  | 1               | 0                    |                |
| 0   | 0  | 1   | 0  | 2               | 0                    |                |
| 0   | 0  | 1   | 1  | 3               | 1                    | S0             |
| 0   | 1  | 0   | 0  | 1               | 0                    |                |
| 0   | 1  | 0   | 1  | 2               | 0                    |                |
| 0   | 1  | 1   | 0  | 3               | 1                    | S0             |
| 0   | 1  | 1   | 1  | 4               | 0                    |                |
| 1   | 0  | 0   | 0  | 2               | 0                    |                |
| 1   | 0  | 0   | 1  | 3               | 1                    | 0              |
| 1   | 0  | 1   | 0  | 4               | 0                    |                |
| 1   | 0  | 1   | 1  | 5               | 0                    |                |
| 1   | 1  | 0   | 0  | 3               | 1                    | S0             |
| 1   | 1  | 0   | 1  | 4               | 0                    |                |
| 1   | 1  | 1   | 0  | 5               | 0                    |                |
| 1   | 1  | 1   | 1  | 6               | 0                    | 0              |

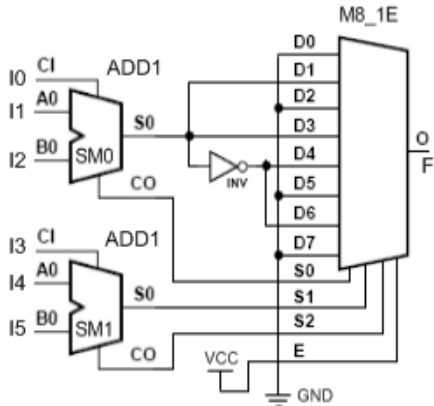


Рис. 30. Логическая схема детектора 3 из 6

На выходах каждого сумматора формируются сигнал суммы  $S0=1$ , если в трёх разрядах входного числа одна или три единицы (нечётное количество), и сигнал переноса  $CO=1$ , если во входном числе две или три единицы. Будем считать, что для каждого сумматора выход суммы  $S0$  имеет вес равный единице, а выход переноса  $CO$  – вес равный двум. В таблице истинности (рис. 30) отмечены веса разрядов выходов сумматоров SM1 и SM0. В графе «сумма единиц» приведены суммы разрядов исходного числа  $I5-I0$ . Вы-

ходная функция  $F=1$ , если исходное число содержит три единицы (сумма равна трем). Для реализации на мультиплексоре M8\_1 проведено объединение по две строки и в таблице истинности приведено значение сформированной функции  $F(SO)$  в зависимости от сигнала суммы  $SO$  сумматора  $SM0$ .

На рис. 31 приведена схема детектора для набора 3-6, когда единицы присутствуют на трех, четырех, пяти или шести входах восьмиразрядного числа  $I7-I0$ .

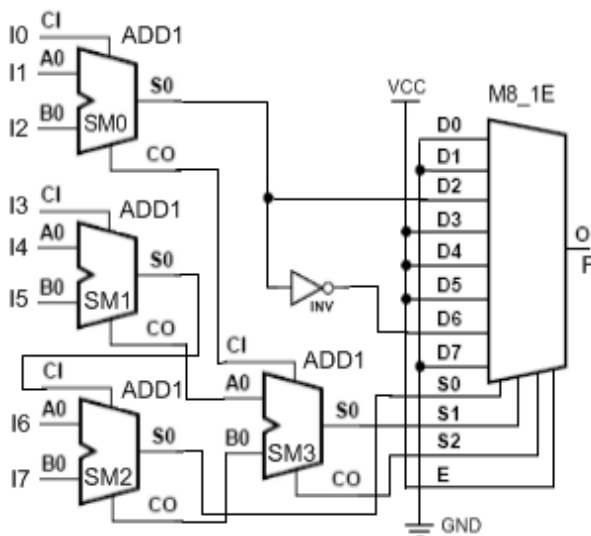


Рис. 31. Логическая схема детектора для набора 3-6 из 8

Схема детектора реализована на трёх полных одноразрядных сумматорах ADD1 и мультиплексоре M8\_1E. На выходах суммы  $S0$  сумматоров  $SM0$ ,  $SM1$  и  $SM2$  формируются сигналы с весами разрядов равными единице. При этом выход суммы  $S0$  сумматора  $SM1$  поступает на вход сумматора  $SM2$ . На входы сумматора  $SM3$  поступают сигналы переносов  $CO$  сумматоров  $SM0$ ,  $SM1$  и  $SM2$  с весами равными двум, поэтому на выходе суммы  $S0$  сумматора  $SM3$  формируется сигнал с весом равным двум, т.е. когда во входном числе присутствует две единицы. На выходе переноса  $CO$  сумматора  $SM3$  формируется сигнал с весом равным четырём, т.е. когда во входном числе присутствуют не менее четырёх единиц. На рис. 32 приведена таблица истинности для реализации детектора для набора 3-6 из 8.

| SM3 |    | SM2 |    | SM0           | Сумма<br>единиц | Выход | Выход |
|-----|----|-----|----|---------------|-----------------|-------|-------|
| CO  | SO | SO  | SO | F 3-6<br>из 8 |                 | F(SO) |       |
| 4   | 2  | 1   | 1  |               |                 |       |       |
| 0   | 0  | 0   | 0  | 0             | 0               | 0     | 0     |
| 0   | 0  | 0   | 1  | 1             | 0               |       |       |
| 0   | 0  | 1   | 0  | 1             | 0               | 0     | 0     |
| 0   | 0  | 1   | 1  | 2             | 0               |       |       |
| 0   | 1  | 0   | 0  | 2             | 0               | SO    | SO    |
| 0   | 1  | 0   | 1  | 3             | 1               |       |       |
| 0   | 1  | 1   | 0  | 3             | 1               | 1     | 1     |
| 0   | 1  | 1   | 1  | 4             | 1               |       |       |
| 1   | 0  | 0   | 0  | 4             | 1               | 1     | 1     |
| 1   | 0  | 0   | 1  | 5             | 1               |       |       |
| 1   | 0  | 1   | 0  | 5             | 1               | 1     | 1     |
| 1   | 0  | 1   | 1  | 6             | 1               |       |       |
| 1   | 1  | 0   | 0  | 6             | 1               | SO    | SO    |
| 1   | 1  | 0   | 1  | 7             | 0               |       |       |
| 1   | 1  | 1   | 0  | 7             | 0               | 0     | 0     |
| 1   | 1  | 1   | 1  | 8             | 0               |       |       |

Рис. 32. Таблица истинности для реализации детектора для набора 3-6 из 8

В таблице приведены значения суммы единиц в исходном числе  $I7-I0$  и значение сформированной функции  $F(SO)$  в зависимости от сигнала суммы  $SO$  сумматора  $SM0$ .

### *Схема формирования сигнала при делении числа на 7*

Данная задача иллюстрирует третий способ реализации функций на мультиплексоре. Экономичное решение реализации комбинационной схемы может быть получено при подходе к системе функций с учётом их взаимосвязей и выделения общих частей.

Примером реализации функций является применение для предварительного преобразования сигналов дешифратора, выходы которого соединяются с входами мультиплексора.

На рис. 33 приведены сокращенные таблицы истинности для формирования сигнала при делении на семь входного шестираз-

рядного числа  $X_5$ – $X_0$ . В таблице на рис. 33,*а* на адресные входы дешифратора D3\_8E подаются три старших разряда  $X_5$ ,  $X_4$  и  $X_3$  входного числа, а на адресные входы мультиплексора M8\_1E – младшие разряды  $X_2$ ,  $X_1$  и  $X_0$ . В таблице представлены все 64 набора от входных переменных – на пересечении выходов дешифратора  $Q_i$  (строка) и информационных входов мультиплексора при соответствующих значениях адресных входов. Наборы деления на 7 отмечены единицами, а значения наборов приведены в крайнем правом столбце. Принято, что на нулевом входном наборе  $X_i=000000$  функция тоже равна единице -  $F_7=1$ . При анализе таблицы видно, что каждый вход мультиплексора должен соединяться с одним выходом дешифратора, кроме входов  $D_0$  и  $D_7$  мультиплексора. Например, первый выход дешифратора  $Q_1$  должен быть соединен с шестым входом  $D_6$  мультиплексора.

Реализация схемы формирования сигнала при делении числа на 7 приведена на рис. 34. Входы  $D_0$  и  $D_7$  мультиплексора соединены с элементом ИЛИ, который объединяет выходы  $Q_0$  и  $Q_7$  дешифратора.

В таблице на рис. 33,*б* на адресные входы дешифратора D3\_8E подаются три младших разряда  $X_2$ ,  $X_1$  и  $X_0$  входного числа, а на адресные входы мультиплексора M8\_1E – старшие разряды  $X_5$ ,  $X_4$  и  $X_3$ . Как видно из таблицы входы мультиплексора аналогично соединяются с выходами дешифратора, но наборы деления на 7 будут соответствовать другим информационным входам мультиплексора (правый столбец таблицы).

### ***Схема быстрого сдвига на $K$ разрядов***

Устройство быстрого сдвига является частью микропроцессора и обычно характеризуется такими параметрами, как направление сдвига (вправо или влево), вид сдвига (логический, циклический или арифметический) и величина сдвига (на  $0$ – $K$  разрядов).

Для  $n$ -разрядных чисел сдвиги можно реализовать с помощью  $n$  мультиплексоров, имеющих  $m \geq \log_2 K$  адресных входов определяющих величину сдвига, а на входы данных поступают соответствующие разряды входных данных. Данное решение будет иметь высокое быстродействие, но большой объем оборудования.

| Дешифратор |    |    |       | Мультиплексор           |     |     |     |     |     |     |     | Наборы  |
|------------|----|----|-------|-------------------------|-----|-----|-----|-----|-----|-----|-----|---------|
| Входы      |    |    | Выход | Адресные входы X2 X1 X0 |     |     |     |     |     |     |     | деления |
| X5         | X4 | X3 |       | 111                     | 110 | 101 | 100 | 011 | 010 | 001 | 000 | на 7    |
| 0          | 0  | 0  | Q0    | 1                       | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 7, 0    |
| 0          | 0  | 1  | Q1    | 0                       | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 14      |
| 0          | 1  | 0  | Q2    | 0                       | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 21      |
| 0          | 1  | 1  | Q3    | 0                       | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 28      |
| 1          | 0  | 0  | Q4    | 0                       | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 35      |
| 1          | 0  | 1  | Q5    | 0                       | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 42      |
| 1          | 1  | 0  | Q6    | 0                       | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 49      |
| 1          | 1  | 1  | Q7    | 1                       | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 63, 56  |

а)

| Дешифратор |    |    |       | Мультиплексор           |     |     |     |     |     |     |     | Наборы<br>деления<br>на 7 |
|------------|----|----|-------|-------------------------|-----|-----|-----|-----|-----|-----|-----|---------------------------|
| Входы      |    |    | Выход | Адресные входы X5 X4 X3 |     |     |     |     |     |     |     |                           |
| X2         | X1 | X0 |       | 111                     | 110 | 101 | 100 | 011 | 010 | 001 | 000 |                           |
| 0          | 0  | 0  | Q0    | 1                       | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 56, 0                     |
| 0          | 0  | 1  | Q1    | 0                       | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 49                        |
| 0          | 1  | 0  | Q2    | 0                       | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 42                        |
| 0          | 1  | 1  | Q3    | 0                       | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 35                        |
| 1          | 0  | 0  | Q4    | 0                       | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 27                        |
| 1          | 0  | 1  | Q5    | 0                       | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 21                        |
| 1          | 1  | 0  | Q6    | 0                       | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 14                        |
| 1          | 1  | 1  | Q7    | 1                       | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 63, 7                     |

б)

Рис. 33. Таблицы истинности для схемы проверки деления числа на 7

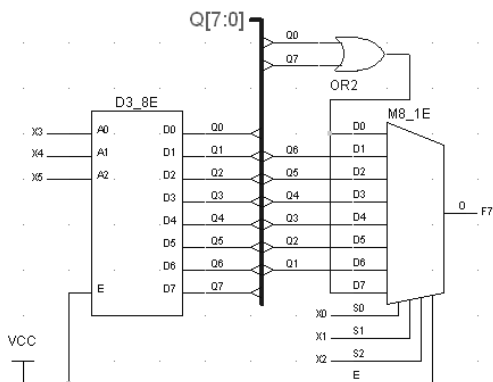


Рис. 34. Схема формирования сигнала при делении числа на 7

Другим вариантом устройства быстрого сдвига является построение схемы на 2-входовых мультиплексорах M2-1. При этом схема содержит  $m$  каскадов последовательно соединённых. В первом каскаде выполняется сдвиг на 0 разрядов или на 1 разряд в зависимости от значения младшего разряда  $k_0$  кода величины сдвига. Выходы первого каскада соединены с входами второго каскада, которые по сигналу  $k_1$  сдвигают входные данные на 0 разрядов или на 2 разряда. В третьем каскаде выполняется сдвиг на 0 разрядов или на 4 разряда под управлением сигнала  $k_2$  и т.д. для последующих каскадов. Сигналы  $k_0, k_1, k_2, \dots$  поступают на адресные входы мультиплексоров M2\_1 соответствующих каскадов.

На рис. 35. приведена логическая схема устройства циклического сдвига на 0-7 разрядов, а на рис. 36. приведены его УГО и таблица истинности.

Сравним два варианта построения схемы быстрого циклического сдвига 8-разрядного числа на 0-7 разрядов. В первом варианте необходимо восемь мультиплексоров M8-1, а во втором варианте двадцать четыре мультиплексора M2-1. Во втором варианте (см. рис. 35) входные данные проходят последовательно через три канала, т.е. схема дает большую задержку. Но если учитывать, что мультиплексор M8\_1 имеет пирамидальную структуру (см. рис. 25) и для его реализации необходимо три ступени и семь мультиплексоров M2\_1, то для первого варианта схемы быстрого сдвига общее число мультиплексоров M2\_1 составит пятьдесят шесть. При этом задержка также составит три мультиплексора M2\_1. Кроме того, в первом варианте схемы увеличивается нагрузка на входные сигналы данных, которые должны поступать на мультиплексоры в каждом разряде. Поэтому необходимо включать буферы, чтобы минимизировать нагрузку, но при этом могут увеличиться задержки.

Для реализации схемы сдвига 16-разрядного числа на 0-15 разрядов выигрыш аппаратуры более значителен: в первом варианте необходимо 16 мультиплексоров M16\_1 (при пирамидальном построении для каждого мультиплексора M16\_1 необходимо четыре ступени и 15 мультиплексоров M2\_1) или 240 мультиплексоров M2\_1, во втором варианте – также четыре каскада и только 64 мультиплексора M2\_1.

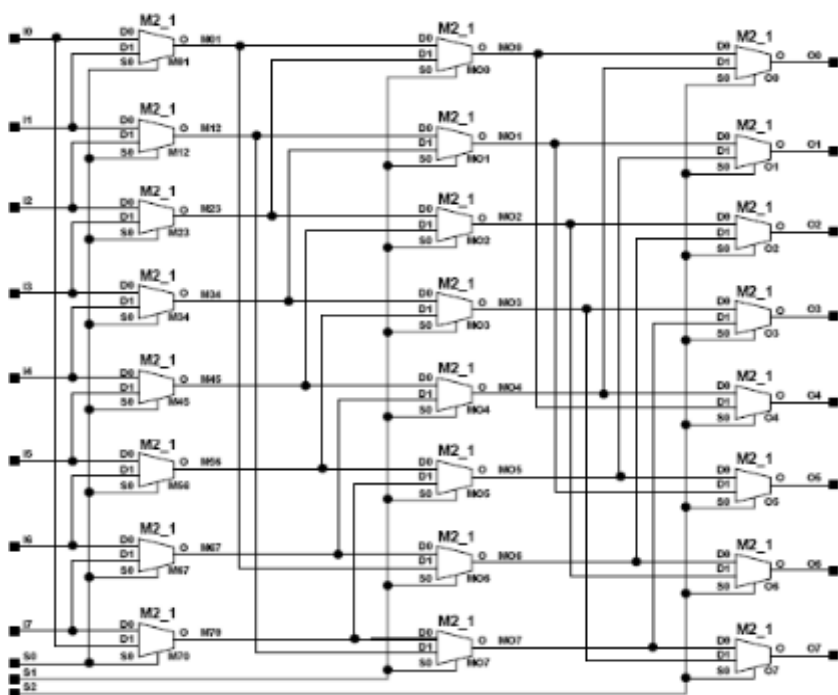


Рис. 35. Логическая схема циклического сдвига на 0-7 разрядов

|    |          | Inputs |    |    |    |    |    |    |    |    |    | Outputs |    |    |    |    |    |    |    |    |
|----|----------|--------|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----|
|    | BRLSHFT8 | S2     | S1 | S0 | I0 | I1 | I2 | I3 | I4 | I5 | I6 | I7      | O0 | O1 | O2 | O3 | O4 | O5 | O6 | O7 |
| 10 |          | 0      | 0  | 0  | a  | b  | c  | d  | e  | f  | g  | h       | a  | b  | c  | d  | e  | f  | g  | h  |
| 11 |          | 0      | 0  | 1  | a  | b  | c  | d  | e  | f  | g  | h       | b  | c  | d  | e  | f  | g  | h  | a  |
| 12 |          | 0      | 1  | 0  | a  | b  | c  | d  | e  | f  | g  | h       | c  | d  | e  | f  | g  | h  | a  | b  |
| 13 |          | 0      | 1  | 1  | a  | b  | c  | d  | e  | f  | g  | h       | d  | e  | f  | g  | h  | a  | b  | c  |
| 14 |          | 1      | 0  | 0  | a  | b  | c  | d  | e  | f  | g  | h       | e  | f  | g  | h  | a  | b  | c  | d  |
| 15 |          | 1      | 0  | 1  | a  | b  | c  | d  | e  | f  | g  | h       | f  | g  | h  | a  | b  | c  | d  | e  |
| 16 |          | 1      | 1  | 0  | a  | b  | c  | d  | e  | f  | g  | h       | g  | h  | a  | b  | c  | d  | e  | f  |
| 17 |          | 1      | 1  | 1  | a  | b  | c  | d  | e  | f  | g  | h       | h  | a  | b  | c  | d  | e  | f  | g  |
| S0 |          |        |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |
| S1 |          |        |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |
| S2 |          |        |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |

Рис. 36. УГО и таблица истинности схемы циклического сдвига на 0-7 разрядов

## *Демультимплексоры и шины*

Демультимплексор (обозначают DMUX, DMS) выполняет функцию, обратную функции выполняемой мультиплексором. Демультимплексор имеет один вход  $D$  данных,  $K$  адресных входов и  $N=2^K$  выходов данных. На выбранном выходе данные совпадают с данными на входе, а на остальных выходах равны 0. Таблица истинности демультимплексора приведена на рис. 37. В качестве демультимплексора можно применить дешифратор с входом разрешения, например, дешифратор D2\_4E, как показано на рис. 37. Вход разрешения дешифратора является входом данных, а адресные входы определяют выходную линию, на которую передается входной сигнал. Поэтому в библиотеках данные элементы обозначают как «дешифратор/демультимплексор».

| Входы |    |    | Выходы |    |    |    |
|-------|----|----|--------|----|----|----|
| D(E)  | A1 | A0 | D3     | D2 | D1 | D0 |
| 0     | 0  | 0  | 0      | 0  | 0  | 0  |
| 1     |    |    |        |    |    | 1  |
| 0     | 0  | 1  | 0      | 0  | 0  | 0  |
| 1     |    |    |        |    | 1  |    |
| 0     | 1  | 0  | 0      | 0  | 0  | 0  |
| 1     |    |    |        | 1  |    |    |
| 0     | 1  | 1  | 0      | 0  | 0  | 0  |
| 1     |    |    | 1      |    |    |    |

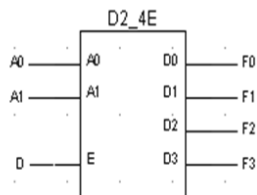


Рис. 37. Таблица истинности и УГО дешифратора-демультимплексора

Для обмена информацией между несколькими источниками применяются шины передачи данных, в которых целесообразно использовать соединение мультиплексоров и демультимплексоров. Такая система с одноразрядной шиной приведена на рис. 38. Мультиплексор выбирает один из  $N$  источников данных для передачи их по шине. Демультимплексор направляет данные к одному из  $M$  адресатов на приёмном конце шины.

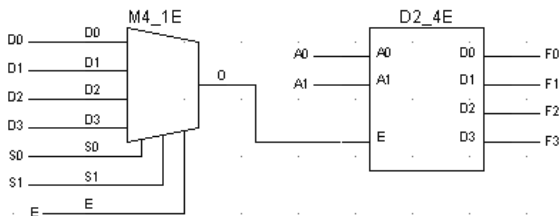


Рис. 38. Организация шины передачи на мультиплексоре и демультиплексоре

Систему передачи на мультиплексоре и демультиплексоре можно использовать для следующих операций:

- коммутации, как отдельных линий, так и многоразрядных шин;

- передачи данных от  $i$ -го источника к  $j$ -му приемнику, при фиксации их значений на соответствующих адресных входах  $S_i$  и  $A_j$  – это передача от 1 к 1;

- преобразования входного параллельного двоичного кода, подаваемого на входы данных мультиплексора, в последовательный код на одном из выходов демультиплексора, при этом фиксируется  $j$ -й адрес на демультиплексоре и последовательно изменяются адреса  $i$  на адресных входах мультиплексора;

- преобразования входного последовательного кода, подаваемого на  $i$ -й адрес мультиплексора, в параллельный на выходах демультиплексора, при этом фиксируется  $i$ -й адрес на мультиплексоре и последовательно изменяются адреса  $j$  на адресных входах демультиплексора.

Важную роль в системах организации шин играет синхронизация передачи и приема данных в источниках и приемниках информации.

#### 1.1.4. Компараторы

**Компаратором** (comparator) называют устройство сравнения, которое определяет отношение между двумя входными словами  $A$  и  $B$ . Компараторы сравнивают два слова и вырабатывают три вы-

ходных сигнала, которые принимают единичные значения (истинны), если соблюдается условие указанное в обозначении функции:

$$F_{A=B}, \text{ если } A=B; F_{A>B}, \text{ если } A>B; F_{A<B}, \text{ если } A<B.$$

Как правило, сравнивается первое слово  $A$  со вторым словом  $B$ , поэтому для краткости записи в обозначении функций указываются только их отношения: «равно» –  $F_{=}$ , «больше» –  $F_{>}$ , «меньше» –  $F_{<}$ . Входные слова могут быть числами как со знаком, так и без знака. В качестве примеров применения компараторов можно указать формирование арифметического соотношения между словами, использование отношений в качестве логических условий в микропрограммах и командах передачи управления, сравнение «идентификатора устройства» с адресом выбираемого устройства в системах с общей шиной, в устройствах автоматики для сигнализации при выходе сигналов за пределы диапазона и т.д.

Компараторы строятся на основе схем поразрядных операций над одноименными разрядами двух слов.

### *Схемы равенства*

Схемы равенства формируют только сигнал отношения «равно», когда входные коды равнозначны. Два слова  $A$  и  $B$  равны, если попарно равны все одноименные разряды слов. На рис. 39 приведены две схемы, реализующие функцию равенства для четырёхразрядных слов на элементах XOR («Исключающее ИЛИ», сумма по модулю два») и XNOR («Схема равнозначности»). Выходную часть схемы компаратора можно значительно упростить, если использовать элементы XOR с открытым коллектором или третьим состоянием, позволяющие реализовать функцию «Монтажное ИЛИ».

Компаратор можно дополнить входом разрешения  $E$  (enable), который используется как для блокирования (выключения) функции сравнения, так и для каскадирования компараторов (увеличение разрядности). На рис. 40 приведена схема последовательного каскадируемого (итерационного) девятиразрядного компаратора. Схема сравнения может быть построена на дешифраторе и мультиплексоре. При этом на их адресные входы подаются сравниваемые коды (рис. 41).

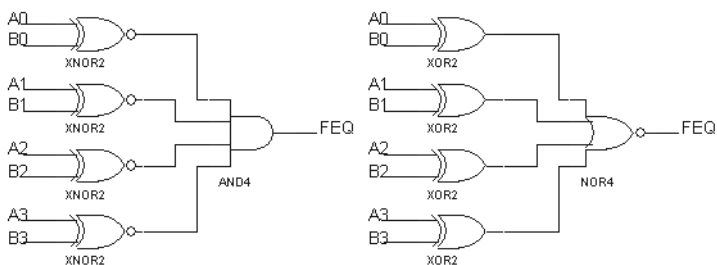


Рис. 39. Схемы равенства четырехразрядных слов

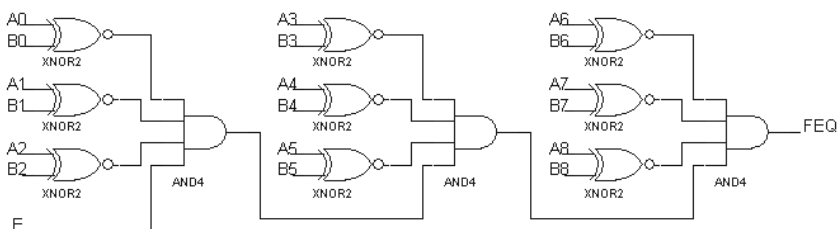


Рис. 40. Схема равенства девятиразрядных слов

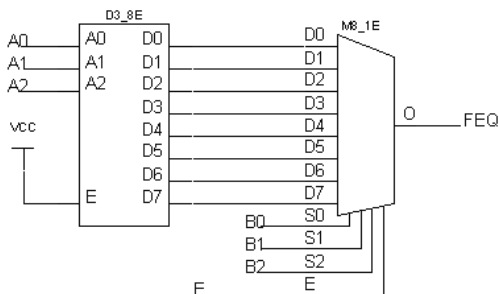


Рис. 41. Схема равенства на основе дешифратора и мультиплексора

## Схемы сравнения

Схемы сравнения формируют выходные сигналы  $F_{=}$ ,  $F_{>}$ ,  $F_{<}$ , которые определяют соотношение между входными словами. Соотношения между числами в позиционных системах счисления, в которых вес любого старшего разряда больше любого меньшего разряда, можно установить на основании последовательного сравнения их одноименных разрядов. Сравнение чисел можно проводить либо со старших, либо с младших разрядов.

Таблица истинности и схема компаратора одноразрядных двоичных чисел приведены на рис. 42.

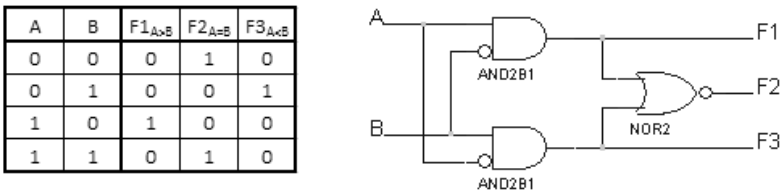


Рис. 42. Схема компаратора одноразрядных двоичных чисел

На основе приведенной ячейки сравнения может быть реализована схема любой разрядности. При анализе отношений со старших разрядов сначала сравнивают два крайних левых разряда. При этом возможны три ситуации: « $A$  больше  $B$ », « $A$  меньше  $B$ » или « $A$  равно  $B$ ». При истинности одного из двух первых отношений, дальнейшее сравнение можно не проводить, так как старший разряд первого числа либо больше, либо меньше второго числа, а значит установлено соотношение чисел. В случае, когда старшие разряды равны, то переходят к проверке следующих более младших разрядов. Если они тоже равны, то переходят к следующим, более младшим разрядам и т.д. Таким образом, дольше всего будут сравниваться числа, если они равны.

Для многоразрядных слов могут быть построены последовательные или параллельные компараторы. В последовательных схемах используется только один одноразрядный компаратор и схема поочередного ввода разрядов и хранения предыдущего соотноше-

ния разрядов. Такое устройство обладает невысоким быстродействием и малыми аппаратными затратами.

В параллельных схемах сокращается время выполнения сравнения чисел, но увеличиваются аппаратные затраты пропорционально числу разрядов. На рис. 43 приведено УГО и схема двухразрядного компаратора.

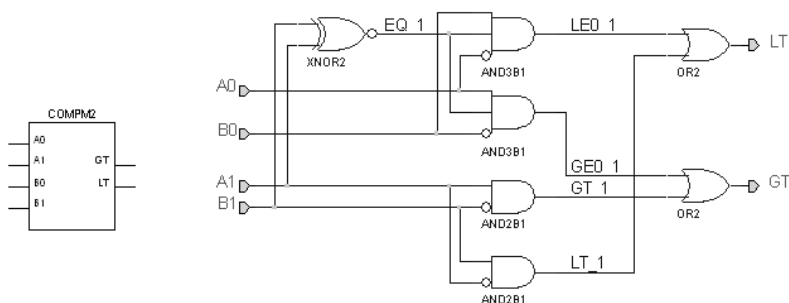


Рис. 43. УГО и схема компаратора двухразрядных двоичных чисел

Для сокращения внутренних аппаратных затрат компараторов определяют только два отношения чисел, например, «больше» и «меньше», а отношение «равно» реализуют на дополнительной схеме в виде функции данных выходов. Для отношения «равно» можно записать

$$F_{A=B} = \neg(F_{A>B} + F_{A<B}).$$

Аналогично можно определить функции для других отношений, таких, как «больше или равно», «меньше или равно», «не равно».

### ***Увеличение разрядности компараторов***

В библиотеки элементов ПЛИС фирмы Xilinx включены параллельные компараторы COMP на 2, 4, 8, 16 разрядов. Для наращивания разрядности компараторов могут вводиться входы отношений, которые подключаются к выходам от предыдущей группы разрядов и используются для каскадного включения компараторов. На рис. 44 приведены УГО и четырёхразрядная каскадируемая схема компаратора. Входы отношений *AGBI*, *ALBI* и *AEBI* могут также использоваться для сравнения разрядов входных кодов.

| Inputs |        |        |        |      |      |      | Outputs |      |      |
|--------|--------|--------|--------|------|------|------|---------|------|------|
| A3, B3 | A2, B2 | A1, B1 | A0, B0 | AGBI | ALBI | AEBI | AGBO    | ALBO | AEBO |
| A3>B3  | X      | X      | X      | X    | X    | X    | 1       | 0    | 0    |
| A3<B3  | X      | X      | X      | X    | X    | X    | 0       | 1    | 0    |
| A3=B3  | A2>B2  | X      | X      | X    | X    | X    | 1       | 0    | 0    |
| A3=B3  | A2<B2  | X      | X      | X    | X    | X    | 0       | 1    | 0    |
| A3=B3  | A2=B2  | A1>B1  | X      | X    | X    | X    | 1       | 0    | 0    |
| A3=B3  | A2=B2  | A1<B1  | X      | X    | X    | X    | 0       | 1    | 0    |
| A3=B3  | A2=B2  | A1=B1  | A0>B0  | X    | X    | X    | 1       | 0    | 0    |
| A3=B3  | A2=B2  | A1=B1  | A0<B0  | X    | X    | X    | 0       | 1    | 0    |
| A3=B3  | A2=B2  | A1=B1  | A0=B0  | 1    | 0    | 0    | 1       | 0    | 0    |
| A3=B3  | A2=B2  | A1=B1  | A0=B0  | 0    | 1    | 0    | 0       | 1    | 0    |
| A3=B3  | A2=B2  | A1=B1  | A0=B0  | 0    | 0    | 1    | 0       | 0    | 1    |
| A3=B3  | A2=B2  | A1=B1  | A0=B0  | 0    | 1    | 1    | 0       | 1    | 1    |
| A3=B3  | A2=B2  | A1=B1  | A0=B0  | 1    | 0    | 1    | 1       | 0    | 1    |
| A3=B3  | A2=B2  | A1=B1  | A0=B0  | 1    | 1    | 1    | 1       | 1    | 1    |
| A3=B3  | A2=B2  | A1=B1  | A0=B0  | 1    | 1    | 0    | 1       | 1    | 0    |
| A3=B3  | A2=B2  | A1=B1  | A0=B0  | 0    | 0    | 0    | 0       | 0    | 0    |

Рис. 44. УГО и схема компаратора четырехразрядных двоичных чисел

Каскадирование схем сравнения при их последовательном включении приведено на рис. 45. Компаратор проводит сравнение 12-ти разрядных двоичных чисел. У младшего каскада на входы отношений «больше» *AGBI* и «меньше» *ALBI* поданы сигналы «0», а на вход «равно» *AEBI* – сигнал «1». Аналогично могут быть построены схемы 4*m*-разрядных чисел с последовательным включением *m* каскадов.

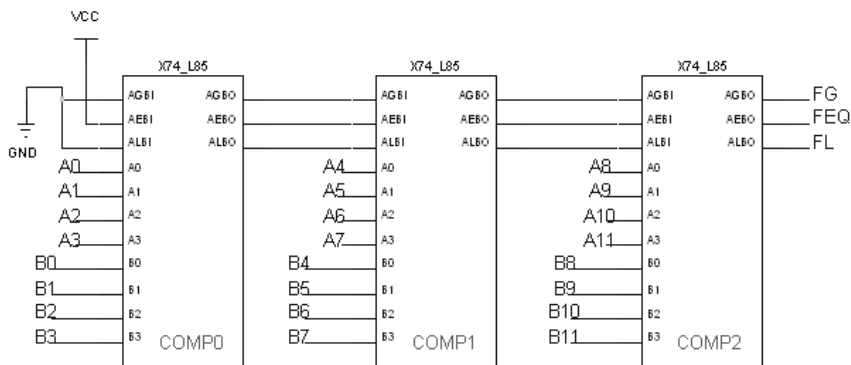


Рис. 45. Схема компаратора с последовательным включением каскадов

Каскадирование схем сравнения можно проводить и при их параллельном включении. На рис. 46 приведена пирамидальная схема сравнения 24-разрядных двоичных чисел при параллельном включении входных каскадов.

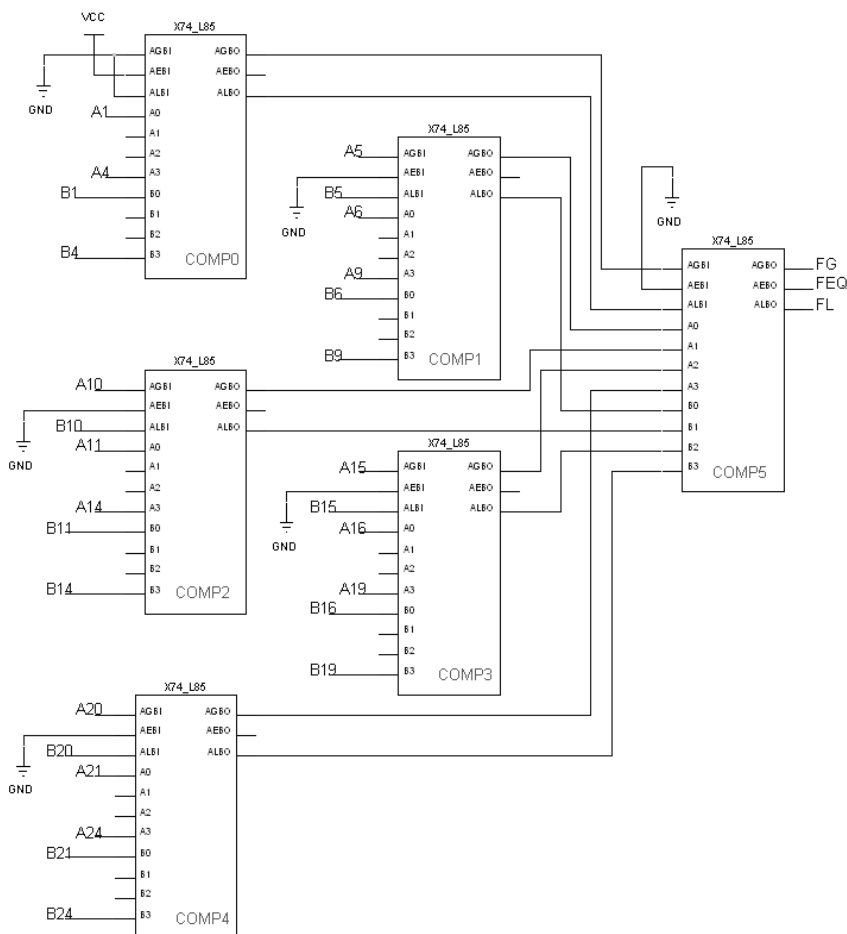


Рис. 46. Схема пирамидального компаратора

На четырех входных каскадах COMP1-COMP4 проводится сравнение пятиразрядных чисел, так как сравниваемые числа подаются также и на входы отношений, а на каскаде COMP0 – срав-

нение четырехразрядных чисел. Входные компараторы первой ступени COMP0-COMP4 преобразуют разрядности групп входных чисел в сигналы отношений в унитарном коде «1 из 3». Компаратор второй ступени COMP5 определяет соотношение между группами и формирует сигнал на одном из выходов *FG*, *FEQ*, *FL*. В данной схеме входные сигналы проходят только через две ступени. Дальнейшее увеличения разрядности можно реализовать в трёхступенчатой схеме сравнения, если входные компараторы заменить на аналогичные 24-разрядные схемы.

### *Компаратор на основе сумматора*

В составе блока операций процессора компаратор может быть построен на основе сумматора. При этом выполняется операция вычитания чисел  $A-B$  и анализируется результат. Для этого на сумматор число  $B$  подаётся в инверсной форме, а на вход переноса  $CI$  подаётся единица (рис. 47).

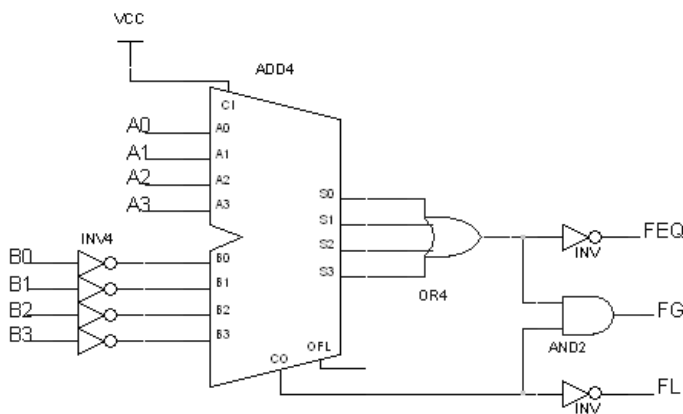


Рис. 47. Схема компаратора на основе сумматора

Рассмотрим примеры выполнения операций. Операция вычитания заменена на прибавление к числу  $A$  обратного кода числа  $B$  (инверсия всех разрядов) и еще плюс «1». Числа рассматриваются как целые положительные коды, без знака. Левый разряд результата показывает значение выходного переноса  $CO$ .

|                                                                                                                                                                                                     |                                                                                                                                                                                                            |                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $  \begin{array}{r}  A=13 \quad + \quad 1101 \\  B=13 \quad + \quad 0010 \text{ (NB)} \\  \hline  + \quad 1 \\  \hline  1. \ 0000 \\  \text{CO}=1 \quad \text{S}=0 \\  \mathbf{A=B}  \end{array}  $ | $  \begin{array}{r}  A=13 \quad + \quad 1101 \\  B=11 \quad + \quad 0100 \text{ (NB)} \\  \hline  + \quad 1 \\  \hline  1. \ 0010 \\  \text{CO}=1 \quad \text{S} \neq 0 \\  \mathbf{A > B}  \end{array}  $ | $  \begin{array}{r}  A=10 \quad + \quad 1010 \\  B=13 \quad + \quad 0010 \text{ (NB)} \\  \hline  + \quad 1 \\  \hline  0. \ 1101 \\  \text{CO}=0 \quad \text{S} \neq 0 \\  \mathbf{A > B}  \end{array}  $ |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Из примеров видно, что равенство суммы нулю будет признаком того, что числа равны  $A=B$ . В этом случае в схеме компаратора единичный сигнал «равно»  $FEQ$  формируется на выходе инвертора. Единица переноса при ненулевой сумме будет признаком того, что  $A$  больше  $B$  (единичный сигнал «больше»  $FG$  формируется на выходе элемента И). Выходной перенос  $CO$  будет равен «0» лишь в том случае, когда  $A$  меньше  $B$  (единичный сигнал «меньше»  $FL$  формируется на выходе второго инвертора).

Аналогично можно вывести значения отношений для сравнения чисел представленных в прямом, дополнительном или обратном коде, когда старший разряд является знаковым.

### 1.1.5. Преобразователи кодов

Преобразователем кода называется функциональный узел компьютера, ставящий соответствие между входным кодом и выходным кодовым словом. С этой точки зрения к данному классу задач можно отнести все комбинационные схемы. Вследствие широкого круга задач, решаемых при преобразованиях, в классе преобразователей кодов выделяют отдельные самостоятельные группы устройств. К таким группам относятся схемы преобразования разнообразных двоичных и двоично-десятичных кодов (прямой, дополнительный, обратный и их модификации), схемы формирования кода Хэмминга и кода Грея, приоритетные и двоичные шифраторы, указатели старшей (младшей) единицы или нуля, клавишные шифраторы, преобразователи 7-сегментного кода.

## *Двоичные шифраторы*

Двоичные шифраторы применяются для формирования адреса активного входа. При возбуждении одного из входов шифратора на его выходе формируется двоичный код номера активной входной линии. Двоичные шифраторы выполняют функцию обратную по отношению к дешифратору – они преобразуют унитарный код «1 из  $N$ » в двоичный код. Полный двоичный шифратор имеет  $2^K$  входов и  $K$  выходов.

На рис. 48 приведена таблица истинности восьмивходового двоичного шифратора.

| I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 | A2 | A1 | A0 |
|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  |
| 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  |
| 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  | 0  |
| 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 1  | 1  |
| 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  |
| 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 0  |
| 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  |

Рис. 48. Таблица истинности восьмивходового двоичного шифратора

Таблица содержит всего восемь строк, хотя количество возможных комбинаций входных сигналов 256, так как на входе не может быть других комбинаций, кроме представленных в унитарном коде. Набор таких функций относится к классу не полностью определенных.

Из анализа таблицы видно, что схема может быть реализована на трех элементах ИЛИ. При этом к входам элементов ИЛИ каждого выходного разряда подключаются те входы шифратора, в двоичном представлении номера которых есть единица в данном разряде. Например, функция выхода  $A_0$  становится равной единице, когда один из нечётных входов  $I_1$ ,  $I_3$ ,  $I_5$  либо  $I_7$  принимают единичные значения, при этом остальные входы равны нулю. Аналогично могут быть определены остальные выходные функции. Вы-

ражения для выходных функций двоичного шифратора в дизъюнктивной нормальной форме будут следующими:

$$A0 = I1+I3+I5+I7; \quad A1 = I2+I3+I6+I7; \quad A2 = I4+I5+I6+I7.$$

Вход  $I0$  не входит в функции, так как единичному сигналу на этом входе соответствует выходной код «все нули». При этом не различается ситуация, когда не подано ни одного входного сигнала. Для фиксации ситуации, когда в группе есть хотя бы одна единица, вводится сигнал  $GS$  (group signal). На рис. 49 приведена схема восьмивходового двоичного шифратора.

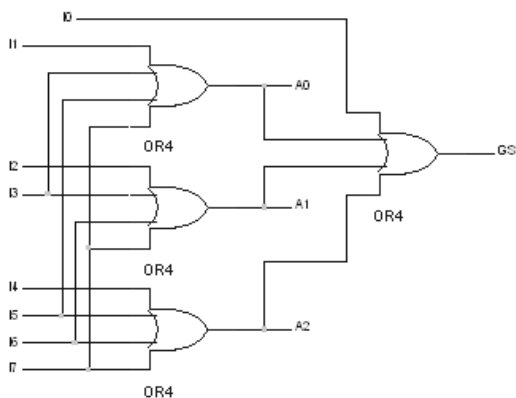


Рис. 49. Схема восьмивходового двоичного шифратора

### ***Приоритетные шифраторы***

При работе устройств компьютера, например, при обмене данными по общей шине, необходимо определение приоритетного претендента на пользование каким-либо ресурсом. При этом несколько конкурентов могут одновременно выставить запросы на обслуживание, и необходимо выбрать того, кому предоставлено право первоочередного обслуживания. Для этого каждому источнику запросов присваивается фиксированный приоритет. Например, пусть высший приоритет имеет запрос со старшим номером, а далее приоритет уменьшается от номера к номеру. Младший приоритет задают у нулевого источника, и он будет обслуживаться только при отсутствии всех других запросов. Если имеется не-

сколько запросов, то обслуживается запрос с наибольшим номером.

Приоритетный шифратор формирует на выходе двоичный адрес (номер) старшего запроса. Так как приоритетные шифраторы осуществляют поиск старшей (левой) единицы во входном слове, то они могут применяться в устройствах нормализации чисел с плавающей запятой. Также приоритетные шифраторы используются для построения шифраторов клавиатуры, контроллеров прерываний.

Логика работы приоритетного шифратора на четыре входа  $I3-I0$  и два выхода  $A1-A0$  приведена в таблице на рис. 50.

| $EI$ | $I3$ | $I2$ | $I1$ | $I0$ | $A1$ | $A0$ | $GS$ | $EO$ |
|------|------|------|------|------|------|------|------|------|
| 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    |
| 1    | 0    | 0    | 0    | 1    | 0    | 0    | 1    | 0    |
| 1    | 0    | 0    | 1    | x    | 0    | 1    | 1    | 0    |
| 1    | 0    | 1    | x    | x    | 1    | 0    | 1    | 0    |
| 1    | 1    | x    | x    | x    | 1    | 1    | 1    | 0    |
| 0    | x    | x    | x    | x    | 0    | 0    | 0    | 0    |

Рис. 50. Таблица истинности четырёхвходового приоритетного шифратора

В таблице введены: вход  $EI$  – сигнал разрешения работы, выход  $EO$  – сигнал отсутствия запросов на входах, используется для разрешения работы следующего (младшего) шифратора при наращивании размерности шифраторов, выход  $GS$  – сигнал наличия запросов на входе данного шифратора.

Сигналы на выходах формируются только при условии  $EI=1$ . При сигнале  $EI=0$  независимо от входных сигналов все выходные сигналы нулевые. Входные переменные в таблице справа от диагонали, образованной цифрами 1, отмечены значением «х» и это означает, что они не должны определять выходной код, так как сигнал с высшим приоритетом блокирует запросы с меньшим приоритетом. На основе таблицы можно получить следующие выражения для функций:

$$A1 = (I3+I2) \cdot EI;$$

$$A0 = (I3+\neg I2 \cdot \neg I1) \cdot EI;$$

$$EO = (I3+\neg I3 \cdot \neg I2 \cdot \neg I1 \cdot \neg I0) \cdot EI;$$

$$GS = I3+I2+I1+I0) \cdot EI.$$

Данные функции определяют внутреннюю структуру приоритетного шифратора.

Из анализа приоритетного шифратора видно, что при наличии только одного запроса он работает так же, как и двоичный.

### ***Увеличение разрядности приоритетного шифратора***

Для увеличения разрядности входных слов используется либо линейное увеличение размерности шифратора, либо каскадирование шифраторов. При линейном увеличении значительно увеличивается объём аппаратных затрат, а при каскадировании увеличивается время задержки. В библиотеку элементов Xilinx включены восьмивходовой каскадируемый приоритетный шифратор X74\_148 и десятивходовой приоритетный шифратор X74\_147 с инверсными входами и выходами.

На рис. 51 приведена схема для приоритетного обслуживания восьмизначного слова при последовательном включении шифраторов (наивысший приоритет имеет вход *I7*). Младший шифратор PR0 включается только в том случае, если нет хотя бы одного запроса на старшем шифраторе PR1 (сигнал *EO*=1). Третий разряд *A2* выходного кода снимается с выхода *GS* шифратора PR1. Итоговый сигнал *GS*=1 формируется только в том случае, если есть хотя бы один запрос.

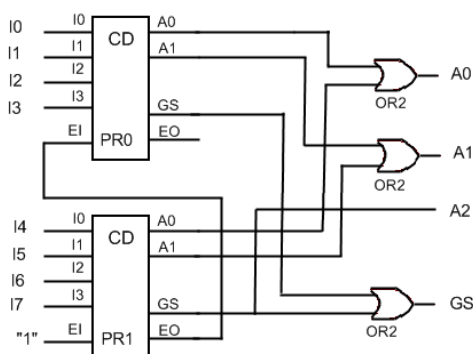


Рис. 51. Схема восьмивходового приоритетного шифратора при последовательном включении каскадов

На рис. 52 приведена схема 16-разрядного приоритетного шифратора при параллельном включении шифраторов (наивысший приоритет имеет вход  $I15$ ). Приоритетный шифратор PR4 второго уровня определяет старшую четвёрку разрядов (тетраду) в которой есть запрос и формирует на выходах  $A3$ ,  $A2$  её номер. Четырёхвходовой двухразрядный мультиплексор MUX выбирает приоритетный запрос в выбранной тетраде и формирует младшие разряды  $A1$ ,  $A0$  выходного кода. Выходной сигнал  $GS$  принимает единичное значение при наличии хотя бы одного запроса.

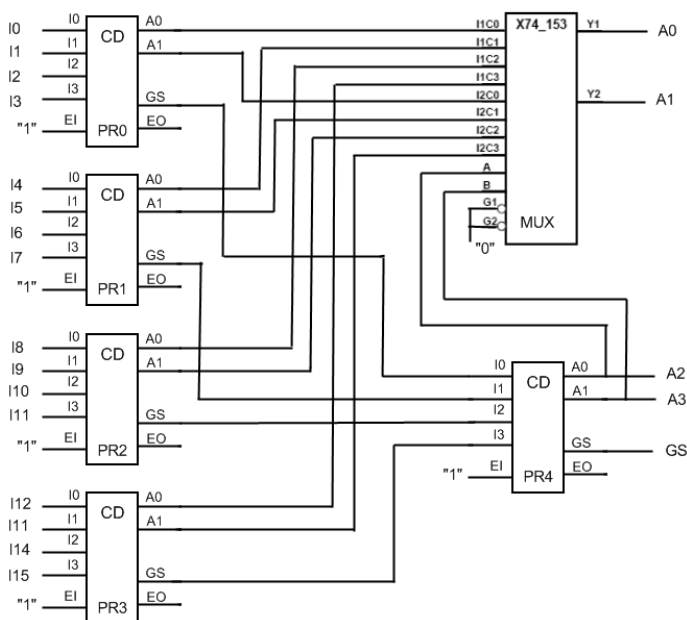


Рис. 52. Схема 16-входового приоритетного шифратора при параллельном включении каскадов

### *Указатели старшей единицы*

Указатели старшей единицы решают ту же задачу, что и приоритетные шифраторы, но вырабатывают результат в виде унитарного кода «1 из  $N$ ». Таким образом, что при наличии на входах не-

скольких запросов на выходе будет активным только один разряд, соответствующий высшему (старшему) приоритету. В схемах указателей число входов равно числу выходов.

Указатели старшей единицы могут быть реализованы подключением двоичного дешифратора к выходам приоритетного шифратора.

Другой вариант построения указателя – реализация схемы путём объединения элементов ИЛИ в цепочку (рис. 53). На выходах элементов ИЛИ входной код преобразуется в вид «00.. 011..1». Далее на выходе только одного элемента запрета AND2B1, на входы которого поданы значения «10», формируется единичное значение, указывающее разряд с высшим приоритетом. На выходах всех остальных элементов запрета AND2B1 формируются нулевые значения.

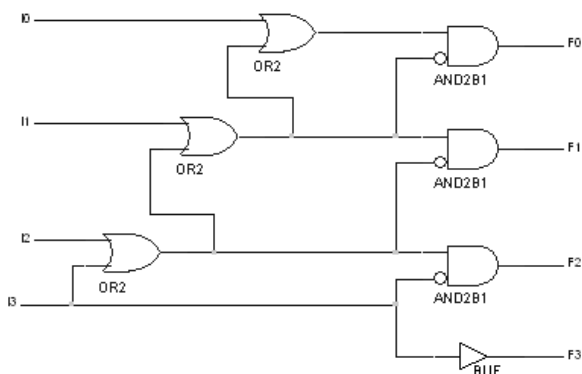


Рис. 53. Схема 4-разрядного указателя старшей единицы

Если к выходам указателя старшей единицы подключить двоичный шифратор, то можно получить схему приоритетного шифратора, который формирует в двоичном коде номер самой старшей единицы.

## 1.2. ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЭЛЕМЕНТЫ И УЗЛЫ

### 1.2.1. Триггеры

**Триггер** — это функциональный элемент последовательно-го типа с двумя устойчивыми состояниями, предназначенный для записи и хранения информации. Триггерами называют такие логические устройства, выходные сигналы которых определяются не только сигналами на входах, но и предысторией их работы, т.е. состоянием в предыдущие моменты времени.

Триггером называют логическую схему с положительной обратной связью, имеющую два устойчивых состояния, которые называют единичным и нулевым состояниями и обозначают как «1» и «0». Под воздействием входных сигналов триггер может переключаться из одного устойчивого состояния в другое. Перевод триггера в единичное состояние называют установкой (set) триггера, а перевод в нулевое состояние называют сбросом или гашением (reset).

Триггер – один из базовых (основных) элементов вычислительной техники. Отличительной особенностью триггера как функционального устройства является его свойство запоминания двоичной информации – триггер хранит один бит информации. Поэтому триггер является элементарной ячейкой памяти. На основе триггеров реализуются регистры и счетчики.

Внутренняя структура триггера содержит запоминающую ячейку (ЗЯ), имеющую прямой и инверсный выходы, и комбинационную схему, которую называют схемой управления (СУ) или входной логики. Запоминающая ячейка реализуется на элементарных триггерных схемах, содержащих два элемента 2х2И-НЕ или два элемента 2х2ИЛИ-НЕ, которые взаимно охвачены обратной связью – включены кольцом. Схема управления преобразует входные логические сигналы в установочные сигналы для ЗЯ. Работа схемы управления может определяться также состоянием входа синхронизации *C*.

#### *Классификация триггерных схем*

По функциональному признаку в зависимости от способов организации логических связей между входами и выходами, триггеры характеризуются числом логических входов и их назначением.

Триггеры делятся на одноходовые, к ним относятся D-, T-триггеры, и двухходовые типа RS-, DV-, JK-и триггеры со сложной логикой. Функционирование триггера может быть задано таблицей переходов, матрицей переходов, графом переходов или описано характеристическими уравнениями.

По способу записи информации триггеры делятся на асинхронные и синхронные. Состояние асинхронного триггера определяется в каждый момент времени только состоянием логических входов. Изменения входных сигналов непосредственно передаются на выход триггера, с некоторой задержкой прохождения входного сигнала, равной сумме задержек на элементах его составляющих. В синхронных триггерах приём входной информации определяется моментом подачи импульса на вход синхронизации  $C$  (clock). Этот вход также обозначают термином «такт». Все триггеры, как правило, имеют асинхронные входы предварительной установки триггера в состояние «0» (вход  $R$ ) и/или состояние «1» (вход  $S$ ).

Синхронные триггеры в зависимости от организации схемы управления разделяют на триггеры со статическим и с динамическим управлением записью по входу синхронизации  $C$ .

В триггерах с динамическим управлением (иное название – «триггер, управляемый фронтом») информация воспринимается только в момент изменения (перепада) сигнала по входу  $C$ . При изменении по фронту из «0» в «1» триггер имеет прямой динамический вход  $C$ , при изменении по спаду из «1» в «0» – инверсный динамический вход  $C$ .

В триггерах со статическим управлением (иное название – «потенциальный триггер») входная информация принимается только тогда, когда значение синхровхода  $C$  соответствует уровню логической единицы (прямой вход) или логического нуля (инверсный вход), т.е. в течение всей длительности сигнала  $C$ . Это означает, что при  $C=1$  (прямой вход) переключение сигналов на логических входах вызывает изменение состояния выхода триггера. Поэтому информацию на логических входах, как правило, изменяют при  $C=0$  (при прямом входе).

Триггеры со статическим управлением, в свою очередь, подразделяются на одноступенчатые и двухступенчатые. Одноступенчатые триггеры (latch, защёлки) состоят из одной ступени и содержат запоминающую ячейку и схему управления. Одноступенчатый триггер на УГО обозначают одной буквой –  $T$ .

Синхронный двухступенчатый триггер – это схема, состоящая из двух частей, т.е. двух последовательно соединенных триггеров со статическим управлением записью, одновременный приём информации в которые запрещён. Это достигается включением инвертора в цепь синхронизирующих импульсов второй ступени. Логика работы следующая. При  $C=1$  информация, в соответствии с логикой работы триггера, записывается в первую ступень (вторая ступень заблокирована для записи). При  $C=0$  вторая ступень принимает (копирует) состояние первой ступени (первая ступень заблокирована для записи). Сигнал на выходах триггера появляется с задержкой, равной задержке срабатывания второй ступени. Следовательно, при  $C=0$  состояния обеих ступеней двухступенчатого триггера одинаково. Двухступенчатый триггер на УГО обозначают двумя буквами – *ТТ*.

