

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И УПРАВЛЕНИЯ В ТЕХНИЧЕСКИХ СИСТЕМАХ

Кафедра информационных технологий и компьютерных систем

СИНТЕЗ ОПЕРАЦИОННЫХ ЭЛЕМЕНТОВ

Методические указания

к циклу лабораторных работ по дисциплине

«Архитектура компьютерных систем»

для студентов направлений

09.03.01 – Информатика и вычислительная техника и

09.03.04 – Программная инженерия

дневной формы обучения

Севастополь

СевГУ

2020

УДК 004.3(076.5)
ББК 32.973-04я7
С380

Р е ц е н з е н т:

В.С. Чернега - канд. техн. наук, доцент,
доцент кафедры информационных систем

Составители: Ю.К. Апраксин, Т.В. Волкова

С380 Синтез операционных элементов : методические указания к циклу лабораторных работ по дисциплине «Архитектура компьютерных систем» для студентов направлений 09.03.01 – Информатика и вычислительная техника и 09.03.04 – Программная инженерия дневной формы обучения / Сост. Ю.К. Апраксин, Т.В. Волкова ; Министерство образования и науки Российской Федерации, Севастопольский государственный университет, Институт информационных технологий и управления в технических системах, кафедра информационных технологий и компьютерных систем. – Севастополь : СевГУ, 2020. – 76 с. – Текст : электронный.

Методические указания предназначены для того, чтобы облегчить выполнение лабораторных работ, включенных в цикл «Синтез операционных элементов», углубить знания студентов в области проектирования операционных автоматов и составляющих их элементов, помочь обрести необходимые навыки сборки и наладки дискретных вычислительных и управляющих устройств. Методические указания к каждой лабораторной работе содержат четкую постановку задачи, варианты заданий, методические рекомендации к домашней подготовке к лабораторной работе, последовательность ее выполнения на лабораторном макете и содержание отчета. Приводится описание лабораторных макетов и справочные данные о наиболее употребительных из используемых в лабораторных работах интегральных схемах.

УДК 004.3(076.5)
ББК 32.973-04я7

Методические указания рассмотрены и утверждены на заседании кафедры информационных технологий и компьютерных систем, протокол № 3 от 6 ноября 2019 г.

Допущено учебно-методическим центром СевГУ в качестве методических указаний.

СОДЕРЖАНИЕ

Введение.....	4
1. Лабораторная работа №1. Исследование работы элементов компьютерной логики.....	7
2. Лабораторная работа №2. Синтез комбинационных и триггерных схем	18
3. Лабораторная работа №3. Выполнение микроопераций в регистрах	19
4. Лабораторная работа №4. Комбинационные двоичные счетчики	26
5. Лабораторная работа №5. Накапливающие двоичные счетчики	31
6. Лабораторная работа №6. Схемы сравнения кодов.....	39
7. Лабораторная работа №7. Дешифраторы и мультиплексоры и их использование для организации передач в системе регистров	44
8. Лабораторная работа №8. Двоичные сумматоры	54
9. Лабораторная работа №9. Распределители сигналов	64
10. Лабораторная установка «Logic-Trigger»	70
Библиографический список	75

ВВЕДЕНИЕ

Методические указания предназначены для выполнения лабораторных работ по дисциплине «Архитектура компьютерных систем».

Назначение компьютера – обработка входных данных в соответствии с программой, заданной пользователем, с целью получения выходных данных.

Архитектура компьютера – это совокупность его свойств и характеристик. Важнейшие архитектурные решения, положенные в основу создания компьютера в целом и отдельных его узлов, задают принципы работы вычислительной системы на функциональном уровне безотносительно физической реализации.

Современный компьютер состоит из совокупности операционных устройств, взаимодействующих друг с другом с целью решения задач пользователей.

Под операционным устройством при этом понимается устройство, выполняющее множество операций F над входными словами A с целью получения результата B (рисунок 1).

В соответствии со структурной теорией цифровых автоматов, к классу которых относятся все типы ОУ компьютера, любое из них можно рассматривать как совокупность двух взаимосвязанных устройств – операционного автомата (ОА) и управляющего автомата (УА) (рисунок 2) [1,2]. В свою очередь, ОА состоит из операционных элементов (ОЭ): двоичных сумматоров, счетчиков, сдвиговых регистров и т.п., и предназначен для преобразования информации под воздействием управляющих сигналов $\{y_i\}$, формируемых УА. ОА может сообщить УА своему состоянию после выполнения того или иного действия осведомительными сигналами $\{x_j\}$, $x_j \in \{0,1\}$.

УА осуществляет управление преобразованием данных в ОА в

соответствии с принципом микропрограммного управления.

Принцип микропрограммного управления состоит в следующем: каждая сложная операция обработки данных (например, умножение двоичных чисел с фиксированной точкой) реализуется как микропрограмма.

Микропрограмма – это последовательность элементарных операций. Каждая i -ая элементарная операция инициируется соответствующим сигналом y_i и выполняется ОА за один

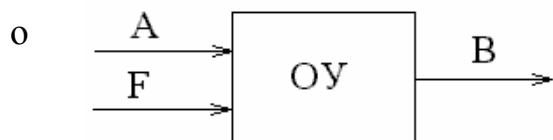


Рисунок 1 — Операционное устройство

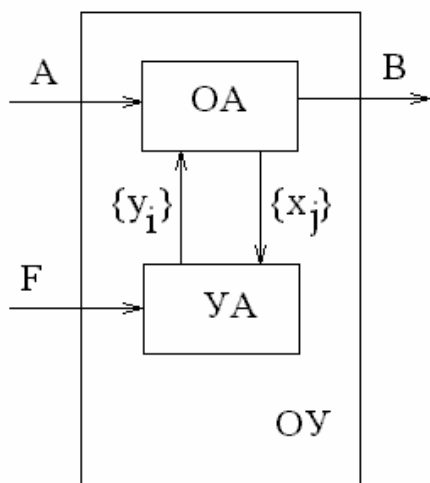


Рисунок 2 — Схема взаимодействия операционного и управляющего автоматов

такт. Под тактом понимается время между двумя синхроимпульсами, синхронизирующими работу ОУ. Элементарные операции называются микрооперациями. Сигналы y_i вырабатываются УА и называются управляющими сигналами или сигналами микроопераций.

Примеры микроопераций:

$y1: P(1:n) := Q(1:n)$ - присвоение n -разрядному регистру P значения, хранящегося на n -разрядном регистре Q ,

$y2: C(1:k) := C(1:k) - 1$ - декрементирование содержимого k -разрядного счетчика C ,

$y3: P(1:n) := P(2:n).0$ - сдвиг слова в регистре P на один разряд влево с доопределением освобождающегося правого разряда нулем.

Осведомительные сигналы x_j вырабатываются ОА и характеризуют промежуточные результаты выполнения микропрограммы.

Примеры осведомительных сигналов:

$x1 = (P(1:n) = 0)$ – сигнал $x1=1$, когда значение регистра P равно нулю, в противном случае $x1=0$ (для реализации сигнала $x1$ необходимо построить комбинационную схему, реализующую булеву функцию $x1 = \overline{p_1} \overline{p_2} \dots \overline{p_n}$, где p_i – выход i -го разряда регистра P),

$x2 = p_1$ – осведомительный сигнал $x2$ снимается непосредственно с первого разряда регистра P и равен значению этого разряда,

$x3 = (P(1:n) \geq Q(1:n))$ – сигнал $x3=1$, когда двоичный код в регистре P больше двоичного кода в регистре Q (такой осведомительный сигнал может быть реализован схемой сравнения кодов на больше, принципы построения цифровых компараторов (схем сравнения) рассмотрены ниже).

Предметом исследования в лабораторных работах являются ОЭ различного назначения. В общем виде структурная схема операционного элемента приведена на рисунке 3.

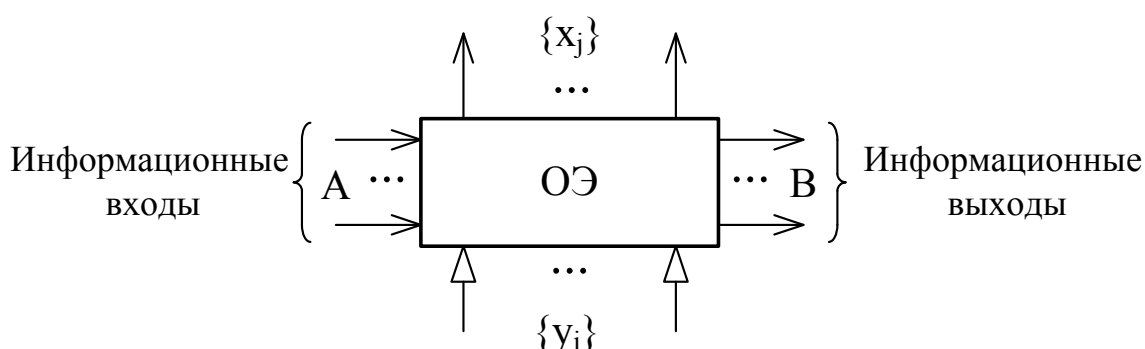


Рисунок 3 – Обобщенная схема операционного элемента

На информационные входы подаются слова для обработки (входные данные). Число информационных входов определяет разрядность обрабатываемых слов. Управляющие входы предназначены для подачи управляющих сигналов y_i , инициирующих выполнение ОЭ микроопераций. На осведомительных выходах формируются сигналы x_j , характеризующие как

состояние ОЭ, так и значения обрабатываемых слов. Информационные выходы представляют значения выходных слов (результатов выполнения микроопераций). Число информационных выходов (разрядность выходных слов) может отличаться от числа информационных входов.

Лабораторные работы преследуют цели изучения типовых схемных решений и приобретения навыков проектирования операционных элементов, изучения наиболее употребительных интегральных цифровых микросхем, а также приобретение опыта наладки и исследования этих элементов. Выполнение каждой лабораторной работы предусматривает внеаудиторную подготовку, выполнение работы в лаборатории и написание отчета. Для облегчения подготовки к лабораторной работе методические указания содержат ссылки на соответствующие литературные источники с указанием разделов, которые необходимо изучить перед выполнением лабораторной работы. После изучения (повторения) теоретического материала необходимо разработать схемы принципиальные электрические устройства в соответствии с индивидуальным заданием. Без такой подготовки студент к лабораторной работе не допускается.

В лаборатории необходимо собрать и отладить схемы на установке «Logic-Trigger». Описание установки приведено в разделе 10. Собирать схему необходимо при отключенном питании. Включать питание установки можно только с разрешения преподавателя, ведущего занятия. После наладки и исследования устройства необходимо продемонстрировать его работу преподавателю. При работе с установкой следует соблюдать правила техники безопасности, утвержденные для лаборатории.

После выполнения лабораторной работы необходимо представить и защитить отчет. Отчет должен содержать следующие разделы:

1. Цель работы.
2. Постановка задачи.
3. Модель разрабатываемого операционного элемента.
4. Тесты для проверки работы операционного элемента.
5. Разработка принципиальной схемы операционного элемента.
6. Результаты наладки и тестирования схемы. В этом разделе должны быть приведены временные диаграммы, основные характеристики схем, последовательность и результаты испытания схемы.
7. Выводы.

В процессе защиты работы студент должен продемонстрировать знание теоретического материала, умение проектировать, знание способов оптимизации схем с целью увеличения их быстродействия или упрощения структуры. Отчет о лабораторной работе должен быть представлен преподавателю на следующем после выполнения работы занятии.

1.ЛАБОРАТОРНАЯ РАБОТА № 1. ИССЛЕДОВАНИЕ РАБОТЫ ЭЛЕМЕНТОВ КОМПЬЮТЕРНОЙ ЛОГИКИ

1.1. Цель работы

Целью работы является исследование логики работы элементов И, ИЛИ, НЕ, получение навыков проектирования более сложных схем на базе этих элементов, изучение лабораторной установки «Logic-Trigger» (L-T), приобретение первых навыков сборки, наладки и исследования комбинационных и триггерных схем.

1.2. Краткие теоретические сведения

1.2.1. Выполнение логических операций в компьютере

Из курсов «Информатика» и «Программирование» известны поразрядные логические операции над двоичными векторами (последовательностями битов). Например, в языке Java подобные операции определены для целых типов данных и выполняются над всеми битами целого значения, включая знаковый в соответствии с таблицей 1.1.

Таблица 1.1 – Правила формирования результата логических операций над битами

i-ые биты слов A и B		AND $\wedge$	OR $\vee$	XOR $\oplus$	NOT $\overline{A_i}$
A_i	B_i	$A_i \& B_i$	$A_i B_i$	$A_i \wedge B_i$	$\sim A_i$
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

В таблице 1.1 использованы следующие обозначения:

& ($\wedge$) – логическое умножение, конъюнкция (И);

| ($\vee$) – логическое сложение, дизъюнкция (ИЛИ);

^ ($\oplus$) – сложение по модулю 2 (исключающее ИЛИ);

~ (–) – логическое отрицание, инверсия (НЕ).

В арифметико-логическом устройстве (АЛУ) компьютера указанные операции выполняются на специальных операционных элементах (ОЭ), построенных на базе логических схем И, ИЛИ, НЕ (элементах булевого базиса).

Логическим элементом называется электронная схема, реализующая элементарную логическую функцию, имеющая количество входов, равное числу аргументов функции, и один выход. Входные сигналы (сигналы,

поступающие на входы логических элементов) определяют значения входных переменных. Значение выходного сигнала является функцией значений входных сигналов. Значение входного или выходного сигнала представляет собой потенциал логического нуля или логической единицы.

На рисунке 1.1 изображены условно-графические обозначения логических элементов И (&), ИЛИ (1), И-НЕ (& с отрицанием (кружком) на выходе), ИЛИ-НЕ (1 с отрицанием на выходе), инвертор – НЕ (1 с одним входом и отрицанием на выходе), исключающее ИЛИ (=1). Элементы И-НЕ и ИЛИ-НЕ после выполнения функции И (ИЛИ) инвертирует результат. Таблицы истинности функций И-НЕ и ИЛИ-НЕ сделайте самостоятельно. Элементы И-НЕ и ИЛИ-НЕ можно использовать в качестве инвертора, если на все входы элемента подать один и тот же сигнал. Правила получения результата, представленные в таблице 1.1, распространяются на элементы с числом входов, большим двух:

- на выходе элемента И потенциал логической единицы будет только тогда, когда на все входы этого элемента поданы потенциалы логической единицы, в противном случае – потенциал логического нуля;
- на выходе элемента ИЛИ потенциал логического нуля будет тогда, когда на все входы подан потенциал логического нуля, в противном случае – потенциал логической единицы.

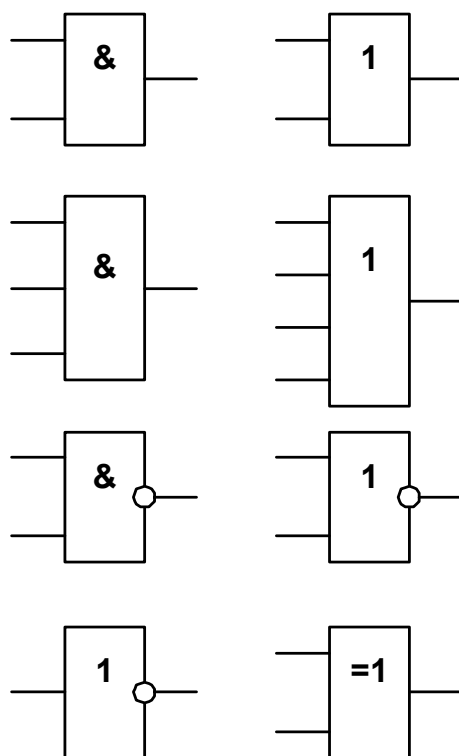


Рисунок 1.1 – Условно-графические обозначения логических элементов

Логические элементы выполняются в виде интегральных микросхем (ИМС), работающих от источника постоянного тока, и их характеристики существенно зависят от схемотехники конкретной серии микросхем. Так для микросхемы К155ЛА2 (серия К155, транзисторно-транзисторная логика, напряжение источника питания: +5В), изображенной на рисунке 1.2, потенциал логической единицы должен быть не менее 2,4 В, потенциал логического нуля – не более 0,4 В. Микросхема обеспечивает время переключения из 1 в 0 – не более 15 нс, время переключения из 0 в 1 – не более 22 нс. Коэффициент разветвления (число входов, которые можно подключить к одному выходу) – 10 [5].

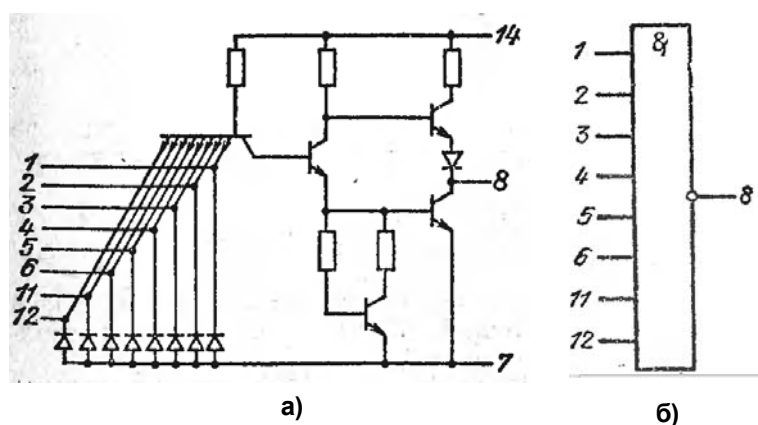


Рисунок 1.2 – Микросхема К155ЛА2:

- а) – схема электрическая (14 вывод должен быть подключен к +5В, 7 – к GND (общему проводу));
- б) – условно-графическое обозначение

Схемы, вычисляющие те или иные булевы функции и состоящие только из логических элементов (не содержащие элементов памяти) называют комбинационными схемами (комбинационными преобразователями).

В качестве примера комбинационной схемы на рисунке 1.3 изображена схема i -го разряда формирователя кодов, выполняющего для двоичных векторов A и B операции конъюнкции, дизъюнкции, инверсии A , инверсии B . Конкретная (j -ая) операция формирователя выбирается управляющим сигналом u_j . При подаче в схему управляющего сигнала u_1 над i -ми разрядами операндов (A_i и B_i) выполняется операция конъюнкции (И), u_2 – дизъюнкции (ИЛИ). Управляющий сигнал u_3 инициирует получение на выходе F_i результата инверсии разряда A_i , управляющий сигнал u_4 инициирует получение на выходе F_i результата инверсии разряда B_i . Таким образом выходной сигнал F_i является функцией от входных сигналов $A_i, B_i, u_1, u_2, u_3, u_4$.

Для создания подобного формирователя для n -разрядных кодов A и B нужно продублировать схему i -го разряда n раз.

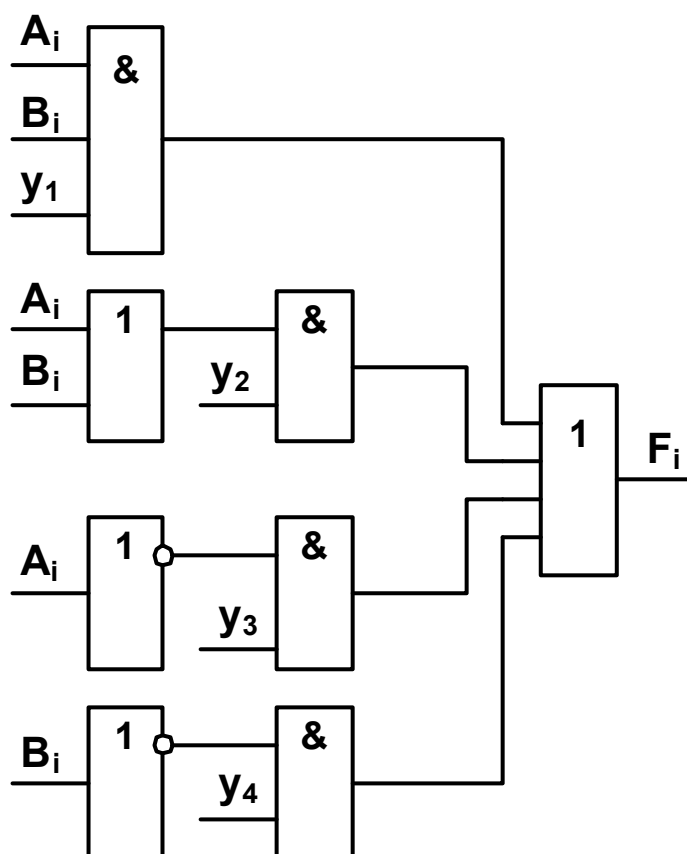


Рисунок 1.3 – Схема i -го разряда комбинационного преобразователя

1.2.2. Хранение информации в компьютере

Для хранения информации в компьютере используются электронные приборы с двумя состояниями. Двоичный элемент, который может переключаться в любое из двух возможных состояний и сохранять это состояние в течение требуемого промежутка времени называется двоичным запоминающим элементом. Состояние элемента в некоторый момент времени определяет бит информации. В логических схемах компьютера, выполняющих обработку информации, в качестве запоминающих элементов наиболее широко используются элементы задержки и триггеры с отдельными и счетными входами.

Элементы задержки.

Для запоминания сигнала на сравнительно небольшой промежуток времени используются элементы задержки, функция которых имеет вид $y(t + D) = x(t)$. Здесь x – входной сигнал, поступающий в момент времени t . Элемент «запоминает» значение x на промежуток времени D и благодаря этому выходной сигнал y переключается с задержкой D по отношению к переключениям входного сигнала x .

В качестве элементов задержки используются специальные электронные схемы, обеспечивающие задержку сигнала на заданную (относительно большую) величину. Для задержки сигнала на небольшие промежутки времени

могут использоваться инверторы, включенные по схеме, изображенной на рисунке 1.4. Если задержка сигнала на одном инверторе составляет τ , то в цепочке из N последовательно включенных инверторов сигнал задерживается на время $D = \tau N$.

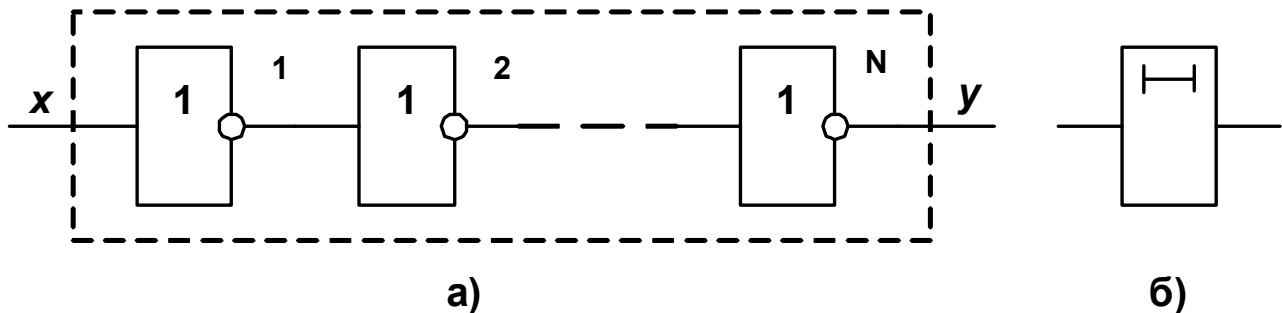


Рисунок 1.4 – Элемент задержки:

- а) – реализация элемента задержки на логических элементах;
 б) – условно-графическое обозначение

Триггеры с раздельными входами.

