МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
к практическим занятиям

«Основы цифровой схемотехники»

для магистров направления
«Прикладная математика и информатика»

профиль
«Математическое моделирование»

Ставрополь, 2026

Основы цифровой схемотехники: Методические указания по выполнению практических занятий / Сост. Шапошников А.В. - Ставрополь: СКФУ, 2019. - 128 с.

Пособие подготовлено в соответствии с Федеральным государственным образовательным стандартом высшего образования. В учебном пособии изложены методические рекомендации по выполнению шестнадцати практических занятий.

Пособие предназначено для изучения дисциплины «Основы цифровой схемотехники» для магистров направления подготовки «Прикладная математика и информатика» (академический магистр).

Авторы:
канд. техн. наук, доц. каф. ПМиММ ИМЕН СКФУ Шапошников А.В.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Основная цель учебной дисциплины - формирование базовой подготовки студентов в области цифровых устройств и микропроцессорных систем и развитии навыков использования цифровой техники. Задачей курса является изучение основ построения и функционирования типовых цифровых устройств.

Освоение дисциплины «Основы цифровой схемотехники» позволит будущему магистру по направлению подготовки «Прикладная математика и информатика» полноценно осуществлять свою профессиональную деятельность, в частности, обладать следующими компетенциями:

- Способен проводить научные исследования и получать новые научные и прикладные результаты самостоятельно и в составе научного коллектива на основе существующих методов в конкретной области профессиональной деятельности (ПК-1),
- Способен управлять информацией с использованием цифровых средств и алгоритмов обработки данных для решения задач профессиональной деятельности (ПК-7).

Студенты, овладевшие указанными компетенциями, должны:

знать: элементную базу современных вычислительных машин, схемотехнику построения типовых цифровых устройств.

уметь: проводить анализ функционирования цифровых устройств, синтезировать цифровые устройства по их описанию.

владеть: методами синтеза микропрограммных устройств, средствами автоматизированного проектирования цифровых устройств.

Аппаратура и материалы

Лабораторный практикум выполняется в штатном компьютерном классе, оборудованном IBM-совместимыми персональными компьютерами под управлением операционной системы Windows. Необходимым является наличие математической программы Matlab с установленным пактом Simulink.

Указания по технике безопасности

Указания по технике безопасности в штатном компьютерном классе разработаны и утверждены администрацией и доводятся каждому студенту под роспись. Дополнительных требований нет.

1 ПРЕОБРАЗОВАНИЕ ЛОГИЧЕСКИХ ВЫРАЖЕНИЙ

Цель занятия:

Закрепить знания, полученные на лекции по логическим преобразованиям.

МЕТОДИЧЕСКИЕ ПОЯСНЕНИЯ И РЕКОМЕНДАЦИИ

Условимся для дальнейшего значение «истина» обозначать цифрой 1, а значение «ложь» – цифрой 0 (в литературе используются также обозначения *T* (true) u *F* (false) и др.) Тогда определения пяти основных логических операций над высказываниями (отрицание, конъюнкцию, дизъюнкцию, импликацию и эквиваленцию) можно свести в одну таблицу (таблица 1.1), которая называется *таблицей истинности*:

Таблица 1.1 – Таблица истинности для определения основных логических операций

x	y	$\bar{x}$	$x\&y$	$x \vee y$	$x \to y$	$x \leftrightarrow y$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Подобные таблицы истинности часто используют для доказательства логических тождеств и равносильностей, законов логики, в различных приложениях булевой алгебры.

На данном практическом занятии на основе определения логических операций и законов логики мы будем устанавливать, к какому классу (тавтологии, тождественно ложные или выполнимые) принадлежат данные формулы алгебры логики. Кроме того, мы научимся выражать одни логические операции через другие, а также находить формулы, двойственные данным формулам. Материал данного практического занятия необходим для закрепления и приобретения простейших навыков для приведения логических функций к СДНФ (СКНФ), их оптимизации и синтезу логических схем.

Напомним, что всякое сложное высказывание, которое может быть получено из элементарных высказываний посредством применения логических операций и круглых скобок, регулирующих порядок их выполнения, называется *формулой алгебры логики*. При этом сами элементарные высказывания, а также константы 0 и 1 также являются простейшими формулами.

Все возможные логические значения формулы в зависимости от значений входящих в неё элементарных высказываний могут быть описаны полностью с помощью таблицы истинности.

По определению формула $f^*(x_1, x_2, ..., x_n)$ называется *двойственной* формуле $f(x_1, x_2, ..., x_n)$, если $f^*(x_1, x_2, ..., x_n) \equiv \overline{f(\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)}$, при этом столбец значений f^* таблицы истинности получается из столбца значений функции f инвертированием по-

следнего и отрицанием.

Для булевых формул двойственная формула может быть получена заменой операции & на $\vee$, операции $\vee$ на &, 1 на 0, 0 на 1 и сохранением структуры формулы (т.е. соответствующего порядка действий). Если в формулах присутствуют операции импликации и эквиваленции, их предварительно выражают через операции отрицания, конъюнкции и дизъюнкции.

Для выполнения тождественных преобразований формул логики высказываний необходимо *знать*:

а) определение логических операций над высказываниями;

б) основные законы алгебры Буля;

уметь: составлять таблицы истинности для сложных высказываний.

Приведем список основных свойств (законов) булевых операций алгебры высказываний. При этом условимся отношение равносильности на множестве высказываний (формул) обозначать знаком «≡».

$x \& x \equiv x$

$x \vee x \equiv x$ - законы идемпотентности

$x \& 1 \equiv x$

$x \vee 1 \equiv 1$

$x \& 0 \equiv 0$

$x \vee 0 \equiv x$ - законы нуля и единицы;

$x \& \bar{x} \equiv 0$ - закон противоречия;

$x \vee \bar{x} \equiv 1$ - закон исключенного третьего;

$x \& (y \vee x) \equiv x$

$x \vee (y \& x) \equiv x$ - законы поглощения;

$\bar{\bar{x}} \equiv x$ - закон двойного отрицания;

$\overline{x \& y} \equiv \bar{x} \vee \bar{y}$

$\overline{x \vee y} \equiv \bar{x} \& \bar{y}$ - законы де Моргана;

$x \& y \equiv y \& x$

$x \vee y \equiv y \vee x$ - коммутативные законы

$x \& (y \& z) \equiv (x \& y) \& z$

$x \vee (y \vee z) \equiv (x \vee y) \vee z$

- ассоциативные законы

$x \& (y \vee z) \equiv (x \& y) \vee (x \& z)$

$x \vee (y \& z) \equiv (x \vee y) \& (x \vee z)$

- дистрибутивные законы

$x \to y \equiv \bar{y} \to \bar{x}$ - закон контрапозиции.

Надо также уметь выражать одни логические операции через другие. Среди таких равносильностей особую роль играют следующие равносильности:

$x \leftrightarrow y \equiv (x \to y) \& (y \to x)$;

$x \to y \equiv \bar{x} \vee y; x \to y \equiv \overline{x \& \bar{y}}$;

Законы де Моргана 12 и 13.

$x \& y \equiv \overline{\bar{x} \vee \bar{y}}$;

$x \vee y \equiv \overline{\bar{x} \& \bar{y}}$;

$\overline{x \to y} \equiv x \& \bar{y}$ и др.

Задачи

Составить таблицы истинности для следующих формул и установить, какие из них являются тавтологиями. К какому классу относятся остальные формулы?

$(x \vee y) \to ((x \& \bar{y} \vee x) \to \bar{y})$;

$(x \& \bar{y}) \to ((y \vee \bar{x}) \to \bar{z}))$;

$(\bar{x} \vee z) \& (y \to (\bar{z} \to x))$;

$\overline{\overline{x \vee y}} \to \overline{\overline{x \& y}}$;

$((x \to y) \& \bar{y}) \to \bar{x}$;

$((p \& (q \to p)) \to \bar{p}$.

В качестве примера рассмотрим формулу 1.2

x	y	z	$\bar{x}$	$\bar{y}$	$\bar{z}$	$x \& \bar{y}$	$y \vee \bar{x}$	$(y \vee \bar{x}) \to \bar{z}$	f
0	0	0	1	1	1	0	1	1	1
0	0	1	1	1	0	0	1	0	1
0	1	0	1	0	1	0	1	1	1
0	1	1	1	0	0	0	1	0	1
1	0	0	0	1	1	1	0	1	1
1	0	1	0	1	0	1	0	1	1
1	1	0	0	0	1	0	1	1	1
1	1	1	0	0	0	0	1	0	1

Формула 1.2 является тавтологией, так как значение этой формулы на всех наборах равно 1.

Используя основные законы логики, с помощью равносильных преобразований доказать следующие соотношения:

$x \vee y \equiv \overline{\bar{x} \,\&\, \bar{y}}$;

$(x \,\&\, y) \vee (x \,\&\, \bar{y}) \equiv x$;

$x \vee x \,\&\, y \equiv x$;

$\bar{x} \vee \bar{y} \equiv y \to \bar{x}$;

$(x \,\&\, y) \vee (\bar{x} \,\&\, \bar{y}) \vee (\bar{x} \,\&\, y) \equiv x \to y$;

$(x \to \bar{y}) \equiv (y \to \bar{x})$.