### ***Таблицы переходов и обозначения триггеров***

На рис. 54 приведены таблицы переходов и характеристические уравнения, описывающие поведение триггеров.

| <b>D-триггер</b>                                                                                                      |        | <b>RS-триггер</b>                       |   |      | <b>NR NS - триггер</b>                  |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
|-----------------------------------------------------------------------------------------------------------------------|--------|-----------------------------------------|---|------|-----------------------------------------|-------|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|--------|---|---|------|---|---|------|---|---|---|---|---|---|--|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|--------|---|---|------|---|---|---|---|---|---|---|---|-------|--|--|
| <table><tr><th>D(t)</th><th>Q(t+1)</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>        | D(t)   | Q(t+1)                                  | 0 | 0    | 1                                       | 1     |  | <table><tr><th>R(t)</th><th>S(t)</th><th>Q(t+1)</th></tr><tr><td>0</td><td>0</td><td>Q(t)</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>X</td></tr></table>    | R(t) | S(t) | Q(t+1) | 0 | 0 | Q(t) | 0 | 1 | 1    | 1 | 0 | 0 | 1 | 1 | X |  |  | <table><tr><th>NR(t)</th><th>NS(t)</th><th>Q(t+1)</th></tr><tr><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>Q(t)</td></tr></table>   | NR(t) | NS(t) | Q(t+1) | 0 | 0 | X    | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | Q(t)  |  |  |
| D(t)                                                                                                                  | Q(t+1) |                                         |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 0      |                                         |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 1      |                                         |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| R(t)                                                                                                                  | S(t)   | Q(t+1)                                  |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 0      | Q(t)                                    |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 1      | 1                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 0      | 0                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 1      | X                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| NR(t)                                                                                                                 | NS(t)  | Q(t+1)                                  |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 0      | X                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 1      | 0                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 0      | 1                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 1      | Q(t)                                    |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| Q(t+1)=D(t)                                                                                                           |        | Q(t+1)=S(t) ∨ NR(t)·Q(t)<br>S(t)·R(t)=0 |   |      | Q(t+1)=S(t) ∨ NR(t)·Q(t)<br>S(t)·R(t)=0 |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| <b>T-триггер</b>                                                                                                      |        | <b>DV-триггер</b>                       |   |      | <b>JK-триггер</b>                       |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| <table><tr><th>D(t)</th><th>Q(t+1)</th></tr><tr><td>0</td><td>Q(t)</td></tr><tr><td>1</td><td>¬Q(t)</td></tr></table> | D(t)   | Q(t+1)                                  | 0 | Q(t) | 1                                       | ¬Q(t) |  | <table><tr><th>V(t)</th><th>D(t)</th><th>Q(t+1)</th></tr><tr><td>0</td><td>0</td><td>Q(t)</td></tr><tr><td>0</td><td>1</td><td>Q(t)</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> | V(t) | D(t) | Q(t+1) | 0 | 0 | Q(t) | 0 | 1 | Q(t) | 1 | 0 | 0 | 1 | 1 | 1 |  |  | <table><tr><th>J(t)</th><th>K(t)</th><th>Q(t+1)</th></tr><tr><td>0</td><td>0</td><td>Q(t)</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>¬Q(t)</td></tr></table> | J(t)  | K(t)  | Q(t+1) | 0 | 0 | Q(t) | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | ¬Q(t) |  |  |
| D(t)                                                                                                                  | Q(t+1) |                                         |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | Q(t)   |                                         |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | ¬Q(t)  |                                         |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| V(t)                                                                                                                  | D(t)   | Q(t+1)                                  |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 0      | Q(t)                                    |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 1      | Q(t)                                    |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 0      | 0                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 1      | 1                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| J(t)                                                                                                                  | K(t)   | Q(t+1)                                  |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 0      | Q(t)                                    |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 0                                                                                                                     | 1      | 0                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 0      | 1                                       |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| 1                                                                                                                     | 1      | ¬Q(t)                                   |   |      |                                         |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |
| Q(t+1)=T(t)·¬Q(t) ∨ ¬T(t)·Q(t)                                                                                        |        | Q(t+1)=D(t)·V(t) ∨ ¬V(t)·Q(t)           |   |      | Q(t+1)=J(t)·¬Q(t) ∨ ¬K(t)·Q(t)          |       |  |                                                                                                                                                                                                                                     |      |      |        |   |   |      |   |   |      |   |   |   |   |   |   |  |  |                                                                                                                                                                                                                                      |       |       |        |   |   |      |   |   |   |   |   |   |   |   |       |  |  |

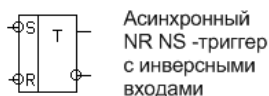
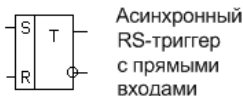
Рис. 54. Таблицы переходов и характеристические уравнения триггеров

Условные графические обозначения триггеров содержат информацию как о функции, так и об основных свойствах и организации

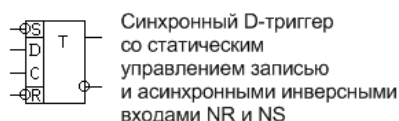
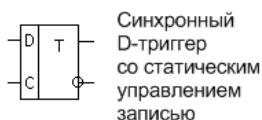
триггера. Знание данных отличий облегчает чтение и понимание работы логических схем цифровых устройств.

На рис. 55 приведены условные графические обозначения триггерных схем.

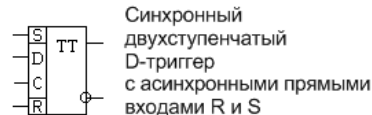
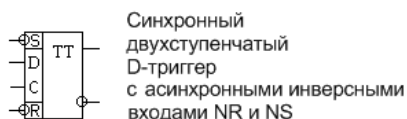
#### АСИНХРОННЫЕ ТРИГГЕРЫ



#### СИНХРОННЫЕ ТРИГГЕРЫ СО СТАТИЧЕСКИМ УПРАВЛЕНИЕМ ЗАПИСЬЮ



#### СИНХРОННЫЕ ДВУХСТУПЕНЧАТЫЕ ТРИГГЕРЫ



#### СИНХРОННЫЕ ТРИГГЕРЫ С ДИНАМИЧЕСКИМ УПРАВЛЕНИЕМ ЗАПИСЬЮ

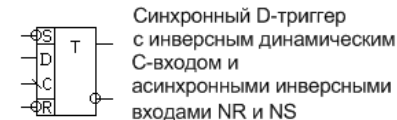
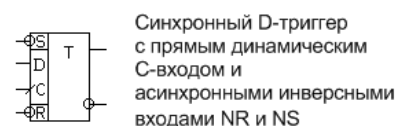


Рис. 55. Условные графические обозначения триггеров

Условные графические обозначения (УГО) различных типов триггеров имеют следующие отличия:

- в УГО синхронных триггеров со статическим управлением записью синхровход C обозначается прямым, что соответствует активному уровню логической единицы, или обозначается как инверсный вход, что соответствует уровню логического нуля;

- в УГО синхронных одноступенчатых триггеров в основном поле стоит одна буква  $T$ ;
- в УГО синхронных двуступенчатых триггеров в основном поле стоят две буквы  $T$  ( $TT$ );
- в УГО синхронных триггеров с динамическим управлением записью синхровход  $C$  отмечен наклонной чертой обозначающей активный фронт. Символ «/» означает активный фронт 0/1, что соответствует прямому динамическому входу, символ «\» означает активный фронт 1/0, что соответствует инверсному динамическому входу;
- асинхронные входы предварительной установки триггера в состояние нуля ( $R$ -вход) и состояние единицы ( $S$ -вход), как правило, располагают в крайних позициях (верхней и нижней) левого поля и отделяют от остальных входов горизонтальной чертой. Входы  $R$  и  $S$  могут быть как прямыми, так и инверсными.

Обозначения триггеров, включенные в библиотеку элементов схемного редактора Schematic Editor САПР Xilinx Foundation, приведены в приложении. Пример из библиотеки Xilinx триггера FTRSLE со сложной логикой приведен на рис. 56.

Данный триггер FTRSLE содержит:

- два логических входа  $D$  и  $T$ ,
- прямой динамический вход  $C$  (фронт 0/1),
- вход разрешения синхроимпульсов  $CE$  (clockenable),
- синхронные входы предварительной установки в «0» ( $R$  – reset) и «1» ( $S$  – set),
- вход разрешения записи  $L$  (load).

Логика работы задается таблицей переходов. Входы синхронной установки  $R$  и  $S$  имеют высший приоритет (первые две строки в таблице), установка проводится по синхровходу  $C$  в момент фронта 0/1. Следующий приоритет у входа загрузки  $L$  – при значении  $L=1$ , независимо значения от входа  $CE$ , проводится запись в триггер значения с входа  $D$ . При значении входа загрузки  $L=0$  и входа разрешения  $CE=0$  триггер хранит значение даже при наличии синхроимпульсов  $C$ . При сигнале  $CE=1$  триггер работает в соответствии с таблицей переходов  $T$ -триггера (две нижние строки в таблице). В последней строке таблицы отмечено переключение состояния триггера (toggle) при значении входов  $CE=1$  и  $T=1$ .

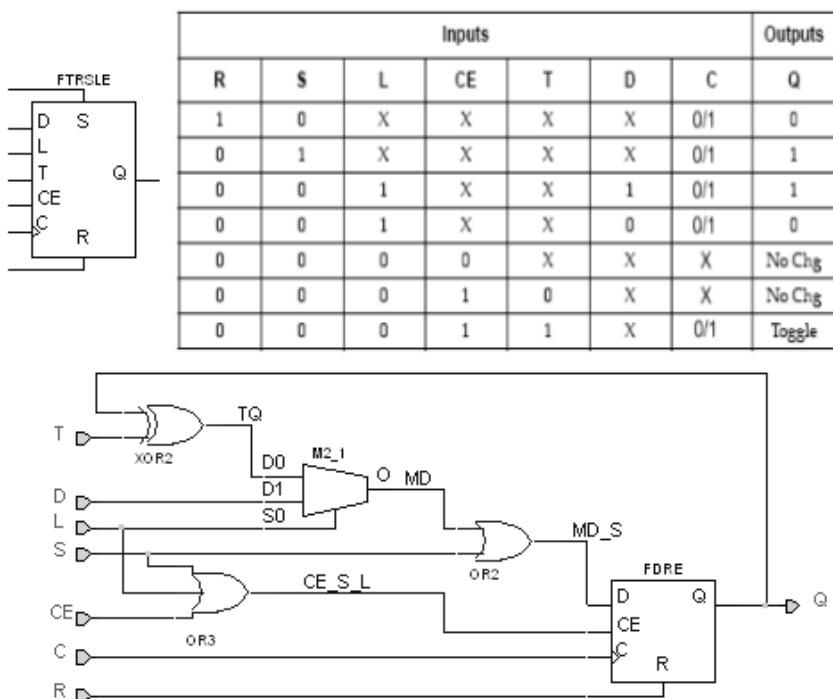


Рис. 56. УГО, таблица переходов и внутренняя структура триггера FTRSLE

Внутренняя структура триггера FTRSLE реализована на основе триггера FDRE – D-триггер с входом разрешения синхросигнала *CE* и входом *R* предварительной установки в ноль. На мультиплексоре *M2\_1* проводится выборка режима *T*- или *D*-триггера под управлением сигнала разрешения загрузки *L*. Функция *T*-триггера реализуется на элементе *XOR2*, который работает в режиме управляемого инвертора для выхода триггера *Q*, в зависимости от значения сигнала на входе *T*. Элемент *OR2* объединяет выход мультиплексора и асинхронный вход *S* предварительной установки в «1» (при сигнале *S*=1 триггер по фронту 0/1 установится в единицу независимо от состояния триггера и выхода мультиплексора). Элемент *OR3* объединяет входы разрешения записи *L* (Load), разрешения синхроимпульсов *CE* (Clock Enable), синхронный вход *S* (Set) предварительной установки в «1».

## *Динамические параметры триггеров*

Основными динамическими параметрами триггерных схем являются:

- задержки переключения выходов триггера по синхронизирующему и установочным входам,
- время предварительной установки сигналов на логических и управляющих входах,
- время удержания сигналов на логических и управляющих входах,
- минимальные длительности синхронизирующих и установочных импульсов.

### *Проектирование триггера с заданной таблицей перехода*

В библиотеке элементов Xilinx базовым является синхронный D-триггер с динамическим управлением записью. На его основе созданы макроэлементы Т-триггера, К-триггера, D-триггера с дополнительными входами.

На основе набора библиотечных триггеров могут быть разработаны триггеры с требуемой таблицей перехода.

***Задача.*** Разработать синхронный E1E2-триггер с асинхронным входом установки в ноль на основе JK-триггера и D-триггера с входом разрешения CE. Таблица переходов E1E2-триггера приведена на рис. 57.

| E1 | E2 | Q(t+1) |
|----|----|--------|
| 0  | 0  | 0      |
| 0  | 1  | NQ     |
| 1  | 0  | 1      |
| 1  | 1  | 1      |

Рис. 57. Таблица переходов E1E2-триггера

Задача синтеза синхронного триггера заключается в составлении функций возбуждения для входов базового триггера и минимизация найденных функций в заданном базисе. Для D-триггера с входом разрешения CE будем считать, что вход CE является входом V для DV-триггера.

Составим таблицу истинности для функций возбуждения (рис. 58).

| N | E1 | E2 | Q(t) | Q(t+1) | Переход | J | K | D  | V      |
|---|----|----|------|--------|---------|---|---|----|--------|
| 0 | 0  | 0  | 0    | 0      | 0–0     | 0 | x | a0 | na0·b0 |
| 1 | 0  | 0  | 1    |        | 1–0     | x | 1 | 0  | 1      |
| 2 | 0  | 1  | 0    | NQ     | 0–1     | 1 | x | 1  | 1      |
| 3 | 0  | 1  | 1    |        | 1–0     | x | 1 | 0  | 1      |
| 4 | 1  | 0  | 0    | 1      | 0–1     | 1 | x | 1  | 1      |
| 5 | 1  | 0  | 1    |        | 1–1     | x | 0 | a5 | a5·b5  |
| 6 | 1  | 1  | 0    | 1      | 0–1     | 1 | x | 1  | 1      |
| 7 | 1  | 1  | 1    |        | 1–1     | x | 0 | a7 | a7·b7  |

Рис. 58. Таблица истинности для функций возбуждения

Аргументами для функций являются входы  $E1$ ,  $E2$  и выход триггера  $Q(t)$  в текущий момент времени. В колонке  $Q(t+1)$  в соответствии с заданием заносим значение триггера в следующий момент времени. В следующей колонке «Переход» отмечаем переходы триггера из состояния в текущий момент времени  $Q(t)$  в состояние в будущий момент времени  $Q(t+1)$  (переход  $Q(t) \rightarrow Q(t+1)$ ).

Для заполнения значений функций будем использовать матрицы переходов триггеров приведенные на рис. 59.

**Матрица переходов JK-триггера**

| Переход<br>Q(t) – Q(t+1) | J | K |
|--------------------------|---|---|
| 0 – 0                    | 0 | x |
| 0 – 1                    | 1 | x |
| 1 – 0                    | x | 1 |
| 1 – 1                    | x | 0 |

**Матрица переходов DV-триггера**

| Переход<br>Q(t) – Q(t+1) | D  | V      |
|--------------------------|----|--------|
| 0 – 0                    | a1 | na1·b1 |
| 0 – 1                    | 1  | 1      |
| 1 – 0                    | 0  | 1      |
| 1 – 1                    | a2 | a2·b2  |

Рис. 59. Матрицы переходов JK- и DV-триггеров

Матрица переходов задает закон функционирования триггера и составляется по таблице переходов триггера.

В матрице переходов JK-триггера символом  $X$  отмечено, что на вход может быть подан сигнал как «1», так и «0». В первой строке таблицы при переходе «0-0» заданы значения  $J=0$  и  $K=X$ , что соответствует двум парам сигналов:  $J=0$  и  $K=0$  или  $J=0$  и  $K=1$  из таблицы переходов JK-триггера. Аналогично заполнены остальные строки матрицы переходов JK-триггера.

В матрице переходов DV-триггера введены коэффициенты  $a_i$  и  $b_i$ , которые могут принимать значения как 1, так и 0. В первой строке при переходе «0-0» на входы  $D$  и  $V$  могут быть поданы следующие сигналы: при  $a_1=1$  и  $b_1=x$  (любое как «0», так и «1») значения входов  $D=1$  и  $V=0$  (так как  $a_1 \cdot b_1=0$ ); при  $a_1=0$  и  $b_1=0$  значения входов  $D=0$  и  $V=0$ ; при  $a_1=0$  и  $b_1=1$  значения входов  $D=0$  и  $V=1$ . Таким образом, переходу «0-0» соответствуют три строки из таблицы переходов DV-триггера. Аналогично четвёртой строке при переходе «1-1» также соответствуют три строки таблицы переходов – на входы  $D$  и  $V$  могут быть поданы следующие пары сигналов: при  $a_2=0$  и  $b_2=x$  значения входов  $D=0$  и  $V=0$ ; при  $a_2=1$  и  $b_2=0$  значения входов  $D=1$  и  $V=0$ ; при  $a_2=1$  и  $b_2=1$  значения входов  $D=1$  и  $V=1$ .

На основании матриц переходов в таблице истинности заполняют значения функций возбуждения в соответствии с типом перехода  $Q(t) \rightarrow Q(t+1)$ . Для облегчения занесения функций на диаграмму Вейча целесообразно у коэффициентов  $a$  и  $b$  ставить индексы, соответствующие номеру набора в таблице истинности для функций возбуждения, например, в нижней строке записываем  $a_7$  и  $a_7 \cdot b_7$ .

Минимизацию целесообразно провести с помощью диаграмм Вейча. Перед заполнением диаграмм Вейча составятся эталонная диаграмма Вейча, в которой все поля обозначаются десятичными цифрами, соответствующими записи двоичного набора конститuenty единицы. Эталонная диаграмма Вейча, диаграммы функций возбуждения и полученные минимальные выражения в дизъюнктивной нормальной форме приведены на рис. 60.

Логические схемы E1E2-триггеров и временная диаграмма работы приведены на рис. 61. Схемы реализованы на триггерах FJKC и FDCE с входами асинхронной установки в «0». Вход разрешения  $CE$  у FDCE-триггера соответствует  $V$ -входу для DV-триггера.

На временной диаграмме входные сигналы  $E1$  и  $E2$  и синхросигнал  $S$  заданы так, чтобы проверить все переходы триггеров – для каждой пары входных сигналов  $E1$  и  $E2$  задаются состояния триггеров как «0», так и «1». Для этого после пятого синхросигнала  $S$  подается сигнал сброса  $CLR=1$ , чтобы установить триггеры в «0» при  $E1=1$  и  $E2=1$ ; остальные переходы проверяется без дополнительной установки триггеров. Выполняемые переходы отмечены на временной диаграмме.

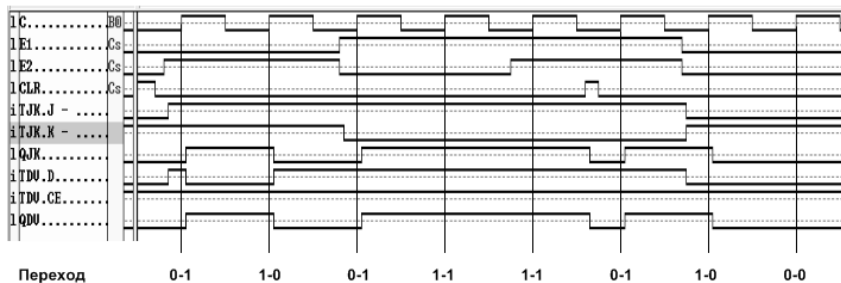
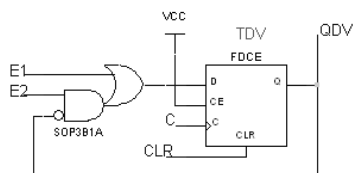
### Эталонная диаграмма Вейча

$$J = E1 + E2$$

K = NE1

$$D = E1 + E2 \cdot NQ$$
$$V = 1$$

The diagram shows a logic circuit for a JK flip-flop. It consists of an OR2 gate with inputs E1 and E2. The output of the OR2 gate is connected to the J input of a JKC flip-flop block. The output of the OR2 gate is also connected to an INV (inverter) gate, whose output is connected to the K input of the JKC flip-flop. The JKC flip-flop block has inputs J, K, and CLR, and an output Q. The output Q is connected to the QJK output. The JKC flip-flop block is labeled with TJK and JKC. The CLR input is connected to a constant low signal (indicated by a line with a bubble). The output QJK is connected to the output of the JKC flip-flop block.



66

### *Реализация триггера с мультиплексором на входе*

Триггер с заданной таблицей переходов можно достаточно легко реализовать на D-триггере и мультиплексоре. Разработаем E1E2-триггер с таблицей переходов, приведённой на рис. 57. Реализация схемы на синхронном динамическом триггере FDC с асинхронным входом установки в «0» и мультиплексоре M4-1E на входе приведена на рис. 62.

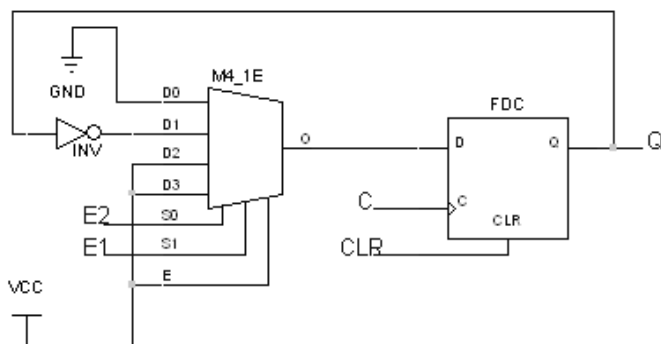


Рис. 62. Схема триггера с мультиплексором на входе

Логические входы  $E1$  и  $E2$  подсоединены к адресным входам  $S1$  и  $S0$  мультиплексора. На информационные входы  $D0-D3$  подаются сигналы в соответствии с таблицей переходов.

### *Делители частоты на триггерах*

Триггеры широко применяются для хранения битов данных и для запоминания результата воздействий управляющих сигналов. Кроме того, триггеры можно использовать в схемах делителей частоты следования цифровых сигналов. Если триггеры, работающие в счётном режиме, соединить последовательно в цепочку, то период следования импульсов на выходе первого младшего триггера  $T0$  будет в два раза больше периода входного сигнала  $C$ , у сигнала на выходе следующего триггера  $T1$  в два раза больше, чем у сигнала с выхода триггера  $T0$  или в четыре раза больше входного сигнала и

т.д. Частота сигналов на выходах каждого триггера будет вдвое меньше, чем на предыдущем разряде, т.е. частота входного сигнала последовательно делится в два раза, затем в четыре раза, восемь раз и т.д. При этом на выходах всех триггеров будут формироваться симметричные последовательности импульсов (меандры) независимо от скважности входных сигналов.

Достаточно часто необходимы счётчики с дробным, недвоичным коэффициентом деления. На рис. 63 приведена схема делителя частоты на три и временная диаграмма работы.

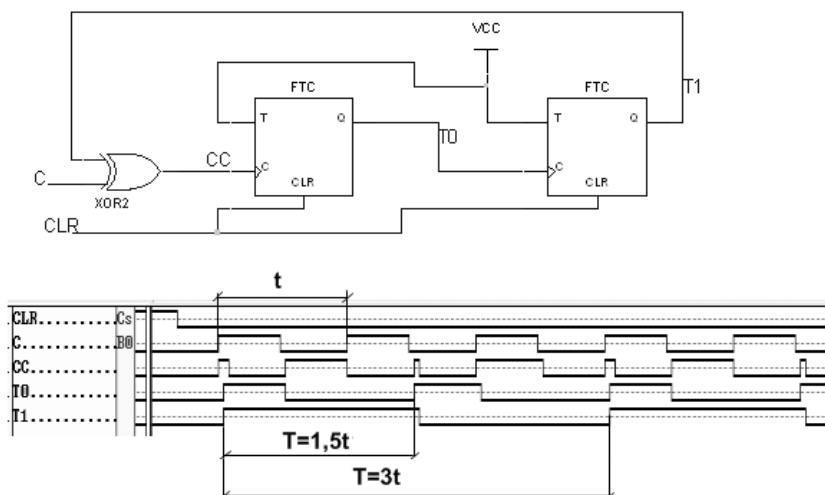


Рис. 63. Схемы делителя частоты на три и временная диаграмма работы

В делителе частоты на три в цепи обратной связи установлен сумматор по модулю два (элемент «Исключающее ИЛИ», XOR2). В исходном состоянии оба триггера установлены в «0». Первый импульс синхронизации  $C$  проходит по цепи  $CC$  на вход первого триггера  $T0$  и вызывает его срабатывание по фронту 0/1 через время задержки. Далее по фронту 0/1 сигнала с выхода триггера  $T0$  срабатывает триггер  $T1$ , вызывая его переход в единичное состояние. При этом сумматор по модулю два переходит в режим инверсии и на его выходе формируется короткий сигнал с длительностью, определяемой задержками срабатывания двух триггеров и элемента XOR2 (см. рис. 63).

Следующие два переключения первого триггера Т0 срабатывают уже по фронту 1/0 входного синхросигнала *C*. Но по фронту 1/0 второго синхроимпульса также переключается второй триггер Т1 в нулевое состояние. При этом следующие импульсы через элемент XOR2 проходят без инверсии и триггер Т0 переключается по фронтам 0/1 синхроимпульса *C*. Далее алгоритм работы повторяется.

Из временной диаграммы видно, что период следования сигналов на выходе первого триггера Т0 равен  $1,5 t$ , а на выходе второго триггера Т1 равен  $3 t$ . Таким образом, коэффициент деления частоты для сигнала с выхода младшего разряда Т0 будет равен 1,5, т.е. дробный коэффициент деления, а для сигнала с выхода триггера Т1 коэффициент будет равен 3. Причем сигнал с выхода триггера Т1 имеет симметричную форму (меандр) и скважность сигнала равную двум  $S=2$  (скважность  $S$  это отношение периода  $T$  следования импульсов к длительности импульса  $t - S=T/t$ ). Для сигнала с выхода первого триггера Т0 скважность равна трем  $S=3$ .

### *Триггеры на мультиплексорах*

Как было показано выше (см. раздел 1.1.3), мультиплексор является многофункциональным узлом, на котором могут быть выполнены специализированные функции, имеющие в ряде случаев схемные преимущества перед специализированными схемами.

На рис. 64 приведена реализация триггеров со статической записью на мультиплексорах М2\_1 и временные диаграммы работы.

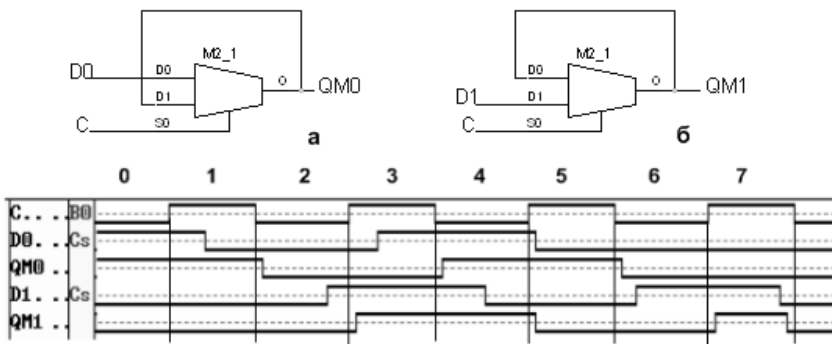


Рис. 64. Схемы триггера на мультиплексоре и временная диаграмма работы

На рис. 64,а показан D-триггер управляемый активным нулем на синхровходе  $C$ . Из временной диаграммы видно, что в нулевой интервал времени при  $C=0$  выход  $QM0=1$  и это значение сохраняется на 1-м интервале при  $C=1$  и входном сигнале  $D0=0$ . Переход выхода  $QM0$  на нулевое значение  $QM0=0$  срабатывает на 2-м интервале при  $C=0$  и сохраняется при  $C=1$  на 3-ем интервале и т.д.

На рис. 64,б приведен D-триггер, управляемый активной единицей на синхровходе  $C$ . Из временной диаграммы видно, что на 3-м интервале времени при  $C=1$  выход  $QM1=1$  и это значение сохраняется на 4-м интервале при  $C=0$  и входном сигнале  $D1=0$ . Переход выхода  $QM1$  на нулевое значение  $QM1=0$  срабатывает на 5-м интервале при  $C=1$  и сохраняется при  $C=0$  на 6-м интервале и т.д. На 7-м интервале времени при  $C=1$  выход  $QM1$  сначала переключается в состояние «1», а затем переходит в нулевое состояние, отрабатывая изменение входного сигнала  $D1=1 \rightarrow 0$ .

### 1.2.2. Счётчики

**Счётчик** – это функциональное устройство последовательностного типа, предназначенное для подсчета числа входных сигналов и запоминания кода этого числа. Число разрешённых устойчивых различных состояний счётчика называют его периодом или модулем счета  $M$ . Диаграмма состояний (граф переходов) счётчика представляет собой кольцо (рис. 65). По состоянию выходов можно определить количество входных воздействий, поступивших на его вход к моменту наблюдения. Входные сигналы поступают на специальный, так называемый счётный вход. Входные сигналы могут поступать как периодически, так и произвольно распределёнными во времени.



Рис. 65. Общий вид диаграммы состояний счётчика

Различные схемы счётчиков могут перебирать свои состояния в самом различном порядке. Наиболее распространенными являются

$n$ -разрядные двоичные счётчики с модулем счёта равным степени два. Такой счётчик состоит из  $n$  триггеров и у него  $M=2^n$  состояний, через которые он проходит в последовательности 0, 1, 2, ...,  $2^n-1$ , 0, 1, ..., он называется счётчиком с двоичной (естественной) системой счисления. Кодом каждого состояния служит соответствующее  $n$ -разрядное двоичное число. Когда количество входных импульсов превысит модуль счёта, то происходит переполнение счётчика, он возвращается в начальное состояние и начинается новый процесс смены состояний. Поэтому модуль также называют коэффициентом пересчёта, так как он определяет цикл работы счётчика, после которого его состояние повторяется.

Список микроопераций счётчика может включать предварительную установку в начальное состояние и/или в нулевое состояние; инкремент (увеличение «+») или декремент (уменьшение «-») хранимого кода на единицу или заданную константу; разрешение выдачи слов параллельным кодом и др.

Разряды счётчиков могут строиться на двухступенчатых триггерах или триггерах с динамической записью. Как правило, триггеры работают в счётном режиме – являются счётчиком по модулю два, т.е. таблица переходов соответствует Т-триггеру при значении  $T=1$ .

### ***Классификация счётчиков***

Счётчики классифицируют по следующим признакам:

- способу кодирования – позиционные и непозиционные;
- модулю счёта – двоичные, двоично-десятичные (декада), с произвольным постоянным или с переменным (программируемым) модулем счёта и прочие (в коде Грея, счётчик Джонсона, в коде «1 из  $N$ » (единичного кодирования) и др.);
- направлению счёта – суммирующие, вычитающие и реверсивные;
- способу организации межразрядных связей – с последовательным, сквозным, параллельным, групповым, кольцевым и комбинированным переносами;
- способу переключения триггера – синхронные и асинхронные.

Возможные режимы работы счётчика – регистрация числа сигналов, поступивших на счётчик, деление частоты.

Важными динамическими параметрами счётчика являются быстроедействие, обычно оцениваемое по максимальной частоте следования входных сигналов, при которой работа счётчика не нарушается, время установки кодов и время формирования сигнала переноса (заёма).

Функционирование счётчика может быть представлено в виде таблицы состояний или графа переходов, последовательности состояний. Работу счётчика можно также описать, используя временные диаграммы для разрядов счётчика, из которых следует порядок состояний счётчика и его модуль  $M$ . Закон функционирования разрядов счётчика описывается функциями возбуждения.

### *Счётчик с произвольным кодированием*

На рис. 66 приведена логическая схема синхронного трехразрядного счётчика с произвольным кодированием. Схема построена на динамических JK-триггерах с асинхронной установкой в «0» и «1» ( $Q_2$  – старший разряд).

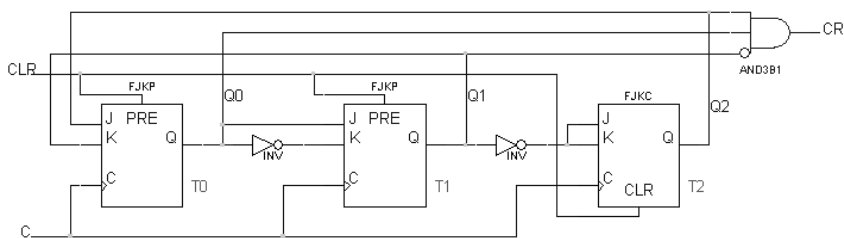


Рис. 66. Схема счётчика с произвольным кодированием

На основе анализа работы счётчика всегда можно определить закон его функционирования. За начальное состояние счётчика принято состояние 3 ( $Q_2=0, Q_1=1, Q_0=1$ ), в которое счётчик устанавливается сигналом  $CLR$ . Для описания работы триггеров можно записать функции возбуждения:

$$\begin{aligned} J_2 &= \neg Q_1, \quad K_2 = \neg Q_1, \\ J_1 &= Q_0, \quad K_1 = \neg Q_0, \\ J_0 &= Q_2, \quad K_0 = Q_1. \end{aligned}$$

Триггеры срабатывают по фронту 0/1 синхросигналов *C*. На рис. 67 приведена временная диаграмма работы счётчика и последовательность состояний счётчика 3, 2, 0, 4, 1, 7, 6, 5. В последнем состоянии последовательности равном 5 формируется **сигнал переноса CR** (carry-out):

$$CR = Q2 \cdot \neg Q1 \cdot Q0.$$

Сигнал переноса (переполнения счётчика) используется для управления следующим счётчиком при увеличении числа разрядов счетчика.

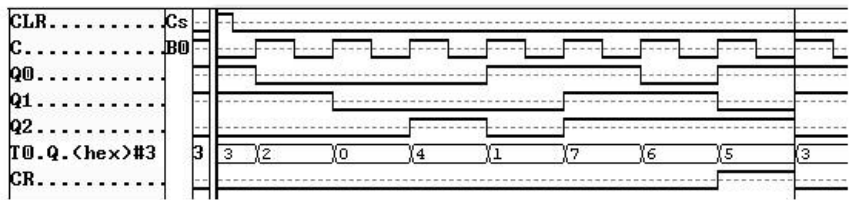


Рис. 67. Временная диаграмма счётчика с произвольным кодированием

По временной диаграмме можно построить граф переходов и составить таблицу переходов (рис. 68). Из анализа работы счётчика легко видеть, что модуль счётчика равен 8.

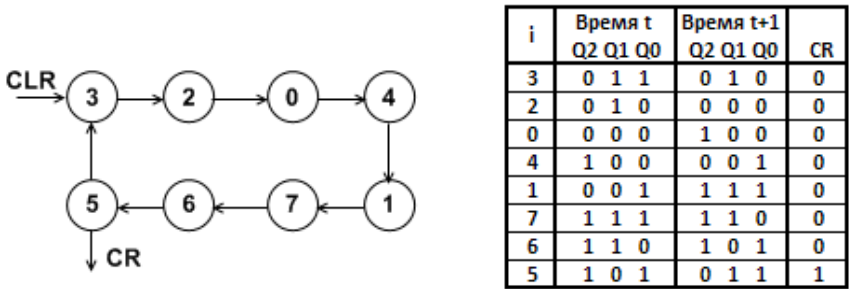


Рис. 68. Граф переходов и таблица состояний счётчика с произвольным кодированием

Достоинством данной схемы является простота функций возбуждения (для их реализации не требуется логических элементов). Недостатком – сложность дешифрации состояний для определения числа пришедших импульсов.

## *Синхронный четырехразрядный двоичный счётчик*

Обозначения счётчиков, включенные в библиотеку элементов схемного редактора Schematic Editor САПР Xilinx Foundation, приведены в приложении. Пример из библиотеки Xilinx синхронного двоичного четырехразрядного счётчика CB4CE приведен на рис. 69.

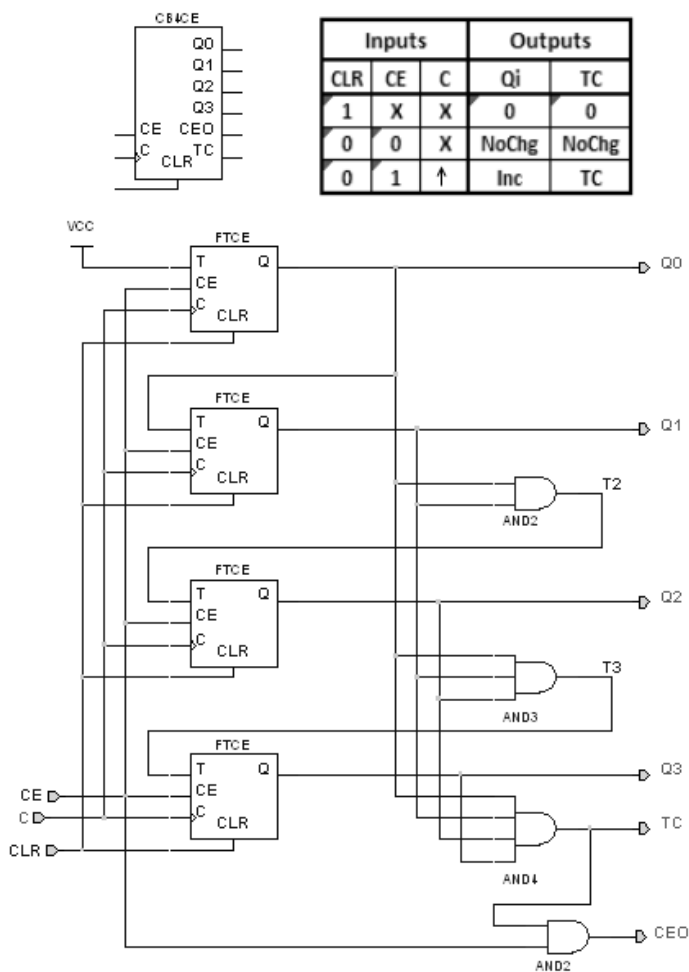


Рис. 69. УГО, таблица переходов и внутренняя структура счётчика CB4CE

Счётчик CB4CE построен на Т-триггерах с входом разрешения синхроимпульсов  $CE$  и входом  $CLR$  асинхронной установки в «0». Асинхронный вход  $CLR$  обладает высшим приоритетом. Межразрядные связи – функции возбуждения разрядов, построены по принципу параллельного переноса. Для этого на входах  $T$  триггеров подключены элементы AND, которые объединяют выходы предыдущих разрядов.

На выходе  $TC$  (terminal count) формируется двоичный сигнал декодирования последнего состояния счётчика, когда все выходы счётчика  $Q3-Q0$  равны «1» –  $TC=Q3 \cdot Q2 \cdot Q1 \cdot Q0$ . Выходной сигнал переноса  $CEO=1$ , когда  $TC=1$  и  $CE=1$  –  $CEO=TC \cdot CE$ .

При сигналах  $CE=0$  и  $CLR=0$  счётчик не изменяет свое состояние и хранит код предыдущего значения. Счётчик работает в режиме счёта (increment) входных импульсов  $C$  при сигналах  $CE=1$  и  $CLR=0$ .

### ***Увеличение разрядности счётчиков***

При необходимости построения многоразрядных счётчиков входящие схемы счётчиков меньшей разрядности каскадируют между собой, для этого счётчики имеют дополнительные управляющие входы. Для асинхронных счётчиков выход старшего разряда каскада или выход переноса соединяется с входом синхронизации последующего каскада.

На рис. 70 приведена схема синхронного шестirazрядного двоичного счётчика по модулю 64 и временная диаграмма работы.

Счётчик реализован на трёх двухразрядных двоичных счётчиках CB2CE с входом разрешения синхроимпульсов  $CE$  и входом  $CLR$  асинхронной установки в «0». Входы синхронизации  $C$  и входы  $CLR$  асинхронной установки в «0» соединены параллельно между собой, так что все внутренние счётчики СТ2, СТ1, СТ0 переключаются и сбрасывается в одно и то же время. Внешний вход разрешения синхроимпульсов  $CE$  соединён с младшим счётчиком СТ0. Выходы  $CEO$  внутренних счётчиков последовательно соединены с входом  $CE$  последующего счётчика. Выходы  $TC2$ ,  $TC1$ ,  $TC0$  последнего состояния счётчиков объединены на элементе AND3, выход которого является внешним выходом  $TC$  шестirazрядного счётчика.

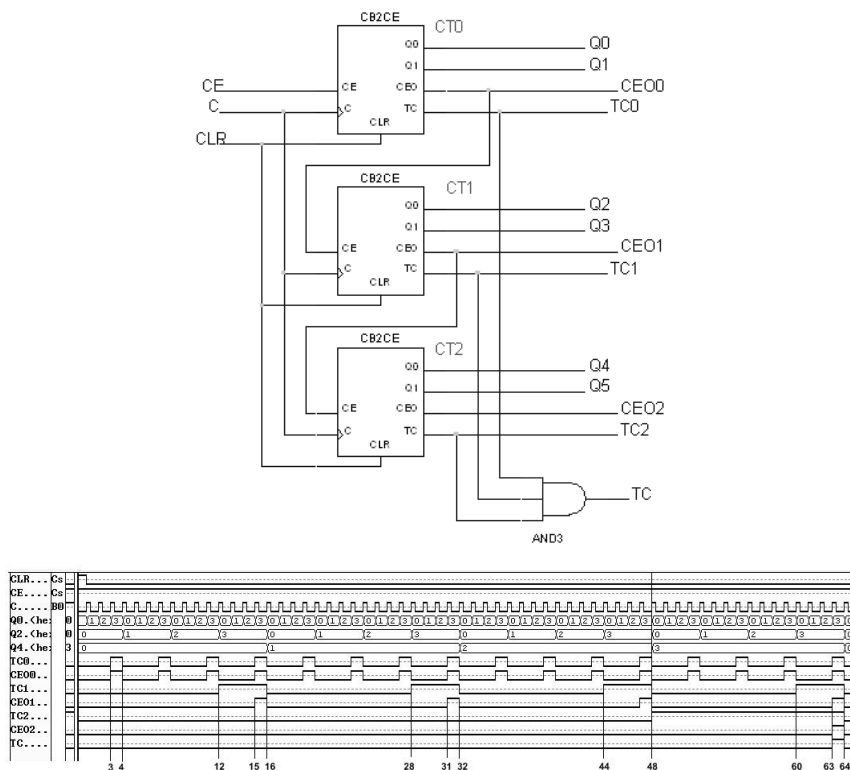


Рис. 70. Схема шестиразрядного счётчика и временная диаграмма работы

Сигнал на выходе  $TC0$  младшего счётчика  $CT0$  формируется только в том случае, когда счётчик находится в состоянии 3. Сигнал на выходе  $CE0$  младшего счётчика  $CT0$  формируется при  $TC0=1$  и подаче сигнала разрешения  $CE=1$ . Далее сигналы на выходах  $TC0$  и  $CE0$  формируются через четыре импульса на состояниях 7, 11, 15, 19, ..., 59, 63 и т.д.

Сигнал на выходе  $TC1$  счётчика  $CT1$  формируется только в том случае, когда счётчик находится в состоянии 3, а для образованного 4-разрядного счётчика в состояниях 12, 13, 14, 15. Сигнал на выходе  $CE1$  счётчика  $CT1$  формируется при  $TC1=1$  и подаче сигнала разрешения  $CE1=1$ , т.е. когда на счётчике установлен код 15. Далее сигналы на выходах  $TC1$  формируются через двенадцать импульсов

на состояниях 28-31, 44-47, 60-63 и т.д. Сигнал  $CE1$  формируется через шестнадцать импульсов на состояниях 15, 31, 47, 63, ... и т.д.

Сигнал на выходе  $TC2$  счётчика формируется в состояниях 48-63, а сигнал разрешения выхода  $CEO2$  только в состоянии 63. Таким образом, сигнал разрешения счёта  $CE$  переходит последовательно от одного двухразрядного счётчика к другому.

На выходе  $TC$  формируется двоичный сигнал декодирования последнего состояния 63 двоичного шестиразрядного счётчика, когда все выходы счётчика  $Q5-Q0$  равны единице, при этом внутренние выходы  $TC2$ ,  $TC1$ ,  $TC0$  тоже равны единице.

Выходы разрядов  $Q5-Q0$ , выходы последнего состояния  $TC$  и разрешения  $CEO2$  являются внешними выходами образованного синхронного двоичного шестиразрядного счётчика.

По этому принципу можно построить счётчик с любым числом разрядов. Быстродействие счёта определяется задержкой прохождения сигнала переноса через все каскады.

### ***Счётчик с заданным модулем***

Счётчики с модулем, не равным степени числа 2, т.е. с произвольным модулем, можно реализовать несколькими методами.

Для построения счётчика с произвольным модулем  $Z$  исходным является двоичный счётчик с модулем  $M=2^n$ , превышающим заданный модуль  $M>Z$ . Такой двоичный счётчик будет иметь  $M-Z$  лишних (неиспользуемых) состояний.

На базе двоичных счётчиков счётчик с заданным модулем можно построить следующими способами:

- управлением сбросом,
- управлением загрузкой,
- управлением сбросом и загрузкой.

В способе управления сбросом используется естественная двоичная последовательность кодов от 0 до  $Z-1$ . Схема счётчика дополняется элементом И, который по состоянию выходов  $Q$  обнаруживает код  $Z-1$  конца счёта, после чего счётчик по синхронному входу  $R$  сбрасывается в «0». В счётчике нельзя использовать асинхронный вход  $CLR$  установки в «0», так как на выходе элемента И возможно появление сигналов помех при переключении разрядов за счёт не одновременного их переключения.

В способе управления загрузкой двоичная  $Z$  последовательность использует коды от  $K$  до  $M-1$ . Перед началом в двоичный счётчик по входам параллельной загрузки загружается код  $K$  начального состояния счёта. Кодом конца счёта в этом случае является штатный сигнал  $CR$  переполнения счётчика. Сигнал  $CR$  поступает на вход разрешения загрузки  $L$  (load), вновь загружая в счётчик код  $K$ .

В третьем способе используются два предыдущих способа. Например, задают пару кодов – начала счёта и конца счёта, разность между которыми будет равна  $Z-1$ . Возможно также применение двух диапазонов счёта  $0-V$  и  $W-P$ , а общем случае может быть и более двух диапазонов. Данный способ может применяться для задач деления частоты входных сигналов.

На рис. 71 приведена схема счётчика по модулю 7 и временная диаграмма работы. Схема реализована на двоично-десятичном четырехразрядном счётчике CD4RLE с входом управления параллельной загрузкой  $L$ , синхронным входом сброса  $R$  и входом разрешения счёта  $CE$ .

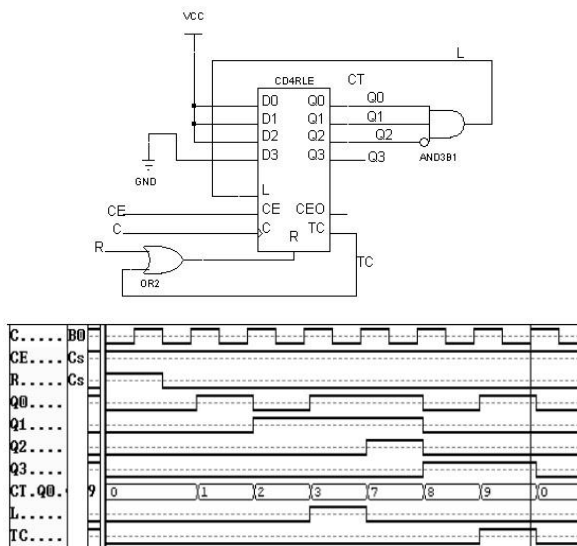


Рис. 71. Схема счетчика по модулю 7 и временная диаграмма работы

По фронту  $0/1$  синхроимпульса и сигнале на входе  $R=1$  счётчик устанавливается в нулевое состояние. При сигнале  $CE=1$  счётчик

работает в счётном режиме. На элементе AND3B1 реализовано условие для определения кода состояния счётчика равного 3. Выход элемента AND3B1 подключён ко входу  $L$  разрешения загрузки и по сигналу на этом входе в счётчик загружается код состояния, заданный на входах  $D3-D0$  параллельной загрузки. В схеме (рис. 71) данный код состояния равен 7. Поэтому после состояния 3 счётчик переходит в состояние 7 и далее вновь переходит в счётный режим. В последнем состоянии счёта 9 на выходе  $TC$  формируется сигнал «1», по которому счётчик переходит в начальное состояние 0.

Таким образом, в счётчике по модулю 7 реализована двоичная последовательность состояний 0, 1, 2, 3, 7, 8, 9. Из анализа временной диаграммы видно, что период сигналов на выходах  $Q$  в семь раз больше периода входного сигнала  $C$ , а следовательно, частота в семь раз меньше входной частоты. На выходах  $Q3, Q2, Q1$  получена равномерная последовательность сигналов. При этом скважность  $S$  сигналов на выходе  $Q2$  равна семи, на выходе  $Q3$  –  $7/2$ , на выходе  $Q1$  –  $7/3$ . На выходе  $Q0$  получена неравномерная последовательность сигналов с частотой сигналов в семь раз меньше входной частоты.

Аналогично могут быть построены счётчики с заданным модулем на стандартных реверсивных счётчиках.

### ***Счётчик с переменным модулем***

Счётчики с переменным модулем счёта позволяют оперативно изменять (программировать) модуль пересчёта.

Одним из способов построения счётчика с управляемым модулем является применение схемы сравнения кодов в цепи синхронного сброса (рис. 72). На одну из групп сравниваемых кодов компаратора подается число  $Z$ . При появлении данного кода на выходах счётчика компаратор формирует единичный сигнал, по которому счётчик переключится в нулевое состояние. Далее процесс будет повторяться. В счётчике используется естественная двоичная последовательность кодов от 0 до  $Z$ . Модуль счёта будет равен значению  $Z+1$ .

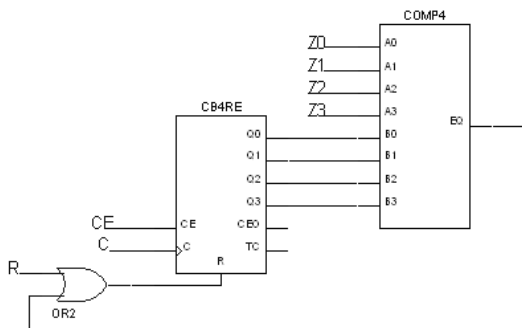


Рис. 72. Схема счётчика с компаратором и переменным модулем

В другом варианте построения счётчика с переменным модулем применяется счётчик с параллельной загрузкой. На рис. 73 приведен счётчик с предварительной установкой на основе суммирующего двоично-десятичного счётчика CD4RLE с модулем счёта  $M=10$ . В начальный момент по фронту 0/1 импульса  $C$  при сигнале  $L=1$  в разряды счётчика заносится код числа  $K$ . Далее счётчик увеличивает коды состояния от  $K$  до числа 9. Когда состояние счётчика равно 9, на выходе  $TC$  формируется единичный сигнал, по которому счётчик переходит в состояние  $K$ . При этом, исключаются состояния от 0 до  $K-1$ . Таким образом, счётчик проходит повторяющуюся последовательность состояний  $K, K+1, \dots, M-1, M$ . Модуль счета будет равным  $Z=M-K$ .

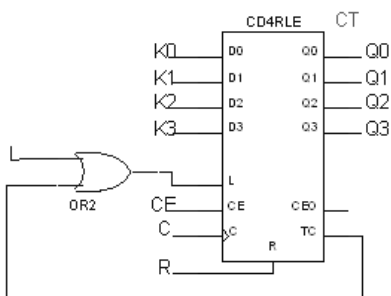


Рис. 73. Схема счётчика с предварительной установкой и переменным модулем

Аналогично можно реализовать счётчик с переменным модулем на вычитающем счётчике. При этом, вычитающий счётчик будет

проходить следующую последовательность состояний:  $K, K-1, \dots, 1, 0, K, K-1, \dots$  и т.д. Модуль счётчика будет равен  $K+1$ . В вычитающем счётчике единичный сигнал заёма ( $BR$  – borrow) на выходе ТС формируется на нулевом состоянии счётчика.

### *Делители частоты*

Делители частоты (frequency divider) представляют собой устройства, частота следования сигналов на выходе которых уменьшается по сравнению с частотой входного сигнала. Они используются, когда частота входного генератора выше необходимой частоты. Коэффициент деления может принимать и дробное значение.

Как отмечалось выше, любой счётчик может выполнять и функции делителя частоты. Частота сигналов с выходов счётчика с модулем  $M$  может быть во столько же раз меньше входных сигналов. В процессе работы на выходах счётчика можно получить сетку частот, находящихся в пропорциональных отношениях. Но на выходах счётчика могут также образовываться неравномерные последовательности сигналов.

При построении делителей частоты важна не только частота выходного сигнала, но и его скважность – отношение периода следования импульсов к длительности этих импульсов. Чаще всего требуется получить меандр – сигнал со скважностью равной двум. Получить меандр из любого сигнала довольно просто: надо использовать дополнительный делитель частоты на 2, при этом частота выходного сигнала уменьшится вдвое.

К примеру, частоты сигналов на выходах двоичного счётчика с естественной двоичной последовательностью состояний будут отличаться друг от друга в два раза и представлять собой симметричные последовательности импульсов (меандры). Двоичные счётчики можно рассматривать как многоразрядный двоичный делитель.

На счётчиках можно строить также управляемые делители частоты, выходная частота которых определяется управляющим кодом. На рис. 74 приведена схема делителя с коэффициентом деления  $2^K$ , где  $K$  – целое число.

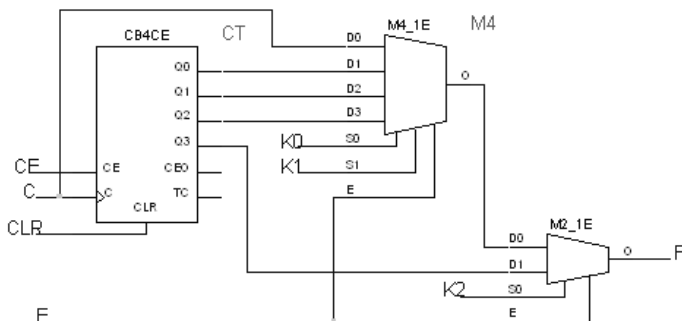


Рис. 74. Схема управляемого делителя частоты

На мультиплексорах M4\_1 и M2\_1 собран мультиплексор M5\_1. Мультиплексор передает на выход входной сигнал  $C$  или один из разрядов счётчика  $CT$ . Выбор номера входа задается трёхразрядным кодом  $K$ . При входном коде  $K=4$  ( $K_2=1$ ,  $K_1=0$ ,  $K_0=0$ ) на выход  $F$  передается сигнал с выхода  $Q_3$  двоичного счётчика на котором частота делится на 16. В момент изменения управляющего кода  $K$  на выходе схемы могут появиться помехи. Поэтому в схеме сначала необходимо установить управляющий код, а затем сигнал  $E$  разрешения работы.

При проектировании счётчиков-делителей можно использовать разложение модуля счёта  $M$  на простые множители счёта, например,  $M_1$  и  $M_2$ , так что их произведение будет равно требуемому модулю  $M=M_1 \cdot M_2$ .

На рис. 75 приведены две схемы счётчиков по модулю 10. Схема счётчика X74\_390 состоит из двух секций счётчиков с модулем 2 (выход  $QA$ ) и модулем 5 (выходы  $QB$ ,  $QC$ ,  $QD$ ). Секция с модулем 2 состоит из одиночного триггера, работающего в счётном режиме, и изменяет свое состояние по фронту  $1/0$  сигнала  $CKA$ . Секция с модулем 5 состоит из трёх триггеров и является синхронным счётчиком. При достижении счётчиком кода 4 по следующему фронту  $1/0$  сигнала  $CKB$  трёхразрядный счётчик сбрасывается в нуль. Секции можно использовать по отдельности или соединять последовательно для получения двоично-десятичного счётчика.

В счётчике-делителе CT1 по модулю 10, приведенном на рис. 75,а, синхровход  $CKB$  трёхразрядного счётчика соединён с выходом  $QA$  младшей одноразрядной секции.