Для запоминания двоичной переменной на сколь угодно большой промежуток времени используются специальные электрические или элементарные логические схемы, называемые триггерными схемами (триггерами). В триггере с раздельными входами выделяют два входных сигнала R и S , два выходных сигнала Q и $\bar{Q}$ и два состояния, обозначаемых как 0 и 1. Триггер функционирует в соответствии с рисунком 1.5.

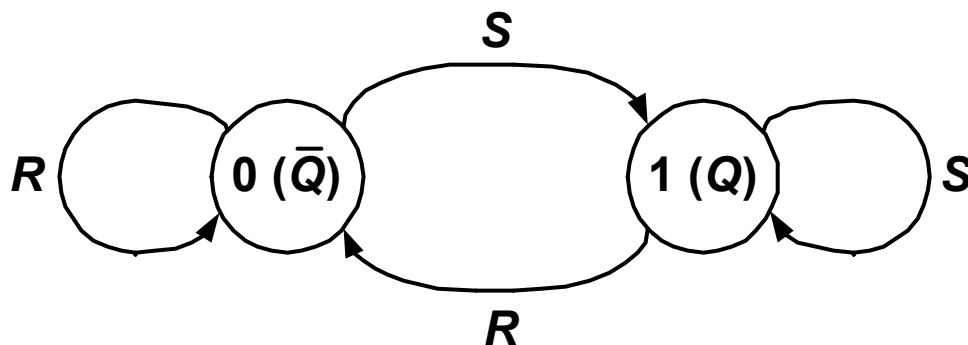


Рисунок 1.5 – Закон функционирования триггера с раздельными входами

Сигнал $R=1$ переключает триггер в нулевое состояние, которому соответствует выходной сигнал $\bar{Q}=1$. Сигнал $S=1$ переключает триггер в единичное состояние, что сопровождается выработкой сигнала $Q=1$. Входная комбинация $R=1, S=1$ является запрещенной.

Специальные электронные схемы триггеров будут рассмотрены в курсе «Компьютерная электроника». Весьма часто триггеры строятся на базе

логических элементов. Ограничиваясь логическими и информационными аспектами функционирования схем, будем рассматривать только логические схемы триггеров.

В базисе ИЛИ-НЕ триггер строится по схеме, изображенной на рисунке 1.6. Схема имеет два входа R и S . Выходные сигналы Q и $\bar{Q}$ связаны с входными сигналами R и S уравнениями:

$$\begin{cases} Q = \overline{\bar{Q} \vee R}; \\ \bar{Q} = \overline{Q \vee S}. \end{cases}$$

Этим уравнениям соответствует таблица 1.2, отображающая зависимость между значениями входных и выходных сигналов.

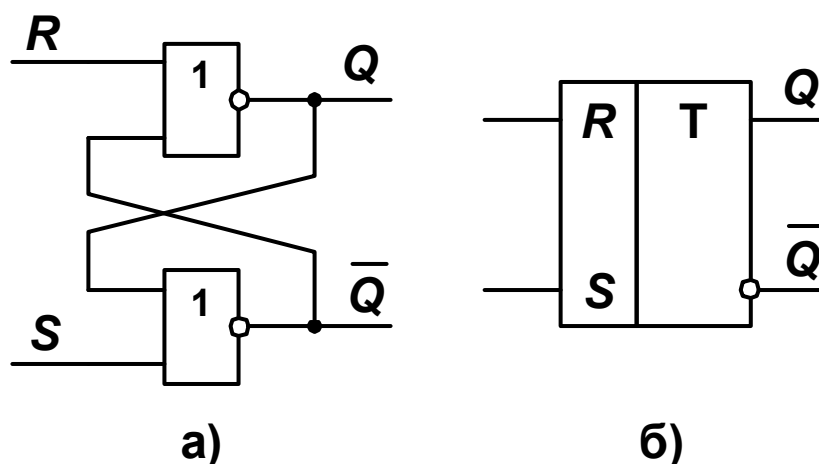


Рисунок 1.6 – Триггер с отдельными входами:

- а) – реализация в базисе ИЛИ-НЕ;
б) – условно-графическое обозначение

Таблица 1.2 – Значения входных и выходных сигналов триггера

Входные сигналы		Выходные сигналы	
R	S	Q	$\bar{Q}$
0	0	0	1
		1	0
0	1	1	0
1	0	0	1
1	1	0	0

Из таблицы 1.2 следует, что при $R=0$ и $S=0$ выходные сигналы Q и $\bar{Q}$ могут принадлежать одному из двух наборов $Q\bar{Q}=01$ или $Q\bar{Q}=10$. При нулевых значениях входных сигналов выходные сигналы сохраняют свои значения сколь угодно долго, т.е. состояния схемы являются устойчивыми.

Триггер имеет два устойчивых состояния: нулевое (0) и единичное (1). Состояние $Q\bar{Q}=01$ называется нулевым состоянием и $Q\bar{Q}=10$ – единичным состоянием триггера. Поскольку сигналы Q и $\bar{Q}$ являются взаимообратными, то состояние триггера определено, если задано значение одного из выходных сигналов, например, сигнала Q .

При поступлении на схему сигналов $R=1$ и $S=1$ выходные сигналы принимают значения $Q=0$ и $\bar{Q}=0$. По окончании действия входных единичных сигналов эта комбинация входных сигналов сохраняться не может (состояние схемы не является устойчивым), и схема должна перейти либо в состояние 0, либо в состояние 1. Оба эти состояния являются равноправными, и, следовательно, переход в конкретное состояние с физической точки зрения определяется случайными обстоятельствами. Если сигнал R переключается в 0 позже, чем сигнал S , то схема переключается в нулевое состояние. Исходя из вышесказанного, комбинация входных сигналов $R=1$ и $S=1$ считается запрещенной.

Поскольку для управления триггером используются два самостоятельных входа, то схема, изображенная на рисунке 1.6, называется триггером с отдельными входами.

В момент включения питающего напряжения триггер случайным образом устанавливается либо в состояние 0, либо в состояние 1. К моменту начала работы триггер должен быть приведен в требуемое состояние, обычно это состояние 0. Эта операция называется начальной установкой (сбросом) триггера.

В дальнейшем триггер с отдельными входами будем обозначать, как показано на рисунке 1.6, б, и использовать для описания его работы таблицу истинности (таблица 1.3).

Таблица 1.3 – Таблица истинности для триггера с отдельными входами (RS-триггера)

R	S	Q
0	0	Сохраняется предыдущее состояние
0	1	1
1	0	0
1	1	Не определено

Триггеры со счетным входом.

Для некоторых применений в триггере организуется счетный вход, по которому триггер переключается как в состояние 0, так и в состояние 1. Триггер со счетным входом функционирует в соответствии с рисунком 1.7.

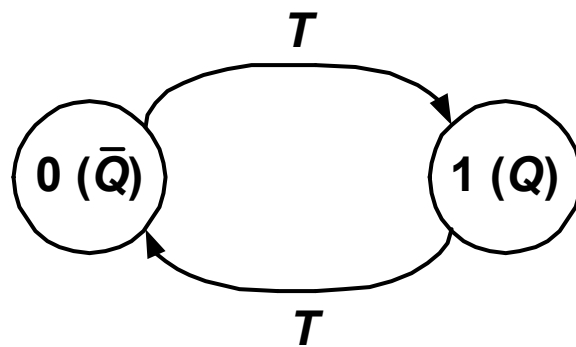


Рисунок 1.7 – Закон функционирования триггера со счетным входом

Последовательность сигналов на входе T приводит к поочередному переключению триггера в состояния $0,1,0,1,\dots$, причем предполагается, что каждый сигнал T должен вызывать единственный переход $(0,1)$ или $(1,0)$.

В базисе ИЛИ-НЕ триггер со счетным входом строится по схеме, изображенной на рисунке 1.8.

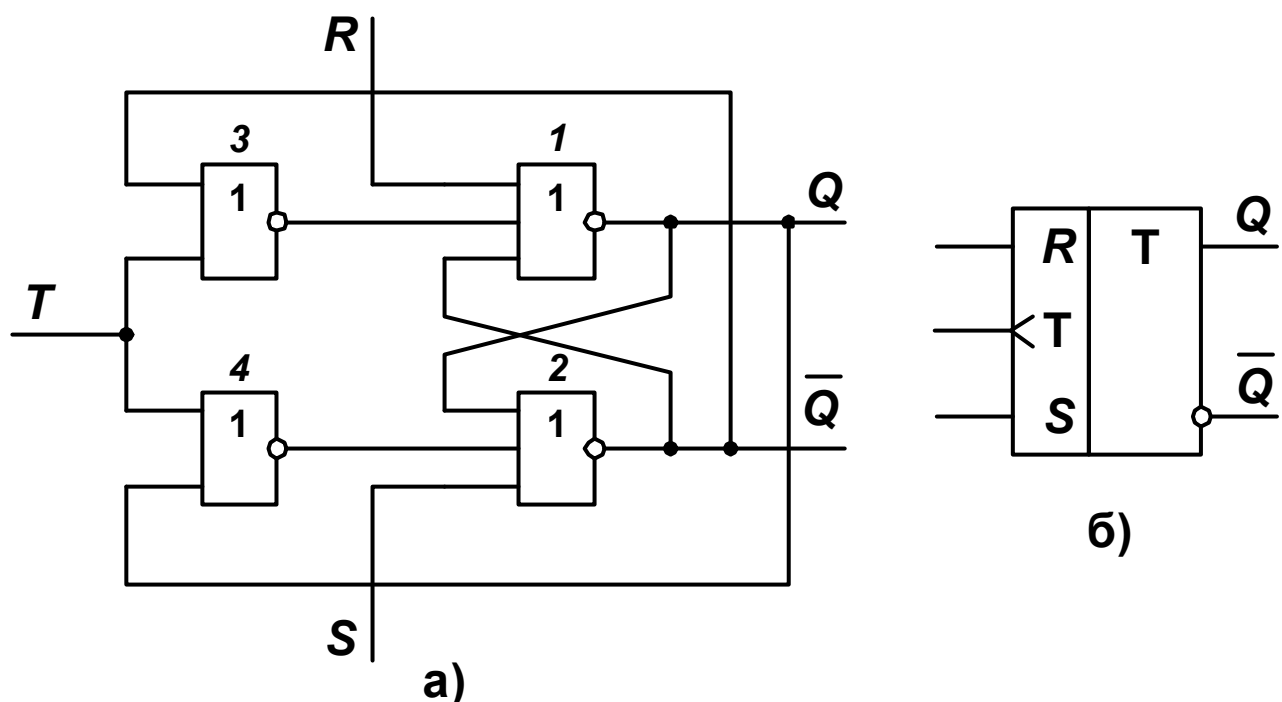


Рисунок 1.8 – Триггер с импульсным счетным входом:

а) – реализация в базисе ИЛИ-НЕ;

б) – условно-графическое обозначение

Элементы 1 и 2 образуют триггер с отдельными входами R и S , а счетный вход T организован с помощью элементов 3 и 4.

Допустим, что на выходах R и S зафиксирован нулевой уровень сигнала ($R=0$, $S=0$), а на вход T поступает серия (последовательность) прямоугольных электрических импульсов C (рисунок 1.9). Тогда при $C=1$ на выходах элементов 3 и 4 будут нули и триггер с отдельными входами, реализованный на элементах 1 и 2, согласно таблице истинности 1.3 будет хранить свое состояние, а при $C=0$ триггер будет переключаться в противоположное состояние, т.к. уровень на выходах элементов 3 и 4 будет в этом случае определяться, соответственно, значениями инверсного и прямого выхода триггера. При правильном подборе параметров синхросерии триггер будет переключаться в противоположное состояние по каждому синхроимпульсу, т.е. считать синхроимпульсы (счет по модулю 2).

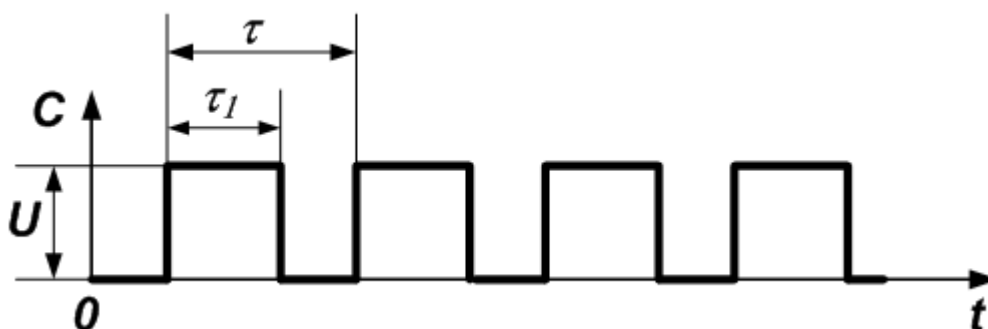


Рисунок 1.9 – Синхросерия с амплитудой импульсов U и периодом τ и скважностью τ_1/τ

Более подробно схемы и особенности работы триггеров со счетным входом (импульсным или потенциальным) описаны в [2, 3].

Синхронизированные триггеры.

В компьютерах наиболее часто синхронизируются моменты переключения триггеров. С этой целью триггеры снабжаются дополнительным входом, по которому поступает синхронизированный сигнал, и логическими элементами, обеспечивающими требуемый порядок переключения состояний. Триггеры с встроенной схемой синхронизации называются синхронизированными триггерами.

Синхронизированный триггер – это элементарная логическая схема, которая при синтезе более сложных схем рассматривается как элемент памяти.

Схема синхронизированного RS-триггера изображена на рисунке 1.10.

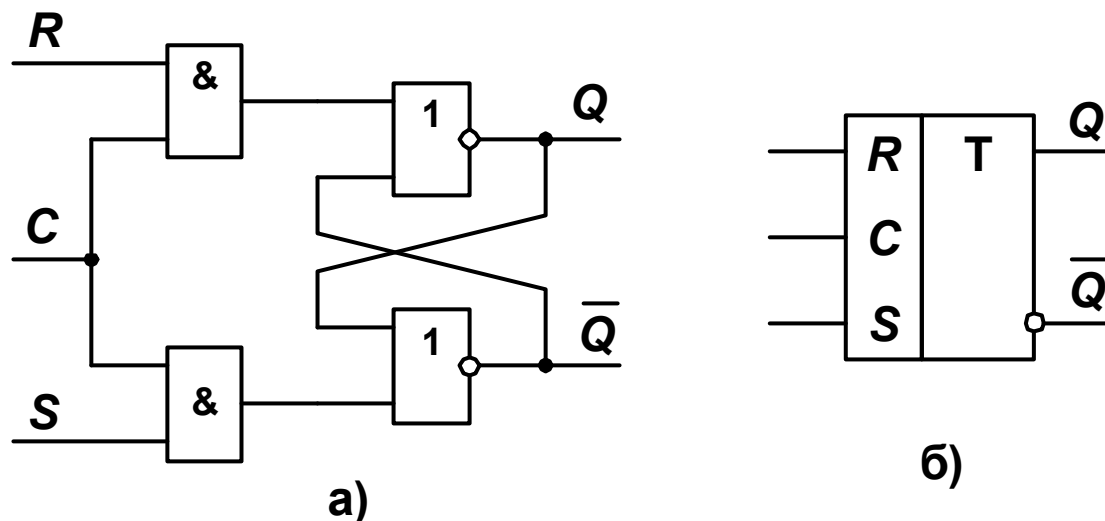


Рисунок 1.10 – Синхронизированный RS-триггер

1.3. Содержание и последовательность выполнения работы

Работа предусматривает следующую последовательность действий:

- 1) изучить раздел 1.2;
- 2) изучить описание лабораторной установки L-T, приведенное в разделе 10;
- 3) изучить описание микросхем, включенных в состав макета;
- 4) проверить логику работы элементов И, ИЛИ, НЕ, Исключающее ИЛИ в соответствии с их таблицами истинности с помощью микросхем серии 74НС**, представленных на макете L-T, сигналы на входы элементов подавать с кнопочного регистра;
- 5) провести макетирование и наладку схемы типового (i-го) разряда формирователя кодов, выполняющего для двоичных векторов А и В операции конъюнкции, дизъюнкции, инверсии А, инверсии В. Конкретная (j-ая) операция формирователя должна выбираться управляющим сигналом u_j . Управляющие сигналы на входы схемы подать с кнопочного регистра.
- 6) провести макетирование и наладку схемы асинхронного RS-триггера на базе логических элементов L-T;
- 7) провести макетирование и наладку схемы синхронного RS-триггера на базе логических элементов L-T.

1.4. Контрольные вопросы

- 1.4.1. Опишите логику работы многовходовых элементов И, ИЛИ, И-НЕ, ИЛИ-НЕ, исключающее ИЛИ.
- 1.4.2. Какая схема называется комбинационной?

1.4.3. Нарисуйте модель комбинационного преобразователя, исследуемого в лабораторной работе (изображение в виде «черного ящика» с указанием входных и выходных сигналов).

1.4.4. Как выбирается конкретная операция комбинационного преобразователя, исследуемого в лабораторной работе?

1.4.5. Объясните логику работы схемы асинхронного RS-триггера, построенную в базисе ИЛИ-НЕ.

1.4.6. Приведите таблицу истинности RS-триггера.

1.4.7. Как синхронизировать RS-триггер?

1.4.8. Что из себя представляет триггер со счетным входом и как он работает.

1.4.9. Что такое синхросерия? Какие параметры имеет синхросерия?

2.ЛАБОРАТОРНАЯ РАБОТА № 2. СИНТЕЗ КОМБИНАЦИОННЫХ И ТРИГГЕРНЫХ СХЕМ.

2.1. Цель работы

Работа имеет цель повторить методы проектирования комбинационных схем, изучить лабораторную установку «Logic-Trigger» (L-T), приобрести первые навыки сборки, наладки и исследования комбинационных и триггерных схем.

2.2. Содержание и последовательность выполнения работы

Работа выполняется без внеаудиторной подготовки и предусматривает следующую последовательность действий:

- 1) изучить описание лабораторной установки L-T, приведенное в разделе 10;
- 2) изучить описание микросхем, включенных в состав макета;
- 3) синтезировать функциональную и принципиальные электрические схемы, реализующие заданную функцию;
- 4) провести макетирование и наладку схемы с использованием L-T;
- 5) реализовать на макете функцию $Y = X1 \vee X2 \vee X3 \vee X4 \vee X5$ с использованием микросхем серии 74НС**;
- 6) построить триггер типа Т на JK- и D-триггерах (серия 74НС), подать на счетный вход сигналы с генератора синхросигналов, снять с помощью осциллографа временные диаграммы работы схем, обратить особое внимание на момент переключения триггера, сопоставив его с фронтами синхросигнала, по которым происходит переключение триггера.

2.3. Варианты заданий

Варианты заданий на лабораторную работу представлены на рисунке 2.1.

2.4. Контрольные вопросы

- 2.4.1. Какие формы задания булевых функций вы знаете?
- 2.4.2. Охарактеризуйте матричную форму задания булевой функции.
- 2.4.3. Как распределяются приоритеты функционирования триггеров с синхронными и асинхронными входами в микросхемах серии 74НС?
- 2.4.4. Представить обобщенную структурную схему операционного элемента.
- 2.4.5. В чем состоит различие между управляющими и осведомительными сигналами?

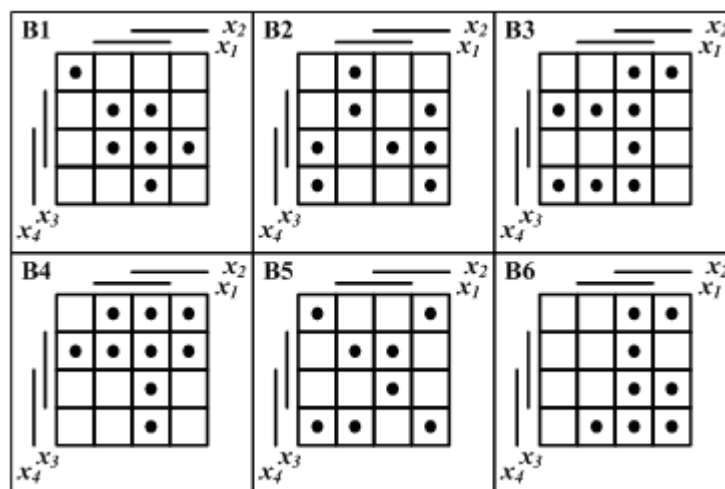


Рисунок 2.1 – Варианты заданий на лабораторную работу № 1

3. ЛАБОРАТОРНАЯ РАБОТА №3. ВЫПОЛНЕНИЕ МИКРООПЕРАЦИЙ В РЕГИСТРАХ

3.1. Цель работы

Целью работы является изучение принципов структурной и функциональной организации регистров, методов выполнения микроопераций установки кода, приема и передачи кода, сдвига и преобразования кода в регистрах, а также формирования осведомительных сигналов.

3.2. Краткие теоретические сведения

Регистром называется упорядоченная последовательность запоминающих элементов, предназначенных для хранения слова информации. В качестве запоминающих элементов используются триггеры. Как составная часть ОА, регистр может выполнять различные микрооперации.

Пример. Спроектировать 4-разрядный регистр $Q(1:4)$, выполняющий следующие микрооперации:

y_0 : $Q(1:4) := 0000$ – обнуление (сброс) регистра,

y_1 : $Q(1:4) := 1010$ – установка кода,

y_2 : $Q(1:4) := T(1:4)$ – прием кода с регистра T (межрегистровая передача),

y_3 : $Q(1:4) := Q(2:4).Q_1$ – циклический сдвиг влево информации в регистре.

Регистр должен вычислять осведомительные сигналы

$x_1 = (Q(1:4) = 1111)$, $x_2 = (Q(1:4) \geq 1010)$.

В качестве элементного базиса для реализации i -го разряда регистра можно взять, например, D-триггер с установочными RS-входами (серия 74НС) – рисунок 3.1. У данного триггера есть входы $D, C, \bar{R}, \bar{S}$, а также

комплиментарные выходы Q и $\bar{Q}$. Входы $\bar{S}$ и $\bar{R}$ асинхронные, потому что они работают независимо от сигнала на тактовом входе C ; активный уровень для них – низкий. Сигнал от входа D передается на выходы Q и $\bar{Q}$ по положительному (от низкого к высокому уровню) перепаду на тактовом входе C . Чтобы триггер переключился правильно, нужный уровень на входе D следует зафиксировать заранее, перед приходом тактового перепада. Если на входы $\bar{S}$ и $\bar{R}$ одновременно подать напряжения низкого уровня, состояние выходов Q и $\bar{Q}$ окажется неопределенным. Асинхронная установка нужного сочетания уровней на выходах получится, когда на входы $\bar{S}$ и $\bar{R}$ поданы взаимно-противоположные логические сигналы. В это время входы C и D отключены. Загрузить в триггер бит данных по входу D можно, если на входы $\bar{S}$ и $\bar{R}$ подано напряжение высокого уровня.