Докажем, например, 1.2.5.

$(x \,\&\, y) \vee (\bar{x} \,\&\, \bar{y}) \vee (\bar{x} \,\&\, y) \equiv (x \,\&\, y) \vee (\bar{x} \,\&\, (\bar{y} \vee y)) \equiv (x \,\&\, y) \vee \bar{x} \equiv (x \vee \bar{x}) \,\&\, (y \vee \bar{x}) \equiv y \vee \bar{x} \equiv x \to y.$

Доказать тождественную истинность или тождественную ложность формул, применяя равносильные преобразования:

$(x \vee (\bar{x} \,\&\, y)) \sim (y \vee x)$;

$(x \to y) \vee (x \to \bar{y})$;

$(x \vee y) \,\&\, \bar{x} \to y$;

$x \,\&\, (x \to y) \,\&\, (x \to \bar{y})$;

$(x \to y) \,\&\, \bar{y} \to \bar{x}$;

$(x \to y) \to (\bar{x} \vee y)$.

Докажем тождественную ложность формулы 1.3.4.

$x \,\&\, (x \to y) \,\&\, (x \to \bar{y}) \equiv x \,\&\, (\bar{x} \vee y) \,\&\, (\bar{x} \vee \bar{y}) \equiv x \,\&\, (\bar{x} \vee (y \,\&\, \bar{y})) \equiv x \,\&\, \bar{x} \vee 0 \equiv x \,\&\, \bar{x} \equiv 0.$

Следующие формулы преобразовать так, чтобы они содержали только операции конъюнкции и отрицания:

$x \vee y$;

$x \to y$;

$x \leftrightarrow y$;

$x \vee (x \leftrightarrow y)$;

$(x \,\&\, \bar{y}) \to (\bar{y} \to x)$;

$(x \vee y) \to (\bar{x} \to z)$.

Рассмотрим, например, 1.4.4:

$x \vee (x \leftrightarrow y) \equiv x \vee ((x \to y) \,\&\, (y \to x)) \equiv x \vee (\bar{x} \vee y) \,\&\, (\bar{y} \vee x) \equiv$

$$\equiv x \vee \left(\overline{(x \,\&\, \bar{y})} \,\&\, \overline{(y \,\&\, \bar{x})} \right) \equiv \overline{\bar{x} \,\&\, \left(\overline{(x \,\&\, \bar{y})} \,\&\, \overline{(y \,\&\, \bar{x})} \right)}$$

1.5. Найти двойственные формулы к формулам:

1.5.1. $x \,\&\, (\bar{y} \vee z)$;

1.5.2. $(x \,\&\, y) \vee (x \,\&\, z)$;

1.5.3. $\overline{(x \vee y)} \,\&\, (x \vee (\bar{y} \,\&\, \bar{z}))$;

1.5.4. $((x \,\&\, y) \vee (y \,\&\, z) \vee (z \,\&\, x)) \,\&\, (\overline{x \vee y \vee z})$;

1.5.5. $((x \vee y) \,\&\, (\overline{x \vee z}) \vee (x \,\&\, y)) \vee (\overline{(x \vee y) \,\&\, z} \vee x)$;

1.5.6. $\left(\overline{x \,\&\, y \,\&\, z}\right) \vee \left(x \,\&\, y \,\&\, \bar{z}\right) \vee \left(x \,\&\, \bar{y} \,\&\, z\right) \vee \left(\bar{x} \,\&\, y \,\&\, z\right)$.

В качестве примера построим формулу f^*, двойственную формуле 1.5.3, исходя из определения двойственной формулы:

$$f^*(x,y,z) \equiv \overline{f(\bar{x},\bar{y},z)} \equiv \overline{\overline{(\bar{x} \vee \bar{y})} \,\&\, (\bar{x} \vee (y \,\&\, z))} \equiv \overline{(x \,\&\, y) \,\&\, (\bar{x} \vee (y \,\&\, z))} \equiv$$

$$\equiv \overline{(x \,\&\, y)} \vee \overline{(\bar{x}(y \,\&\, z))} \equiv \overline{(x \,\&\, y)} \vee (x \,\&\, \overline{(y \,\&\, z)}) \equiv \overline{(x \,\&\, y)} \vee (x \,\&\, (\bar{y} \vee \bar{z})),$$

что совпадает с формулой, которая получается из формулы 1.5.3 заменой операции & на операцию $\vee$ и наоборот.

При решении задач 1.1 - 1.5 можно придерживаться следующей методики. Одну задачу разобрать подробно у доски. Затем доску разделить на две части и пригласить к ней двух студентов, которые будут решать одновременно еще две задачи. Всю группу разбить на две подгруппы, каждая из которых будет следить за решением задачи у доски своим представителем. Тем самым будет более широкий охват заданий, у доски поработает большее количество студентов и будет экономия времени.

ОТЧЕТНОСТЬ ЗА ЗАНЯТИЕ

В результате выполненной работы заполняется бланк отчёта, в котором должны быть отражены:

1. Тема, программа и цель работы.
2. Задание для практического занятия.
3. Решение задач.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Всякое ли предложение естественного языка является высказыванием?
2. Что означает союз «или» в неисключающем смысле?
3. Дать словесные определения логических операций над высказываниями и привести их обозначения.
4. Определить понятие формулы алгебры высказываний.
5. Какая формула называется тавтологией?
6. Какая формула называется тождественно ложной?
7. Какая формула называется выполнимой?
8. Какие формулы называются равносильными?
9. Записать основные законы логики (тавтологии).
10. Сформулировать закон двойственности для булевых формул.

2 СИНТЕЗ ДИСКРЕТНЫХ АВТОМАТОВ БЕЗ ПАМЯТИ

Цель работы:

Синтезировать структуру комбинационного цифрового устройства и исследовать функционирование его компьютерной модели.

Методические указания по выполнению задания на практическое занятие

Приведём пример выполнения практического занятия для варианта № 30. Заданная логическая функция принимает значения 1 на наборах: 2, 3, 5, 8, 9, 10 , 14, 15.

Минимизацию логической функции проведем графическим методом. Карта Карно для нахождения минимальной дизъюнктивной нормальной формы (МДНФ) заданной функции приведена на рисунке 1.

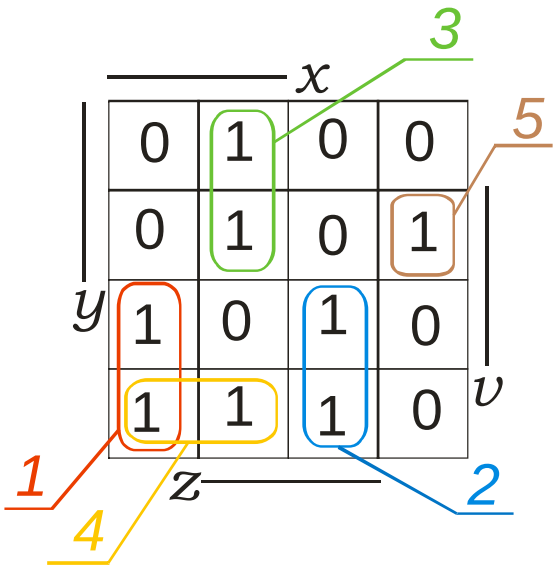


Рисунок 2.1 – Карта Карно для МДНФ

МДНФ для приведенной логической функции будет иметь вид:

$$f = \bar{x}\bar{y}z \vee \bar{x}y\bar{z} \vee xyz \vee x\bar{y}\bar{v} \vee \bar{x}yz\bar{v}. \quad (1)$$

Карта Карно для нахождения минимальной конъюнктивной нормальной формы (МКНФ) заданной функции приведена на рисунке 2.

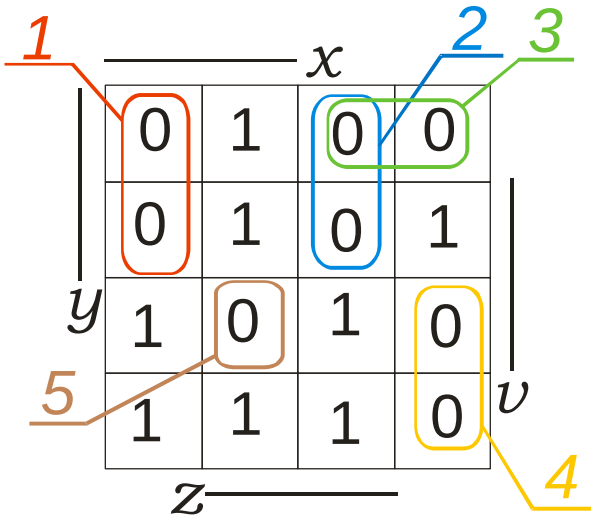


Рисунок 2.2 – Карта Карно МКНФ

МКНФ для приведенной логической функции будет иметь вид:

$$f = (\bar{x} \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (x \vee \bar{y} \vee v) \wedge (x \vee y \vee z) \wedge (\bar{x} \vee y \vee \bar{z} \vee \bar{v}). \quad (2)$$

Для синтеза логической схемы комбинационного цифрового устройства можно выбрать любую из минимальных форм так как они равнозначны. Логическая схема разрабатываемого комбинационного цифрового автомата приведена на рисунке 2.3.