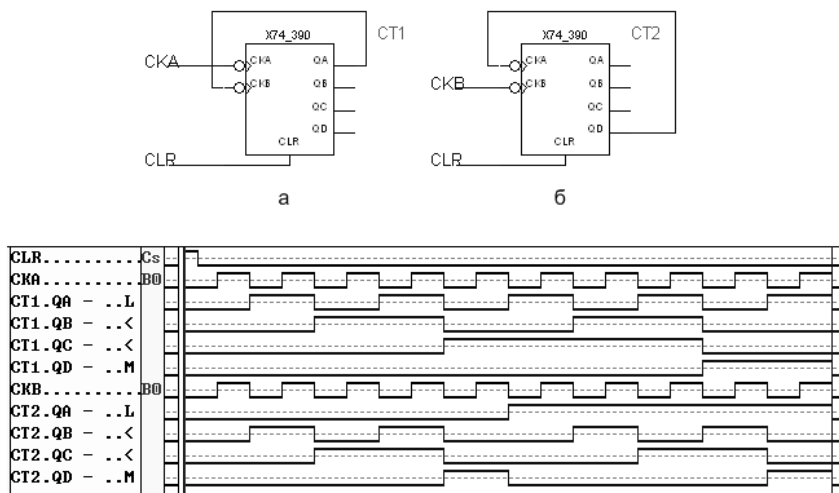


Рис. 75. Схемы счётчика по модулю 10 и временная диаграмма работы

Из временных диаграмм видно, что в схеме на рис. 75,а частота выходного сигнала  $QA$  счётчика  $CT1$  вдвое меньше входной частоты и имеет форму меандра, а на выходах  $QB$ ,  $QC$ ,  $QD$  частота делится на 10. При этом скважность сигналов на выходе  $QC$  равна 2,5, а на выходе  $QD$  равна 5. На выходе  $QB$  получена неравномерная последовательность сигналов.

В счётчике-делителе  $CT2$ , приведённом на рис. 75,б, одноразрядный внутренний счётчик включён после трёхразрядного внутреннего счётчика. Трёхразрядный счётчик делит частоту входного сигнала на 5, но выходные импульсы имеют скважность не равную двум. Сигналы на выходе  $QC$  имеют скважность равную 2,5, а на выходе  $QD$  равную 5. На выходе  $QB$  получена неравномерная последовательность сигналов. Одноразрядный счётчик делит частоту еще вдвое, т.е. делит на 10 входную частоту сигнала на входе  $CKB$  и одновременно формирует меандр на выходе  $QA$ .

В счётчиках с недвоичными модулями отношения между частотами их следования могут быть дробными. При этом импульсы могут формироваться как по фронту 0/1, так и по фронту 1/0. Для этого в схему синхронного счётчика вводятся одна группа разрядов, переключаемых по фронту 0/1, и вторая группа, переключаемых по фронту 1/0. Для получения импульса заданной длительности объе-

диняют соответствующие разряды двух групп. В асинхронных счётчиках с дробным коэффициентом деления можно в цепи обратной связи установить сумматор по модулю два (см. рис. 75).

### Формирователи импульсов

На основе счётчиков можно построить формирователи временных интервалов с длительностью, задаваемой внешним кодом  $K$ , и формирователи одиночной пачки (группы), содержащей заданное число импульсов  $K$ , вырезанных из непрерывной входной последовательности импульсов  $C$  (рис. 76).

Схема формирователя импульсов реализована на реверсивном двоичном 4-разрядном счётчике CB4CLED с входами параллельной загрузки  $DI$ , асинхронным сбросом  $R$  и входом разрешения счета  $CE$ . Вход  $UP$  задает направление счёта:  $UP=1$  увеличение,  $UP=0$  уменьшение. В схеме вход  $UP=0$ , поэтому счётчик работает в режиме вычитающего счётчика. Счётчик срабатывает по фронту 1/0 сигнала  $C$ , так как к синхровходу счётчика подключен инвертор НЕ. Сигнал заёма на выходе  $TC$  формируется при нулевом состоянии всех разрядов счётчика.

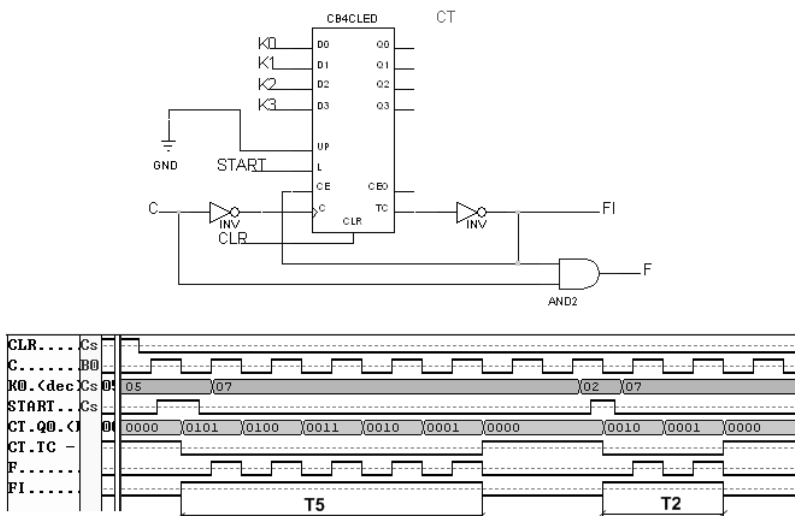


Рис. 76. Схемы формирователя импульсов и временная диаграмма работы

Перед началом работы по сигналу  $CLR$  счётчик сбрасывается в «0», поэтому выход  $TC=1$  и на выходах  $F$  и  $FI$  также установлены нули. При коротком сигнале  $START=1$  после синхроимпульса  $C$  по его по фронту 1/0 значение кода  $K$  заносится в счётчик. Так как состояние счётчика не равно 0, то изменяется состояние выхода  $TC=0$  и разрешается прохождение импульсов с входов  $C$  через элемент  $AND2$  на выход  $F$ . После того как на выход  $F$  пройдёт  $K$  импульсов, на счётчике установится нулевое состояние и сигнал  $TC=1$ , который приведёт к запрету прохождения импульсов на выход  $F$ . Дальнейшая работа формирователя возобновится по следующему сигналу  $START=1$ . На временной диаграмме первая пачка  $T5$  состоит из пяти импульсов, вторая пачка  $T2$  – из двух импульсов.

Входные положительные синхроимпульсы  $C$  поступают на выходной элемент  $AND2$ , а на синхровход счётчика отрицательные синхроимпульсы  $C$ . Это позволяет сохранять длительность входных синхроимпульсов на выходе  $F$ , а также запретить прохождение возможных помех с выхода  $TC$ , образуемых за счёт одновременного переключения разрядов счётчика.

Из временной диаграммы видно, что по сигналам  $START$  на выходе  $FI$  формируются временные интервалы с длительностью, задаваемой кодом  $K$  (на рис. 76 интервалы  $T5$  и  $T2$ ). Длительность формируемого временного интервала  $T$  равна числу  $K$  периодов  $t$  входного синхросигнала:  $T=K \cdot t$ .

**Измерители длительности входных импульсов.** В этом случае входной сигнал поступает на вход  $CE$  разрешения счёта. Счётчик работает в режиме суммирующего счётчика. Перед началом работы по сигналу  $START$  счётчик сбрасывается в нуль. После окончания входного сигнала в счётчике будет установлен код, пропорциональный длительности этого сигнала.

**Измерение частоты следования входных импульсов.** В схеме применяется метод подсчёта входных импульсов в течение заданного временного интервала  $T_{\text{зад}}$ . Перед началом работы по сигналу  $START$  счётчик сбрасывается в нуль. На входе разрешения счёта формируется сигнал длительностью  $T_{\text{зад}}$ . По окончании счёта будет зафиксировано количество периодов  $K$  входного сигнала, которому пропорциональна входная частота. Если выбрать временной интервал  $T_{\text{зад}}=1$  мкс, то код  $K$  будет равен частоте входного сигнала в мегагерцах.

## Преобразователи код–частота

Преобразователи кода в частоту применяют в цифровых синтезаторах частоты. В зависимости от применения они называются цифро-частотными умножителями, цифро-частотными интеграторами, интерполяторами. Цифровые синтезаторы формируют сигналы, частота следования которых прямо пропорциональна входному коду, т.е. они умножают входную частоту  $F_0$  на коэффициент  $K$ . В таких устройствах интервал времени  $T$  заполняется импульсами с частотой  $F_0$ , причем их количество определяется числом  $K$ .

Схема трёхрядного синтезатора частоты и временная диаграмма работы приведены на рис. 77.

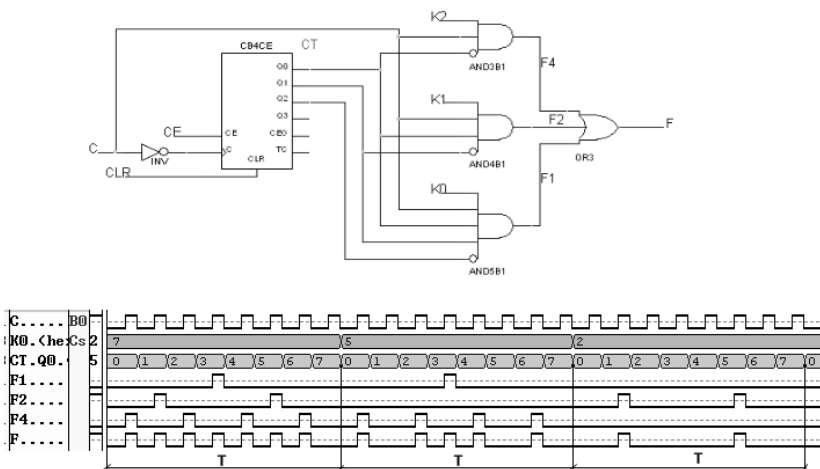


Рис. 77. Схемы синтезатора частоты и временная диаграмма работы

Интервал времени периода  $T$  составляет 8, а код заполнения  $K$  принимает значения от 0 до 7. Основу схемы составляет синхронный двоичный счётчик. На элементах AND объединены инверсное значение данного разряда и прямые значения всех более младших разрядов, а также соответствующие разряды входного кода  $K$ . На выходах элементов AND формируются соответствующие импульсные последовательности: на выходе  $F4$  – четыре импульса, выходе  $F2$ , соответственно, два импульса, выходе  $F1$  – один импульс. Причем импульсы не совпадают друг с другом во времени, что видно на временной диаграмме (см. рис. 77), а частота последовательно-

стей уменьшается вдвое. Данные сигналы формируются лишь при наличии единиц в соответствующих разрядах кода  $K_2$ ,  $K_1$ ,  $K_0$ .

Выходной элемент OR3 проводит логическое сложение импульсов во времени. На выходе  $F$  каждый частотный компонент может быть представлен независимо от наличия или отсутствия других. При этом количество импульсов на интервале  $T$ , равном восьми периодам частоты входного сигнала, определяется соотношением  $K=K_0+2\cdot K_1+4\cdot K_2$ . На временной диаграмме приведены три интервала  $T$  для  $K=7$ ,  $K=5$ ,  $K=1$ .

Следует обратить внимание, что выходные импульсы для произвольных кодов могут быть распределены во времени неравномерно.

### 1.2.3. Регистры

**Регистр** – это функциональное устройство последовательностного типа, предназначенное для хранения  $n$ -разрядных двоичных чисел и выполнения преобразований над ними. Регистр представляет собой упорядоченную совокупность из двух или большего числа триггеров с общим входом тактового сигнала и соединённых между собой определённым образом при помощи комбинационных схем.

Регистры имеют широкий диапазон применения. Они используются для преобразования форматов представления чисел из последовательного в параллельный, а также обратного преобразования, в линиях передачи данных и отображения; для создания управляющих и запоминающих схем, генераторов кодов; применяются во внутренних регистрах процессора и арифметических блоках и т.д.

#### *Классификация регистров*

Регистры классифицируются по способам ввода-вывода и представления входной и выходной информации на два класса: параллельные и последовательные или сдвиговые регистры. Общими для всех разрядов являются цепи управления – тактирования, сброса/установки, разрешения выхода или приема.

**В параллельных (статических) регистрах** хранения каждый триггер имеет свой независимый информационный вход и свой независимый информационный выход. Схемы разрядов не обмениваются данными между собой. Разряды регистров хранения могут быть реализованы на триггерах любого типа, реализующих функцию D-триггера.

**Последовательные (сдвигающие) регистры** представляют собою цепочку разрядных схем, связанных цепями переноса. Под действием сигнала тактирования (синхронизации) происходит перемещение (сдвиг) разрядов кода между триггерами. Особенность построения регистра сдвига состоит в том, что при выполнении сдвига каждый разряд должен одновременно выдавать информацию в следующий разряд и принимать новую информацию из предыдущего разряда. Поэтому нельзя применять одноступенчатые триггеры, управляемые уровнем. При построении регистров сдвига применяют синхронные триггеры с динамическим управлением записью либо двухступенчатые. Внутренняя структура данных триггерных схем предусматривает разделение во времени этапов приема входной информации и смены выходного сигнала триггера. По одному фронту синхронизирующего импульса происходит прием информации, по другому – смена выходного сигнала.

Регистры различают по типу ввода (записи, приема, загрузки) и вывода (выдачи) информации:

- с последовательным вводом и выводом информации;
- с параллельным вводом и выводом информации;
- комбинированные: с параллельно-последовательным вводом и последовательно-параллельным выводом.

Регистры по направлению передачи информации различают на однонаправленные и реверсивные (двунаправленные), в которых используются триггеры с третьим состоянием на выходе или открытым коллектором, что позволяет использовать их в качестве буферных регистров или подключать непосредственно на магистраль данных.

По числу линий представления одного разряда различают однофазные и парафазные связи; при однофазных каждый разряд передается по одной линии, а при парафазном – по двум линиям, одновременно передается прямое и инверсное значение разряда.

Типичными для регистров являются следующие микрооперации:

- параллельный ввод (прием, запись) данных;
- последовательный ввод (сдвиг) данных – преобразование последовательного кода в параллельный и наоборот;
- направление сдвига: влево, вправо, реверсивный;
- вид сдвига: логический, циклический, арифметический;
- преобразование данных: прямой, обратный или дополнительный код;
- сдвиг на заданное число разрядов;
- хранение данных;
- разрешение выдачи данных – последовательно или параллельно;
- установка в начальное состояние (сброс).

### *Операции сдвигов*

**Сдвиг** – это одновременное пространственное перемещение двоичного слова в разрядной сетке с сохранением порядка следования единиц и нулей. Операции сдвига позволяют реализовать различные преобразования данных, работать с отдельными битами, быстро выполнять умножение и деление чисел на степень 2, а также применяются для нормализации чисел. Различают три типа сдвигов – логический, циклический и арифметический; направление сдвига – вправо или влево; количество разрядов, на которые выполняется сдвиг.

На рис. 78 приведены механизмы сдвига битов в регистрах.

**Логический сдвиг.** При логическом сдвиге вправо одновременно все биты разрядов внутри регистра сдвигаются в сторону младших разрядов, а самый младший бит выдвигается и теряется. В старший разряд поступает значение сигнала с входа *SRI*. При логическом сдвиге влево все биты разрядов внутри регистра сдвигаются в сторону старших разрядов, а самый старший бит выдвигается и теряется. В младший разряд поступает значение сигнала с входа *SLI*.

Логический сдвиг вправо можно использовать для деления целых чисел без знака на степень 2. С помощью сдвига влево можно быстро умножать числа на степень 2. Кроме того, с помощью логи-

ческого сдвига выполняется преобразование параллельного кода в последовательный и наоборот.

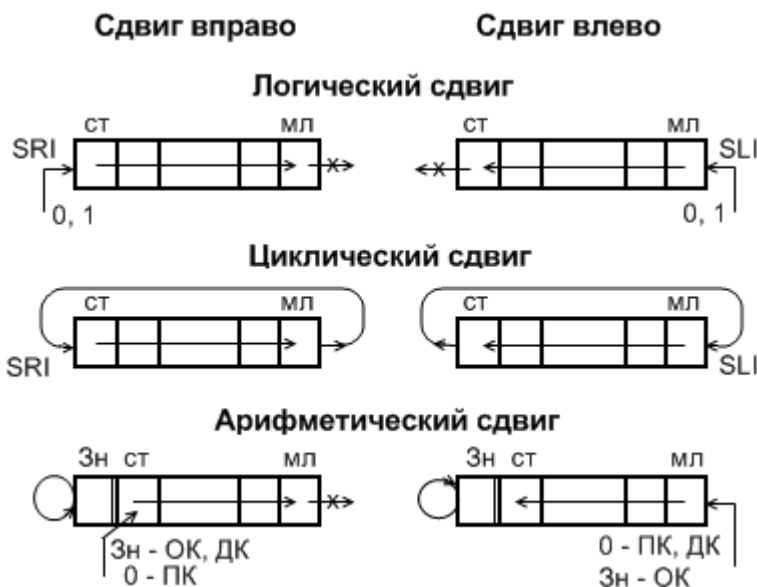


Рис. 78. Операции сдвигов

**Циклический сдвиг.** Для выполнения циклического сдвига в регистре устанавливается кольцевая связь между старшим и младшим разрядами. Вход *SLI* соединяется с выходом старшего разряда, а вход *SRI* с выходом младшего разряда регистра. При циклическом сдвиге все биты разрядов внутри регистра одновременно сдвигаются вправо или влево, а выдвигаемые биты поступают на соответствующие входы *SLI* или *SRI*.

**Арифметический сдвиг.** Арифметический сдвиг предназначен для чисел со знаком. Арифметический сдвиг отличается от логического тем, что он не изменяет значение старшего бита. При арифметическом сдвиге биты числовых разрядов сдвигаются, соответственно, влево или вправо, а знаковый разряд не изменяется. Выдвигаемые биты удаляются. Значения, поступающие при сдвиге в числовые разряды, зависят от кода представления чисел (прямой, обратный или дополнительный код). При сдвиге вправо в старший

разряд поступает логический нуль в прямом коде и значение знакового разряда в дополнительном и обратном кодах. При сдвиге влево в младший разряд поступает логический нуль в прямом и дополнительном кодах и значение знакового разряда в обратном коде.

Арифметический сдвиг вправо можно использовать для деления чисел со знаком на степень 2. С помощью арифметического сдвига влево можно быстро умножать числа на степень 2. Но в результате выполнения операции может быть получено переполнение разрядной сетки.

### *Универсальные регистры*

В сериях ИС и библиотеках элементов ПЛИС имеется много различных вариантов регистров. Регистры, способные выполнять набор микроопераций, называются многорежимными (многофункциональными) или универсальными. Входы регистра, задающие вид выполняемой микрооперации, называются управляющими.

Примером универсального регистра является 4-разрядный реверсивный сдвиговый регистр X74\_194 с инверсным асинхронным входом сброса, последовательными и параллельными входами данных и параллельными выходами (рис. 79 и рис. 80). Регистр построен на четырёх D-триггерах с асинхронной установкой в «0» и динамическим входом синхронизации. Регистр содержит четыре входа параллельной загрузки  $A$ ,  $B$ ,  $C$ ,  $D$  ( $A$  – старший разряд,  $D$  – младший разряд). Регистр имеет два входа последовательного ввода:  $SRI$  (shift-right input) и  $SLI$  (shift-left input).

В регистре X74\_194 управляющие входы  $S1$  и  $S0$  задают четыре режима работы:

$S1=0, S0=0$  – хранение;

$S1=0, S0=1$  – сдвиг вправо в сторону младших разрядов –  
 $SRI \rightarrow QA, QA \rightarrow QB, QB \rightarrow QC, QC \rightarrow QD$ ;

$S1=1, S0=0$  – сдвиг влево в сторону младших разрядов –  
 $SLI \rightarrow QD, QD \rightarrow QC, QC \rightarrow QB, QB \rightarrow QA$ ;

$S1=1, S0=1$  – параллельная загрузка по входам  $D3-D0$ .

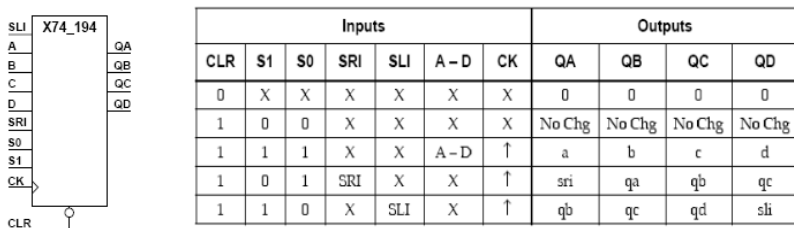


Рис. 79. УГО и таблица микроопераций регистра X74\_194

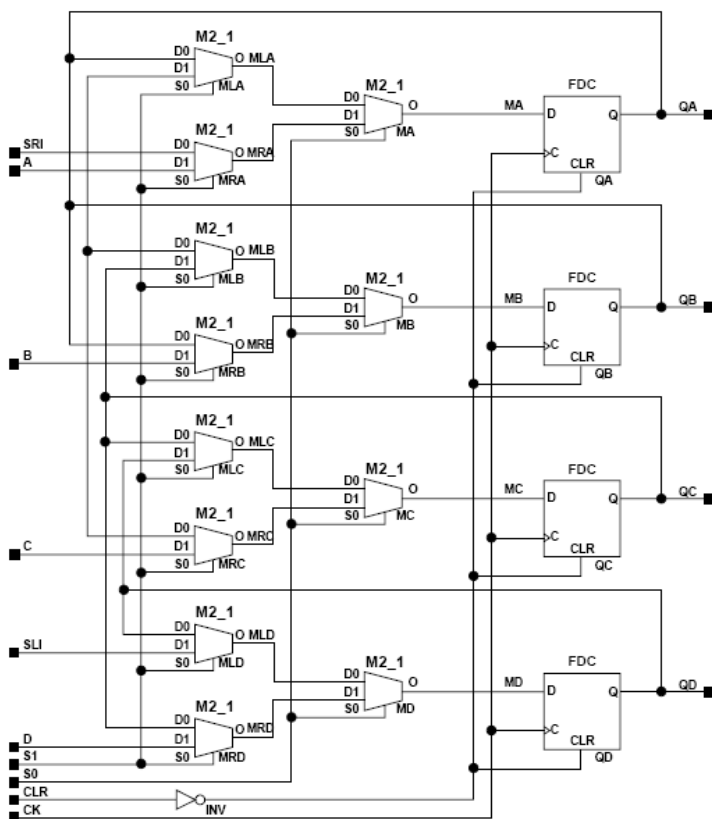


Рис. 80. Логическая схема регистра X74\_194

Управляющие входы  $S1$  и  $S0$  осуществляют мультиплексирование сигналов. Так как триггеры не имеют входа разрешения синхроимпульсов  $CE$ , то операция хранения выполняется передачей на входы  $D_i$  регистров значения выходов  $Q_i$  этих же разрядов. Все микрооперации выполняются по фронту 0/1 синхроимпульса  $СК$ .

Установка в нулевое состояние проводится при подаче на вход  $CLR$  нулевого значения:  $CLR=0$ .

### Увеличение разрядности регистров

Каскадирование сдвигающих регистров для увеличения их разрядности осуществляется соединением последовательного выхода одного регистра с последовательным входом другого регистра. Построенный таким способом реверсивный 12-разрядный регистр сдвига с входами параллельной загрузки приведен на рис. 81

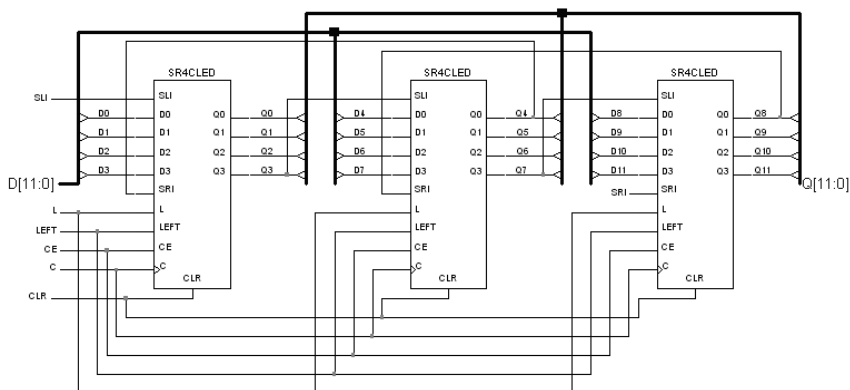


Рис. 81. Схема 12-разрядного регистра сдвига

У реверсивного 4-разрядного регистра  $SR4CLED$  вход  $LEFT$  задает направление сдвига: при  $LEFT=0$  выполняется сдвиг вправо,  $LEFT=1$  – сдвиг влево. Входы управления  $LEFT$ ,  $L$ ,  $CE$ , синхровход  $C$  и вход  $CLR$  асинхронной установки в нулевое состояние для построения 12-разрядного регистра объединяются, соответственно, между собой и являются внешними входами регистра. Вход после-

довательного ввода *SLI* (сдвиг влево) младшего регистра и вход ввода *SRI* (сдвиг вправо) старшего регистра являются внешними входами 12-разрядного регистра. Входы параллельной загрузки *D* регистров составляют 12-разрядную входную шину  $D[11:0]$ , а выходы *Q* соответственно 12-разрядную выходную шину  $Q[11:0]$ .

### ***Счетчик на регистре сдвига***

Работа регистра сдвига может быть описана графом переходов. Закон функционирования 3-разрядного регистра сдвига при сдвиге влево на один разряд характеризуется графом переходов, приведенным на рис. 82. В узлах указаны значения состояний регистра, а на дугах значение сигнала на входе последовательного ввода *SLI*.

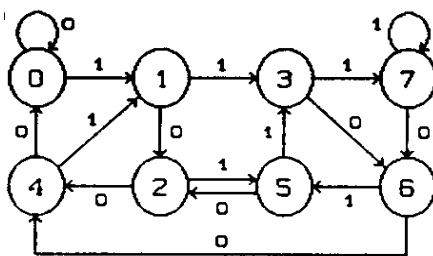


Рис. 82. Граф переходов 3-разрядного регистра сдвига

Из графа переходов видно, что последовательность переходов между некоторыми внутренними состояниями образует замкнутое кольцо. Следовательно, на сдвигающих регистрах можно синтезировать счётчики с модулем переходов до восьми. Например, при последовательности переходов  $0 \rightarrow 1 \rightarrow 3 \rightarrow 6 \rightarrow 4 \rightarrow 0$  получим счётчик по модулю пять. Разработав таблицу истинности для функций переходов счетчика, можно синтезировать функцию возбуждения для входа ввода *SLI*. Схемы счётчика по модулю пять и временные диаграммы работы приведены на рис. 83: на трёх отдельных триггерах *FD* (рис. 83,а), на сдвиговом регистре *SR4CE* (рис. 83,б), и регистре хранения *FD4CE* (рис. 83,в).

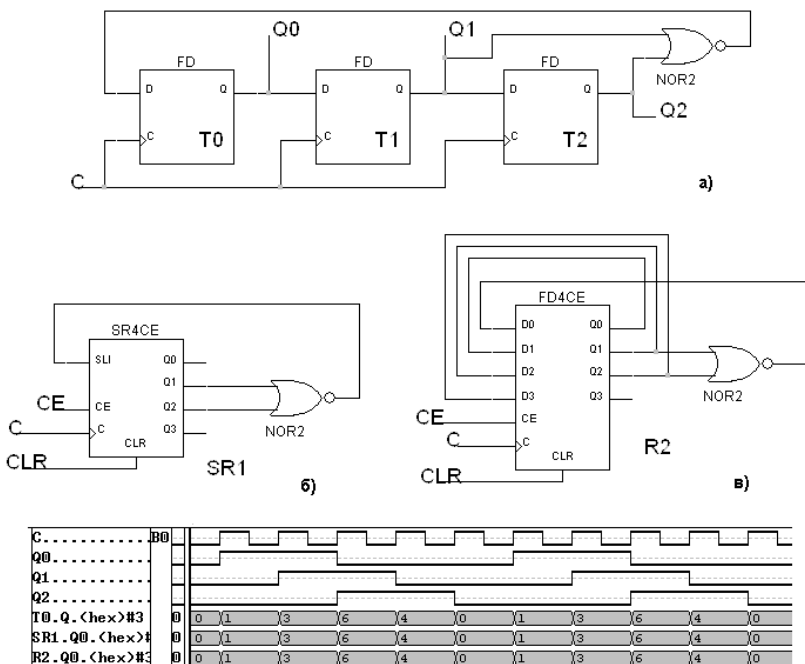


Рис. 83. Счётчик по модулю пять на регистре сдвига

Счётчик имеет три неиспользуемых состояния 2, 5, 7. При случайном попадании в эти состояния счётчик вернется в заданную последовательность, что видно из графа переходов: так как в этих состояниях назначение входа  $SLI = 0$ , то будут следующие переходы:  $2 \rightarrow 4 \rightarrow 0$ , или  $5 \rightarrow 2 \rightarrow 4 \rightarrow 0$ , или  $7 \rightarrow 6 \rightarrow 4 \rightarrow 0$  и т.д.

На рис. 84 приведен полный граф переходов 4-разрядного сдвигающего регистра при сдвиге влево на один разряд. По графу переходов можно выбрать замкнутый цикл из  $M$  состояний и синтезировать счётчик с модулем  $M=2, 3, \dots, 16$ .

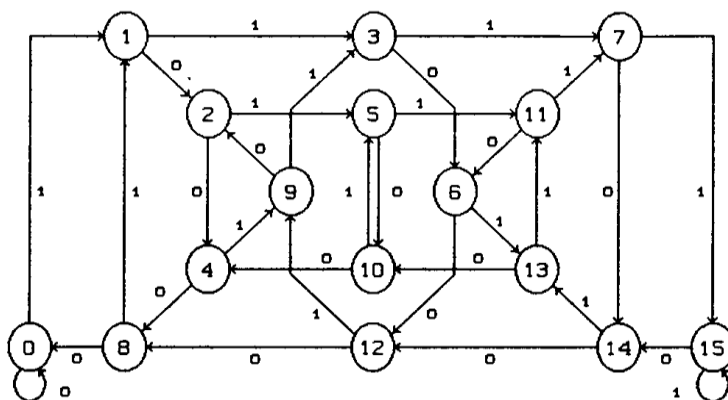


Рис. 84. Граф переходов 4-разрядного регистра сдвига

### ***Кольцевые счётчики. Распределители тактов***

**Кольцевые счётчики** – это замкнутые «в кольцо» цепью обратной связи регистры сдвига. В таких регистрах выход последнего разряда подсоединён к входу *SLI* последовательного ввода в младший разряд. Кольцевые счётчики реализуют счёт в коде «1 из  $N$ ». Они находят применение в системах синхронизации и управления. На их основе можно получить импульсные последовательности с заданными временными параметрами.

В простейшем случае по кольцу циркулирует одна единица, так что коэффициент пересчёта равен числу входящих в него триггеров. В кольцевом счётчике в каждый момент времени только один выходной сигнал равен «1» (счёт в прямом унитарном коде 0010...00) или «0» (счёт в инверсном унитарном коде 1101...11). Дешифрирование состояний кольцевого счётчика осуществляется очень легко и не требует никаких дополнительных элементов. Порядковый номер триггера, находящегося в состоянии единицы, соответствует числу счётных импульсов, поступивших на вход счётчика к моменту наблюдения.

**Распределителями тактов** называют функциональные узлы, распределяющие входной поток импульсов последовательно, импульс за импульсом, по нескольким выходам.

Распределители могут быть построены на двоичном счётчике с дешифратором, подключенным к его выходам. Однако при этом на

выходах дешифратора могут появляться короткие всплески – «иголочки», ширина которых определяется задержками срабатывания триггеров друг относительно друга. Роль распределителя может играть и кольцевой сдвиговый регистр.

Распределители тактов делятся на распределители уровней (когда отсутствуют паузы между сигналами соседних выходов) и распределителями импульсов РИ (когда импульсы стробируются импульсами задающего генератора). Распределители импульсов реализуются на основе распределителей уровней путем включения в их выходные цепи конъюнкторов, вторые входы которых соединены с генератором.

На рис. 85 приведена схема 5-разрядного распределителя тактов на основе кольцевого регистра с самовосстановлением. На выходах  $U0-U4$  формируются сигналы распределителя уровней, а на выходах  $C0-C4$  сигналы распределителя импульсов.

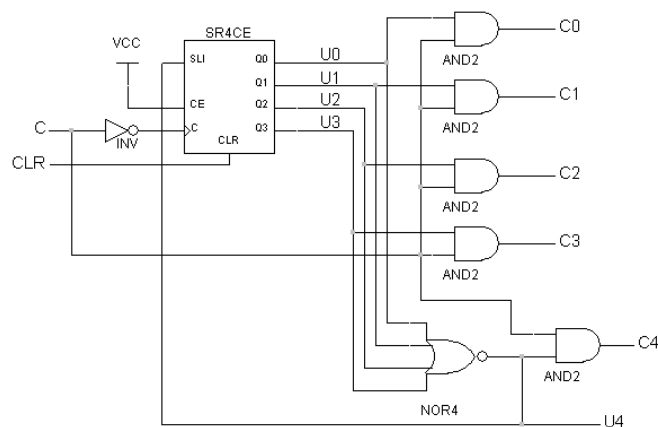


Рис. 85. Распределитель тактов с самовосстановлением  
и временные диаграммы работы

Недостатком схемы кольцевого регистра является возможное нарушение работы при сбое. Например, появление двух единиц в регистре приведёт к дальнейшей их циркуляции и схема не вернется в рабочее состояние. Если требуется разработать систему, функционирующую по заданному алгоритму без сбоев, то необходимо обеспечить условия, при которых система восстанавливала бы свою работоспособность. Один из подходов заключается в разработке схемы контроля состояний системы, которая при неправильной работе автоматически возвращает ее в рабочее состояние.

В схеме распределителя на рис. 85 элемент NOR4, соединенный с входом *SLI* последовательного ввода, контролирует наличие единиц в разрядах регистра. При наличии в регистре одной и более единиц на входе формируется логический ноль и только при всех нулевых значениях формируется логическая единица. Таким образом, за несколько тактов, пока в регистре имеется хотя бы одна единица, на вход регистра подаются нули и удаляются лишние единицы. При этом реализуется самовосстановление рабочего цикла с циркуляцией одной единицы. Потеря единственной единицы также будет исключена. Выход логического элемента NOR4, выполняющего самовосстановление схемы, даёт ещё один дополнительный выход.

В схеме распределителя импульсов (см. рис. 85) на входе синхроимпульсов *C* регистра установлен инвертор, чтобы разнести моменты переключения регистра и формирования выходных импульсов *C4–C0*. Если регистр будет срабатывать по фронту 0/1 синхроимпульса и по единичному уровню этого импульса будут формироваться выходные сигналы *C4–C0*, то на выходах возможно появление коротких импульсов. Пример двухфазного распределителя импульсов *C1* и *C2* и триггером, срабатывающим по фронту 0/1, приведен на рис. 86. Из временной диаграммы работы видно наличие коротких всплесков импульсов.

Распределители на кольцевых регистрах применяются при малом числе выходов, когда необходимо иметь по триггеру на каждый выход. Достоинством распределителей на кольцевых регистрах является отсутствие дешифраторов и, как следствие, отсутствие паразитных импульсов и высокое быстродействие (задержка переключения в новое состояние равна времени переключения триггера).

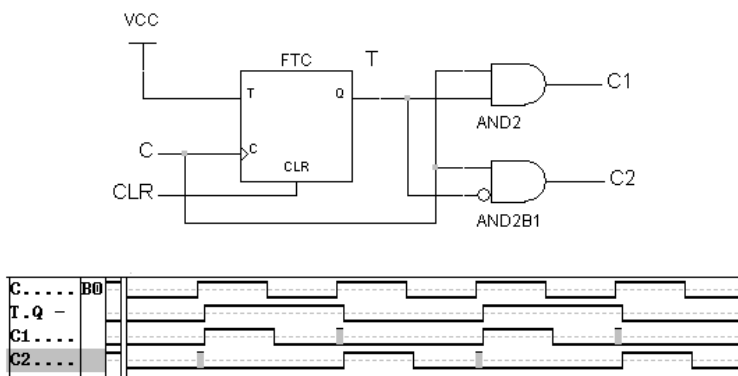


Рис. 86. Двухфазный распределитель импульсов

### Счётчик Джонсона

**Счётчиком Джонсона** (счётчик Мебиуса, счётчик Либау–Крейга, кольцо Реженера, скрученный кольцевой счётчик) называется кольцевой счётчик с перекрестной обратной связью, подаваемой с инверсного выхода старшего разряда регистра на его вход *SLI* последовательного ввода в младший разряд.

Счётчик Джонсона имеет  $2n$  состояний, где  $n$  – количество разрядов регистра, т.е. при той же разрядности вдвое больше, чем обычный кольцевой счётчик ( $n+1$  состояние для самовосстанавливающегося кольцевого счётчика).

Особенность работы счётчика Джонсона состоит в том, что при счёте вначале от первого до последнего разряда регистра распространяется «волна единиц», а затем «волна нулей». Выход счётчика Джонсона не представлен в коде «1 из  $N$ », поэтому для применения в качестве распределителя тактов требуется преобразование кодов. Для преобразования требуется добавление на каждый выход распределителя одного двухвходового элемента И или И-НЕ. Принцип дешифрации состоит в выявлении на временной диаграмме границы между зонами нулей и единиц. В двух случаях, для кодов состоящих только из единиц и только из нулей, состояние определяется анализом крайних разрядов.

На рис. 87 приведена схема и временная диаграмма работы счётчика Джонсона. На выходах счетчика получены сигналы для распределителя тактов «1 из 8».

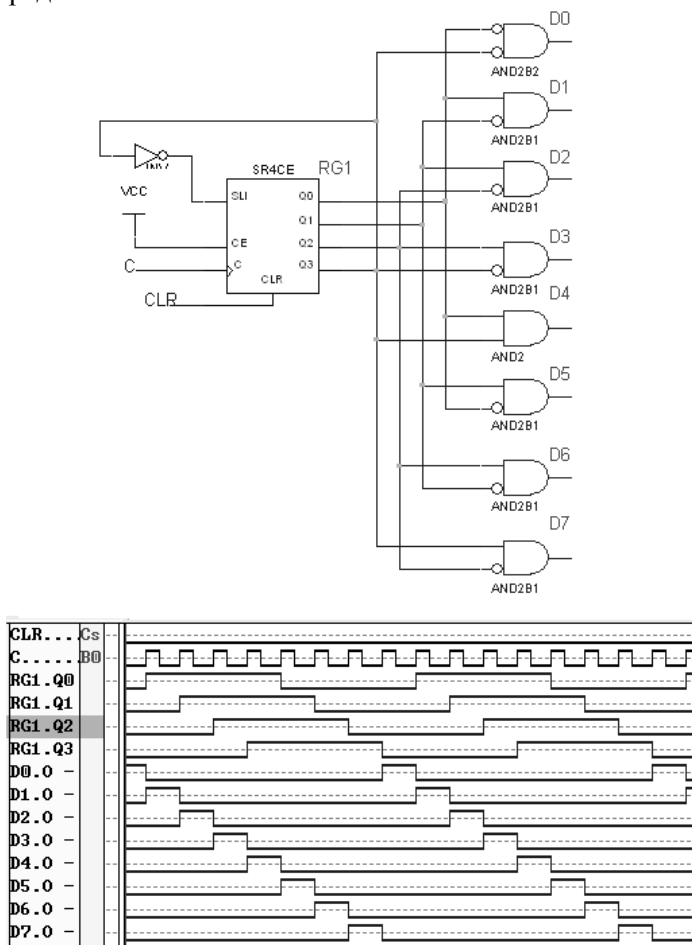


Рис. 87. Счётчик Джонсона и временная диаграмма работы

Счётчик Джонсона из четырёх разрядов имеет модуль пересчёта равный 8 и ему соответствует граф переходов, приведённый на рис. 88,а. Легко показать, что при попадании счетчика в одно из неиспользуемых состояний он будет циркулировать в соответствии с графом переходов, приведённом на рис. 88,б.

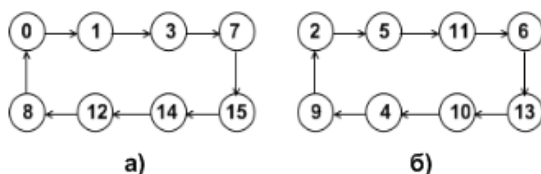


Рис. 88. Граф переходов счётчика Джонсона

Для автоматического выхода счётчика из неиспользуемых состояний можно изменить функцию возбуждения входа  $SLI$ . Функция  $SLI$  определена лишь на восьми наборах переменных 0, 1, 3, 7, 15, 14, 12, 8, которые являются рабочими, а на остальных комбинациях 2, 5, 11, 6, 13, 10, 4, 9 значения функции могут задаваться произвольным образом. Минимальной форме СДНФ соответствует нижнее покрытие  $SLI = \neg Q_3$  в диаграмме Вейча, приведённой на рис. 89. На диаграмме введены два дополнительных покрытия, которые позволяют решить задачу восстановления счётчика в рабочий цикл. На рисунке 89 приведена схема счётчика Джонсона с самовосстановлением. Из анализа работы счётчика следует, что самый длинный путь самовосстановления в рабочий цикл займет три такта, например, переходы  $2 \rightarrow 5 \rightarrow 11 \rightarrow 7$  или  $6 \rightarrow 13 \rightarrow 11 \rightarrow 7$  или  $10 \rightarrow 5 \rightarrow 11 \rightarrow 7$ , или два такта  $4 \rightarrow 9 \rightarrow 3$ .

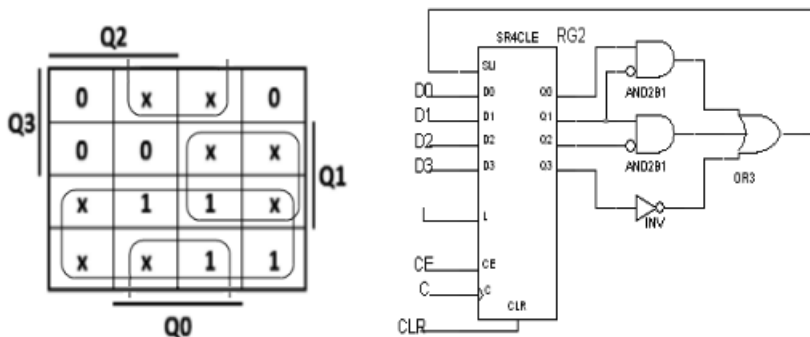


Рис. 89. Счётчик Джонсона с самовосстановлением

## Генераторы псевдослучайных последовательностей

**Генераторы псевдослучайных последовательностей (ГПСП)** реализуют алгоритм, генерирующий кодовую последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному). ГПСП применяются в задачах защиты информации, при моделировании систем и решении математических задач с учётом случайного распределения параметров и др.

В ГПСП двоичных символов вероятность появления на следующем шаге последовательности значений «0» или «1» равновероятна и равна 0,5. Для таких последовательностей заранее нельзя предсказать, каким будет следующий символ, но в последовательности любой длины количество единиц и нулей одинаково.

На рис. 90 приведена простая схема обеспечения секретности передачи сообщений. Операции шифрования и расшифрования реализуются на двух элементах суммирования по модулю два.



### Последовательности:

|                 |    |                  |
|-----------------|----|------------------|
| Сообщение       | DI | 1111000011110000 |
| Случайная       | XI | 1001101001010110 |
| Зашифрованная   | YI | 0110101010100110 |
| Восстановленная | FI | 1111000011110000 |

Рис. 90 Схема обеспечения секретности передачи сообщений

На входы схемы поступает заданная цифровая последовательность информации  $DI$  (например, на рис. 90 это символы с выхода счетчика Джонсона) и случайная последовательность  $XI$  такой же длины. На первом элементе XOR проводится смешивание сигналов и на выходе получается зашифрованная последовательность двоичных символов  $YI = DI \oplus XI$ , отличная от последовательностей  $DI$  и  $XI$ . При декодировании вновь выполняют операцию XOR –  $FI = YI \oplus XI$ . Видно, что на выходе  $FI$  восстановлена исходная последовательность  $DI$ , так как  $DI \oplus XI \oplus XI = DI$ .

Таким образом, случайная последовательность может служить ключом как для шифрования так и для расшифрования сообщений.

Аппаратно ГПСП реализуются на счётчиках, построенных на регистрах сдвига с линейными обратными связями, реализованными через элементы «Исключающее ИЛИ» (сумматоры по модулю два). При этом на вход *SLI* последовательного ввода регистра сдвига поступает сумма по модулю два определённых выходов разрядов регистра. Эта обратная связь определяет последовательность состояний, через которые проходит счётчик.

Такие счётчики будут генерировать псевдослучайную последовательность максимальной длины, содержащую символы единиц и нулей. Общее количество символов в последовательности на единицу меньше максимального возможного числа символов, так как состояние «все нули» является запрещённым. Поэтому количество единиц будет на единицу больше, чем нулей.

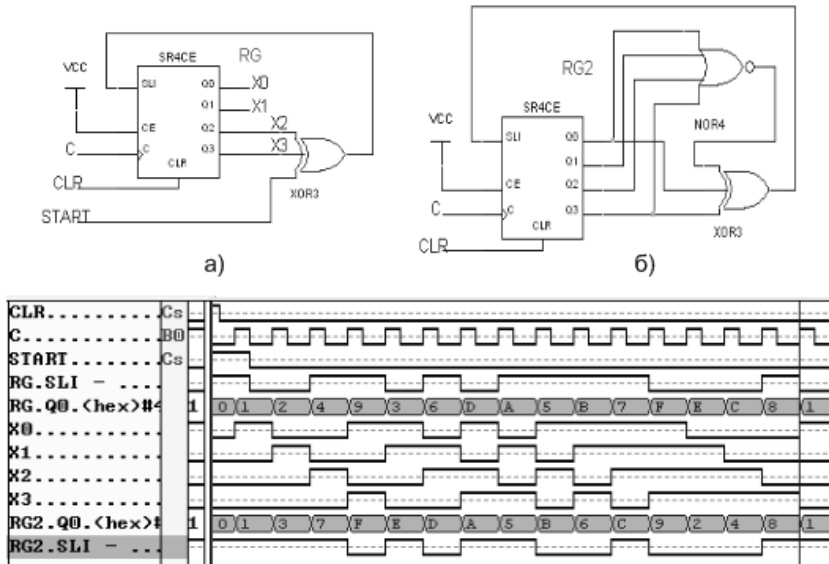


Рис. 91. Схема генератора псевдослучайной последовательности

На рис. 91 приведены два варианта схемы генератора псевдослучайной последовательности и временная диаграмма работы.

Один из входов элемента «сумма по модулю два» всегда подключается к выходу последнего триггера (иначе его подключение теряет смысл), а второй вход подключается к выходу других разрядов. При этом будут генерироваться различные последовательности символов, не всегда максимальной длины.

Так как в схемах используются регистры сдвига, то, как видно из временных диаграмм, с каждого выхода регистра можно снимать идентичные последовательности с модулем  $M=2^4-1=15$ , смещенные во времени. Данные последовательности называются псевдослучайными, так как они, с одной стороны, обладают свойствами случайных сигналов, а с другой – повторяются через определенные интервалы времени. Кроме того, если существует алгоритм генерации, то появление в следующем такте нуля или единицы для конкретного ГПСР можно предсказать. Однако если период повторения последовательности значительно больше интервала наблюдения, то предсказание может стать вычислительно неразрешимым. Если ГПСР содержит 20-разрядный регистр сдвига, то модуль счетчика уже составит  $M=1048575$  состояний.

Схема ГПСР приведенная на рис. 91,а останавливается сигналом сброс *CLR* и запускается сигналом *START*, при котором по первому синхроимпульсу *C* записывается единица в младший разряд регистра через третий вход элемента «сумма по модулю два».

На рис. 91,б приведена схема ГПСР, обладающая свойством самозапуска. В данном ГПСР при состоянии регистра равном нулю, на выходе элемента ИЛИ-НЕ формируется единица,

### ***Регистр последовательного приближения***

Данный регистр предназначен для построения аналого-цифровых преобразователей, работающих по принципу последовательного приближения.

Принцип последовательного приближения соответствует процедуре взвешивания предмета. При этом набор «разновесов» подобран по двоичной системе:  $1/2$  от предельной массы, на которую рассчитаны весы,  $1/4$ ,  $1/8$ ,... На весы устанавливаем предмет, проверяем его массу наибольшей гирей ( $1/2$  шкалы весов) и в регистр записываем 1. Если гиря перевешивает, то этот старший разряд придется с весов снять (записать в старший разряд регистра 0). Ес-

ли предмет перевешивает, гирию на чашке оставляем (оставляем в регистре 1). Далее добавляем гирию в 1/4 шкалы. После анализа «больше — меньше» (в электронной схеме это делает компаратор), либо записываем 1 (гирию оставляем), либо 0 (гирию снимаем) и продолжаем далее до младшего разряда.

Рассмотрим алгоритм работы для 4-разрядного регистра. Перед началом работы в старший разряд регистра записывается единица, а в остальные разряды нуль. При этом в регистре будет зафиксирован код числа 8. В первом такте в старший разряд регистра заносится значение входного сигнала  $DI$ , а содержимое остальных разрядов сдвигается вправо в сторону младших разрядов. Так как сигнал  $DI$  может быть равен «0» или «1», то в регистре будет сформирован код числа 12 (при  $DI=1$ ) или 4 (при  $DI=0$ ). В следующем такте содержимое старшего разряда  $Q3$  сохраняется, сигнал  $DI$  заносится в разряд  $Q2$ , а данные в младших разрядах сдвигаются вправо. На рис. 92 приведены переходы состояний регистра.

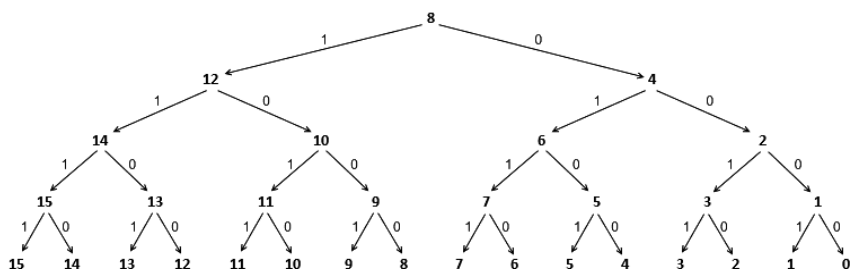


Рис. 92. Переходы состояний регистра последовательного приближения

В регистре последовательного приближения в отличие от регистра сдвига входные данные  $DI$  не перемещаются по разрядам регистра, а заносятся и фиксируются в соответствующих разрядах. В зависимости от входных значений  $DI$  на выходах регистра может быть сформирован код любого числа в пределах от 0 до 15.

Существуют различные варианты построения регистров последовательного приближения, которые также входят в состав библиотечных и серийных элементов. На рис. 93 приведена схема, построенная на JK-триггерах Т0–Т3 с асинхронными входами предварительной установки и входом  $CE$  разрешения синхросигнала  $C$ .



Сигнал с входа  $DI$  поступает на  $K$ -входы всех триггеров. В начальный момент времени элементы AND разрешают приём во все триггеры  $T0$ – $T3$ . При этом на входах  $J0$  и  $J1$  триггеров  $T0$  и  $T1$  установлено значение нуля, а вход  $J2=1$ . Поэтому по первому синхроимпульсу  $C$ , независимо от значения сигнала  $DI$ , поданного на  $K$ -входы триггеров  $T0$ – $T2$ , триггеры  $T0$  и  $T1$  останутся в нулевом состоянии, а триггер  $T2$  перейдёт в единичное состояние. Одновременно триггер  $T3$ , установленный в единичное состояние, перейдёт в нулевое состояние или останется в единичном состоянии. Таким образом единица из триггера  $T3$  запишется в триггер  $T2$ .

Далее разрешается приём только в триггеры  $T2$ – $T0$ . С приходом второго синхроимпульса  $C$  сигнал со входа  $DI$  запишется в разряд  $T2$ , и одновременно выполнится сдвиг данных остальных разрядов вправо. На выходе триггера  $T1$  будет сформирован единичный сигнал.

По третьему синхроимпульсу  $C$  сигнал с входа  $DI$  заносится в триггер  $T1$  и одновременно единица перемещается в триггер  $T0$ . По четвёртому синхроимпульсу  $C$  в триггерах будет установлено состояние, соответствующее входной комбинации сигналов  $DI$ . Одновременно триггер управления  $T4$  переключится в нулевое состояние, запрещая работу триггеров по следующим синхросигналам. При этом на выходе  $CO$  установится единичный сигнал. Следующий цикл работы регистра начнется после подаче сигнала установки  $CLR$ .

Регистры последовательного приближения находят применение при построении цифровых устройств для реализации ряда математических операций, в частности, деления чисел, извлечения корня и т.п.

### ***Счётчики на регистрах сдвига с линейными обратными связями***

**Общая схема регистра сдвига с линейными обратными связями (РСЛОС).** Исходная информация для построения РСЛОС – так называемый характеристический многочлен, определяющий разрядность регистра и характер обратных связей. Общая схема регистра РСЛОС, соответствующего двоичному характеристическому многочлену степени  $n$ :

$$\Phi(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + \dots + a_2 x^2 + a_1 x + a_0 ,$$

где  $a_n = a_0 = 1$ ,  $a_i \in \{0, 1\}$ ,  $i = 1, \overbrace{1, \dots, n-1}$ , и имеет вид, показанный на рис.94. (Прямоугольниками обозначены D-триггеры, кружками с крестом – схемы сложения по модулю 2) Характер обратных связей РСЛОС определяется ненулевыми коэффициентами  $\Phi(x)$ . Если в качестве  $\Phi(x)$  выбран примитивный многочлен, то число состояний РСЛОС будет максимально возможным и равным  $2^n - 1$ . Диаграмма состояний РСЛОС в этом случае состоит из цикла длиной  $2^n - 1$  и одного тривиального цикла, включающего только состояние «все нули».

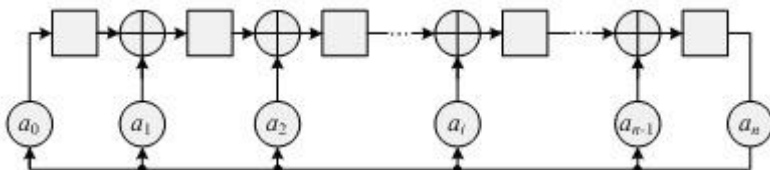


Рис. 94. Общая схема РСЛОС. Начальное состояние – любое ненулевое

**Счётчики по модулю  $2^n - 1$ .** Для построения счётчика по модулю  $2^n - 1$  выбираем в качестве характеристического примитивный многочлен степени  $n$ . Соответствующий ему РСЛОС и будет являться искомым счётчиком.

*Пример 1.* Для построения счётчика по модулю  $2^3 - 1 = 7$  выбираем примитивный многочлен третьей степени  $\Phi(x) = x^3 + x^2 + 1$ . Соответствующий ему РСЛОС (суть счётчик по модулю 7) имеет вид, показанный на рис. 95.

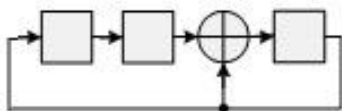


Рис. 95. Функциональная схема счётчика по модулю 7.  
Начальное состояние – любое ненулевое

**Счётчики по модулю  $2^n - 2$ .** Для построения счётчика по модулю  $2^n - 2$  выбираем в качестве характеристического многочлен вида  $\Phi(x) = (x + 1)\phi(x)$ , где  $\phi(x)$  – примитивный многочлен степени  $n - 1$ . Соответствующий ему регистр (рис. 96) и будет являться искомым счётчиком.

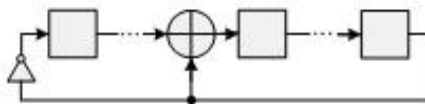


Рис. 96. Функциональная схема РСЛОС с числом состояний  $2^n-2$  или  $2^n-4$

*Пример 2.* Для построения счётчика по модулю  $2^3-3=6$  выбираем характеристический многочлен вида  $\Phi(x) = (x+1)(x^2+x+1) = x^3+1$ . Соответствующий ему регистр (суть счётчик по модулю 6) имеет вид, показанный на рис. 97.

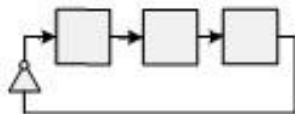


Рис. 97. Функциональная схема счётчика по модулю 6

**Счётчики по модулю  $2^n-4$ .** Для построения счётчика по модулю  $2^n-4$  выбираем в качестве характеристического многочлен вида  $\Phi(x) = (x+1)^2 \varphi(x)$ , где  $\varphi(x)$  – примитивный многочлен степени  $n-2$ . Соответствующий ему регистр и будет являться искомым счётчиком.

*Пример 3.* Для построения счётчика по модулю  $2^4-4=12$  выбираем характеристический многочлен вида  $\Phi(x) = (x+1)^2(x^2+x+1) = x^4+x^3+x+1$ . Соответствующий ему регистр (суть счётчик по модулю 12) имеет вид, показанный на рис. 98.

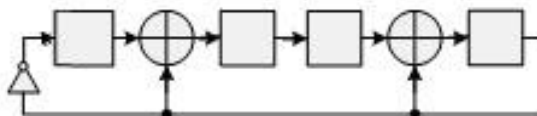


Рис. 98. Функциональная схема счётчика по модулю 12

**Счётчики по модулю  $2^n$ .** Для построения счётчика по модулю  $2^n$  выбираем в качестве характеристического примитивный многочлен степени  $n$ . Исключаем у соответствующего ему РСЛОС запрещённое ранее нулевое состояние и получаем в результате искомым счётчик.