В таблицу 3.1 сведены все правила переходов D -триггера. Для реализации регистра необходимо четыре таких триггера. Обозначим входы триггеров 1 – 4 соответственно R1-R4, S1-S4, D1-D4, C1-C4, выходы – Q1-Q4.

Выходы внешнего регистра T(1:4), с которого принимается информация на регистр Q, обозначим T1-T4.

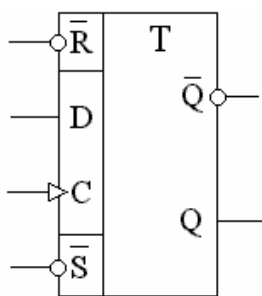


Рисунок 3.1 — Условно-графическое обозначение D-триггера

Необходимо синтезировать функции возбуждения входов регистра для осуществления набора заданных микроопераций. Построим таблицу зависимости функций возбуждения входов от управляющих сигналов и входной информации (таблица 3.2). Операции установки и приема кода целесообразно выполнять при помощи специально для

этого предназначенных установочных RS-входов регистра. Эти входы асинхронные и имеют приоритет перед другими входами (что отражено в таблице 3.1). При этом различают двухтактные и одноктактные операции установки и приема. Для двухтактных операций характерно, что в первом такте в схему (на R входы, ответственные за установку нуля) подается управляющий сигнал y_0 , вследствие чего регистр обнуляется. Во втором такте в схему подается непосредственно сигнал установки (или приема) – y_1 (или y_2). Этот сигнал подается только на S входы, ответственные за установку единицы (так как нули уже установлены в предыдущем такте).

При одноктактной установке и нули и единицы входного слова должны быть приняты на регистр за один такт под воздействием единственного сигнала y_1 (или y_2). В таблице 3.2 элементы, необходимые для одноктактной установки (приема) и лишние для двухтактной, заключены в квадратные скобки. Пустые клетки в таблице 3.2 означают, что соответствующий сигнал имеет низкий уровень (равен логическому нулю).

Таблица 3.1

$C(t)$	$D(t)$	$\bar{R}(t)$	$\bar{S}(t)$	$Q(t+1)$
X	X	0	0	Не определено
X	X	0	1	0
X	X	1	0	1
0	X	1	1	$Q(t)$
1	0	1	1	0
1	1	1	1	1

Анализ таблицы 3.2 позволяет сделать вывод, что функции для R-входов при однократной установке усложняются, что является своеобразной платой за выигрыш в быстродействии.

Примечание. У микросхемы D-триггера для исключения запрещенной комбинации 11 на RS-входах, эти входы выполнены инверсными. При составлении таблицы 3.2 инверсии на установочных входах триггера не учитывались. Эти инверсии будут учтены при составлении булевых функций возбуждения входов по таблице 3.2 (так как инвертироваться должна вся функция, а не отдельные входящие в нее переменные).

Микрооперацию сдвига целесообразно выполнять с использованием синхронного D-триггера. При этом синхроимпульсы будут использоваться для отсчета сдвигов: на один синхроимпульс будет выполняться один сдвиг. Для этого управляющий сигнал y_3 , домноженный на синхроимпульс, целесообразно подать на входы С триггеров.

Внимание! В схему регистра не может быть подано два или более управляющих сигнала одновременно, так как в конкретный момент времени на одном операционном элементе может выполняться только одна микрооперация.

Таблица 3.2

y	R1	S1	R2	S2	R3	S3	R4	S4
y_0	1		1		1		1	
y_1		1	[1]			1	[1]	
y_2	$[\bar{T}1]$	$T1$	$[\bar{T}2]$	$T2$	$[\bar{T}3]$	$T3$	$[\bar{T}4]$	$T4$
y	D1	D2	D3	D4	C1	C2	C3	C4
y_3	Q2	Q3	Q4	Q1	с	с	с	с

Из таблицы 3.2 получаем следующие функции возбуждения входов триггеров.

$$\begin{aligned}
\overline{R1} &= \overline{y0 \vee [y2 \overline{T1}]}; & \overline{S1} &= \overline{y1 \vee y2 \overline{T1}}; \\
\overline{R2} &= \overline{y0 \vee [y1 \vee y2 \overline{T2}]}; & \overline{S2} &= \overline{y2 \overline{T2}}; \\
\overline{R3} &= \overline{y0 \vee [y2 \overline{T3}]}; & \overline{S3} &= \overline{y1 \vee y2 \overline{T3}}; \\
\overline{R4} &= \overline{y0 \vee [y1 \vee y2 \overline{T4}]}; & \overline{S4} &= \overline{y2 \overline{T4}}; \\
D1 &= Q2; \quad D2 = Q3; \quad D3 = Q4; \quad D4 = Q1; \\
C1 &= C2 = C3 = C4 = c \vee y3.
\end{aligned}$$

Осведомительный сигнал $x1$ вычисляется по формуле

$$x1 = Q1 Q2 Q3 Q4.$$

На рисунке 3.1 показаны коды состояний регистра, на которых осведомительный сигнал $x2$ должен принимать единичные значения. Минимальная ДНФ функции $x2$:

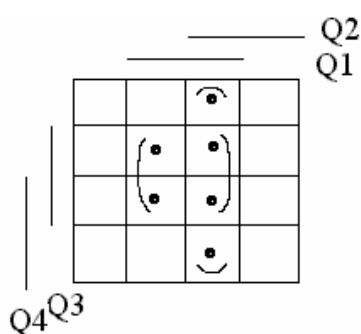


Рисунок 3.1 — Наборы, на которых осведомительный сигнал $x2$ принимает единичное значение

$$x2(Q1, Q2, Q3, Q4) = Q1 Q2 \vee Q1 Q3.$$

Для реализации функции в базисе «И-НЕ» ее следует представить в виде:

$$x2 = \overline{Q1 Q2 \cdot Q1 Q3} \quad (\text{доказать справедливость преобразования функции самостоятельно}).$$

Схема регистра (вариант двухтактной установки (приема) кода) представлена на рисунке 3.2. В дальнейшем под словом схема будем понимать функциональную схему

устройства. Если будут подразумеваться другие виды схем, это будет оговорено.

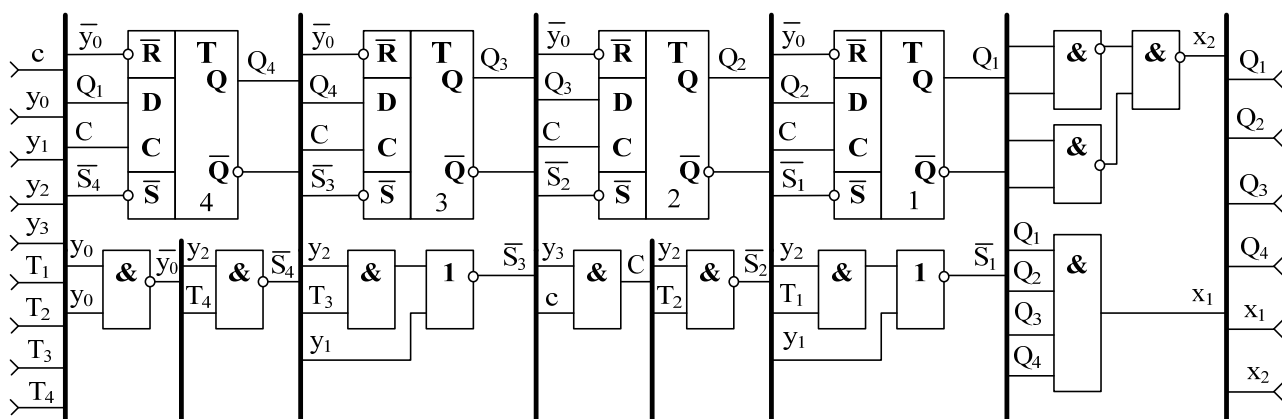


Рисунок 3.2 — Схема регистра, выполняющего микрооперации сброса, установки и приема кода, сдвига

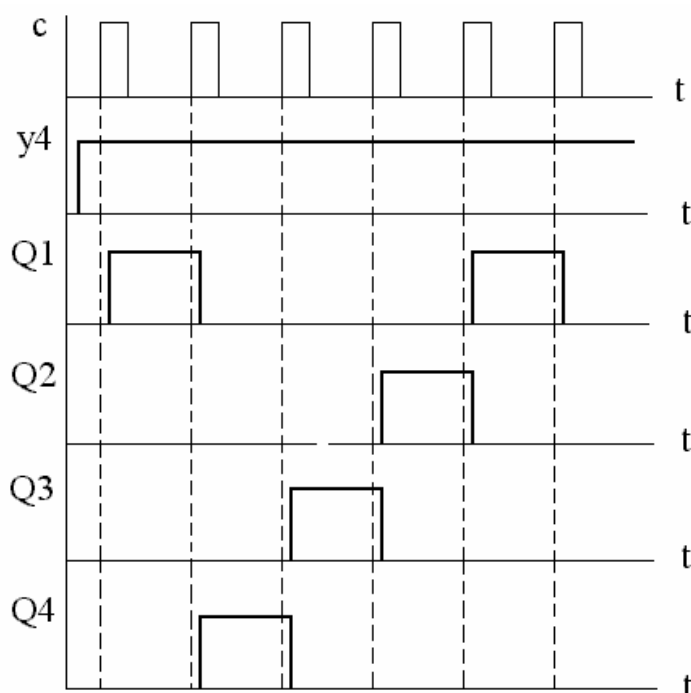


Рисунок 3.3 — Временные диаграммы регистра при выполнении микрооперации сдвига в динамическом режиме

Временные диаграммы регистра при выполнении им микрооперации сдвига в динамическом режиме (при подаче на С-входы последовательности синхроимпульсов) приведены на рисунке 3.3. D-триггеры (триггеры-защелки) переключаются по нарастающему фронту синхросигнала. Временные интервалы, в частности, задержки переключения триггеров относительно начала синхроимпульса, на рисунке 3.3 показаны условно без соблюдения масштаба. Параметры синхросерии (частота, скважность) обычно рассчитываются исходя из принципов взаимодействия УА и ОА и задержек срабатывания операционных элементов. В данном случае, например, важно, чтобы за время положительного уровня на входе С, триггер успел сработать.

Микрооперации установки и приема кода в данной схемной реализации осуществляются асинхронно при подаче в схему единичного значения соответствующего управляющего сигнала (можно предположить, что время формирования этих значений задается (синхронизируется) управляющим автоматом).

3.3. Содержание работы

Задание предусматривает проектирование трехразрядного регистра, выполняющего микрооперации установки кода, приема кода с регистра кнопок, операцию сдвига, а также синтез схем формирования осведомительных сигналов, отражающих состояние регистра. Разработанная схема должна быть собрана и отлажена на макете L-T.

В результате исполнения операции установки кода регистр переходит в заранее известное состояние. Микрооперация приема кода обеспечивает прием кода с трехразрядного регистра кнопок. Обе эти микрооперации могут выполняться за один или два такта. В схемах, проектируемых с использованием двухтактной установки или приема кода, по приходу сигнала первого такта регистр сбрасывается в нулевое состояние, а затем по сигналу второго такта осуществляется переключение только тех триггеров регистра, которые должны быть установлены в единичное состояние. Количество тактов микроопераций (МО) установки и приема кода и сам код определяется индивидуальным заданием (таблица 3.3). Сигналы микроопераций приема и установки кода (в том числе и сигнал первого такта для двухтактных схем) рекомендуется подавать с помощью кнопок. Соответствующие переключения триггеров целесообразно выполнять, используя асинхронные входы R и S триггеров.

Микрооперацию сдвига наиболее удобно реализовать по сигналу, снимаемому с генератора синхроимпульсов с использованием синхронных входов триггеров. Заданием определяется тип триггера (JK или D), тип сдвига (логический с доопределением освобождающегося разряда – L, циклический – C), направление сдвига (влево – L, вправо – R). Константа сдвига равна 1. Освобождающийся при логическом сдвиге разряд доопределяется нулем.

Осведомительные сигналы задаются в терминах отношений. Например: $X_1 = (A=B_1)$, $X_2 = (A \geq B_2)$, $X_3 = (A < B_3)$, где B_1, B_2, B_3 константы. Естественно, коды A и B_i интерпретируются как двоичные числа. Для формирования осведомительных сигналов реализуются соответствующие комбинационные схемы. Перечень осведомительных сигналов определяется индивидуальным заданием.

3.4. Варианты индивидуальных заданий

Варианты индивидуальных заданий на лабораторную работу приведены в таблице 3.3.

Таблица 3.3 – Варианты заданий на лабораторную работу № 2

№ вар.	Тип триггера	Устанавливаемые коды	МО приема кода	Количество тактов приема, установки кода	Тип сдвига	Направление сдвига	Осведомительный сигнал
1	D	000 001	$A(1:3) := T(1:3)$	2	L	L	$A(1:3) \leq 110$
2	D	000 101	$A(1:3) := \neg T(1:3)$	2	C	L	$A(1:3) = 101$
3	D	000 011	$A(1:3) := T(1:3)$	2	L	R	$A(1:3) = 011$
4	D	000 110	$A(1:3) := \neg T(1:3)$	2	C	R	$A(1:3) \geq 110$
5	D	000 100	$A(1:3) := T(1:3)$	2	C	L	$A(1:3) \geq 101$

3.5. Внеаудиторная подготовка

3.5.1. Изучить особенности функциональной и структурной организации регистров: [2] стр. 119-127, [2] стр. 197-199.

3.5.2. Разработать схему регистра в соответствии с индивидуальным заданием.

3.5.3. Продумать последовательность наладки схемы. Рекомендуется с целью упрощения наладки сначала собрать часть схемы, реализующую одну или две микрооперации. После наладки последовательно макетировать дополнительные цепи, реализующие оставшиеся микрооперации. В процессе подготовки следует спланировать тестовые входные комбинации и ожидаемую реакцию на них, которые будут использованы при проверке работоспособности схемы на каждом этапе наладки.

3.6. Выполнение работы в лаборатории.

3.6.1. В соответствии с подготовленным планом собрать и наладить схему.

3.6.2. Установить в регистре унитарный код 100 или 011 и с помощью осциллографа снять временные диаграммы циклического сдвига. На временной диаграмме отразить состояние всех трех триггеров регистра.

3.7. Контрольные вопросы

3.7.1. Какие микрооперации выполняются в регистре?

3.7.2. Для чего в сдвиговых регистрах используются двухступенчатые триггеры?

3.7.3. Какие преимущества и недостатки двухтактных (однофазных) схем приемки и установки кодов?

3.7.4. Как изменится разработанная схема регистра, если использовать одноктактные схемы приема и установки кодов.

3.7.5. Чем определяется время исполнения микроопераций в регистрах?

3.7.6. Сформулируйте алгоритм получения функций возбуждения триггеров регистра, выполняющего заданный набор микроопераций.

4. ЛАБОРАТОРНАЯ РАБОТА №4. КОМБИНАЦИОННЫЕ ДВОИЧНЫЕ СЧЕТЧИКИ

4.1. Цель работы

Изучить схемные реализации и методы проектирования комбинационных схем счета, получить навыки сборки и наладки таких схем.

4.2. Краткие теоретические сведения

Различают комбинационные и накапливающие счетчики. Комбинационный счетчик – это комбинационная схема, формирующая на своих выходах код (целое двоичное число), который больше (меньше) входного кода на целое число m . Как правило, $m = 2^k$, где $k = 1, 2, 3 \dots$. Накапливающий счетчик – это последовательностная схема (схема с памятью), которая в каждом такте счета увеличивает (уменьшает) код своего состояния на целое число, не обязательно являющееся степенью двойки.

В данной работе рассматриваются комбинационные счетчики, выполняющие следующие микрооперации:

$$y_1: B(1:n) := A(1:n) + 1, \quad (4.1)$$

$$y_2: B(1:n) := A(1:n) - 1, \quad (4.2)$$

где $A(1:n)$ - n -разрядное двоичное число на входе схемы, а $B(1:n)$ результат выполнения микрооперации счета, представленной на выходе схемы. Счетчики, выполняющие микрооперацию y_1 (формула 4.1), называются суммирующими или инкрементирующими, микрооперацию y_2 (формула 4.2) – вычитающими или декрементирующими. Счетчики, выполняющие обе эти операции, называются реверсивными.

Синтезируем схему типового i -го разряда счетчика (рисунок 4.1,а) и наберем из этих схем требуемую разрядность счетчика, как это показано на рисунке 4.1,б. Подобный подход - один из основных подходов при синтезе операционных элементов.

На входы i -го разряда счетчика (рисунок 4.1,а) поступают сигналы a_i – i -ый разряд входного кода A и p_i – перенос (в вычитающем счетчике – заем) в этот разряд из предыдущего младшего разряда счетчика. Счетчик суммирует два указанных сигнала и выдает на выход два сигнала – b_i (i -ый разряд выходного кода B) - сумма входных сигналов, p_{i-1} – перенос (заем) в следующий старший разряд. Перенос в самый младший разряд n -разрядного счетчика (рисунок 4.1,б) p_n равен 1, т.к. счетчик выполняет функцию суммирования (декрементирующий счетчик – вычитания) единицы из младшего разряда. Перенос (заем) из старшего разряда p_0 может быть использован для наращивания разрядности счетчика.

Зависимость выходов одnorазрядного суммирующего счетчика от входов приведена в таблице 4.1. Подобная информация для вычитающего счетчика приведена в таблице 4.2.

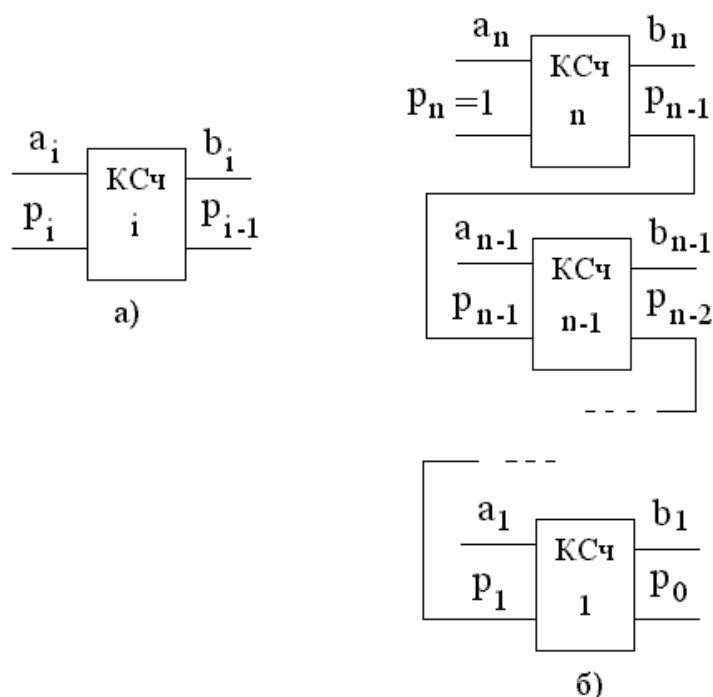


Рисунок 4.1 — Комбинаторные двоичные счетчики:
а) — одноразрядный, б) — n-разрядный

Таблица 4.1

a_i	p_i	b_i	p_{i-1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Таблица 4.2

a_i	p_i	b_i	p_{i-1}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Выходные сигналы для одnorазрядного суммирующего счетчика вычисляются по формулам:

$$b_i = a_i \overline{p_i} \vee \overline{a_i} p_i; \quad p_{i-1} = a_i p_i. \quad (4.3)$$

Аналогичные формулы для вычитающего счетчика:

$$b_i = a_i \overline{p_i} \vee \overline{a_i} p_i; \quad p_{i-1} = \overline{a_i} p_i. \quad (4.4)$$

Анализ формул (4.3 и 4.4) позволяет сделать заключение о том, что сигнал b_i вычисляется одинаково как в суммирующем, так и в вычитающем счетчике (сложение по модулю два сигналов a_i и p_i).

Суммирующий счетчик отличается от вычитающего только тем, что в нем значение выходного сигнала p_{i-1} зависит от прямого значения входного

сигнала a_i , а в вычитающем – от инверсного.

Перенос в самый младший разряд как вычитающего, так и суммирующего счетчика должен быть тождественно равен 1 ($p_n \equiv 1$), чтобы выполнялась микрооперация счета.

С учетом этого

$$b_n = a_n \cdot 0 \vee \overline{a_n} \cdot 1 = \overline{a_n},$$

$$p_{n-1} = a_n \cdot 1 = a_n \text{ для суммирующего счетчика и}$$

$$p_{n-1} = \overline{a_n} \cdot 1 = \overline{a_n} \text{ для вычитающего счетчика.}$$

Следовательно, самый младший разряд комбинационного счетчика может быть реализован с помощью инвертора.

Важно понимать, что в суммирующем счетчике перенос в следующий старший разряд определяется прямыми значениями a_i всех предыдущих младших разрядов, а в вычитающем - инверсными. В следующих примерах как при сложении так и при вычитании переносы во все разряды счетчика равны 1.

$$\begin{array}{r} \overline{1000} \\ \underline{1} \\ 0111 \end{array} \quad \begin{array}{r} \overline{0111} \\ \underline{1} \\ 1000 \end{array}$$

Перенос в вычитающем счетчике называют заемом, что не принципиально при построении электрической схемы.

Различают комбинационные счетчики с последовательным (сквозным) и параллельным переносом. В счетчиках с последовательным переносом для вычисления сигнала p_{i-1} используется формула $p_{i-1} = a_i p_i$ для суммирующего счетчика, $p_{i-1} = \overline{a_i} p_i$ для вычитающего счетчика.

В счетчиках с параллельным переносом для вычисления сигнала p_{i-1} используется формула $p_{i-1} = a_n a_{n-1} a_{n-2} \dots a_i$ для суммирующего счетчика и $p_{i-1} = \overline{a_n} \overline{a_{n-1}} \overline{a_{n-2}} \dots \overline{a_i}$ для вычитающего счетчика. Схема с последовательным переносом имеет меньшую сложность (2 входа элемента «И» для реализации переноса в каждый разряд, начиная с (n-2)-го). Недостаток схемы – сравнительно невысокое быстродействие (значение выходного кода станет действительным только после того как входной сигнал пройдет через все логические элементы, формирующие переносы). Платой за высокое быстродействие схемы с параллельным переносом (задержка формирования переноса в любой разряд, начиная с (n-2)-го, равна задержке одного элемента, формирующего перенос) является увеличение сложности (число входов элемента «И», формирующего перенос возрастает с увеличением номера разряда счетчика).

Компромиссное решение между количеством используемого оборудования и быстродействием счетчика может быть получено при использовании принципа групповой организации переноса между разрядами счетчика. В этом случае n -разрядный счетчик разбивается на последовательность групп, по m соседних разрядов в каждой группе. В пределах разрядов одной группы переносы

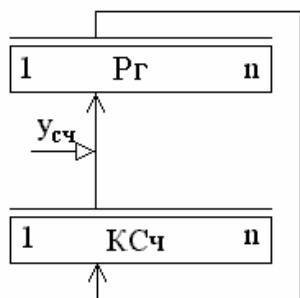


Рисунок 4.2 — Схема накапливающего счетчика, реализованного на основе комбинационного счетчика и регистра

вырабатываются параллельно для всех разрядов групп, а между группами переносы распространяются последовательно. Более подробно схемотехника комбинационных счетчиков изложена в [1], стр. 131-157.