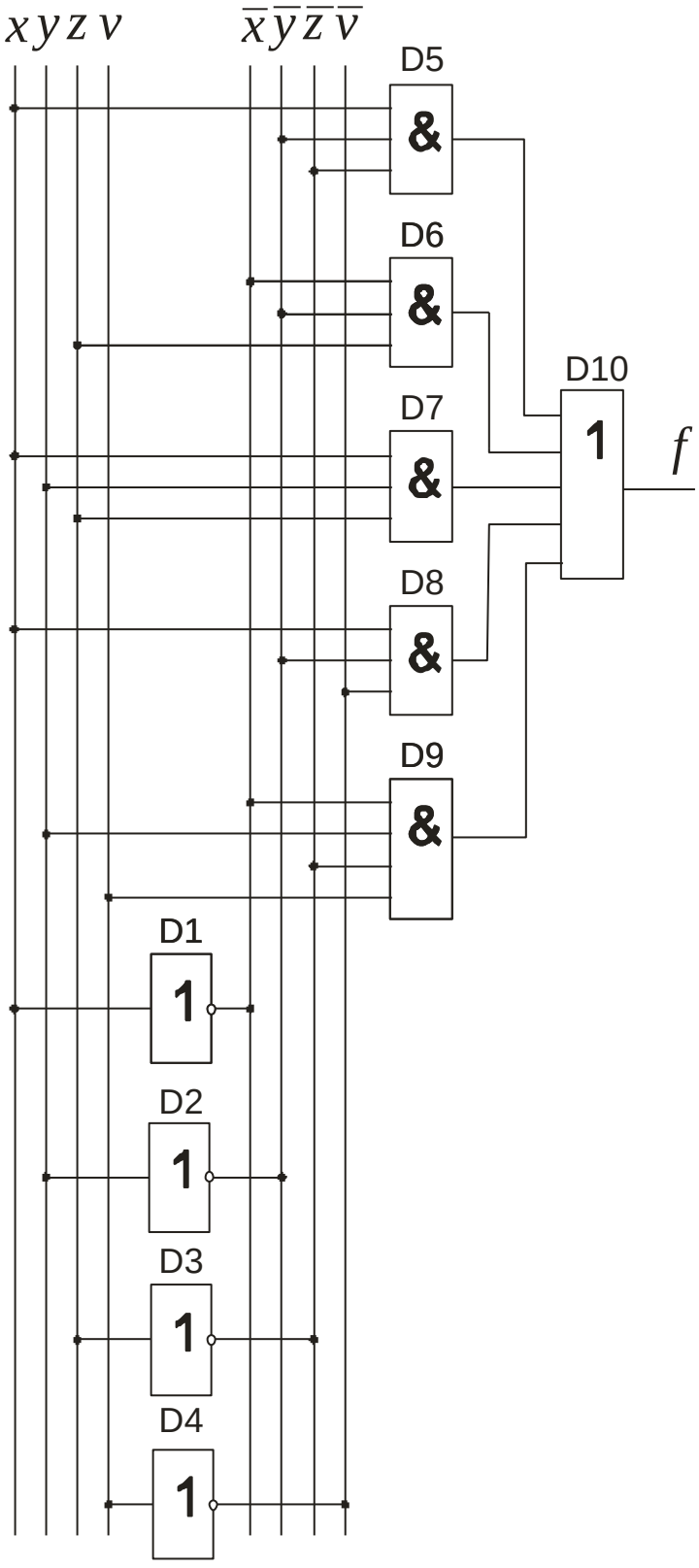


Рисунок 2.3 – Логическая схема комбинационного цифрового автомата

Для моделирования разработанной схемы необходимо собрать ее в средстве Simulink программы MatLab.

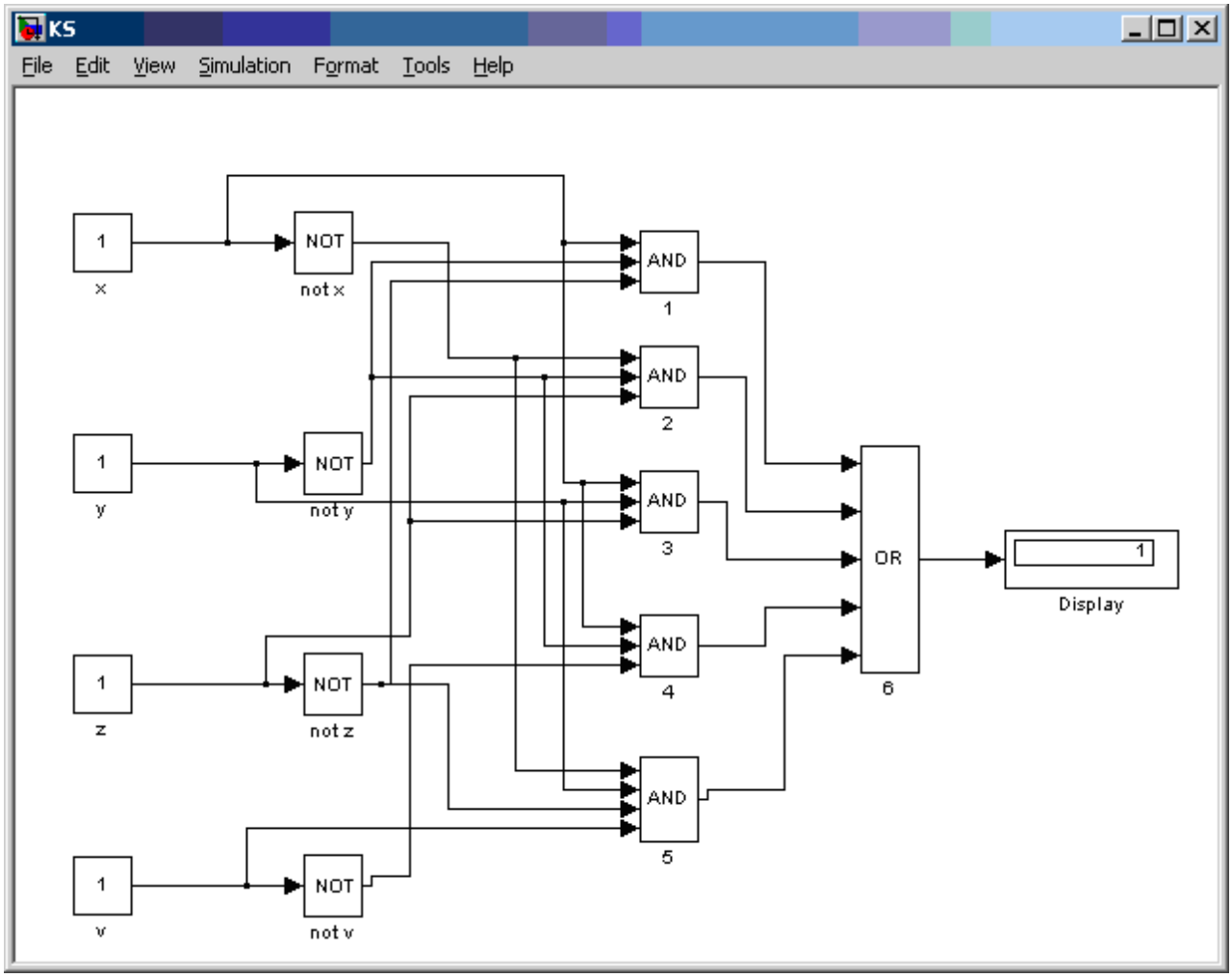


Рисунок 2.4 – Модель комбинационного цифрового автомата в Simulink

В приведённой на рисунке 2.3 схеме использованы элементы: Constant (для задания значений логических переменных), Display (для отображения значения полученной функции) и Logical Operator (логические операторы). Недостаток приведенной схемы заключается в том, что исследовать работу схемы можно только в статическом режиме, меняя значения логических переменных в блоке Constant и запуская моделирование кнопкой Start .

Для исследования приведённой модели в динамическом режиме (рисунок 5) необходимо блоки Constant заменить на блоки From File (из файла), а Display на Scope (осциллограф).

