

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТРАНСПОРТА**

Кафедра автоматики и телемеханики

К. А. БОЧКОВ, Ю. Ф. БЕРЕЗНЯЦКИЙ

АНАЛИЗ ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ И СИНТЕЗ ДИСКРЕТНЫХ УСТРОЙСТВ

**Лабораторный практикум
по дисциплине «Теория дискретных устройств»**

Часть III

Гомель 2001

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТРАНСПОРТА

Кафедра автоматики и телемеханики

К. А. БОЧКОВ, Ю. Ф. БЕРЕЗНЯЦКИЙ

АНАЛИЗ ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ И СИНТЕЗ ДИСКРЕТНЫХ УСТРОЙСТВ

Лабораторный практикум
по дисциплине «Теория дискретных устройств»

Часть III

Одобен методическими комиссиями
факультета безотрывного обучения и электротехнического факультета

Гомель 2001

УДК 656.25:681.32 (075.8)
Б 866

Бочков К. А., Березняцкий Ю. Ф.

Б 866 Анализ функциональной структуры и синтез дискретных устройств: Лабораторный практикум по дисциплине «Теория дискретных устройств». Ч. III / Белорус. гос. ун-т трансп. – Гомель: БелГУТ, 2001. – 38 с.

Рассматриваются вопросы анализа функциональной структуры и синтеза дискретных устройств с применением программы «Электронная лаборатория».

Предназначен для студентов третьего курса факультета безотрывного обучения и студентов второго курса электротехнического факультета, обучающихся по специальности Т 11.05 «Автоматика, телемеханика и связь на транспорте».

Р е ц е н з е н т – канд.техн.наук, доцент кафедры «Микропроцессорная техника и информационно-управляющие системы»
С. Н. Харлап

ВВЕДЕНИЕ

Целью данного цикла лабораторных работ является изучение принципов построения, логической структуры и функциональных особенностей дискретных устройств на базе программы «Электронная лаборатория» (*“Electronics Workbench”*) фирмы *“Interactive Image Technologies”*, а также получение практических навыков анализа и синтеза дискретных устройств различного назначения.

В третьей части лабораторного практикума рассмотрены вопросы синтеза и анализа регистров, счетчиков а также надежных комбинационных схем с проверкой правильности их функционирования.

Лабораторная работа № 1

ИЗУЧЕНИЕ СЧЕТЧИКОВ ИМПУЛЬСОВ

Ц е л ь р а б о т ы. Изучить назначение, структуры, принцип действия и особенности счетчиков импульсов различных видов на базе программы «Электронная лаборатория».

1 Сведения из теории

Счетчиком называется устройство, подсчитывающее число импульсов и представляющее собой последовательное устройство с одним двоичным входом и определенным числом внутренних состояний, отождествляемых с некоторым числовым кодом. Значение числового кода является одновременно выходным словом счетчика, представляющим результат счета. Возможна более широкая трактовка счетчика как генератора числовых кодов (например, в программных устройствах, в том числе в ЭВМ).

В устройствах цифровой обработки информации счетчики широко применяют как самостоятельные изделия в качестве компонентов более сложных функциональных устройств: в счетных схемах, схемах образования

и измерения временных интервалов, распределителях импульсов, в программных устройствах и др.

В счетчиках используют числовые коды с различными основаниями счета m . Наиболее часто применяют двоичные ($m = 2$) и десятичные ($m = 10$) счетчики, а также восьмеричные ($m = 8$) и шестнадцатеричные ($m = 16$). Максимальное число импульсов, которое может сосчитать счетчик, называется *коэффициентом (модулем) счета* $K_{сч}$. Счетчики могут быть одноразрядными, если $K_{сч}$ не превышает основание счета m , многоразрядными, если $K_{сч} > m$. В многоразрядном счетчике выходное слово будет представлять n -разрядное число, где n – число разрядов счетчика. При этом многоразрядное число будет отображаться комбинацией состояний одноразрядных счетчиков: $N = N_0 + mN_1 + m^2N_2 + \dots + m^{n-1}N_{n-1}$, где $0 \leq i \leq n-1$ – показание (цифра) i -го разряда. Коэффициент счета многоразрядного счетчика $K_{сч} = m^n$.

Чтобы отображать все символы числового кода, каждый разряд счетчика должен иметь столько же состояний, сколько цифр в используемой системе счисления. В счетчиках *статического типа* состояния изображаются потенциалами на его выходах, число которых равно числу состояний. Состояние каждого разряда может отображаться логической единицей (прямые выходы) или логическим нулем (инверсные выходы) на соответствующем выходе. На остальных выходах должны быть логические инверсные значения. Выходы счетчика, его состояния и цифры обозначают $0, 1, \dots, m-1$ (рисунок 1.1). Символом счетчиков на схемах служат буквы CT , которые могут дополняться числом, характеризующим коэффициент (модуль) счета.

Состояние $N = 0$ называется начальным. Для перевода счетчика в начальное состояние в нем предусматривается установочный вход R . Операция «установка 0» заключается в подаче управляющего сигнала на вход R .

Если число входных импульсов превышает коэффициент счета, то счетчик переполняется и затем возвращается в исходное состояние. Таким образом, коэффициент счета характеризует число импульсов, доступных счету за один цикл. После каждого цикла счета на последнем выходе возникает перепад напряжения, чем объясняется второе функциональное назначение счетчиков: деление числа входных импульсов. Если входной сигнал периодический с частотой $F_{вх}$, частота выходного сигнала $F_{вых} = F_{вх} K_{сч}^{-1}$. Счетчики можно снабжать специальными выходами, указывающими возникновение ситуации переполнения.

Импульсы считаются как в направлении увеличения, так и в направлении уменьшения числового кода.

По направлению счета счетчики делятся на *суммирующие, вычитающие*

и реверсивные.

В суммирующем счетчике каждый очередной импульс на входе увеличивает значение кода на единицу, а в вычитающем счетчике – уменьшает на единицу. Реверсивный счетчик может работать как суммирующий или как вычитающий счетчик в зависимости от управляющих сигналов, определяющих направление счета.

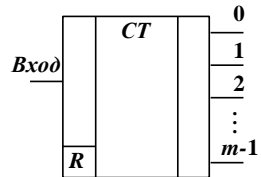


Рисунок 1.1 – Логическая схема одноразрядного счетчика

Современная элементная база позволяет строить самые разнообразные счетчики. Базовыми элементами для счетчиков служат *JK*- и *D*-триггеры. Любой из счетчиков можно реализовать по синхронному или асинхронному принципу. Синхронные (тактируемые) счетчики строят на синхронных триггерах. В них каждый разряд, кроме информационного (счетного) входа *T*, имеет синхронизирующий (тактовый) вход *C*. Асинхронные счетчики строят на асинхронных триггерах или на синхронных, работающих в асинхронном режиме, когда тактирующий вход *C* используется как информационный.

Двоичным счетчиком называется счетчик, у которого выходной код представляет собой число в двоичной системе счисления: $N = N_0 + 2N_1 + 2^2N_2 + \dots + 2^{n-1}N_{n-1}$, где $N_0 - N_{n-1}$ – двоичные числа.

Модуль счета двоичного счетчика равен целой степени числа два – 2^n .

Образовать двоичный счетчик можно, соединяя каскадно счетчики с основанием счета $m = 2$. При $m = 2$ счетчик имеет только два внутренних состояния и может быть реализован на одном триггере со счетным входом (*T*-триггере). Кроме триггера, в его составе могут быть логические схемы для формирования управляющих сигналов и сигналов переноса. Выходная двоичная переменная триггера Q определяет значение числового кода, т. е. $N = Q$.

Триггеры со счетным входом не выпускают в виде самостоятельных изделий, а образуют их из более универсальных *JK*- и *D*-триггеров. Счетный вход у интегральных *JK*-триггеров образуется объединением входов *J* и *K* только в случае синхронного *T*-триггера. В асинхронном режиме роль счетного входа исполняет синхронизирующий вход *C*. Триггер *D*-типа преобразуется в счетный соединением входа *D* с выходом. Счетные импульсы подаются на вход *C*.

Структура связей между разрядами зависит от способа передачи (переноса) информации между разрядами и используемых в счетчике

триггеров, которые могут иметь прямые или инверсные, статические или динамические входы, а также отличаться способом образования T -входа. Применяют два способа передачи информации: последовательный от низшего разряда к высшему и параллельный (одновременно во всех разрядах).

В счетчиках с *последовательным* переносом триггер i -го разряда переключается выходным сигналом триггера $(i - 1)$ -го разряда счетчика. В счетчиках с *параллельным* переносом для формирования сигналов переноса в схему вводят конъюнктуры. На все триггеры счетчика одновременно воздействуют входной (счетный) сигнал и сигналы с выходов других триггеров.

В счетчиках *суммирующего* типа двоичный разряд переполняется при единичном состоянии триггера, и на прямом выходе образуется логический перепад от 1 к 0. Следовательно, при использовании триггеров с прямым динамическим входом в качестве сигнала переноса необходимо применять инверсный выход триггера. В счетчиках *вычитающего* типа разряд переполняется при нулевом состоянии триггера, и в схеме переноса необходимо использовать прямой выход.

Проиллюстрируем работу двоичного счетчика на T -триггерах (рисунок 1.2, *а*) для случая $n = 3$, принимая в качестве числового кода комбинации логических сигналов $Q3 Q2 Q1$ на прямых выходах триггеров. Пусть в исходном состоянии $N = 0$, т. е. все триггеры находятся в состоянии логического нуля и выходное слово 000. Для связи между разрядами использованы инверсные выходы \bar{Q} , поэтому на информационных входах второго и третьего разрядов – логические единицы.

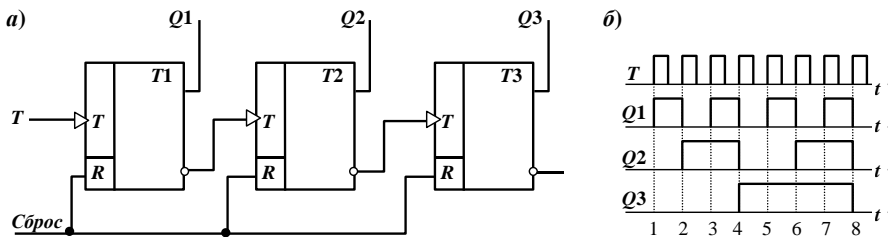


Рисунок 1.2 – Логическая схема двоичного асинхронного счетчика на T -триггерах (*а*) и временные диаграммы его работы (*б*)

Первый импульс (логический перепад от 0 к 1) переключает триггер $T1$ в состояние 1, на его инверсном выходе возникает перепад от 1 к 0 (рисунок 1.2, *б*). Триггер $T2$ (а следовательно, и $T3$) своего состояния не меняет. Выходной код принимает значения 001, приведенные в таблице 1.1. На входе $T2$ возникает состояние логического нуля, а на входе $T3$ сохраняется

логическая единица. Второй импульс возвратит первый разряд в состояние нуль. На инверсном выходе $T1$ сформируется сигнал переноса (логический перепад от 0 к 1), и $T2$ переключится в единичное состояние. На входе $T3$ будет логический перепад от 1 к 0, и он сохранит свое значение. Выходное слово примет вид 010.