## 2. ЗАДАЧИ НА ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ СХЕМ

### 2.1. Проектирование и анализ комбинационных схем

#### 2.1.1. Синтез комбинационных схем

Для каждого варианта дополнительным условием может быть требование реализации схемы в заданном элементном базисе:

- 2И-НЕ,
- 2ИЛИ-НЕ
- 2ИЛИ, 2И-НЕ,
- 2ИЛИ, 2И, НЕ,
- И-НЕ,
- ИЛИ-НЕ,
- на любых элементах.

**Задача 1.** Разработать схему, реализующую функцию, заданную вариантом:

| Вариант | Функция                                                                                                                   |
|---------|---------------------------------------------------------------------------------------------------------------------------|
| 1       | $F(X_2, X_1, X_0) = \Sigma(0, 1, 2, 5, 7)$                                                                                |
| 2       | $F(X_3, X_2, X_1, X_0) = \Sigma(1, 3, 4, 6, 7, 12, 13, 15)$                                                               |
| 3       | $F(X_2, X_1, X_0) = \Sigma(0, 1, 2, 3, 4, 5, 6)$                                                                          |
| 4       | $F(X_3, X_2, X_1, X_0) = \Sigma(2, 8, 10, 12, 13, 14)$                                                                    |
| 5       | $F(X_3, X_2, X_1, X_0) = \Sigma(8, 9, 10, 11, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 31)$                        |
| 6       | $F(X_3, X_2, X_1, X_0) = X_1 \cdot \neg X_3 \cdot X_0 + \neg X_2 \cdot X_3 \cdot (\neg X_0 \cdot \neg X_1 + X_0)$         |
| 7       | $F(X_3, X_2, X_1, X_0) = X_0 \cdot X_2 \cdot X_3 + \neg X_2 \cdot X_1 \cdot (\neg X_0 + X_3)$                             |
| 8       | $F(X_2, X_1, X_0) = X_0 \cdot \neg X_1 + \neg X_0 \cdot X_2 + X_0 \cdot \neg X_1 \cdot X_2$                               |
| 9       | $F(X_2, X_1, X_0) = X_0 \cdot \neg X_1 + X_0 \cdot \neg X_2 + X_0 \cdot X_1 \cdot X_2$                                    |
| 10      | $F(X_2, X_1, X_0) = X_0 \cdot \neg X_1 + \neg X_0 \cdot \neg X_2 + X_0 \cdot \neg X_1 \cdot \neg X_2$                     |
| 11      | $F(X_2, X_1, X_0) = \neg X_0 \cdot \neg X_1 + \neg X_0 \cdot X_2 + X_0 \cdot \neg X_1 \cdot X_2$                          |
| 12      | $F(X_2, X_1, X_0) = \neg X_0 \cdot X_1 + X_1 \cdot \neg X_2 + X_0 \cdot \neg X_1 \cdot X_2 + X_0 \cdot (X_1 + \neg X_2)$  |
| 13      | $F(X_2, X_1, X_0) = X_0 \cdot \neg X_1 + X_1 \cdot \neg X_2 + X_0 \cdot X_1 \cdot \neg X_2 + X_0 \cdot (\neg X_1 + X_2)$  |
| 14      | $F(X_2, X_1, X_0) = X_0 \cdot \neg X_2 + X_1 \cdot \neg X_2 + X_0 \cdot X_1 \cdot \neg X_2 + X_0 \cdot (X_1 + \neg X_2)$  |
| 15      | $F(X_2, X_1, X_0) = \neg X_0 \cdot X_1 \cdot \neg X_2 + X_0 \cdot \neg X_1 \cdot X_2 + \neg X_0 \cdot \neg X_1 \cdot X_2$ |

Примечание. В вариантах 1-5 и далее функции заданы в виде десятичных номеров конститuent совершенной дизъюнктивной нормальной формы, т.е. тех наборов, на которых функции равны единице. В вариантах 6-15 и далее символ «» означает операцию И, символ «+» – операцию ИЛИ, символ «¬» – инверсию.

**Задача 2.** Разработать схему, реализующую функции  $F1(X3,X2,X1,X0)$ ,  $F2(X3,X2,X1,X0)$  и  $F3(X3,X2,X1,X0)$ , заданные вариантом:

| Вариант 1.                              | Вариант 2.                                 |
|-----------------------------------------|--------------------------------------------|
| $F1 = \Sigma (0,2,6,7, 12,13,14,15)$    | $F1 = \Sigma (0, 2, 4, 5, 11, 12, 13, 15)$ |
| $F2 = \Sigma (0, 2, 4, 12, 13, 14, 15)$ | $F2 = \Sigma (2, 3, 4, 5, 12, 13, 15)$     |
| $F3 = \Sigma (4, 5, 12, 13, 14, 15)$    | $F3 = \Sigma (2, 3, 4, 5, 12, 13)$         |

| Вариант 3.                                 | Вариант 4.                                |
|--------------------------------------------|-------------------------------------------|
| $F1 = \Sigma (2, 4, 5, 9, 10, 11, 13, 15)$ | $F1 = \Sigma (0, 4, 5, 8, 9, 10, 11, 14)$ |
| $F2 = \Sigma (0, 4, 5, 9, 11, 13, 15)$     | $F2 = \Sigma (0, 6, 7, 8, 9, 10, 11)$     |
| $F3 = \Sigma (0, 2, 9, 11, 13, 15)$        | $F3 = \Sigma (0, 6, 8, 9, 10, 11)$        |

| Вариант 5.                               | Вариант 6.                                |
|------------------------------------------|-------------------------------------------|
| $F1 = \Sigma (0, 1, 2, 3, 6, 8, 13, 15)$ | $F1 = \Sigma (0, 1, 6, 8, 9, 11, 14, 15)$ |
| $F2 = \Sigma (0, 1, 2, 3, 6, 8, 15)$     | $F2 = \Sigma (0, 1, 6, 8, 9, 11, 15)$     |
| $F3 = \Sigma (0, 1, 2, 3, 13, 15)$       | $F3 = \Sigma (0, 1, 6, 8, 9, 14)$         |

| Вариант 7.                                 | Вариант 8.                                 |
|--------------------------------------------|--------------------------------------------|
| $F1 = \Sigma (0, 4, 6, 9, 10, 11, 12, 14)$ | $F1 = \Sigma (1, 4, 5, 9, 10, 11, 14, 15)$ |
| $F2 = \Sigma (4, 5, 6, 10, 11, 12, 14)$    | $F2 = \Sigma (1, 5, 7, 10, 11, 14, 15)$    |
| $F3 = \Sigma (4, 6, 9, 11, 12, 14)$        | $F3 = \Sigma (4, 8, 10, 11, 14, 15)$       |

| Вариант 9.                            | Вариант 10.                            |
|---------------------------------------|----------------------------------------|
| $F1 = \Sigma (0, 1, 3, 5, 7, 12, 13)$ | $F1 = \Sigma (1, 2, 6, 7, 11, 12, 13)$ |
| $F2 = \Sigma (6, 7, 8, 9, 10, 11)$    | $F2 = \Sigma (10, 11, 12, 13, 14, 15)$ |
| $F3 = \Sigma (2, 4, 6, 7, 8)$         | $F3 = \Sigma (0, 1, 2, 3, 4)$          |

| Вариант 11.                            | Вариант 12.                          |
|----------------------------------------|--------------------------------------|
| $F1 = \Sigma (2, 3, 5, 8, 10, 13, 14)$ | $F1 = \Sigma (3, 4, 5, 6, 7, 8, 9)$  |
| $F2 = \Sigma (9, 10, 11, 12, 13, 14)$  | $F2 = \Sigma (8, 9, 10, 11, 12, 13)$ |
| $F3 = \Sigma (1, 2, 3, 4, 5)$          | $F3 = \Sigma (2, 3, 4, 5, 6)$        |

| Вариант 13.                          | Вариант 14.                           |
|--------------------------------------|---------------------------------------|
| $F1 = \Sigma (4, 5, 6, 7, 8, 9, 10)$ | $F1 = \Sigma (5, 6, 7, 8, 9, 10, 11)$ |
| $F2 = \Sigma (7, 8, 9, 10, 11, 12)$  | $F2 = \Sigma (6, 7, 8, 9, 10, 11)$    |
| $F3 = \Sigma (3, 4, 5, 6, 7)$        | $F3 = \Sigma (4, 5, 6, 7, 8)$         |

| Вариант 15.                            |
|----------------------------------------|
| $F1 = \Sigma (6, 7, 8, 9, 10, 11, 12)$ |
| $F2 = \Sigma (5, 6, 7, 8, 9, 10)$      |
| $F3 = \Sigma (5, 6, 7, 8, 9)$          |

### ***2.1.2. Синтез комбинационных схем при словесном описании их функций***

Для каждого варианта дополнительным условием может быть требование реализации схемы в заданном элементном базисе:

- 2И-НЕ,
- 2ИЛИ-НЕ
- 2ИЛИ, 2И-НЕ,
- 2ИЛИ, 2И, НЕ,
- И-НЕ,
- ИЛИ-НЕ,
- на любых элементах.

**Задача 1.** Разработать комбинационную схему, заданную вариантом.

Вариант 1. Четырёхразрядная схема сложения по mod 2.

Вариант 2. Схема для получения признака чётности числа единиц в 4-разрядном двоичном коде («0» – чётно, «1» – нечётно).

Вариант 3. Схема, реализующая функцию равнозначности (эквивалентности) трёх аргументов.

Вариант 4. Схема, реализующая функцию нечётности числа единиц в 3-разрядном двоичном слове.

Вариант 5. Схема, реализующая функцию нечётности числа нулей в 3-разрядном двоичном слове.

Вариант 6. Схема, реализующая функцию равнозначности четырёх аргументов.

Вариант 7. Схема, реализующая функцию чётности числа единиц в 3-разрядном двоичном слове.

Вариант 8. Схема, реализующая функцию чётности числа нулей в 3-разрядном двоичном слове.

Вариант 9. Схема, обнаруживающая цифры, кратные 3, во входных данных в коде «8421».

Вариант 10. Схему, которая будет показывать, когда большинство цифр пятиразрядного двоичного числа равно «1».

Вариант 11. Схема, реализующая функцию для двухразрядных чисел  $A(A1,A0)$  меньше или равно  $B(B1,B0)$ .

Вариант 12. Схема, реализующая функцию для двухразрядных чисел  $A(A1,A0)$  меньше  $B(B1,B0)$ .

Вариант 13. Схема, реализующая функцию для двухразрядных чисел  $A(A1,A0)$  больше  $B(B1,B0)$ .

Вариант 14. Схема, реализующая функцию для двухразрядных чисел  $A(A1,A0)$  больше или равно  $B(B1,B0)$ .

Вариант 15. Схема, реализующая функцию голосования «2 из 3».

**Задача 2.** Разработать комбинационную схему, заданную вариантом.

Вариант 1. Схема сравнения двух двухразрядных двоичных чисел  $A$  и  $B$ . Схема должна показывать, является ли: 1)  $A > B$ ; 2)  $A = B$ ; 3)  $A < B$ .

Вариант 2. Схема определения количества единиц (двоичное число) в 4-разрядном двоичном коде.

Вариант 3. Схема определения количества нулей (двоичное число) в 4-разрядном двоичном коде.

Вариант 4. Схема деления по mod 5 4-разрядного двоичного числа.

Вариант 5. Приоритетная схема формирования номера запроса прерывания. Входные сигналы – четыре запроса на прерывание:  $INT4$ ,  $INT3$ ,  $INT2$ ,  $INT1$ . Высший приоритет у запроса с меньшим номером.

Вариант 6. Схема получения остатка по mod 3 от 4-разрядного двоичного числа.

Вариант 7. Схема получения целой части (двоичное число) корня квадратного от 4-разрядного двоичного числа.

Вариант 8. Схема сложения по mod 3 двух двухразрядных двоичных чисел.

Вариант 9. Схема получения произведения двух двухразрядных чисел без знаков.

Вариант 10. Схема получения квадратов от всех возможных значений трёхразрядного двоичного числа.

Вариант 11. Схема определения результатов судейства. Входные сигналы  $X_3$ ,  $X_2$ ,  $X_1$ ,  $X_0$  – оценки результатов «да», «нет» от четырёх судей. При равенстве результатов «да» и «нет» решение принимается по результату главного судьи ( $X_3$ ).

Вариант 12. Схема управления охранной сигнализацией. Система сигнализации приводится в действие, если срабатывают одновременно датчик появления дыма и датчик повышения температуры или срабатывает один датчик повышения температуры, при наличии сигнала снятия электропитания с силового оборудования и сигнала наличия охлаждающего агента в системе пожаротушения. В остальных случаях система пожаротушения должна быть отключена.

Вариант 13. Схема управления запуском двигателя. Двигатель запускается при наличии масла в системе, наличии топлива и отсутствии сигнала с датчика температуры воды и при включённом тумблере «Запуск». Двигатель выключается при отсутствии масла или топлива или при повышении температуры воды. Двигатель запускается от кнопки «Аварийный запуск» при наличии топлива и независимо от состояния остальных датчиков.

Вариант 14. Схема управления холодильной установкой. Двигатель холодильной установки включается, если срабатывает первый датчик повышения температуры в камере при наличии сигнала герметизации камеры и сигнала о наличии хладагента, а также включенного состояния тумблера «Размораживание». Двигатель включается при включении тумблера «Размораживание». Двигатель включается при выключенном положении тумблера «Размораживание» и срабатывании первого и второго датчиков повышения температуры.

Вариант 15. Схема управления освещением. Блок квартир размещается на четырёх этажах. Схема должна позволять включать или выключать свет, освещающий межэтажную лестницу, на любом этаже.

**Задача 3.** Разработать комбинационную схему, заданную вариантом.

Вариант 1. Ограничитель сигналов при превышении диапазона 7. Схема имеет пять информационных входов  $D_4-D_0$  и три выхода  $Y_2-Y_0$ . Если значение входного числа менее или равно 7, то значение входов передается на выходы. При значениях больше 7 на выходах также формируется число 7 и вырабатывается флаг переполнения  $FC$ .

Вариант 2. Ограничитель сигналов, заданных в обратном коде, при превышении диапазона 7. Схема имеет шесть информационных входов  $D_5-D_0$  и четыре выхода  $Y_3-Y_0$ . Левые старшие разряды  $D_5$  и  $Y_3$  знаковые. Если значение входного числа по модулю менее или равно 7, то значение входов передается на выходы. При значениях входного числа по модулю больше 7 на выходах формируется число +7 для положительных чисел или -7 для отрицательных чисел в обратном коде и вырабатывается флаг переполнения  $FC$ .

Вариант 3. Ограничитель сигналов, заданных в дополнительном коде, при превышении диапазона 7. Схема имеет шесть информационных входов  $D_5-D_0$  и три выхода  $Y_3-Y_0$ . Левые старшие разряды  $D_5$  и  $Y_3$  знаковые. Если значение входного числа по модулю менее или равно 7, то значение входов передается на выходы. При значениях входного числа больше 7 на выходах формируется число +7 для положительных чисел, а при значениях входного числа меньше -8 для отрицательных чисел в дополнительном коде формируется число -8 и вырабатывается флаг переполнения  $FC$ .

Вариант 4. Схема подсчёта количества левых старших единиц в 8-разрядном входном числе  $X_7-X_0$ . На выходе формируется четырёхразрядный двоичный код  $Y_3-Y_0$ .

Вариант 5. Схема подсчёта количества левых старших нулей в 8-разрядном входном числе  $X_7-X_0$ . На выходе формируется четырёхразрядный двоичный код  $Y_3-Y_0$ .

Вариант 6. Схема для нормализации чисел. На вход поступает 8-разрядное входное число  $X7-X0$ , заданное в обратном коде (разряд  $X7$  знаковый). Схема должна выполнить подсчёт количества левых старших нулей или единиц в зависимости от знака и на выходе сформировать трёхразрядный двоичный код  $Y2-Y0$ . Если нормализации не требуется, то формируется флаг  $FN$ .

Вариант 7. Схема для нормализации чисел. На вход поступает 8-разрядное входное число  $X7-X0$ , заданное в дополнительном коде (разряд  $X7$  знаковый). Схема должна выполнить подсчёт количества левых старших нулей или единиц в зависимости от знака и на выходе сформировать трёхразрядный двоичный код  $Y2-Y0$ . Если нормализации не требуется, то формируется флаг  $FN$ .

Вариант 8. Схема для определения номера левой старшей единицы в 8-разрядном входном числе  $X7-X0$ . На выходе формируется четырёхразрядный двоичный код  $Y3-Y0$ .

Вариант 9. Схема для определения номера левого старшего нуля в 8-разрядном входном числе  $X7-X0$ . На выходе формируется четырёхразрядный двоичный код  $Y3-Y0$ .

Вариант 10. Схема подсчёта количества единиц в 8-разрядном входном числе  $X7-X0$ . На выходе формируется четырёхразрядный двоичный код  $Y3-Y0$ .

Вариант 11. Схема подсчёта количества нулей в 8-разрядном входном числе  $X7-X0$ . На выходе формируется четырёхразрядный двоичный код  $Y3-Y0$ .

Вариант 12. Схема подсчёта количества единиц (нулей) в 8-разрядном входном числе  $X7-X0$ . На выходе формируется четырёхразрядный двоичный код  $Y3-Y0$ . Счёт единиц проводится при входном сигнале  $K=1$ , счёт нулей – при входном сигнале  $K=0$ .

Вариант 13. Схема двухпорогового компаратора для проверки соответствия заданному диапазону входного трёхразрядного двоичного числа  $A(A2, A1, A0)$ . Границы диапазона задаются трёхразрядными двоичными кодами:  $YM$  – верхний порог,  $YL$  – нижний порог. На выходе формируются сигналы:  $G$  – больше,  $E$  – равно,  $L$  – меньше.

Вариант 14. Схема, которая в зависимости от значения управляющего сигнала  $Y$  выполняет функции инкремента ( $Y=0$ ) или декремента ( $Y=1$ ) трёхразрядных двоичных чисел. Схема должна вырабатывать сигналы переноса и заёма.

Вариант 15. Схема, которая в зависимости от значения управляющего сигнала  $Y$  выполняет функции инкрементора ( $Y=0$ ) или декрементора ( $Y=1$ ) трёхразрядных двоичных чисел.

**Задача 4.** Разработать комбинационную схему, определенную вариантом.

Вариант 1. Схема уплотнения (сдвига) единиц вправо в восьмиразрядном входном числе  $X_7-X_0$ . На выходе формируется восьмиразрядный двоичный код  $Y_7-Y_0$ . Например, для числа 01101001 должно быть получено 00001111.

Вариант 2. Схема уплотнения (сдвига) единиц влево в восьмиразрядном входном числе  $X_7-X_0$ . На выходе формируется восьмиразрядный двоичный код  $Y_7-Y_0$ . Например, для числа 01001001 должно быть получено 11100000.

Вариант 3. Схема уплотнения (сдвига) нулей влево в восьмиразрядном входном числе  $X_7-X_0$ . На выходе формируется восьмиразрядный двоичный код  $Y_7-Y_0$ . Например, для числа 01101101 должно быть получено 00011111.

Вариант 4. Схема уплотнения (сдвига) нулей вправо в восьмиразрядном входном числе  $X_7-X_0$ . На выходе формируется восьмиразрядный двоичный код  $Y_7-Y_0$ . Например, для числа 01001001 должно быть получено 11100000.

Вариант 5. Схема преобразования позиционного трёхразрядного двоичного кода  $X_2-X_0$  в унитарный позиционный ряд (число единиц) длиной равной входному коду. Например, для входного кода 011 формируется ряд 0000111. Схема имеет семь выходов  $Y_6-Y_0$ .

Вариант 6. Схема преобразования позиционного трёхразрядного двоичного кода  $X_2-X_0$  в инверсный унитарный позиционный ряд (число нулей) длиной, равной входному коду. Например, для входного кода 011 формируется ряд 1111000. Схема имеет семь выходов  $Y_6-Y_0$ .

Вариант 7. Схема преобразования позиционного трёхразрядного двоичного числа  $X_2-X_0$ , заданного в обратном коде, в двухсторонний унитарный позиционный ряд (число единиц) длиной, равной входному коду. В зависимости от знака унитарный ряд формируется от середины влево для отрицательных чисел или вправо для положительных чисел. Разряд  $X_2$  знаковый. Схема имеет шесть выхо-

дов  $Y_5$ – $Y_0$ . Например, для входного кода 0.10 формируется ряд 000110, а для кода 1.01 формируется ряд 011000.

Вариант 8. Схема преобразования позиционного трёхразрядного двоичного числа  $X_2$ – $X_0$ , заданного в дополнительном коде, в двухсторонний унитарный позиционный ряд (число единиц) длиной равной входному коду. В зависимости от знака унитарный ряд формируется от середины влево для отрицательных чисел или вправо для положительных чисел. Разряд  $X_2$  знаковый. Схема имеет семь выходов  $Y_6$ – $Y_0$ . Например, для входного кода 0.10 формируется ряд 0000110, а для кода 1.10 формируется ряд 0011000.

Вариант 9. Для варианта 7 добавляется возможность наращивания разрядности схемы.

Вариант 10. Для варианта 8 добавляется возможность наращивания разрядности схемы.

Вариант 11. Схема сложения двух чисел  $A$  и  $B$ , заданных в унитарном коде «1 из 3». Примеры задания чисел: число ноль – 001, один – 001, два – 100. На выходе результат  $C$  также формируется в унитарном коде «1 из 3» и, кроме того, формируется перенос  $CO$ .

Вариант 12. Схема сложения двух чисел  $A$  и  $B$ , заданных в унитарном коде «1 из 4». Примеры задания чисел: число ноль – 0001, один – 0010, два – 0100, три – 1000. На выходе результат  $C$  также формируется в унитарном коде «1 из 4» и, кроме того, формируется перенос  $CO$ .

Вариант 13. Схема умножения двух чисел  $A$  и  $B$ , заданных в унитарном коде «1 из 3». Примеры задания чисел: число ноль – 001, один – 010, два – 100. На выходе результат  $C$  также формируется в унитарном коде «1 из 3» и, кроме того, формируются переносы:  $P_2$  – двойной перенос и  $P_1$  – одинарный перенос.

Вариант 14. Схема сложения двух чисел  $A$  и  $B$ , заданных в унитарном коде «1 из 4». Примеры задания чисел: число ноль – 0001, один – 0010, два – 0100, три – 1000. На выходе результат  $C$  также формируется в унитарном коде «1 из 4» и, кроме того, формируются переносы:  $P_2$  – двойной перенос и  $P_1$  – одинарный перенос.

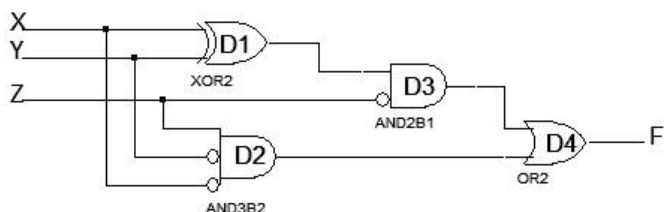
Вариант 15. Схема преобразования двоично-десятичных чисел, представленных в коде «3321» в числа, представленные в коде «2521».

### 2.1.3. Построение временной диаграммы работы комбинационной схемы с учётом задержек элементов

**Задача 1.** Построить временную диаграмму работы схемы при заданной последовательности изменений сигналов  $Z, Y, X$  на входах. Определить задержку переключения сигнала  $F$  при соотношении задержек каждого элемента  $t_3^{01} = 2t_3^{10}$ .

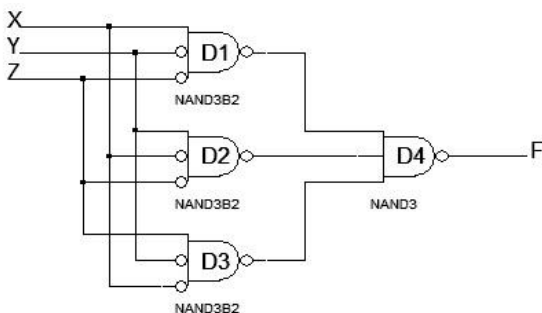
Вариант 1.

$Z, Y, X =$   
 $= 2, 3, 4, 5, 1.$



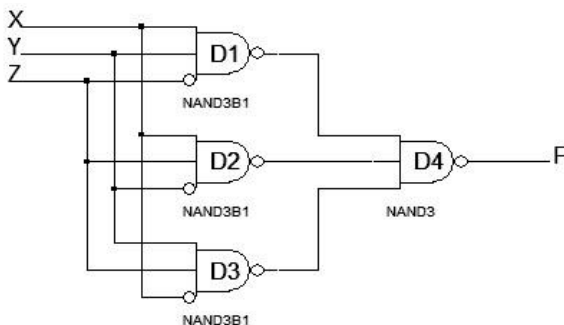
Вариант 2.

$Z, Y, X =$   
 $= 0, 1, 2, 4, 2.$



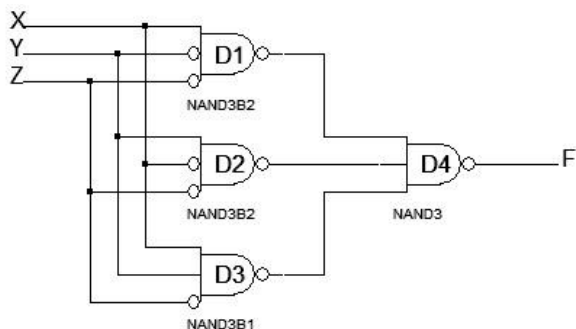
Вариант 3.

$Z, Y, X =$   
 $= 4, 5, 6, 0, 3.$



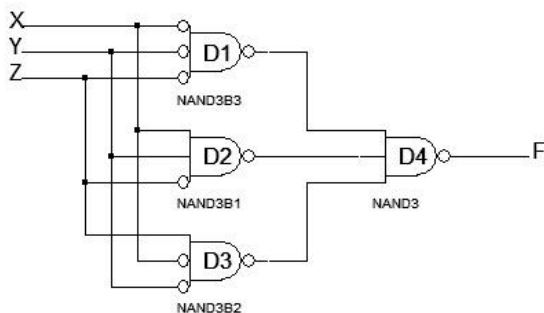
Вариант 4.

$Z,Y,X =$   
 $= 1, 0, 2, 3, 4.$



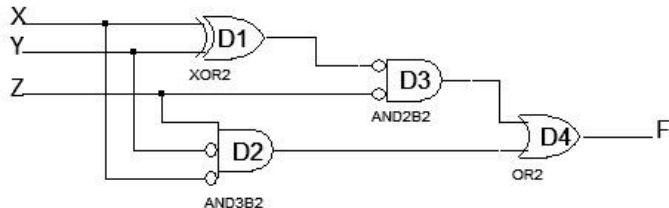
Вариант 5.

$Z,Y,X =$   
 $= 0, 4, 2, 3, 1.$



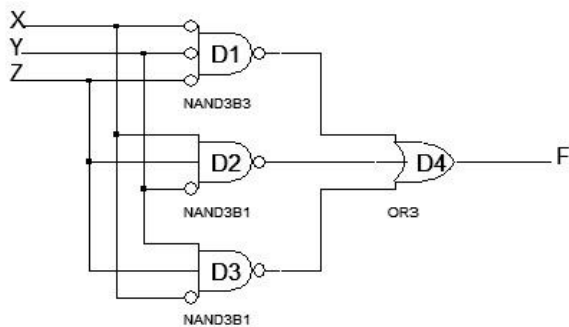
Вариант 6.

$Z,Y,X =$   
 $= 0, 1, 2, 3, 4,$   
 $5.$



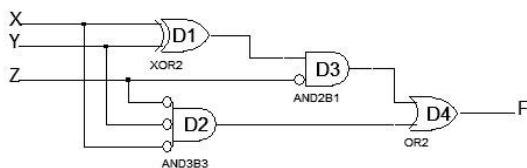
Вариант 7.

$Z,Y,X =$   
 $= 5, 0, 2, 4, 3.$



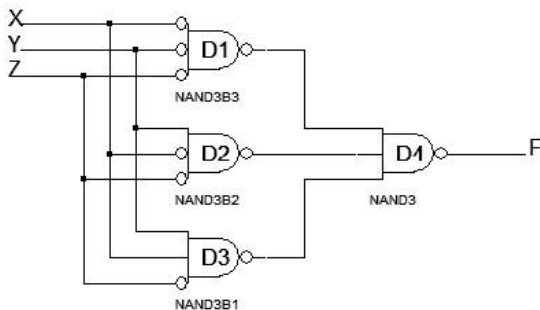
Вариант 8.

$Z,Y,X =$   
 $= 4, 1, 2, 3, 0.$



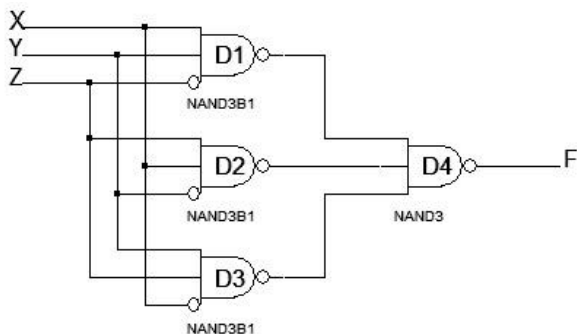
Вариант 9.

$Z,Y,X =$   
 $= 0, 4, 2, 1, 3.$



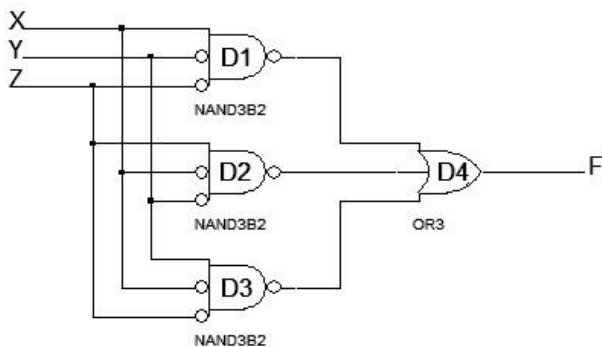
Вариант 10.

$Z,Y,X =$   
 $= 3, 4, 0, 1, 2.$



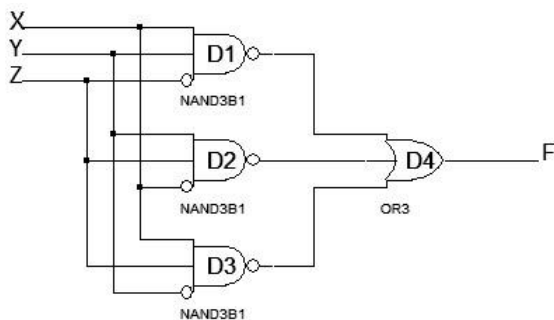
Вариант 11.

$Z,Y,X =$   
 $= 6, 4, 5, 3, 4.$



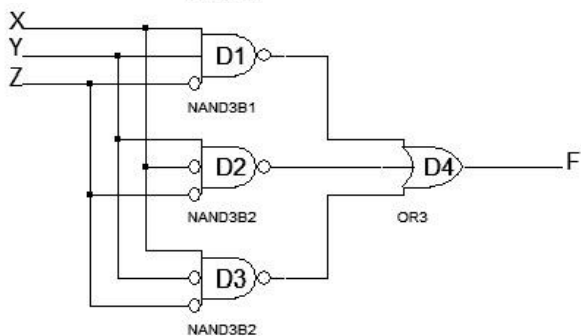
Вариант 12.

$Z,Y,X =$   
 $= 0, 1, 4, 2, 3.$



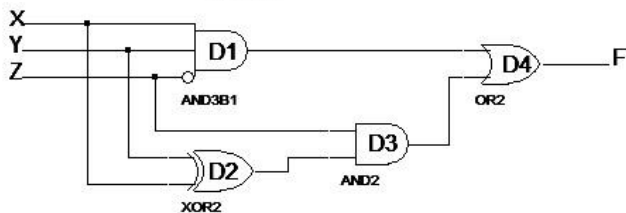
Вариант 13.

$Z,Y,X =$   
 $= 0, 3, 1, 4, 2.$



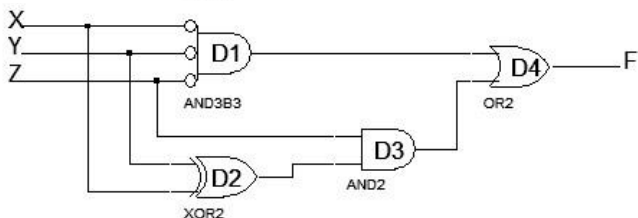
Вариант 14.

$Z,Y,X =$   
 $= 3, 4, 5, 6, 7.$



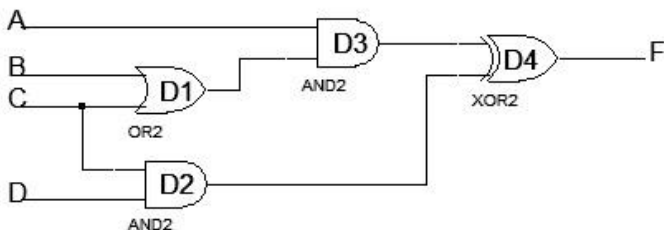
Вариант 15.

$Z,Y,X =$   
 $= 4, 5, 6, 7, 0.$

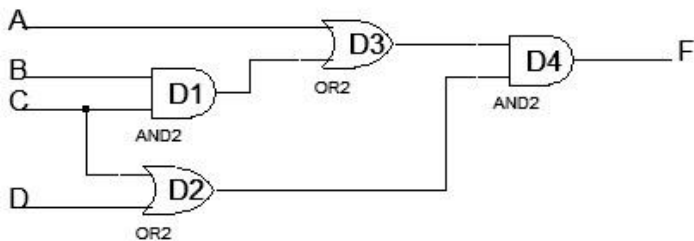


**Задача 2.** Построить временную диаграмму работы схемы при заданной последовательности изменений сигналов  $A, B, C, D$  на входах. Определить задержку переключения выходного сигнала при заданных задержках элементов.

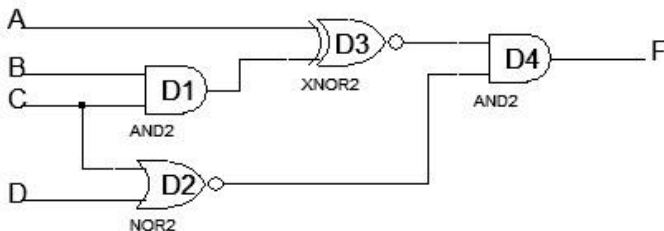
Вариант 1.  $A, B, C, D = 1000, 1010$ . Задержка  $D1, D4 - 6$  нс, задержка  $D2, D3 - 3$  нс.



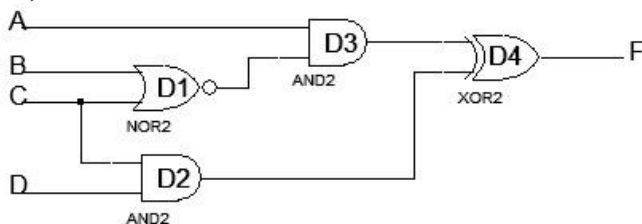
Вариант 2.  $A, B, C, D = 0110, 0100$ . Задержка  $D1, D4 - 10$  нс, задержка  $D2, D3 - 20$  нс.



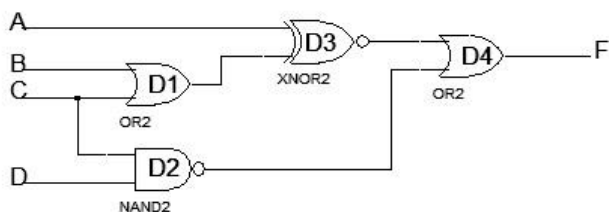
Вариант 3.  $A, B, C, D = 1111, 1101$ . Задержка  $D1, D4 - 8$  нс, задержка  $D2, D3 - 4$  нс.



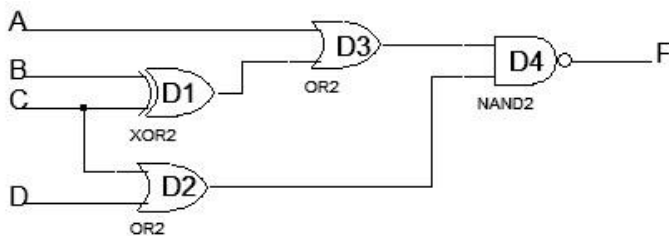
Вариант 4.  $A, B, C, D = 1011, 1001$ . Задержка  $D2, D3 - 6$  нс, задержка  $D1, D4 - 12$  нс.



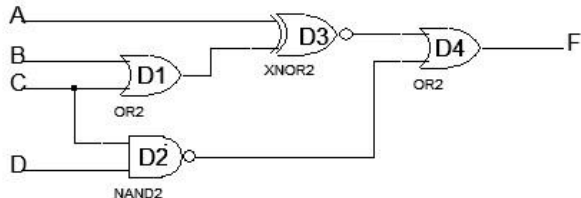
Вариант 5.  $A, B, C, D = 0001, 0011$ . Задержка  $D1, D4 - 9$  нс, задержка  $D2, D3 - 18$  нс.



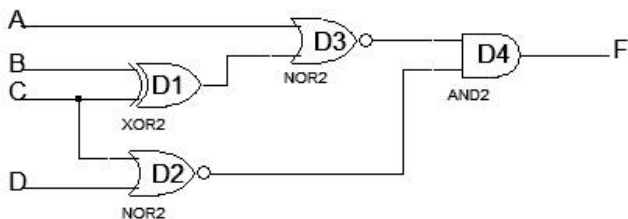
Вариант 6.  $A, B, C, D = 0110, 0100$ . Задержка  $D1, D4 - 26$  нс, задержка  $D2, D3 - 13$  нс.



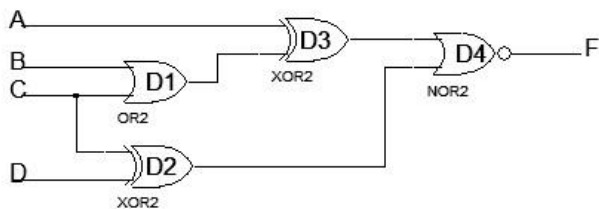
Вариант 7.  $A, B, C, D = 1011, 1011$ . Задержка  $D2, D3 - 11$  нс, задержка  $D1, D4 - 22$  нс.



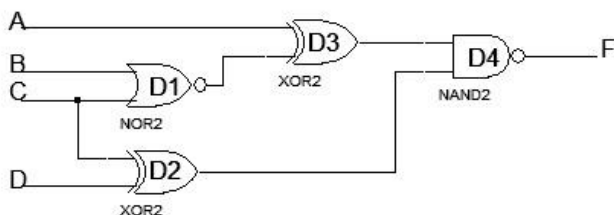
Вариант 8.  $A, B, C, D = 0010, 0000$ . Задержка  $D2, D3 - 14$  нс, задержка  $D1, D4 - 7$  нс.



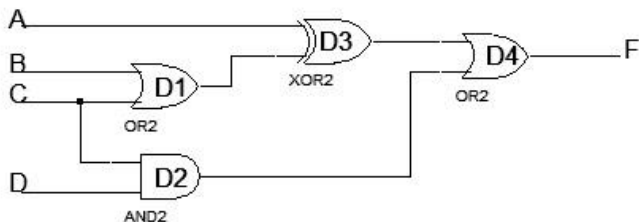
Вариант 9.  $A, B, C, D = 0000, 0010$ . Задержка  $D1, D4 - 5$  нс, задержка  $D2, D3 - 10$  нс.



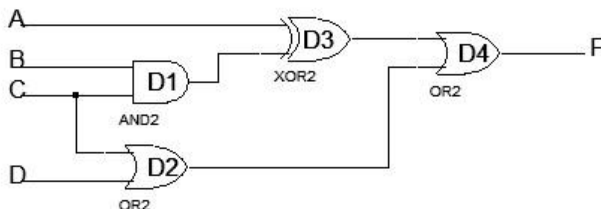
Вариант 10.  $A, B, C, D = 1010, 1000$ . Задержка  $D1, D2 - 2$  нс, задержка  $D3, D4 - 4$  нс.



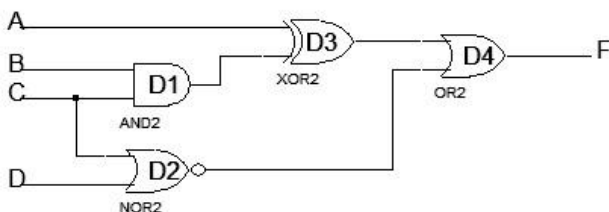
Вариант 11.  $A, B, C, D = 1011, 1001$ . Задержка  $D1, D4 - 16$  нс, задержка  $D2, D3 - 8$  нс.



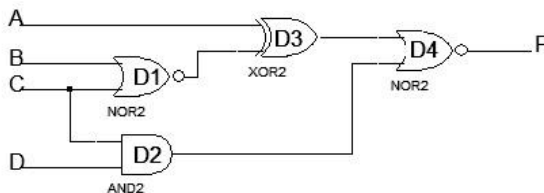
Вариант 12.  $A, B, C, D = 0110, 0100$ . Задержка  $D1, D2 - 7$  нс, задержка  $D3, D4 - 14$  нс.



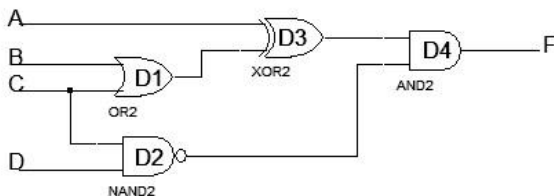
Вариант 13.  $A, B, C, D = 0100, 0110$ . Задержка  $D1, D3 - 8$  нс, задержка  $D2, D4 - 4$  нс.



Вариант 14.  $A, B, C, D = 1011, 1001$ . Задержка  $D1, D4 - 8$  нс, задержка  $D2, D3 - 16$  нс.



Вариант 15.  $A, B, C, D = 1011, 1001$ . Задержка  $D1, D2 - 9$  нс, задержка  $D3, D4 - 18$  нс.



### 2.1.4. Синтез комбинационных схем на основе дешифраторов

**Задача 1.** Реализовать на дешифраторах D2\_4E функцию  $F(X1, X0)$  от двух переменных (варианты 1-10):

| Вариант 1. |    |             | Вариант 2. |    |             |
|------------|----|-------------|------------|----|-------------|
| X1         | X0 | $F(X1, X0)$ | X1         | X0 | $F(X1, X0)$ |
| 0          | 0  | 1           | 0          | 0  | 1           |
| 0          | 1  | 0           | 0          | 1  | 1           |
| 1          | 0  | 0           | 1          | 0  | 0           |
| 1          | 1  | 1           | 1          | 1  | 0           |

| Вариант 3. |    |             | Вариант 4. |    |             |
|------------|----|-------------|------------|----|-------------|
| X1         | X0 | $F(X1, X0)$ | X1         | X0 | $F(X1, X0)$ |
| 0          | 0  | 1           | 0          | 0  | 0           |
| 0          | 1  | 0           | 0          | 1  | 1           |
| 1          | 0  | 1           | 1          | 0  | 1           |
| 1          | 1  | 0           | 1          | 1  | 0           |

| Вариант 5. |    |             | Вариант 6. |    |             |
|------------|----|-------------|------------|----|-------------|
| X1         | X0 | $F(X1, X0)$ | X1         | X0 | $F(X1, X0)$ |
| 0          | 0  | 0           | 0          | 0  | 0           |
| 0          | 1  | 1           | 0          | 1  | 0           |
| 1          | 0  | 0           | 1          | 0  | 1           |
| 1          | 1  | 1           | 1          | 1  | 1           |

| Вариант 7. |    |             | Вариант 8. |    |             |
|------------|----|-------------|------------|----|-------------|
| X1         | X0 | $F(X1, X0)$ | X1         | X0 | $F(X1, X0)$ |
| 0          | 0  | 1           | 0          | 0  | 1           |
| 0          | 1  | 1           | 0          | 1  | 1           |
| 1          | 0  | 0           | 1          | 0  | 1           |
| 1          | 1  | 1           | 1          | 1  | 0           |

| Вариант 9. |    |             | Вариант 10. |    |             |
|------------|----|-------------|-------------|----|-------------|
| X1         | X0 | $F(X1, X0)$ | X1          | X0 | $F(X1, X0)$ |
| 0          | 0  | 0           | 0           | 0  | 1           |
| 0          | 1  | 1           | 0           | 1  | 0           |
| 1          | 0  | 1           | 1           | 0  | 1           |
| 1          | 1  | 1           | 1           | 1  | 1           |

Реализовать на двухходовом дешифраторе с инверсными выходами функцию  $F(X1, X0)$  от двух переменных (варианты 11-15):

| Вариант 11. |    |             | Вариант 12. |    |             |
|-------------|----|-------------|-------------|----|-------------|
| X1          | X0 | $F(X1, X0)$ | X1          | X0 | $F(X1, X0)$ |
| 0           | 0  | 0           | 0           | 0  | 0           |
| 0           | 1  | 1           | 0           | 1  | 0           |
| 1           | 0  | 1           | 1           | 0  | 1           |
| 1           | 1  | 0           | 1           | 1  | 1           |

| Вариант 13. |    |             | Вариант 14. |    |             |
|-------------|----|-------------|-------------|----|-------------|
| X1          | X0 | $F(X1, X0)$ | X1          | X0 | $F(X1, X0)$ |
| 0           | 0  | 1           | 0           | 0  | 1           |
| 0           | 1  | 0           | 0           | 1  | 0           |
| 1           | 0  | 0           | 1           | 0  | 1           |
| 1           | 1  | 1           | 1           | 1  | 0           |

| Вариант 15. |    |             |
|-------------|----|-------------|
| X1          | X0 | $F(X1, X0)$ |
| 0           | 0  | 0           |
| 0           | 1  | 1           |
| 1           | 0  | 0           |
| 1           | 1  | 1           |

**Задача 2.** Реализовать с помощью дешифраторов и логических элементов И-НЕ функцию, заданную вариантом:

Вариант 1. Умножение двух двухразрядных чисел.

Вариант 2. Равнозначности (эквивалентности) четырёх аргументов.

Вариант 3. Нечётности числа «1» в 3-разрядном двоичном слове.

Вариант 4. Нечётности числа «0» в 3-разрядном двоичном слове.

Вариант 5. Голосования «2 из 3».

Вариант 6. Нечётности числа «1» в 4-разрядном двоичном слове.

Вариант 7. Нечётности числа «0» в 4-разрядном двоичном слове.

Вариант 8. Чётности числа «1» в 3-разрядном двоичном слове.

Вариант 9. Чётности числа «1» в 4-разрядном двоичном слове.

Вариант 10. Чётности числа «0» в 3-разрядном двоичном слове.

Вариант 11. Определение номера позиции, где, начиная слева, располагается первая «1» в 4-разрядном двоичном слове, отличном от 0000.

Вариант 12. Чётности числа «0» в 4-разрядном двоичном слове.

Вариант 13. Определение номера позиции, где, начиная справа, располагается первая «1» в 4-разрядном двоичном слове, отличном от 0000.

Вариант 14. Определение номера позиции, где, начиная справа, располагается первый «0» в 4-разрядном двоичном слове, отличным от 1111.

Вариант 15. Определение номера позиции, где, начиная слева, располагается первый «0» в 4-разрядном двоичном слове, отличным от 1111.

**Задача 3.** Реализовать на дешифраторах D2\_4E функцию  $F(X_2, X_1, X_0)$  от трёх переменных.

| Вариант | Функция                                 |
|---------|-----------------------------------------|
| 1       | $F(X_2, X_1, X_0) = \Sigma(2, 3, 6, 7)$ |
| 2       | $F(X_2, X_1, X_0) = \Sigma(1, 2, 3, 6)$ |
| 3       | $F(X_2, X_1, X_0) = \Sigma(0, 2, 5, 6)$ |
| 4       | $F(X_2, X_1, X_0) = \Sigma(0, 1, 3, 7)$ |
| 5       | $F(X_2, X_1, X_0) = \Sigma(0, 2, 4, 6)$ |
| 6       | $F(X_2, X_1, X_0) = \Sigma(1, 4, 5, 6)$ |
| 7       | $F(X_2, X_1, X_0) = \Sigma(2, 4, 5, 7)$ |
| 8       | $F(X_2, X_1, X_0) = \Sigma(1, 2, 4, 7)$ |
| 9       | $F(X_2, X_1, X_0) = \Sigma(3, 4, 6, 7)$ |
| 10      | $F(X_2, X_1, X_0) = \Sigma(3, 5, 6, 7)$ |
| 11      | $F(X_2, X_1, X_0) = \Sigma(0, 1, 2, 6)$ |
| 12      | $F(X_2, X_1, X_0) = \Sigma(0, 2, 3, 7)$ |
| 13      | $F(X_2, X_1, X_0) = \Sigma(2, 3, 4, 6)$ |
| 14      | $F(X_2, X_1, X_0) = \Sigma(3, 4, 5, 6)$ |
| 15      | $F(X_2, X_1, X_0) = \Sigma(1, 3, 5, 7)$ |

**Задача 4.** Реализовать функцию на дешифраторах D2\_4Е.

| Вариант | Функция    |
|---------|------------|
| 1       | 3ИЛИ       |
| 2       | 4XOR       |
| 3       | 3И-НЕ      |
| 4       | 4ИЛИ       |
| 5       | 3И         |
| 6       | 4XNOR      |
| 7       | 4И         |
| 8       | 3ИЛИ-НЕ    |
| 9       | 3XNOR      |
| 10      | 3XOR       |
| 11      | 4И-НЕ      |
| 12      | 4ИЛИ-НЕ    |
| 13      | 2И-2ИЛИ-НЕ |
| 14      | 2И-2ИЛИ    |
| 15      | 2И-3И-ИЛИ  |

**Задача 5.** Реализовать функцию с помощью дешифратора D3\_8Е и логических элементов ИЛИ.

| Вариант | Функция                                          |
|---------|--------------------------------------------------|
| 1       | 4XOR                                             |
| 2       | 4ИЛИ-НЕ                                          |
| 3       | 4XNOR                                            |
| 4       | 4И-НЕ                                            |
| 5       | 4И                                               |
| 6       | 4ИЛИ                                             |
| 7       | $X \cdot \neg Y + Z$                             |
| 8       | $(\neg X + Y) \cdot Z$                           |
| 9       | $(X \cdot Y + \neg X \cdot \neg Y) \cdot \neg Z$ |
| 10      | $X \cdot \neg Y + \neg Z$                        |
| 11      | $(X \cdot Y + \neg X \cdot \neg Y) \cdot Z$      |
| 12      | $X \cdot \neg Y + Z$                             |
| 13      | $X \cdot Y + \neg X \cdot \neg Z$                |
| 14      | $X \cdot \neg Y + \neg X \cdot \neg Z$           |
| 15      | $(X \cdot Y + X \cdot \neg Y) \cdot Z$           |

**Задача 6.** Реализовать на дешифраторах D2\_4Е или D3\_8Е функцию сравнения двух двухразрядных чисел  $A(A1,A0)$  и  $B(B1,B0)$  и принимающую истинное значение в соответствии с условием варианта.

| Вариант | Условие, при котором функция принимает истинное значение |
|---------|----------------------------------------------------------|
| 1       | $A(A1,A0) > B(B1,B0)$                                    |
| 2       | $A(A1,A0) \geq B(B1,B0)$                                 |
| 3       | $A(A1,A0) < B(B1,B0)$                                    |
| 4       | $A(A1,A0) \leq B(B1,B0)$                                 |
| 5       | $A(A1,A0) = B(B1,B0)$                                    |
| 6       | $A(A1,A0) > \neg B(B1,B0)$                               |
| 7       | $A(A1,A0) > \neg B(B1,B0)$                               |
| 8       | $A(A1,A0) \geq \neg B(B1,B0)$                            |
| 9       | $A(A1,A0) < \neg B(B1,B0)$                               |
| 10      | $A(A1,A0) \leq \neg B(B1,B0)$                            |
| 11      | $\neg A(A1,A0) = \neg B(B1,B0)$                          |
| 12      | $\neg A(A1,A0) > \neg B(B1,B0)$                          |
| 13      | $\neg A(A1,A0) > \neg B(B1,B0)$                          |
| 14      | $\neg A(A1,A0) \geq \neg B(B1,B0)$                       |
| 15      | $\neg A(A1,A0) < \neg B(B1,B0)$                          |

### 2.1.5. Синтез комбинационных схем на основе мультиплексоров

**Задача 1.** Указать восьмиразрядное слово  $X(X7-X0)$ , которое надо подать на входы мультиплексора MUX 8-1 для реализации логической функции  $F(A,B,C)$  при заданном порядке подачи переменных  $A,B,C$  на адресные входы  $S2,S1,S0$ .

| Вариант | $F(A,B,C)$                                                                           | $S2,S1,S0$ |
|---------|--------------------------------------------------------------------------------------|------------|
| 1       | $A \cdot B \cdot \neg C + A \cdot \neg B \cdot C + \neg A \cdot B \cdot C$           | $A,B,C$    |
| 2       | $\neg A \cdot \neg B \cdot \neg C + A \cdot B \cdot C + \neg A \cdot B \cdot C$      | $B,C,A$    |
| 3       | $A \cdot B \cdot \neg C + A \cdot \neg B \cdot C + \neg A \cdot B \cdot C$           | $B,A,C$    |
| 4       | $\neg A \cdot B \cdot C + \neg A \cdot B \cdot C + A \cdot \neg B \cdot \neg C$      | $C,A,B$    |
| 5       | $A \cdot B \cdot \neg C + A \cdot \neg B \cdot C + \neg A \cdot B \cdot C$           | $C,B,A$    |
| 6       | $A \cdot \neg B \cdot \neg C + A \cdot B \cdot \neg C + A \cdot \neg B \cdot C$      | $A,C,B$    |
| 7       | $\neg A \cdot \neg B \cdot \neg C + A \cdot B \cdot \neg C + A \cdot \neg B \cdot C$ | $A,B,C$    |

| Вариант | $F(A,B,C)$                                                                                | $S_2, S_1, S_0$ |
|---------|-------------------------------------------------------------------------------------------|-----------------|
| 8       | $A \cdot \neg B \cdot \neg C + \neg A \cdot B \cdot \neg C + \neg A \cdot \neg B \cdot C$ | $B, C, A$       |
| 9       | $A \cdot \neg B \cdot C + \neg A \cdot B \cdot \neg C + A \cdot \neg B \cdot \neg C$      | $B, A, C$       |
| 10      | $\neg A \cdot B \cdot C + A \cdot \neg B \cdot \neg C + A \cdot B \cdot C$                | $C, A, B$       |
| 11      | $A \cdot \neg B \cdot C + \neg A \cdot B \cdot \neg C + \neg A \cdot \neg B \cdot \neg C$ | $C, B, A$       |
| 12      | $A \cdot B \cdot \neg C + A \cdot \neg B \cdot \neg C + \neg A \cdot \neg B \cdot C$      | $A, C, B$       |
| 13      | $A \cdot B \cdot C + \neg A \cdot B \cdot \neg C + A \cdot \neg B \cdot \neg C$           | $B, A, C$       |
| 14      | $A \cdot B \cdot \neg C + \neg A \cdot \neg B \cdot C + \neg A \cdot B \cdot \neg C$      | $C, A, B$       |
| 15      | $\neg A \cdot B \cdot \neg C + A \cdot B \cdot C + \neg A \cdot \neg B \cdot C$           | $A, C, B$       |

**Задача 2.** Составить схему воспроизведения функции  $F(X_3, X_2, X_1, X_0)$  с помощью мультиплексора MUX 16-1.

| Вариант | $F(X_3, X_2, X_1, X_0)$                                                       |
|---------|-------------------------------------------------------------------------------|
| 1       | $\neg X_0 + X_1 + \neg X_2 \cdot X_3$                                         |
| 2       | $\neg X_0 + \neg X_1 + X_2 \cdot \neg X_3$                                    |
| 3       | $X_0 \cdot \neg X_2 + X_1 + X_2 \cdot X_3$                                    |
| 4       | $X_0 \cdot \neg X_1 + \neg X_0 \cdot X_2 + X_0 \cdot X_2 \cdot \neg X_3$      |
| 5       | $\neg X_0 \cdot \neg X_1 + X_0 \cdot \neg X_2 + \neg X_0 \cdot X_2 \cdot X_3$ |
| 6       | $\neg X_0 \cdot X_1 + \neg X_1 \cdot X_2 + X_1 \cdot \neg X_2 \cdot X_3$      |
| 7       | $X_0 \cdot \neg X_3 + X_1 \cdot \neg X_2 + X_0 \cdot X_2 \cdot X_3$           |
| 8       | $X_0 \cdot X_2 + X_1 \cdot \neg X_3 + X_0 \cdot X_2 \cdot \neg X_3$           |
| 9       | $X_1 + X_0 \cdot \neg X_2 + \neg X_1 \cdot X_3$                               |
| 10      | $X_2 + X_1 \cdot \neg X_2 + \neg X_0 \cdot X_3$                               |
| 11      | $\neg X_1 + X_2 \cdot \neg X_3 + \neg X_2 \cdot X_4$                          |
| 12      | $\neg X_3 + X_0 \cdot \neg X_1 \cdot X_3 + X_1 \cdot \neg X_2$                |
| 13      | $X_1 + X_0 \cdot \neg X_2 \cdot X_3 + X_1 \cdot \neg X_2$                     |
| 14      | $\neg X_0 + X_0 \cdot \neg X_1 \cdot X_3 + X_2 \cdot \neg X_3$                |
| 15      | $X_2 + X_0 \cdot \neg X_1 \cdot X_3 + \neg X_1 \cdot \neg X_2$                |