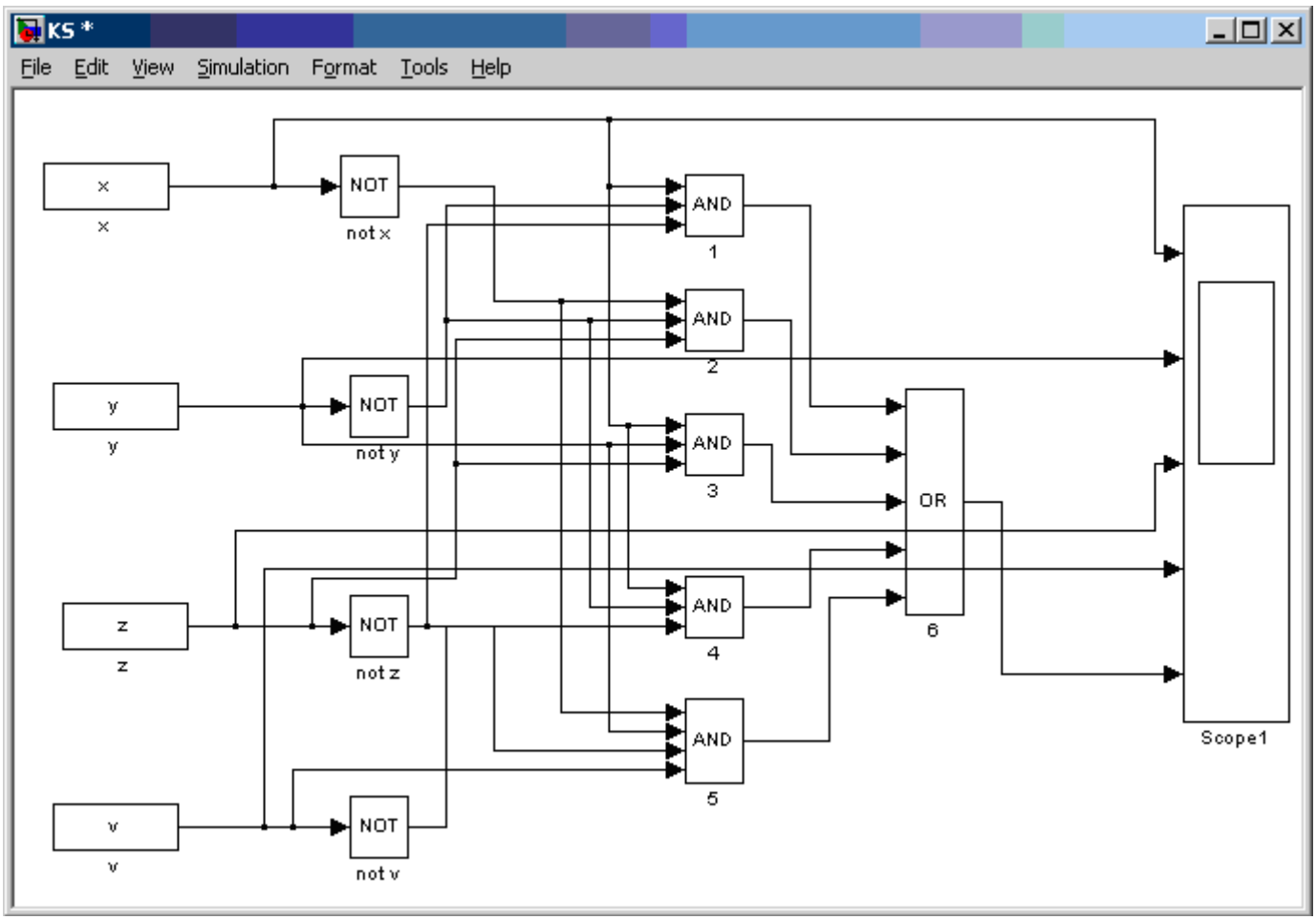


Рисунок 2.5 – Модель комбинационного цифрового автомата для исследования в динамическом режиме

Для задания параметров блока From File необходимо создать файл с переменной, которая будет описывать ее изменение во времени. Каждая переменная будет представлять собой матрицу первая строка которой описывает временные отсчеты, а вторая содержит значение переменной.

```
>> x=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15;0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1]
>> y=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15;0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1]
>> z=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15;0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1]
>> v=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15;0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1]
```

После выполнения этих команд в рабочей области будет сформировано четыре переменные x, y, z, v (рисунок 2.6).

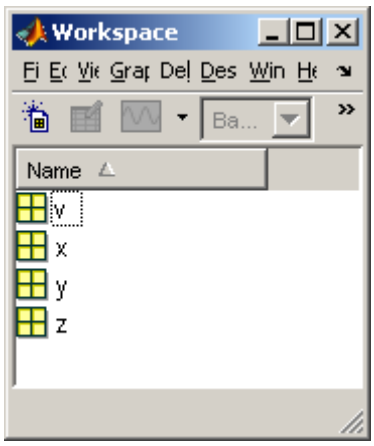


Рисунок 2.6 – Окно рабочей области

Затем необходимо сохранить каждую переменную в файл с расширением *.mat. Для этого необходимо кликнуть правой кнопкой мышки по переменной в рабочей области и выполнить команду Save as … Изменения можно отследить в окне текущего каталога (рисунок 2.7).

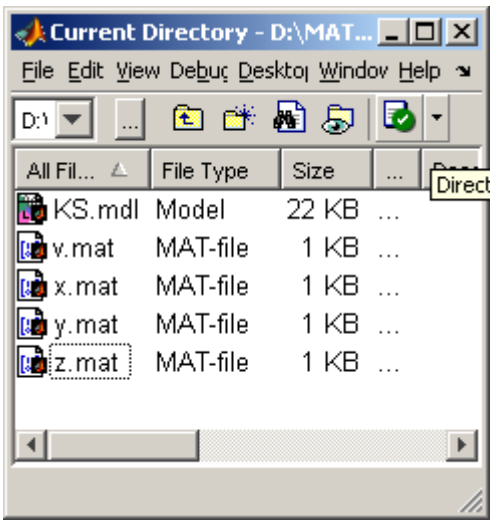


Рисунок 2.7 – Окно рабочей области

В параметрах блока From File необходимо указать имя файла на который ссылается блок и параметр единицы времени, который равен 1 (рисунок 8).

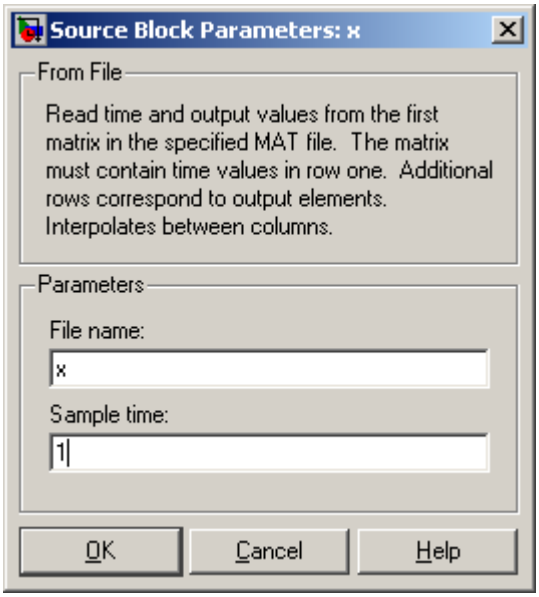


Рисунок 2.8 – Настройка параметров блока From File

В параметрах блока Scope необходимо указать количество входов – 5 (4 переменных + выход схемы) и промежуток времени, в течении которого будет производится моделирование (рисунок 2.9).

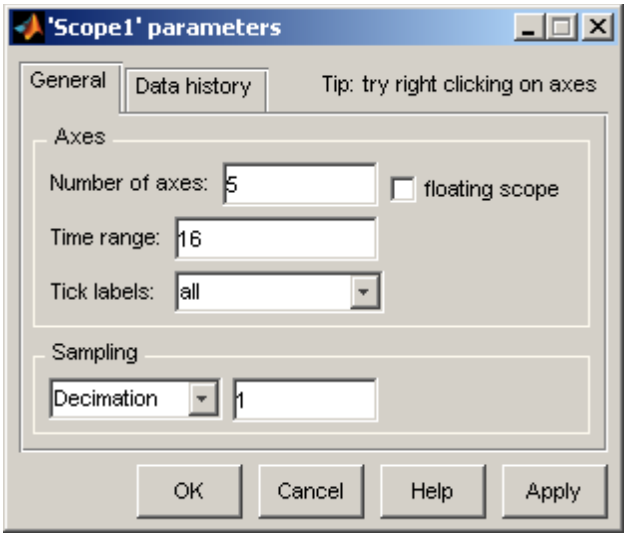


Рисунок 2.9 – Настройка параметров блока Scope

В параметрах модели (Simulation→Configuration Parameters) необходимо указать время начала и окончания моделирования (рисунок 10).