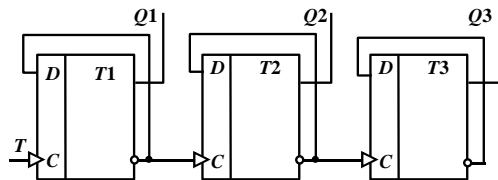
Т а б л и ц а 1.1 – Таблица состояний суммирующего счетчика

Число импульсов	Состояние счетчика	Числовой код		
		Q3	Q2	Q1
0	0	0	0	0
1	1	0	0	1
2	2	0	1	0
3	3	0	1	1
4	4	1	0	0
5	5	1	0	1
6	6	1	1	0
7	7	1	1	1
8	0	0	0	0

Дальнейшая работа протекает аналогично. Третий импульс вновь переключит $T1$, но $T2$ сохранит значение (код 011). На четвертом импульсе $T1$ и $T2$ переходят в состояние 0, при этом в цепи связи между вторым и третьим разрядами сформируется сигнал переноса, и $T3$ переключится в состояние 1 (код 100). После отсчета семи импульсов (код 111) схема вернется в исходное состояние 000.

Реализовать такой счетчик можно также и на обычных D -триггерах с прямым счетным входом (рисунок 1.3).

Рисунок 1.3 – Логическая схема двоичного асинхронного счетчика на D -триггерах



Для перехода к вычитающему счетчику достаточно в цепи переноса заменить инвертирующий выход прямым. Работу вычитающего счетчика иллюстрирует таблица 1.2.

Т а б л и ц а 1.2 – Таблица состояний вычитающего счетчика

Число импульсов	Состояние счетчика	Числовой код		
		Q3	Q2	Q1
0	7	1	1	1
1	6	1	1	0
2	5	1	0	1

3	4	1	0	0
4	3	0	1	1
5	2	0	1	0
6	1	0	0	1
7	0	0	0	0
8	7	1	1	1

В синхронных счетчиках (рисунок 1.4) счетный сигнал T одновременно подается на входы синхронизации триггеров всех разрядов.

В таких триггерах при $C = 1$ предварительно запоминается информация, а ввод ее осуществляется при $C = 0$, т. е. в структурном отношении они эквивалентны триггеру с инверсным счетным входом. Поэтому в случае суммирующего счетчика в цепи переноса должен использоваться прямой выход.

Временная диаграмма синхронного счетчика совпадает с диаграммой асинхронного счетчика, если не учитывать временные сдвиги переключения старших разрядов в асинхронном режиме.

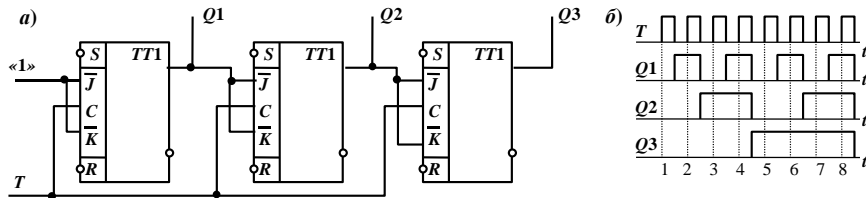


Рисунок 1.4 – Логическая схема синхронного счетчика на JK -триггерах с MS -структурой (а) и временная диаграмма его работы (б)

Временные диаграммы двоичных счетчиков (рисунки 1.2, б и 1.4, б) показывают, что они одновременно являются делителями частоты. Действительно, если частота импульсов на входе счетчика F , то на выходе первого разряда частота $F_1 = F/2$, на выходе второго разряда – $F_2 = F/4$, на выходе третьего разряда – $F_3 = F/8$ и т. д.

Главное достоинство данных счетчиков (с последовательным переносом) – простота схемы. Сигнал переноса с этих счетчиков передается от одного разряда к другому в результате последовательного срабатывания триггеров одного за другим. Это приводит к возникновению временно ложных выходных слов и создает опасность появления помехи на выходах логических схем, подключенных к счетчику. Кроме того, снижается быстроедействие счетчика.

Ускорение переноса в счетчиках основано на применении логических элементов, обеспечивающих одновременное формирование сигналов переноса для всех разрядов. Единица передается в старший разряд при переполнении младших разрядов и, следовательно, сигнал переноса T_j на

входе j -го разряда конъюнктивно связан с состоянием триггеров предшествующих разрядов 1, 2, ..., $(j - 1)$: $T_j = TQ_1Q_2 \dots Q_{j-1}$.

Реализация параллельного переноса сводится к реализации вышеуказанных функциональных зависимостей для каждого разряда счетчика. Для этого применяют TV -триггеры (рисунок 1.5), на T -входы которых подаются счетные импульсы, а на V -входы – сигналы переноса, формируемые согласно формуле для T_j . Триггеры, на V -входе которых имеется сигнал переноса, одновременно переключаются с приходом очередного счетного импульса, и в счетчике устанавливается новое состояние.

Ускорение переноса усложняет структуры счетчика особенно при большом числе разрядов. Поэтому используют комбинированные счетчики, в которых каскадное соединение сочетается с параллельным переносом в каждом каскаде.

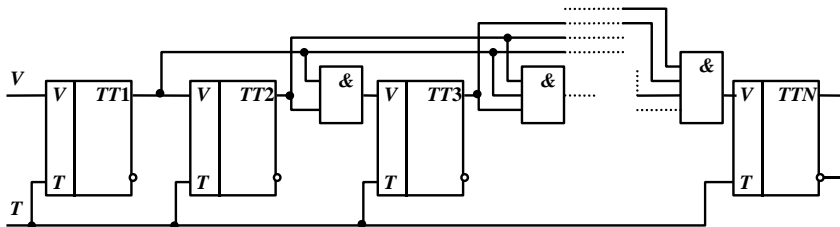


Рисунок 1.5 – Логическая схема двоичного счетчика с параллельным переносом на TV -триггерах

Кольцевые счетчики применяют при основании системы счета больше двух ($m > 2$). Число триггеров в кольцевом счетчике равно числу состояний (основанию счета). В каждом состоянии в нем существует только одна единица, которая по мере поступления счетных импульсов на вход перемещается из одного триггера в другой. Цикл работы кольцевого счетчика равен числу триггеров. В конце цикла единица из последнего триггера вновь поступает в первый триггер.

Кольцевые счетчики выполняются на D - или JK -триггерах. В первом случае информерционный вход D и оба входа J и K во втором случае соединяют с выходом или с обоими выходами предшествующей ступени, образуя замкнутое кольцо (рисунок 1.6).

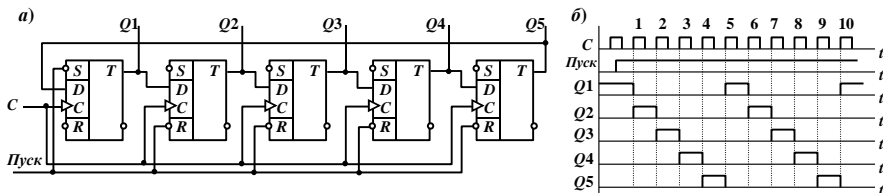


Рисунок 1.6 – Логическая схема пятиразрядного кольцевого счетчика (а) и временная диаграмма его работы (б)

Во время установки схемы в начальное состояние сигналом «Пуск» в первый триггер заносится единица, которая при каждом тактовом импульсе передается в следующую ступень. Остальные триггеры устанавливаются в нуль. Требуемый числовой код образуется непосредственно сигналами Q_1, \dots, Q_5 без дешифратора.

При сбое кольцевых счетчиков в них могут возникать недопустимые состояния (например, несколько единиц в выходном слове), которые не устраняются сами собой. Этот недостаток исключается введением цепи логической коррекции. Так, например, работу счетчика на JK -триггерах (рисунок 1.7) корректирует конъюнктор $DD1$, блокирующий вход первой ступени, до тех пор пока счетчик не будет приведен в начальное состояние. Благодаря этому работа счетчика при любых сбоях восстанавливается к началу очередного цикла.

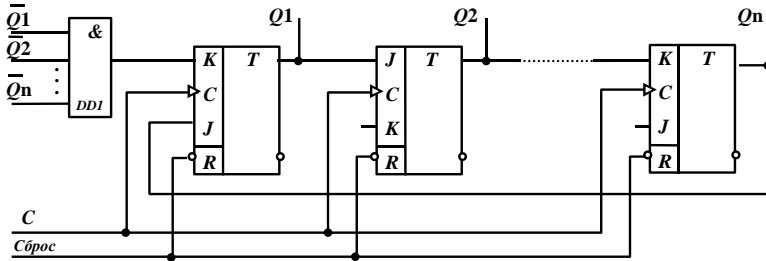


Рисунок 1.7 – Логическая схема счетчика Джонсона с коррекцией состояний

Двоичные счетчики с обратными связями являются наиболее экономичными в отношении расхода триггеров. Например, для реализации десятичного кольцевого счетчика требуется 10 триггеров, а используя двоичные счетчики, 10 цифр можно отобразить при помощи четырех триггеров. Для устройств на дискретных компонентах это обстоятельство может иметь решающее значение.

Двоичные счетчики могут быть обращены в недвоичные с коэффициентом счета $K_{сч} \neq 2^n$ введением дополнительных логических связей. Число двоичных разрядов n выбирают равным числу разрядов коэффициента $K_{сч}$, записанного в виде двоичного числа. Например, для счетчика с модулем $K_{сч} = 10_{10} = 1010_2$ необходимы четыре разряда. При таком выборе число состояний 2^n будет ближайшим числом больше заданного $K_{сч}$. Изменяя естественный порядок счета, дополнительные связи позволяют исключить $2^n - K_{сч}$ лишних состояний.

Наибольшее распространение получили десятичные (декадные) счетчики, считающие в двоично-десятичном коде (рисунок 1.8, а). Чтобы

избежать дополнительных логических элементов, в них использованы триггеры с входной логикой, имеющие несколько J (K) входов, объединенных логической операцией И.

До цифры 7 счетчик (рисунок 1.8, б) работает как асинхронный двоичный. В состоянии 7 (0111) на входы J четвертого разряда поступают две единицы из второго и третьего разрядов, и при счете 8 в него заносится единица.

Логический ноль с инверсного выхода четвертого разряда поступает на вход J второго разряда, удерживая второй и третий разряды в состоянии ноль. Переключение первого разряда при счете 9 подготавливает сброс четвертого разряда, состояние входов которого соответствует комбинации $J = 0, K = 1$. При счете 10 счетчик возвращается в нулевое состояние.