**Задача 3.** Реализовать с помощью мультиплексоров MUX 16-1 функцию от трех переменных  $F(A,B,C)$ . Порядок подачи сигналов на адресные входы  $S_1$  и  $S_0$  определяется условием варианта.

| Вариант | $F(A,B,C)$            | $S_1, S_0$ |
|---------|-----------------------|------------|
| 1       | $\Sigma(0,2,3,5,7)$   | $A, B$     |
| 2       | $\Sigma(1,3,4,6,7)$   | $B, C$     |
| 3       | $\Sigma(0,2,4,5,6,7)$ | $A, C$     |
| 4       | $\Sigma(2,3,4,5,7)$   | $B, A$     |

| Вариант | $F(A,B,C)$          | $S1,S0$ |
|---------|---------------------|---------|
| 5       | $\Sigma(1,3,4,5,6)$ | $C,B$   |
| 6       | $\Sigma(3,4,6,7)$   | $C,A$   |
| 7       | $\Sigma(0,1,5,6)$   | $A,B$   |
| 8       | $\Sigma(1,3,6,7)$   | $B,C$   |
| 9       | $\Sigma(2,4,5,7)$   | $A,C$   |
| 10      | $\Sigma(1,2,4,6)$   | $B,A$   |
| 11      | $\Sigma(0,1,4,7)$   | $C,B$   |
| 12      | $\Sigma(3,5,6,7)$   | $C,A$   |
| 13      | $\Sigma(1,3,5,6)$   | $B,A$   |
| 14      | $\Sigma(2,4,6,7)$   | $A,C$   |
| 15      | $\Sigma(1,2,4,6,7)$ | $C,A$   |

**Задача 4.** Реализовать с помощью мультиплексоров MUX 2-1 функцию от четырёх переменных  $F(A,B,C,D)$ .

| Вариант | $F(A,B,C,D)$                  |
|---------|-------------------------------|
| 1       | $\Sigma(0,4,5,7,8,11,14)$     |
| 2       | $\Sigma(1,2,4,6,9,10,12)$     |
| 3       | $\Sigma(0,3,4,8,12,13,15)$    |
| 4       | $\Sigma(2,5,7,8,9,12,14)$     |
| 5       | $\Sigma(3,5,7,9,10,11,12)$    |
| 6       | $\Sigma(4,6,7,8,10,11,13)$    |
| 7       | $\Sigma(4,5,6,8,12,14,15)$    |
| 8       | $\Sigma(5,7,8,9,11,13,15)$    |
| 9       | $\Sigma(3,4,6,7,9,10,11,14)$  |
| 10      | $\Sigma(1,3,5,6,8,10,12,13)$  |
| 11      | $\Sigma(4,5,7,8,11,13,14)$    |
| 12      | $\Sigma(2,4,6,7,9,10,11,12)$  |
| 13      | $\Sigma(5,6,7,9,10,11,12,14)$ |
| 14      | $\Sigma(1,4,6,8,10,11,13,15)$ |
| 15      | $\Sigma(4,5,7,8,9,11,13,15)$  |

**Задача 5.** Реализовать преобразователь кода Грея в двоично-десятичный код с помощью четырёх мультиплексоров MUX 2-1 и дополнительных логических элементов. Таблица истинности для кода «8421» приведена ниже. Значения выходных сигналов для кодов, отсутствующих в таблице, являются избыточными.

| Входы |    |    |    | Выходы |    |    |    |
|-------|----|----|----|--------|----|----|----|
| X3    | X2 | X1 | X0 | Y3     | Y2 | Y1 | Y0 |
| 0     | 0  | 0  | 0  | 0      | 0  | 0  | 0  |
| 0     | 0  | 0  | 1  | 0      | 0  | 0  | 1  |
| 0     | 0  | 1  | 1  | 0      | 0  | 1  | 0  |
| 0     | 0  | 1  | 0  | 0      | 0  | 1  | 1  |
| 0     | 1  | 1  | 0  | 0      | 1  | 0  | 0  |
| 0     | 1  | 1  | 1  | 0      | 1  | 0  | 1  |
| 0     | 1  | 0  | 1  | 0      | 1  | 1  | 0  |
| 0     | 1  | 0  | 0  | 0      | 1  | 1  | 1  |
| 1     | 1  | 0  | 0  | 1      | 0  | 0  | 0  |
| 1     | 1  | 0  | 1  | 1      | 0  | 0  | 1  |

Варианты:

| Вариант | Двоично-десятичный код |
|---------|------------------------|
| 1       | 3321                   |
| 2       | 5211                   |
| 3       | 6421                   |
| 4       | 7421                   |
| 5       | 4221                   |
| 6       | 5421                   |
| 7       | 6311                   |
| 8       | 7321                   |
| 9       | 4421                   |
| 10      | 5121                   |
| 11      | 6321                   |
| 12      | 4321                   |
| 13      | 5321                   |
| 14      | 6221                   |
| 15      | 2621                   |

**Задача 6.** Реализовать функцию  $F(A,B,C,D)$  от четырёх переменных на мультиплексоре MUX 4-1Е. Построить временную диаграмму для заданной схемы.

| Вариант | $F(A,B,C,D)$                        |
|---------|-------------------------------------|
| 1       | $\Sigma(2, 3, 7, 8, 9, 11, 12, 14)$ |
| 2       | $\Sigma(0, 2, 6, 7, 9, 13, 14, 15)$ |

| Вариант | $F(A,B,C,D)$                             |
|---------|------------------------------------------|
| 3       | $\Sigma(4, 5, 6, 7, 8, 10, 11, 15)$      |
| 4       | $\Sigma(0, 2, 3, 7, 10, 11, 12, 15)$     |
| 5       | $\Sigma(1, 3, 5, 7, 8, 10, 11, 15)$      |
| 6       | $\Sigma(0, 1, 2, 3, 5, 9, 10, 13)$       |
| 7       | $\Sigma(2, 3, 4, 9, 10, 11, 12, 13)$     |
| 8       | $\Sigma(0, 1, 5, 7, 11, 13, 14, 15)$     |
| 9       | $\Sigma(1, 2, 5, 7, 9, 10, 12, 13)$      |
| 10      | $\Sigma(0, 2, 5, 7, 10, 12, 14, 15)$     |
| 11      | $\Sigma(4, 6, 7, 9, 10, 11, 12, 13, 15)$ |
| 12      | $\Sigma(1, 4, 5, 7, 8, 9, 12, 14)$       |
| 13      | $\Sigma(1, 2, 4, 5, 8, 10, 13, 15)$      |
| 14      | $\Sigma(3, 4, 7, 10, 11, 12, 14, 15)$    |
| 15      | $\Sigma(2, 4, 5, 7, 8, 9, 11, 13, 15)$   |

**Задача 7.** Реализовать с помощью мультиплексоров MUX 4-1 и логических элементов И-НЕ функцию от четырёх переменных  $F(A,B,C,D)$ . Порядок подачи сигналов на адресные входы  $S_1$  и  $S_0$  определяется условием варианта.

| Вариант | $F(A,B,C,D)$                      | $S_1, S_0$ |
|---------|-----------------------------------|------------|
| 1       | $\Sigma(0,1,3,5,6,8,9,11,12,13)$  | $A, B$     |
| 2       | $\Sigma(0,7,8,9,10,11,13,14,15)$  | $B, C$     |
| 3       | $\Sigma(0,1,3,5,10,11,13,14,15)$  | $A, C$     |
| 4       | $\Sigma(1,8,9,10,12,13,14,15)$    | $B, A$     |
| 5       | $\Sigma(1,3,5,6,8,9,10,12,13)$    | $C, B$     |
| 6       | $\Sigma(0,3,4,6,7,9,10,11,14)$    | $C, A$     |
| 7       | $\Sigma(0,1,3,5,6,8,10,11,13,15)$ | $A, B$     |
| 8       | $\Sigma(1,3,4,5,7,9,10,12,14,15)$ | $B, C$     |
| 9       | $\Sigma(2,4,6,7,8,9,11,13,15)$    | $A, C$     |
| 10      | $\Sigma(1,2,3,6,8,10,11,12,14)$   | $B, A$     |
| 11      | $\Sigma(0,1,4,5,8,10,11,13,15)$   | $C, B$     |
| 12      | $\Sigma(1,3,5,6,7,9,10,11,14,15)$ | $C, A$     |
| 13      | $\Sigma(1,2,4,6,7,8,10,12,13,14)$ | $B, A$     |
| 14      | $\Sigma(2,3,6,8,9,10,11,13,15)$   | $A, C$     |
| 15      | $\Sigma(1,2,4,5,7,8,10,11,12,13)$ | $C, A$     |

**Задача 8.** Реализовать функцию с помощью мультиплексоров.

| Вариант | Функция | Мультиплексор |
|---------|---------|---------------|
| 1       | 4И      | MUX 4-1E      |
| 2       | 4ИЛИ    | MUX 4-1E      |
| 3       | 4XOR    | MUX 4-1E      |
| 4       | 4ИЛИ-НЕ | MUX 4-1E      |
| 5       | 4И-НЕ   | MUX 4-1E      |
| 6       | 4XNOR   | MUX 4-1E      |
| 7       | 3XNOR   | MUX 4-1E      |
| 8       | 4XOR    | MUX 8-1E      |
| 9       | 4ИЛИ    | MUX 8-1E      |
| 10      | 4XNOR   | MUX 8-1E      |
| 11      | 4ИЛИ-НЕ | MUX 8-1E      |
| 12      | 4И-НЕ   | MUX 8-1E      |
| 13      | 3ИЛИ    | MUX2-1E       |
| 14      | 3И      | MUX2-1E       |
| 15      | 3ИЛИ-НЕ | MUX2-1E       |

### **2.1.6. Синтез комбинационных схем на основе мультиплексоров при словесном задании функции**

**Задача 1.** Реализовать схему, выполняющую функцию, заданную вариантом, с помощью мультиплексоров.

Вариант 1. Схема сравнения двух одноразрядных переменных.

Вариант 2. Схема одноразрядного двоичного сумматора.

Вариант 3. Схема одноразрядного двоичного вычитателя.

Вариант 4. Схема получения признака чётности числа нулей в 3-разрядном двоичном коде («0» – чётно, «1» – нечётно).

Вариант 5. Схема, реализующая функцию голосования «2 из 3».

Вариант 6. Схема получения признака чётности числа единиц в 3-разрядном двоичном коде («0» – чётно, «1» – нечётно).

Вариант 7. Схема, реализующая функцию равнозначности (эквивалентности) трёх аргументов.

Вариант 8. Схема, реализующая функцию равнозначности (эквивалентности) четырёх аргументов.

Вариант 9. Схема получения признака чётности числа единиц в 4-разрядном двоичном коде («0» – чётно, «1» – нечётно).

Вариант 10. Схема голосования «2 из 4».

Вариант 11. Схема получения признака чётности числа нулей в 4-разрядном двоичном коде («0» – чётно, «1» – нечётно).

Вариант 12. Схема умножения двух двухразрядных чисел.

Вариант 13. Схема голосования «3 из 4».

Вариант 14. Схема голосования «2 из 3».

Вариант 15. Схема определения номера бита старшей единицы в 4-разрядном двоичном слове, отличном от 0000.

**Задача 2.** Реализовать на мультиплексорах MUX 4-1 функцию сравнения двух двухразрядных чисел  $A(A1,A0)$  и  $B(B1,B0)$  и принимающую истинное значение в соответствии с условием варианта.

| Вариант | Условие, при котором функция принимает истинное значение |
|---------|----------------------------------------------------------|
| 1       | $A(A1,A0) > B(B1,B0)$                                    |
| 2       | $A(A1,A0) \geq B(B1,B0)$                                 |
| 3       | $A(A1,A0) < B(B1,B0)$                                    |
| 4       | $A(A1,A0) \leq B(B1,B0)$                                 |
| 5       | $A(A1,A0) = B(B1,B0)$                                    |
| 6       | $A(A1,A0) > \neg B(B1,B0)$                               |
| 7       | $A(A1,A0) < \neg B(B1,B0)$                               |
| 8       | $A(A1,A0) \geq \neg B(B1,B0)$                            |
| 9       | $A(A1,A0) \leq \neg B(B1,B0)$                            |
| 10      | $A(A1,A0) \leq \neg B(B1,B0)$                            |
| 11      | $\neg A(A1,A0) = \neg B(B1,B0)$                          |
| 12      | $\neg A(A1,A0) > \neg B(B1,B0)$                          |
| 13      | $\neg A(A1,A0) < \neg B(B1,B0)$                          |
| 14      | $\neg A(A1,A0) \geq \neg B(B1,B0)$                       |
| 15      | $\neg A(A1,A0) < \neg B(B1,B0)$                          |

### **2.1.7. Синтез преобразователей кодов**

**Задача.** Разработать схему преобразователя кодов, заданную вариантом:

Вариант 1. Схема получения дополнительного кода от 4-разрядного двоичного числа (старший разряд – знаковый).

Вариант 2. Схема преобразования двоичного кода «8421» (цифры 0-9) в код «с избытком 3».

Вариант 3. Схема преобразования двоичного кода «5421» (цифры 0-9) в код «с избытком 3».

Вариант 4. Схема преобразования двоичного кода «с избытком 3» (цифры 0-9) в код «5421».

Вариант 5. Схема преобразования двоичного кода «с избытком 3» (цифры 0-9) в код «8421».

Вариант 6. Схема преобразования двоичного кода «8421» (цифры 0-9) в код «5421».

Вариант 7. Приоритетный шифратор для преобразования 8-разрядного входного кода в двоичный код числа  $i$ , где  $i$  – номер позиции старшей единицы во входном коде.

Вариант 8. Схема, с помощью которой можно получить дополнение до 9 для каждой десятичной цифры. Десятичные цифры представляются в коде «8421».

Вариант 9. Преобразователь кодов для перевода цифр от 1 до 8 кода «8421» в код «2 из 5».

Вариант 10. Преобразователь кодов для перевода цифр от 2 до 9 кода «2421» в код «2 из 5».

Вариант 11. Преобразователь кодов для перевода цифр от 0 до 7 кода «с избытком 3» в код «2 из 5».

Вариант 12. Преобразователь кодов для перевода цифр от 0 до 7 кода «3а+2» в код «2 из 5».

Вариант 13. Преобразователь кодов для перевода цифр от 1 до 8 кода «7421» в код «2 из 5».

Вариант 14. Преобразователь кодов для перевода цифр от 2 до 9 кода «8421» в код «2 из 5».

Вариант 15. Преобразователь кодов для перевода цифр от 1 до 8 кода «2421» в код «2 из 5».

### ***2.1.8. Синтез шифраторов и дешифраторов***

***Задача.*** Разработать схему, заданную вариантом:

Вариант 1. Шифратор для перевода цифр десятичного кода от 3 до 9 в код «8421».

Вариант 2. Шифратор для перевода цифр десятичного кода от 1 до 9 в код «7421».

Вариант 3. Шифратор для перевода цифр десятичного кода от 1 до 8 в код «с избытком 3».

Вариант 4. Шифратор для перевода цифр десятичного кода от 2 до 8 в код «2421».

Вариант 5. Шифратор для перевода цифр десятичного кода от 2 до 9 в код «7421».

Вариант 6. Шифратор для перевода цифр десятичного кода от 1 до 9 в код «с избытком 3».

Вариант 7. Шифратор для перевода цифр десятичного кода от 0 до 9 в код «2421».

Вариант 8. Шифратор для перевода цифр десятичного кода от 0 до 9 в код «8421».

Вариант 9. Шифратор для перевода цифр десятичного кода от 0 до 9 в код «2 из 5».

Вариант 10. Дешифратор для перевода цифр от 2 до 9 кода «8421» в десятичный код.

Вариант 11. Дешифратор для перевода цифр от 1 до 7 кода «7421» в десятичный код.

Вариант 12. Дешифратор для перевода цифр от 2 до 9 кода «с избытком 3» в десятичный код.

Вариант 13. Дешифратор для перевода цифр от 0 до 8 кода «2 из 5» в десятичный код.

Вариант 14. Дешифратор для перевода цифр от 0 до 9 кода «2421» в десятичный код.

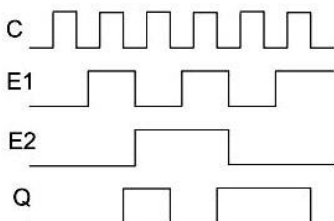
Вариант 15. Дешифратор для перевода цифр от 0 до 7 кода «2 из 5» в десятичный код.

## 2.2. Проектирование и анализ последовательностных схем

### 2.2.1. Определение типа триггера по временной диаграмме

**Задача.** Какая из двух представленных в варианте таблиц переходов, описывающих поведение двухступенчатого триггера, соответствует приведенной временной диаграмме?

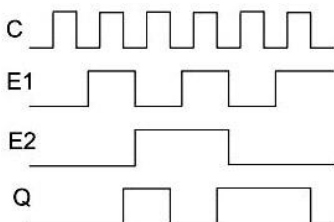
Вариант 1.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | Q(t)   |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | 1      |

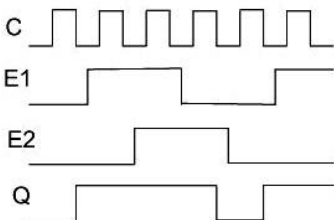
Вариант 2.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | Q(t)   |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | 1      |

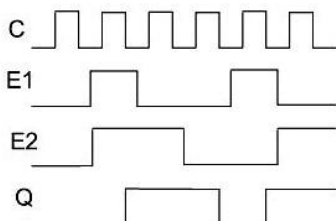
Вариант 3.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | Q(t)   |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 1      |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | 1      |

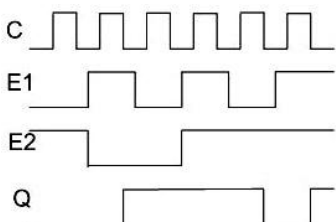
### Вариант 4.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

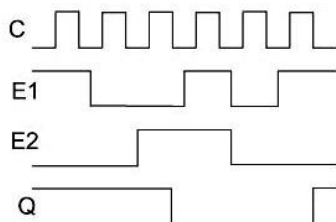
### Вариант 5.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 1      |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | 1      |

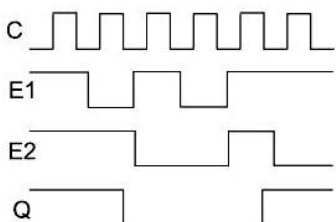
### Вариант 6.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | Q(t)   |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 0      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

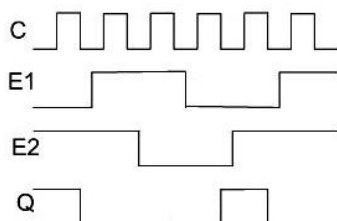
### Вариант 7.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 1      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 1      |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | 1      |

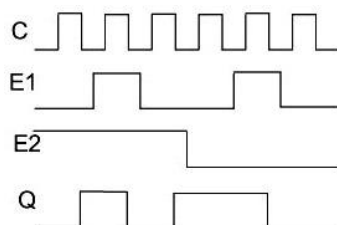
### Вариант 8.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | Q(t)   |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 1      |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | Q(t)   |

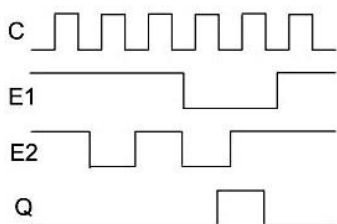
### Вариант 9.



| №1    |       |                   |
|-------|-------|-------------------|
| E1(t) | E2(t) | Q(t+1)            |
| 0     | 0     | Q(t)              |
| 0     | 1     | 1                 |
| 1     | 0     | $\overline{Q(t)}$ |
| 1     | 1     | 0                 |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

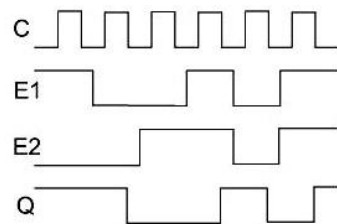
### Вариант 10.



| №1    |       |                   |
|-------|-------|-------------------|
| E1(t) | E2(t) | Q(t+1)            |
| 0     | 0     | Q(t)              |
| 0     | 1     | $\overline{Q(t)}$ |
| 1     | 0     | 1                 |
| 1     | 1     | 0                 |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 1      |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | Q(t)   |

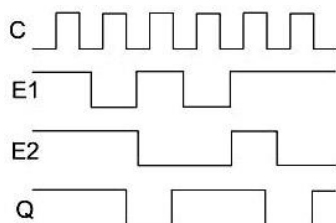
### Вариант 11.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | Q(t)   |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 0      |
| 0     | 1     | Q(t)   |
| 1     | 0     | Q(t)   |
| 1     | 1     | 1      |

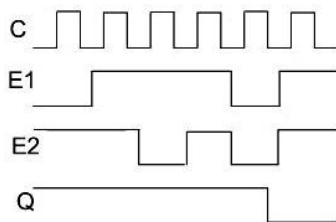
### Вариант 12.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 1      |
| 0     | 1     | 0      |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | 1      |

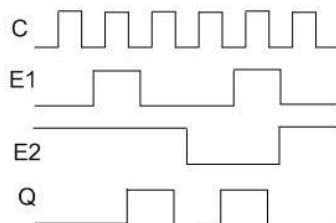
### Вариант 13.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 1      |
| 1     | 0     | 0      |
| 1     | 1     | 1      |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 0      |
| 0     | 1     | 1      |
| 1     | 0     | Q(t)   |
| 1     | 1     | Q(t)   |

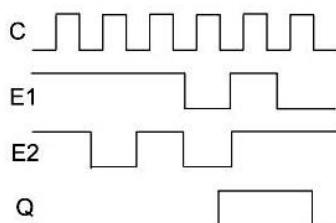
### Вариант 14.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | 1      |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 1      |
| 0     | 1     | 0      |
| 1     | 0     | 0      |
| 1     | 1     | 1      |

### Вариант 15.



| №1    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | Q(t)   |
| 0     | 1     | Q(t)   |
| 1     | 0     | 1      |
| 1     | 1     | 0      |

| №2    |       |        |
|-------|-------|--------|
| E1(t) | E2(t) | Q(t+1) |
| 0     | 0     | 1      |
| 0     | 1     | 0      |
| 1     | 0     | Q(t)   |
| 1     | 1     | Q(t)   |

### 2.2.2. Синтез триггера по заданной таблице переходов

**Задача.** Разработать триггер с логическими входами  $E1, E2$ , с таблицей переходов, заданной вариантом.

Для каждого варианта дополнительным условием может быть требование реализации триггера на основе триггера типа D, T, RS, JK или DV, а также требование наличия дополнительных входов:

1. Асинхронной установки в ноль  $CLR$ .
2. Асинхронной установки в единицу  $PRE$ .
3. Асинхронной установки в ноль  $R$ .
4. Синхронной установки в единицу  $S$ .
5. Асинхронной установки в ноль  $CLR$  и синхронной установки в единицу  $S$ .
6. Асинхронной установки в единицу  $PRE$  и синхронной установки в ноль  $R$ .
7. Синхронной установки в ноль  $R$  и в единицу  $S$ .
8. Асинхронной установки в ноль  $CLR$  и логического входа  $D$ .
9. Асинхронной установки в единицу  $PRE$  и логического входа  $D$ .
10. Синхронной установки в ноль  $R$  и логического входа  $D$ .
11. Синхронной установки в единицу  $S$  и логического входа  $D$ .
12. Асинхронной установки в ноль  $CLR$  и синхронной установки в единицу  $S$  и логического входа  $D$ .
13. Асинхронной установки в единицу  $PRE$  и синхронной установки в ноль  $R$  и логического входа  $D$ .
14. Синхронной установки в ноль  $R$  и в единицу  $S$  и логического входа  $D$ .

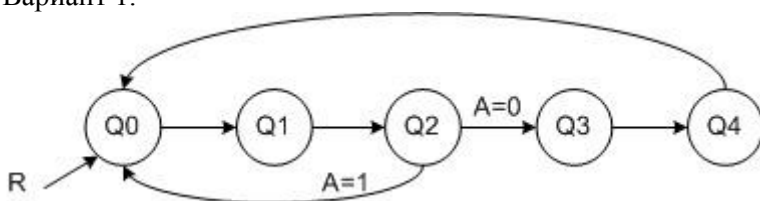
| Входы<br>триггера |         | Варианты |          |          |          |          |          |
|-------------------|---------|----------|----------|----------|----------|----------|----------|
|                   |         | 1        | 2        | 3        | 4        | 5        | 6        |
| $E1(t)$           | $E2(t)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ |
| 0                 | 0       | 0        | 0        | 0        | $Q(t)$   | 0        | 0        |
| 0                 | 1       | 0        | $Q(t)$   | 0        | 0        | $Q(t)$   | 0        |
| 1                 | 0       | 1        | $NQ(t)$  | 0        | 1        | 1        | $Q(t)$   |
| 1                 | 1       | $Q(t)$   | 1        | 1        | 0        | 1        | 1        |
| Входы<br>триггера |         | Варианты |          |          |          |          |          |
|                   |         | 7        | 8        | 9        | 10       | 11       | 12       |
| $E1(t)$           | $E2(t)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ |
| 0                 | 0       | $Q(t)$   | 1        | 1        | 0        | $Q(t)$   | $Q(t)$   |
| 0                 | 1       | $Q(t)$   | $Q(t)$   | 0        | 1        | 0        | 1        |
| 1                 | 0       | 1        | 1        | $Q(t)$   | $Q(t)$   | 1        | 0        |
| 1                 | 1       | 0        | 0        | 0        | $Q(t)$   | 1        | 0        |
| Входы<br>триггера |         | Варианты |          |          |          |          |          |
|                   |         | 13       | 14       | 15       | 16       | 17       | 18       |
| $E1(t)$           | $E2(t)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ |
| 0                 | 0       | 0        | 1        | 0        | 0        | 1        | 0        |
| 0                 | 1       | $Q(t)$   | 1        | 0        | $Q(t)$   | 0        | 1        |
| 1                 | 0       | 1        | 0        | 1        | 0        | $Q(t)$   | $Q(t)$   |
| 1                 | 1       | $Q(t)$   | $Q(t)$   | 0        | 1        | $Q(t)$   | 1        |
| Входы<br>триггера |         | Варианты |          |          |          |          |          |
|                   |         | 19       | 20       | 21       | 22       | 23       | 24       |
| $E1(t)$           | $E2(t)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ |
| 0                 | 0       | 1        | 1        | $Q(t)$   | 0        | $Q(t)$   | 1        |
| 0                 | 1       | 1        | 0        | 1        | 1        | 1        | 1        |
| 1                 | 0       | 1        | 1        | $Q(t)$   | 0        | 0        | $Q(t)$   |
| 1                 | 1       | 0        | $Q(t)$   | 0        | $Q(t)$   | 1        | 0        |
| Входы<br>триггера |         | Варианты |          |          |          |          |          |
|                   |         | 25       | 26       | 27       | 28       | 29       | 30       |
| $E1(t)$           | $E2(t)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ | $Q(t+1)$ |
| 0                 | 0       | 0        | $Q(t)$   | 0        | $Q(t)$   | 1        | $Q(t)$   |
| 0                 | 1       | 1        | $Q(t)$   | 1        | $Q(t)$   | 0        | 0        |
| 1                 | 0       | 0        | 0        | 1        | 1        | 1        | 1        |
| 1                 | 1       | 0        | 1        | 1        | 0        | 1        | $NQ(t)$  |

### 2.2.3. Проектирование схем на основе триггеров по графу переходов

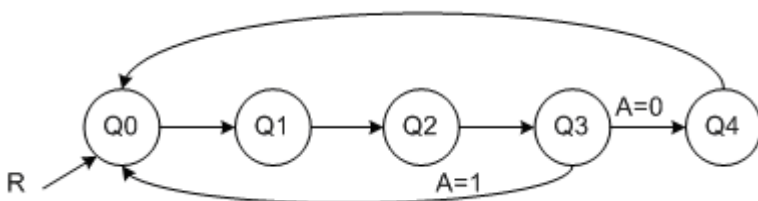
**Задача.** Разработать схему синхронного автомата (унитарный код) с условным переходом по условию  $A$ , работающую в соответствии с заданным графом переходов. Схема имеет асинхронный вход  $R$  установки в начальное состояние, вход синхронизации  $C$ , управляющий вход  $A$  и пять выходов  $Q_4$ - $Q_0$ .

Для проверки работы схемы построить временную диаграмму.

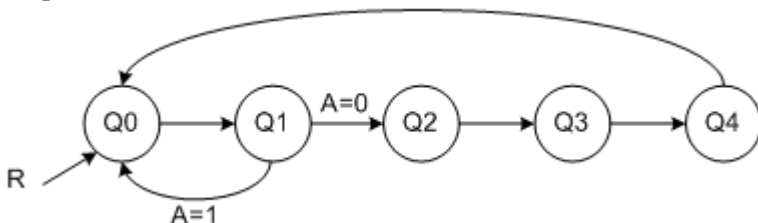
Вариант 1.



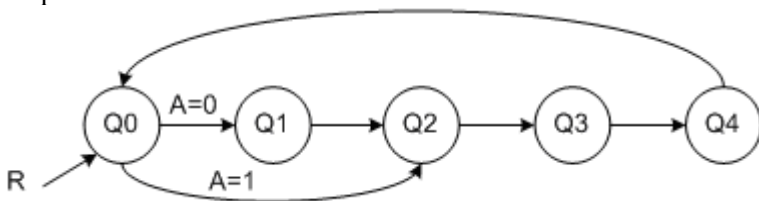
Вариант 2.



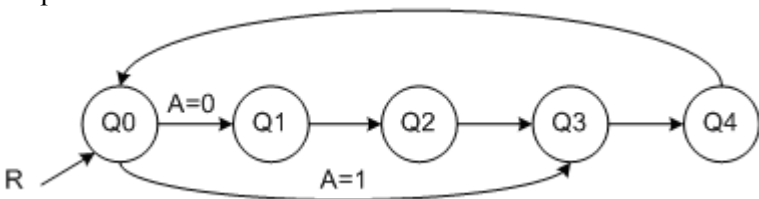
Вариант 3.



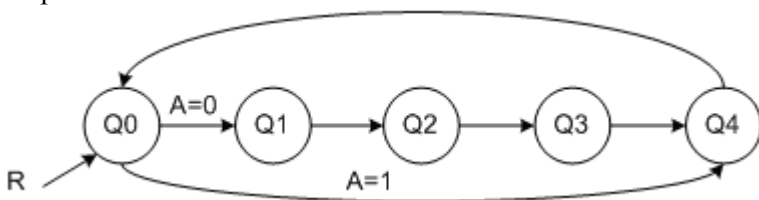
Вариант 4.



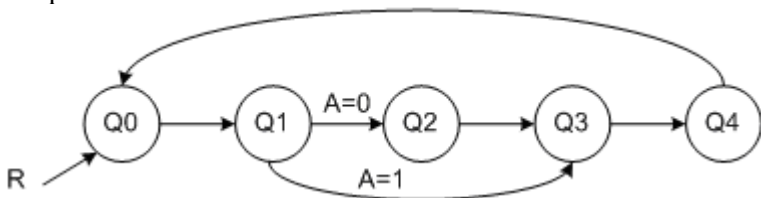
Вариант 5.



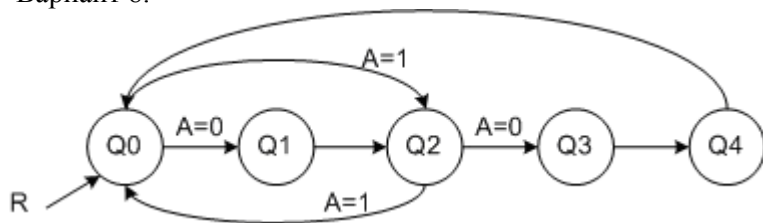
Вариант 6.



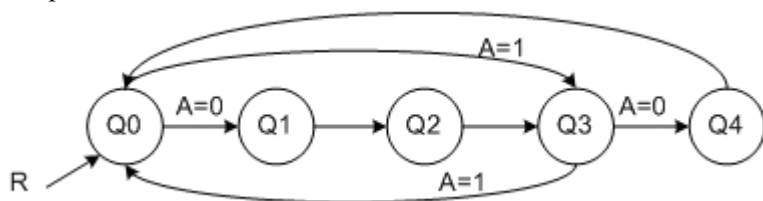
Вариант 7.



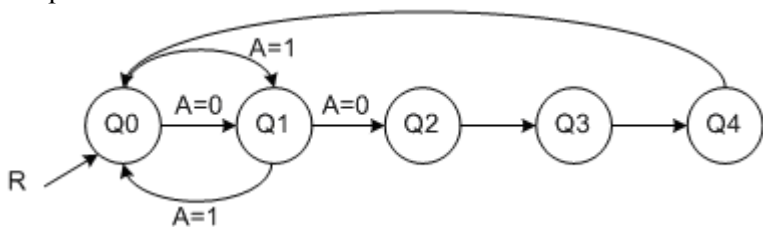
Вариант 8.



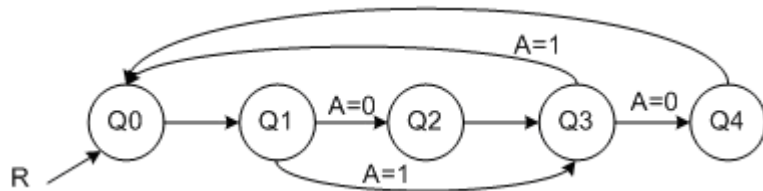
Вариант 9.



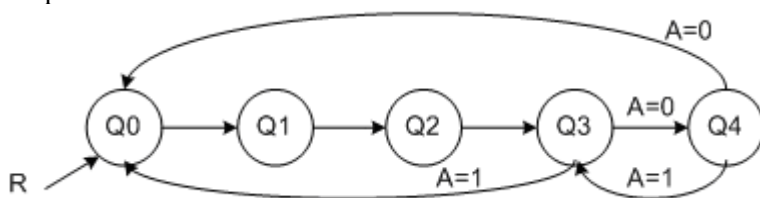
Вариант 10.



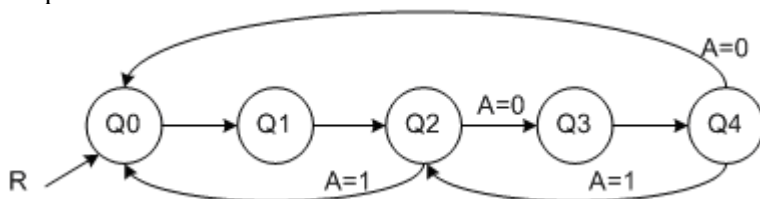
Вариант 11.



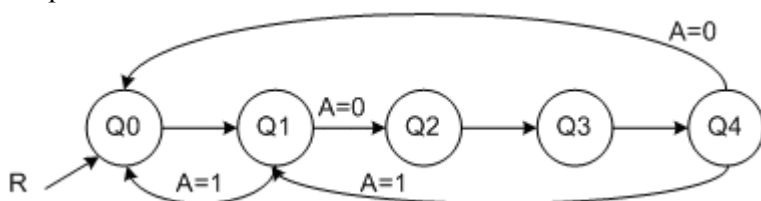
Вариант 12.



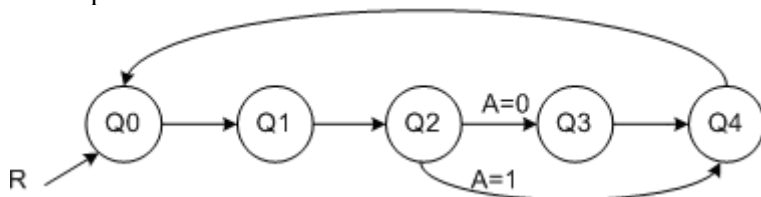
Вариант 13.



Вариант 14.



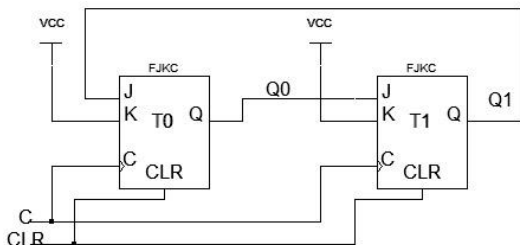
Вариант 15.



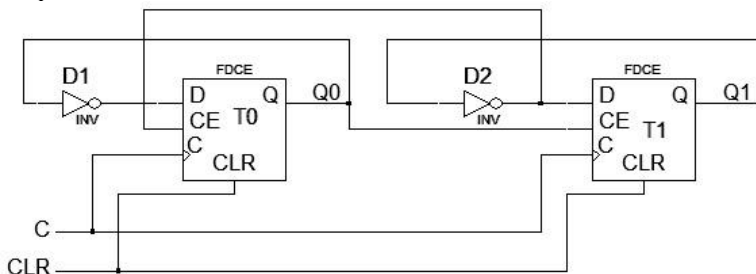
### 2.2.4. Построение временных диаграмм работы схем на основе триггеров

**Задача 1.** Построить временную диаграмму работы схемы до момента повторения последовательности состояний триггеров или до момента повторения одного и того же состояния.

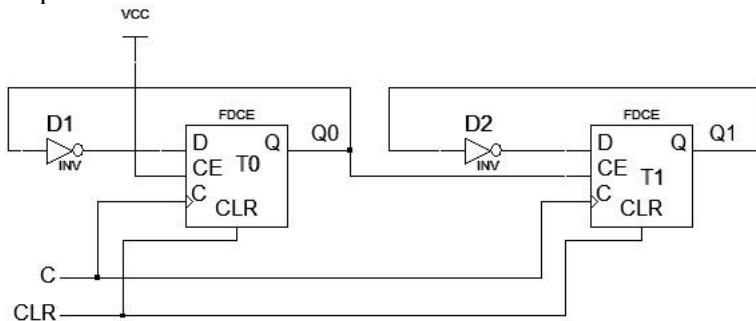
Вариант 1.



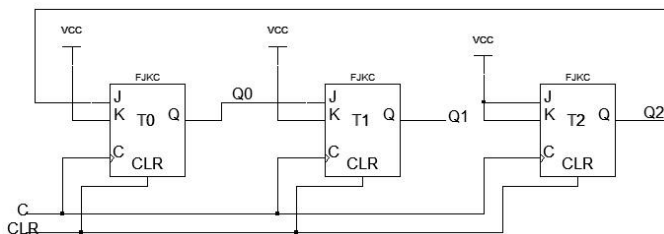
Вариант 2.



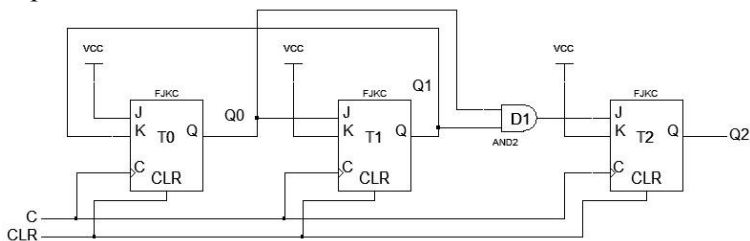
Вариант 3.



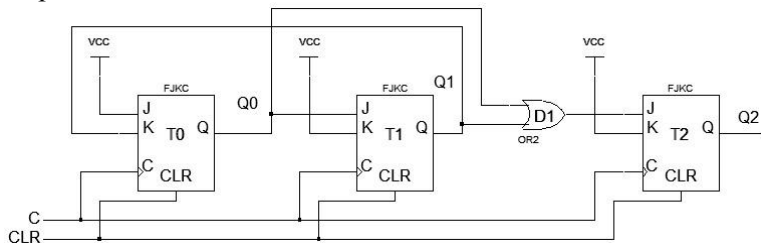
#### Вариант 4.



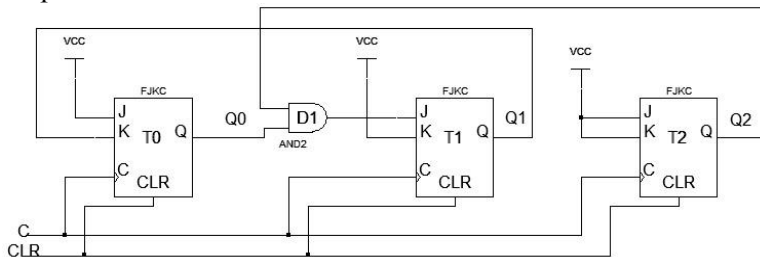
#### Вариант 5.



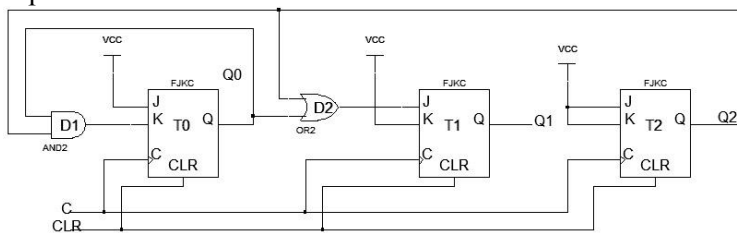
#### Вариант 6.



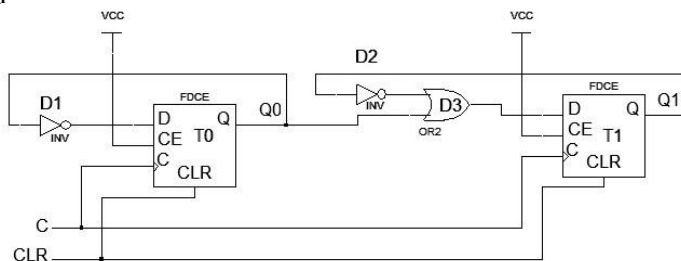
#### Вариант 7.



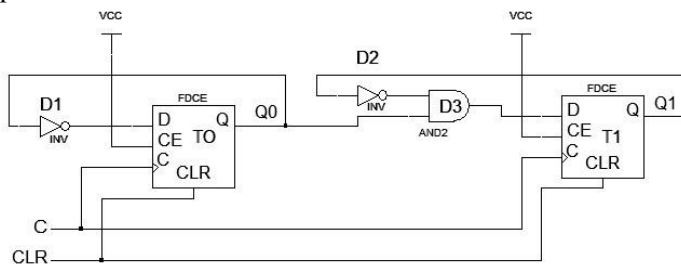
### Вариант 8.



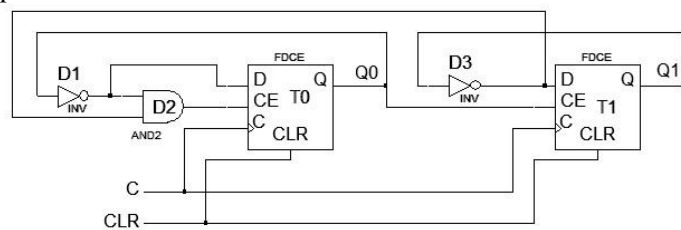
### Вариант 9.



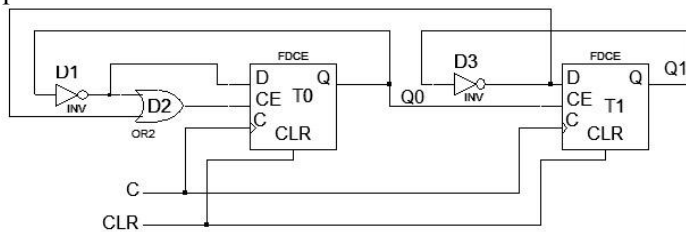
### Вариант 10.



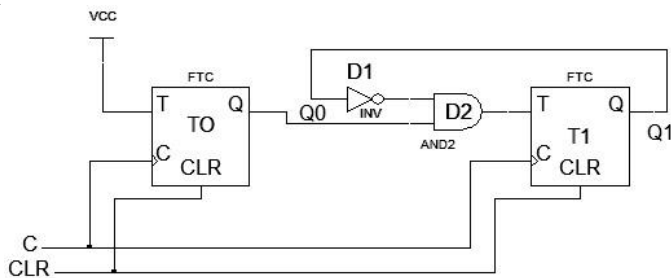
### Вариант 11.



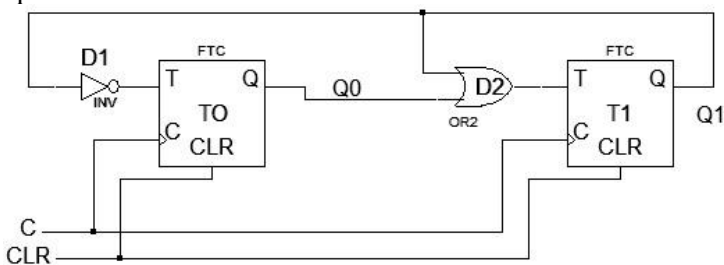
Вариант 12.



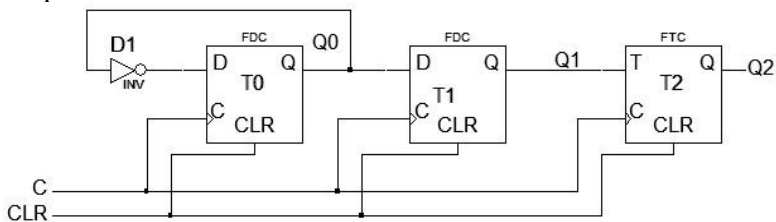
Вариант 13.



Вариант 14.

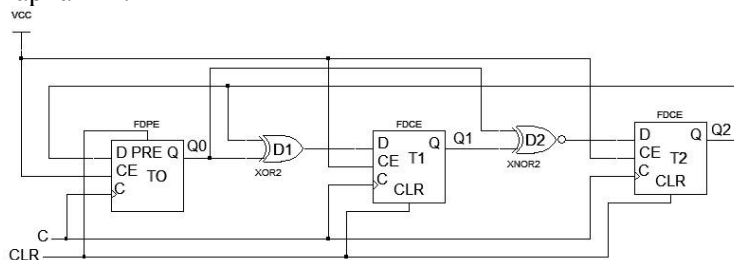


Вариант 15.

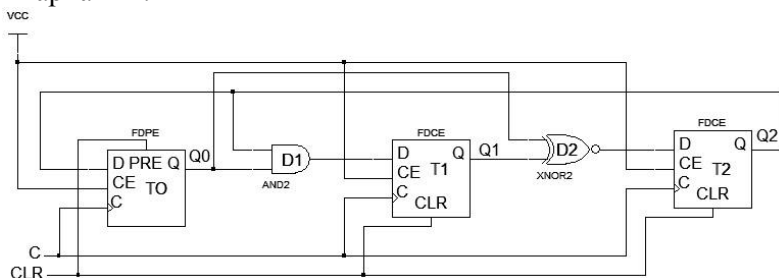


**Задача 2.** Построить временную диаграмму работы схемы до момента повторения последовательности состояний триггеров или до момента повторения одного и того же состояния.

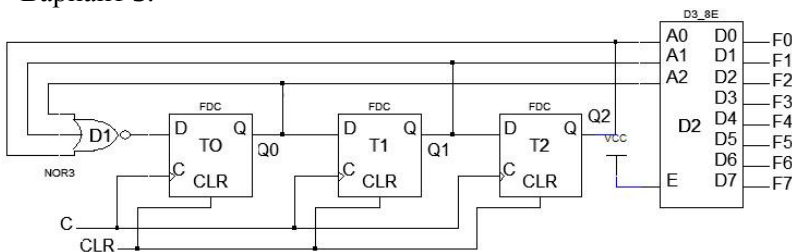
Вариант 1.



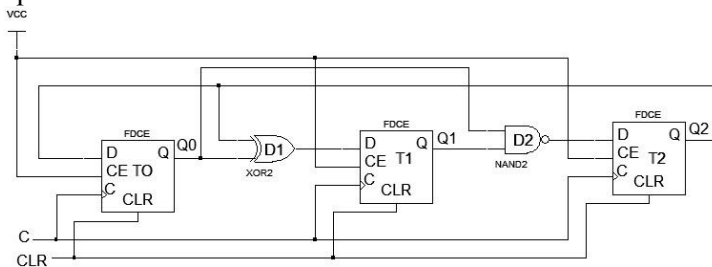
Вариант 2.



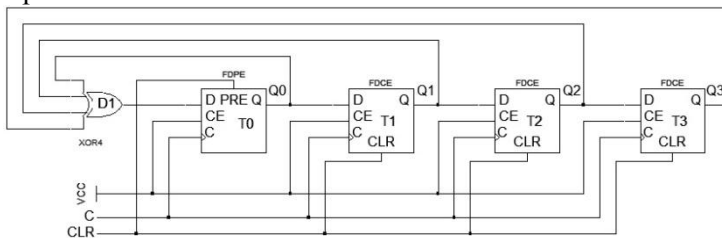
Вариант 3.



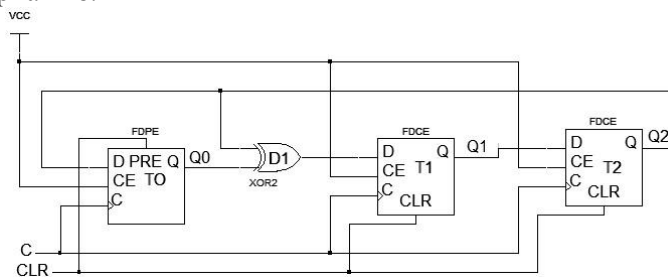
### Вариант 4.



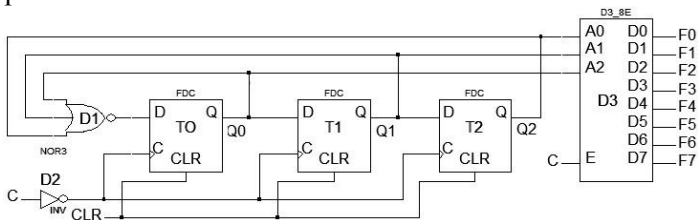
### Вариант 5.



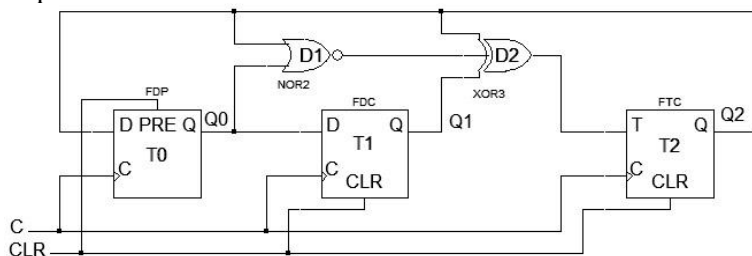
### Вариант 6.



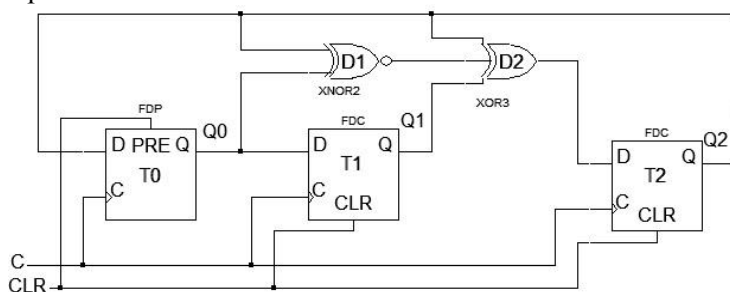
### Вариант 7.



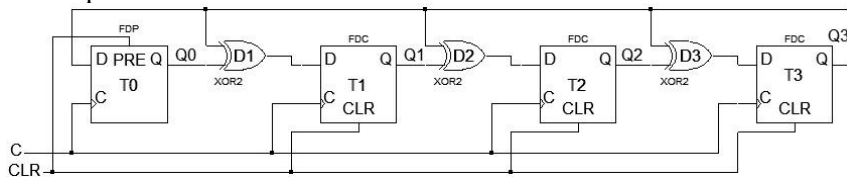
Вариант 8.



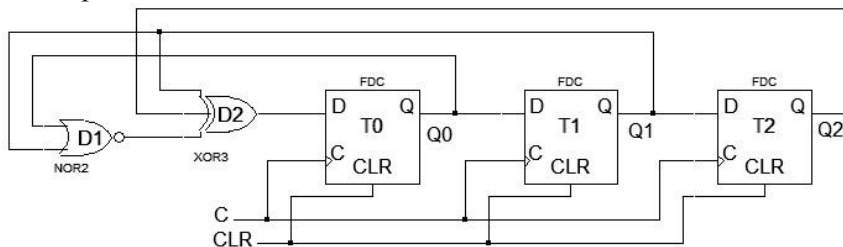
Вариант 9.



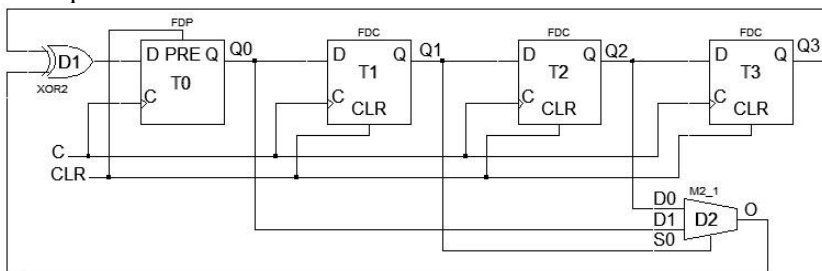
Вариант 10.



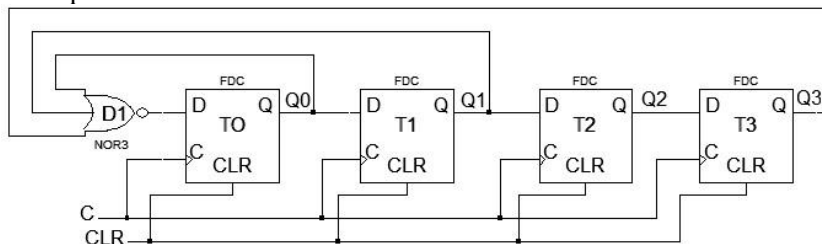
Вариант 11.



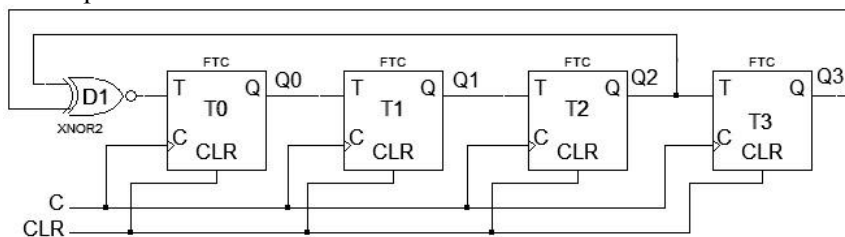
### Вариант 12.



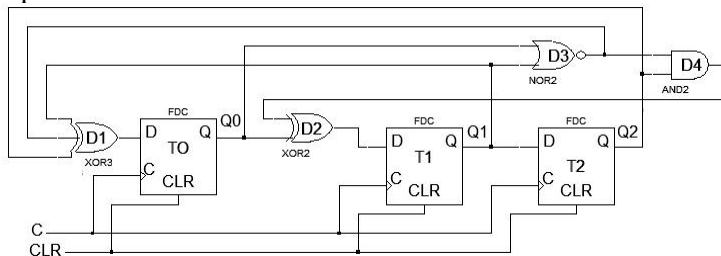
### Вариант 13.



### Вариант 14.



### Вариант 15.

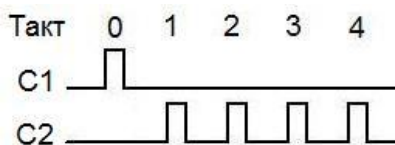


### 2.2.5. Синтез схем на основе регистров

**Задача 1.** Разработать схему, сдвигающую код, находящийся в 8-разрядном регистре, на  $K$  разрядов в сторону старших разрядов (влево) или в сторону младших разрядов (вправо). Освобождающиеся разряды заполнить нулями. На какое количество разрядов необходимо сдвинуть код и направление сдвига указывает трехразрядный код  $Y_2, Y_1, Y_0$ , подаваемый на схему. Значение разряда  $Y_2$  указывает направление сдвига:  $Y_2 = 1$  – сдвиг влево,  $Y_2 = 0$  – сдвиг вправо. Значение разрядов  $Y_1, Y_0$  задаёт  $K$  – количество разрядов, на которое необходимо сдвинуть код, хранимый в регистре:

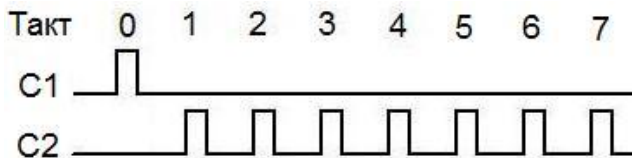
00 –  $K = 1$ , 01 –  $K = 2$ , 10 –  $K = 3$ , 11 –  $K = 4$ .