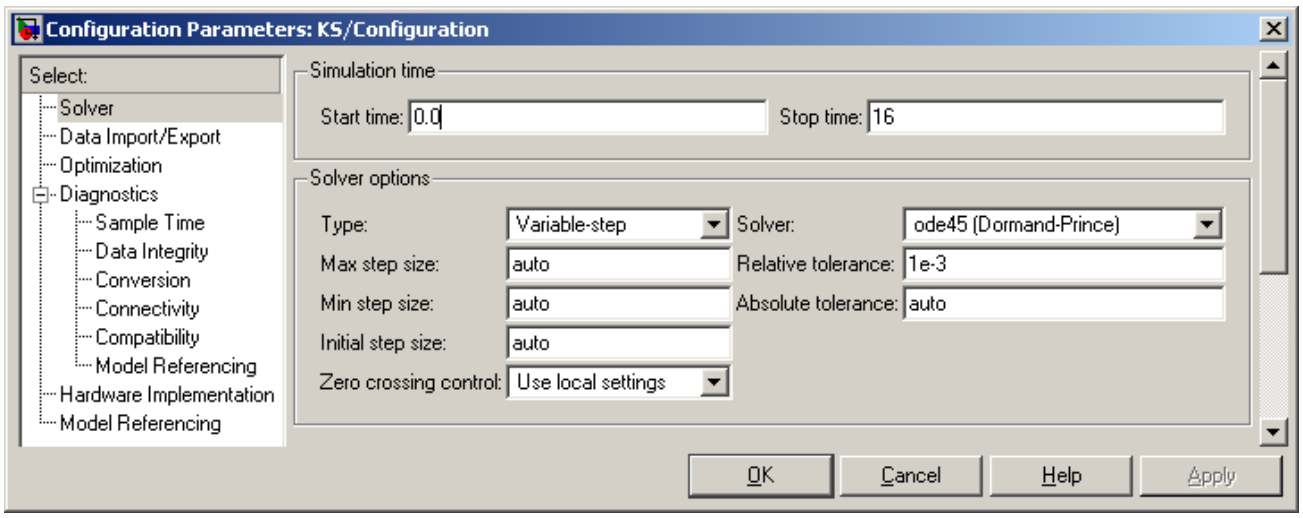


Рисунок 2.10 – Настройка параметров модели

После проведения моделирования (кнопка Start) необходимо проанализировать полученные временные эпюры блока Scope (рисунок 2.11)

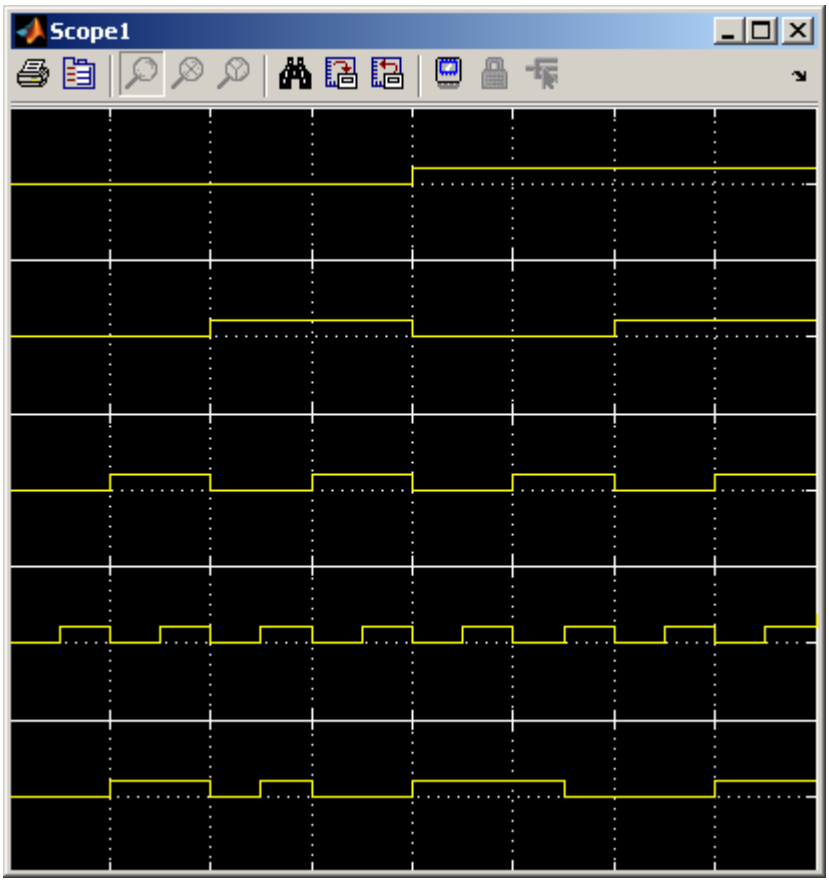


Рисунок 2.11 – Временные эпюры блока Scope

и убедиться в правильности синтеза логической схемы цифрового комбинационного устройства.

Задание для практического занятия:

Получить минимальную форму логической функции, заданной номерами наборов, на которых она принимает значение логической единицы (таблица 1).

Синтезировать схему комбинационного цифрового устройства, заданного логической функцией.

Провести моделирование разработанного комбинационного устройства с использование средства Simulink программы MATLAB.

Варианты заданий

Вариант	**Номера наборов, на которых ЛФ равна 1**
1	3, 4, 5, 6, 10, 12, 13, 15
2	1, 2, 4, 5, 6, 12, 13, 14, 15
3	0, 1, 2, 4, 4, 8, 10, 13, 14
4	0, 1, 4, 8, 9, 14, 15
5	1, 2, 5, 7, 9, 10, 11, 13
6	1, 4, 5, 6, 7, 8, 9, 15
7	2, 5, 6, 7, 10, 11, 12, 14
8	0, 1, 3, 6, 8, 11, 12, 14
9	1, 2, 4, 5, 6, 11,12, 13
10	1, 3, 7, 9, 11, 12, 14
11	3, 4, 5, 7, 8, 11, 12, 15
12	0, 1, 4, 6, 7, 11, 12, 13
13	0, 1, 3, 4, 5, 8, 10 12
14	0, 4, 5, 7, 11, 13, 14
15	1, 3, 5, 6, 9,13, 14
16	0, 1, 2, 6, 9, 10, 13, 14
17	0, 3, 7, 8, 9, 10, 11
18	2, 3, 5, 6, 7, 10, 11, 13
19	2, 4, 5, 7, 9, 10, 13,15
20	0, 1, 7, 8, 9, 11,14
21	2, 6, 7, 9, 11, 13, 14, 15
22	0, 1, 3, 4, 8, 9,13,15
23	3, 5, 6, 7, 9, 11, 12, 15
24	0, 3, 4, 6, 8,12, 14, 15
25	0, 2, 4, 7, 8, 10, 11, 12,15
26	1, 2, 3, 6, 8, 9, 10, 12, 15
27	0, 2, 3, 10 , 11,12, 13, 14
28	0, 2, 4, 5, 6, 8, 10, 13, 15
29	0, 3, 7, 8, 9, 10, 14, 15
30	2, 3, 5, 8, 9, 10 , 14, 15

Содержание отчета по практическому занятию:

1. Задание.
2. Минимизация заданной логической функции.
3. Схема комбинационного цифрового устройства, заданного логической функцией.
4. Результаты моделирования функционирования разработанного цифрового устройства.

3 СТРУКТУРНЫЙ СИНТЕЗ КОНЕЧНЫХ АВТОМАТОВ

Цель работы: закрепить знания, полученные на лекции, посвящённой автоматам с памятью, синтезировать конечный автомат согласно варианту, собрать его компьютерную модель и исследовать ее работу.

Пример выполнения практического занятия

Синтезировать конечный автомат Мили, который под воздействием входного сигнала X_0 не изменяет своего состояния, при подаче каждого сигнала X_1 последовательно переходит в состояние a_0, a_1, a_2, a_3, a_4, a_5, a_0, a_1, a_2,... и т.д., выдавая при переходе из a_5 в a_0 выходной сигнал Y_1. Во всех остальных случаях автомат выдаёт сигнал Y_0.

Синтез конечного автомата проведём придерживаясь следующей методики:

1. Задание автомата в канонической форме.
2. Выбор структуры сигналов на физических входах и выходах.
3. Кодирование входных, выходных сигналов и состояний автомата в соответствии с выбором сигналов.
4. Выбор типа элементарных автоматов и базиса.
5. Составление кодированных таблиц переходов и выходов (кодированного графа) автомата.
6. Составление таблиц функций возбуждения элементарных автоматов, входящих в синтезируемый конечный автомат.
7. Получение минимальных форм этих функций и функций выходов.
8. Выбор типа логических элементов и выражение полученных минимальных форм с учётом базиса.
9. Составление структурной схемы конечного автомата.
10. Проверка функционирования схемы.

1.Зададим закон функционирования автомата в канонической форме

В соответствии с формулировкой, приведенной в примере, построим таблицу переходов и выходов (таблица 3.1), или граф (рисунок 3.1).

Таблица 3.1 – Таблица переходов и выходов конечного автомата

a_j / X_i	a_0	a_1	a_2	a_3	a_4	a_5
X_0	a_0 / Y_0	a_1 / Y_0	a_2 / Y_0	a_3 / Y_0	a_4 / Y_0	a_5 / Y_0
X_1	a_1 / Y_0	a_2 / Y_0	a_3 / Y_0	a_4 / Y_0	a_5 / Y_0	a_0 / Y_1

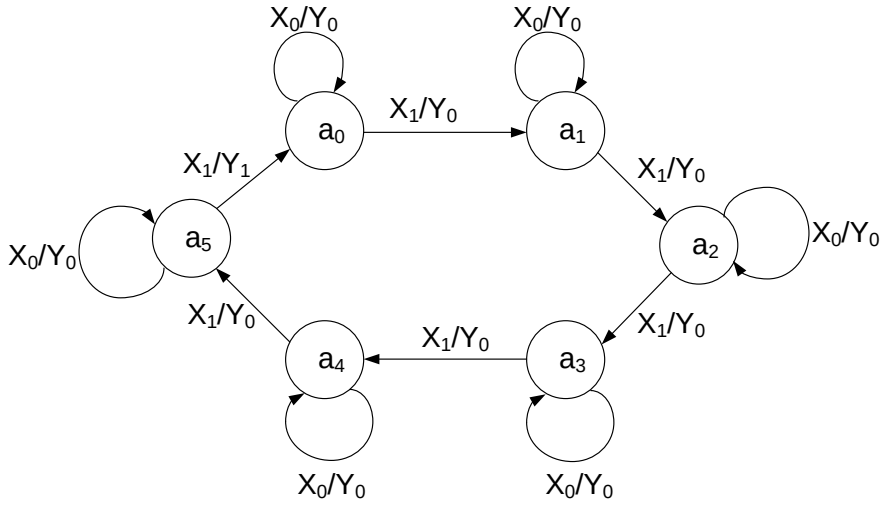


Рисунок 3.1 – Граф конечного автомата

2. Выбор структуры сигналов

В качестве физических сигналов выбираем электрические сигналы, способные принимать любое из двух значений 0 или 1.