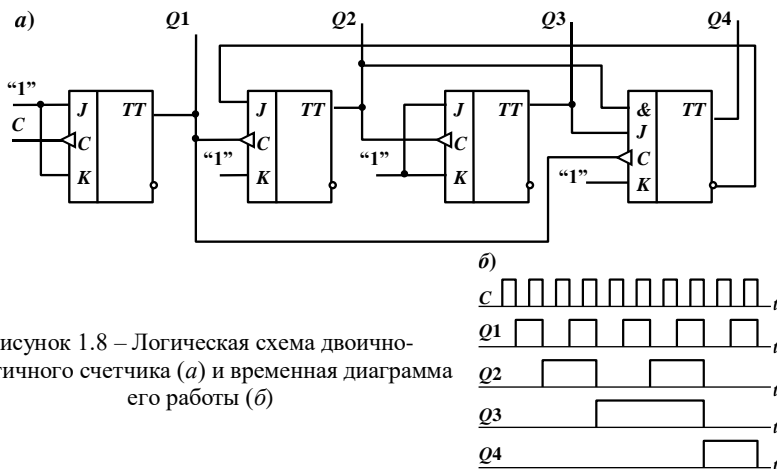


Рисунок 1.8 – Логическая схема двоично-десятичного счетчика (а) и временная диаграмма его работы (б)

В состав многих серий, кроме *декадных счетчиков*, входят также счетчики-делители с разными коэффициентами деления (3, 5 и т. д.). Широко распространены счетчики-делители с последовательным переносом в коде 8421, в схему которых вводится дополнительный логический элемент, исключающий лишние состояния. Счетчик работает как двоичный до достижения некоторого состояния, определяемого коэффициентом счета $K_{сч}$, которое дешифрируется логическим элементом, и на его выходе формируется сигнал сброса. По цепи обратной связи сигнал сброса подается одновременно на R -входы всех триггеров, возвращая счетчик в нулевое состояние.

Дешифрируемая комбинация в таком счетчике совпадает с двоичным кодом коэффициента счета $K_{сч}$ (например, 1010 при $K_{сч} = 10$). Дешифратор представляет собой логический элемент И либо И-НЕ (в случае инверсных

Рисунок 1.10 – Логическая схема реверсивного счетчика

В режиме обратного счета $C_b = 1$ ($C_c = 0$) в цепь переноса вводится инверсный выход. В зависимости от управляющих импульсов складывается или вычитается двоичное число, записанное в счетчике, с числом, представленным в единичной системе, записанным одними единицами, например, 11111 – 5, в виде последовательности импульсов на входе счетчика.

Счетчики повышенного уровня интеграции имеются в составе многих серий микросхем, оформленные как самостоятельные изделия. Многие из них обладают универсальными свойствами и позволяют решать большинство практических задач, связанных с применением счетчиков. Несколько таких микросхем имеются в составе серий ТТЛ и КМОП: К155ИЕ2, К155Е4, К155ИЕ6, К155ИЕ7, К155ИЕ8, К155ИЕ9 в составе ТТЛ-серии К155 широкого назначения, 564ИЕ16, 564ИЕ8, 564ИЕ9, 564ИЕ10, 564ИЕ11, 564ИЕ14 в составе КМОП-серии 564.

Универсальность счетчиков ТТЛ с последовательным переносом (К155ИЕ5, К155ИЕ4, К155ИЕ2) обеспечивается возможностью изменения их логической структуры. Каждый из счетчиков состоит из четырех *JK*-триггеров, разбитых на две секции (рисунок 1.11, а): три соединены в последовательную цепочку (элемент *DD1*), один выполнен самостоятельно (элемент *DD2*). Каждая из микросхем имеет логический элемент (*DD3*) для перевода всех триггеров в нулевое состояние. При каскадном соединении обеих секций и использовании элемента установки микросхемы позволяют организовать самые различные коды счета (например, 8421, 6421, 5421 и т. д.) и коэффициент счета (до 16).

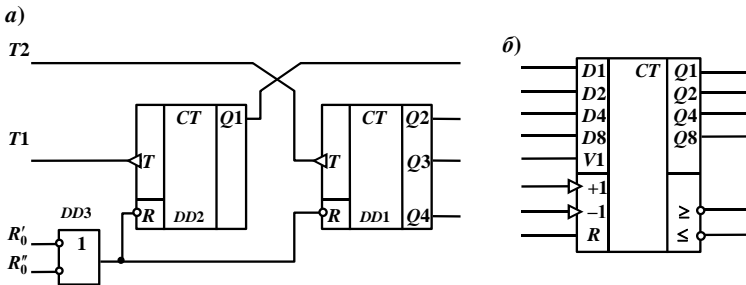


Рисунок 1.11 – Логические схемы интегральных счетчиков с последовательным (а) и параллельным (б) переносами

Несколько иные принципы применяют в счетчиках ТТЛ с параллельным переносом (К155ИЕ9, К155ИЕ6, К155ИЕ7). Их универсальность обеспечивается за счет использования предварительной записи информации. Для этого каждый из четырех разрядов счетчика (*Q1 Q2 Q4 Q8*) имеет вход

предварительной установки (входы $D1 D2 D4 D8$) (рисунок 1.11, б). Желаемое состояние записывается по сигналу ввода информации (вход $V1$).

Счетчики имеют выход переноса \geq и выход заема \leq . Наличие входов установки и выходов переноса позволяет применять их в качестве программируемых счетчиков делителей, если сигнал с выхода переноса подать на вход разрешения записи. Коэффициент счета в этом случае будет определяться по формуле $K_{сч} = K_{сч} - N_0$, и его можно программировать, изменяя число N_0 , записываемое в счетчик через входы D .

У реверсивных счетчиков предусматриваются два счетных входа: $+1$ и -1 . В режиме суммирования последовательность входных импульсов подается на счетный вход $+1$, а сигнал переноса формируется на выходе \geq во время перехода из последнего состояния в нулевое. В режиме вычитания используется счетный вход -1 и сигнал переноса на выходе \leq при изменении состояния 000. Аналогичные принципы построения применяют в КМОП-счетчиках.

2 Порядок выполнения работы

1 Изучить сведения из теории по данным методическим указаниям и литературе [1, с. 67 – 76; 3, с. 162 – 196; 4, с. 211 – 219].

2 Используя программу “Электронная лаборатория”, собрать и смоделировать работу схемы счётчика, указанного преподавателем в таблице 1.3.

Таблица 1.3 – Варианты заданий

Вариант	Номер рисунка	Количество разрядов счетчика	Вариант	Номер рисунка	Количество разрядов счетчика
1	1.2	5	12	1.4	5
2	1.3	5	13	1.5	5
3	1.4	4	14	1.6	6
4	1.5	4	15	1.7	4
5	1.6	5	16	1.7	3
6	1.7	5	17	1.9	5
7	1.8	4	18	1.10	6
8	1.9	4	19	1.2	7
9	1.10	5	20	1.3	7
10	1.2	6	21	1.4	6
11	1.3	6	22	1.5	6

Содержание отчета

Наименование и цель работы; условное обозначение счетчиков; рабочая

схема заданного счетчика и его временные диаграммы (полученные при помощи логического анализатора (или осциллографа) программы «Электронная лаборатория»); ответы на заданные преподавателем контрольные вопросы; выводы по работе.

Контрольные вопросы

- 1 Опишите принцип действия вычитающего счетчика.
- 2 Что называется счетчиком?
- 3 Какие методы синтеза счетчиков Вам известны?
- 4 В чем заключается более широкая трактовка счетчика?
- 5 Каковы области применения счетчиков?
- 6 Какие типы счетчиков Вам известны?
- 7 Особенности счетчиков с обратными связями.
- 8 Как определяется частота деления счетчика?
- 9 Каким образом по схемному исполнению можно отличить суммирующий и вычитающий счетчики?
- 10 Каковы особенности кольцевых счетчиков?
- 11 Отличия счетчиков с последовательным и параллельным переносом.
- 12 Преимущества и недостатки счетчиков с последовательным переносом.
- 13 Опишите принцип действия суммирующего счетчика.
- 14 Особенности реверсивных счетчиков.
- 15 Как задается счет чисел не равных 2^n ?
- 16 Приведите примеры микросхем счетчиков промышленного исполнения.

Лабораторная работа № 2

ИЗУЧЕНИЕ РЕГИСТРОВ

Ц е л ь р а б о т ы. Изучить назначение, структуру, функциональные особенности и принцип действия регистров.

1 Сведения из теории

1.1 Определение, назначение и классификация регистров

Регистрами называются устройства, выполняющие функции приема, хранения, логической обработки и передачи информации. Информация в регистре хранится в виде числа (слова), представленного комбинацией сигналов "0" и "1". Каждому разряду числа, записываемому в регистр, соответствует свой разряд регистра. Обычно регистры выполняются на

триггерах. Триггер с номером i является i -м разрядом регистра и представляет i -й разряд двоичного слова $x_1, x_2, x_3, \dots, x_n$, где x_i ($i = 1, 2, 3, \dots, n$) – двоичное значение переменной данного разряда, равное 0 или 1. Число возможных состояний n -разрядного регистра равно 2^n . Для сокращения записи двоичного числа, отображающего состояние регистра, иногда используется восьмеричное и шестнадцатеричное представление двоичных чисел. При этом регистр разбивается условно на трех- или четырехразрядные подрегистры. На рисунке 2.1 изображен регистр на триггерах и его обозначение на функциональных схемах.

В регистре могут выполняться следующие основные операции над n -разрядным входным кодом $x_1, x_2, x_3, \dots, x_n$:

- а) предварительная установка регистра;
- б) прием и хранение кода числа $x_1, x_2, x_3, \dots, x_n$ из другого устройства;
- в) передача кода числа из регистра в другое устройство (в сумматор, в запоминающее устройство, в другой регистр и т. д.);
- г) сдвиг хранимого в регистре кода на один или несколько разрядов вправо или влево;
- д) преобразование последовательного кода в параллельный и наоборот;
- е) поразрядные логические операции.

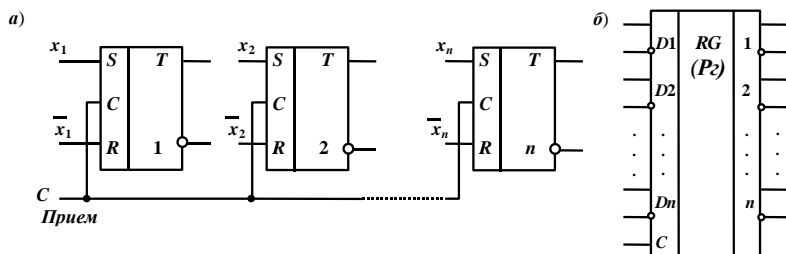


Рисунок 2.1 – Схема параллельного регистра на синхронных RS-триггерах (а) и обозначение регистров на функциональных схемах (б)

Схемы конкретных регистров могут выполнять лишь некоторые из указанных операций. В зависимости от назначения регистра в нем кроме основных элементов (триггеров) применяют различные логические схемы.

Классифицируются регистры по способу приема и выдачи кода числа (*параллельные, последовательные, параллельно-последовательные*), по количеству каналов передачи информации (*парафазные и однофазные*), по способу тактирования (*однотактного и многотактного действия*).

1.2 Выполнение операции установки

Операция установки служит для переключения регистра в фиксированное состояние. Установка регистра производится совокупностью

сигналов управления, поступающих на установочные входы триггеров регистра. Обычно все триггеры регистра перед приемом кода числа устанавливают в положение "0" или "1". Однако не исключается и установка регистра в состояние, отличное от 000...0 или 111...1. Такую операцию часто называют предварительной установкой. При этом установочные входы триггеров объединяются в общую шину (рисунок 2.2), на которую подается управляющий сигнал (Y).