Схема имеет ещё два внешних входа, на которые подаются только пять импульсов в соответствии с временной диаграммой.



Коды  $Y_2, Y_1, Y_0$  поддерживаются неизменными в течение всей временной диаграммы.

**Задача 2.** В регистре хранится 8-разрядный код. Разработать схему подсчёта в коде числа групп из рядом стоящих единиц. Граница группы – либо ноль, либо граница кода. Например, в коде 11011001 две таких группы. Результат зафиксировать. На схему подаются только восемь импульсов в соответствии с временной диаграммой.



**Задача 3.** В регистре хранится 8-разрядный код. Разработать схему подсчёта общего числа групп нулей и единиц в этом коде.

Группой считается произвольное количество одинаковых рядом стоящих двоичных разрядов. Например, в коде 10011101 содержится пять групп разрядов. Результат зафиксировать.

На схему подаются только восемь импульсов в соответствии с временной диаграммой (см. рис. к задаче 2).

**Задача 4.** В регистре хранится 8-разрядный код. Разработать схему подсчёта общего числа изолированных нулей в коде, т.е. количества в коде комбинаций 101 из трёх разрядов. Ноль считается изолированным, если он с обеих сторон имеет единицы. Крайний ноль изолированным не считается. Например, в коде 10101110 содержится два изолированных нуля. Результат зафиксировать.

На схему подаются только восемь импульсов в соответствии с временной диаграммой (см. рис. к задаче 2).

**Задача 5.** В регистре хранится 8-разрядный код. Разработать схему подсчёта общего числа групп нулей и единиц в этом коде. Группой нулей (единиц) считается не менее двух рядом стоящих нулей (единиц), отделённых единицами (нулями) или границей кода. Например, код 11001000 содержит три группы. Результат зафиксировать.

На схему подаются только восемь импульсов в соответствии с временной диаграммой (см. рис. к задаче 2).

**Задача 6.** В регистре хранится 8-разрядный код. Разработать схему подсчёта общего числа изолированных единиц в коде, т.е. количества в коде комбинаций 010 из трёх разрядов. Единица считается изолированной, если она с обеих сторон имеет нули. Крайняя единица изолированной не считается. Например, в коде 01011001 содержится одна изолированная единица. Результат зафиксировать.

На схему подаются только восемь импульсов в соответствии с временной диаграммой (см. рис. к задаче 2).

**Задача 7.** В регистре хранится 8-разрядный код. Разработать схему подсчёта общего числа изолированных нулей и единиц в этом коде. Ноль (единица) считается изолированным, если он (она) с обеих сторон имеет единицы (нули). Крайний разряд не может

быть изолированным. Например, для кода 10110100 результат равен трём (одна изолированная единица и два изолированных нуля).

На схему подаются только восемь импульсов в соответствии с временной диаграммой (см. рис. к задаче 2).

**Задача 8.** Разработать схему, сдвигающую влево двоичный код  $Q_3, Q_2, Q_2, Q_0$  на число разрядов, равное числу единиц в этом коде. Освобождающиеся разряды заполнить нулями. Значения выдвигаемых разрядов теряются. Код 0000 можно сдвигать на любое число разрядов. Исходный код хранится в 8-разрядном регистре. Схема имеет два входа, на которые подаются только пять импульсов в соответствии с временной диаграммой (см. рис. к задаче 1).

**Задача 9.** Разработать схему, сдвигающую влево двоичный код  $Q_3, Q_2, Q_1, Q_0$  на число разрядов, равное числу единиц в этом коде за один такт.

**Задача 10.** Восьмиразрядный регистр сдвига выполняет микрооперации сдвига вправо на один разряд и сдвига влево на два разряда. Сдвиги могут выполняться только блоками из двух однотипных микроопераций с входными данными «1», «0». Первоначально в регистр загружен код 00000000. Определить, какую последовательность блоков микроопераций необходимо выполнить, чтобы сформировать в регистре значение 00111110.

**Задача 11.** Восьмиразрядный регистр сдвига выполняет микрооперации сдвига вправо на один разряд и сдвига влево на два разряда. Сдвиги могут выполняться только блоками из двух однотипных микроопераций с входными данными «1», «0». Первоначально в регистр загружен код 11111111. Определить, какую последовательность блоков микроопераций необходимо выполнить, чтобы сформировать в регистре значение 00111100.

**Задача 12.** Восьмиразрядный регистр сдвига выполняет микрооперации сдвига вправо на один разряд и сдвига влево на два разряда. Сдвиги могут выполняться только блоками из двух однотипных микроопераций с входными данными «1», «0». Первоначально в регистр загружен код 00001111. Определить, какую последовательность блоков микроопераций необходимо выполнить, чтобы сформировать в регистре значение 00110011.

**Задача 13.** Разработать двухразрядный регистр на E1E2-триггерах, таблица переходов которых приведена ниже. Регистр должен выполнять две микрооперации: параллельную загрузку кода по входу  $C$  регистра и загрузку инверсного кода, хранимого в регистре, также по входу  $C$ .

| $E1(t)$ | $E2(t)$ | $Q(t+1)$ |
|---------|---------|----------|
| 0       | 0       | $Q(t)$   |
| 0       | 1       | 0        |
| 1       | 0       | 1        |
| 1       | 1       | 0        |

**Задача 14.** Разработать двухразрядный регистр на E1E2-триггерах, таблица переходов которых приведена ниже. Регистр должен выполнять две микрооперации: параллельную загрузку кода по входу  $C$  регистра и последовательную загрузку по входу  $D0$ , также по входу  $C$ .

| $E1(t)$ | $E2(t)$ | $Q(t+1)$ |
|---------|---------|----------|
| 0       | 0       | $Q(t)$   |
| 0       | 1       | 0        |
| 1       | 0       | 1        |
| 1       | 1       | 0        |

**Задача 15.** Спроектировать двухразрядный регистр на E1E2-триггерах, таблица переходов которых приведена ниже. Регистр должен выполнять две микрооперации: параллельную загрузку кода по входу  $C$  регистра и загрузку инверсного кода, хранимого в регистре, также по входу  $C$ .

| $E1(t)$ | $E2(t)$ | $Q(t+1)$ |
|---------|---------|----------|
| 0       | 0       | 0        |
| 0       | 1       | $Q(t)$   |
| 1       | 0       | 1        |
| 1       | 1       | 1        |

### 2.2.6. Построение синхронных счётчиков на регистрах сдвига с линейными и нелинейными обратными связями

**Задача.** Разработать счётчик с модулем, заданным вариантом, на регистре сдвига с линейными или нелинейными обратными связями.

| Вариант | Тип счетчика                    | Модуль |
|---------|---------------------------------|--------|
| 1       | Счётчик по модулю $2^n-1$       | 15     |
| 2       | «»                              | 31     |
| 3       | Счётчик по модулю $2^n-2$       | 14     |
| 4       | «»                              | 30     |
| 5       | Счётчик по модулю $2^n-4$       | 4      |
| 6       | «»                              | 28     |
| 7       | Счётчик по модулю $2^n$         | 8      |
| 8       | «»                              | 16     |
| 9       | «»                              | 32     |
| 10      | Счётчик по произвольному модулю | 9      |
| 11      | «»                              | 13     |
| 12      | «»                              | 15     |
| 13      | «»                              | 23     |
| 14      | «»                              | 25     |
| 15      | «»                              | 29     |

### 2.2.7. Проектирование счётчиков с заданным модулем счета

**Задача 1.** Разработать схему синхронного реверсивного двух-разрядного счётчика с модулем  $M=3$  при ограничениях, заданных вариантом. Счётчик имеет вход синхронизации  $C$ , асинхронный вход установки начального состояния  $CLR$ , управляющий вход  $Y$ , выходы  $Q1, Q0$  и выход переноса/заёма  $CR$ .

Для проверки работы схемы построить временную диаграмму.

Вариант 1. Вход  $Y$  управляет направлением счёта: при  $Y=0$  – вычитание,  $Y=1$  – суммирование. В качестве элемента памяти использовать JK-триггер. Начальное состояние  $Q1, Q0 = 00$ .

Вариант 2. Вход  $Y$  управляет направлением счёта: при  $Y=1$  – вычитание,  $Y=0$  – суммирование. В качестве элемента памяти использовать JK-триггер. Начальное состояние  $Q1, Q0 = 11$ .

Вариант 3. Вход  $Y$  управляет направлением счёта: при  $Y=0$  – вычитание,  $Y=1$  – суммирование. В качестве элемента памяти использовать JK-триггер. Начальное состояние  $Q1, Q0 = 11$ .

Вариант 4. Вход  $Y$  управляет направлением счёта: при  $Y=1$  – вычитание,  $Y=0$  – суммирование. В качестве элемента памяти использовать JK-триггер. Начальное состояние  $Q1, Q0 = 10$ .

Вариант 5. Вход  $Y$  управляет направлением счёта: при  $Y=1$  – вычитание,  $Y=0$  – суммирование. В качестве элемента памяти использовать DV-триггер. Начальное состояние  $Q1, Q0 = 01$ .

Вариант 6. Вход  $Y$  управляет направлением счёта: при  $Y=0$  – вычитание,  $Y=1$  – суммирование. В качестве элемента памяти использовать DV-триггер. Начальное состояние  $Q1, Q0 = 11$ .

Вариант 7. Вход  $Y$  управляет направлением счёта: при  $Y=1$  – вычитание,  $Y=0$  – суммирование. В качестве элемента памяти использовать DV-триггер. Начальное состояние  $Q1, Q0 = 00$ .

Вариант 8. Вход  $Y$  управляет направлением счёта: при  $Y=0$  – вычитание,  $Y=1$  – суммирование. В качестве элемента памяти использовать DV-триггер. Начальное состояние  $Q1, Q0 = 01$ .

Вариант 9. Вход  $Y$  управляет направлением счёта: при  $Y=0$  – вычитание,  $Y=1$  – суммирование. В качестве элемента памяти использовать Т-триггер. Начальное состояние  $Q1, Q0 = 11$ .

Вариант 10. Вход  $Y$  управляет направлением счёта: при  $Y=1$  – вычитание,  $Y=0$  – суммирование. В качестве элемента памяти использовать Т-триггер. Начальное состояние  $Q1, Q0 = 00$ .

Вариант 11. Вход  $Y$  управляет направлением счёта: при  $Y=0$  – вычитание,  $Y=1$  – суммирование. В качестве элемента памяти использовать Т-триггер. Начальное состояние  $Q1, Q0 = 01$ .

Вариант 12. Вход  $Y$  управляет направлением счёта: при  $Y=1$  – вычитание,  $Y=0$  – суммирование. В качестве элемента памяти использовать Т-триггер. Начальное состояние  $Q1, Q0 = 11$ .

Вариант 13. Вход  $Y$  управляет направлением счёта: при  $Y=1$  – вычитание,  $Y=0$  – суммирование. В качестве элемента памяти использовать D-триггер. Начальное состояние  $Q1, Q0 = 00$ .

Вариант 14. Вход  $Y$  управляет направлением счёта: при  $Y=0$  – вычитание,  $Y=1$  – суммирование. В качестве элемента памяти использовать D-триггер. Начальное состояние  $Q1, Q0 = 01$ .

Вариант 15. Вход  $Y$  управляет направлением счёта: при  $Y=1$  – вычитание,  $Y=0$  – суммирование. В качестве элемента памяти использовать D-триггер. Начальное состояние  $Q1, Q0 = 11$ .

**Задача 2.** Разработать схему счётчика по модулю  $M$  на основе четырёхразрядного двоичного регистра сдвига и мультиплексора М4\_1.

| Вариант | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8 |
|---------|----|----|----|----|----|----|----|---|
| $M$     | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 |

**Задача 3.** Разработать схему счётчика по модулю  $M$  на основе трёхразрядного двоичного регистра сдвига и мультиплексора М4\_1.

| Вариант | 1 | 2 | 3 | 4 |
|---------|---|---|---|---|
| $M$     | 8 | 7 | 6 | 5 |

**Задача 4.** Разработать схему реверсивного счётчика по модулю  $M$  на основе трёхразрядного двоичного регистра сдвига и мультиплексора М4\_1.

| Вариант | 1 | 2 | 3 |
|---------|---|---|---|
| $M$     | 7 | 6 | 5 |

## 2.2.8. Проектирование пересчётных схем

**Задача 1.** Разработать схему преобразователя частоты, который выполняет набор операций прореживания импульсной последовательности  $C$  в зависимости от управляющих сигналов  $Y1, Y0$  в соответствии с таблицей.

| $Y1$ | $Y0$ | $Q$    |
|------|------|--------|
| 0    | 0    | $C/N0$ |
| 0    | 1    | $C/N1$ |
| 1    | 0    | $C/N2$ |
| 1    | 1    | $C/N3$ |

При выполнении прореживания по правилу  $C/N$  на выход устройства  $Q$  проходит каждый  $N$ -й импульс исходной последовательности  $C$ .

| Вариант | $N0$ | $N1$ | $N2$ | $N3$ |
|---------|------|------|------|------|
| 1       | 12   | 11   | 7    | 5    |
| 2       | 11   | 9    | 6    | 3    |
| 3       | 13   | 10   | 5    | 2    |
| 4       | 10   | 9    | 4    | 3    |
| 5       | 9    | 7    | 6    | 4    |
| 6       | 11   | 8    | 3    | 2    |
| 7       | 7    | 6    | 4    | 3    |
| 8       | 14   | 11   | 9    | 5    |
| 9       | 12   | 10   | 6    | 3    |
| 10      | 10   | 7    | 5    | 2    |
| 11      | 9    | 6    | 5    | 4    |
| 12      | 6    | 5    | 4    | 3    |
| 13      | 8    | 5    | 3    | 2    |
| 14      | 10   | 8    | 6    | 4    |
| 15      | 12   | 10   | 8    | 6    |

**Задача 2.** Разработать схему делителя частоты на  $N$  и скважностью  $M$  (скважность это отношение периода следования к длительности импульса).

Для каждого варианта дополнительным условием может быть требование реализации схемы на основе:

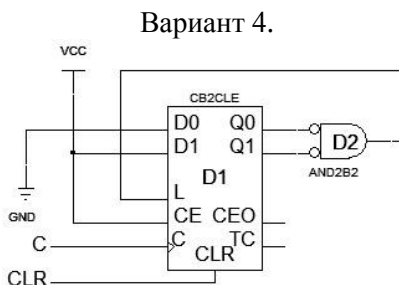
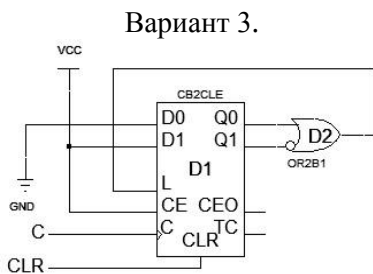
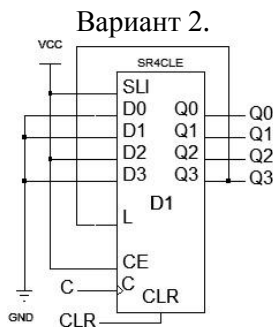
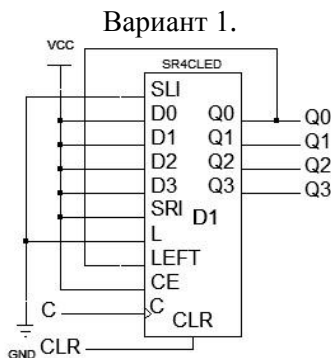
- JK-, DV- или Т-триггера,
- двоичного счётчика,
- регистра сдвига.

| Вариант | $N$ | $M$  |
|---------|-----|------|
| 1       | 5   | 2,5  |
| 2       | 5   | 1,25 |
| 3       | 6   | 2    |
| 4       | 6   | 3    |
| 5       | 7   | 3,5  |
| 6       | 7   | 1,75 |
| 7       | 8   | 4    |

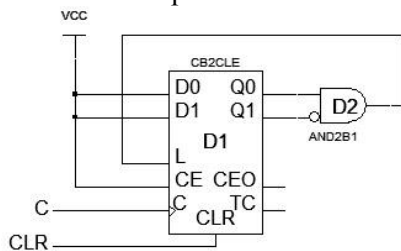
| Вариант | $N$ | $M$ |
|---------|-----|-----|
| 8       | 8   | 1,6 |
| 9       | 9   | 3   |
| 10      | 10  | 2,5 |
| 11      | 12  | 4   |
| 12      | 12  | 3   |
| 13      | 12  | 6   |
| 14      | 14  | 7   |
| 15      | 14  | 2   |

### 2.2.9. Построение временных диаграмм работы последовательностных схем на основе счётчиков и регистров

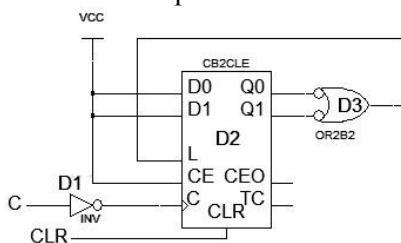
**Задача.** Построить временную диаграмму работы схемы до момента повторения последовательности состояний счётчика или регистра.



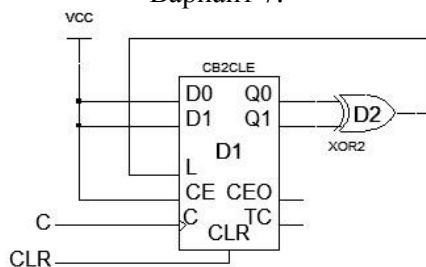
Вариант 5.



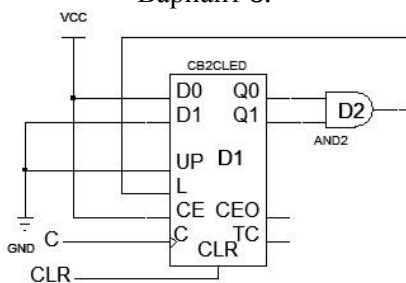
Вариант 6.



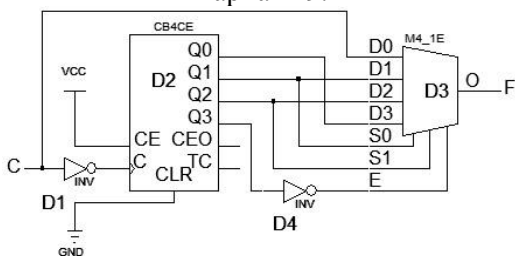
Вариант 7.



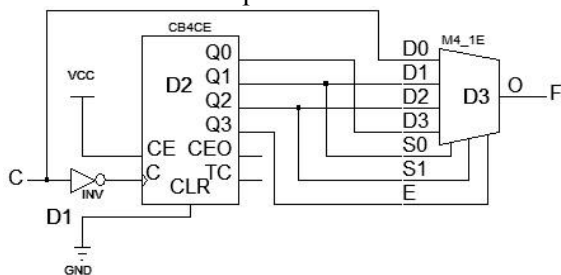
Вариант 8.



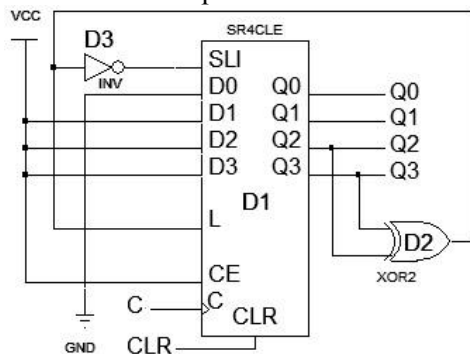
Вариант 9.



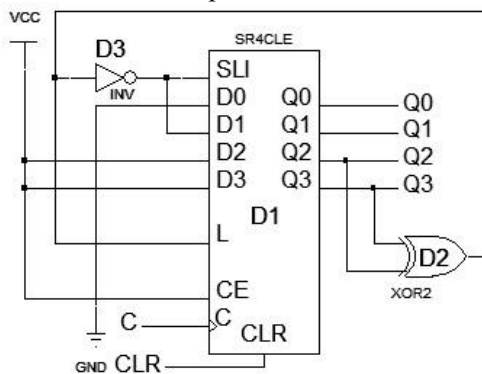
Вариант 10.



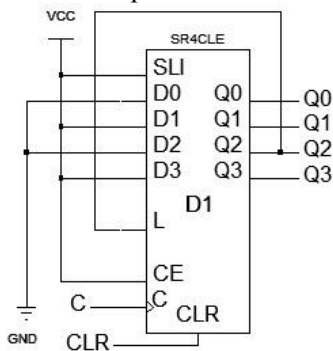
Вариант 11.



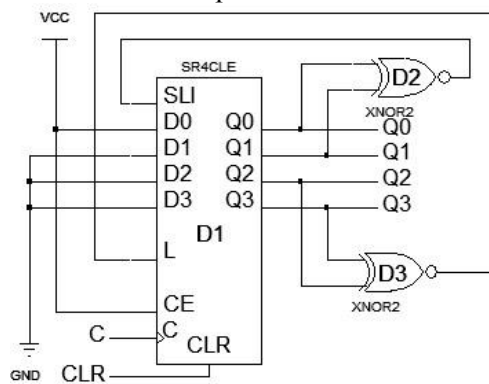
Вариант 12.



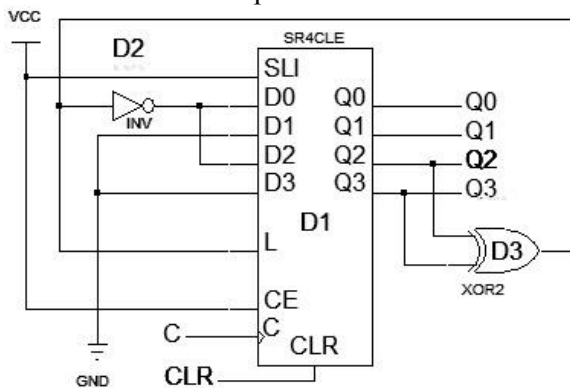
Вариант 13.



Вариант 14.



Вариант 15.

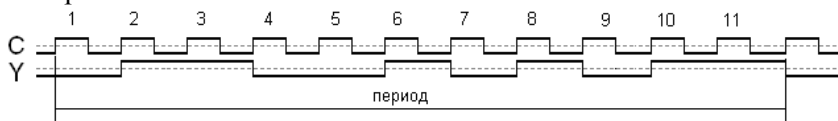


### 2.2.10. Синтез генераторов последовательностей сигналов по заданной временной диаграмме

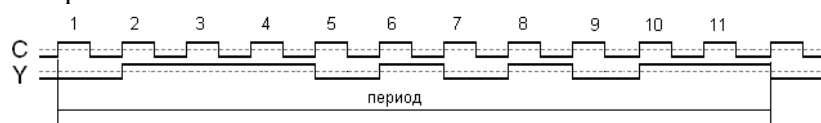
**Задача 1.** Разработать схему генератора последовательности сигналов по заданной временной диаграмме. Период последовательности задается типовым двоичным счётчиком. Цифры в варианте обозначают номер такта (состояние счётчика).

Для проверки работы схемы построить временную диаграмму.

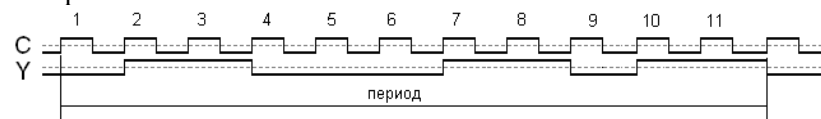
Вариант 1.



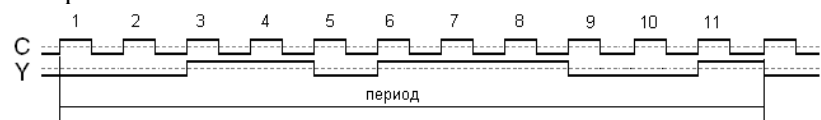
Вариант 2.



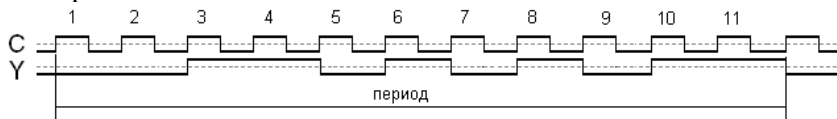
Вариант 3.



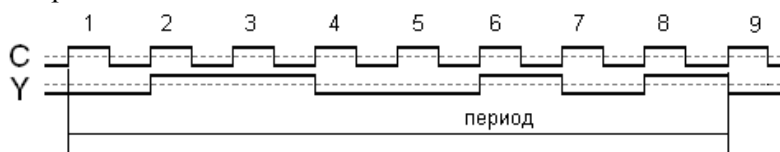
Вариант 4.



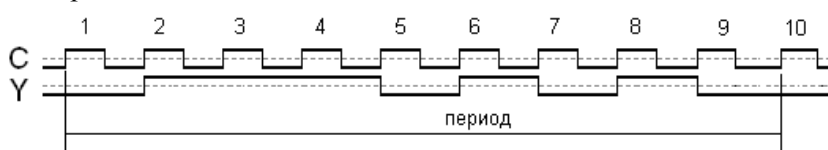
Вариант 5.



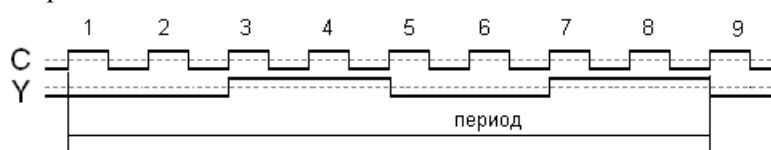
Вариант 6.



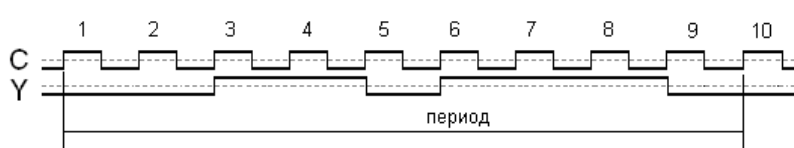
Вариант 7.



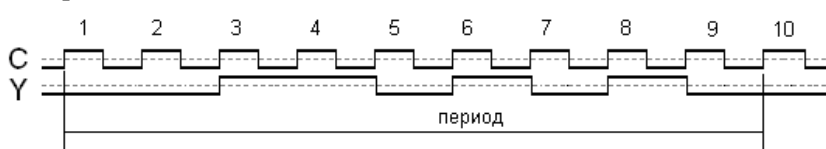
Вариант 8.



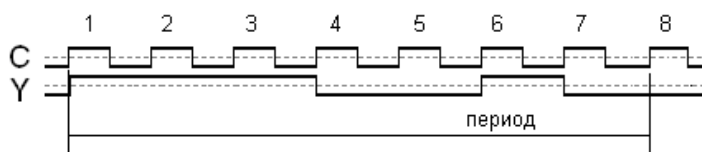
Вариант 9.



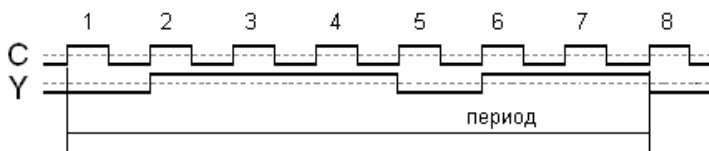
Вариант 10.



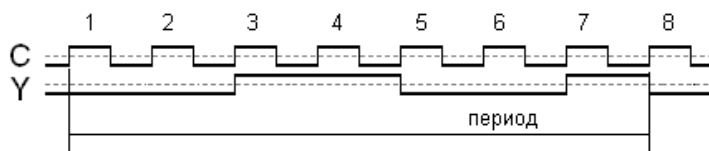
Вариант 11.



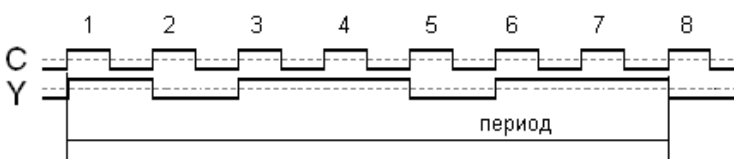
Вариант 12.



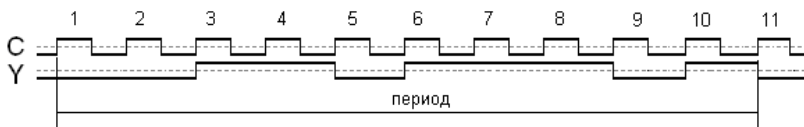
Вариант 13.



Вариант 14.



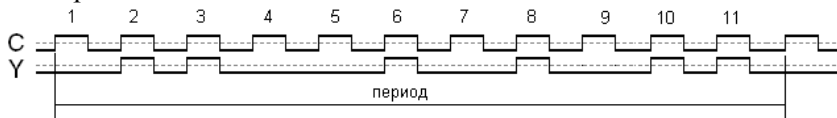
Вариант 15.



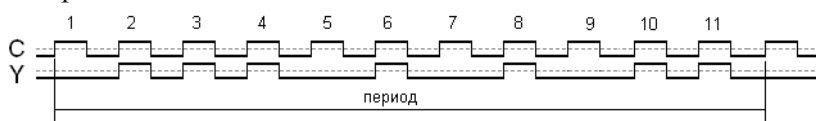
**Задача 2.** Разработать схему генератора последовательности сигналов по заданной временной диаграмме. Период последовательности задается типовым двоичным счётчиком. *Цифры в варианте обозначают номер такта (состояние счётчика).*

Для проверки работы схемы построить временную диаграмму.

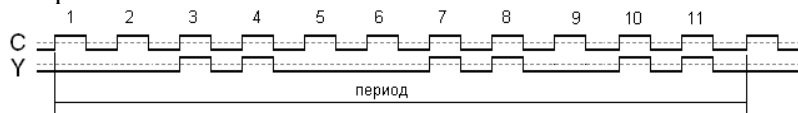
Вариант 1.



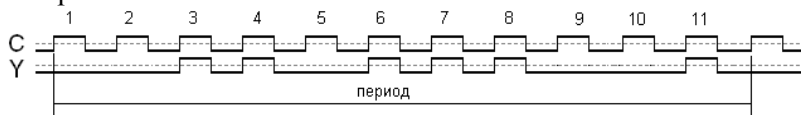
### Вариант 2.



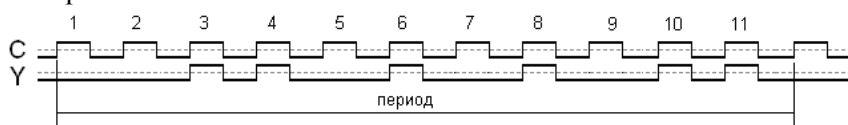
### Вариант 3.



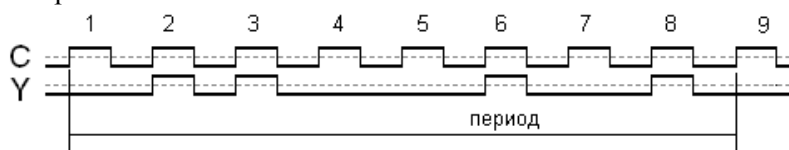
### Вариант 4.



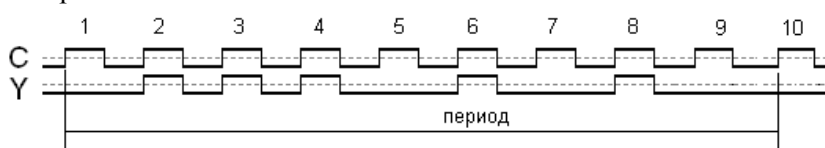
### Вариант 5.



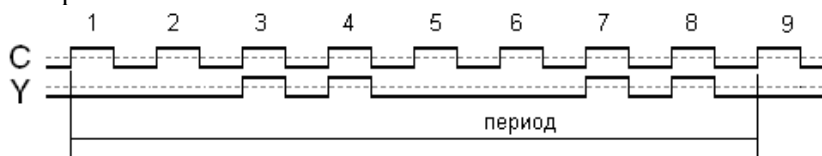
### Вариант 6.



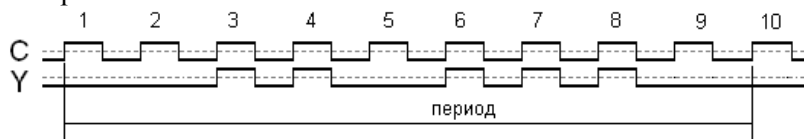
### Вариант 7.



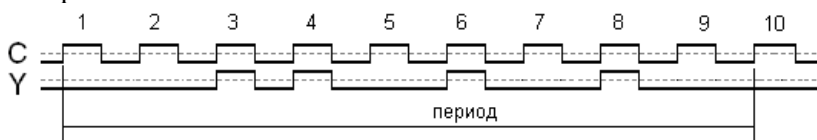
### Вариант 8.



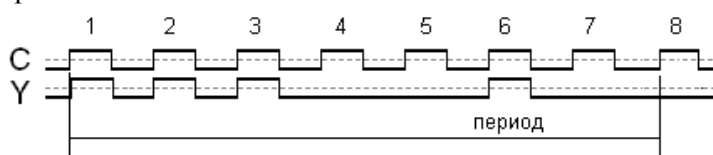
Вариант 9.



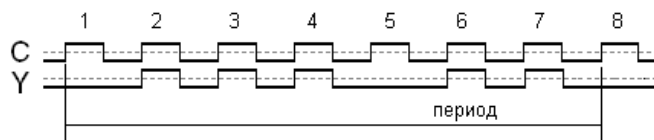
Вариант 10.



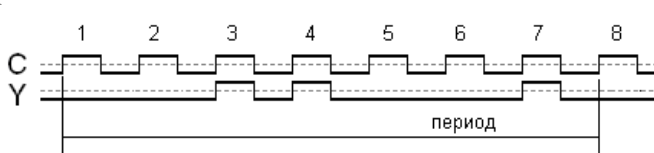
Вариант 11.



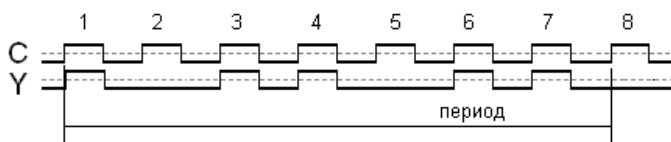
Вариант 12.



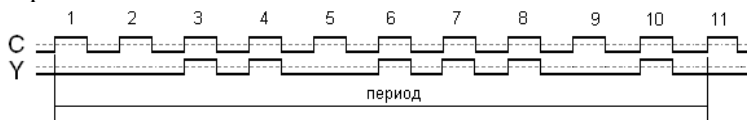
Вариант 13.



Вариант 14.

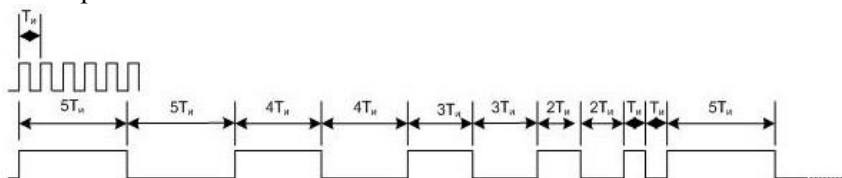


### Вариант 15.

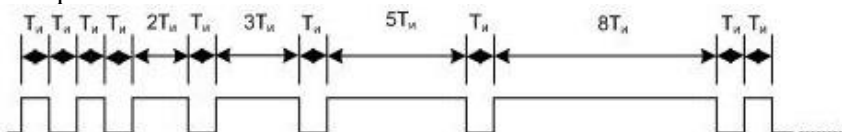


**Задача 3.** Разработать устройство, формирующее периодическую последовательность временных интервалов, синхронизируемую тактовыми сигналами по закону, представленному на временной диаграмме.

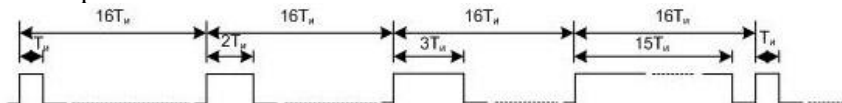
### Вариант 1.



### Вариант 2.

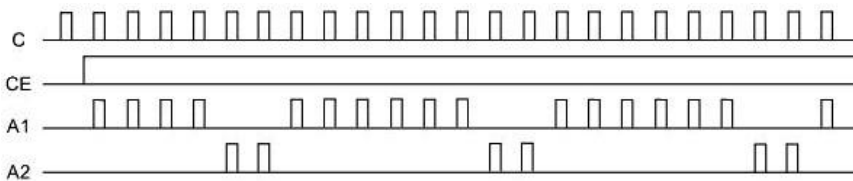


### Вариант 3.

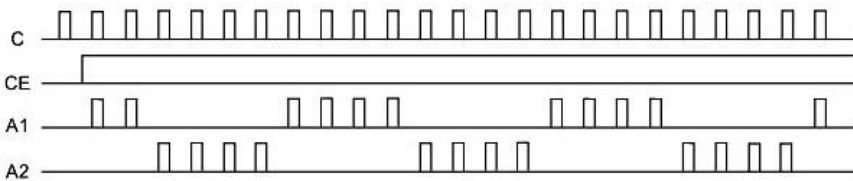


**Задача 4.** Разработать устройство, которое при наличии сигнала разрешения  $CE$  распределяет входные импульсы  $C$  на два выхода  $A1$  и  $A2$  в соответствии с диаграммой, заданной вариантом. При отсутствии сигнала  $CE$  выходы устройства устанавливаются в состояние «0». Сигнал  $CE$  разрешает распределение, начиная с выхода  $A1$ .

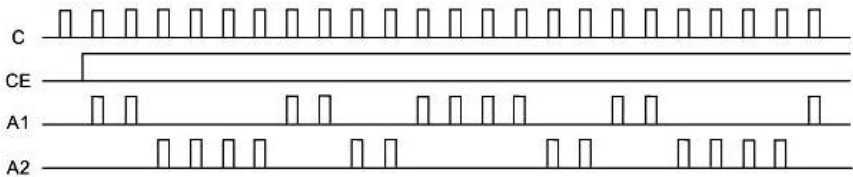
### Вариант 1.



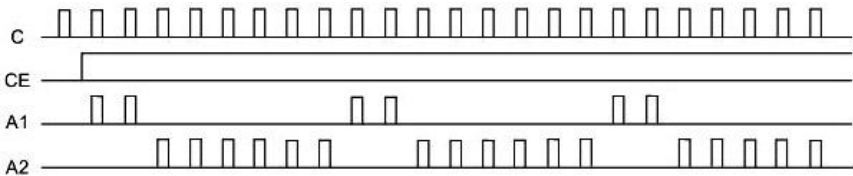
### Вариант 2.



### Вариант 3.

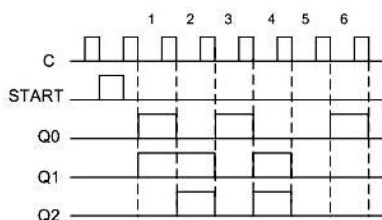


### Вариант 4.

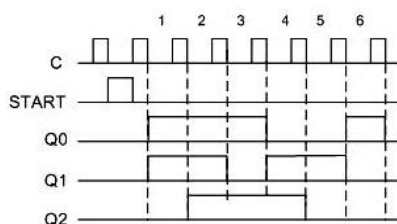


**Задача 5.** Разработать устройство, которое по сигналу *START*, поступающему в отсутствие синхросигналов *C*, однократно вырабатывает серию выходных сигналов *Q2, Q1, Q0* в соответствии с временной диаграммой, заданной вариантом.

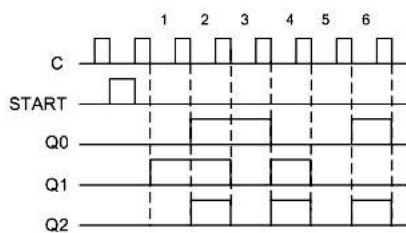
Вариант 1.



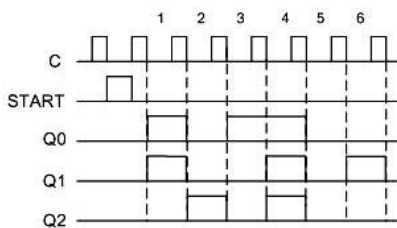
Вариант 2.



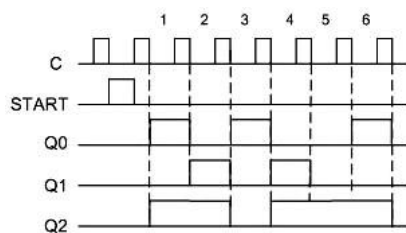
Вариант 3.



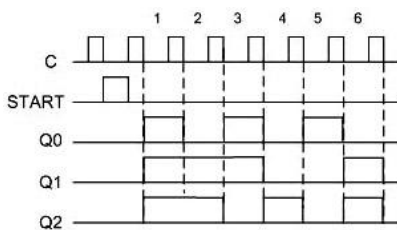
Вариант 4.



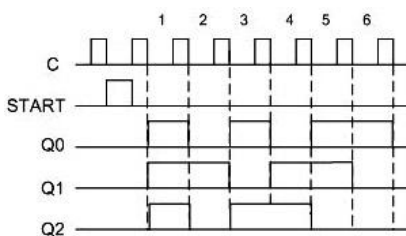
Вариант 5.



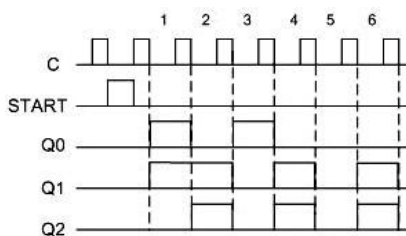
Вариант 6.



Вариант 7.



Вариант 8.



### 2.2.11. Синтез генераторов последовательностей сигналов

**Задача 1.** Разработать схему, вырабатывающую по синхросигналу  $C$  в зависимости от значения входного сигнала  $Y$  различные циклические последовательности выходных сигналов  $Q1, Q0$ . В качестве элемента памяти использовать JK-триггер. Схема должна иметь асинхронный вход  $CLR$  установки в начальное состояние.

| Вариант | Последовательность комбинаций сигналов $Q1, Q0$ при |           |
|---------|-----------------------------------------------------|-----------|
|         | $Y = 0$                                             | $Y = 1$   |
| 1       | 0 1 2 3 0                                           | 0 3 2 1 0 |
| 2       | 0 1 3 2 0                                           | 0 2 3 1 0 |
| 3       | 0 2 1 3 0                                           | 0 3 1 2 0 |
| 4       | 0 2 3 1 0                                           | 0 1 3 2 0 |
| 5       | 0 3 1 2 0                                           | 0 2 1 3 0 |
| 6       | 0 3 2 1 0                                           | 0 1 2 3 0 |
| 7       | 1 0 2 3 1                                           | 1 3 2 0 1 |
| 8       | 1 0 3 2 1                                           | 1 2 3 0 1 |
| 9       | 1 2 3 0 1                                           | 1 0 3 2 1 |
| 10      | 1 2 0 3 1                                           | 1 3 0 2 1 |
| 11      | 1 3 0 2 1                                           | 1 2 0 3 1 |
| 12      | 1 3 2 0 1                                           | 1 0 2 3 1 |
| 13      | 2 0 1 3 2                                           | 2 3 1 0 2 |
| 14      | 2 0 3 1 2                                           | 2 1 3 0 2 |
| 15      | 2 1 0 3 2                                           | 2 3 0 2 1 |

**Задача 2.** Разработать устройство, которое имеет четыре входа ( $START, C, X1, X0$ ) и выход  $Y$ . После прихода сигнала начала работы на вход  $START$  (в паузе между импульсами синхронизации  $C$ ) устройство однократно и синхронно с импульсной последовательностью  $C$  формирует последовательности импульсных сигналов  $Y$  согласно таблице, заданной вариантом.

Вариант 1.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 0 (нет сигналов) |
| 0     | 1     | 1100101          |
| 1     | 0     | 1011             |
| 1     | 1     | 0 (нет сигналов) |

Вариант 2.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 10101011         |
| 0     | 1     | 0 (нет сигналов) |
| 1     | 0     | 10101            |
| 1     | 1     | 0 (нет сигналов) |

Вариант 3.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 10101011         |
| 0     | 1     | 0 (нет сигналов) |
| 1     | 0     | 1011             |
| 1     | 1     | 0 (нет сигналов) |

Вариант 4.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 10101011         |
| 0     | 1     | 0011             |
| 1     | 0     | 1011             |
| 1     | 1     | 0 (нет сигналов) |

Вариант 5.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 10000011         |
| 0     | 1     | 0 (нет сигналов) |
| 1     | 0     | 10101            |
| 1     | 1     | 10110111         |

Вариант 6.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 10101011         |
| 0     | 1     | 0011             |
| 1     | 0     | 0 (нет сигналов) |
| 1     | 1     | 0011             |

Вариант 7.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 10101011         |
| 0     | 1     | 0 (нет сигналов) |
| 1     | 0     | 1011             |
| 1     | 1     | 0 (нет сигналов) |

Вариант 8.

| $X_1$ | $X_0$ | $Y$  |
|-------|-------|------|
| 0     | 0     | 101  |
| 0     | 1     | 011  |
| 1     | 0     | 111  |
| 1     | 1     | 0111 |

Вариант 9.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 0 (нет сигналов) |
| 0     | 1     | 0111             |
| 1     | 0     | 1011             |
| 1     | 1     | 1111             |

Вариант 10.

| $X_1$ | $X_0$ | $Y$    |
|-------|-------|--------|
| 0     | 0     | 101011 |
| 0     | 1     | 01101  |
| 1     | 0     | 1011   |
| 1     | 1     | 0101   |

Вариант 11.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 0 (нет сигналов) |
| 0     | 1     | 1011             |
| 1     | 0     | 0 (нет сигналов) |
| 1     | 1     | 100111           |

Вариант 12.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 11011            |
| 0     | 1     | 0 (нет сигналов) |
| 1     | 0     | 0 (нет сигналов) |
| 1     | 1     | 10101            |

Вариант 13.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 1010             |
| 0     | 1     | 10111            |
| 1     | 0     | 101101           |
| 1     | 1     | 0 (нет сигналов) |

Вариант 14.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 1011             |
| 0     | 1     | 11011            |
| 1     | 0     | 101101           |
| 1     | 1     | 0 (нет сигналов) |

Вариант 15.

| $X_1$ | $X_0$ | $Y$              |
|-------|-------|------------------|
| 0     | 0     | 111001           |
| 0     | 1     | 1101             |
| 1     | 0     | 0 (нет сигналов) |
| 1     | 1     | 101              |

**Задача 3.** Разработать схему, которая с приходом разрешающего потенциального сигнала на вход *START* вырабатывает на выходе  $Y$  синхронно с импульсами синхронизации  $C$  одну из следующих последовательностей импульсных сигналов:

1. 10011
2. 110010011

Выбор требуемой последовательности определяется потенциальными сигналами, поданными на два входа  $X_1, X_0$  схемы: код  $X_1, X_0 = 01$  соответствует первой импульсной последовательности сигналов; код  $X_1, X_0 = 10$  – второй последовательности.

После выдачи требуемой последовательности импульсных сигналов схема прекращает работу (сигналы на входах *START*,  $X_1$ ,  $X_0$  остаются неизменными). Повторная выдача сигналов возможна только при повторной подаче сигнала *START*.

При значениях 00 и 11 на входах  $X_1$  и  $X_0$  схема не должна выдавать сигналы. Никакие другие сигналы на схему не подаются.

**Задача 4.** Разработать схему, которая при соответствующем коде на входах  $X_1$  и  $X_0$  с приходом разрешающего потенциального сигнала на вход  $START$  вырабатывает синхронно с импульсами синхронизации  $C$  на одном из двух выходов  $Y_1$  и  $Y_2$  последовательность импульсных сигналов:

при значениях  $X_1, X_0 = 10$  на выходе  $Y_1$  – 1010001110111;

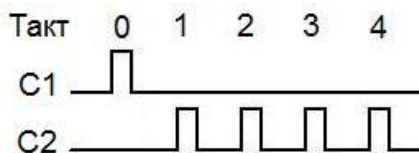
при значениях  $X_1, X_0 = 01$  на выходе  $Y_2$  – 110010001.

После выдачи требуемой последовательности импульсных сигналов схема прекращает работу (сигналы на входах  $START$ ,  $X_1$ ,  $X_0$  остаются неизменными). Повторная выдача сигналов возможна только при повторной подаче сигнала  $START$ .

При значениях 00 и 11 на входах  $X_1$  и  $X_0$  схема не должна выдавать сигналы. Никакие другие сигналы на схему не подаются.

**Задача 5.** Разработать схему, выдающую на один из четырёх выходов  $Y_4$ – $Y_1$  непрерывную серию импульсов. Номер выхода и число импульсов в серии определяется числом единиц в 4-разрядном двоичном коде  $X_3, X_2, X_1, X_0$ , подаваемом на схему. Например, если код равен 1010, то на выходе  $Y_2$  должно выработаться два импульса в первом и во втором тактах.

Кроме входов для исходного кода  $X_3, X_2, X_1, X_0$  схема должна иметь ещё два входа  $C_1$  и  $C_2$ , на которые подаются пять импульсов в соответствии с временной диаграммой.



Исходный код  $X_3, X_2, X_1, X_0$  остается неизменным в течение всей приведенной временной диаграммы. При проектировании положить, что код 0000 не появляется на входах схемы.

**Задача 6.** Разработать схему, выдающую в соответствующем такте на один из четырёх выходов  $Y_3$ – $Y_0$  один импульс. Номер такта и номер шины определяется числом единиц в 3-разрядном двоичном коде  $X_2, X_1, X_0$ , подаваемом на схему. Например, при коде 000 схема должна выдать импульс на выходе  $Y_0$  в нулевом такте, при коде 110 импульс должен появиться во втором такте на выходе  $Y_2$  и т.д.

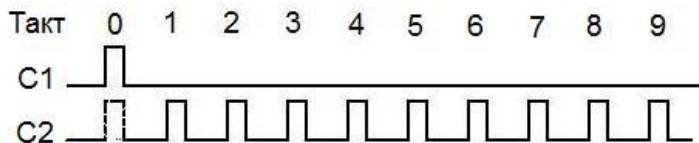
Кроме входов для исходного кода  $X_2, X_1, X_0$  схема должна иметь ещё два входа  $C_1$  и  $C_2$ , на которые подаются пять импульсов в соответствии с временной диаграммой (см. рис. к задаче 5).

Исходный код  $X_2, X_1, X_0$  остается неизменным в течение всей приведенной временной диаграммы.

**Задача 7.** Разработать схему, передающую синхронно с тактовым импульсом  $C_2$ :

состояние информационного входа 1 на выход 1 в первом такте,  
состояние информационного входа 2 на выход 1 во втором такте,  
состояние информационного входа 4 на выход 1 в третьем такте,  
состояние информационного входа 4 на выход 2 в четвёртом такте,  
состояние информационного входа 4 на выход 4 в пятом такте,  
состояние информационного входа 3 на выход 4 в шестом такте,  
состояние информационного входа 1 на выход 4 в седьмом такте,  
состояние информационного входа 1 на выход 3 в восьмом такте.

Кроме четырёх информационных входов схема должна иметь ещё два управляющих входа  $C_1$  и  $C_2$ , на которые подаются девять (десять) импульсов в соответствии с временной диаграммой. Первый импульс на входе  $C_2$  (такт 0) можно при необходимости считать отсутствующим.



**Задача 8.** Разработать устройство, которое имеет 10 входов и три выхода  $Q_2$ ,  $Q_1$  и  $Q_0$ . На входы 1-8 поступают битовые векторы ( $X_3, X_2, X_1, X_0$ ) и ( $Y_3, Y_2, Y_1, Y_0$ ). На вход  $C$  поступают импульсы син-

хронизации. Вход *START* предназначен для сигналов начала работы, которые могут поступать только в паузах между сигналами синхронизации.

При подаче сигнала *START* устройство выполняет один трёх-тактный цикл формирования сигналов на выходах *Q2*, *Q1* и *Q0*. С приходом первого после сигнала *START* импульса *C* фиксируются значения векторов ( $X_3, X_2, X_1, X_0$ ) и ( $Y_3, Y_2, Y_1, Y_0$ ) и в последующих трёх тактах устройство передаёт на один из выходов сигнал *C* по правилу, соответствующему таблице.

| Такт | Выходные сигналы |                 |                 |                 |                 |                 |
|------|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|      | <i>Q2</i>        |                 | <i>Q1</i>       |                 | <i>Q0</i>       |                 |
|      | $Y_1 \cdot Y_3$  | $Y_0 \cdot Y_2$ | $Y_0 \cdot Y_3$ | $Y_2 \cdot Y_3$ | $Y_0 \cdot Y_3$ | $Y_1 \cdot Y_2$ |
| 1    | 0                | $X_1 \cdot X_2$ | 0               | 0               | $X_0 \cdot X_1$ | 0               |
| 2    | 0                | 0               | $X_1 \cdot X_2$ | $X_2 \cdot X_3$ | 0               | 0               |
| 3    | $X_2 \cdot X_3$  | 0               | 0               | 0               | 0               | $X_0 \cdot X_1$ |

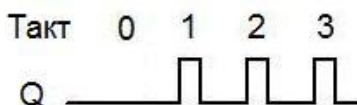
**Задача 9.** Разработать схему, которая в зависимости от числа единиц в двоичном коде, поступающем на входы устройства  $X_2, X_1, X_0$ , выдаёт на выход такое же количество импульсов.

При кодах 001, 010, 100 схема должна выдать импульс в четвёртом такте, при кодах 101, 011, 110 – в третьем и четвёртом тактах, при коде 111 – во втором, третьем и четвёртом тактах, при коде 000 – в первом такте. Схема имеет ещё два входа *C1* и *C2*, на которые подаются только пять импульсов в соответствии с временной диаграммой (см. рис. к задаче 5). Код  $X_2, X_1, X_0$  сохраняется неизменным в течение всей временной диаграммы.

**Задача 10.** Разработать схему, выдающую на выход *Q* импульс в такте, номер которого определяется числом единиц в 4-разрядном двоичном коде, подаваемом на схему. Кроме входов  $X_3, X_2, X_1, X_0$  для исходного кода схема должна иметь ещё два входа *C1* и *C2*, на которые подаются пять импульсов в соответствии с временной диаграммой (см. рис. к задаче 5).

Например, при коде 1011 схема должна выдать импульс в третьем такте. При проектировании считать, что код 0000 не появляется на входах схемы. Исходный код остается неизменным в течение всей временной диаграммы.

**Задача 11.** Разработать схему, выдающую на выход  $Q$  серию импульсов (без пропуска), число импульсов в которой равно количеству единиц в 4-разрядном коде, подаваемом на входы  $X_3, X_2, X_1, X_0$  схемы. Схема имеет ещё два входа  $C1$  и  $C2$ , на которые подаются только пять импульсов в соответствии с временной диаграммой (см. рис. к задаче 5). Например, при коде 1011 на выходе  $Q$  схемы должна быть следующая временная диаграмма:

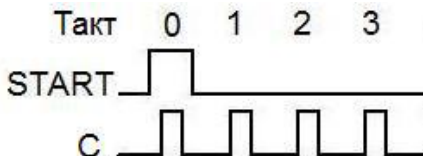


При коде 0000 схема не должна выдавать сигналы. Исходный код остается неизменным в течение всей временной диаграммы.

**Задача 12.** Разработать схему, имеющую три информационных входа  $X_2, X_1, X_0$ , входы  $START$  и  $C$  и один выход  $Q$ . Сигнал на выходе схемы должен формироваться синхронно с синхросигналами  $C$  согласно следующим условиям:

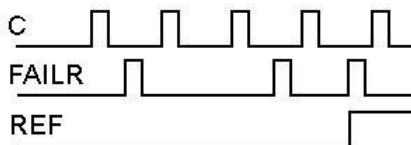
- в первом такте, если на информационных входах имеется одна единица и два нуля;
- в первом и во втором такте, если на информационных входах имеются две единицы и один ноль;
- в первом, втором и третьем тактах, если на всех трёх информационных входах имеются единицы.

Временная диаграмма управляющих сигналов приведена на рисунке. Информационные сигналы в процессе работы схемы не меняются.

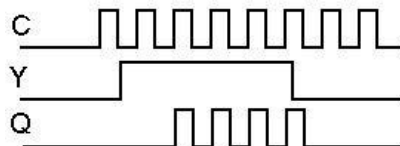


**Задача 13.** На вход схемы поступает периодический синхросигнал  $C$ , в паузе которого возможно появление одиночного импульса  $FAILR$ , свидетельствующего о внешнем сбое.

Разработать схему, выдающую сигнал  $REF$  при появлении сигнала  $FAILR$  в двух смежных циклах  $C$ .