Если в качестве входного кода комбинационного счетчика использовать код, хранимый в регистре, а выходной код комбинационного счетчика загружать в этот регистр по соответствующему

4.3. Содержание работы

Задание на лабораторную работу предусматривает синтез, макетирование и наладку комбинационного счетчика с использованием макета L-T. Варианты задания отличаются типом выполняемых микроопераций счета и способом распространения переноса. После сборки и наладки комбинационного счетчика необходимо соединить его с регистром для построения накапливающего счетчика и снять временную диаграмму.

4.4. Варианты индивидуальных заданий

Варианты заданий представлены в таблице 4.3.

Таблица 4.3

№ вар	Тип счетчика	Способ распространения переноса
1	Декрементирующий	последовательный
2	Инкрементирующий	последовательный
3	Декрементирующий	параллельный
4	Инкрементирующий	параллельный
5	Декрементирующий	последовательный

4.5. Внеаудиторная подготовка

4.5.1. Изучить принципы построения комбинационных счетчиков:

[3], стр. 247-250.

4.5.2. Синтезировать схему комбинационного счетчика в соответствии с индивидуальным заданием.

4.5.3. Построить схему накапливающего счетчика с использованием комбинационного счетчика и регистра на отдельных триггерах.

4.6. Выполнение работы в лаборатории

4.6.1. Собрать и отладить схему трехразрядного комбинационного счетчика. В качестве сигналов счета рекомендуется использовать сигналы с кнопочных регистров, входной код $A(1:3)$ также набирать на кнопочном регистре. Младший разряд (b_3) можно реализовать с помощью элемента И-НЕ (за счет подстановки в формулу для b_3 значения переноса $p_3=1$). Для реализации разрядов b_2 и b_1 удобно использовать элементы «Исключающее ИЛИ» ($=1$). Правильность работы схемы проверить в статическом режиме.

4.6.2. Построить схему накапливающего счетчика, присоединив к комбинационному счетчику регистр. Занесение информации на регистр рекомендуется синхронизировать импульсами генератора, сигнал установки в начальное состояние (000 – для инкрементирующего, 111 для декрементирующего счетчика) подавать с кнопки, управляющей работой генератора одиночных импульсов. Убедиться в правильности работы счетчика в однократном режиме. Снять временную диаграмму в непрерывном режиме.

4.7. Контрольные вопросы

4.7.1. Где в компьютере используются комбинационные счетчики?

4.7.2. Построить структурную схему счетчика с групповым переносом.

4.7.3. Какова минимальная задержка сигнала переноса в одноразрядном счетчике?

4.7.4. Сравните быстродействие счетчиков с параллельным и последовательным переносами, считая, что задержка сигнала на любом вентиле равна t .

4.7.5. Как на комбинационном счетчике наряду с микрооперацией $B:=A+1$ реализовать микрооперации $B:=A+2$?

5. ЛАБОРАТОРНАЯ РАБОТА № 5. НАКАПЛИВАЮЩИЕ ДВОИЧНЫЕ СЧЕТЧИКИ

5.1. Цель работы

Изучить принципы построения, получить навыки проектирования и наладки счетчиков, ознакомиться с методами использования счетчиков в интегральном исполнении.

5.2. Краткие теоретические сведения

Накапливающий счетчик, выполняющий микрооперацию $y_1 : A(1:4) := A(1:4) + 1$ или $y_2 : A(1:4) := A(1:4) - 1$ называется соответственно инкрементирующим или декрементирующим. Если счетчик выполняет обе микрооперации, то его называют реверсивным. Анализ реализуемых функций позволяет сделать вывод о том, что накапливающий счетчик – это схема, содержащая элементы памяти.

В качестве примера на рисунке 5.1 приведена функциональная схема четырехразрядного суммирующего накапливающего счетчика с последовательным переносом. В накапливающих счетчиках последовательный перенос отличается от сквозного. Типовой разряд счетчика (рисунок 5.1) реализуется Т-триггером. Таблица истинности Т-триггера (триггер со счетным входом) – таблица 5.1 (Т – тактовый вход триггера, Е – вход разрешения переключения триггера). Для обеспечения счетного режима работы триггера (т.е. режима переключения в противоположное состояние по каждому импульсу, поступающему на тактовый вход) в соответствии с таблицей 5.1 на входы Е всех триггеров подана логическая 1. Сигналы межразрядных переносов поданы на Т-входы соответствующих триггеров.

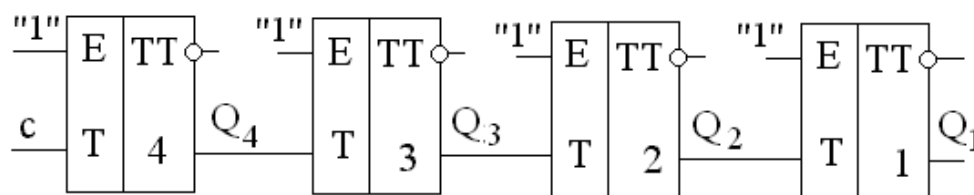


Рисунок 5.1 — Четырехразрядный накапливающий
инкрементирующий счетчик с последовательным
переносом

Таблица 5.1

E(t)	T(t)	$Q(t+1)$
0	с	$\underline{Q(t)}$
1	с	$\overline{Q(t)}$

Вычитающий счетчик будет отличаться от суммирующего только тем, что перенос в следующий старший разряд будет определяться инверсным значением предыдущих разрядов (то есть в данном

случае сигнал переноса будет сниматься не с прямого, а с инверсного выхода соответствующего триггера).

Результат микрооперации счета является действительным через промежуток времени $t_{\text{счета}} = 4t_{\text{тр}}$ с момента подачи в схему синхросигнала, где $t_{\text{тр}}$ – время переключения триггера. Это объясняется тем, что триггеры срабатывают последовательно. Отсюда следует, что период тактирующей серии импульсов (такт операционного автомата) для стабильной работы счетчика должен превышать указанное время.

Для увеличения быстродействия счетчика применяются схемы со сквозным и параллельным переносом. В этих схемах синхросигнал подается параллельно на Т-входы всех триггеров, т.е. все триггеры переключаются одновременно. К моменту переключения триггеров на логических элементах формируются межразрядные переносы по формуле:

$P_{i-1} = Q_i P_i$, $i = n, n-1, n-2, \dots, 1$ (для n -разрядного счетчика со сквозным переносом),

$P_{i-1} = Q_n \cdot Q_{n-1} \cdot Q_{n-2} \cdot Q_i$ (для n -разрядного счетчика с параллельным переносом), где P_i – перенос в i -ый разряд триггера, Q_i – прямой выход i -го триггера.

Приведенные выше формулы справедливы для суммирующих счетчиков. В соответствующие формулы для вычитающих счетчиков сигналы Q_i войдут с инверсией (будут сниматься с инверсных входов триггеров). Функциональная схема четырехразрядного вычитающего счетчика со сквозным переносом приведена на рисунке 5.2. В этой схеме $t_{\text{счета}} = t_{\text{тр}} + 2t_{\text{кс}}$, где $t_{\text{кс}}$ – время задержки комбинационной схемы, формирующей сигнал переноса. Для счетчика с параллельным переносом $t_{\text{счета}} = t_{\text{тр}} + t_{\text{кс}}$.

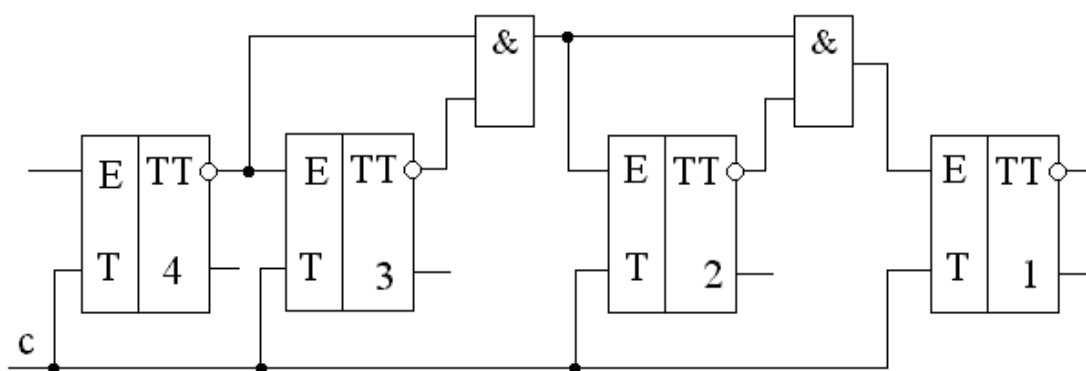


Рисунок 5.2 — Четырехразрядный накапливающий декрементирующий счетчик со сквозным переносом

Обратите внимание, что в схеме используются двухступенчатые триггеры. Первая ступень двухступенчатого триггера синхронизируется, как правило, прямым значением синхросигнала и, следовательно, срабатывает по нарастающему фронту синхросигнала (будем называть его просто фронтом), вторая ступень синхронизируется инверсным значением синхросигнала и

соответственно срабатывает по ниспадающему фронту синхроимпульса (будем называть его срезом). Следовательно, переключение JK-триггера осуществляется по ниспадающему фронту (срезу) синхроимпульса (это необходимо учитывать при построении временных диаграмм счета).

Преимущество двухступенчатых элементов памяти состоит в том, что момент приема информации на триггер с его входов (срабатывание первой ступени) и момент выдачи информации на выходы триггера (срабатывание второй ступени) разделены во времени (на время длительности синхроимпульса минус время срабатывания первой ступени триггера). Это позволяет схемам, в которых используется двухступенчатая память, работать устойчиво, когда один и тот же запоминающий элемент (триггер) в одном и том же такте является и источником и приемником информации. Примером может служить рассматриваемая схема счетчика (рисунок 5.2). По фронту синхроимпульса в схеме счетчика переключаются первые ступени всех триггеров в соответствии со сформированными на логических элементах (вентиллях) сигналами межразрядных переносов (длительность синхроимпульса должна быть больше, чем время переключения первой ступени триггеров). Переключение осуществится корректно, т. к. во время длительности синхроимпульса информация на выходах триггеров (выходах второй ступени), используемых для вычисления сигналов переноса, не изменяется. По срезу синхроимпульса информация с первой ступени триггеров переписывается на вторую ступень. Следующее срабатывание схемы произойдет по следующему синхроимпульсу.

Постройте временные диаграммы работы схем, изображенных на рисунках 5.1 и 5.2, и убедитесь, что эти схемы действительно выполняют микрооперацию счета.

Если речь идет о синтезе счетчиков, считающих в заданной системе счисления (по заданному модулю), то необходимо синтезировать счетчик, считающий по модулю q , где q – основание системы счисления. Например, нужно синтезировать двоичный счетчик, считающий по модулю 10 ($q=10$). В этом случае разрабатывается m -разрядный двоичный счетчик $m = \lceil \log_2 q \rceil$, обеспечивающий переключение из состояния $(q-1)$ в состояние 0 для инкрементирующего режима и из состояния 0 в состояние $(q-1)$ для декрементирующего режима.

Пусть требуется синтезировать инкрементирующий накапливающий двоичный счетчик по модулю десять ($q=10$) с параллельным переносом. Для реализации такого счетчика необходимо четыре триггерных элемента ($m=4$). I-ый разряд накапливающего счетчика можно реализовать на JK-триггере серии 74НС (вспомните одно из заданий первой лабораторной работы: синтезировать Т-триггер на JK-триггере).

Пусть кроме микрооперации счета операционный элемент будет выполнять микрооперации сброса y_0 : $C(1:4) := 0000$ и установки y_1 : $C(1:4) := 0101$ и выдавать осведомительный сигнал $x = (C(1:4) := 0000)$.

Таблица истинности JK-триггера с установочными RS-входами, изображенного на рисунке 5.3, – таблица 5.2.

На основе анализа таблиц 5.1 и 5.2 можно обосновать организацию Е-

входа путем объединения J и K входов. При таком объединении на J и K вход триггера будет поступать один и тот же сигнал. Если этот сигнал равен нулю, триггер хранит свое состояние, в противном случае триггер переключается в противоположное состояние по срезу синхроимпульса, поступившего на вход C (используется в качестве тактового входа T), согласно таблице 5.2.

Таблица 5.2

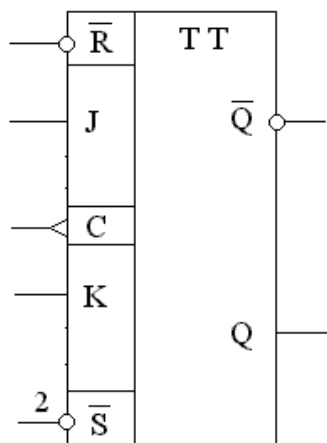


Рисунок 5.3 Условно-графическое изображение JK-триггера

$C(t)$	$J(t)$	$K(t)$	$\bar{R}(t)$	$\bar{S}(t)$	$Q(t+1)$
X	X	X	0	0	Не определено
X	X	X	0	1	0
X	X	X	1	0	1
0	X	X	1	1	$Q(t)$
1	0	0	1	1	$Q(t)$
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	1	1	$\bar{Q}(t)$

Пусть микрооперация счета инициируется управляющим сигналом y_2 . Тогда для организации параллельного переноса на C-входы всех триггеров необходимо подать сигнал $c \cdot y_2$, а JK-входы использовать для подачи межразрядного переноса.

Обычный суммирующий четырехразрядный двоичный счетчик автоматически устанавливает комбинацию 0000 после комбинации 1111 (15 в двоичной системе). Это счетчик по модулю 16.

Суммирующий двоичный счетчик по модулю 10 (основание системы счисления) должен считать от 0000 до 1001 (9 в двоичной системе счисления), а затем сбрасываться в 0000. Сброс счетчика можно осуществлять по установочным RS-входам при помощи сигнала $x_{\text{mod}} = (C(1:4)=1010)$. При этом комбинация 1010 будет появляться на выходах регистра временно ($t_{\text{тр}} + t_{\text{кз}}$), что не повлияет на логику работы схемы, хотя может сказаться на надежности. Другими словами, при переходе счетчика из состояния 1001 в состояние 0000 через состояние 1010, присутствующее на выходах счетчика в течение времени выработки осведомительного сигнала и времени переключения триггеров, на выходе Q_3 возникает кратковременный остроконечный импульс, вызывающий переходный процесс.

Построим таблицу зависимости функций возбуждения RS-входов от управляющих сигналов и сигнала x_{mod} (таблица 5.3).

Таблица 5.3

y	R1	S1	R2	S2	R3	S3	R4	S4
y0	1		1		1		1	
y1	1			1	1			1
x _{mod}	1		1		1		1	

На основе данной таблицы строятся функции возбуждения входов.

$$\begin{aligned} \overline{R1} = \overline{R3} &= \overline{y0 \vee y1 \vee x_{\text{mod}}}; & \overline{S1} = \overline{S3} &= 1; \\ \overline{R2} = \overline{R4} &= \overline{y0 \vee x_{\text{mod}}}; & \overline{S2} = \overline{S4} &= \overline{y1} \end{aligned}$$

Осведомительный сигнал $x=(C(1:4):=0000)$ может быть сформирован в соответствии с функцией

$$x = \overline{Q_1} \cdot \overline{Q_2} \cdot \overline{Q_3} \cdot \overline{Q_4}$$

Схема счетчика приведена на рисунке 5.4. Постройте временные диаграммы работы счетчика, особое внимание уделите моменту переключения в из состояния 1001 в состояние 0000 через состояние 1010 (кратковременно присутствующее на выходах счетчика).

Примечание – Вычитающий счетчик по модулю 10 считает от 1001 до 0000, а затем устанавливается в 1001. Установку счетчика в состояние 1001 можно осуществлять по установочным RS-входам при помощи сигнала $x_{\text{mod}} = (C(1:4)=1111)$. Недостаток организации счета по модулю при помощи RS-входов рассмотрен выше.

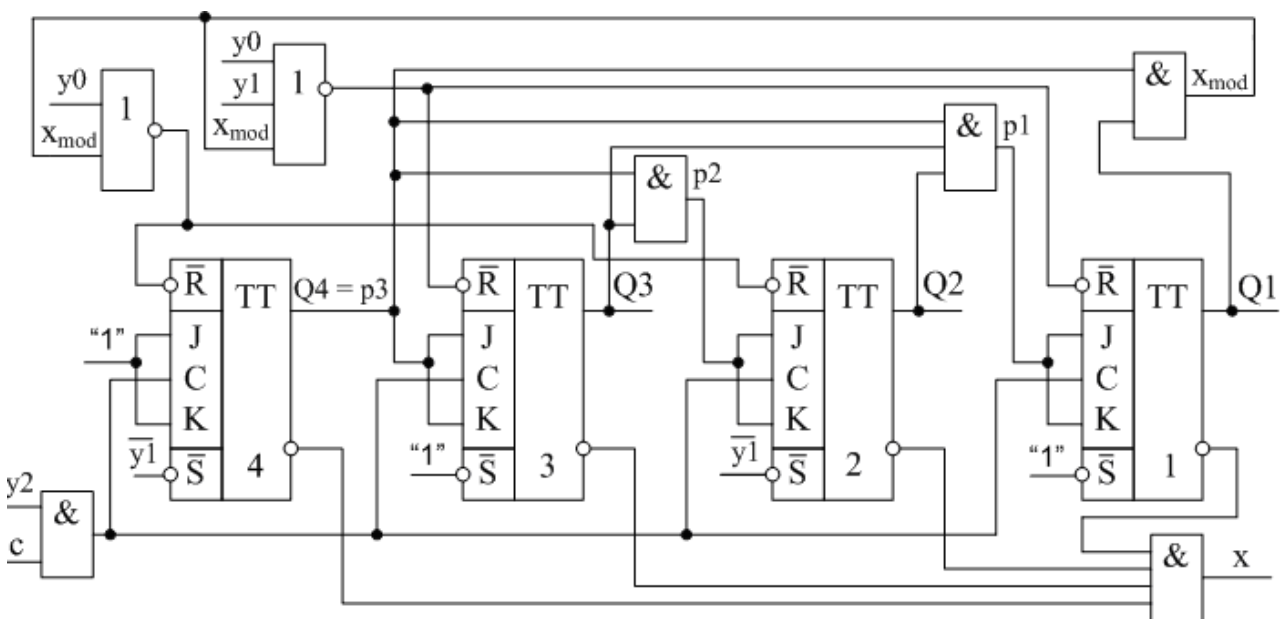


Рисунок 5.4 – Схема четырехразрядного накапливающего инкрементирующего счетчика по модулю 10

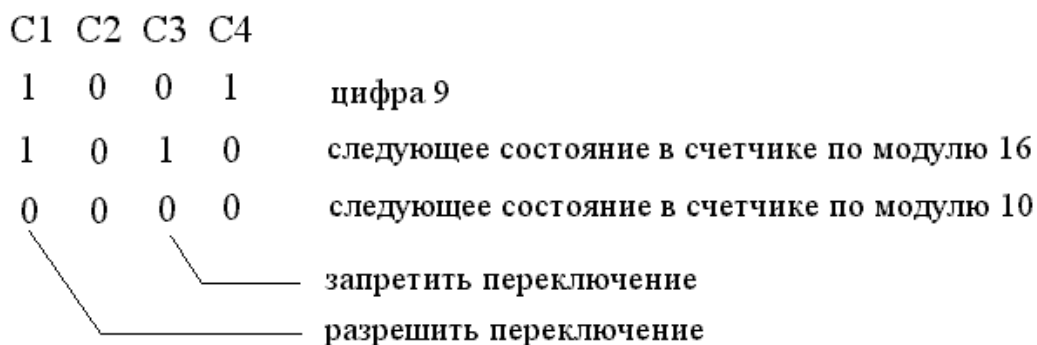


Рисунок 5.6. – Стратегия коррекции переносов в инкрементирующем счетчике по модулю 10.

5.3. Содержание работы

В лабораторной работе необходимо разработать, собрать и отладить схему синхронного трехразрядного двоичного счетчика на JK-триггерах, считающего по модулю М (счетчик должен выполнять только микрооперацию счета). Тип счетчика, способ организации цепей переноса и модуль счета М определяются индивидуальным заданием. В ходе тестирования схемы нужно убедиться в правильности работы счетчика, подавая на его вход импульсы от генератора одиночных импульсов.

В последней части задания требуется разработать двухразрядный восьмеричный счетчик, соединив выход F/8 счетчика-делителя (старший восьмеричный разряд), имеющегося в составе макета «Trigger», и вход трехразрядного накапливающего суммирующего счетчика с последовательным переносом на отдельных JK-триггерах (младший восьмеричный разряд).

5.4. Варианты индивидуальных заданий

Варианты заданий приведены в таблице 5.4.

Таблица 5.4 – Варианты заданий для синхронного счетчика на JK-триггерах.

№ вар.	Тип счетчика	Способ распространения переноса	Модуль счета М
1	Вычитающий	Параллельный	5
2	Суммирующий	Сквозной	6
3	Суммирующий	Параллельный	5
4	Вычитающий	Сквозной	6
5	Вычитающий	Параллельный	7

5.5. Внеаудиторная подготовка

5.5.1. Изучить классификацию счетчиков, методы их проектирования и схемные реализации: [1], стр. 308-318; [2], стр.128-140; [3], стр. 216-244; [4]. Обратить особое внимание на способы распространения переносов, т. к. они несколько отличаются от способов распространения переносов в комбинационных счетчиках.

5.5.2. Разработать схему трехразрядного накапливающего счетчика на раздельных JK-триггерах в соответствии с индивидуальным заданием (счетчик должен выполнять только микрооперацию счета). В качестве сигналов счета рекомендуется использовать синхроимпульсы. Провести анализ работы схемы.

5.5.3. Разработать схему трехразрядного накапливающего суммирующего счетчика с последовательным переносом на раздельных JK-триггерах.

Изучить микросхемы счетчиков K155ИЕ6, K155ИЕ7: [5], стр. 87-91. Эти микросхемы имеют аналоги в серии 74НС**. Одним из аналогов является счетчик-делитель, входящий в состав макета «Trigger».

5.5.4. Разработать двухразрядный восьмеричный счетчик, соединив выход F/8 счетчика-делителя (младший восьмеричный разряд), имеющегося в составе макета «Trigger», и вход трехразрядного накапливающего суммирующего счетчика с последовательным переносом на раздельных JK-триггерах (старший восьмеричный разряд).

5.6. Выполнение работы в лаборатории

Последовательно собрать и отладить счетчики, схемы которых разработаны в процессе домашней подготовки. Проверить их функционирование, подавая на вход импульсы с генератора одиночных сигналов. Двоичные счетчики проверить в динамическом режиме с помощью осциллографа, снять временные диаграммы.

5.7. Контрольные вопросы

5.7.1. Назовите характеристики быстродействия счетчика.

5.7.2. Сравните по быстродействию схемы с последовательным, сквозным и параллельным переносами.

5.7.3. В каких случаях целесообразно использовать цепи группового переноса?