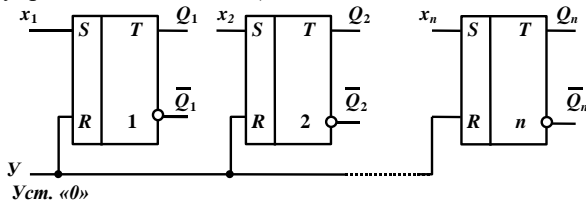


Рисунок 2.2 – Схема предварительной установки регистра

Объединение установочных вводов в общую шину допускает однозначную установку триггеров, т. е. всегда в одно и то же состояние (например, 000...0, см. рисунок 2.2). Для выполнения нескольких операций установки используют логические схемы.

1.3 Прием и выдача кодов в регистрах

Для приема любого кода числа $x_1, x_2, x_3, \dots, x_n$ и его выдачи применяются системы входных и выходных логических схем.

На рисунке 2.3 приведена функциональная схема регистра на асинхронных RS-триггерах, которая позволяет принимать числа, заданные в параллельном коде потенциальными однофазными сигналами, поступающими на входы системы входных схем "И", и осуществлять выдачу этого числа или с выходов Q (прямой код числа), или с выходов \bar{Q} (инверсный код) с помощью элементов 2И-ИЛИ.

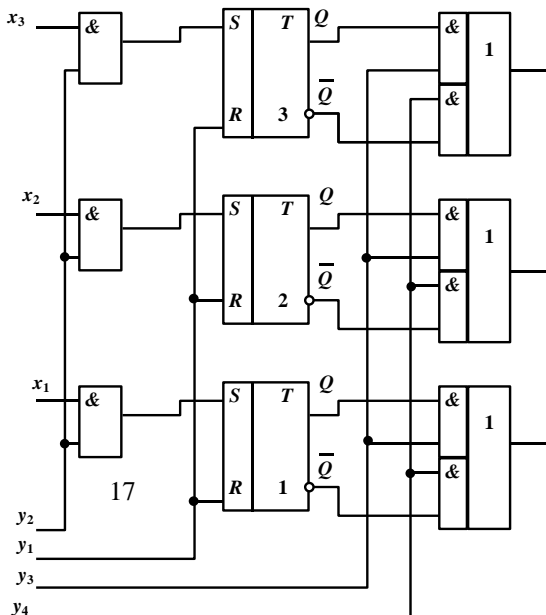


Рисунок 2.3 – Схема

После подачи сигнала V_1 триггеры регистра устанавливаются в состояние 000. Поданный на шину приема числа импульсный сигнал V_2 проходит на единичные (*S*) входы триггеров регистра только в тех разрядах, где $x_i = 1$. Если код числа x_1, x_2, x_3 задан в виде импульсных сигналов, то необходима синхронизация передачи сигналов информационного кода и сигнала приема кода. При использовании синхронных *RS*-триггеров входные логические элементы И исключаются, а функции сигнала V_2 выполняет синхросигнал, подаваемый в триггер. Функциональная схема регистра на синхронных *RS*-триггерах приведена на рисунке 2.1. Здесь прием информации в регистр осуществляется в парафазном коде ($x_1, \overline{x_1}; x_2, \overline{x_2}; x_3, \overline{x_3}$) без предварительной установки и за один такт. При этом на *S*-вход каждого триггера подается код данного разряда числа (x_i), на вход *R* – его инверсия ($\overline{x_i}$).

Выдача числа из регистра может осуществляться как в прямом, так и в инверсном коде. С прямых выходов триггеров регистра снимается прямой код, с инверсных – инверсный. При *парафазной* выдаче числа используются одновременно оба выхода каждого триггера, при *однофазной* – только один из них. Так, в схеме, представленной на рисунке 2.3 для выдачи информации в прямом коде используется сигнал V_3 , а в инверсном – V_4 . При парафазной выдаче прямого и инверсного кодов используются две схемы 2И-ИЛИ на один разряд.

1.4 Межрегистровые передачи

При необходимости передачи кода слова с одного регистра на другой выполняется операция межрегистровой передачи. Она может осуществляться *однофазным* и *парафазным* способами. В схеме на рисунке 2.4 управляющим сигналом V_1 вначале производят предварительную установку регистра *B* в состояние 000 (сброс). Затем управляющим

сигналом Y_2 производится прием в регистр B прямого кода числа, содержащегося в регистре A . Для передачи инверсного кода информация снимается с инверсных выходов триггеров регистра A .

Таким образом, операция *однофазной межрегистровой передачи* в схеме, представленной на рисунке 2.4, осуществляется последовательной подачей двух управляющих сигналов Y_1 и Y_2 , т. е. за два такта. При *парафазной передаче* (рисунок 2.5) не требуется предварительно очищать регистр B , так как его нулевые и единичные входы подключаются к прямым и инверсным выходам триггеров регистра A , и межрегистровая передача выполняется за один такт.

При этом на одном из входов триггера обязательно присутствует сигнал "1", который и устанавливает триггер в нужное состояние независимо от той информации, которая в нем хранилась.

Как правило, операция передачи кода с регистра объединяется с операцией приема этого кода на другой регистр. Это показано на рисунке 2.5, где информация передается с регистра A в регистр B под действием сигнала Y_2 , управляющего приемом информации на регистр.

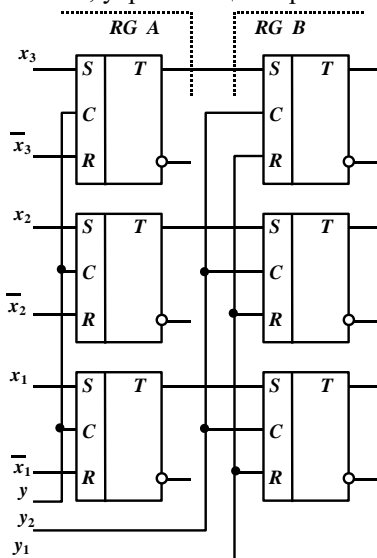


Рисунок 2.4 – Схема однофазной межрегистровой передачи

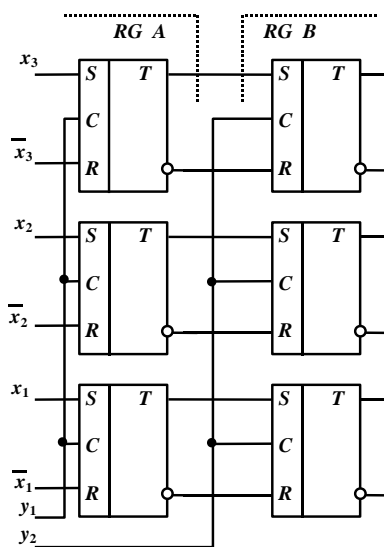


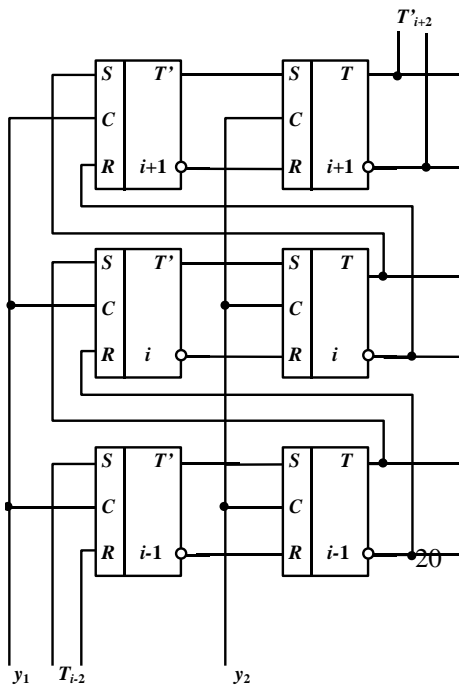
Рисунок 2.5 – Схема парафазной передачи

1.5 Сдвигающие регистры

Сдвигающий регистр служит для хранения кода числа и

выполнения операции сдвига кода. Сдвиг кода – это перемещение в регистре кода числа на какое-либо число разрядов влево или вправо. В этом случае разряды числа, вышедшие из разрядной сетки регистра влево или вправо, теряются, а в освободившиеся при сдвиге разряды регистра записываются нули. Поскольку для позиционных систем счисления вес каждого символа кода определяется его положением в коде, то сдвиг кода числа на один разряд влево относительно принятой разрядной сетки соответствует умножению величины числа на основание системы счисления, а сдвиг вправо – делению при условии, что справа и слева есть свободные разряды и не происходит потеря единиц. Сдвигающий регистр используется также для преобразования параллельного кода в последовательный и наоборот.

Для осуществления операции сдвига вводят специальные цепи сдвига. Структура сдвигающего регистра зависит от типа используемых элементов, от требуемого быстродействия и от ограничений, налагаемых на количество оборудования. На рисунке 2.6 изображена схема парафазного сдвигающего регистра на RS -триггерах. Здесь кроме основных триггеров, выполняющих функции хранения кода числа, имеются дополнительные, буферные или промежуточные триггеры (понятие основных и дополнительных чисто условное). Очевидно, что нельзя записывать в триггер T_i , код числа, хранящийся в T_{i-1} триггере, пока из T_i не считан код в T_{i+1} . Поэтому при построении сдвиговых регистров на одноярусных триггерах необходимо применять в цепях передачи кода элементы задержки при однократном сдвиге или дополнительные (промежуточные триггеры) при двухтактном сдвиге.



В схеме на рисунке 2.6 будем считать триггеры T_{i-1} , T_i , T_{i+1} основными, в которых хранится код числа, подлежащего сдвигу, а триггеры T'_{i-1} , T'_i , T'_{i+1} – дополнительными или промежуточными.

При подаче управляющего сигнала Y_1 код $i - 1$ разряда слова передается в промежуточный триггер T'_{i-1} , а T'_i в T'_{i+1} и т. д. Затем управляющим сигналом Y_2 код $i - 1$ разряда из промежуточного триггера T'_i передается в основной T_i . Аналогично происходит передача

кодов остальных разрядов слова. Таким образом, двумя управляющими сигналами (двумя тактами) производится сдвиг кода слова на один разряд вверх по схеме (рисунок 2.6).

Рисунок 2.6 – Схема сдвигающего регистра

Естественно, что в регистрах, построенных на двухъярусных триггерах, не требуется применение элементов задержки или дополнительных промежуточных триггеров, так как начало (фронт) тактового импульса в таком триггере производит запись в ведущий триггер, а конец тактового импульса (срез) переписывает этот код из ведущего в ведомый, т. е. функции дополнительного триггера в каждом разряде выполняет ведущий триггер, а основного – ведомый, с которого и снимается информация. На рисунке 2.7 изображена схема регистра сдвига вправо на двух- ярусных *JK*-триггерах.

Работает схема следующим образом. Подачей сигнала u_1 все триггеры устанавливаются в состояние "0". Далее производится запись кода числа x_1, x_2, x_3, x_4 , например, 1101, по установочным входам *S*-триггеров. Затем подачей тактовых импульсов u_2 на шину синхронизации *C* производится сдвиг вправо кода числа, записанного в регистр. Состояния триггеров регистра после подачи тактовых импульсов $T_1 - T_4$ сведены в таблицу 2.1.