**Задача 14.** Разработать синхронную последовательностную схему, в которой синхронизирующий сигнал  $C$  стробируется сигналом  $Y$ . При стробировании на выходе схемы  $Q$  с задержкой на один период следования импульсов  $C$  вырабатываются только полные синхроимпульсы. Временная диаграмма работы схемы приведена на рисунке.



**Задача 15.** Разработать схему, которая выдает на выход  $Q$  импульс в первом такте, если:

а) в двоичном коде  $X_3, X_2, X_1, X_0$ , поданном на схему, разряд  $X_0 = 1$ , а состояние двух других входов схемы  $Y_1, Y_0 = 00$ ;

б) в двоичном коде  $X_3, X_2, X_1, X_0$  разряд  $X_1 = 1$ , а  $Y_1, Y_0 = 01$ .

Схема должна выдавать на выход  $Q$  импульс в четвёртом такте, если:

а) разряд  $X_2 = 1$ , а  $Y_1, Y_0 = 10$ ;

б) разряд  $X_3 = 1$ , а  $Y_1, Y_0 = 11$ .

Схема имеет ещё два внешних входа  $C1$  и  $C2$ , на которые подаются только пять импульсов в соответствии с временной диаграммой (см. рис. к задаче 5).

Коды  $X_3, X_2, X_1, X_0$  и  $Y_1, Y_0$  поддерживаются неизменными в течение всей временной диаграммы.

### 2.2.12. Синтез устройств анализа цифровых последовательностей

**Задача 1.** Разработать устройство, которое имеет десять входов и четыре выхода. На вход  $C$  поступают синхросигналы. На восемь входов синхронно с сигналами  $C$  поступают битовые векторы  $(X_3, X_2, X_1, X_0)$  и  $(Y_3, Y_2, Y_1, Y_0)$ . Вход  $CLR$  предназначен для установки устройства в начальное состояние (при необходимости). В каждом  $i$ -м такте анализируются матрицы:

$$\left| \begin{array}{cccc} X_3^i, & X_2^i, & X_1^i, & X_0^i \\ X_3^{i-1}, & X_2^{i-1}, & X_1^{i-1}, & X_0^{i-1} \end{array} \right| \quad \left| \begin{array}{cccc} Y_3^i, & Y_2^i, & Y_1^i, & Y_0^i \\ Y_3^{i-1}, & Y_2^{i-1}, & Y_1^{i-1}, & Y_0^{i-1} \end{array} \right|$$

Матрицы составлены из значений входных векторов в  $i$ -м и  $(i-1)$ -м тактах.

Вариант 1. На выходы устройства  $Q_3, Q_2, Q_1, Q_0$  выдаётся тот входной вектор, для которого в соответствующей ему матрице содержится наибольшее количество единиц. При равенстве значений выдаётся тот же самый вектор, который выдавался в предыдущем такте (или произвольный вектор в начале работы устройства).

Вариант 2. На выходы устройства  $Q_3, Q_2, Q_1, Q_0$  выдаётся входной вектор, в котором по сравнению с предыдущим тактом наибольшее количество изменившихся координат. При равенстве значений выдаётся тот же самый вектор, который выдавался в предыдущем такте (или произвольный вектор в начале работы устройства).

Вариант 3. На выходы устройства  $Q_3, Q_2, Q_1, Q_0$  выдаётся входной вектор, для которого сумма со значением в предыдущем такте максимальна. При равенстве значений выдаётся тот же самый вектор, который выдавался в предыдущем такте (или произвольный вектор в начале работы устройства).

Вариант 4. На выходы устройства  $Q_3, Q_2, Q_1, Q_0$  выдаётся входной вектор, для которого разность со значением другого вектора в предыдущем такте максимальна. При равенстве значений выдаётся тот же самый вектор, который выдавался в предыдущем такте (или произвольный вектор в начале работы устройства).

Вариант 5. Задана матрица, которая накладывается на исходные матрицы для векторов  $(X3, X2, X1, X0)$  и  $(Y3, Y2, Y1, Y0)$ :

$$\begin{vmatrix} 0, & 1, & 1, & 0 \\ 1, & 0, & 0, & 1 \end{vmatrix}$$

Подсчитываются количества совпадений для обоих векторов. На выходы устройства Q3, Q2, Q1, Q0 выдаётся тот вектор, для которого количество совпадений максимально. При равенстве значений выдаётся тот же самый вектор, который выдавался в предыдущем такте (или произвольный вектор в начале работы устройства).

Вариант 6. Задана матрица:

$$\begin{vmatrix} 0, & 1, & 1, & 0 \\ 1, & 0, & 0, & 1 \end{vmatrix}$$

Для этой матрицы и матриц входных векторов поэлементно выполняется операция конъюнкции. Подсчитывается количество выделенных единиц для каждого входного вектора. На выходы устройства Q3, Q2, Q1, Q0 выдаётся тот вектор, для которого количество выделенных единиц максимально. При равенстве значений выдаётся тот же самый вектор, который выдавался в предыдущем такте (или произвольный вектор в начале работы устройства).

Вариант 7. Из матриц входных векторов выбираются те элементы, которые расположены в тех же позициях, что и единичные элементы в селектирующей матрице:

$$\begin{vmatrix} 0, & 1, & 0, & 0 \\ 0, & 0, & 1, & 0 \end{vmatrix}$$

Далее над ними выполняется операция сложения по модулю 2. На выходы устройства Q3, Q2, Q1, Q0 выдаётся тот вектор, для которого результат операции равен «1», при условии, что для другого он равен «0». При равенстве значений выдаётся тот же самый вектор, который выдавался в предыдущем такте (или произвольный вектор в начале работы устройства).

**Задача 2.** Разработать устройство, которое имеет пять входов и один выход. Три входа  $X_2$ ,  $X_1$  и  $X_0$  предназначены для последовательного ввода данных. Ввод данных синхронизируется сигналами на входе  $C$ . После прихода сигнала асинхронной установки  $CLR$ , устройство передает на выход  $Q$  синхросигнал  $C$  всякий раз, как только обнаруживает из периодической последовательности очередную кодовую комбинацию заданную вариантом (на входы схемы могут поступать любые двоичные комбинации в любом порядке).

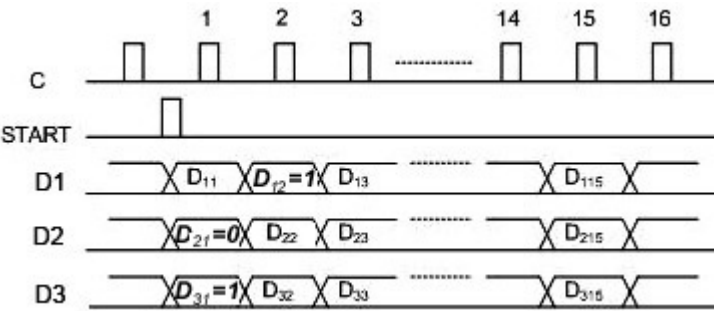
| Вариант | Последовательность обнаружения кодовых комбинаций на входах $X_2$ , $X_1$ , $X_0$ |
|---------|-----------------------------------------------------------------------------------|
| 1       | 001, 011, 101, 111, 001, ...                                                      |
| 2       | 000, 001, 100, 011, 000, ...                                                      |
| 3       | 101, 010, 000, 001, 101, ...                                                      |
| 4       | 010, 011, 001, 000, 010, ...                                                      |
| 5       | 011, 010, 110, 111, 011, ...                                                      |
| 6       | 100, 011, 111, 110, 100, ...                                                      |
| 7       | 010, 101, 100, 001, 010, ...                                                      |
| 8       | 001, 000, 011, 111, 001, ...                                                      |
| 9       | 110, 101, 000, 001, 110, ...                                                      |
| 10      | 110, 111, 000, 011, 110, ...                                                      |
| 11      | 111, 100, 101, 001, 111, ...                                                      |
| 12      | 110, 101, 001, 000, 110, ...                                                      |
| 13      | 101, 100, 111, 011, 101, ...                                                      |
| 14      | 000, 101, 110, 011, 000, ...                                                      |
| 15      | 001, 010, 101, 110, 001, ...                                                      |

**Задача 3.** Разработать устройство, которое имеет пять входов и четыре выхода. Информационные входы  $D_1$ ,  $D_2$  и  $D_3$  служат для последовательного ввода данных. Ввод данных синхронизируется сигналами, поступающими на вход  $C$ , значения данных изменяются в паузах между сигналами  $C$ . Вход  $START$  служит для сигнала начала работы устройства.

После прихода в паузе между сигналами  $C$  сигнала  $START$  устройство в течение последующих 15 периодов  $C$ , начиная со второго (третьего), осуществляет селекцию (выделение) и подсчёт битовых комбинаций на линиях входных данных, заданных вариантом. Например, на рисунке жирным курсивным шрифтом показана ситуа-

ция выделения комбинации  $D1_i = 1, D2_{(i-1)} = 0, D3_{(i-1)} = 1$  ( $2 \leq i \leq 15$ ).  
 Результат подсчёта в такте 16 фиксируется на выходах устройства Q4, Q3, Q2 и Q1.

В обозначении бита входных данных  $Dji$ :  $j$  – определяет номер входной линии данных, а  $i$  – номер такта.



Вариант 1.

| Такт \ Вход | $i-1$ | $i$ |
|-------------|-------|-----|
| D1          | X     | 1   |
| D2          | 0     | X   |
| D3          | 0     | X   |

Вариант 2.

| Такт \ Вход | $i-1$ | $i$ |
|-------------|-------|-----|
| D1          | 1     | X   |
| D2          | X     | 1   |
| D3          | 0     | X   |

Вариант 3.

| Такт \ Вход | $i-1$ | $i$ |
|-------------|-------|-----|
| D1          | 0     | 1   |
| D2          | 0     | X   |
| D3          | 0     | X   |

Вариант 4.

| Такт \ Вход | $i-1$ | $i$ |
|-------------|-------|-----|
| D1          | 1     | 1   |
| D2          | 0     | 1   |
| D3          | 1     | 0   |

Вариант 5.

| Такт \ Вход | $i-1$ | $i$ |
|-------------|-------|-----|
| D1          | 1     | 0   |
| D2          | 0     | 1   |
| D3          | 0     | 0   |

Вариант 6.

| Такт \ Вход | $i-1$ | $i$ |
|-------------|-------|-----|
| D1          | 0     | 1   |
| D2          | 1     | 0   |
| D3          | 1     | 0   |

Вариант 7.

| Такт \ Вход | $i$ |
|-------------|-----|
| $D1$        | 1   |
| $D2$        | 1   |
| $D3$        | 1   |

Вариант 8.

| Такт \ Вход | $i$ |
|-------------|-----|
| $D1$        | 0   |
| $D2$        | 1   |
| $D3$        | 0   |

Вариант 9.

| Такт \ Вход | $i$ |
|-------------|-----|
| $D1$        | 1   |
| $D2$        | 1   |
| $D3$        | 0   |

Вариант 10.

| Такт \ Вход | $i-2$ | $i-1$ | $i$ |
|-------------|-------|-------|-----|
| $D1$        | X     | X     | 1   |
| $D2$        | X     | 1     | X   |
| $D3$        | 1     | X     | X   |

Вариант 11.

| Такт \ Вход | $i-2$ | $i-1$ | $i$ |
|-------------|-------|-------|-----|
| $D1$        | 0     | 1     | 0   |
| $D2$        | 0     | 1     | 0   |
| $D3$        | 0     | 1     | 0   |

Вариант 12.

| Такт \ Вход | $i-2$ | $i-1$ | $i$ |
|-------------|-------|-------|-----|
| $D1$        | 0     | X     | X   |
| $D2$        | X     | 0     | X   |
| $D3$        | X     | X     | 0   |

В вариантах 13–15 требуется подсчитать количество выполненных заданных отношений между селектируемыми данными двух типов. В скобках записан двоичный код: старший бит, указан слева, младший бит – справа.

Вариант 13.  $(D1_{(i-1)}, D2_i) > (D2_{(i-1)}, D3_i)$ . (Селекция с такта 2).

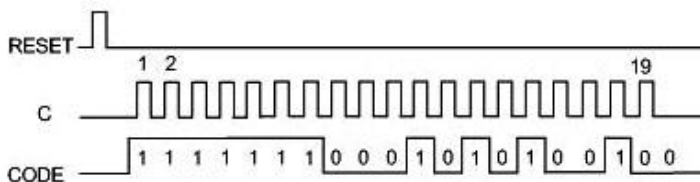
Вариант 14.  $(D3_i, D2_i, D1_i) \geq (D3_{(i-1)}, D2_{(i-1)}, D1_{(i-1)})$ . (Селекция с такта 2).

Вариант 15.  $(D3_{(i-2)}, D2_{(i-2)}, D1_{(i-1)}) \leq (D3_{(i-1)}, D2_{(i-1)}, D1_i)$ . (Селекция с такта 3).

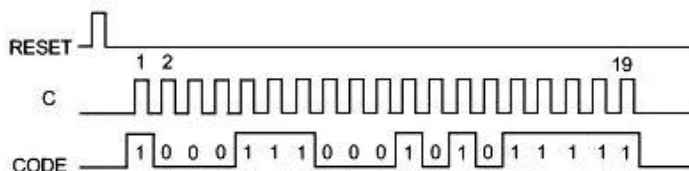
**Задача 4.** Разработать схему для подсчёта общего количества групп изолированных нулей в  $N$ -разрядном коде. Группа из любого количества нулей изолирована, если она справа и слева ограничена единицами.

Код поступает на вход CODE схемы последовательно в сопровождении  $N$  синхроимпульсов  $C$ . Значения разрядов кода на входе CODE изменяются в паузе между импульсами  $C$ . Перед поступлением кода на устройство подается сигнал начальной установки **RESET**. Ниже приведены варианты последовательностей с четырьмя группами изолированных нулей для  $N=19$  и  $N=18$ .

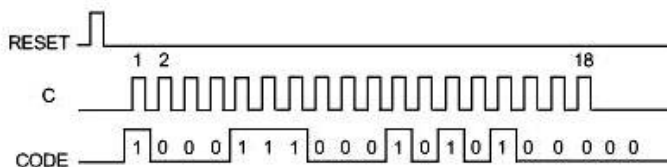
Вариант 1.



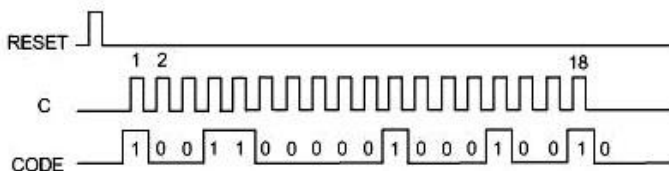
Вариант 2.



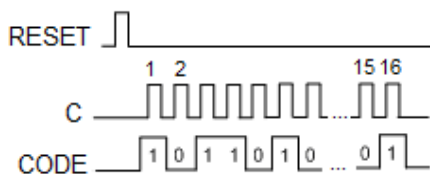
Вариант 3.



Вариант 4.

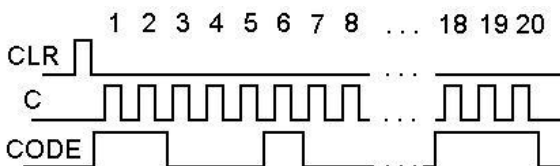


**Задача 5.** Разработать схему для подсчёта общего количества изолированных нулей и единиц в 16-разрядном двоичном коде. Нуль (единица) считается изолированным (ой), если он (она) с обеих сторон имеет единицы (нули). Крайние разряды считаются неизолированными. Например, для кода 1011010...01 результат равен 5 (три изолированных «0» и две изолированных «1»).



Код поступает на вход CODE схемы последовательно в сопровождении синхроимпульсов *C*. Значения разрядов кода на входе CODE изменяются в паузе между импульсами *C*. Перед поступлением кода и синхроимпульсов на схему подается одиночный импульс *RESET* для установки схемы в начальное состояние. Результат должен быть зафиксирован.

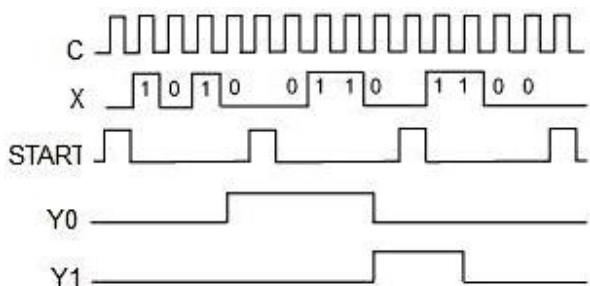
**Задача 6.** Разработать схему для подсчёта общего количества групп нулей и единиц в 20-разрядном коде. Группой нулей (единиц) считается не менее двух рядом стоящих нулей (единиц), отделенных единицами (нулями) или левой границей кода (со стороны старших разрядов). Например, код 1100010...0111 содержит три группы.



Код поступает на вход CODE схемы последовательно со старших разрядов в сопровождении двадцати синхроимпульсов *C*. Значения разрядов кода на входе CODE изменяются в паузе между импульсами *C*. Перед поступлением кода и импульсов *C* на схему подается одиночный импульс *CLR* для установки схемы в начальное состояние. Результат должен быть зафиксирован.

**Задача 7.** Разработать схему, которая вырабатывает на выходе *Y* сигнал единичного уровня, если сигнал на входе *X* имеет нулевое значение для шести и более тактовых интервалов.

**Задача 8.** Четырёхразрядные послылки двоичных данных в последовательной форме, синхронизируемые тактовыми импульсами *C*, поступают на вход *X* логической схемы. Перед началом каждой послылки на вход *START* схемы синхронно с импульсами *C* подаётся стартовый сигнал. Разработать синхронную последовательностную схему, которая при обнаружении среди входных послылок любой из трёх комбинаций, заданных вариантом, вырабатывает на выходах *Y1* и *Y0* код номера этой комбинации. При обнаружении любой другой комбинации на выходах выставляется код 00. Рисунок иллюстрирует работу схемы для варианта 1.



| Вариант | Кодовые комбинации |        |        |
|---------|--------------------|--------|--------|
|         | Первая             | Вторая | Третья |
| 1       | 1010               | 0110   | 0010   |
| 2       | 0001               | 1011   | 0101   |
| 3       | 0010               | 1101   | 0110   |
| 4       | 0111               | 1001   | 1100   |
| 5       | 0011               | 0110   | 1000   |
| 6       | 0100               | 1010   | 1100   |
| 7       | 1110               | 0010   | 0101   |
| 8       | 1011               | 0111   | 1111   |
| 9       | 1001               | 0101   | 1110   |
| 10      | 0101               | 1011   | 1000   |
| 11      | 1100               | 1010   | 0001   |
| 12      | 0110               | 0011   | 1100   |
| 13      | 0001               | 0011   | 0111   |
| 14      | 0101               | 0011   | 1111   |
| 15      | 0010               | 1110   | 0100   |

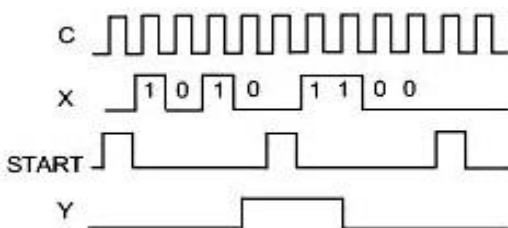
**Задача 9.** Разработать схему, которая имеет четыре входа  $START$ ,  $C$ ,  $D$ ,  $X$  и три выхода  $Y_2, Y_1, Y_0$ . В каждом цикле работы схемы на входы  $START$ ,  $C$  и  $D$  подаются сигналы в соответствии с временной диаграммой:



После подачи сигнала  $START$  в одном из последующих восьми тактов (1-8) по фронту 0/1 импульса  $C$  на вход  $X$  приходит потенциальный сигнал, который разрешает подачу на схему импульсов  $C$  и сохраняется до начала следующего цикла. После окончания цикла работы схемы на выходах  $Y_2, Y_1, Y_0$  должен быть сформирован номер такта, в котором пришёл сигнал на вход  $X$ .

| Вариант | Битовая последовательность на входе $D$ в тактах |   |   |   |   |   |   |   |   |   |    |    |
|---------|--------------------------------------------------|---|---|---|---|---|---|---|---|---|----|----|
|         | 0                                                | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1       | 0                                                | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1  | 0  |
| 2       | 0                                                | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1  | 0  |
| 3       | 0                                                | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0  | 0  |
| 4       | 0                                                | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1  | 1  |
| 5       | 0                                                | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1  | 1  |
| 6       | 0                                                | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1  | 0  |
| 7       | 0                                                | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1  | 1  |
| 8       | 0                                                | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1  | 0  |
| 9       | 0                                                | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0  | 0  |
| 10      | 0                                                | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1  | 1  |
| 11      | 0                                                | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0  | 0  |
| 12      | 0                                                | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1  | 1  |
| 13      | 0                                                | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1  | 0  |
| 14      | 0                                                | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1  | 0  |
| 15      | 0                                                | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1  | 1  |

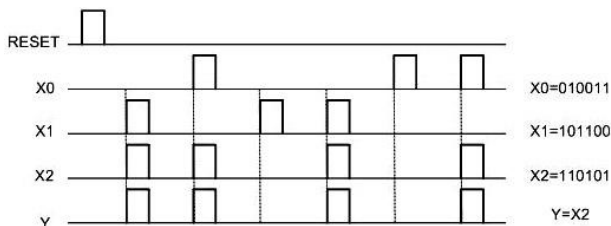
**Задача 10.** Четырёхразрядные послылки двоичных данных в последовательной форме, синхронизируемые тактовыми импульсами  $C$ , поступают на вход  $X$  логической схемы. Перед началом каждой послылки на вход  $START$  схемы синхронно с импульсами  $C$  подаётся стартовый сигнал. Разработать схему, которая вырабатывает выходной сигнал  $Y$  при обнаружении среди входных послылок любой из трёх комбинаций, заданных вариантом. Если выход  $Y$  был предварительно установлен в «1», то он должен сбрасываться в «0», если текущая послылка не относится к трём заданным комбинациям. Рисунок иллюстрирует работу схемы для варианта 1.



| Вариант | Кодовые комбинации |
|---------|--------------------|
| 1       | 1010, 0110, 0010   |
| 2       | 0001, 1011, 0101   |
| 3       | 0010, 1101, 0110   |
| 4       | 0111, 1001, 1100   |
| 5       | 0011, 0110, 1000   |
| 6       | 0100, 1010, 1100   |
| 7       | 1110, 0010, 0101   |
| 8       | 1011, 0111, 1111   |
| 9       | 1001, 0101, 1110   |
| 10      | 0010, 0101, 1100   |
| 11      | 0101, 1101, 0011   |
| 12      | 0111, 1001, 0100   |
| 13      | 1000, 0110, 1110   |
| 14      | 1101, 0101, 1001   |
| 15      | 0001, 0111, 1011   |

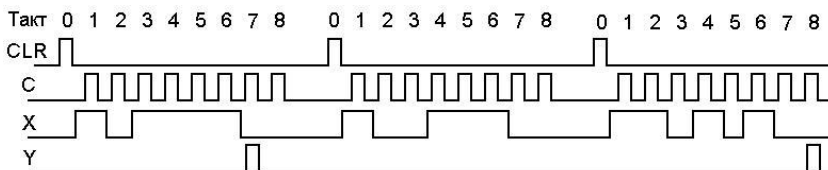
**Задача 11.** Разработать устройство, которое имеет четыре входа и один выход. После подачи сигнала начальной установки на вход  $RESET$  на входы  $X_2$ ,  $X_1$  и  $X_0$  начинают синхронно и последова-

тельно, начиная со старших разрядов, подаваться коды 6-разрядных беззнаковых чисел. Устройство должно передавать на выход  $Y$  код максимального из трёх чисел, подаваемых на его входы (пример показан на рисунке).

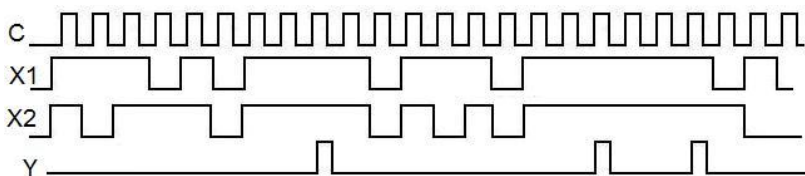


**Задача 12.** Разработать схему, которая вырабатывает на выходе  $Y$  сигнал единичного уровня в течение шести тактовых импульсов после того, как сигнал на входе  $X$  имеет единичное значение в течение одного тактового интервала времени. Если сигнал на входе  $X$  принимает единичное значение снова в течение шести тактовых интервалов, то схема не реагирует на это изменение. Сигнал  $X$  изменяет свое значение по фронту 1/0 сигнала синхронизации  $C$ .

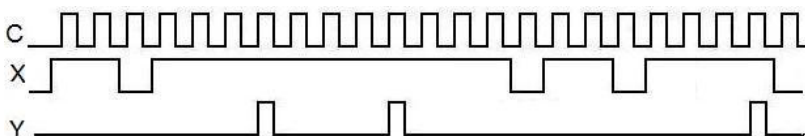
**Задача 13.** Разработать схему анализа 6-разрядной кодовой последовательности, которая поступает на вход  $X$  схемы одновременно с синхроимпульсами  $C$ . Кодовые сигналы изменяются в промежутках между импульсами  $C$ . Схема должна выдать на выход  $Y$  импульс в 7 такте, если число единиц в 3, 5 и 6 тактах кодовой последовательности больше числа единиц в 1, 2 и 4 тактах. Если меньше, то схема должна выдать импульс в 8 такте. Если же число единиц в тактах 3, 5 и 6 равно числу единиц в тактах 1, 2 и 4, то импульс на выходе схемы не вырабатывается. Перед поступлением кодовой последовательности на схему подается сигнал  $CLR$  установки в начальное состояние.



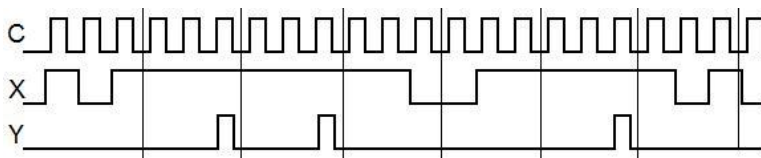
**Задача 14.** Разработать схему, которая выдает на выход  $Y$  импульс всякий раз, когда в двух кодовых последовательностях  $X1$  и  $X2$ , поступающих на отдельные входы схемы, одновременно появляется подряд (без пропуска) три единицы. Кодовые сигналы в последовательностях  $X1$  и  $X2$  изменяются в промежутках между синхроимпульсами  $C$ .



**Задача 15.** Разработать схему, которая выдает на выход  $Y$  импульс всякий раз, когда в поступающей на ее вход  $X$  кодовой последовательности появляется подряд (без пропуска) четыре единицы. Кодовые сигналы изменяются в промежутках между синхроимпульсами  $C$ .



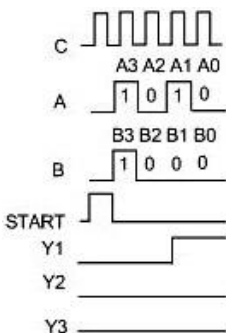
**Задача 16.** На один вход схемы поступает кодовая последовательность  $X$ , на другой – сопровождающая её последовательность синхроимпульсов  $C$ . Кодовые сигналы изменяются в промежутках между импульсами  $C$ . Схема должна выдавать на выход  $Y$  каждый третий импульс  $C$ , если сигналы кодовой последовательности в каждом из трёх тактов равны «1».



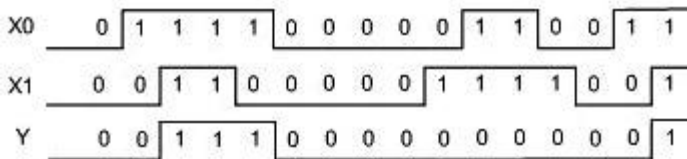
**Задача 17.** На вход схемы сравнения последовательного действия синхронно с тактовыми импульсами  $C$  поступают четырехразрядные двоичные числа:  $A_3, A_2, A_1, A_0$  и  $B_3, B_2, B_1, B_0$ , начиная со старших разрядов. Подаче чисел на входы  $A$  и  $B$  схемы предшествует подача стартового сигнала на вход  $START$ .

Разработать синхронную последовательностную схему, на выходах  $Y_1$ ,  $Y_2$  и  $Y_3$  которой формируются сигналы по следующему правилу:

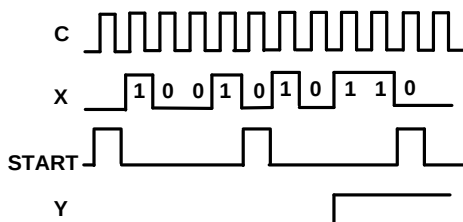
$Y_1=1$  при  $A>B$ ,  $Y_2=1$  при  $A=B$ ,  $Y_3=1$  при  $A<B$ .



**Задача 18.** Разработать асинхронную схему обнаружения последовательности, которая имеет два входа  $X_1$  и  $X_0$  и один выход  $Y$ . Выходной сигнал  $Y$  становится равным «1» только при последовательности входных состояний  $X_1, X_0$ , принимающих значения 00, 01, 11. Выходной сигнал  $Y$  становится равным «0» только после появления на входах последовательности состояний 11, 01, 00.



**Задача 19.** Двоично-десятичные данные в последовательной форме, синхронизируемые импульсами  $C$ , начиная со старших разрядов, поступают на вход  $X$  логической схемы. Перед началом каждой кодовой посылки на вход  $START$  схемы синхронно с импульсами  $C$  подается стартовый сигнал. Разработать схему, которая вырабатывает выходной сигнал  $Y$ , как только обнаруживается подача на её вход неиспользуемой в данном коде кодовой комбинации. На рисунке представлен пример для варианта 1.



| Вариант | Двоично-десятичный код |
|---------|------------------------|
| 1       | 8421                   |
| 2       | 8421+3                 |
| 3       | 5421                   |
| 4       | 5421+3                 |
| 5       | 2421                   |
| 6       | 7421                   |
| 7       | 3321                   |
| 8       | 5211                   |
| 9       | 2521                   |
| 10      | 3411                   |
| 11      | 2621                   |
| 12      | 4271                   |
| 13      | 4621                   |
| 14      | 5221                   |
| 15      | 6321                   |

**Задача 20.** Двоичные данные в последовательной форме, синхронизируемые тактовыми импульсами  $C$ , поступают по линии  $X$  на логическую последовательностную схему. Данные изменяются по фронту 1/0 сигнала  $C$ .

Разработать схему, в которой на выходе  $Y$  сигнал появляется каждый раз, как только во входной последовательности  $X$  появляется набор 1101.

## Приложение

### Справочные данные по элементам

Данное приложение содержит краткие сведения по тем функциональным группам библиотечных элементов, которые используются при выполнении технического задания на проектирование процессора.

Библиотечные элементы схемного редактора САПР Xilinx Foundation разбиты на следующие основные группы:

- Gate — логические элементы;
- Flip-Flop/Latch — триггеры/защелки;
- MUX/DMUX — мультиплексоры;
- Coder/Decoder — дешифраторы;
- Arithmetic Circuit — арифметические устройства;
- Counter — счетчики;
- Register — регистры;
- Memory — память.

Элементы, включенные в библиотеку, являются либо базовыми элементами, либо макроэлементами. Базовый элемент (Primitive) — такой элемент, который не может быть разбит на меньшие компоненты. Макроэлемент (Macro) — элемент, имеющий внутреннюю структуру, состоящую из базовых элементов. Внутреннюю структуру макроэлемента всегда можно увидеть в схемном редакторе, используя команду Hierarchy Push/Pop.

Как правило, в состав каждой группы входят элементы одного типа, различающиеся разрядностью или количеством входов. Разрядность указывается цифрой в обозначении компонента.

### Логические элементы

Группа Gate содержит следующий набор логических элементов:

- инвертор НЕ (INV);
- элементы И (конъюнктор, AND) с инверсными и не инверсными входами от 2 до 9;
- элементы ИЛИ (дизъюнктор, OR) с инверсными и не инверсными входами от 2 до 9;

- элементы И-НЕ (NAND) с инверсными и не инверсными входами от 2 до 9;
- элементы ИЛИ-НЕ (NOR) с инверсными и не инверсными входами от 2 до 9;
- элементы «Исключающее ИЛИ» (сложение по модулю 2, XOR) с числом входов от 2 до 9;
- элементы, выполняющие функцию логическая равнозначность (XNOR) с числом входов от 2 до 9;
- элементы 2-2И-2ИЛИ (SOP4) с инверсными и не инверсными входами;
- элементы 1-2И-2ИЛИ (SOP3) с инверсными и не инверсными входами.

Расшифровка обозначений элементов этой группы достаточно проста, поскольку тип элемента явно указан в его названии, а число обозначает количество входов. Кроме того, у элементов некоторые входы могут быть проинвертированы, что отмечается символом *B* в обозначении элемента. Пример соглашения, принятого для обозначения логических элементов, приведён на рис. П.1.

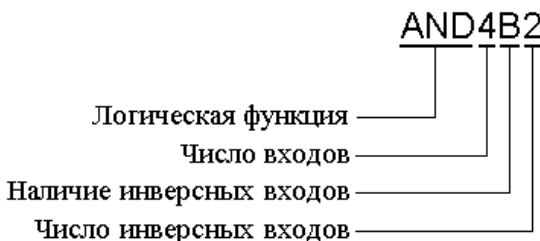


Рис. П.1. Обозначение логических элементов

### ***Инвертор НЕ (INV)***

Условное графическое обозначение инвертора, приведено на рис. П.2. Инвертор является базовым элементом.

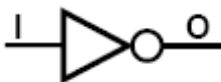


Рис. П.2. Условное графическое обозначение инвертора

## Элементы И (AND)

Условные графические обозначения элементов AND приведены на рис. П.3. Элементы AND до пяти входов включительно являются базовыми элементами и могут иметь инверсные входы в любой комбинации.

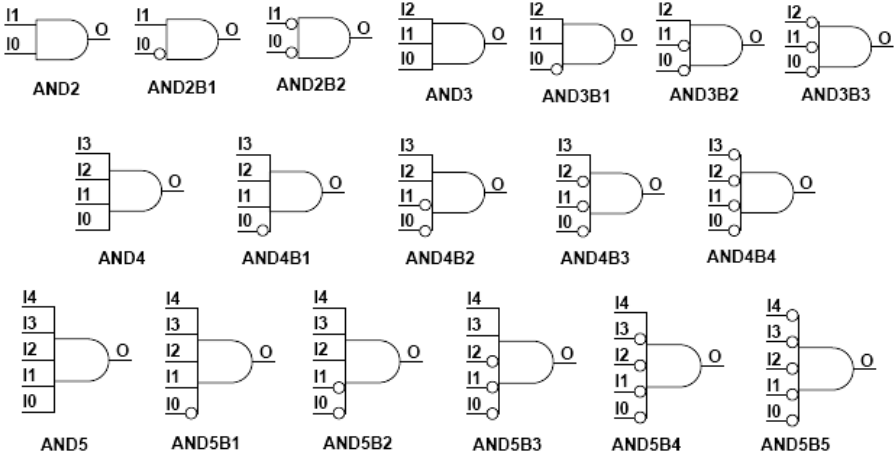


Рис. П.3. Условное графическое обозначение элементов AND

Элементы AND с числом входов от 6 до 9 не имеют инверсных входов и являются макроэлементами.

## Элементы ИЛИ (OR)

Условные графические обозначения элементов OR приведены на рис. П.4. Элементы OR до пяти входов включительно являются базовыми элементами и могут иметь инверсные входы в любой комбинации.

Элементы OR с числом входов от 6 до 9 не имеют инверсных входов и являются макроэлементами.

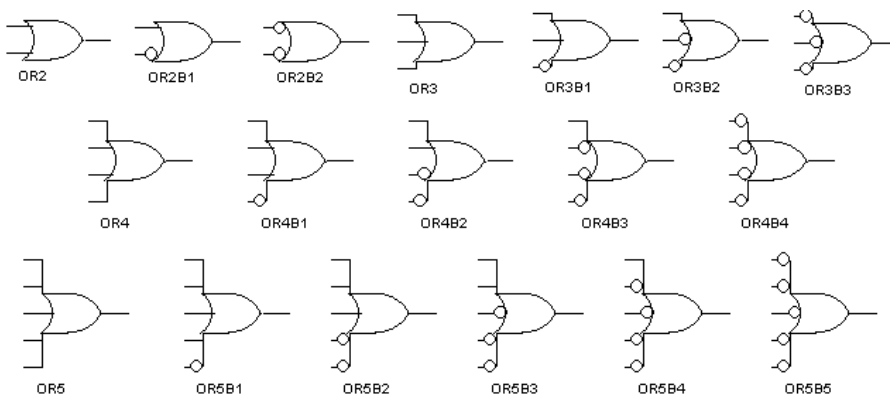


Рис. П.4. Условное графическое обозначение элементов OR

### ***Элементы ИНЕ (NAND)***

Условные графические обозначения элементов NAND приведены на рис. П.5. Элементы NAND до пяти входов включительно являются базовыми элементами и могут иметь инверсные входы в любой комбинации.

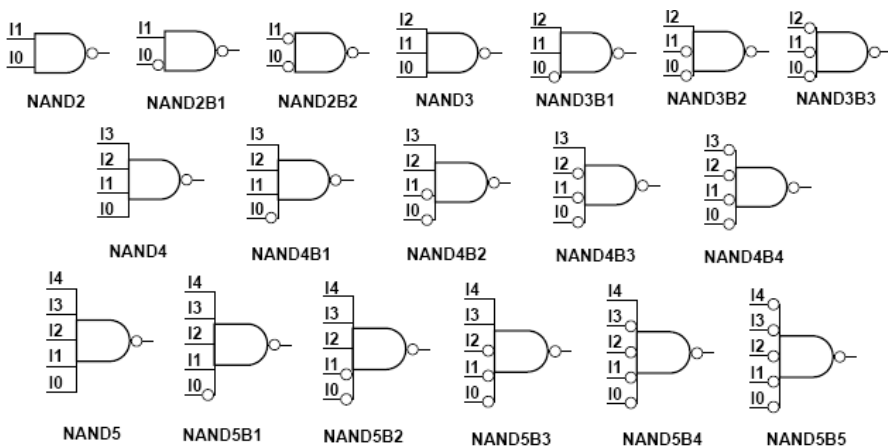


Рис. П.5. Условное графическое обозначение элементов NAND

### ***Элементы ИЛИНЕ (NOR)***

Условные графические обозначения элементов NOR приведены на рис. П.6. Элементы NOR до пяти входов включительно являются базовыми элементами и могут иметь инверсные входы в любой комбинации.

ся базовыми элементами и могут иметь инверсные входы в любой комбинации. Элементы NOR с числом входов от 6 до 9 не имеют инверсных входов и являются макроэлементами.

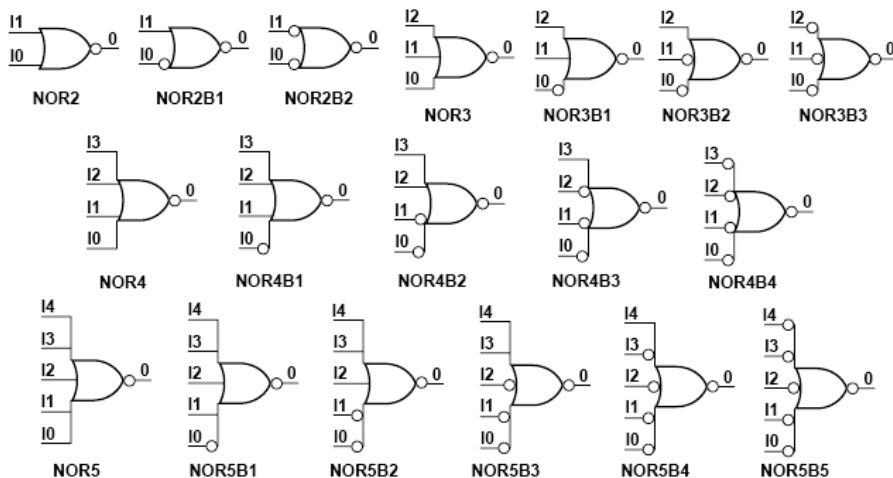


Рис. П.6. Условное графическое обозначение элементов NOR

### *Элементы 2-2И-2ИЛИ (SOP4)*

Условные графические обозначения элементов SOP4 приведены на рис. П.7. Элементы SOP4 являются макроэлементами и могут иметь инверсные входы в любой комбинации.

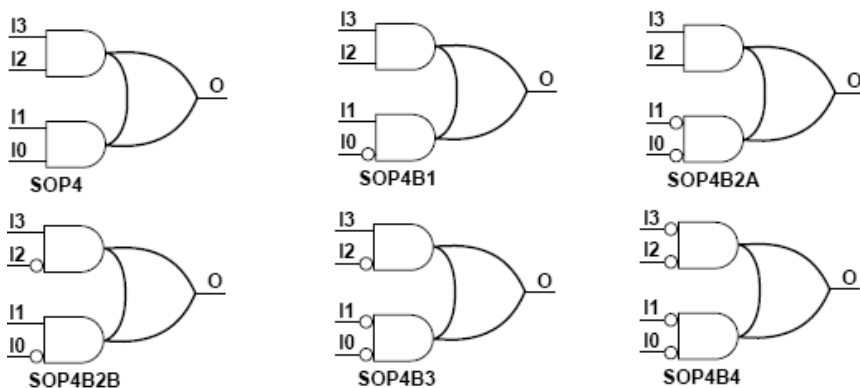


Рис. П.7. Условное графическое обозначение элементов SOP4

### ***Элементы 1-2И-2ИЛИ (SOP3)***

Условные графические обозначения элементов SOP3 приведены на рис. П.8. Элементы SOP3 являются макроэлементами и могут иметь инверсные входы в любой комбинации.

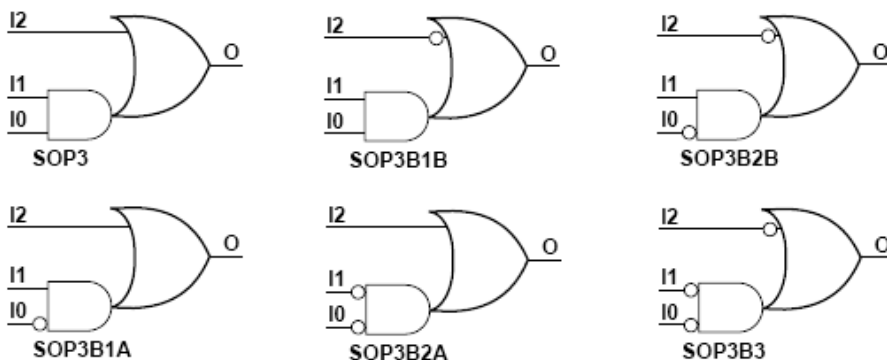


Рис. П.8. Условное графическое обозначение элементов SOP3

### ***Элементы ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR) и ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ (XNOR)***

В библиотеке элементы «Исключающее ИЛИ» (сложение по модулю 2, XOR) и элементы «Исключающее ИЛИ-НЕ» (логическая равнозначность, XNOR) имеют число входов от 2 до 9.

Условное графическое обозначение элементов XOR2, XNOR2 с двумя входами приведено на рис. П.9, а их таблица истинности представлена табл. П.1.



Рис. П.9. Условное графическое обозначение элементов XOR и XNOR

Таблица истинности элементов XOR2, XNOR2

| Входы |    | Выход элемента XOR | Выход элемента XNOR |
|-------|----|--------------------|---------------------|
| I0    | I1 | Q                  | Q                   |
| 0     | 0  | 0                  | 1                   |
| 0     | 1  | 1                  | 0                   |
| 1     | 0  | 1                  | 0                   |
| 1     | 1  | 0                  | 1                   |

## Триггеры

### *Основные сведения*

В библиотеке элементов имеются три типа синхронных триггеров с динамическим управлением записью:

D-триггеры;

JK-триггеры;

T-триггеры.

JK- и T-триггеры имеют прямой динамический синхронизирующий вход *C*. D-триггеры могут иметь как прямой, так и инверсный синхронизирующий динамический вход *C*. Кроме того, указанные триггеры могут иметь вход разрешения для синхроимпульсов *CE* (clockenable).

Каждый из перечисленных типов триггеров может иметь асинхронный вход предварительной установки триггера или в «0» или «1».

Также триггеры могут иметь синхронный вход предварительной установки триггера в «0» и/или «1».

В системе Xilinx Foundation приняты следующие сокращения установочных входов на условном графическом обозначении триггера:

*CLR* (Clear), *PRE* (Preset) — асинхронные входы установки триггера в «0» и «1» соответственно;

$R$  (Reset),  $S$  (Set) — *синхронные* входы установки триггера в «0» и «1» соответственно.

### **Обозначение триггеров**

Наличие тех или входов у конкретного библиотечного триггера можно выяснить из его текстового обозначения. Пример соглашения для обозначения триггеров в библиотеке элементов приведён на рис. П.10.

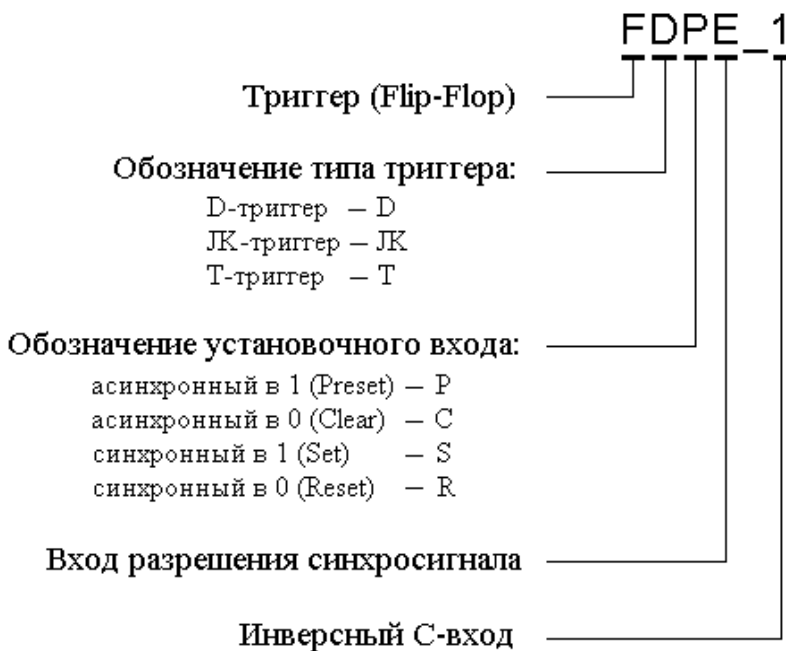


Рис. П.10. Обозначение триггеров

### **Библиотечный набор триггеров**

#### ***D-триггеры***

FDCE — D-триггер с входом разрешения синхросигнала и асинхронной установкой в «0».

FDPE — D-триггер с входом разрешения синхросигнала и асинхронной установкой в «1».

FDCE\_1 — D-триггер с инверсным синхровходом, входом разрешения синхросигнала и асинхронной установкой в «0».

FDPE\_1 — D-триггер с инверсным синхровходом, входом разрешения синхросигнала и асинхронной установкой в «1».

FD — D-триггер.

FD\_1 — D-триггер с инверсным синхровходом.

FDC — D-триггер с асинхронной установкой в «0».

FDC\_1 — D-триггер с инверсным синхровходом и асинхронной установкой в «0».

FDP — D-триггер с асинхронной установкой в «1».

FDP\_1 — D-триггер с инверсным синхровходом и асинхронной установкой в «1».

FDR — D-триггер с синхронной установкой в «0».

FDS — D-триггер с синхронной установкой в «1».

FDRS — D-триггер с синхронной установкой в «0» и «1».

FDSR — D-триггер с синхронной установкой в «1» и «0».

FDRE — D-триггер с входом разрешения синхросигнала и синхронной установкой в «0».

FDSE — D-триггер с входом разрешения синхросигнала и синхронной установкой в «1».

FDRSE — D-триггер с синхронной установкой в «0» и «1» и с входом разрешения синхросигнала.

FDSRE — D-триггер с синхронной установкой в «1» и «0» и с входом разрешения синхросигнала.

### ***JK-триггеры***

FJKC — JK-триггер с асинхронной установкой в «0».

FJKCE — JK-триггер с входом разрешения синхросигнала и асинхронной установкой в «0».

FJKP — JK-триггер с асинхронной установкой в «1».

FJKPE — JK-триггер с входом разрешения синхросигнала и асинхронной установкой в «1».

FJKRSE — JK-триггер с синхронной установкой в «0» и «1» и с входом разрешения синхросигнала.

FJKSRE — JK-триггер с синхронной установкой в «1» и «0» и с входом разрешения синхросигнала.

### ***T-триггеры***

FTC — Т-триггер с асинхронной установкой в «0».

FTCE — Т-триггер с входом разрешения синхросигнала и с асинхронной установкой в «0».

FTCLE — Т-триггер с функцией синхронной загрузки, с входом разрешения синхросигнала и с асинхронной установкой в «0».

FTCLEX — Т-триггер с функцией синхронной загрузки, с входом разрешения синхросигнала и с асинхронной установкой в «0».

FTP — Т-триггер с асинхронной установкой в «1».

FTPE — Т-триггер с входом разрешения синхросигнала и с асинхронной установкой в «1».

FTPLE — Т-триггер с функцией синхронной загрузки, с входом разрешения синхросигнала и с асинхронной установкой в «1».

FTRSE — Т-триггер с входом разрешения синхросигнала и с синхронной установкой в «0» и «1».

FTRSLE — Т-триггер с функцией синхронной загрузки, с входом разрешения синхросигнала и с синхронной установкой в «0» и «1».

FTSRE — Т-триггер с входом разрешения синхросигнала и с синхронной установкой в «1» и «0».

FTSRLE — Т-триггер с функцией синхронной загрузки, с входом разрешения синхросигнала и с синхронной установкой в «1» и «0».

### ***Краткое описание отдельных триггеров***

#### ***D-триггеры***

Практически все триггеры являются макроэлементами, и только два триггера являются базовыми. Это D-триггеры FDCE и FDPE, на их основе и создано все многообразие перечисленных выше триггеров.

FDCE — D-триггер с входом разрешения синхросигнала  
и асинхронной установкой в «0»

Условное графическое обозначения D-триггера FDCE приведено на рис. П.11, а его таблица переходов представлена табл. П.2.

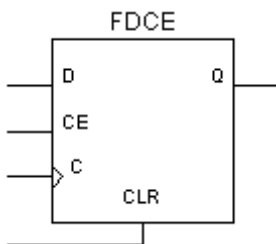


Рис. П.11. Условное графическое обозначение D-триггера FDCE

Таблица П.2

Таблица переходов D-триггера FDCE

| Входы |    |   |     | Выход |
|-------|----|---|-----|-------|
| CLR   | CE | D | C   | Q     |
| 1     | X  | X | X   | 0     |
| 0     | 0  | X | X   | Q     |
| 0     | 1  | 1 | 0/1 | 1     |
| 0     | 1  | 0 | 0/1 | 0     |

FDPE — D-триггер с входом разрешения синхросигнала  
и асинхронной установкой в «1»

Условное графическое обозначения D-триггера FDPE приведено на рис. П.12, а его таблица переходов представлена табл. П.3.

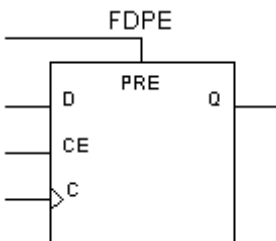


Рис. П.12. Условное графическое обозначение D-триггера FDPE

Таблица П.3

Таблица переходов D-триггера FDPE

| Входы      |           |          |          | Выход    |
|------------|-----------|----------|----------|----------|
| <i>PRE</i> | <i>CE</i> | <i>D</i> | <i>C</i> | <i>Q</i> |
| 1          | X         | X        | X        | 1        |
| 0          | 0         | X        | X        | <i>Q</i> |
| 0          | 1         | 1        | 0/1      | 1        |
| 0          | 1         | 0        | 0/1      | 0        |

FDRE — D-триггер с входом разрешения синхросигнала и синхронной установкой в «0»

Условное графическое обозначения D-триггера FDRE приведено на рис. П.13, а его таблица переходов представлена табл. П.4.

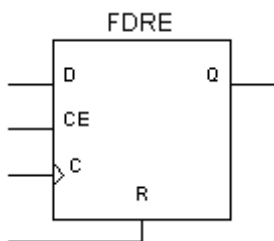


Рис. П.13. Условное графическое обозначение D-триггера FDRE

Таблица П.4

Таблица переходов D-триггера FDRE

| Входы    |           |          |          | Выход    |
|----------|-----------|----------|----------|----------|
| <i>R</i> | <i>CE</i> | <i>D</i> | <i>C</i> | <i>Q</i> |
| 1        | X         | X        | 0/1      | 0        |
| 0        | 0         | X        | X        | <i>Q</i> |
| 0        | 1         | 1        | 0/1      | 1        |
| 0        | 1         | 0        | 0/1      | 0        |

Данный триггер является макроэлементом, его структура приведена на рис. П.14.

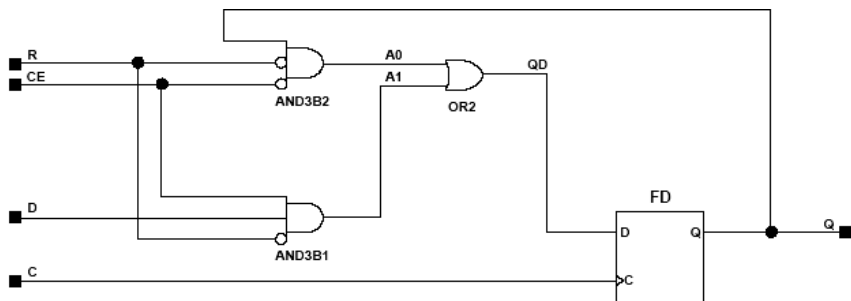


Рис. П.14. Внутренняя структура D-триггера FDRE

*FDSE — D-триггер с входом разрешения синхросигнала и синхронной установкой в «1»*

Условное графическое обозначения D-триггера FDSE приведено на рис. П.15. а его таблица переходов представлена табл. П.5.

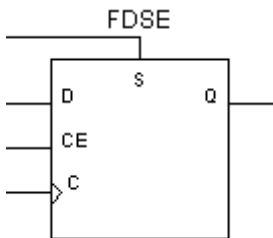


Рис. П.15. Условное графическое обозначение D-триггера FDSE

Таблица П.5

Таблица переходов D-триггера FDSE

| Входы |    |   |     | Выход |
|-------|----|---|-----|-------|
| S     | CE | D | C   | Q     |
| 1     | X  | X | 0/1 | 1     |
| 0     | 0  | X | X   | Q     |
| 0     | 1  | 1 | 0/1 | 1     |
| 0     | 1  | 0 | 0/1 | 0     |

Триггер FDSE является макроэлементом, его структура приведена на рис. П.16.

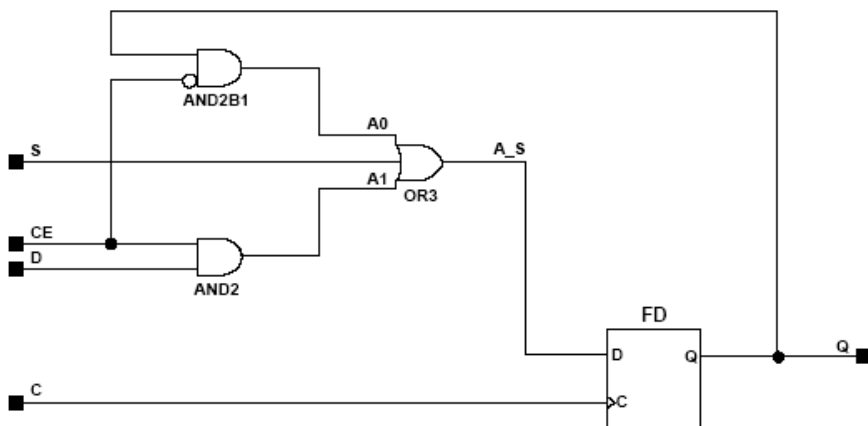


Рис. П.16. Внутренняя структура D-триггера FDSE

### ***JK-триггеры***

FJKC — JK-триггер с асинхронной установкой в «0»

Условное графическое обозначения JK-триггера FJKC приведено на рис. П.17, а его таблица переходов представлена табл. П.6.

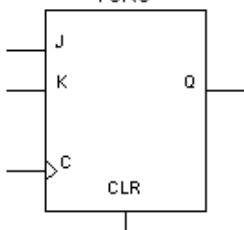


Рис. П.17. Условное графическое обозначение JK-триггера FJKC

Таблица П.6

Таблица переходов JK-триггера FJKC

| Входы      |          |          |          | Выход    |
|------------|----------|----------|----------|----------|
| <i>CLR</i> | <i>J</i> | <i>K</i> | <i>C</i> | <i>Q</i> |
| 1          | X        | X        | X        | 0        |
| 0          | 0        | 0        | 0/1      | <i>Q</i> |
| 0          | 0        | 1        | 0/1      | 0        |
| 0          | 1        | 0        | 0/1      | 1        |
| 0          | 1        | 1        | 0/1      | $\neg Q$ |

Триггер FJKС является макроэлементом, его структура приведена на рис. П.18.