3. Кодирование входных и выходных сигналов, и состояний автомата с помощью выбранных физических сигналов

Количество входов и выходов состояний определяется из выражения

$$2^k \geq N > 2^{k-1} \quad k=]\log_2 N[,$$

где N - количество состояний (значений входного или выходного сигналов) синтезируемого автомата;

k - количество двоичных переменных (элементарных автоматов).

В данном случае требуется один вход и один выход. $k_{вх} = 1$, $k_{вых} = 1$.

Т.к. число состояний автоматов 6, то количество элементарных автоматов $k_{эл} = 3$. Кодирование входных и выходных сигналов выполним, например, следующим образом (таблица 3.2,3.3):

Таблица 3.2 – Кодирование входных сигналов. Таблица 5 – Кодирование выходных сигналов

i	
0	
1	

i	
0	
1	

Кодирование состояний автомата (таблица 6) проведём произвольно.

Таблица 3.3 – Кодирование состояний автомата

a_j	Q_1	Q_2	Q_3
a_0	0	0	0
a_1	1	1	0
a_2	0	1	1

a_3	1	0	1
a_4	0	1	0
a_5	1	0	0

Наборы 001, 111 не используются, т.е. запрещены, что может быть использовано при минимизации.

4. Выбор типа элементарного автомата.

В качестве элементарного автомата выберем синхронный JK- триггер (рисунок 3.2).

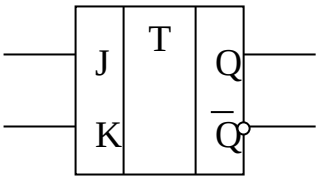


Рисунок 3.2 – Условно-графическое обозначении JK-триггера

Функция переходов JK-триггера приведена в таблице 3.4.

Таблица 3.4 – Функция переходов JK-триггера

$q_j(t)$	$q_k(t)$	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Матрица переходов JK-триггера имеет вид (таблица 3.5).

Таблица 3.5 – Матрица переходов JK-триггера

Q(t)	Q(t+1)	q_j	q_k
0	0	0	α
0	1	1	α
1	0	α	1
1	1	α	0

5. Составление таблиц функций возбуждения элементарных автоматов, входящих в синтезируемый конечный автомат

На основании проведённого кодирования и выбранного типа элементарного автомата (JK-триггера) строим кодированную таблицу переходов, выходов и функций возбуждения.

Кодированная таблица переходов и выходов определяет зависимость состояния элементарных автоматов, входящих в синтезируемый автомат Мили, в момент t+1 и сигналов на его физических выходах в момент *t* от значений сигналов на физических входах и состояний элементарных автоматов в момент времени t.

Чтобы происходили заданные кодированной таблицей переходов и выходов переходы элементарных автоматов, необходимо для каждого из них вырабатывать соответствующие входные воздействия, определяемые функциями возбуждения.

Функции возбуждения каждого элементарного автомата определяют значения его управляющих сигналов в момент времени *t* в зависимости от состояния всех элементарных автоматов и значений входных сигналов синтезируемого автомата в тот же момент времени.

Если Q_1, Q_2,..., $Q_к$ - состояние элементарных автоматов, а V_1, V_2,..., V_m - сигналы на физических входах конечного автомата, то в общем случае функция возбуждения q_{ij} для i-го входа j-го элементарного автомата может быть записана следующим образом:

$$q_{ij}(t) = F_{ij}[V_1(t), V_2(t), \dots , V_m(t), Q_1(t), Q_2(t), \dots , Q_k(t)]. \quad (1)$$

В нашем случае k = 3, а m = 1. Аргументами функций возбуждения являются V(t), $Q_1(t)$ и $Q_2(t)$, и $Q_3(t)$, тогда

$$q_{J1}(t) = F_{J1}[V(t), Q_1(t), Q_2(t), Q_3(t)]$$

$$q_{K1}(t) = F_{K1}[V(t), Q_1(t), Q_2(t), Q_3(t)] \quad (2)$$

Функции возбуждения определяются аргументами, действующими в момент времени *t*, поэтому синтезировать на их основе схемы можно по способу комбинационных схем. Все аргументы действуют в момент времени *t*.

Функция выхода записывается в виде

$$Z(f) = F[V(f), Q_1(t), Q_2(f), Q_3(f)]. \quad (3)$$

Кодированная таблица переходов, выходов и функций возбуждения (таблица 3.6).

Таблица 3.6 – Таблица переходов, выходов и функций возбуждения автомата

V(t)	$Q_1(t)$	$Q_2(t)$	$Q_3(t)$	$Q_1(t+1)$	$Q_2(t+1)$	$Q_3(t+1)$	Z(t)	$q_{J1}(t)$	$q_{K1}(t)$	$q_{J2}(t)$	$q_{K2}(t)$	$q_{J3}(t)$	$q_{K3}(t)$
0	0	0	0	0	0	0	0	0	α	0	α	0	α
1	0	0	0	1	1	0	0	1	α	1	α	0	α
0	1	1	0	1	1	0	0	α	0	α	0	0	α
1	1	1	0	0	1	1	0	α	1	α	0	0	α
0	0	1	1	0	1	1	0	0	α	α	0	α	0
1	0	1	1	1	0	1	0	1	α	α	1	α	0
0	1	0	1	1	0	1	0	α	0	0	α	α	0
1	1	0	1	0	1	0	0	α	1	1	α	α	1
0	0	1	0	0	1	0	0	0	α	α	0	0	α
1	0	1	0	1	0	0	0	1	α	α	1	0	α
0	1	0	0	1	0	0	0	α	0	0	α	0	α
1	1	0	0	0	0	0	1	α	1	0	α	0	α

В последних девяти столбцах приведены значения функций возбуждения, полученные на основании переходов элементарных автоматов Q_1, Q_2 и Q_3 (таблица 3.6) и матрицы переходов (таблица 8).

Каждая из функций возбуждения является неполностью заданной. Миними-

зируем функции возбуждения с помощью карт Карно (рисунок 3.3).

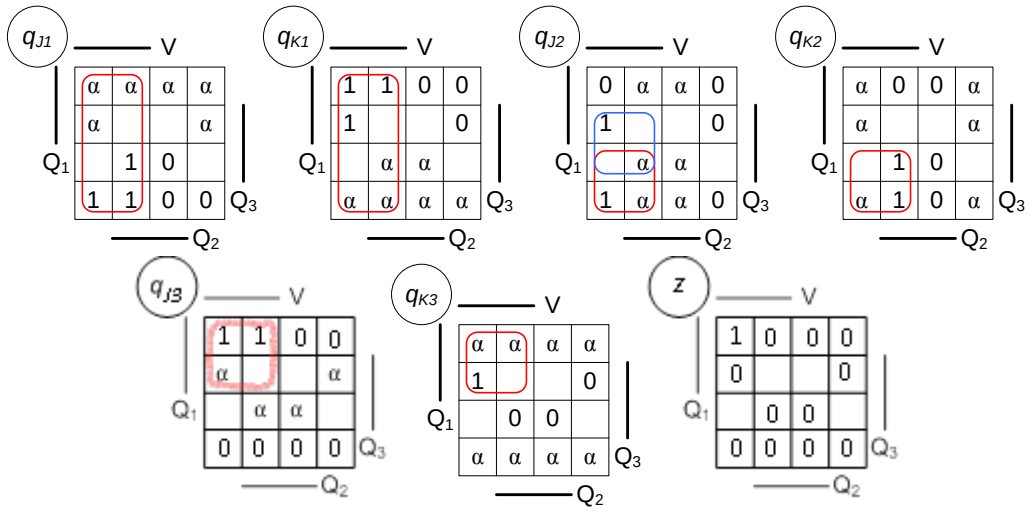


Рисунок 3.3 − Карты Карно

Минимальные формы функций возбуждения автоматов:

$$q_{J1} = q_{K2} = V, \quad q_{J2} = VQ_3 \vee V\overline{Q}_1, \quad q_{K2} = V\overline{Q}_1, \quad q_{J3} = VQ_1, \quad q_{K3} = VQ_1, \quad Z = VQ_1\overline{Q}_2\overline{Q}_3. \quad (4)$$

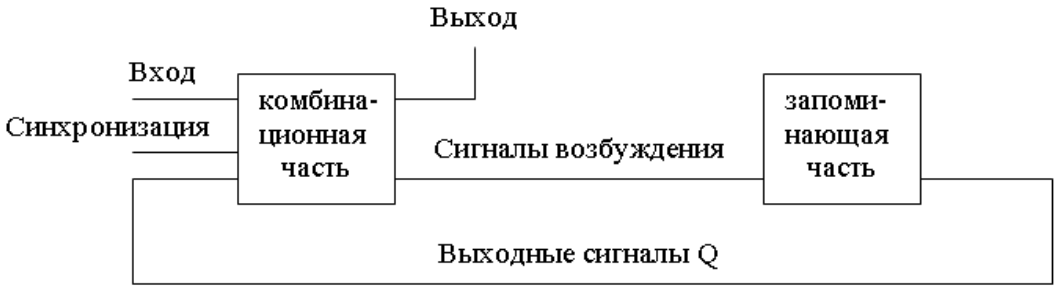


Рисунок 3.4 - Обобщенная схема конечного автомата

На основании минимальных форм функций возбуждения и функции выхода строится структурная схема конечного автомата, обобщенная схема которого состоит из запоминающей и комбинационной частей (рисунок 4).