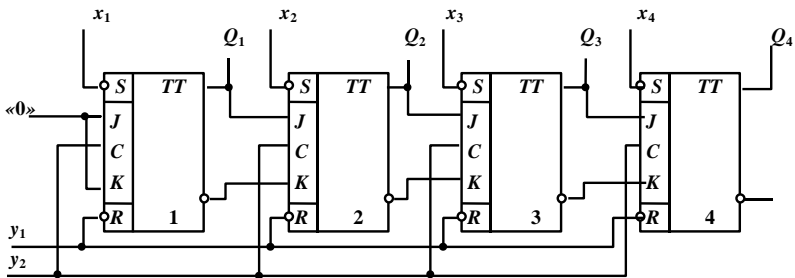


Рисунок 2.7 – Схема регистра сдвига вправо на JK -триггерах

Т а б л и ц а 2.1 – Состояния триггеров регистра

Тактовые импульсы	Выходы триггеров			
	Q_1	Q_2	Q_3	Q_4
T_0	1	1	0	1
T_1	0	1	1	0
T_2	0	0	1	1
T_3	0	0	0	1
T_4	0	0	0	0

Следует отметить, что входы J и K первого триггера заземлены, что соответствует коду "0", который и записывается в первый триггер после прихода каждого тактового импульса.

На рисунке 2.8 приведены временные диаграммы, поясняющие принцип работы схемы.

Существуют схемы реверсивных сдвиговых регистров, которые позволяют производить сдвиг кода числа как вправо, так и влево. Они могут быть одноктактными и двухтактными с однофазными и парафазными цепями передачи, с параллельной и последовательной записью.

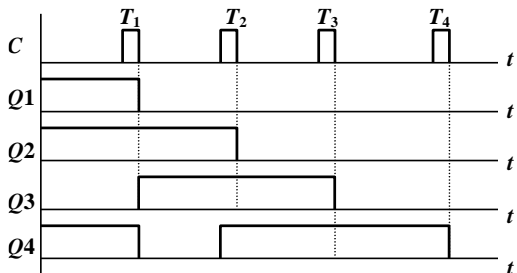


Рисунок 2.8 – Временная диаграмма регистра

2 Порядок выполнения работы

1 Используя программу «Электронная лаборатория», собрать схему регистра, указанного преподавателем в таблице 2.2.

2 Исследовать работу схемы, используя логический анализатор, генератор импульсов и логический преобразователь.

3 Сделать выводы по функционированию исследованной схемы.

Т а б л и ц а 2.2 – **Варианты заданий**

Вариант	Номер рисунка	Количество разрядов регистра	Вариант	Номер рисунка	Количество разрядов регистра
1	2.1	5	12	2.7	4
2	2.3	5	13	2.1	7
3	2.4	4	14	2.3	6
4	2.5	4	15	2.4	3
5	2.6	5	16	2.5	5
6	2.7	5	17	2.6	3
7	2.1	4	18	2.7	3
8	2.3	4	19	2.1	6
9	2.4	5	20	2.3	3
10	2.5	6	21	2.4	6
11	2.6	4	22	2.5	3

Содержание отчета

Наименование и цель работы, схема исследованного регистра, временные диаграммы его работы, ответы на контрольные вопросы, выводы по работе.

Контрольные вопросы

1 По каким признакам классифицируются регистры? Какое назначение имеют регистры?

2 Как выполняется операция установки в регистрах?

3 Как осуществляется прием и выдача кодов в регистрах на синхронных и асинхронных триггерах?

4 Как осуществляется операция межрегистровой передачи на синхронных и асинхронных RS-триггерах?

5 Какие преимущества и недостатки имеет однофазная межрегистровая передача перед парафазной?

6 Какое назначение и применение имеют сдвигающие регистры?

7 Как осуществляется операция сдвига?

8 Какие преимущества и недостатки имеет схема парафазного сдвигающего регистра по сравнению с однофазным?

9 Как работает схема сдвигового регистра, построенная на одноярусных синхронных *RS*-триггерах.

10 Как работает схема сдвигового регистра, построенного на *JK*-триггерах?

11 Как работает схема сдвигового регистра, построенного на *D*-триггерах?

Лабораторная работа № 3

СИНТЕЗ НАДЕЖНЫХ КОМБИНАЦИОННЫХ СХЕМ С ПРОВЕРКОЙ ПРАВИЛЬНОСТИ ИХ ФУНКЦИОНИРОВАНИЯ

Ц е л ь р а б о т ы. Научиться создавать избыточные структуры путем необходимого резервирования, изучить способности корректирования ошибок различными логическими схемами.

1 Сведения из теории

1.1 Избыточные устройства с восстанавливающими органами

Резервирование дискретного устройства предусматривает установку одного или нескольких идентичных экземпляров устройства. Резервироваться может как все устройство в целом, так и отдельные его элементы или блоки. При *поблочном резервировании* каждый функциональный блок дискретного устройства заменяют несколькими идентичными блоками. На рисунке 3.1 приведена структурная схема резервирования дискретного устройства в целом. Число k устанавливаемых идентичных устройств называют кратностью резервирования или избыточностью. Выходы y_1, y_2, \dots, y_k отдельных дискретных устройств (*ДУ*) подаются на входы восстанавливающего органа (*ВО*), который исправляет ошибки, возникающие на выходах *ДУ*. Если все *ДУ* исправны, двоичные сигналы y_i на их выходах равны между собой. На выходе *ВО* в этом случае образуется сигнал $z = y_1 = y_2 = \dots = y_k$.

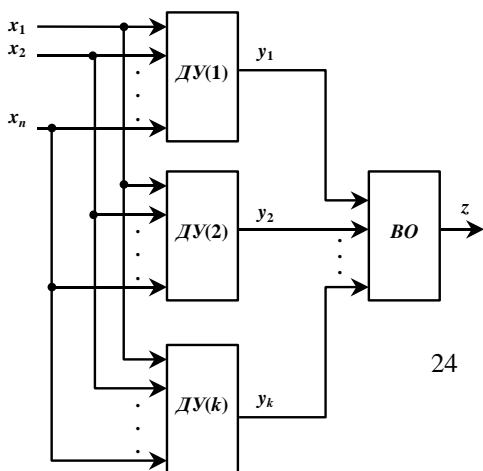


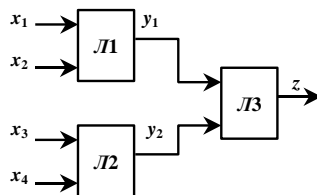
Рисунок 3.1 – Структурная схема избыточного устройства с восстанавливающим органом

При неисправности какого-либо из $ДУ$ на его выходе может появиться ложный сигнал. Ошибки могут быть двух типов: вместо правильного сигнала логического 0 появляется ложный сигнал логической 1 (ошибка типа $0 \rightarrow 1$) или вместо правильного сигнала логической 1 – ложный сигнал логического 0 ($1 \rightarrow 0$). Принято также появление сигнала логической 1 вместо логического 0 называть ложным срабатыванием, а появление 0 вместо 1 – ложным несрабатыванием. Правильный сигнал z на выходе избыточной структуры восстанавливается только в случае непревышения количеством ложных сигналов на выходах $ДУ$ некоторого числа, т. е. $ВО$ исправляет ограниченное число ошибочных сигналов y_i .

Восстанавливающая способность зависит от избыточности и логической функции $ВО$.

Рисунки 3.2 и 3.3 на конкретном примере поясняют принцип поблочного резервирования дискретных устройств.

Рисунок 3.2 – Структурная схема нерезервированного дискретного устройства с одним выходом



Рассматриваемое устройство состоит из трех логических функциональных блоков $Л1 - Л3$.

При резервировании каждый блок заменяют k (в данном случае $k = 2$) идентичными блоками с установкой отдельного $ВО$.

Как правило, поблочное резервирование дает большее повышение надежности по сравнению с резервированием всего устройства, но и требует большего числа восстанавливающих органов.

При решении конкретной задачи резервирования дискретного устройства следует выбрать значение избыточности k и логическую функцию $ВО$. Очевидно, что чем выше избыточность k , тем больше ошибок на выходах функциональных блоков будут исправляться. Значение k выбирают, исходя из конкретного заданного значения параметра надежности разрабатываемого устройства. Логическую функцию $ВО$ выбирают в зависимости от того, какой отказ дискретного устройства является наиболее

нежелательным с точки зрения воздействия на объект управления (ложное срабатывание, несрабатывание или то и другое).

Наиболее часто в качестве *ВО* применяют элемент, реализующий пороговую логическую функцию

$$z = M_k^r = f(y_1, y_2, \dots, y_k) = \begin{cases} 0, & \text{если } \sum_{i=1}^k y_i - r < 0; \\ 1, & \text{если } \sum_{i=1}^k y_i - r \geq 0, \end{cases}$$

где r – порог *ВО*;

k – избыточность;

y_i – логический сигнал на выходе i -го функционального блока.

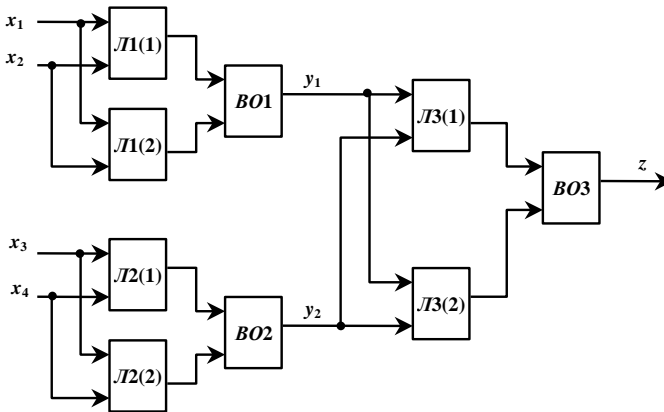


Рисунок 3.3 – Структурная схема резервированного дискретного устройства

Восстанавливающий орган, описываемый приведенным выражением, работает так. Если на r и более из k входов подается сигнал логической 1, на выходе также формируется сигнал логической 1, в противном случае – сигнал логического 0.

Вычислим функцию $z = M_k^r = \bigvee_{f=1}^{C_k^r} m_f$, где m_f ($f \in \{1, 2, \dots, C_k^r\}$) – конъюнкция r переменных без отрицаний из множества $\{y_1, y_2, \dots, y_k\}$.

Определим, например, все возможные функции вида M_3^r : $M_3^1 = y_1 \vee y_2 \vee y_3$;

$M_3^2 = y_1 y_2 \vee y_1 y_3 \vee y_2 y_3$; $M_3^3 = y_1 y_2 y_3$.