5.7.4. Сформулируйте общий метод настройки счетчика на заданный модуль счета.

5.7.5. Как синтезировать счетчик, меняющий состояния в заданной последовательности? Синтезируйте трехразрядный счетчик, считающий в последовательности 0,3,5,2,7,0,3,... .

6. ЛАБОРАТОРНАЯ РАБОТА №6. СХЕМЫ СРАВНЕНИЯ КОДОВ

6.1. Цель работы

Изучить способы формирования осведомительных сигналов $A=B$, $A>B$, получить навыки наладки и использования схем сравнения кодов, ознакомиться с описанием микросхем компараторов [2], стр. 158-167.

6.2. Краткие теоретические сведения

Схемой сравнения называется ОЭ, на котором вычисляется значение отношения между двумя позиционными кодами равной длины. Основными отношениями являются выражения $A=B$ и $A>B$, которые в зависимости от значений операндов принимают значения ложь (0) или истина (1).

В микропрограммах отношения используются как логические условия и их значения представляются осведомительными сигналами, вырабатываемыми на выходе соответствующих схем сравнения в ОА. В зависимости от типа вычисляемых отношений выделяются :

- 1) схемы сравнения на равенство (неравенство), вычисляющие отношения $A=B$ ($A \neq B$);
- 2) схемы сравнения на больше (меньше), вычисляющие отношения $A>B$ ($A<B$).

Нетрудно убедиться, что схема сравнения на равенство позволяет вычислять отношения $A=B$ и $A \neq B$ и схема сравнения на больше позволяет вычислять отношения $A>B$, $A<B$, $A \geq B$, $A \leq B$. После ознакомления с данной главой объясните почему.

Схемы сравнения на равенство

Сравнение на равенство (неравенство) выполняется поразрядно над одноименными разрядами операндов. Операнды равны, если все одноименные разряды операндов имеют одинаковые значения. Операнды не равны, если хотя бы в одном разряде операнды имеют различное значение.

Значение признака равенства i -ых разрядов ($i = 1, 2, \dots, n$) сравниваемых слов $A(1:n)$ и $B(1:n)$:

$$r_i = \overline{a_i} \overline{b_i} \vee a_i b_i = (a_i \equiv b_i)$$

Значение признака неравенства i -ых разрядов сравниваемых слов:

$$q_i = \overline{a_i} b_i \vee a_i \overline{b_i} = a_i \oplus b_i \text{ (исключающее или – сложение по модулю 2)}$$

Выше приведены функции в дизъюнктивной нормальной форме. Те же функции в конъюнктивной нормальной форме:

$$r_i = (a_i \vee \overline{b_i})(\overline{a_i} \vee b_i);$$

$$q_i = (a_i \vee b_i)(\overline{a_i} \vee \overline{b_i})$$

Признак равенства R двух n -разрядных двоичных слов $A(1:n)$ и $B(1:n)$ вычисляется как конъюнкция

$$R = r_1 r_2 \dots r_n$$

и признак неравенства Q – как дизъюнкция

$$Q = q_1 \vee q_2 \vee \dots \vee q_n$$

Схема сравнения $R=(A(1:3)=B(1:3))$ приведена на рисунке 6.1. Заметим, что на макете L-Т всего два элемента «Исключающее Или» ($=1$). Третий можно спроектировать на основе элементов «И», «ИЛИ», «НЕ».

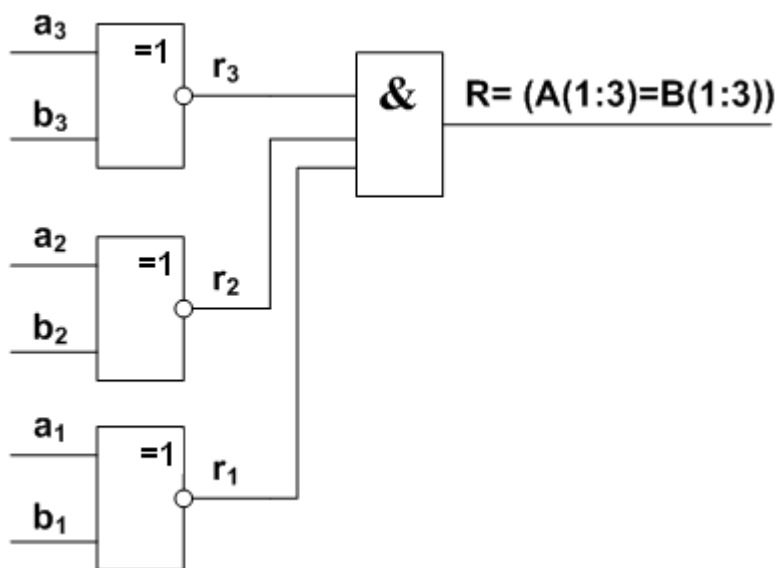


Рисунок 6.1 – Схема сравнения $R=(A(1:3)=B(1:3))$

Микрооперация сравнения слов на равенство может выполняться на регистре, состоящем из триггеров со счетным входом. Регистр хранит код $A(1:n)$. По управляющему сигналу y на счетные входы регистра (входы, переключающие триггеры в противоположное состояние при поступлении на них активного уровня сигнала) подается код $B(1:n)$, в результате чего регистр A устанавливается в состояние $A = A \oplus B$. Если операнды A и B равны, то регистр устанавливается в состояние $A = 0$, отмечаемое осведомительным сигналом $x = (A(1:n) = 00\dots 0)$, который после выполнения микрооперации y сигнализирует о равенстве значений слов A и B .

Микрооперация сравнения слов на равенство может выполняться на сумматорах, если на момент выполнения микрооперации заблокировать цепи межразрядных переносов. При этом сумматор выполняет сложение операндов по модулю 2 и нулевое значение на выходе сумматора соответствует равенству операндов.

Схемы сравнения на больше (меньше)

Сравнение на больше $A > B$ можно выполнять на сумматоре (или вычитателе) путем определения знака разности $(B - A)$: если разность имеет отрицательный знак, то отношение $A > B$ истинно, в противном случае оно ложно. Когда необходимость в использовании сумматора отсутствует и требуется вычислять отношения вида $A > B$, целесообразно использовать схему сравнения, поскольку она экономичнее сумматора в смысле количества оборудования.

Вычисление отношения $A > B$ между двумя n -разрядными словами $A(1:n)$ и $B(1:n)$ сводится к последовательному сравнению разрядов операндов, начиная от старших. Принцип сравнения ясен из рисунка 6.2.

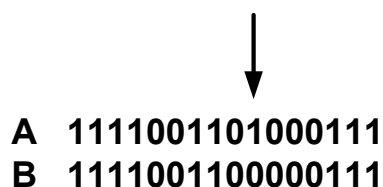


Рисунок 6.2 – Сравнение кодов «на больше»

Если в i -ом разряде ($i = 1, 2, \dots, n$) имеем $a_i b_i = 10$, то отношение $A > B$ истинно; если $a_i b_i = 01$, то отношение $A > B$ ложно, и, если $a_i b_i = 00$ или $a_i b_i = 11$ ($i \neq n$), то значение отношения определяется путем анализа значений в следующем младшем разряде, а при $i = n$ отношение ложно. Определим булеву функцию, на основе которой может быть вычислено значение двоичной переменной S , представляющей отношение $A > B$. Пусть S_i - переменная, принимающая значение 1, если факт истинности отношения вытекает из анализа значений в разряде i .

Из правила сравнения следует:

$$S_i = \begin{cases} a_1 \bar{b}_1 & \text{при } i = 1; \\ R_{i-1} a_i \bar{b}_i & \text{при } i = 2, \dots, n \end{cases},$$

где $R_{i-1} = r_1 r_2 \dots r_{i-1}$ - признак равенства значений операндов в разрядах от 1 до $(i-1)$ включительно и r_j - признак равенства j -ых разрядов сравниваемых кодов. Если $S_i = 1$, то и $S = 1$, т.е.

$$S = S_1 \vee S_2 \vee \dots \vee S_n.$$

Схема сравнения $S=(A(1:3)>B(1:3))$ приведена на рисунке 6.3.

Оцените максимальную задержку сигнала S по отношению к моменту поступления операндов A и B . Если необходимо увеличить быстродействие схемы, то по аналогии со способами ускорения сложения n -разрядная схема делится на группы по k разрядов и организуются обходные цепи передачи сигналов.

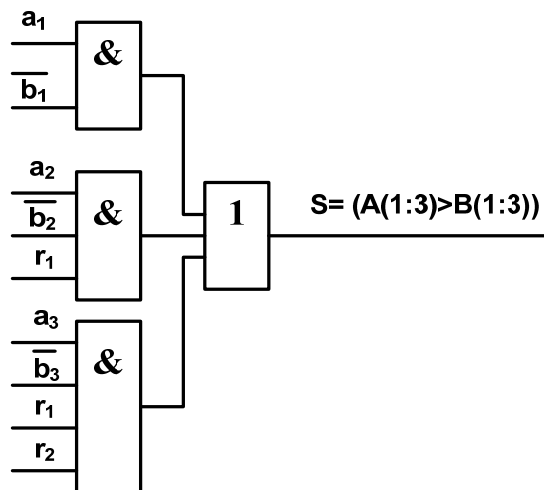


Рисунок 6.3 – Схема сравнения $S=(A(1:3)>B(1:3))$

6.3. Содержание работы

В процессе работы необходимо синтезировать схемы сравнения на равенство и на больше (меньше) трехразрядных двоичных кодов и промоделировать их работу на макете L-T. Затем, используя схему сравнения на равенство и схему трехразрядного накапливающего счетчика, построить двоичный счетчик с настраиваемым модулем счета (рисунок 6.4).

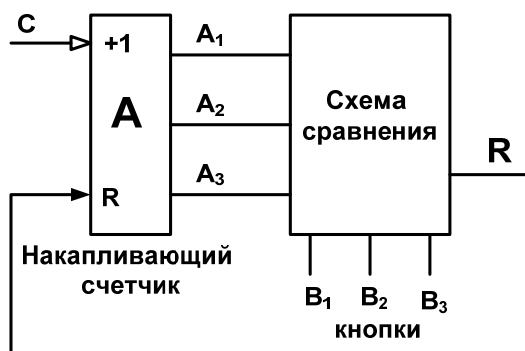


Рисунок 6.4 – Трехразрядный двоичный счетчик с настраиваемым модулем счета

В данной схеме значение модуля счета набирается на кнопочном регистре B . В каждом такте счетчик A выполняет микрооперацию счета, увеличивая свое состояние на 1. В момент равенства кодов A и B сигнал $R=1$ сбрасывает счетчик в нулевое состояние.

6.4. Внеаудиторная подготовка

6.4.1. Изучить принципы построения схем сравнения: [2], стр. 157-167; [5], стр. 142-150.

6.4.2. Разработать схему трехразрядного компаратора, формирующего осведомительный сигнал $X_1=(A(3:1)>B(3:1))$ или $X_2=(A(3:1)<B(3:1))$.

6.4.3. Разработать схему трехразрядного компаратора, формирующего осведомительный сигнал $X_3=(A(3:1)=B(3:1))$.

6.4.4. На основе схемы сравнения на равенство и схемы накапливающего счетчика по модулю 8, синтезировать трехразрядный счетчик, модуль счета которого задается кнопчным регистром. Сигнал X_3 использовать для формирования сброса счетчика.

6.5. Выполнение работы в лаборатории

6.5.1. Собрать и наладить схему сравнения кодов на больше или меньше (задается преподавателем). Подать на ее входы числа А и В с кнопчных регистров и, изменяя их значения, убедиться в том, что сигналы сравнения X_1 (X_2) вырабатываются правильно.

6.5.2. Собрать и отладить схему сравнения кодов на равенство. Изменяя значения А и В убедиться, что сигнал X_3 принимает значение 1 только при $A=B$.

6.5.3. Исследовать схему в динамическом режиме. С этой целью одно из чисел (например, А) подать с выхода счетчика. При фиксированном значении кнопчного регистра В и изменяемом по синхросигналам состоянии счетчика снять с помощью осциллографа временную диаграмму.

6.5.4. Собрать схему счетчика, считающего по модулю, задаваемому кнопчным регистром. Убедиться в правильности работы. Снять временную диаграмму и сравнить ее с построенной заранее.

6.6. Варианты заданий

Предполагается, что каждый студент синтезирует, проектирует и тестирует три операционных элемента: а) схема сравнения на больше, б) схема сравнения на равно, в) трехразрядный счетчик с настраиваемым модулем счета.

6.7. Контрольные вопросы

6.7.1. Где в компьютере могут использоваться схемы сравнения кодов? Приведите примеры.

6.7.2. Оцените быстродействие схем сравнения. Какие вы видите пути увеличения быстродействия?

6.7.3. Какие вы знаете микросхемы компараторов? Как их использовать в схемах?

6.7.4. Как можно вычислить значение осведомительных сигналов $A=B$, $A>B$, не используя схемы сравнения?

7. ЛАБОРАТОРНАЯ РАБОТА № 7. ДЕШИФРАТОРЫ (ШИФРАТОРЫ), МУЛЬТИПЛЕКСОРЫ (ДЕМУЛЬТИПЛЕКСОРЫ) И ИХ ИСПОЛЬЗОВАНИЕ ПРИ ПРОЕКТИРОВАНИИ ЦИФРОВЫХ УСТРОЙСТВ РАЗЛИЧНОГО НАЗНАЧЕНИЯ

7.1. Цель работы.

Изучить методы проектирования и схемные реализации двоичных шифраторов, дешифраторов, мультиплексоров и демультимплексоров, получить навыки синтеза, наладки, исследования и применения схем подобного класса.

7.2. Краткие теоретические сведения

Перечисленные в названии лабораторной работы операционные элементы комбинационного типа предназначены для преобразования многоразрядного входного кода в выходной код, построенный по иному, чем первый правилу. Закон функционирования таких схем обычно задается таблицей входов и выходов.

Дешифратор осуществляет преобразование n -разрядного двоичного позиционного кода в 2^n -разрядный унарный код. Из всех 2^n выходов дешифратора только на одном, а именно на том, номер которого равен поданному на вход двоичному коду, будет логическая единица.

Шифратор выполняет функцию, обратную дешифратору, т.е. преобразует унарный двоичный код в позиционный, значение которого определяет номер единичной компоненты входного набора.

Синтез схем шифраторов и дешифраторов сводится к построению для каждого разряда выходного слова булевой функции, определяющей связь данного разряда с входными наборами двоичных переменных.

В таблицах 7.1, 7.2 представлены, соответственно правила преобразования для кодов: «код 2,1» и «код 1 из 4».

Таблица 7.1.

Входной код «2,1»		Выходной код «1 из 4»			
x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Таблица 7.2

Выходной код «1 из 4»				Входной код «2,1»	
x_3	x_2	x_1	x_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Таблица 7.1 задает работу дешифратора, а таблица 7.2 – шифратора. Соответствующие системы булевых функций представлены ниже:

$$\left\{ \begin{array}{l} y_0 = \overline{x_1} \overline{x_0} \\ y_1 = \overline{x_1} x_0 \\ y_2 = x_1 \overline{x_0} \\ y_3 = x_1 x_0 \end{array} \right. ; \quad \left\{ \begin{array}{l} y_0 = \overline{x_3} \overline{x_2} \overline{x_1} \overline{x_0} \vee x_3 \overline{x_2} \overline{x_1} \overline{x_0} \\ y_1 = \overline{x_3} \overline{x_2} \overline{x_1} x_0 \vee x_3 \overline{x_2} \overline{x_1} x_0 \end{array} \right.$$

В цифровых устройствах часто решается задача передачи информации от m различных источников к n приемникам через общую шину (канал общего использования). Для решения такой задачи на входе шины устанавливается операционный элемент, называемый мультиплексором, который в зависимости от кода (адреса A_m) подключает ко входу шины один из источников информации. При этом, установленный на выходе шины операционный элемент, называемый демultipлексором, обеспечивает передачу информации приемнику, определяемому кодом (адресом A_n).

Логика функционирования мультиплексора для $m=4$ представлена таблицей 7.3, где x_0, x_1, x_2, x_3 – выходы независимых источников информации. Таблица 7.4 определяет логику функционирования демultipлексора в случае $n=4$, где y_0, y_1, y_2, y_3 – входы приемников информации.

Таблица 7.3

Входы	Адрес $A_1 \quad A_0$		Выход y
x_3, x_2, x_1, x_0	0	0	x_0
	0	1	x_1
	1	0	x_2
	1	1	x_3

Таблица 7.4

Вход	Адрес $A_1 \quad A_0$		Выход			
	A_1	A_0	y_3	y_2	y_1	y_0
x	0	0	0	0	0	x
	0	1	0	0	x	0
	1	0	0	x	0	0
	1	1	x	0	0	0

Мультиплексор и демultipлексор содержат дешифратор адреса. Дешифратор управляет логическими вентилями, разрешающими передачу информации (при конкретном адресе) только через один из вентиляей.

На рисунках 7.1 и 7.2 изображены схемы мультиплексора и демultipлексора соответственно. Поведение операционных элементов специфицировано таблицами 7.3 и 7.4.

Как уже говорилось выше, мультиплексоры, дешифраторы, демultipлексоры могут применяться для синтеза операционных элементов, например, шин для передачи данных.

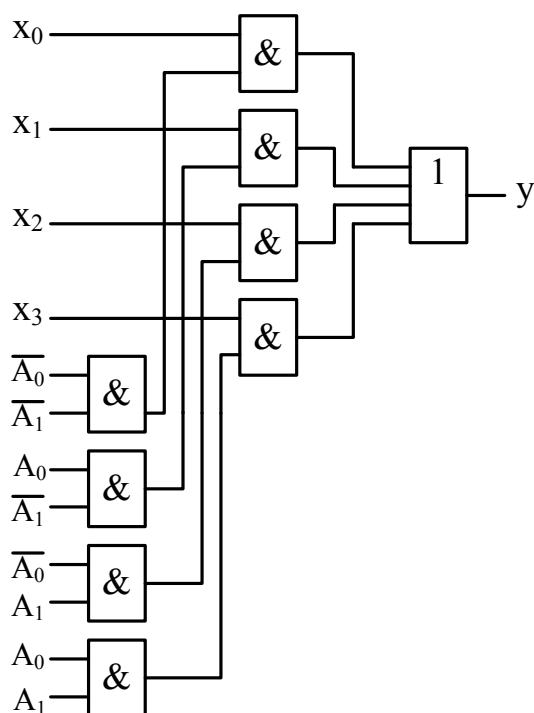


Рисунок 7.1 – Мультиплексор

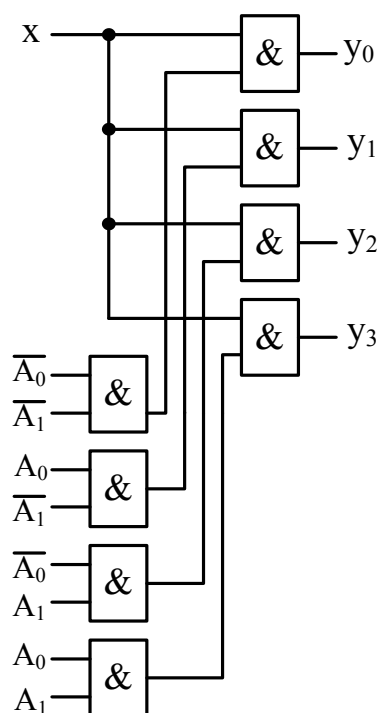


Рисунок 7.2 – Демультимплексор

Шина – операционный элемент (ОЭ), служащий для выполнения микрооперации передачи информации, например, между регистрами:

y1: $B(1:n) := A(1:n)$,

y2: $B(1:n) := \neg A(1:n)$ (инверсная передача).

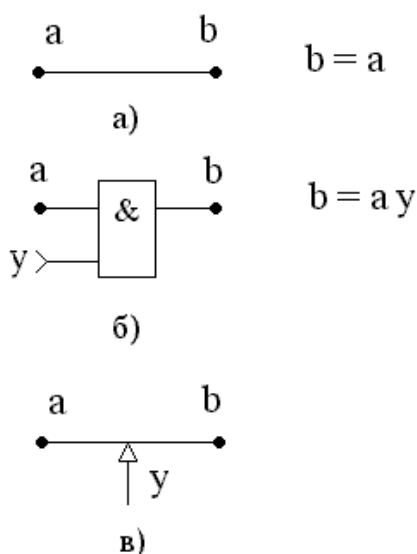


Рисунок 7.3 — Условные обозначения неуправляемой (а) и управляемой (б, в) цепей передачи данных

Регистр А называют источником информации, а регистр В – приемником.

Шины состоят из цепей.

Цепь – ОЭ, служащий для передачи одного бита информации. В простейшем случае цепь – это проводник. Информация на его концах совпадает ($b = a$, рисунок 7.3, а). На рисунке 7.3,б показана управляемая цепь. Информация передается по цепи только при наличии управляющего сигнала y ($b = a \cdot y$). Условное изображение управляемой шины на структурных схемах ОУ приведено на рисунке 7.3, в.

Виды цепей:

- однофазные,
- парафазные.

Однофазные цепи (управляемые или неуправляемые) передают только прямое или только инверсное значение сигнала. Парафазные цепи передают оба значения сигнала (рисунок 7.4).

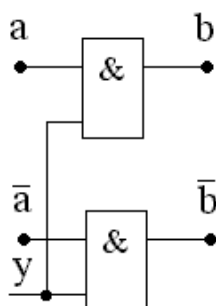


Рисунок 7.4 Парафазная управляемая цепь

Сложность парафазной шины увеличивается в два раза по сравнению с однофазной, что позволяет достичь выигрыша по быстродействию, если приемниками информации являются RS-триггеры (JK-триггеры). В данном случае парафазная шина позволяет передать информацию за один такт.

Для однофазной шины используется два такта.

1) В первом такте приемный регистр обнуляется (на R-входы всех триггеров регистра подаются единицы $R_i=1$).

2) Во втором такте устанавливаются только единицы в нужных разрядах регистра ($S_i=a$).

Проектирование одноктактных и двухтактных (парафазных и однофазных) цепей приема и установки кода рассматривалось выше в примере 1 (синтез регистра на отдельных триггерах).

Прием информации на D-триггер с помощью однофазной цепи осуществляется за один такт.

N-разрядная шина состоит из N одноразрядных цепей.

Шины могут быть отдельными и общими. При реализации передач в системе регистров отдельными шинами каждая передача осуществляется по собственной шине, связывающей соответствующий источник с соответствующим приемником, при подаче в схему соответствующего управляющего сигнала y_i . Общая шина (магистраль) – это схемное решение, в котором все возможные передачи между регистрами осуществляются по единственной шине.

Раздельные шины

Если необходимо на один регистр-приемник передать информацию с нескольких регистров-источников, то соответствующие цепи объединяются при помощи логических элементов. Пусть на регистр В может передаваться информация с регистров $A_1, A_2, \dots, A_k$:

$$y_1: B(1:n) := A_1(1:n);$$

$$y_2: B(1:n) := A_2(1:n);$$

...

$$y_k: B(1:n) := A_k(1:n).$$

Рисунок 7.5,а иллюстрирует идеологию раздельной шины для

осуществления указанных микроопераций. На рисунке 7.5,б приведена схема связей между i -ми разрядами регистров без привязки к конкретной серии микросхем.