В нашем примере запоминающая часть включает 3 элементарных автомата (JK-триггера), а комбинационная часть зависит от выбора функционального полного набора логических элементов. Допустим, для данного случая принята основная функционально полная система элементов, состоящая из элементов И, ИЛИ, НЕ. Тогда в основу построения схемы положим выражения (4).

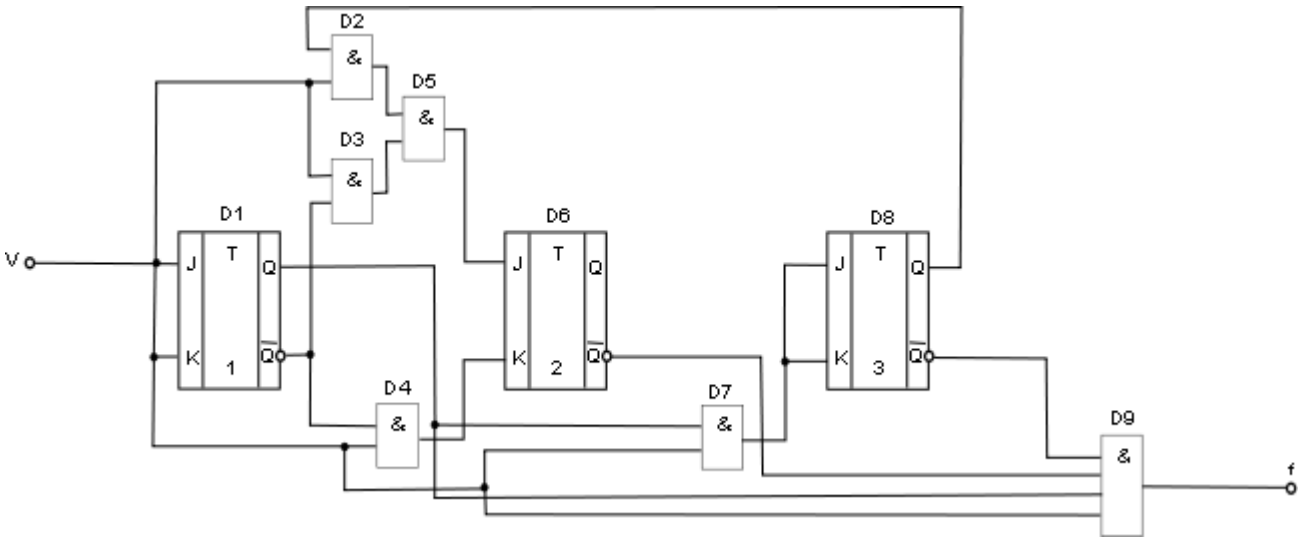


Рисунок 3.5 − Схема конечного цифрового автомата

Модель цифрового автомата в Simulinke представлен на рисунке 3.6.

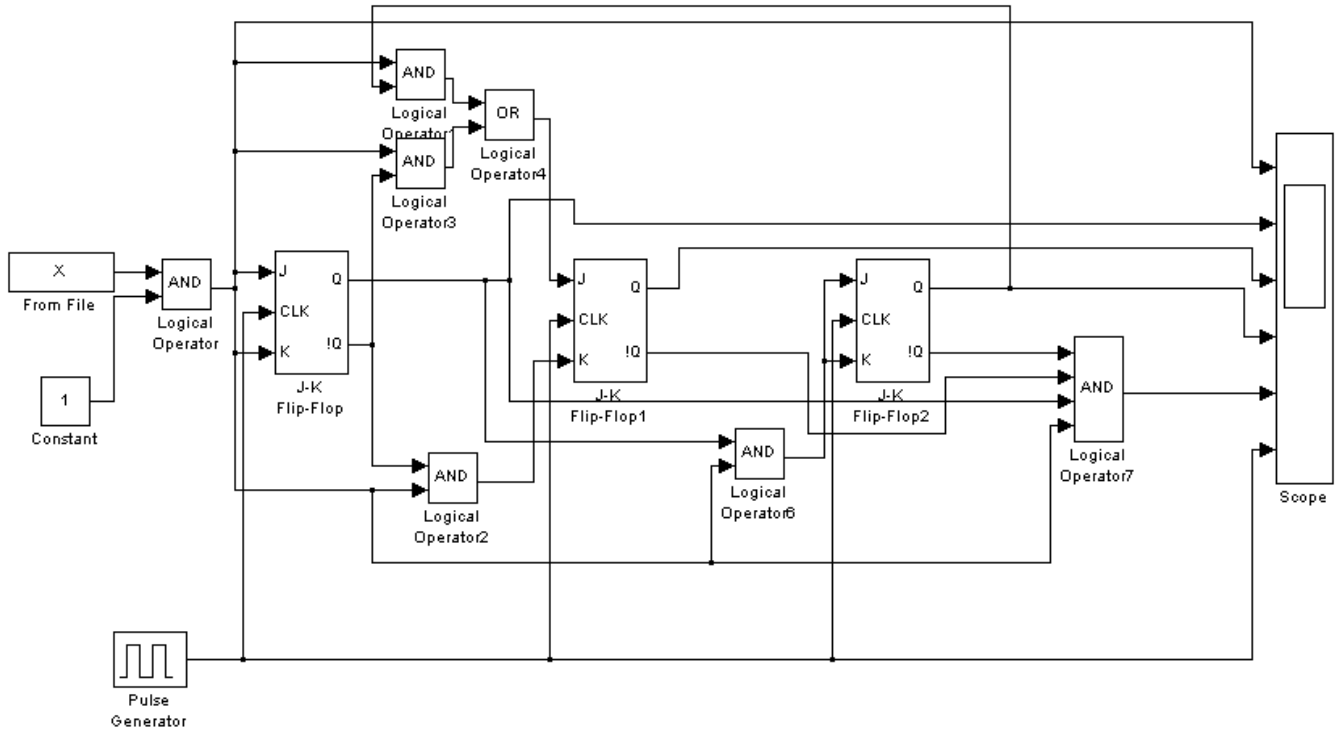


Рисунок 3.6 - Модель цифрового автомата в пакете Simulink

Для проверки функционирования цифрового автомата необходимо сформировать такую последовательность входных сигналов, чтобы автомат прошел все состояния и переходы. В рассматриваемом примере зададим такую последовательность, соответствующую таблице 9. Для задания входной последовательности необходимо в рабочей области Matlab задать:

```
>> V=[0 1 2 3 4 5 6 7 8 9 10 11;0 1 0 1 0 1 0 1 0 1 0 1]
V =0    1    2    3     4     5     6      7      8      9     10
11
   0    1    0   1    0      1      0      1      0      1      0
```

1

и сохранить полученную матрицу в виде файла.

Временные диаграммы (рисунок 7), полученные при симуляции модели, позволяют судить о правильности функционирования автомата, что говорит о верности проведенного синтеза.

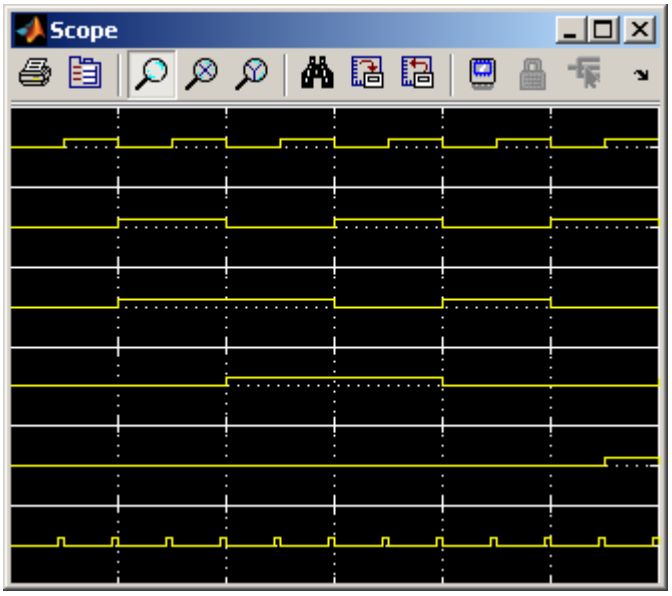


Рисунок 3.6 - Временные диаграммы функционирования цифрового автомата

Задание для практического занятия:

Провести синтез конечного автомата и построить структурную схему, закон функционирования которого задан таблицей переходов и выходов (табл. 3.6). Содержание клеток № 1- 24 этой таблицы приведено в одноименных по номеру столбцах табл. 3.7.