Рассмотрим более подробно случай дублирования дискретных устройств ($k = 2$). В этом случае могут быть использованы два типа *ВО*, реализующие функции $M_2^1 = y_1 \vee y_2$ и $M_2^2 = y_1 y_2$. На рисунке 3.4 показаны

соответствующие избыточные структуры. Если в качестве BO применен элемент ИЛИ, в схеме исправляется одна ошибка типа $1 \rightarrow 0$, но не исправляются ошибки типа $0 \rightarrow 1$. Действительно, пусть на выходе $ДУ(1)$ появляется ошибка типа $1 \rightarrow 0$, а $ДУ(2)$ в это время исправен (на его выходе есть сигнал логической 1). Тогда в соответствии со свойствами элемента ИЛИ на выходе восстанавливающего органа $z = 1$, т. е. происходит исправление указанной ошибки. С другой стороны, если на выходе одного из $ДУ$ возникает ошибка типа $0 \rightarrow 1$, она в соответствии со свойствами элемента ИЛИ непосредственно передается на выход BO вне зависимости от состояния другого $ДУ$. Если же в качестве BO используется элемент И, то, наоборот, в схеме исправляется одна ошибка типа $0 \rightarrow 1$, но не исправляются ошибки типа $1 \rightarrow 0$. В связи с этим, если с точки зрения воздействия на объект управления необходимо уменьшить вероятность ложного срабатывания схемы, то в качестве BO следует применять элемент И, если же необходимо уменьшить вероятность ложного несрабатывания, – то элемент ИЛИ.

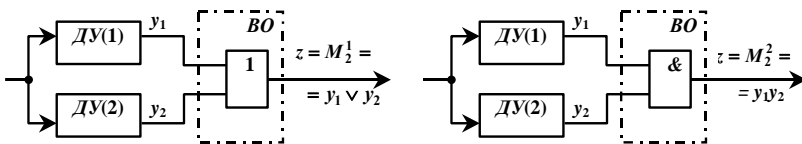


Рисунок 3.4 – Схемы дублирования дискретного устройства

Оценим надежность дублирования схемы с элементом ИЛИ в качестве BO . Введем обозначения: p_0 (q_0) – вероятность отсутствия (появления) на выходе функционального блока ошибки типа $1 \rightarrow 0$; p_1 (q_1) – вероятность отсутствия (появления) на выходе функционального блока ошибки типа $0 \rightarrow 1$. Имеют место равенства: $p_0 = 1 - q_0$; $p_1 = 1 - q_1$. Расчет надежности проводится с использованием таблицы повреждений, которой в рассматриваемом случае является таблица 3.1. В ней показаны значения выходной переменной избыточной схемы z при различных состояниях исправности всех k ее логических блоков.

Переменные V_i ($i \in \{1, 2, \dots, k\}$) используют для обозначения состояний логических блоков. Если $V_i = 0$, то соответствующий блок исправен и выдает сигнал $y_i^* = y_i$,

если же $V_i = 1$, то неисправен и выдает сигнал $y_i^* = \bar{y}_i$. В двух последних

Таблица 3.1 – Таблица повреждений

V_1	V_2	z	
		при $y_i = 0$	при $y_i = 1$
0	0	0	1
1	0	1	1
0	1	1	1
1	1	1	0

столбцах таблицы записывают значения выходной переменной z при заданном наборе состояний всех блоков для двух значений правильного сигнала на выходе логического блока $y_i = 0$ и $y_i = 1$. Для получения значений z в логическую функцию BO следует подставить значения величин $y_i^* = y_i$, если $V_i = 0$, и $y_i^* = \bar{y}_i$, если $V_i = 1$. Таблица повреждений имеет 2^k строк, соответствующих всем возможным комбинациям значений V_i . Например, таблица 3.1 содержит четыре строки, так как при двух блоках существуют четыре комбинации значений V_i . Первая строка соответствует случаю исправного состояния обоих блоков, вторая и третья строки – соответственно случаям неисправного состояния первого и второго блоков, четвертая строка – случаю неисправного состояния обоих блоков. Существенными повреждениями избыточной структуры называют повреждения логических блоков, вызывающие появление сигнала $z = \bar{y}_i$. В таблице 3.1 для случая $y_i = 0$ существенными являются неисправности, соответствующие второй – четвертой строкам таблицы, а для случая $y_i = 1$ – четвертой строке.

Пользуясь таблицей повреждений, можно определить характеристики надежности различных избыточных структур при произвольной функции BO , считая, что последний абсолютно надежен. Повреждения различных блоков являются независимыми событиями, а появление различных состояний повреждений блоков – несовместимыми событиями, составляющими полную группу событий. Поэтому, рассматривая, например, в таблице 3.1 столбец z при $y_i = 1$, можно найти вероятность Q_0 появления существенных повреждений избыточной структуры, вызывающих ошибку типа $1 \rightarrow 0$, и вероятность P_0 правильного срабатывания структуры суммированием вероятностей появления тех сочетаний повреждений блоков, при которых имеет место указанная ошибка (или, если это удобнее, при которых не имеет место указанная ошибка). Рассматривая столбец z при $y_i = 1$, определяем, что ошибка типа $1 \rightarrow 0$ имеет место только в одном случае, когда $V_1 = V_2 = 1$, т. е. при неисправности обоих блоков дублированной структуры. Поэтому

$$Q_0 = q_0^2 = (1 - p_0)^2; P_0 = 1 - Q_0 = 1 - (1 - p_0)^2 = 2p_0 - p_0^2.$$

Рассматривая столбец z при $y_i = 0$, можно найти вероятность P_1 отсутствия существенных повреждений избыточной структуры, вызывающих на выходе ошибку типа $0 \rightarrow 1$. Из таблицы 3.1 видно, что ошибка типа $0 \rightarrow 1$ возможна при повреждении одного или двух логических блоков и невозможна только в случае исправности обоих блоков. Поэтому $P_1 = p_1^2$.

Если в схеме дублирования BO реализует функцию И, надежность

избыточной структуры: $P_0 = p_0^2$; $P_1 = 1 - (1 - p_1)^2 = 2p_1 - p_1^2$.

Таким образом, при избыточности $k = 2$ использование BO , реализующего логическую функцию ИЛИ, увеличивает P_0 и уменьшает P_1 , а использование BO , реализующего функцию И, наоборот, увеличивает P_1 и уменьшает P_0 по сравнению с неизбыточной структурой.

Часто резервирование дискретного устройства с использованием логического элемента ИЛИ называют параллельным, а элемента И – последовательным резервированием.

Если необходимо исправлять более одной ошибки одного типа или одновременно ошибки обоих типов, приходится выбирать избыточность $k > 2$. При $k = 3$ возможны три вида BO , реализующие функции M_3^1 , M_3^2 , M_3^3 . Использование функций M_3^1 (ИЛИ) и M_3^3 (И) позволяет исправлять по две ошибки соответственно типов $1 \rightarrow 0$ и $0 \rightarrow 1$, а использование функции $M_3^2 = y_1 y_2 \vee y_1 y_3 \vee y_2 y_3$ – одиночные ошибки обоих типов.

Рассмотрим работу схемы резервирования с использованием BO , реализующего функцию M_3^2 (рисунок 3.5) при одиночных неисправностях.

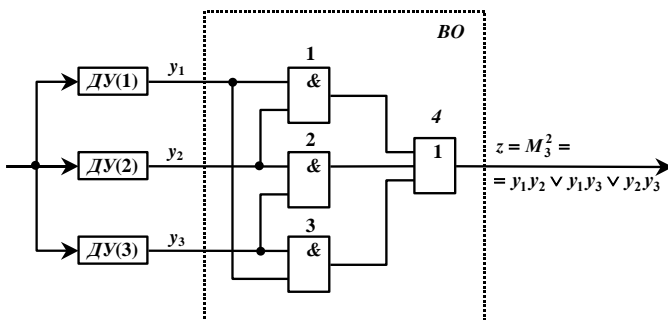


Рисунок 3.5 – Структурная схема резервирования с использованием мажоритарного элемента

Пусть на выходе $DV(1)$ имеется ошибка типа $1 \rightarrow 0$ ($y_1 = 0$). Остальные два блока при этом исправны ($y_2 = y_3 = 1$). Тогда на выходах элементов И1 и 3 также возникает ошибка типа $1 \rightarrow 0$, но на выходе элемента И2 сигнал логической 1 сохраняется, так как этот элемент не связан с выходом неисправного блока. Данный сигнал через элемент ИЛИ передается на выход устройства, и, следовательно, ошибка исправляется. Если на выходе есть ошибка типа $0 \rightarrow 1$ ($y_1 = 1$), а на выходах двух остальных исправных блоков – сигнал логического 0, на выходах всех элементов И будет присутствовать сигнал логического 0, так как каждый из них связан с

выходом хотя бы одного исправного блока. В этом случае на выходе устройства также будет сигнал логического 0, и, следовательно, указанная ошибка исправляется. В то же время две ошибки в схеме не исправляются. Если, например, на выходах двух блоков возникает ошибка типа $1 \rightarrow 0$, на выходах всех элементов $И$ устанавливается сигнал логического 0, и ошибка передается на выход устройства.

Функция M_3^2 получила название мажоритарной, а реализующий ее элемент – мажоритарным. Работает *мажоритарный элемент* в соответствии со следующим правилом: на выходе мажоритарного элемента присутствует сигнал логической 1 в том и только в том случае, если на большинство его входов поданы сигналы логической 1. Число входов мажоритарного элемента всегда нечетное. Например, на выходе элемента, реализующего функцию M_3^2 сигнал логической 1 появляется при подаче такого же сигнала на любые из двух входов элемента (рисунок 3.5). Такой элемент получил наиболее широкое распространение на практике в качестве $ВО$, так как он при сравнительно небольшой избыточности (в 3 раза) обеспечивает повышение надежности относительно ошибок обоих типов. Характеристики надежности в этом случае следующие (с условием абсолютной надежности $ВО$): $P_0 = 3p_0^2 - 2p_0$; $P_1 = 3p_1^2 - 2p_1$. В общем случае при построении резервированного устройства с избыточностью k и использовании $ВО$, реализующего функцию M_k^r , в нем исправляются $k - r$ ошибок типа $1 \rightarrow 0$ и $r - 1$ ошибок типа $0 \rightarrow 1$, а характеристики надежности следующие: $P_0 = \sum_{i=r}^k C_k^i p_0^i (1 - p_0)^{k-i}$; $P_1 = \sum_{i=k-r+1}^k C_k^i p_1^i (1 - p_1)^{k-i}$.

При достаточно высокой надежности логических блоков p_0 и p_1 выбор большого k позволяет получить высокую надежность логической части избыточной структуры. В этом случае надежность структуры в целом будет определяться и ограничиваться надежностью восстанавливающего органа. Если дискретное устройство резервируется целиком (рисунок 3.1), структура содержит только один $ВО$. Поэтому следует обратить особое внимание на повышение надежности $ВО$ (например, резервированием его внутренних элементов). Очевидно, что надежность $ВО$ должна быть, по крайней мере, не ниже надежности остальной части избыточной структуры.

При поблочном резервировании (рисунок 3.3) избыточная структура содержит большое число $ВО$. Поэтому влияние $ВО$ на общую надежность устройства увеличивается. Для повышения надежности возможна установка вместо одного k одинаковых восстанавливающих органов $ВО$ (рисунок 3.6).