Оценим сложность схемы, приведенной на рисунке 7.5,б, по числу входов логических элементов. Для n -разрядных регистров она составит $C = 3kn$, где k – количество источников. В общем случае, если в системе не один, а m приемников, сложность схемы составит $C = 3kmt$. Если количество передач между регистрами $M = km$ в системе достаточно велико, то реализация этих передач отдельными шинами может быть сравнительно дорогой. В связи с этим появилась идея реализации всех передач одной – общей – шиной.

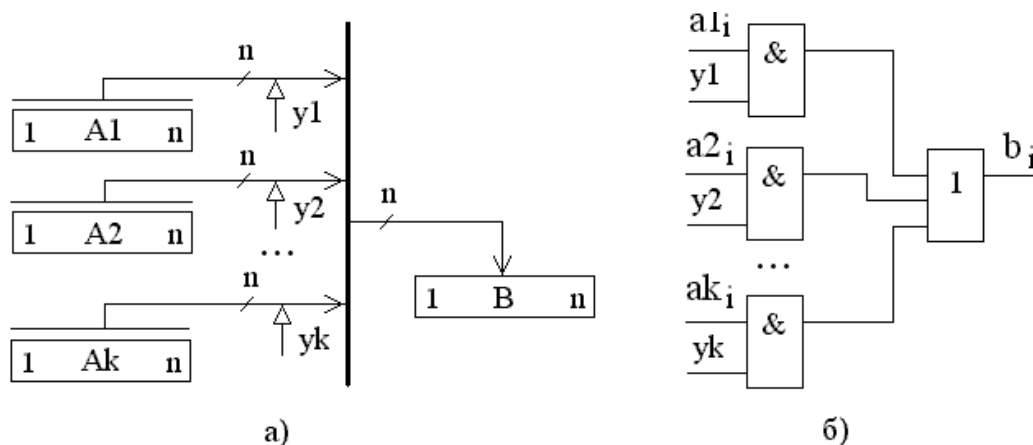


Рисунок 7.5 – Реализация передач в системе регистров
раздельными шинами: а) структурная схема;
б) функциональная организация для i -го разряда регистров

Пример. Синтезировать раздельные парафазные шины (функциональная схема i -го разряда) для осуществления следующих передач в системе из трех регистров:

$$y1: P1 := \overline{P2}; \quad y2: P2 := P1; \quad y3: P3 := P1; \quad y4: P2 := \overline{P3}$$

Парафазные шины применяются, когда приемником является регистр, построенный на RS (JK)-триггерах для того, чтобы межрегистровые передачи могли быть осуществлены за один такт. Поэтому в схеме в качестве i -го разряда каждого регистра будет использован RS-триггер. Для осуществления прямой передачи на S-вход приемника должно быть передано прямое значение источника, домноженное на соответствующий управляющий сигнал, а на R-вход приемника – инверсное значение источника, домноженное на тот же сигнал.

Для осуществления инверсной передачи на S-вход приемника должно быть передано инверсное значение источника, домноженное на соответствующий управляющий сигнал, а на R-вход приемника – прямое значение источника, домноженное на тот же сигнал.

Функции возбуждения входов триггеров:

$$R1 = y1 P2; \quad S1 = y1 \overline{P2};$$

$$R2 = y2 \overline{P1} \vee y4 P3; \quad S2 = y2 P1 \vee y4 \overline{P3};$$

$$R3 = y3 \overline{P1}; \quad S3 = y3 P1$$

Функциональная схема i-го разряда отдельных шин, осуществляющих заданные передачи, приведена на рисунке 7.6. Схема выполнена без привязки к конкретной серии микросхем.

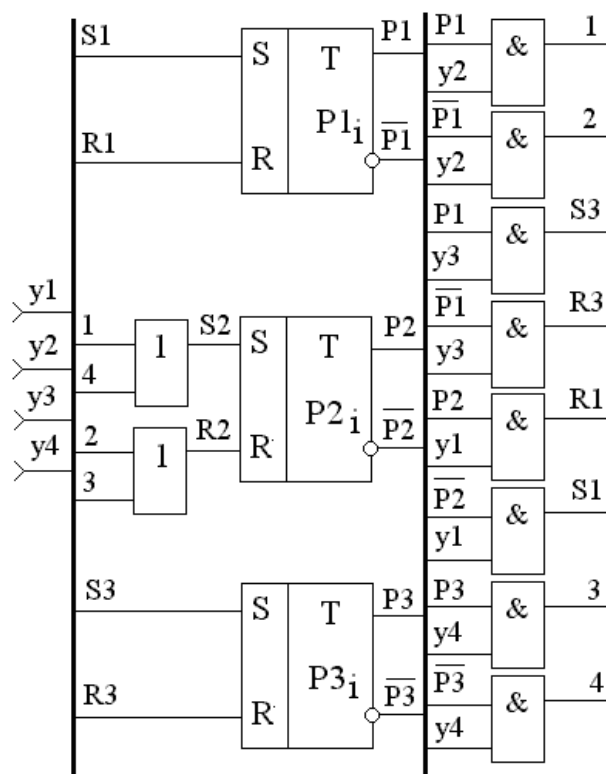


Рисунок 7.6 – Реализация передач в системе регистров отдельными шинами

Общая шина (магистраль)

Структурная схема общей шины приведена на рисунке 7.7,а. Функциональная схема i-го разряда общей шины приведена на рисунке 7.7,б.

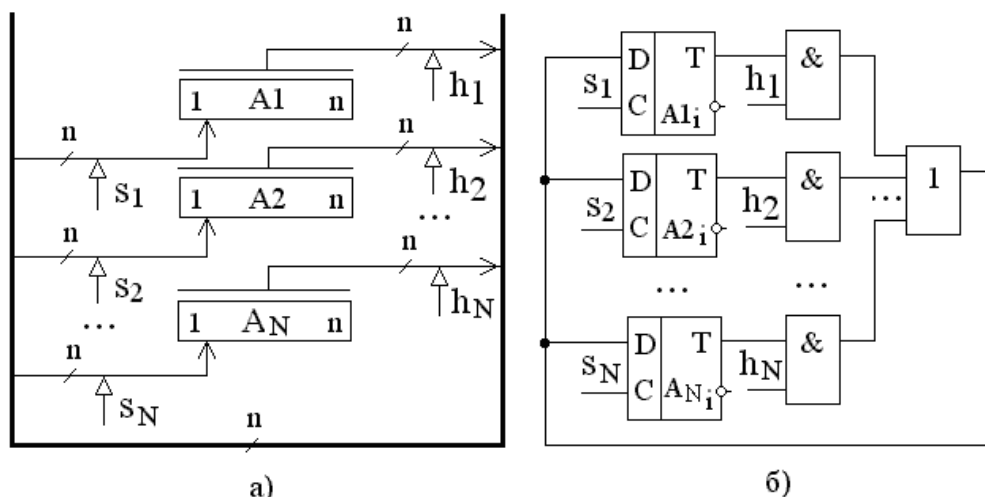


Рисунок 7.7 – Реализация межрегистровых передач общей шиной:

а) структурная схема общей шины;

б) функциональная схема i -го разряда общей шины

При осуществлении передачи $A_k := A_j$ через общую шину выход источника подключается ко входу общей по сигналу h_j , выход общей шины подключается ко входу приемника по сигналу s_k . Для осуществления указанной передачи управляющие сигналы h_j и s_k должны быть поданы в схему одновременно. Это значит что для осуществления любой передачи управляющий автомат должен выработать два соответствующих этой передаче унитарных кода:

$h_1 h_2 \dots h_N$ и $s_1 s_2 \dots s_N$. То есть управление в данном случае осуществляется унитарными управляющими кодами. Часто на практике УА для осуществления передачи по общей шине вырабатывает позиционные коды номера источника и номера приемника (управление позиционными управляющими кодами). В этом случае выбор источника и подключение его ко входу общей шины осуществляется посредством мультиплексора, подключением нужного приемника ко входу общей шины управляет дешифратор (рисунок 7.8).

Следует отметить, что в рассмотренных схемах общих шин (рисунки 7.7, 7.8) передаются только прямые значения с регистров-источников. Если в системе возможны передачи инверсных значений, то число источников увеличивается в общем случае в два раза (так как будут использоваться не только прямые, но и инверсные выходы триггеров). В дальнейшем под источником будем понимать номер регистра и тип передачи (прямая или инверсная).

Сравнение общей и раздельной шины.

1) По быстродействию. Более производительными считаются раздельные шины, так как они позволяют совместить некоторые передачи в одном такте. Передачи $y1: A := V$ и $y2: C := D$ могут производиться одновременно (но не $y1: A := V$ и $y2: C := A$).

7.4. Варианты индивидуальных заданий

Варианты индивидуальных заданий приведены в таблице 7.5.

Таблица 7.5

№ варианта	Реализуемые передачи данных	Тип триггера
1	$P0 := P3; P0 := \overline{P2}; P1 := P2; P2 := P0; P3 := \overline{P1}$	JK
2	$P0 := \overline{P3}; P1 := \overline{P2}; P1 := P0; P2 := P3; P3 := P1$	D
3	$P0 := \overline{P1}; P1 := \overline{P2}; P2 := \overline{P3}; P2 := P0; P3 := P2$	RS
4	$P0 := P2; P1 := \overline{P3}; P2 := \overline{P1}; P3 := \overline{P0}; P3 := P0$	D
5	$P0 := \overline{P2}; P1 := \overline{P3}; P2 := P1; P3 := P0; P3 := \overline{P2}$	JK

7.5. Внеаудиторная подготовка к работе

7.5.1. Изучить классификацию дешифраторов. Мультиплексоров и демультиплексоров, методы их проектирования и схемы реализации [1], стр. 173-178, [2], стр. 141-157, [3], стр. 110-132.

7.5.2. Изучить описание микросхем серий ТТЛ-логики (K155, 74НС). По описаниям микросхем построить таблицы их функционирования.

7.5.3. Построить схемы реализации дешифратора и мультиплексора, необходимые для выполнения индивидуального задания на макете.

7.5.4. Разработать схему передачи данных (i-ый разряд) в системе регистров в соответствии с индивидуальным заданием. Для этого требуется синтезировать общую шину. Номера источников и приемников информации задаются в виде двоичных кодов. Передача информации синхронизируется синхроимпульсами. На входе шины используется мультиплексор, на адресные входы которого подается номер источника информации. Информация по адресам приемников распределяется с помощью демультиплексора. В случае, если выход шины подключается ко всем входам D-триггеров регистров, которые могут принимать информацию, то достаточно однофазного сигнала. Если прием информации осуществляется на триггеры с RS или JK-входами, необходимо организовать парафазный сигнал с выхода шины.

7.5.5. Разработать схему i-го разряда отдельной шины для осуществления заданных передач. Сравнить с предыдущей схемой по сложности и быстродействию.

7.6. Выполнение работы в лаборатории

7.6.1. Собрать схему дешифратора (2 входа на 4 выхода). Соединив его входы с кнопочным регистром, провести эксперимент и сравнить экспериментальные данные с теоретическими. Аналогичный эксперимент

провести с шифратором.

7.6.2. По аналогии с предыдущим пунктом собрать и исследовать схемы мультиплексора и демultipлексора с двумя адресными входами.

7.7. Контрольные вопросы

7.7.1. Сравнить по быстродействию и сложности различные структуры дешифраторов.

7.7.2. Как построить сеть из дешифраторов с заданным количеством входов с целью увеличения разрядности дешифрируемого кода?

7.7.3. Что такое мультиплексор? Как с помощью мультиплексора реализовать произвольную булеву функцию?

7.7.4. Как подключить к общей шине n двоичных сигналов, если $n > k$, где k – число информационных входов мультиплексора?

7.7.5. Где в компьютере используются мультиплексоры и дешифраторы. Привести примеры.

7.7.6. Приведите формулы для оценки сложности общей и отдельной шины.

8. ЛАБОРАТОРНАЯ РАБОТА № 8. ДВОИЧНЫЕ СУММАТОРЫ

8.1. Цель работы

Изучить методы проектирования и схемные реализации одноразрядных сумматоров, получить навыки сборки, наладки и использования схем двоичных сумматоров, приобрести опыт работы с сумматорами в интегральном исполнении.

8.2. Краткие теоретические сведения

Сумматор – это операционный элемент, предназначенный для выполнения микрооперации сложения чисел. Если операнды и результат сложения представляются в двоичной системе счисления, то сумматор называется двоичным. Сумматоры подразделяются на два типа: комбинационные и накапливающие. Комбинационные сумматоры не содержат запоминающих элементов и реализует микрооперацию сложения в виде $C:=A+B$. Накапливающие сумматоры содержат регистр, на котором перед началом микрооперации хранится слагаемое и на момент окончания сложения – сумма. Накапливающие сумматоры реализуют микрооперацию сложения в виде $C:=C+A$.

Структура сумматоров зависит от типа используемых логических и запоминающих элементов и ограничений, налагаемых на длительность микрооперации сложения или количество оборотов.

Операционное устройство может производить арифметические и логические операции в соответствии с заложенной в его основу микропрограммой параллельно, последовательно и параллельно-последовательно. Параллельный способ выполнения операции суммирования двух чисел с фиксированной точкой предполагает, что микрооперация сложения, используемая в микропрограмме суммирования, выполняется над всеми разрядами операндов одновременно (параллельно). Последовательный способ выполнения операции суммирования означает, что соответствующая микрооперация сложения выполняется последовательно над каждым i -ым разрядом суммируемых чисел, начиная с младшего. Поэтому для осуществления сложения двух n -разрядных чисел на последовательном двоичном сумматоре потребуется n тактов суммирования (потери времени при этом компенсируются существенной экономией аппаратуры). Компромиссным способом является параллельно-последовательный способ выполнения сложения на двоичном сумматоре. Он предполагает, что микрооперация сложения выполняется над m разрядами суммируемых чисел (т.е. m разрядов суммируются параллельно), $m < n$, n кратно m . При этом в микропрограмме сложения должна быть предусмотрена микрооперация сдвига регистра, на который записывается сумма, на m разрядов вправо для того, чтобы освободить место для записи суммы следующих по старшинству m разрядов

суммируемых слов.

На микрооперации сложения основаны алгоритмы (микропрограммы) выполнения основных арифметических операций в арифметико-логическом устройстве (сложение, вычитание, умножение, деление). Три перечисленных способа выполнения суммирования слов на соответствующих операционных элементах применимы и к выполнению поразрядных логических операций над словами. В соответствии с вышесказанным, арифметико-логические устройства компьютера по способу выполнения операций делятся на параллельные, последовательные и параллельно-последовательные.

Параллельный комбинационный сумматор с последовательным переносом. При построении применяется стандартный метод – синтез типового

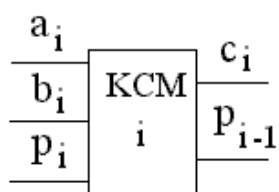


Рисунок 8.1 — Одноразрядный комбинационный сумматор

разряда. N-разрядный параллельный сумматор с последовательным переносом строится из n одноразрядных комбинационных сумматоров (ОКСм). Функция одноразрядного комбинационного сумматора (рисунок 8.1) представлена в таблице 8.1.

Таблица 8.1

Номер входного набора	Входы			Выходы	
	p_i	a_i	b_i	p_{i-1}	c_i
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

Схема, изображенная на рисунке 1, суммирует a_i и b_i - значения i -ых разрядов слов $A(1:n)$ и $B(1:n)$ – а также p_i - перенос в i -ый разряд суммы из младшего $(i+1)$ -го разряда.

Из таблицы 8.1 получены следующие функции для определения выходных сигналов ОКС.

Значение i -го разряда суммы:

$$c_i = \overline{p_i} \overline{a_i} b_i \vee \overline{p_i} a_i \overline{b_i} \vee p_i \overline{a_i} \overline{b_i} \vee p_i a_i b_i ;$$

Значение переноса в старший $(i-1)$ -ый разряд суммы:

$$p_{i-1} = \overline{p_i} a_i b_i \vee p_i \overline{a_i} b_i \vee p_i a_i \overline{b_i} \vee p_i a_i b_i = a_i b_i \vee p_i b_i \vee p_i a_i$$

По этим функциям строится двухуровневый двоичный одноразрядный комбинационный сумматор с ценой 25 (входов логических элементов) и временем задержки 2τ , где τ - задержка на логическом элементе.

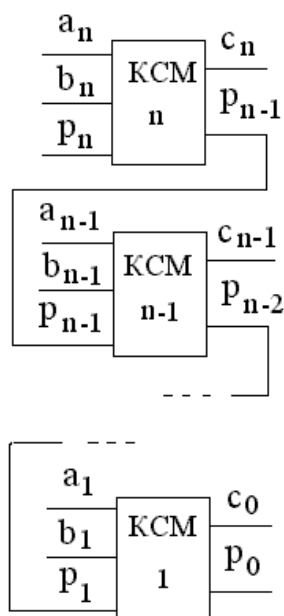


Рисунок 8.2 — N-разрядный комбинационный сумматор

Схема n-разрядного комбинационного сумматора, построенного на базе ОКС представлена на рисунке 8.2.

В данном сумматоре сигналы переносов распространяются последовательно от младших разрядов к старшим. По этой причине указанный сумматор называется сумматором с последовательным переносом. Длительность микрооперации сложения определяется суммой времени распространения переносов и времени формирования суммы в одном разряде:

$$T = L \cdot \tau_{пер} + \tau_{сум} + \mathcal{G},$$

где L — максимальная длина переноса, равная числу соседних разрядов, через которые распространяется сигнал переноса при выполнении сложения, $\mathcal{G}$ — небольшой

запас времени для надежности. При наличии переноса между всеми разрядами сумматора (например, при сложении чисел $A=11\dots11$ и $B=00\dots01$ перенос формируется в младшем разряде сумматора и проходит через все остальные разряды) $L = n$, и длительность микрооперации сложения составляет

$$T = n \cdot \tau_{пер} + \tau_{сум} + \mathcal{G}$$

Поскольку фактическое время распространения переноса в схеме рисунка 8.2 не определяется, то для выполнения микрооперации сложения отводится такт, длительность которого определяется максимальным временем распространения переносов.

Методы ускорения микрооперации сложения

Случай сложения с наличием переносов между всеми разрядами сумматора достаточно маловероятен. Доказано, что в n-разрядном сумматоре, на вход которого поступают слагаемые, равномерно распределенные в интервале $[2^n - 1]$, средняя длина самого длинного переноса, затрагивающего группу соседних разрядов, не превышает $L_{cp} = \log_2 n$. Так, в 32-х разрядном сумматоре переносы пробегают в среднем не более чем через 5 соседних разрядов. Если в сумматор встроить схему, определяющую момент окончания сложения, то время сложения уменьшается в $n/\log_2 n$ раз. Сумматор, в котором вырабатывается осведомительный сигнал, отмечающий окончание сложения, называется асинхронным сумматором.

Еще одним методом уменьшения длительности микрооперации сложения является использование сумматора с групповым переносом. Сумматор делится на n/k групп, каждая из которых содержит k соседних разрядов. В сумматор вводятся цепи, ускоряющие процесс распространения переносов между группами и в пределах каждой группы.

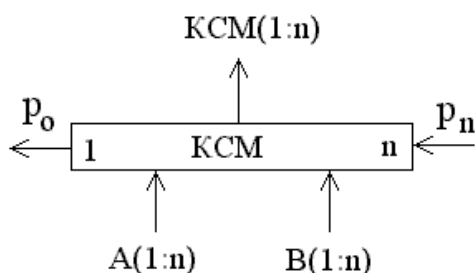


Рисунок 8.3 — Условное обозначение комбинационного сумматора на операционных схемах

Комбинационные сумматоры независимо от их внутренней структуры обозначаются на операционных (структурных) схемах как представлено на рисунке 8.3. Если перенос в младший разряд сумматора всегда равен нулю, то вход P_n на операционных схемах не обозначается. Если перенос из старшего разряда сумматора никак не используется (теряется), то выход P_0 также не

обозначается. Выход P_0 может быть использован для увеличения разрядности сумматора (тогда он подается на вход P_n следующего (старшего) сумматора). При сложении чисел в обратных кодах в сумматоре организуется цепь циклического переноса путем подсоединения выхода P_0 (старшего сумматора, если необходимую разрядность обеспечивают несколько сумматоров) ко входу P_n (младшего сумматора).

Накапливающие сумматоры

В накапливающем сумматоре значение суммы вычисляется путем сложения предыдущего ее значения со слагаемым, т.е. микрооперация сложения выполняется в виде $C := C + A$. Для присваивания сумме начального значения в накапливающем сумматоре обычно реализуется операция установки $C := 0$. Накапливающий сумматор можно рассматривать как композицию регистра, используемого для хранения значения C и комбинационного сумматора KCM , вычисляющего сумму $C + A$. Структурная схема параллельного накапливающего сумматора изображена на рисунке 8.4,а, условное обозначение на операционных схемах – на рисунке 8.4,б.

Время такта суммирования в данном случае определяется задержкой комбинационного сумматора и временем приема значения суммы на регистр.

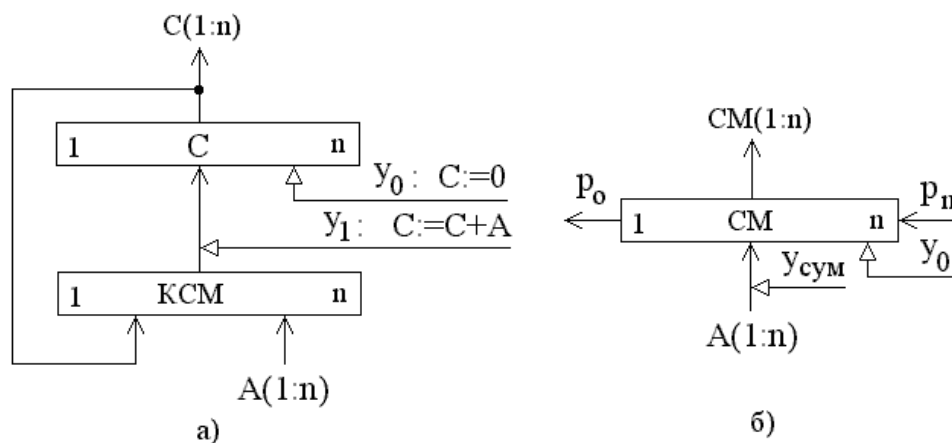


Рисунок 8.4 — Параллельный накапливающий сумматор:
а) - структура, б) - обозначение на операционных схемах

Последовательный накапливающий сумматор.

Синтезируем последовательный n -разрядный накапливающий сумматор на базе ОКСм (рассмотрен выше), сдвигающего регистра $C(1:n)$ и триггера P (для запоминания переноса).

Структурная схема последовательного сумматора приведена на рисунке 8.5. Первое слагаемое загружается в регистр $C(1:n)$, с внешнего регистра $T(1:n)$. Второе слагаемое загружается в регистр $B(1:n)$, с внешнего регистра $Q(1:n)$.