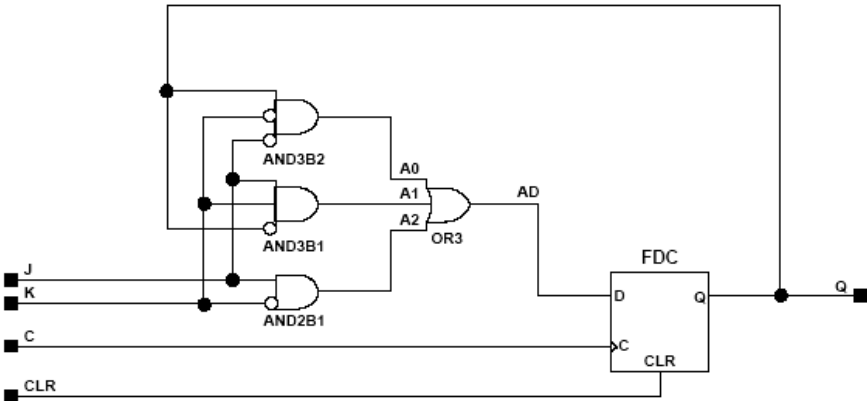


Рис. П.18. Внутренняя структура JK-триггера FJKС

FJKCE — JK-триггер с входом разрешения синхросигнала и асинхронной установкой в «0»

Условное графическое обозначения JK-триггера FJKCE приведено на рис. П.19, а его таблица переходов представлена табл. П.7.

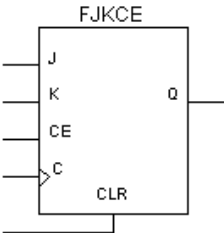


Рис. П.19. Условное графическое обозначение JK-триггера FJKCE

Таблица П.7

Таблица переходов JK-триггера FJKCE

| Входы |    |   |   |     | Выход    |
|-------|----|---|---|-----|----------|
| CLR   | CE | J | K | C   | Q        |
| 1     | X  | X | X | X   | 0        |
| 0     | 0  | X | X | X   | Q        |
| 0     | 1  | 0 | 0 | X   | Q        |
| 0     | 1  | 0 | 1 | 0/1 | 0        |
| 0     | 1  | 1 | 0 | 0/1 | 1        |
| 0     | 1  | 1 | 1 | 0/1 | $\neg Q$ |

Триггер FJKCE является макроэлементом, его структура приведена на рис. П.20.

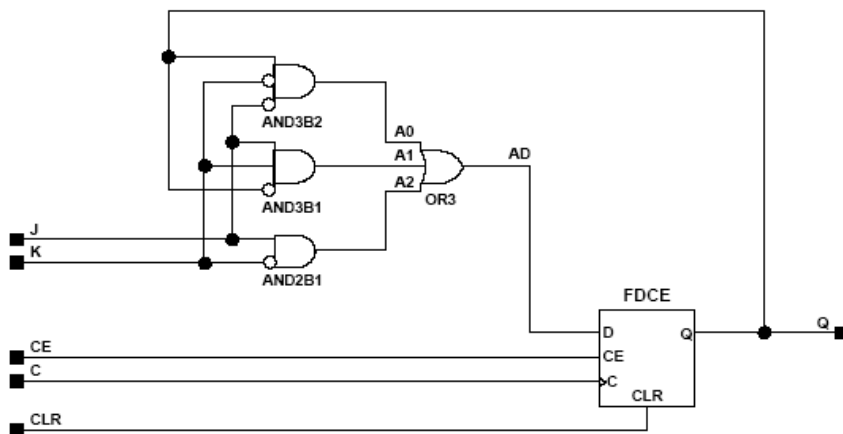


Рис. П.20. Внутренняя структура JK-триггера FJKCE

### ***T-триггеры***

FTCE — T-триггер с входом разрешения синхросигнала и асинхронной установкой в «0»

Условное графическое обозначения T-триггера FTCE приведено на рис. П.21, а его таблица переходов представлена табл. П.8.

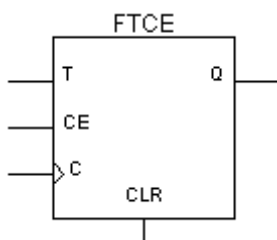


Рис. П.21. Условное графическое обозначение T-триггера FTCE

Таблица переходов Т-триггера FTCE

| Входы |      |     |     | Выход    |
|-------|------|-----|-----|----------|
| $CLR$ | $CE$ | $T$ | $C$ | $Q$      |
| 1     | X    | X   | X   | 0        |
| 0     | 0    | X   | X   | $Q$      |
| 0     | 1    | 0   | 0/1 | $Q$      |
| 0     | 1    | 1   | 0/1 | $\neg Q$ |

Триггер FTCE является макроэлементом, его структура дана на рис. П.22.

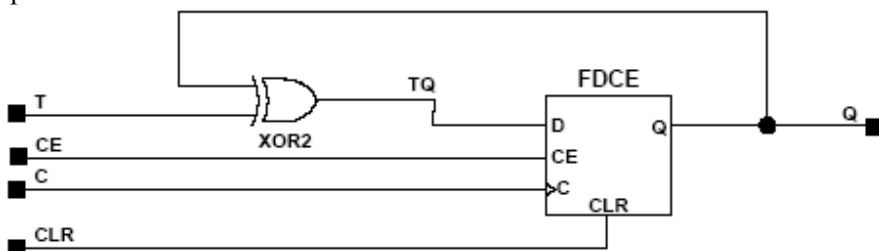


Рис. П.22. Внутренняя структура Т-триггера FTCE

FTCLE — Т-триггер с функцией синхронной загрузки,  
входом разрешения синхросигнала и асинхронной  
установкой в «0»

Условное графическое обозначения Т-триггера FTCLE приведено на рис. П.23, а его таблица переходов представлена табл. П.9.

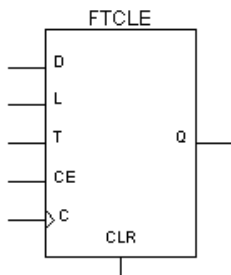


Рис. П.23. Условное графическое обозначение Т-триггера FTCLE

Таблица переходов Т-триггера FTCLE

| Входы |   |    |   |   |     | Выход    |
|-------|---|----|---|---|-----|----------|
| CLR   | L | CE | T | D | C   | Q        |
| 1     | X | X  | X | X | X   | 0        |
| 0     | 1 | X  | X | 1 | 0/1 | 1        |
| 0     | 1 | X  | X | 0 | 0/1 | 0        |
| 0     | 0 | 0  | X | X | X   | Q        |
| 0     | 0 | 1  | 0 | X | 0/1 | Q        |
| 0     | 0 | 1  | 1 | X | 0/1 | $\neg Q$ |

Триггер FTCLE является макроэлементом, его структура приведена на рис. П.24.

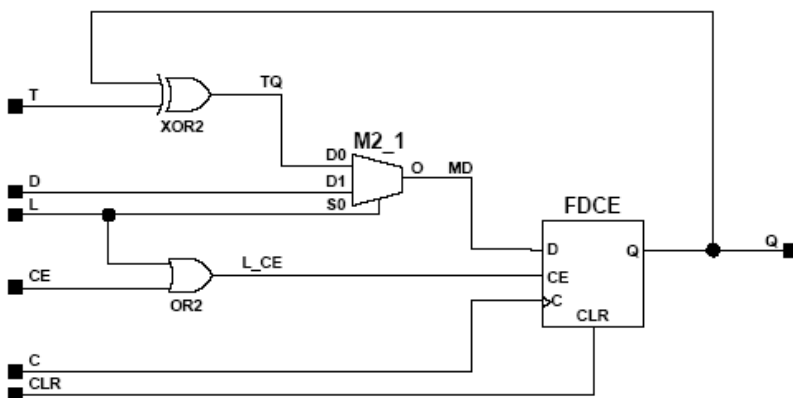


Рис. П.24. Внутренняя структура Т-триггера FTCLE

FTCLEX — Т-триггер с функцией синхронной загрузки,  
входом разрешения синхросигнала и асинхронной  
установкой в «0»

Условное графическое обозначения Т-триггера FTCLEX приведено на рис. П.25, а его таблица переходов представлена табл. П.10.

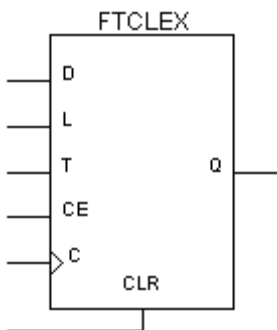


Рис. П.25. Условное графическое обозначение Т-триггера FTCLEX

Таблица П.10

Таблица переходов Т-триггера FTCLEX

| Входы      |          |           |          |          |          | Выход    |
|------------|----------|-----------|----------|----------|----------|----------|
| <i>CLR</i> | <i>L</i> | <i>CE</i> | <i>T</i> | <i>D</i> | <i>C</i> | <i>Q</i> |
| 1          | X        | X         | X        | X        | X        | 0        |
| 0          | 1        | 1         | X        | 1        | 0/1      | 1        |
| 0          | 1        | 1         | X        | 0        | 0/1      | 0        |
| 0          | 0        | 0         | X        | X        | X        | <i>Q</i> |
| 0          | 0        | 1         | 0        | X        | 0/1      | <i>Q</i> |
| 0          | 0        | 1         | 1        | X        | 0/1      | $\neg Q$ |

Триггер FTCLEX является макроэлементом, его структура дана на рис. П.26.

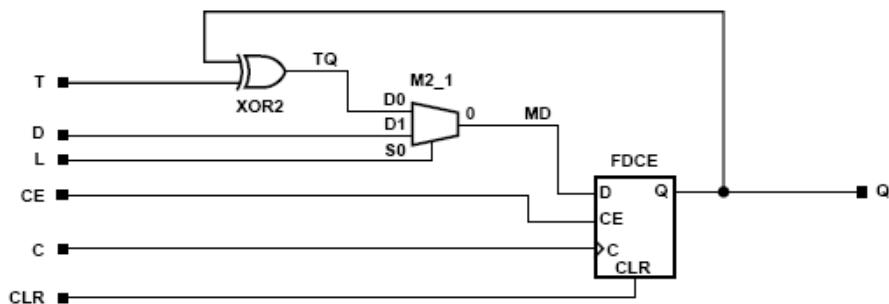


Рис. П.26. Внутренняя структура Т-триггера FTCLEX

Примечание.: Как видно из рис. П.23 и П.25, графическое обозначение и наименование выводов триггеров FTCLE и FTCLEX полностью совпадают. Однако

рассмотрение способов их реализации позволяет выявить различие при выполнении микрооперации загрузки (см. рис. П.24 и П.26).

Если активен сигнал на входе  $L$ , то по фронту тактового сигнала триггер принимает состояние, определяемое входом  $D$ . Однако для триггера FTCLE такая загрузка возможна при отсутствии активного сигнала на входе разрешения  $CE$  (такое разрешение формируется элементом OR2 при наличии активного уровня на входе  $L$ ), а для триггера FTCLEX загрузка возможна только при наличии активного уровня на входе  $CE$  (см. табл. П.9 и П.10).

## Мультиплексоры

Пример соглашения для обозначения мультиплексоров в библиотеке элементов приведен на рис. П.27.

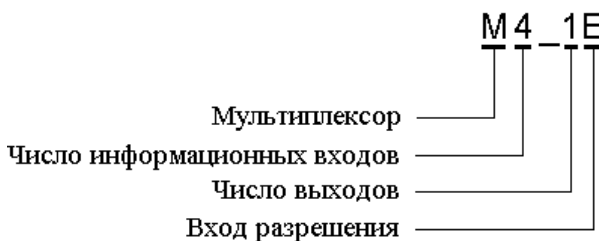


Рис. П.27.Обозначение мультиплексоров

### Мультиплексор 2-1 ( $M2\_1$ )

Мультиплексор выбирает один информационный вход из двух источников ( $D1$  или  $D0$ ) в зависимости от состояния управляющего входа  $S0$ . Выход ( $O$ ) принимает состояние выбранного входа данных. При  $S0 = 0$  на выходе появляется состояние входа  $D0$ , при  $S0 = 1 - D1$ .

Условное графическое обозначение мультиплексора приведено на рис. П.28, а его таблица истинности представлена табл. П.11.

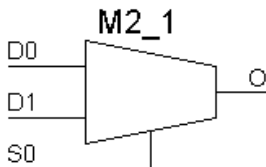


Рис. П.28. Условное графическое обозначение мультиплексора  $M2\_1$

Таблица истинности мультиплексора M2\_1

| Входы |    |    | Выход |
|-------|----|----|-------|
| S0    | D1 | D0 | O     |
| 1     | 1  | X  | 1     |
| 1     | 0  | X  | 0     |
| 0     | X  | 1  | 1     |
| 0     | X  | 0  | 0     |

### ***Мультиплексоры 2-1 с инверсными входами (M2\_1B1 и M2\_1B2)***

Условные графические обозначения мультиплексоров с инверсными входами приведены на рис. П.29. Данные мультиплексоры отличаются от предыдущего наличием одного или двух инверсных входов, что отражено на их условном графическом обозначении.



Рис. П.29. Условное графическое обозначение мультиплексоров с инверсными входами

### ***Мультиплексор 2-1 с входом разрешения (M2\_1E)***

Условное графическое обозначение мультиплексора M2\_1E приведено на рис. П.30.

Вход *E* разрешает работу мультиплексора. Когда разрешающий вход  $E = 1$ , то мультиплексор M2\_1E выбирает один из информационных входов (*D1* или *D0*) в зависимости от состояния управляющего входа *S0*. При  $E = 0$  состояние выхода равно «0».

Таблица истинности мультиплексора M2\_1E представлена табл. П.12.

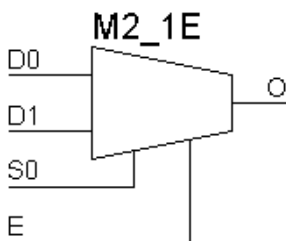


Рис. П.30. Условное графическое обозначение мультиплексора M2\_1E

Таблица П.12

Таблица истинности мультиплексора M2\_1E

| Входы |    |    |    | Выход |
|-------|----|----|----|-------|
| E     | S0 | D1 | D0 | O     |
| 0     | X  | X  | X  | 0     |
| 1     | 0  | X  | 1  | 1     |
| 1     | 0  | X  | 0  | 0     |
| 1     | 1  | 1  | X  | 1     |
| 1     | 1  | 0  | X  | 0     |

### **Мультиплексор 4-1 с входом разрешения (M4\_1E)**

Условное графическое обозначение мультиплексора M4\_1E приведено на рис. П.31.

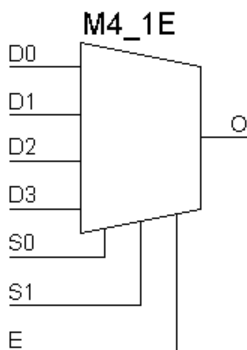


Рис. П.31. Условное графическое обозначение мультиплексора M4\_1E

Вход E разрешает работу мультиплексора. Когда разрешающий вход  $E = 1$ , то мультиплексор M4\_1E выбирает один из четырех информационных входов (D3, D2, D1 или D0) в зависимости от со-

стояния управляющих (адресных) входов  $S1$  и  $S0$ . При  $E = 0$  состояние выхода равно «0».

Таблица истинности мультиплексора  $M4\_1E$  представлена табл. П.13.

Таблица П.13

Таблица истинности мультиплексора  $M4\_1E$

| Входы |      |      |      |      |      |      | Выход |
|-------|------|------|------|------|------|------|-------|
| $E$   | $S1$ | $S0$ | $D0$ | $D1$ | $D2$ | $D3$ | $O$   |
| 0     | X    | X    | X    | X    | X    | X    | 0     |
| 1     | 0    | 0    | $D0$ | X    | X    | X    | $D0$  |
| 1     | 0    | 1    | X    | $D1$ | X    | X    | $D1$  |
| 1     | 1    | 0    | X    | X    | $D2$ | X    | $D2$  |
| 1     | 1    | 1    | X    | X    | X    | $D3$ | $D3$  |

Структура мультиплексора  $M4\_1E$  приведена на рис. П.32.

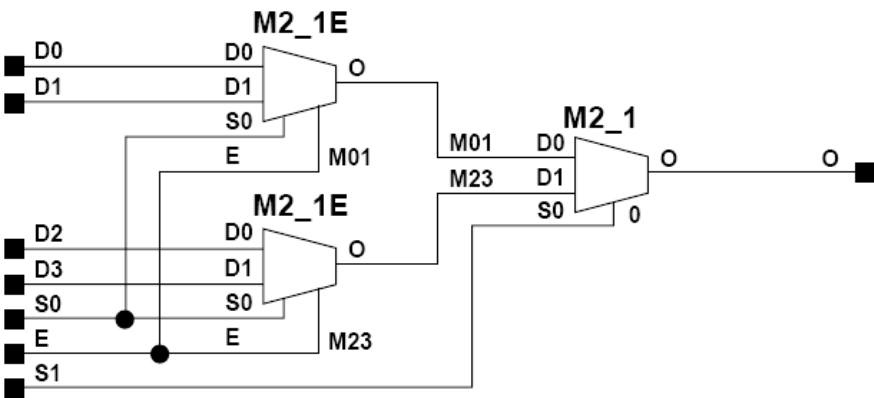


Рис. П.32. Внутренняя структура мультиплексора  $M4\_1E$

### Дешифратор

#### *Дешифратор 2-4 с входом разрешения ( $D2\_4E$ )*

Условное графическое обозначение дешифратора  $D2\_4E$  приведено на рис. П.33.

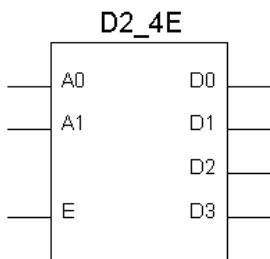


Рис. П.33. Условное графическое обозначение дешифратора D2\_4E

Дешифратор преобразует двухразрядный двоичный код, поступающий на входы *A1*, *A0*, в четырехразрядный унитарный код на выходах *D0÷D3*, при наличии сигнала высокого уровня на входе разрешения *E* (табл. П.14).

Дешифратор D2\_4E является макроэлементом, его структура приведена на рис. П.34.

Таблица П.14

Таблица истинности дешифратора D2\_4E

| Входы    |           |           | Выходы    |           |           |           |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>E</i> | <i>A1</i> | <i>A0</i> | <i>D0</i> | <i>D1</i> | <i>D2</i> | <i>D3</i> |
| 0        | X         | X         | 0         | 0         | 0         | 0         |
| 1        | 0         | 0         | 1         | 0         | 0         | 0         |
| 1        | 0         | 1         | 0         | 1         | 0         | 0         |
| 1        | 1         | 0         | 0         | 0         | 1         | 0         |
| 1        | 1         | 1         | 0         | 0         | 0         | 1         |

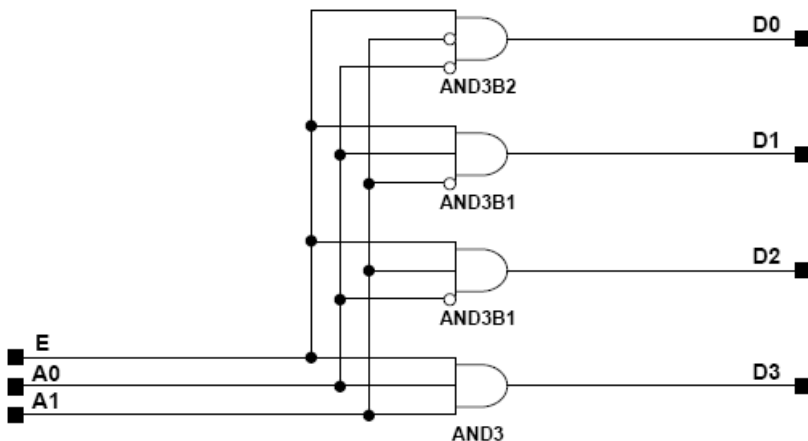


Рис. П.34. Внутренняя структура дешифратора D2\_4E

## Сумматоры

### *Четырехразрядный и восьмиразрядный сумматоры (ADD4 и ADD8)*

Условные графические обозначения сумматоров ADD4 и ADD8 приведены на рис. П.35.

Сумматоры складывают два операнда  $A$  и  $B$  с учетом состояния входа  $CI$ . На выходах сумматоров формируется сумма  $S$ , сигнал переноса ( $CO$  — carry-out) или сигнал переполнения ( $OFL$  — overflow). В зависимости от того, какое толкование пользователь дает операндам  $A$  и  $B$ , он может использовать или выходы  $S$  и  $CO$ , или  $S$  и  $OFL$  (табл. П.15).

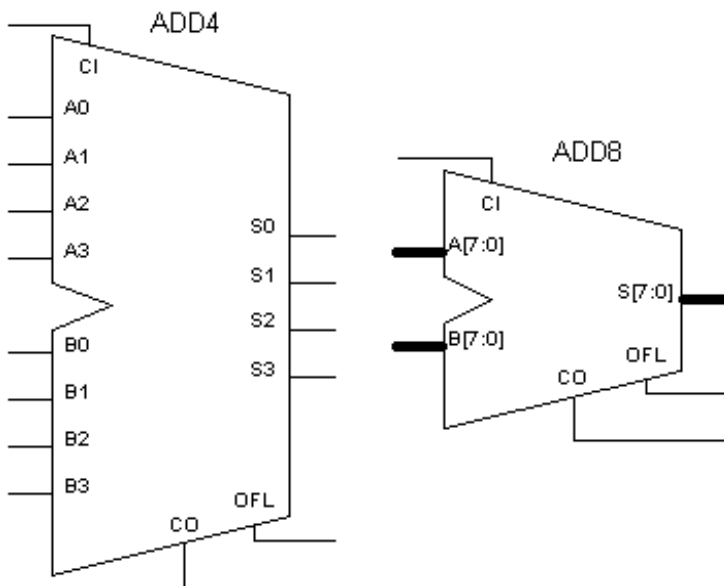


Рис. П.35. Условное графическое обозначение сумматоров ADD4 и ADD8:  $A$ ,  $B$  — входы данных;  $CI$  — вход переноса;  $S$  — сумма входных сигналов;  $OFL$  — выход переполнения (для операндов со знаком, представленных в дополнительном коде);  $CO$  — выход переноса (для операндов без знака)

Использование выходов  $CO$  и  $OFL$  сумматоров  $ADD4$  и  $ADD8$ 

| $S$      | $CO$ | $OFL$ | Режим работы                                                                                                                                      |
|----------|------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| $A+B+CI$ | 0    | X     | Суммирование двоичных кодов ( $S < 16$ для $ADD4$ и $S < 256$ для $ADD8$ )                                                                        |
| $A+B+CI$ | 1    | X     | Суммирование двоичных кодов ( $S \geq 16$ для $ADD4$ и $S \geq 256$ для $ADD8$ )                                                                  |
| $A+B+CI$ | X    | 0     | Суммирование двух операндов со знаком, представленных в дополнительном коде ( $S > -9$ или $< 8$ для $ADD4$ ; $S > -129$ или $< 128$ для $ADD8$ ) |
| $A+B+CI$ | X    | 1     | Суммирование двух операндов со знаком, представленных в дополнительном коде ( $S > 7$ или $< -8$ для $ADD4$ ; $S > 127$ или $< -128$ для $ADD8$ ) |

Данные сумматоры могут использоваться в двух режимах: или как сумматоры двоичных кодов, или как сумматоры чисел, представленных в дополнительном коде. В последнем случае крайний левый разряд операндов и результата является знаком числа. Функциональное различие между этими режимами состоит в определении «переполнения».

Если интерпретировать операнды как двоичные коды, то переполнение фиксируется на выходе  $CO$ , а состояние выхода  $OFL$  игнорируется (см. табл. П.15).

Если операнды рассматриваются как числа в дополнительном коде, то переполнение как положительное, так и отрицательное определяется на выходе  $OFL$  в виде единичного сигнала, а состояние выхода  $CO$  игнорируется (см. табл. П.15).

### Счётчики

Библиотека элементов содержит большое количество разнообразных счётчиков. Расшифровка обозначения счётчиков приведена на рис. П.36.

Двоичные счётчики в библиотеке элементов имеют разрядность 2, 4, 8 и 16. Их построение однотипно, поэтому рассмотрение двоичных счётчиков будет проведено на примере двухразрядных двоичных счётчиков с различным набором микроопераций.

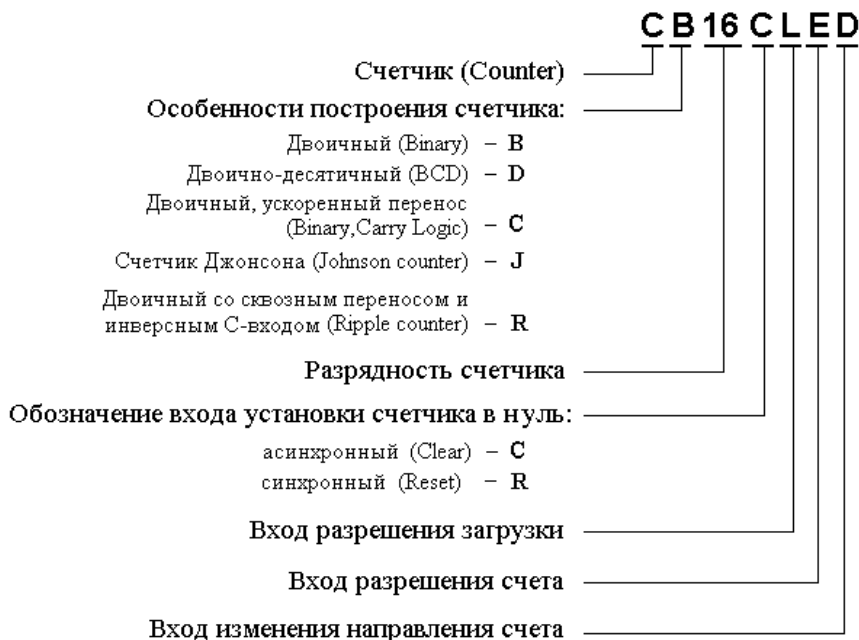


Рис. П.36. Обозначение счётчиков

### ***Двоичный счётчик с асинхронным сбросом и входом разрешения счёта (CB2CE, CB4CE, CB8CE, CB16CE)***

Условное графическое обозначение двухразрядного двоичного счётчика CB2CE приведено на рис. П.37, а его внутренняя структура — на рис. П.38.

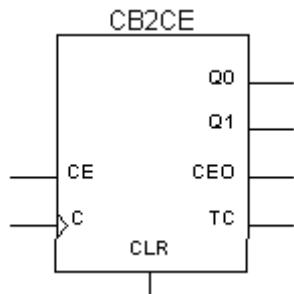
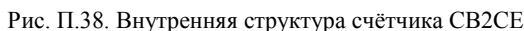


Рис. П.37. Условное графическое обозначение счётчика CB2CE: *C* — тактовый вход; *CE* — вход разрешения счёта; *CLR* — вход сброса; *Q1-Q0* — выходы (1-0 разряды); *TC* — выход переноса (выход завершения счёта); *CEO* — выход разрешения счёта ( $CEO = TC \cdot CE$ )



Многоразрядные счётчики получают путём соединения выхода *CEO* одного счётчика с входом *CE* следующего и объединением входов *C* и *CLR* всех счётчиков соответственно.

Условное графическое обозначение двухразрядного двоичного счётчика CB2CLE показано на рис. П.39. Микрооперации, выполняемые счётчиком CB2CLE, приведены в табл. П.16, а его внутренняя структура — на рис. П.40.

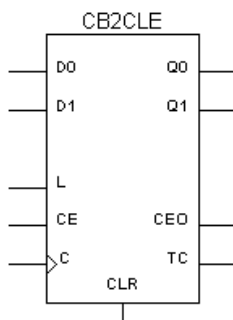


Рис. П.39. Условное графическое обозначение счётчика CB2CLE:  $C$  — тактовый вход;  $CE$  — вход разрешения счёта;  $L$  — вход разрешения параллельной загрузки;  $CLR$  — вход сброса;  $D1-D0$  — параллельные входы данных;  $Q1-Q0$  — выходы (1-0 разряды);  $TC$  — выход переноса (выход завершения счёта);  $CEO$  — выход разрешения счёта ( $CEO = TC \cdot CE$ )

Таблица П.16

Микрооперации счётчика CB2CLE

| Входы |     |      |     | Микрооперация            |
|-------|-----|------|-----|--------------------------|
| $CLR$ | $L$ | $CE$ | $C$ |                          |
| 1     | X   | X    | X   | Асинхронный сброс в нуль |
| 0     | 1   | X    | 0/1 | Параллельная загрузка    |
| 0     | 0   | 1    | 0/1 | Счёт                     |
| 0     | 0   | 0    | X   | Хранение                 |

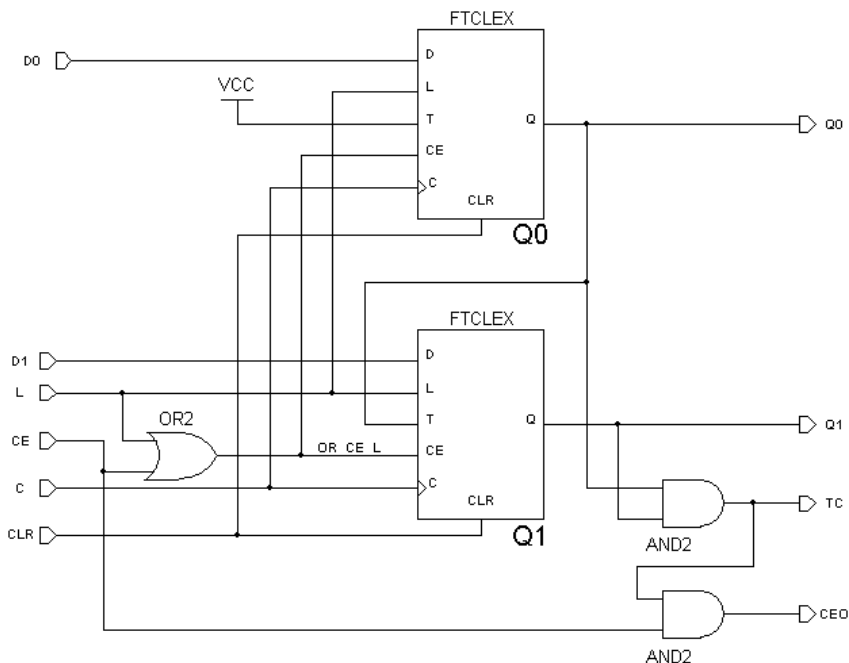


Рис. П.40. Внутренняя структура счётчика CB2CLE

**Двоичный реверсивный счётчик с параллельной загрузкой,  
асинхронным сбросом и входом разрешения счёта  
(CB2CLED, CB4CLED, CB8CLED, CB16CLED)**

Условное графическое обозначение двухразрядного двоичного счётчика CB2CLED показано на рис. П.41. Микрооперации, выполняемые счётчиком CB2CLED, приведены в табл. П.17, а его внутренняя структура — на рис. П.42.

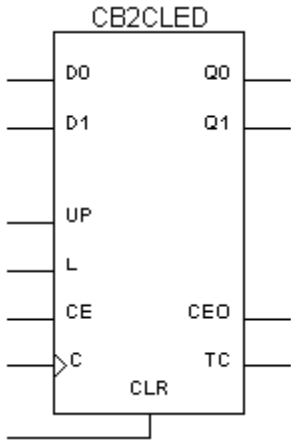


Рис. П.41. Условное графическое обозначение счётчика CB2CLED: *C* — тактовый вход; *CE* — вход разрешения счёта; *L* — вход разрешения параллельной загрузки; *UP* — вход изменения направления счёта; *CLR* — вход сброса; *D1-D0* — параллельные входы данных; *Q1-Q0* — выходы (1-0 разряды); *TC* — выход переноса (выход завершения счёта); *CEO* — выход разрешения счёта ( $CEO = TC \cdot CE$ )

Таблица П.17

Микрооперации счётчика CB2CLED

| Входы      |          |           |          |           | Микрооперация            |
|------------|----------|-----------|----------|-----------|--------------------------|
| <i>CLR</i> | <i>L</i> | <i>CE</i> | <i>C</i> | <i>UP</i> |                          |
| 1          | X        | X         | X        | X         | Асинхронный сброс в нуль |
| 0          | 1        | X         | 0/1      | X         | Параллельная загрузка    |
| 0          | 0        | 1         | 0/1      | 1         | Счёт на увеличение       |
| 0          | 0        | 1         | 0/1      | 0         | Счёт на уменьшение       |
| 0          | 0        | 0         | X        | X         | Хранение                 |

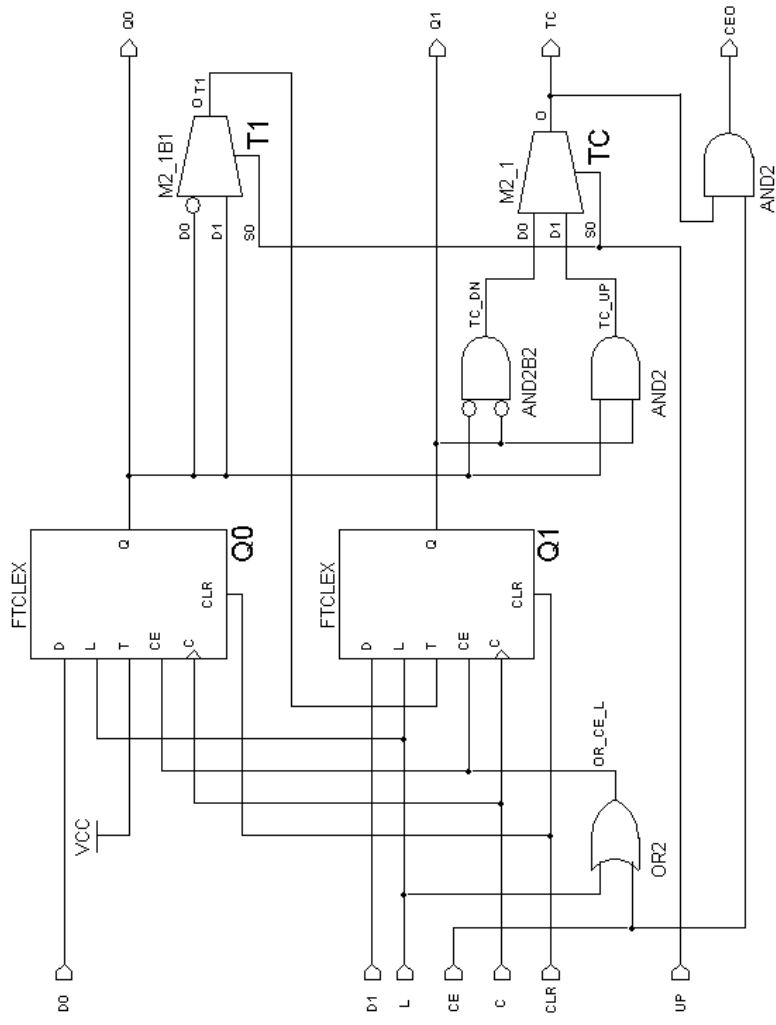


Рис. П.42. Внутренняя структура счётчика CB2CLED

## Регистры

Регистры по количеству возможных вариантов дополнительных управляющих входов почти так же многочисленны, как триггеры и счетчики. Расшифровка обозначения этих дополнительных управляющих входов полностью совпадает с их обозначением у счётчиков (см. рис. П.36). Обозначение регистров отличается от обозначения счётчиков в первых двух символах.

Регистры хранения (Data Register), которые представляют собой совокупность D-триггеров, объединённых общими управляющими входами, обозначаются как FDxxx. Регистры сдвига (Shift Register) имеют обозначение SRxxx.

Регистры в библиотеке элементов имеют разрядность 4, 8 и 16. Их построение однотипно, поэтому рассмотрение будет проведено на примере четырехразрядных регистров с различным набором микроопераций.

### *Регистр хранения с асинхронным сбросом и разрешением тактового сигнала (FD4CE, FD8CE, FD16CE)*

Условное графическое обозначение четырёхразрядного регистра хранения FD4CE показано на рис. П.43, а его внутренняя структура — на рис. П.44.

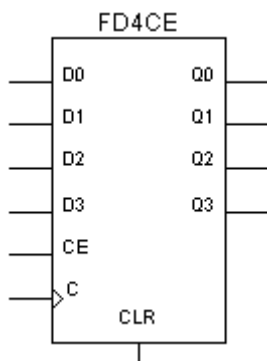


Рис. П.43. Условное графическое обозначение регистра FD4CE: D0-D3 — информационные входы D; C — вход синхронизации; CLR — вход сброса регистра в нуль; CE — вход разрешения тактового сигнала; Q0-Q3 — выходы регистра

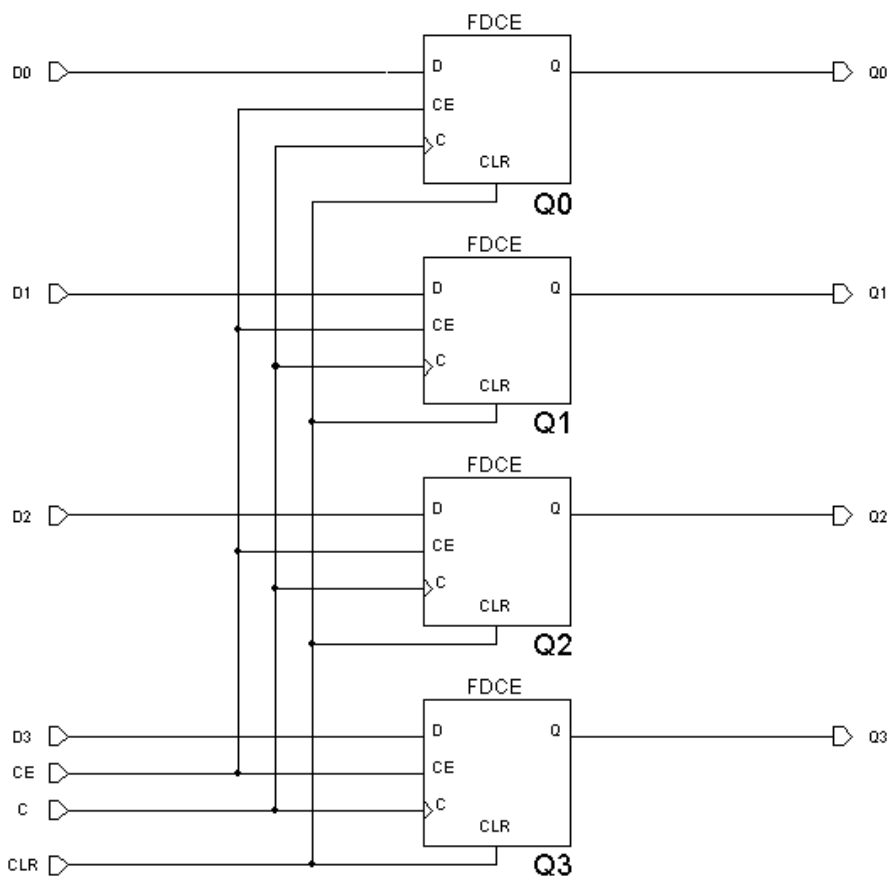


Рис. П.44. Внутренняя структура регистра FD4CE

***Сдвиговый регистр с асинхронным сбросом,  
входом разрешения тактового сигнала, последовательным  
и параллельными входами данных и параллельными выходами  
(SR4CLE, SR8CLE, SR16CLE)***

Условное графическое обозначение четырёхразрядного регистра сдвига SR4CLE показано на рис. П.45. Микрооперации, выполняе-

мые регистром сдвига SR4CLE, приведены в табл. П.18, а его внутренняя структура — на рис. П.46.

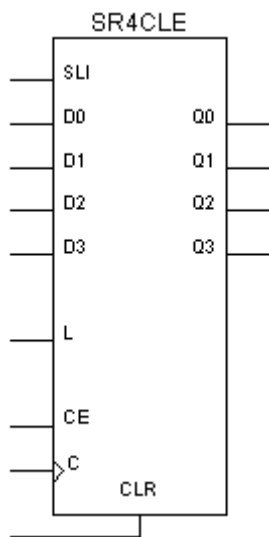


Рис. П.45. Условное графическое обозначение регистра сдвига SR4CLE: *C* — вход синхронизации; *CE* — вход разрешения тактового сигнала; *CLR* — вход сброса регистра в нуль; *SLI* — последовательный вход данных; *L* — вход разрешения параллельной загрузки; *D0-D3* — параллельные входы данных 0-3 разряды; *Q0-Q3* — выходы (0-3 разряды)

Таблица П.18

Микрооперации регистра сдвига SR4CLE

| Входы      |          |           |            |          | Выходы    |           |           |           |
|------------|----------|-----------|------------|----------|-----------|-----------|-----------|-----------|
| <i>CLR</i> | <i>L</i> | <i>CE</i> | <i>SLI</i> | <i>C</i> | <i>Q3</i> | <i>Q2</i> | <i>Q1</i> | <i>Q0</i> |
| 1          | X        | X         | X          | X        | 0         | 0         | 0         | 0         |
| 0          | 1        | X         | X          | 0/1      | <i>D3</i> | <i>D2</i> | <i>D1</i> | <i>D0</i> |
| 0          | 0        | 1         | 0          | 0/1      | <i>Q2</i> | <i>Q1</i> | <i>Q0</i> | 0         |
| 0          | 0        | 1         | 1          | 0/1      | <i>Q2</i> | <i>Q1</i> | <i>Q0</i> | 1         |
| 0          | 0        | 0         | X          | X        | <i>Q3</i> | <i>Q2</i> | <i>Q1</i> | <i>Q0</i> |

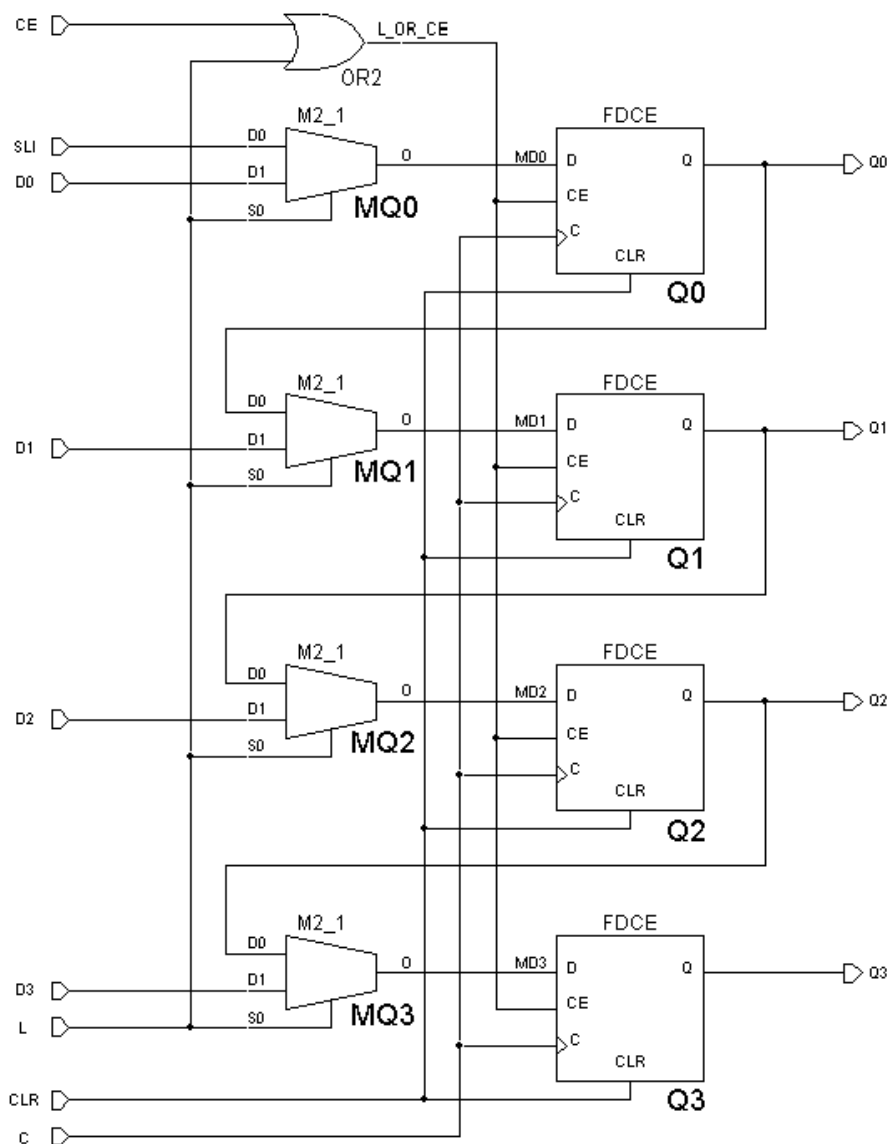


Рис. П.46. Внутренняя структура регистра сдвига SR4CLE

**Реверсивный сдвиговый регистр с асинхронным сбросом,  
входом разрешения тактового сигнала, последовательным  
и параллельными входами данных и параллельными выходами  
(SR4CLED, SR8CLED, SR16CLED)**

Условное графическое обозначение четырехразрядного реверсивного регистра сдвига SR4CLED показано на рис. П.47. Микрооперации, выполняемые регистром сдвига SR4CLED, приведены в табл. П.19, а его внутренняя структура — на рис. П.48.

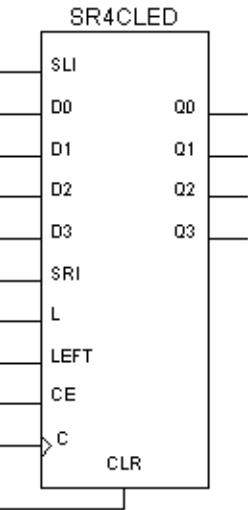


Рис. П.47. Условное графическое обозначение регистра сдвига SR4CLED: *C* — вход синхронизации; *CE* — вход разрешения тактового сигнала; *CLR* — вход сброса регистра в нуль; *SLI* — последовательный вход данных при сдвиге от Q0 к Q3; *SRI* — последовательный вход данных при сдвиге от Q3 к Q0; *L* — вход разрешения параллельной загрузки; *LEFT* — вход сигнала направления сдвига, *D0-D3* — параллельные входы данных (0-3 разряды); *Q0-Q3* — выходы (0-3 разряды)

Таблица П.19

Микрооперации регистра сдвига SR4CLED

| Входы      |          |           |             |          | Выходы     |           |           |            |
|------------|----------|-----------|-------------|----------|------------|-----------|-----------|------------|
| <i>CLR</i> | <i>L</i> | <i>CE</i> | <i>LEFT</i> | <i>C</i> | <i>Q3</i>  | <i>Q2</i> | <i>Q1</i> | <i>Q0</i>  |
| 1          | X        | X         | X           | X        | 0          | 0         | 0         | 0          |
| 0          | 1        | X         | X           | 0/1      | <i>D3</i>  | <i>D2</i> | <i>D1</i> | <i>D0</i>  |
| 0          | 0        | 1         | 0           | 0/1      | <i>SRI</i> | <i>Q3</i> | <i>Q2</i> | <i>Q1</i>  |
| 0          | 0        | 1         | 1           | 0/1      | <i>Q2</i>  | <i>Q1</i> | <i>Q0</i> | <i>SLI</i> |
| 0          | 0        | 0         | X           | X        | <i>Q3</i>  | <i>Q2</i> | <i>Q1</i> | <i>Q0</i>  |

У реверсивного сдвигового регистра направление сдвига определяется состоянием управляющего входа *LEFT*. В случае сдвига «вправо» (в сторону младших индексов) старший разряд принимает значение, равное состоянию на входе *SRI*.

При сдвиге влево предыдущее состояние выходов сдвигается в сторону больших индексов. При этом в самый младший разряд вдвигается новое значение, которое присутствует на информационном входе *SLI*.

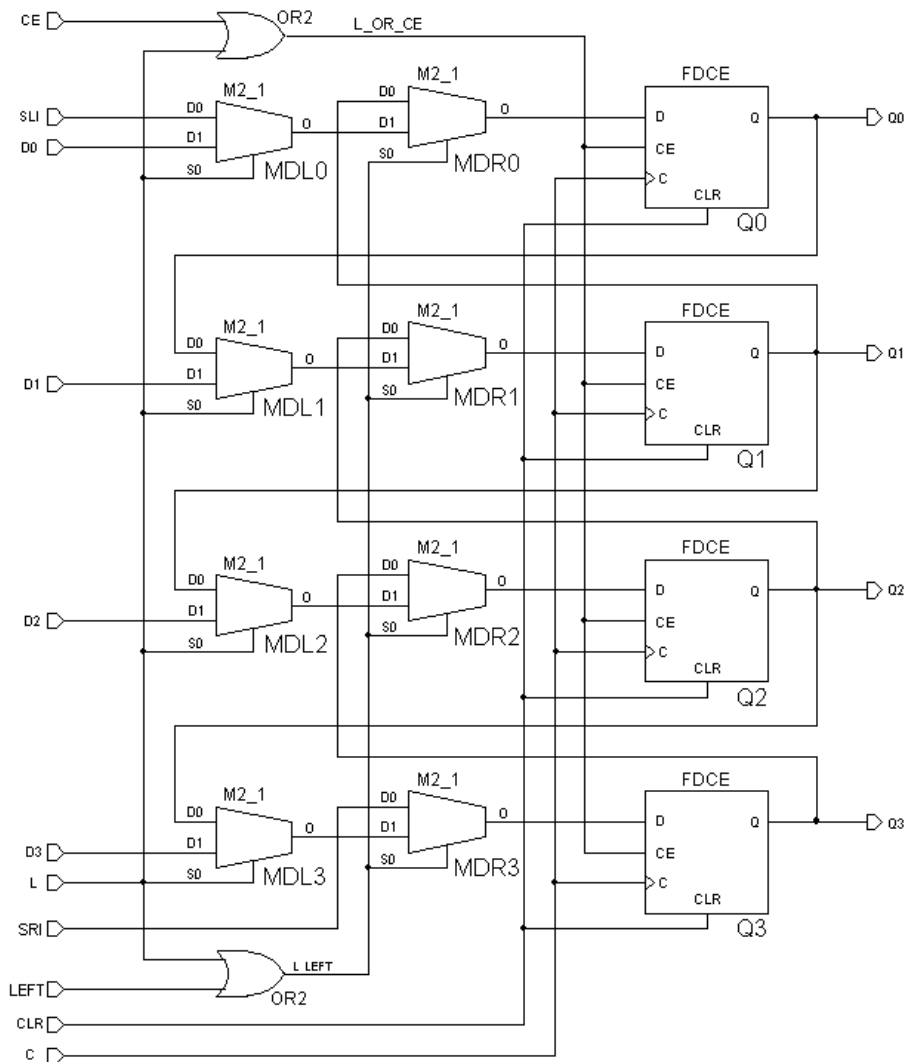


Рис. П.48. Внутренняя структура реверсивного регистра сдвига SR4CLED

## Список литературы

1. Бабич Н.П., Жуков И.А. Основы цифровой схемотехники: Учебн. пос. – М.: Изд. дом «Додэка-XXI», К.: «МК-Пресс», 2007. – 480 с.
2. Блейсли Т.Р. (США). Проектирование цифровых устройств с малыми и большими интегральными схемами: Пер. с англ. – Киев: Вища школа. Головное изд-во, 1981. – 336 с.
3. Будинский Я. Логические цепи в цифровой технике. Пер. с чешск. К. Юнга. /Под ред. Б.А. Калабекова. М.: Связь, 1977. – 392 с.
4. Бурдаев О.В., Иванов М.А., Тетерин И.И. Ассемблер в задачах защиты информации. /Под ред. И.Ю. Жукова. – М.: КУДИЦ-ОБРАЗ, 2002. – 318 с.
5. Голдсуорт Б. Проектирование цифровых логических устройств. /Пер. с англ. М.В. Сергиевского. /Под ред. Ю.И. Топчиева. – М.: Машиностроение, 1985. – 288 с.
6. Дж. Ф. Уэйкерли. Проектирование цифровых устройств. В 2-х т. М.: Постмаркет, 2002. – Т. 1, 544 с.; Т. 2, 544 с.
7. Ковригин Б.Н. Введение в инструментальные средства проектирования и отладки цифровых устройств на ПЛИС: Учебно-методическое пособие. – М.: МИФИ, 2006. – 192 с.
8. Лехин С.Н. Схемотехника ЭВМ. – СПб.: БХВ-Петербург, 2010. – 672 с.
9. Микроэлектронные устройства программного и логического управления. Принципы построения / Ю.О. Абугув., К.И. Диденко, Г.И. Загарий и др. – М.: Машиностроение, 1979. – 208 с.
10. Воробьев Н. Мультиплексор как многофункциональный узел. CHIPNEWS, 1999, №2.
11. Новиков Ю.В. Введение в цифровую схемотехнику. Учебн. пос. – М.: Интернет-университет информационных технологий; Бином. Лаборатория знаний, 2006. – 272 с. (Серия «Основы информационных технологий»).
12. Потемкин И.С. Функциональные узлы цифровой автоматики. – М.: Энергоатомиздат, 1988. – 320 с.
13. Применение интегральных схем: Практическое руководство. В 2-х кн. Пер с англ. /Под ред. А. Уильямса. – М.: Мир, 1987. – Кн. 1, 432 с.; Кн. 2, 413 с.

14. Проектирование микроэлектронных цифровых устройств. М.: Сов. радио, 1977. – 272 с. /Авторы: О.А. Пятлин, П.И. Овсищер, И.М. Лазер и др.

15. Пухальский Г.И., Новосельцева Т.Я. Цифровые устройства: Учебн. пос. для втузов. – СПб.: Политехника, 1996. – 885 с.

16. Стохастические методы и средства защиты информации в компьютерных системах и сетях. /Иванов М.А., Михайлов Д.М., Чугунков И.В., Ковалев А.В., Мацук Н.А.: Ред. И.Ю. Жуков: – М.: КУДИЦ-ПРЕСС, 2009. – 510 с.

17. Схемотехника ЭВМ. Лабораторный практикум: учебн. пос. /Под ред. Б.Н. Ковригина. Изд. 3-е, перераб. и доп. – М.: МИФИ, 2006. – 212 с. /Авторы: М.Н. Ёхин, Б.Н. Кальнин, Б.Н. Ковригин, В.Г. Тышкевич, И.М. Ядыкин.

18. Титце У., Шенк К. Полупроводниковая схемотехника: Справочное руководство. /Пер. с нем. – М.: Мир, 1982. – 512 с.

19. Токхейм Р. Основы цифровой электроники: Пер. с англ. – М.: Мир, 1988. – 392 с.

20. Трачик В. Дискретные устройства автоматики: /Пер. с польск. /Под ред. Д.А. Поспелова. – М.: Энергия, 1978. – 456 с.

21. Угрюмов Е.П. Цифровая схемотехника: учебн. пособие для вузов. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2010. – 816 с.

22. Фридман А., Менон П. Теория и проектирование переключательных схем. – М.: Мир, 1978. – 582 с.

23. Хоуп Г. Проектирование цифровых вычислительных устройств на интегральных схемах. /Пер. с англ. – М.: Мир, 1984. – 400 с.

24. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения: Справочник. – 2-е изд. перераб. и доп. – М.: Радио и связь, 1990. – 512 с.

25. Янсен Й. Курс цифровой электроники: В 4-х т. Т. 2. Проектирование устройств на цифровых ИС. /Пер. с голланд. – М.: Мир, 1987. – 368 с.

26. Янсен И. Курс цифровой электроники: В 4-х т. Т. 3. Сложные ИС для устройств передачи данных. /Пер. с голл. Фридман А., Менон П. Теория и проектирование переключательных схем. – М.: Мир, 1978. : Мир, 1987. – 412 с.

27. Libraries Guide 3.3.06i. Xilinx Development System. – 990 p.

Никита Алексеевич Дмитриев  
Михаил Николаевич Ёхин  
Михаил Александрович Иванов  
Борис Николаевич Ковригин  
Григорий Григорьевич Новиков  
Владимир Георгиевич Тышкевич  
Игорь Михайлович Ядыкин

## **СХЕМОТЕХНИКА ЭВМ**

Сборник задач

Учебное пособие

Редактор Коцарева Е.К.

Подписано в печать 15.11.2012 Формат 60х84 1/16  
Печ. л. 15,0. Уч.-изд. л. 15,0. Тираж 350 экз.  
Изд. №34/1 Заказ №48

*Национальный исследовательский ядерный университет «МИФИ»  
115409, Москва, Каширское ш., 31*

*ООО «Полиграфический комплекс «Курчатовский»  
144000, Московская область, г. Электросталь, ул. Красная, д. 42*