Каждый из $ВО$ имеет k входов (в данном случае $k = 2$), на которые поступают сигналы с выходов всех логических блоков. Выходной сигнал с

каждого BO подается только на один соответствующий логический блок последующего уровня схемы. Повреждение BO для последующего уровня схемы эквивалентно повреждению следующего за BO логического блока, и поэтому избыточная структура остается работоспособной при повреждениях одного или нескольких BO в зависимости от выбранного значения k . Однако, и в этом случае остается проблема повышения надежности BO , выход которого является выходом всего устройства [$BO(3)$].

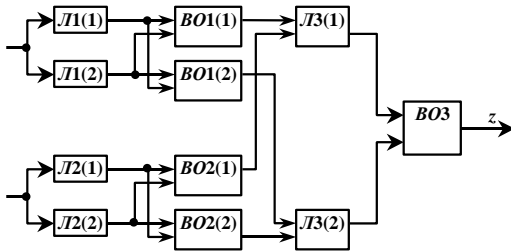


Рисунок 3.6 – Схема избыточной структуры с несколькими восстанавливающими органами

На рисунке 3.7 приведен фрагмент избыточной структуры, в которой вместо одного устанавливают $k = 3$ восстанавливающих органа, реализующих мажоритарную функцию M_3^2 .

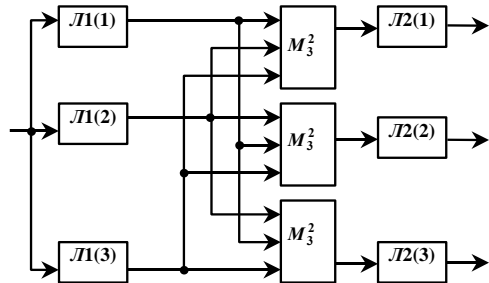


Рисунок 3.7 – Схема троированной структуры

Такая структура получила название троированной. Она находит широкое применение при построении надежных вычислительных и логических устройств. Структура не отказывает при любых повреждениях одного из $k = 3$ одинаковых логических блоков и одного из $k = 3$ одинаковых восстанавливающих органа.

1.2 Надежные комбинационные схемы

Рассмотрим вопросы повышения надежности комбинационных схем, построенных на функциональных логических элементах (ЛЭ). Во внутреннюю структуру ЛЭ входят отдельные детали: транзисторы, диоды, резисторы и т. д. Повреждение какой-либо детали может изменить логическую функцию, реализуемую ЛЭ. Очевидно, что надежность ЛЭ

может быть повышена последовательным или параллельным соединением входящих в его структуру деталей.

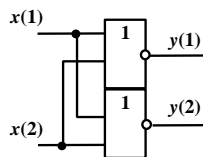
Применение резервированных элементов позволяет просто строить надежные логические схемы: достаточно в избыточной схеме каждый ЛЭ заменить его резервированной модификацией. В этом случае можно добиться существенного увеличения надежности относительно отказов деталей внутренней структуры ЛЭ, но обрывы соединений между входами и выходами элементов схемы будут приводить к выходу ее из строя. В то же время следует отметить, что вероятность указанных обрывов в современных устройствах остается довольно высокой, так как обрывы связаны с некачественной пайкой и нарушением контакта в механических разъемах. Число же соединений в схемах, как правило, очень велико.

Влияние обрывов на правильную работу схемы исключается в комбинационных схемах с многократными связями. В них каждый ЛЭ выполняет логические операции над тождественными копиями входных сигналов, которые поступают по пучкам входных связей.

Назовем собственными ошибками ЛЭ ошибки на его выходе, вызванные повреждениями деталей внутренней структуры элемента, а входными ошибками – искажения сигналов на входе из-за повреждений деталей в предыдущих каскадах устройства или повреждений соединительных линий. Каждый ЛЭ в избыточной схеме с многократными связями должен обладать свойствами логического элемента для выполнения логических функций и восстанавливающего органа для восстановления правильного сигнала коррекцией входных или собственных ошибок.

Резервированный элемент НЕ с двойными связями (рисунок 3.8) состоит из двух отдельных элементов НЕ, в которых осуществлено параллельное резервирование транзисторов.

Рисунок 3.8 – Условное обозначение резервированного элемента НЕ с двойными связями



Входные и выходные сигналы элемента передаются по двум линиям. В исправном состоянии $x(1) = x(2)$ и $y(1) = y(2)$. Схема, построенная на соединенных между собой элементах с двойными связями (рисунок 3.9) работает правильно при любых одиночных повреждениях деталей ЛЭ и обрыве одной из параллельных линий, связывающих между собой два элемента.

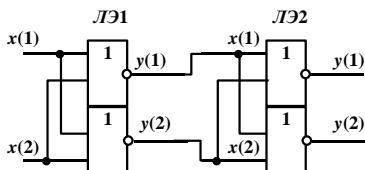


Рисунок 3.9 – Схема последовательного соединения элементов НЕ с двойными связями

Хорошие результаты по повышению надежности дают схемы с тройными связями, в которых для исправления ошибок используется мажоритарная функция M_3^2 (рисунок 3.10). Сигналы передаются по пучку из трех линий. На входах элемента сначала осуществляется исправление сигналов элементами M_3^2 , а затем реализация логической операции. Элемент корректирует входные ошибки обоих типов, собственные одиночные ошибки исправляются на входах последующего каскада. Такие элементы строят на резистивно-транзисторных схемах, в которых мажоритарная функция реализуется на входных резисторах элементов.

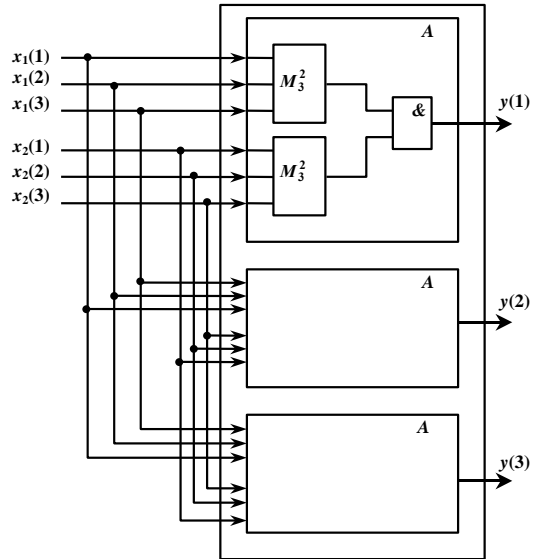


Рисунок 3.10 – Схема логической структуры резервированного элемента А с тройными связями

В схемах с двойными и тройными связями эффект исправления ошибок достигается благодаря специальным свойствам используемых ЛЭ. Такие свойства получают усложнением внутренней структуры элементов. Однако в распоряжении разработчика могут не всегда быть такие элементы. Поэтому особый интерес представляют схемы с учетверенными связями, в которых никакие особые требования к ЛЭ не предъявляются.

Рассмотрим идею *метода учетверения*. Он использует естественные исправляющие свойства элементов, на входы которых многократно подан один и тот же сигнал. На рисунке 3.11, а показана реакция на ошибки на

входах элемента И.

Ошибка типа $1 \rightarrow 0$ передается на выход, а ошибка типа $0 \rightarrow 1$ исправляется элементом. Элемент ИЛИ наоборот, исправляет ошибку типа $1 \rightarrow 0$ (рисунок 3.11, б). Так как элементы И и ИЛИ исправляют ошибки разных типов, то возникает возможность исправления ошибок обоих типов последовательным включением этих элементов. Схема приведенная на рисунке 3.11, в, исправляет ошибки обоих типов на входах элементов И.

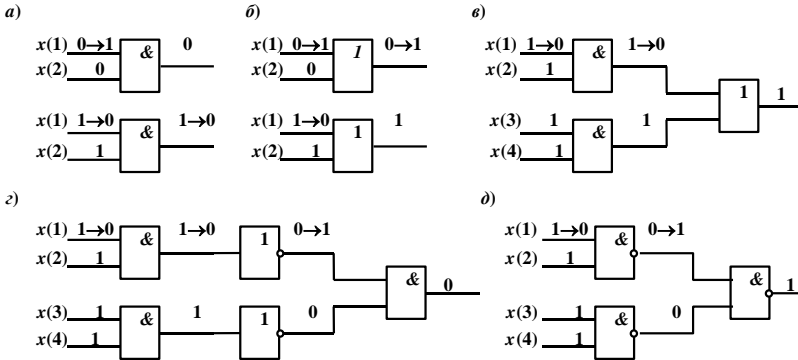


Рисунок 3.11 – Схемы исправления ошибок логическими элементами

Ошибки типа $0 \rightarrow 1$ исправляются самими элементами И, а ошибки типа $1 \rightarrow 0$ – элементом ИЛИ, который составляет второй каскад схемы. Для того чтобы добиться данного свойства схемы, пришлось установить два элемента И, а число независимых линий, по которым передается входная переменная, увеличить до четырех. В полученной схеме исправляются все ошибки на входах элементов, кроме ошибок типа $0 \rightarrow 1$ на входах элемента ИЛИ. Для исправления данных ошибок в схему следует ввести еще большую избыточность, а именно третий каскад схемы (рисунок 3.12, а).

Ошибка типа $0 \rightarrow 1$ на входе элемента ИЛИ исправляется на выходе элемента И третьего каскада. Число элементов И первого каскада схемы возросло до четырех, число независимых копий входного сигнала – до восьми. Дальнейший рост числа каскадов схемы вызывает увеличение числа элементов в каждом предыдущем каскаде и числа независимых входных сигналов. Однако все ошибки могут быть исправлены схемами с четырьмя одинаковыми элементами в каждом каскаде благодаря специальному соединению между собой входов элементов одного каскада. Схема (см. рисунок 3.12, б) обладает теми же свойствами по исправлению ошибок, что и схема, приведенная на рисунке 3.12, а. На этой схеме объединены входы элементов первого каскада. В результате вместо восьми потребовалось четыре независимых входных сигнала.

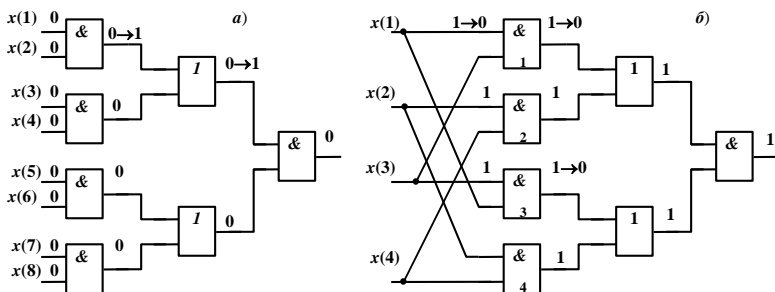


Рисунок 3.12 – Схемы логические трехкаскадные

Объединение входов элементов каскада приводит к появлению на выходах каскада двойной ошибки, обусловленной одиночной ошибкой.

Например, ошибка $1 \rightarrow 0$ на входе $x(1)$ (см. рисунок 3.12, б) вызывает такую же ошибку на выходах элементов 1 и 3 первого каскада схемы. Однако обе эти ошибки исправляются следующим каскадом схемы, так как поступают на входы различных его элементов. Очевидно, что если бы обе ошибки поступили на вход одного элемента ИЛИ второго каскада, это вызвало бы появление ошибки на его выходе, а, следовательно, и на выходе всей схемы.