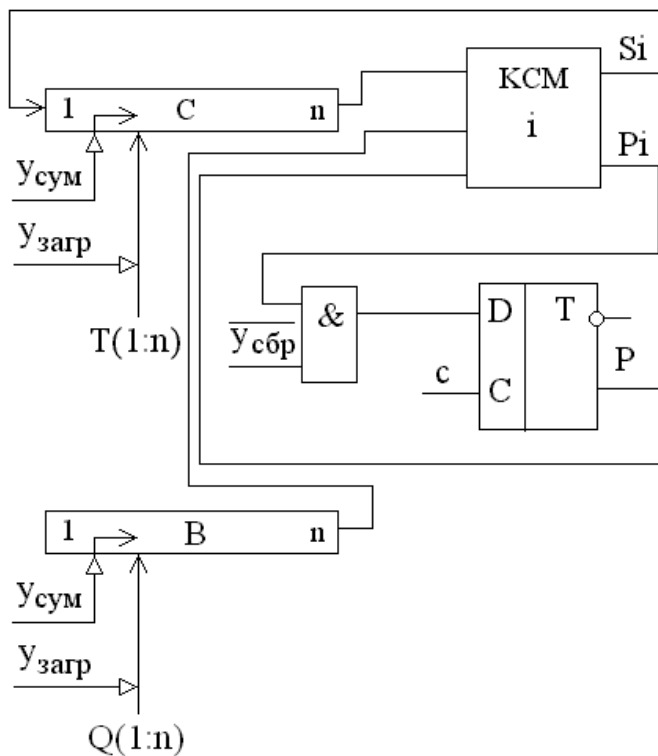


Рисунок 8.5 — Структурная схема последовательного накапливающего сумматора ($c = Y_{сум} \vee Y_{сбр}$)

Сложение осуществляется за n тактов. В очередном такте на ОКС складываются очередные цифры слагаемых C_n и B_n и перенос P , полученный в предыдущем такте при сложении предыдущих младших цифр. Перенос P_i , полученный в текущем такте, запоминается на триггере для использования в следующем такте. По истечении n тактов сумма должна оказаться в регистре C . Для записывания в регистр C очередного разряда суммы S_i и подачи следующей цифры первого и второго слагаемых на ОКС используется микрооперация сдвига (при сдвиге в n -ном разряде регистра C оказывается следующая по старшинству цифра первого

слагаемого, а освобождающийся старший разряд регистра C доопределяется значением очередной вычисленной цифры суммы). При сложении младших разрядов перенос должен быть равен нулю, поэтому необходимо предусмотреть микрооперацию сброса триггера P . Если в триггере есть установочные входы, то сигнал $Y_{сбр}$ можно подавать на установочный R-вход триггера.

Процесс сложения при помощи последовательного сумматора является не микрооперацией, а микропрограммой (почему?).

8.3. Содержание работы

В процессе выполнения работы необходимо синтезировать одноразрядный двоичный комбинационный сумматор, собрать и исследовать его на макете L-T, построить на его основе и с использованием сдвигающего регистра последовательный сумматор. После этого необходимо построить схему двухразрядного параллельного накапливающего сумматора на основе ОКСм,

полусумматора и запоминающих элементов и исследовать его работу в динамическом режиме. Индивидуальное задание может быть сформулировано преподавателем в виде особых требований к схемной реализации одноразрядного сумматора (по быстродействию, количеству элементов и т. д.).

8.4. Внеаудиторная подготовка к работе

8.4.1. Изучить классификацию сумматоров, методы проектирования и схемные реализации комбинационных и накапливающих сумматоров: ([1], стр. 181-184; [2], стр. 173-185; [3], стр. 133-141).

8.4.2. Разработать принципиальную электрическую схему одноразрядного сумматора. Оценить ее быстродействие.

8.4.3. Изучить микросхемы К155ИМ3 ([5], стр. 157-158) и К155ИР1 ([5], стр. 104-107) – аналоги 74НС83 и 74НС95).

9.4.4. Синтезировать параллельный четырехразрядный накапливающий сумматор, используя схемы 74НС83 и 74НС95.

8.5. Выполнение работы в лаборатории

8.5.1. Собрать и наладить схему одноразрядного комбинационного сумматора (ОКСм) на макете L-Т. Убедиться в ее работоспособности, подавая с кнопок на входные полюса все возможные комбинации и фиксируя значения сигналов суммы и переноса.

Для обеспечения возможности размещения схемы одноразрядного сумматора на макете Logic, рекомендуется преобразовать функции ОКСм следующим образом:

$$\begin{aligned}
 C &= \bar{a}\bar{b}p \vee \bar{a}b\bar{p} \vee a\bar{b}\bar{p} \vee abp = (\bar{a}\bar{b} \vee ab)p \vee (\bar{a}b \vee a\bar{b})\bar{p} = \\
 &= (\overline{a \oplus b})p \vee (a \oplus b)\bar{p} = (a \oplus b) \oplus p \\
 P &= \bar{a}b\bar{p} \vee \bar{a}b\bar{p} \vee ab\bar{p} \vee abp = (\bar{a}b \vee ab)p \vee ab(\bar{p} \vee p) = \\
 &= (a \oplus b)p \vee ab
 \end{aligned}$$

Схема размещения ОКСм на макете Logic изображена на рисунке 8.6.

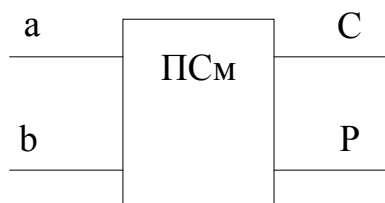


Рисунок 8.8 – Полусумматор

Таблица 8.2 – Таблица истинности полусумматора

a	b	C	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

По таблице 8.2 получены следующие функции для выходов полусумматора. Они преобразованы с целью обеспечения возможности размещения схемы на макете Logic.

$$C = \bar{a}b \vee a\bar{b} = (a \vee b)(\bar{a} \vee \bar{b}) = (a \vee b)\bar{a}\bar{b};$$

$$P = ab$$

Схема полусумматора изображена на рисунке 8.9.

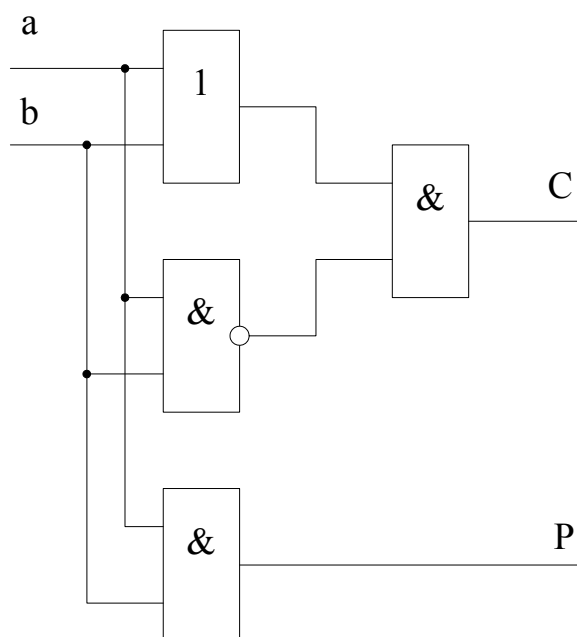


Рисунок 8.9 – Схема полусумматора

8.5.4. На основе схем ОКСм и ПСм собрать и наладить схему накапливающего двухразрядного параллельного сумматора. Проверить схему в статическом и динамическом режиме. В последнем случае поставить схему в режим работы счетчика и снять осциллограммы на выходах разрядов сумматора. Определить минимальное время такта суммирования (через время задержки срабатывания логических и запоминающих элементов).

Схема параллельного двухразрядного накапливающего сумматора изображена на рисунке 8.10. Данный операционный элемент выполняет две микрооперации:

$$y_0 : A(1:0) := 00; \quad y_{сум} : A(1:0) := A(1:0) + B(1:0)$$

Управляющий сигнал y_0 можно подать с кнопки, управляющий сигнал $y_{сум}$ целесообразно подать с генератора одиночных импульсов.

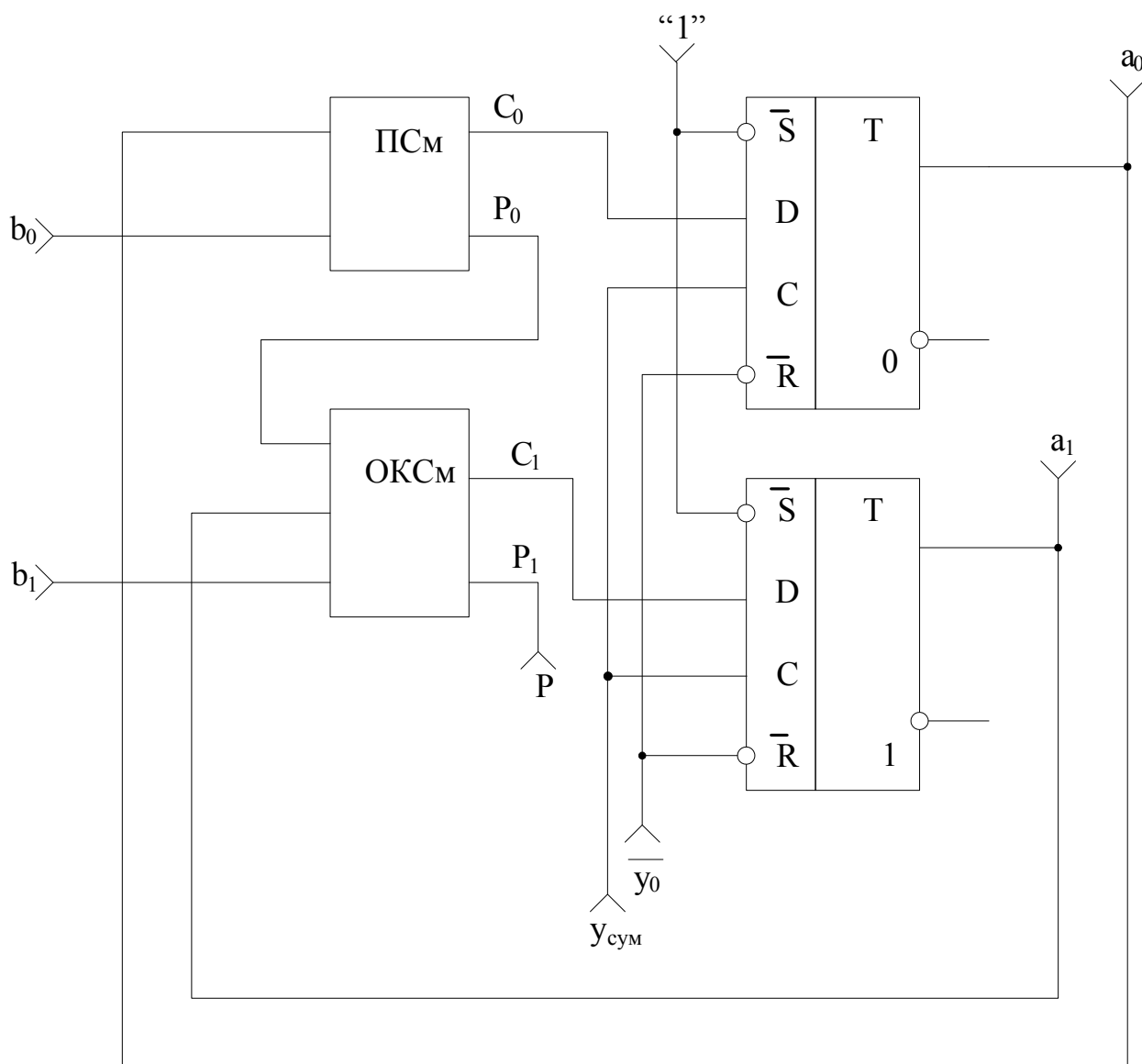


Рисунок 8.10 – Двухразрядный параллельный накапливающий сумматор

Схема параллельного двухразрядного накапливающего сумматора в элементном базисе макета «Logic-Trigger» изображена на рисунке 8.11.

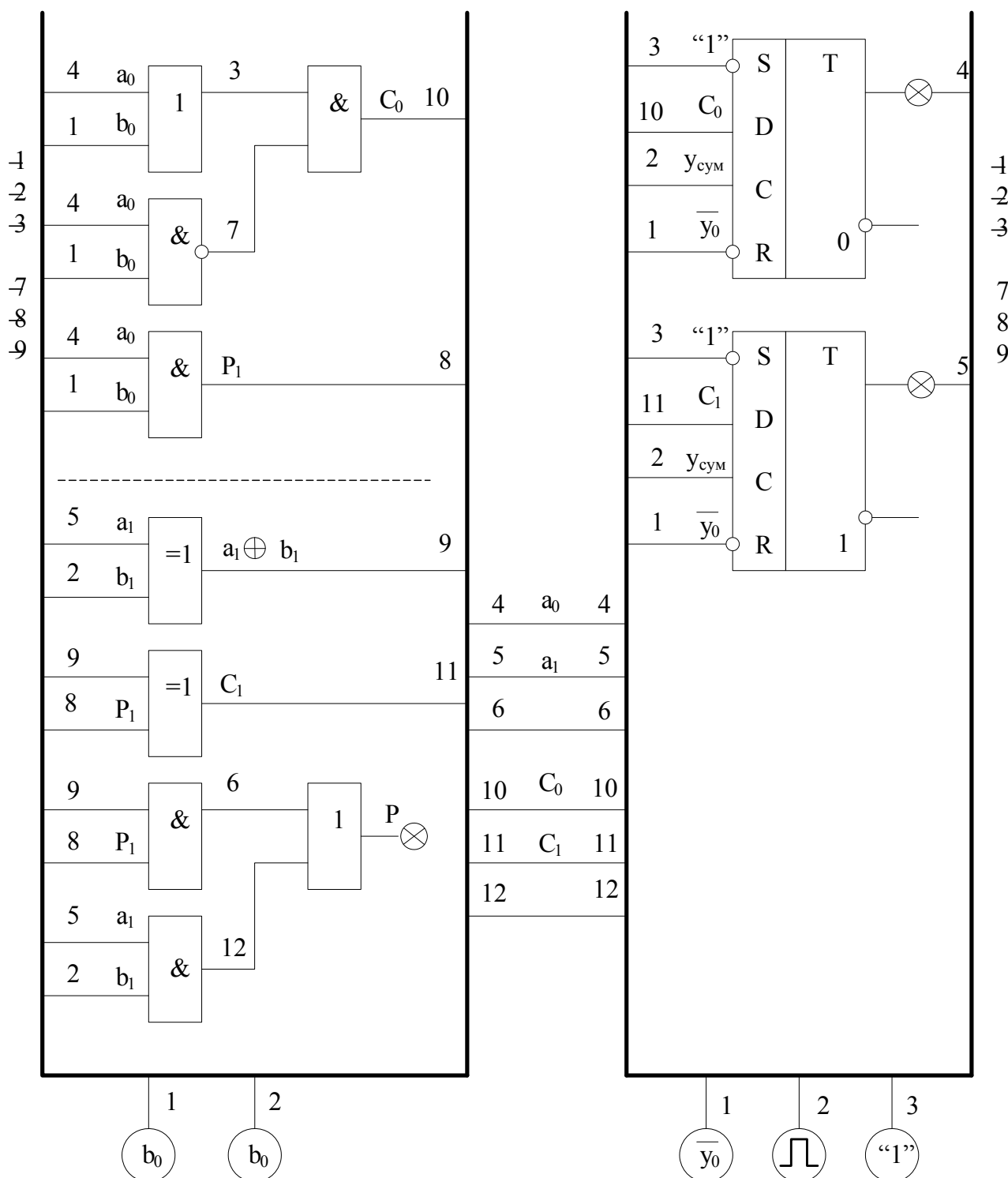


Рисунок 8.11 – Схема двухразрядного параллельного накапливающего сумматора на макете «Logic-Trigger»

8.6. Контрольные вопросы

8.6.1. Чем отличается комбинационный сумматор от накапливающего, последовательный от параллельного?

8.6.2. Какова минимальная задержка сигнала переноса в одноразрядном двоичном сумматоре?

8.6.3. Что такое полусумматор? Как строится полный одноразрядный сумматор из полусумматоров?

8.6.4. Чем определяется время суммирования в параллельном одноразрядном сумматоре? Какие вы знаете способы уменьшения этого времени?

8.6.5. Как с помощью сумматора выполнить операцию вычитания?

9. ЛАБОРАТОРНАЯ РАБОТА № 9. РАСПРЕДЕЛИТЕЛИ СИГНАЛОВ

9.1. Цель работы

Изучить способы построения распределителей сигналов, приобрести навыки проектирования и наладки схем этого класса.

9.2. Краткие теоретические сведения

Распределитель сигналов (РС) – это операционный элемент, вырабатывающий выходные сигналы $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$, принимающие единичные значения в моменты времени $t_0, t_1, \dots, t_{n-1}$ соответственно. Специфика применения распределителей сигналов состоит в том, что они обычно используются для построения управляющих, а не операционных автоматов. На РС удобно строить запоминающую часть автомата, работающего по линейной или легко линеаризуемой микропрограмме. При этом с выходов α_i ($i=0, 1, 2, \dots, n-1$) РС снимаются сигналы состояний автомата. Выход α_{n-1} может использоваться для перевода (сброса) РС в состояние α_0 . РС может быть переведен в состояние α_0 и специальным управляющим сигналом y_0 . Состояния РС сменяются последовательно (единица переходит на очередной выход) по синхросигналу c (рисунок 9.1).

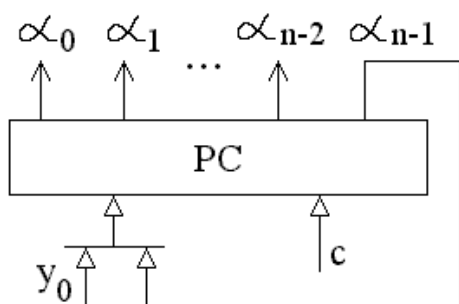


Рисунок 9.1 — Обозначение распределителя сигналов на структурных схемах

Распределитель сигналов может быть реализован как композиция счетчика и дешифратора (рисунок 9.2,а), а также при помощи регистра сдвига (рисунок 9.2,б).

Синхросигнал c инициирует микрооперацию счета (увеличение значения счетчика на единицу) – рисунок 9.2,а. Позиционный код состояния счетчика преобразуется дешифратором в унитарный код, расположение единицы в котором

соответствует коду в счетчике. Счетчик может быть сброшен в ноль сигналом $y_{сбр}$ или сигналом на $(n-1)$ -ом выходе дешифратора. Сигнал, равный дизъюнкции указанных сигналов, должен быть подан на вход сброса (R) счетчика.

Примечание. Если $n=2^k$, то после значения $(n-1)$ нулевое значение на счетчике устанавливается автоматически.

При реализации РС на основе регистра, хранящего унитарный код (рисунок 9.2,б), синхросигнал c инициирует микрооперацию сдвига в регистре. При этом единица сдвигается на очередной выход регистра, а значение освобождающегося нулевого разряда регистра доопределяется значением $(n-1)$ -го разряда (организован циклический сдвиг). Таким образом, обеспечивается автоматический переход регистра из состояния α_{n-1} в состояние α_0 . Управляющий сигнал $y_{загр}$ позволяет установить РС в состояние α_0 путем загрузки в регистр кода $100\dots 0$. Состояния РС иногда называют тактами, а сам РС – распределителем тактов, хотя это название не очень удачно.

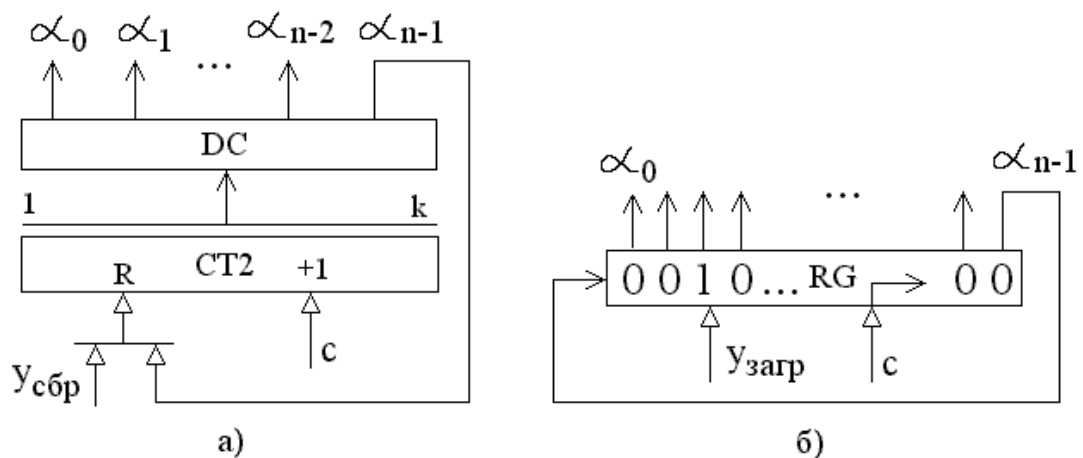


Рисунок 9.2 — Структура распределителя сигналов:
а) на основе счетчика и дешифратора ($k = \lceil \log_2 n \rceil$),
б) на основе регистра сдвига

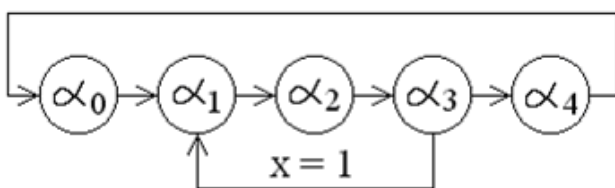


Рисунок 9.3 — Последовательность смены тактов управляемого РС

Иногда требуется в зависимости от значения некоторого осведомительного сигнала x изменять естественную последовательность смены состояний (тактов). Тогда строят управляемый РС. Например, граф, изображенный на рисунке 9.3 задает распределитель

Элемент задержки на рисунке 9.4 должен обеспечивать задержку сигнала φ_1 относительно сигнала α_3 на длину синхроимпульса, чтобы распределитель находился в состоянии α_3 в течение целого такта, а переход в состояние α_1 произошел в следующем такте. В начальное состояние распределитель переводится либо управляющим сигналом y_0 , либо активным уровнем сигнала на выходе α_5 .

9.3. Содержание работы

В процессе выполнения лабораторной работы необходимо разработать, собрать и отладить схему управляемого распределителя, формирующего заданную графом последовательность тактовых сигналов. Распределитель сигналов строится на основе счетчика и дешифратора. В качестве примера на рисунке 9.6 изображена принципиальная схема распределителя сигналов, работающего в соответствии с графом смены состояний, приведенном на рисунке 9.5.

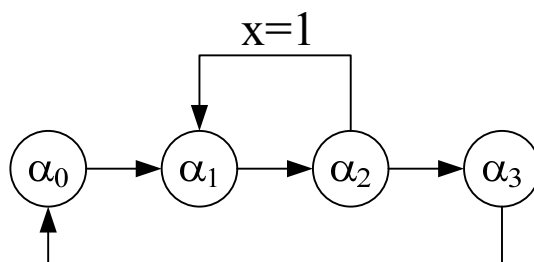


Рисунок 9.5 – Граф смены состояний распределителя сигналов

9.4. Варианты заданий

Варианты заданий приведены на рисунке 9.7.