Рассмотрим трехкаскадную схему с учетверенными связями (рисунок 3.13).

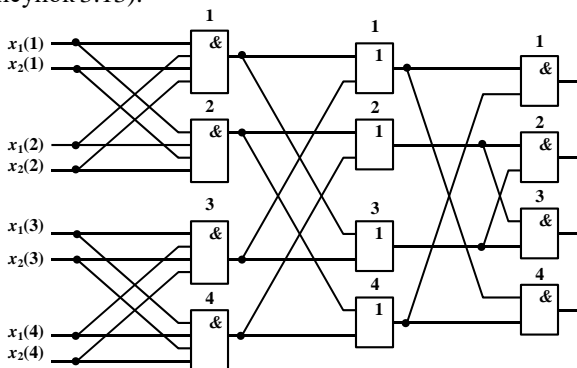


Рисунок 3.13 – Схема И-ИЛИ-НЕ с учетверенными связями

Элементы И первого каскада имеют по два сдвоенных входа, т. е. помимо задачи исправления ошибок, они в данном случае выполняют и свою основную задачу реализацию логических функций. Аналогично могут быть включены элементы и других каскадов (соответствующие их входы не показаны для упрощения рисунка). Входы элементов каждого каскада объединены в соответствии с определенной формулой. Для первого каскада имеет место формула $[1, 2 - 3, 4]$ (объединены входы первого и второго, а также третьего и четвертого элементов), для второго каскада формула $[1, 3 -$

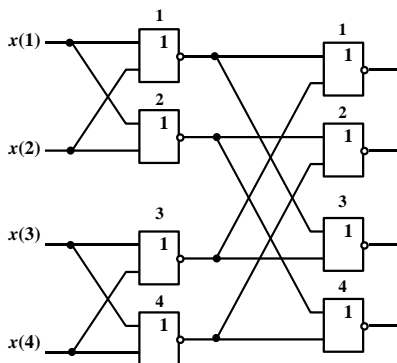
2, 4] и для третьего каскада формула [1, 4 – 2, 3].

Необходимым условием исправления ошибок является объединение входов элементов двух соседних каскадов схемы в соответствии с различными формулами. Это обеспечивает невозможность попадания двух ошибок, возникающих на выходе какого-либо каскада в результате одиночной ошибки предыдущего каскада, на входы одного и того же элемента последующего каскада. В этом случае (как было показано на примере схемы, см. рисунок 3.12, б) двойные ошибки исправляются. Если в соответствии с логической структурой схемы необходимо соединить последовательно несколько каскадов из одинаковых элементов, их включают по одной и той же формуле. В этом случае ошибки обоих каскадов исправляются первым последующим каскадом из элементов другого типа.

Вернемся к схеме, приведенной на рисунке 3.11, в. Включим на выходе элементов И первого каскада инверторы (рисунок 3.11, з). Инвертор меняет на противоположный тип ошибки, имеющейся на его входе. Поэтому в данной схеме для исправления ошибок на входах элементов первого каскада после инверторов вместо элемента ИЛИ следует установить элемент И. Из этого следует, что схемы (рисунок 3.11, д), представляющие собой последовательное включение каскадов из элементов И-НЕ (или ИЛИ-НЕ) обладают способностью исправлять одиночные ошибки обоих типов.

Избыточную комбинационную схему с учетверенными связями (рисунок 3.14) строят по структуре избыточной схемы заменой каждого ее элемента каскадом из четырех соответствующих элементов со двоянными входами.

Рисунок 3.14 – Схема из двух каскадов элементов ИЛИ-НЕ с учетверенными связями



Необходимо правильно чередовать формулы объединения входов элементов различных каскадов. Элементы двух последних каскадов схемы должны исправлять ошибки обоих типов. На рисунке 3.15 показан пример построения схемы, реализующей функцию $f = x_1x_2 \vee \bar{x}_3$. В ней для каждого элемента с учетверенными связями указана формула объединения

ВХОДОВ.

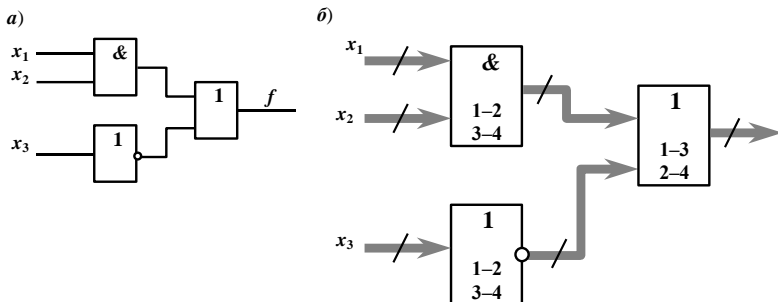


Рисунок 3.15 – Неизбыточный (а) и избыточный (б) варианты избыточной схемы с учетверенными связями

2 Порядок выполнения работы

1 Изучить сведения из теории по данным методическим указаниям или учебнику [2, с.181 – 196].

2 Построить комбинационную схему без ее резервирования по заданному в таблице 3.2 варианту.

Таблица 3.2 – Варианты заданий

Вариант	Формула	Вариант	Формула
1	$f = x_1 \bar{x}_2 \vee \bar{x}_3$	9	$f = \bar{x}_1 x_2 \vee x_3 x_4$
2	$f = \bar{x}_1 x_2 \vee \bar{x}_3$	10	$f = \bar{x}_1 \bar{x}_2 \vee x_3 x_4$
3	$f = x_1 x_2 \vee x_3 x_4$	11	$f = \bar{x}_1 x_2 \vee x_3$
4	$f = x_1 \bar{x}_2 \vee x_3 x_4$	12	$f = x_1 x_2 \vee \bar{x}_3 x_4$
5	$f = \bar{x}_1 \bar{x}_2 \vee \bar{x}_3$	13	$f = x_1 \bar{x}_2 \vee x_3$
6	$f = x_1 x_2 \vee x_3$	14	$f = x_1 x_2 \vee x_3 \bar{x}_4$
7	$f = \bar{x}_3 \vee x_1 \bar{x}_2 x_4$	15	$f = x_1 x_2 \vee \bar{x}_3 \bar{x}_4$
8	$f = x_1 \vee x_2 x_3$	16	$f = x_3 \vee \bar{x}_1 x_2 x_4$

3 Построить таблицу истинности (ТИ) заданной схемы.

4 Выполнить резервирование заданной схемы по аналогии со схемой, приведенной на рисунке 3.3. В качестве $BO1$ и $BO2$ использовать элементы И, а $BO3$ – элемент ИЛИ.

5 Внося ошибки типов $0 \rightarrow 1$ и $1 \rightarrow 0$ в точках, указанных преподавателем, проверить работоспособность схемы. Построить ТИ резервированной схемы при внесении ошибок типа $0 \rightarrow 1$ и $1 \rightarrow 0$.

6 Сделать выводы о работоспособности резервированной схемы при

внесении ошибок.

7 Выполнить резервирование исходной схемы, используя в качестве восстанавливающих органов $BO1 - BO3$ мажоритарный элемент (рисунок 3.5).

8 Внося неисправности типов $0 \rightarrow 1$ и $1 \rightarrow 0$ в тех же точках, что и в п. 5, проверить работоспособность схемы. Построить ТИ резервированной схемы.

9 Сделать выводы о работоспособности резервированной схемы при внесении ошибок.

10 Построить избыточный вариант схемы по принципу схемы с учетверенными связями (см. рисунок 3.15).

11 Внося ошибки типов $0 \rightarrow 1$ и $1 \rightarrow 0$ в прежних точках, проверить работоспособность схемы и построить ее ТИ.

12 Сделать выводы по сравнению трех вариантов резервирования исходной схемы.

Содержание отчета

Наименование и цель работы, исходная схема и три варианта ее резервирования с таблицами истинности, ответы на заданные преподавателем контрольные вопросы, выводы по работе.

Контрольные вопросы

- 1 Что понимают под резервированием дискретного устройства?
- 2 Что называют кратностью резервирования?
- 3 Какой элемент чаще всего применяют для реализации BO ?
- 4 Чем отличается параллельное резервирование от последовательного?
- 5 Каких три вида BO возможны при избыточности $k = 3$.
- 6 Опишите принцип действия мажоритарного элемента.
- 7 Особенности поблочного резервирования.
- 8 Особенности троированных структур.
- 9 В чем заключается отличие между собственными и входными ошибками логических элементов?
- 10 Опишите работу схемы логической структуры резервированного элемента И с тройными связями.
- 11 В чем заключается идея метода учетверения?
- 12 Приведите формулы для расчета надежности дублированной избыточной структуры при использовании в качестве BO элемента И.
- 13 Приведите формулы для расчета надежности дублированной избыточной структуры при использовании в качестве BO мажоритарного элемента.

14 Приведите формулы для расчета надежности резервированного дискретного устройства с избыточностью k и использованием BO , реализующим функцию M_k^r .

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1 Автоматизация систем энергоснабжения: Учебник для вузов ж.-д. трансп. /Ю. И. Жарков, В. Я. Овласюк, Н. Г. Сергеев, Н. Д. Сухопрудский, А. С. Шилов; Под ред. Н. Д. Сухопрудского. М.: Транспорт, 1990. 359 с.

2 Сапожников В. В. и др. Дискретные устройства железнодорожной автоматики, телемеханики и связи: Учебник для вузов ж.-д. трансп. М.: Транспорт, 1988. 255 с.

3 Букреев И. Н. и др. Микроэлектронные схемы цифровых устройств. М.: Советское радио, 1975. 264 с.

4 Алексеенко А. Г., Шагурин И. И. Микросхемотехника. М.: Радио и связь, 1982. 414 с.

О Г Л А В Л Е Н И Е

Введение	3
<i>Лабораторная работа № 1</i> Изучение счетчиков импульсов.....	3
<i>Лабораторная работа № 2</i> Изучение регистров	15
<i>Лабораторная работа № 3</i> Синтез надежных комбинационных схем с проверкой правильности их функционирования	23
Рекомендуемая литература.....	38

Учебное издание

Константин Афанасьевич БОЧКОВ
Юрий Федорович БЕРЕЗНЯЦКИЙ

Анализ функциональной структуры и синтез дискретных устройств

Лабораторный практикум по дисциплине «Теория дискретных устройств»

Часть III

Редактор Т. М. Р и з е в с к а я
Технический редактор В. Н. К у ч е р о в а
Корректоры Н. А. Д а ш к е в и ч, О. В. З а н и н а

Подписано в печать .01. 2001 г. Формат бумаги 60x84_{1/16}.
Бумага газетная. Гарнитура Таймс. Печать офсетная.
Усл. печ. л. 2,32. Уч.-изд. л. 2,23. Тираж 200 экз.
Зак. № 2459. Изд. № 3582.

Редакционно-издательский отдел БелГУТа, 246653, г.Гомель, ул. Кирова, 34.
Лицензия ЛВ № 57 от 22.10.97 г.

Типография БелГУТа, 246022, г. Гомель, ул. Кирова, 34.
Лицензия ЛП № 360 от 26.07.99 г.