9.5. Внеаудиторная подготовка

9.5.1. Изучить функциональное описание и способы реализации распределителей сигналов: [2], стр. 261-267.

9.5.2. Синтезировать схему распределителя сигналов в соответствии с вариантом задания. Распределитель сигналов построить на основе счетчика и дешифратора. Суммирующий счетчик с последовательным переносом реализовать на D-триггерах, имеющих в составе макета «Trigger». Для построения дешифратора использовать логические элементы макета «Logic». Для обеспечения задержки сигнала φ_i (переводит счетчик в режим параллельной загрузки) относительно сигнала α_j (состояние, после которого

надо организовать разветвление) на длину синхроимпульса, удобно использовать D-триггер, срабатывающий по инверсному значению синхросигнала, если счетчик срабатывает по прямому значению синхросигнала.

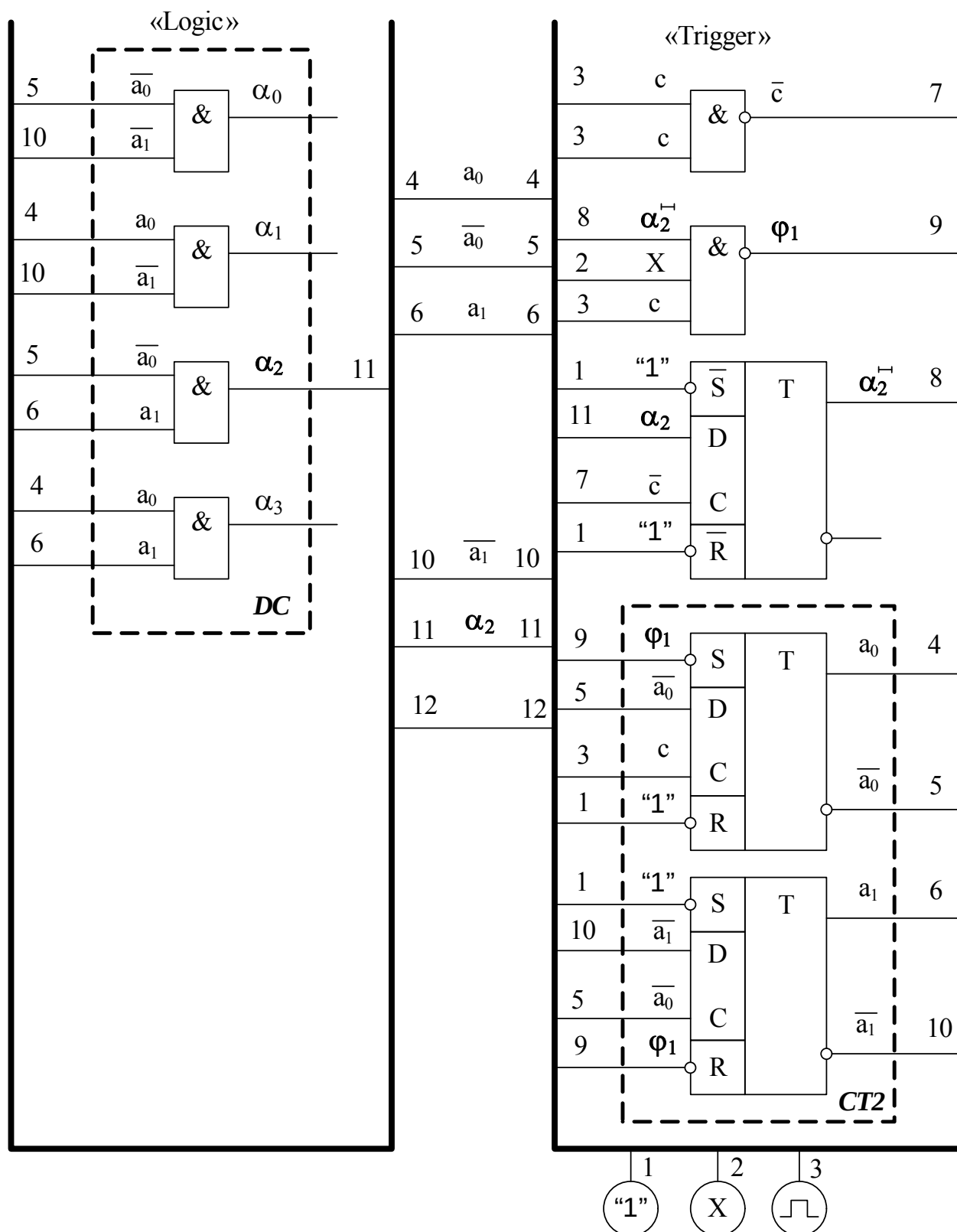


Рисунок 9.6 – Принципиальная схема управляемого распределителя сигналов, заданного графом, изображенным на рисунке 9.5.

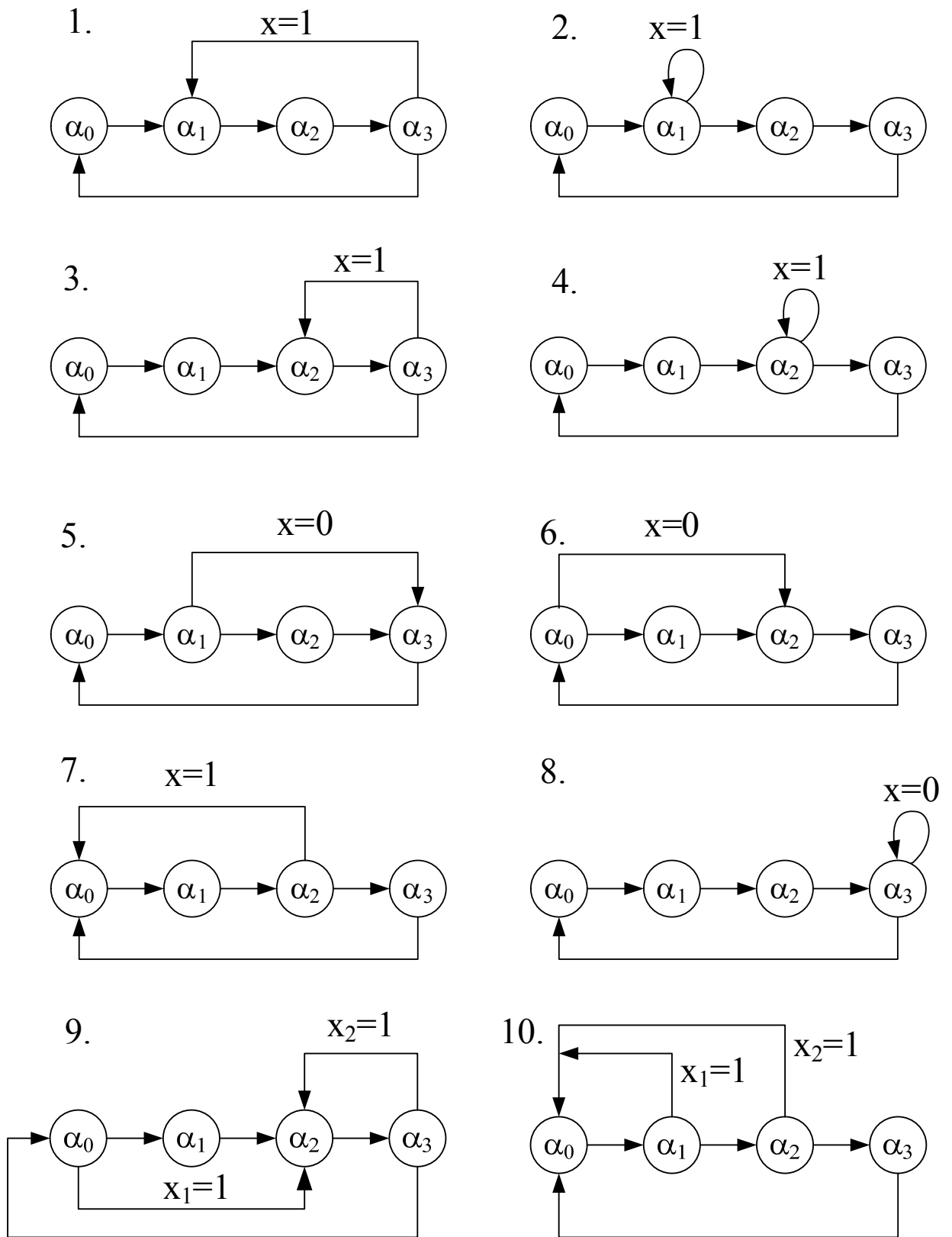


Рисунок 9.7 – Варианты заданий

9.6. Выполнение работы в лаборатории

9.6.1. Собрать схему распределителя сигналов. Значение осведомительного сигнала задать при помощи кнопки.

9.6.2. Отладить схему в статическом режиме, подавая синхросигналы с выхода генератора одиночных импульсов и анализируя состояние схемы с помощью индикаторных лампочек.

9.6.3. Снять с помощью двухлучевого осциллографа временную диаграмму распределителя сигналов, подав на его вход синхросигналы с генератора синхроимпульсов. Определить, по какому фронту синхросигнала осуществляется переключение состояния каждого элемента схемы.

9.7. Контрольные вопросы

9.7.1. Дайте функциональное описание распределителя сигналов. Как он реализуется?

9.7.2. Синтезируйте распределитель сигналов в соответствии с индивидуальным заданием на основе циклического сдвигающего регистра.

9.7.3. Для чего используется триггер в цепях управления сменой состояний счетчика?

9.7.4. Где в компьютере используются распределители сигналов?

10. ЛАБОРАТОРНАЯ УСТАНОВКА «LOGIC-TRIGGER» (L-T)

10.1. Назначение установки

Установка L-T предназначена для изучения методов построения логических схем на потенциальных элементах интегрального комплекса 74HG+. Установка состоит из двух блоков (Logic и Trigger, соединенных шиной, состоящей из шести проводов (рисунок 10.1) [4]. Основная часть установки – наборное поле, которое состоит из ряда функциональных и логических элементов. Каждому из этих элементов соответствует условное графическое изображение на лицевой панели установки (рисунок 10.2). Входы и выходы элементов наборного поля выведены локальные коммутационные поля (ЛКП). Схемы устройств собираются с помощью шины путем подключения к ней с помощью ЛКП. Состав наборного поля отдельных блоков:

1. Набор логических (блок Logic) и триггерных (блок Trigger) элементов
2. Кнопочный регистр для задания логических уровней $FIX_1 - FIX_4$
3. Генератор синхроимпульсов (Impuls/Generator режимы работы).
4. Генератор одиночных импульсов (режим Impuls).
5. Локальные коммутационные поля (ЛКП), обеспечивающие подключение к шине элементов схемы.

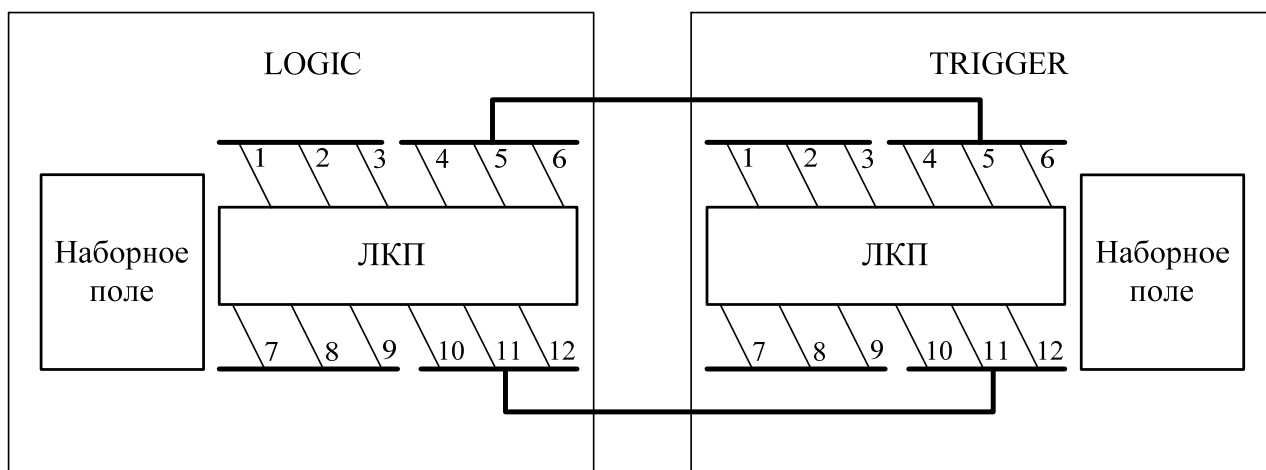


Рисунок 10.1 – Схема установки «LOGIC-TRIGGER» (L-T)

10.2. Логические и триггерные элементы установки

В составе установки имеются логические элементы типа 2И-НЕ, 2И, 3И-НЕ, 3И, 4И-НЕ, 4И, 2ИЛИ-НЕ, 2ИЛИ, 2XOR-НЕ, 2XOR и триггерные элементы типа асинхронных RS и синхронных JK и D-триггеров с установочными инверсными RS-входами. Каждый элемент подключается к шине через ЛКП с помощью перемычки (джампера), которая соединяет вход (выход) с нужным каналом (1,2,..., 12). На рисунке 10.3 приведен пример коммутации одного из выводов логического (триггерного) элемента с линией шины номер 3.

Логические элементы, имеющие как прямой, так и инверсный выход подключаются к ЛКП с помощью разъема. На рисунке 10.3 с помощью разъема и перемычки к ЛКП подключен инверсный выход логического элемента. Каждый вывод элементов наборного поля можно подключить к расположенному рядом светодиодному индикатору.

На незадействованные входы используемых элементов наборного поля должен быть подан потенциал логической «1». Уровень логической «1» может быть выставлен с помощью любой из кнопок «FIX» или кнопки «VAR».

Триггеры типа JK и D, входящие в состав лабораторной установки, используются как элементы памяти.

Триггер типа D может работать в двух режимах: синхронном, при котором управление производится по входу D (при наличии импульсов положительной полярности на входе C), и асинхронном – по RS-входам. Таблица 10.1 – таблица переходов триггера.

Таблица 10.1 – Таблица переходов D-триггера с установочными RS- входами

$C(t)$	$D(t)$	$\bar{R}(t)$	$\bar{S}(t)$	$Q(t+1)$
X	X	0	0	Не определено
X	X	0	1	0
X	X	1	0	1
0	X	1	1	$Q(t)$
1	0	1	1	0
1	1	1	1	1

В таблице 10.1 $C(t)$, $D(t)$, $\bar{R}(t)$, $\bar{S}(t)$ – сигналы на входах C , D , $\bar{R}$, $\bar{S}$ в момент t ; $Q(t)$ – состояние триггера в момент t ; $Q(t+1)$ - состояние триггера в момент $t+1$; X - состояния входов, значения которых не влияют на $Q(t+1)$.

Для синхронного входа t и $t+1$ означают время до и после прихода синхронизирующего импульса. Для организации счетного режима работы триггера необходимо инверсный выход триггера подсоединить к входу D .

Триггер JK-типа также может работать в синхронном и асинхронном режимах. Таблица 10.2 – таблица переходов триггера.

Таблица 10.2 – Таблица переходов JK-триггера с установочными RS- входами

$C(t)$	$J(t)$	$K(t)$	$\bar{R}(t)$	$\bar{S}(t)$	$Q(t+1)$
X	X	X	0	0	Не определено
X	X	X	0	1	0
X	X	X	1	0	1
0	X	X	1	1	$Q(t)$
1	0	0	1	1	$Q(t)$
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	1	1	$\bar{Q}(t)$

В таблице 10.2 $J(t), K(t)$ - сигналы на входах J и K в момент t .

10.3. Кнопочный регистр

В состав наборного поля установки входит 4-разрядный кнопочный регистр (FIX1 – FIX4, рис. 1), который предназначен для задания парафазным кодом логических уровней "1" и "0". Кнопки FIX изменяют состояние (логический уровень) при каждом нажатии.

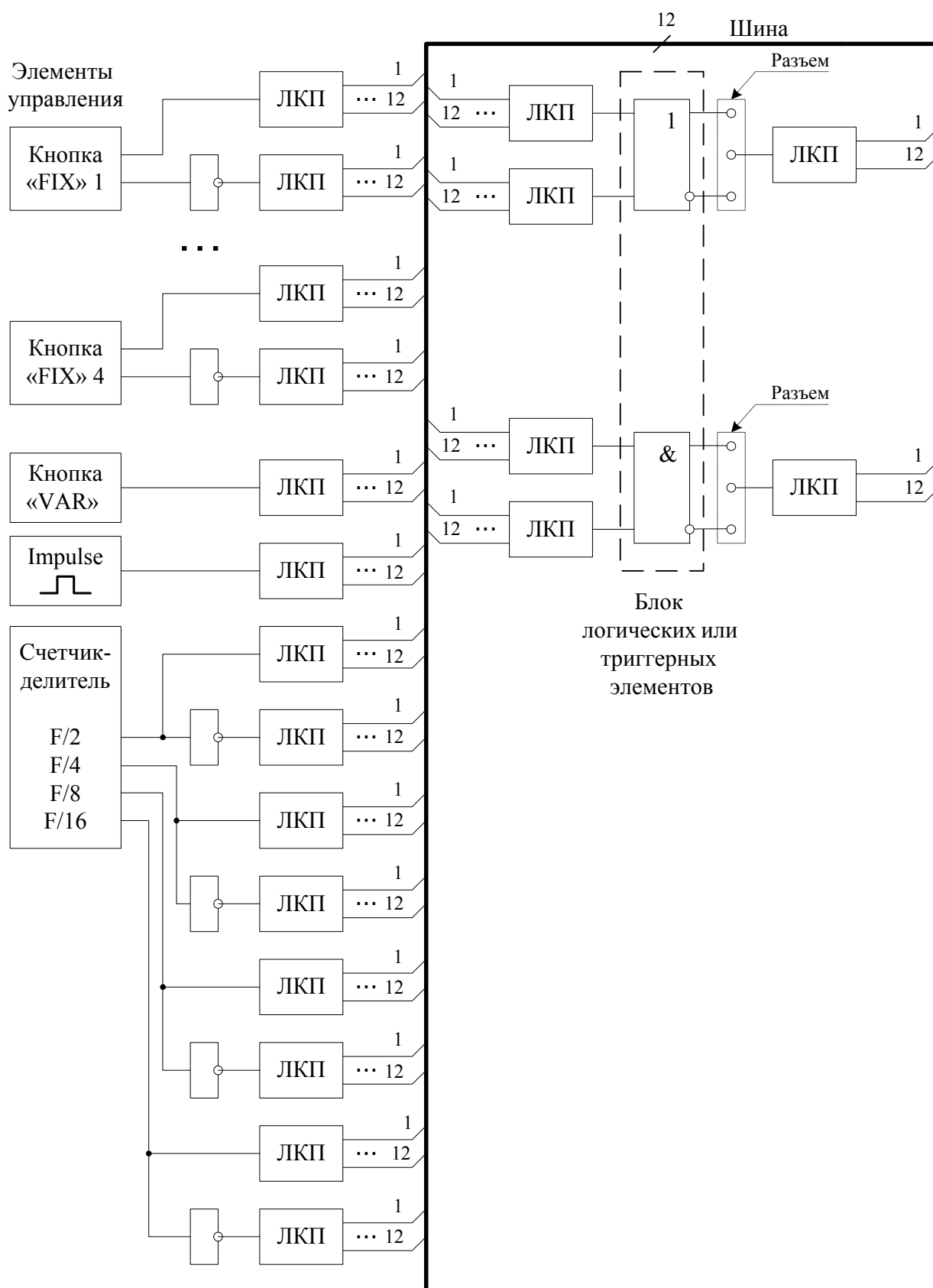


Рисунок 10.2 – Структурная схема блока логических (LOGIC) или триггерных (TRIGGER) элементов с элементами управления

10.4. Генератор синхроимпульсов и одиночных импульсов

Кнопка «Impuls-Generator» управляет одновибратором, формирует одиночный импульс длительностью 20 мс. При длительном нажатии кнопки можно переключиться на режим мультивибратора (генератор). Генератор формирует прямоугольные импульсы, которые подаются на счетчик «Counter Output» который работает как делитель частоты. На коммутационных полях F/2, F/4, F/8, F/16 с помощью перемычек устанавливается связь с необходимой линией шины.

10.5. Порядок работы на установке

10.5.1. Произвести коммутацию элементов установки в соответствии с заданной схемой и описанием лабораторной работы.

10.5.2. Получить разрешение преподавателя на проведение эксперимента. Включить питание.

10.5.3. При работе с осциллографом необходимо соединить корпус осциллографа с корпусом установки (гнезда "_|_"), установить на осциллографе режим внешней синхронизации. Сигнальный вход осциллографа соединить с нужной точкой схемы. Коммутация производится специальными шнурами.

10.5.4. После выполнения работы выключить питание, затем разобрать схему. Коммутационные перемычки собрать в специальный контейнер.

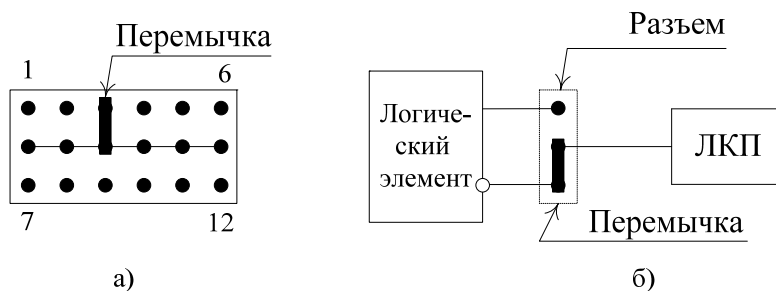


Рисунок 10.3 – Элементы коммутации:

а) локальное коммутационное поле (ЛКП);

б) разъем на выходе логического элемента

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Апраксин Ю.К. Основы теории и проектирования цифровых автоматов: Учеб. пособие для вузов/Ю.К.Апраксин. – Севастополь: Изд-во СевГТУ, 2001.–345 с.
2. Бабич Н.П. Компьютерная схемотехника. Методы построения и проектирования: учеб. пособие/ Н.П. Бабич, И.А. Жуков. – К.: МК-Пресс, 2004. – 576 с.
3. Схемотехника электронных систем. Цифровые устройства/ В.И. Бойко, А.Н. Гуржий, В.Я Жуйков и др. – СПб.: БХВ-Петербург, 2004. – 512 с.
4. Цифровая электроника: методические указания к лабораторным работам. – Хмельницкий: Open System, 2004. – 98 с.
5. Шило В.Л. Популярныe цифровые микросхемы: справочник/ В.Л. Шило. – М: Радио и связь, 1988. – 352 с.

СИНТЕЗ ОПЕРАЦИОННЫХ ЭЛЕМЕНТОВ

*Методические указания
к циклу лабораторных работ по дисциплине*

Составители: **Апраксин** Юрий Константинович,
Волкова Татьяна Викторовна

В авторской редакции

Изд. № 81/2020. Объем 4,75 п.л.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»