Таблица 3.6.

a_j / x_i	a_0	a_1	a_2	a_3
x_0	1 / 13	2 / 14	3 / 15	4 / 16
x_1	5 / 17	6 / 18	7 / 19	8 / 20
x_2	9 / 21	10 / 22	11 / 23	12 / 24

Содержание отчета по практическому занятию:

1. Задание на практическое занятие.

2. Синтез структуры конечного автомата.
3. Компьютерная модель исследуемого устройства и временные диаграммы его функционирования.

Таблица 3.7.

№ вари-анта	№ клеток таблицы переходов и выходов																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	a_0	a_1	a_2	a_3	a_1	a_2	a_3	a_0	a_0	a_1	a_2	a_3	y_1	y_2	y_3	y_0	y_0	y_0	y_1	y_2	y_1	y_0	y_2	y_1
2	a_1	a_2	a_3	a_0	a_2	a_3	a_0	a_1	a_2	a_3	a_0	a_2	y_3	y_0	y_1	y_2	y_3	y_0	y_1	y_3	y_2	y_0	y_1	y_3
3	a_2	a_3	a_0	a_1	a_2	a_0	a_3	a_0	a_0	a_1	a_2	a_0	y_3	y_0	y_1	y_2	y_3	y_0	y_1	y_3	y_2	y_0	y_1	y_3
4	a_3	a_0	a_1	a_2	a_0	a_1	a_1	a_2	a_3	a_0	a_1	a_1	y_0	y_1	y_2	y_3	y_0	y_1	y_2	y_3	y_0	y_1	y_2	y_3
5	a_0	a_0	a_0	a_1	a_1	a_2	a_3	a_0	a_1	a_2	a_3	a_0	y_1	y_2	y_3	y_0	y_0	y_2	y_3	y_1	y_0	y_2	y_3	y_0
6	a_2	a_1	a_3	a_0	a_1	a_3	a_0	a_2	a_2	a_0	a_0	a_1	y_2	y_1	y_1	y_1	y_0	y_0	y_0	y_2	y_3	y_3	y_0	y_1
7	a_0	a_1	a_2	a_3	a_0	a_0	a_1	a_1	a_2	a_3	a_1	a_0	y_3	y_0	y_0	y_0	y_1	y_2	y_0	y_3	y_3	y_3	y_1	y_2
8	a_1	a_1	a_2	a_3	a_2	a_3	a_0	a_1	a_2	a_1	a_0	a_1	y_1	y_2	y_3	y_1	y_2	y_3	y_1	y_2	y_3	y_0	y_0	y_0
9	a_0	a_2	a_3	a_3	a_3	a_0	a_1	a_2	a_1	a_1	a_1	a_0	y_1	y_2	y_0	y_0	y_1	y_2	y_3	y_0	y_3	y_2	y_1	y_3
10	a_2	a_1	a_3	a_3	a_3	a_3	a_3	a_0	a_1	a_2	a_1	a_0	y_1	y_1	y_1	y_2	y_2	y_2	y_3	y_3	y_3	y_0	y_0	y_0
11	a_1	a_0	a_2	a_2	a_0	a_0	a_1	a_1	a_2	a_3	a_0	a_1	y_2	y_3	y_3	y_3	y_3	y_0	y_0	y_0	y_1	y_2	y_3	y_1
12	a_2	a_2	a_2	a_0	a_0	a_1	a_1	a_2	a_2	a_0	a_1	a_3	y_1	y_3	y_0	y_2	y_0	y_2	y_0	y_1	y_0	y_3	y_0	y_2
13	a_3	a_3	a_0	a_0	a_2	a_2	a_2	a_1	a_1	a_1	a_0	a_2	y_1	y_2	y_3	y_3	y_3	y_0	y_0	y_0	y_2	y_2	y_1	y_1
14	a_2	a_2	a_0	a_0	a_3	a_2	a_3	a_2	a_3	a_1	a_3	a_2	y_2	y_2	y_2	y_3	y_3	y_3	y_0	y_1	y_2	y_3	y_0	y_1
15	a_1	a_1	a_1	a_1	a_0	a_0	a_0	a_2	a_2	a_2	a_3	a_3	y_1	y_3	y_0	y_2	y_1	y_3	y_0	y_2	y_1	y_3	y_2	y_0
16	a_3	a_2	a_1	a_0	a_0	a_0	a_1	a_1	a_2	a_2	a_1	a_3	y_2	y_0	y_1	y_3	y_1	y_0	y_1	y_3	y_0	y_1	y_1	y_2
17	a_0	a_1	a_2	a_2	a_2	a_1	a_1	a_0	a_0	a_1	a_2	a_3	y_1	y_3	y_2	y_2	y_0	y_2	y_0	y_3	y_0	y_1	y_2	y_1
18	a_3	a_0	a_1	a_1	a_1	a_2	a_3	a_3	a_3	a_0	a_0	a_0	y_0	y_1	y_0	y_2	y_0	y_3	y_0	y_1	y_2	y_3	y_0	y_1
19	a_1	a_2	a_3	a_0	a_0	a_0	a_1	a_1	a_2	a_3	a_3	a_1	y_1	y_0	y_3	y_3	y_3	y_2	y_1	y_0	y_1	y_1	y_0	y_1

.

4 СИНТЕЗ РЕГИСТРОВ

Цель занятия:

Закрепить знания по регистрам, полученные на лекциях, получить практические навыки и умения в сборке, отладке и экспериментальном исследовании регистров в потенциальной системе элементов.

Методические указания по выполнению практического занятия

Регистром называется цифровое устройство, предназначенное для записи временного хранения и выдачи кода числа, а также для выполнения некоторых логических преобразований над кодами чисел.

В соответствии с назначением различают параллельные регистры и регистры сдвига.

При построении регистров в качестве запоминающих элементов обычно используют триггеры, количество которых определяет разрядность регистра. Как правило, регистры снабжают дополнительными цепями и логическими элементами, позволяющими выполнять несколько микроопераций таких как установка регистра в нуль, приём, выдачи и сдвиг, преобразование кодов, поразрядные операции логического сложения, логического умножения, суммирования по модулю два (сравнения) кодов.

Для выполнения указанных микроопераций в регистрах предусматривается наличие управляющих сигналов, которые подаются на соответствующие шины.

Конструктивно регистр может быть выполнен как в виде совокупности триггеров и логических элементов, определённым образом соединённых на печатной плате, так и в виде микросхемы, сформированной на единой подложке (Рисунок 4.1).

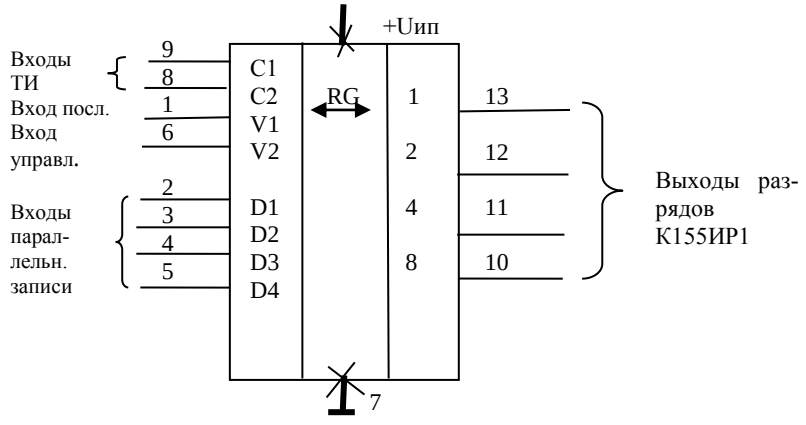


Рисунок 4.1 –Условное графическое обозначение регистров

Приём кода в регистр можно осуществить параллельно или последовательно.

Возможны два способа записи информации в параллельный регистр:

1. с предварительной установкой регистра в нуль.
2. без предварительной установки регистра в нуль (парафазным к дом X_i $\overline{X_i}$).

На Рисунок 4.2. приведена функциональная электрическая схема параллельного регистра, в котором возможна запись информации первым способом.

Перед записью в регистр кода X управляющим сигналом «Уст.0» все триггеры устанавливаются в нулевое состояние. Разряды кода X_i подводятся к триггерам, затем управляющим сигналом «Запись» осуществляется занесение этого кода в регистр. Записанный код может храниться сколько угодно долго (конечно при наличии напряжения питания).

На Рисунок 4.3. показана функциональная электрическая схема параллельного регистра с записью информации парафазам входным кодом X_i Xi.

В процессе передачи информации с одного регистра RGY на другой RGX возможно выполнение указанных выше поразрядных логических операций. Вид операции определяется организацией схем вентилей обмена информацией между регистрами и наличием управляющих сигналов.

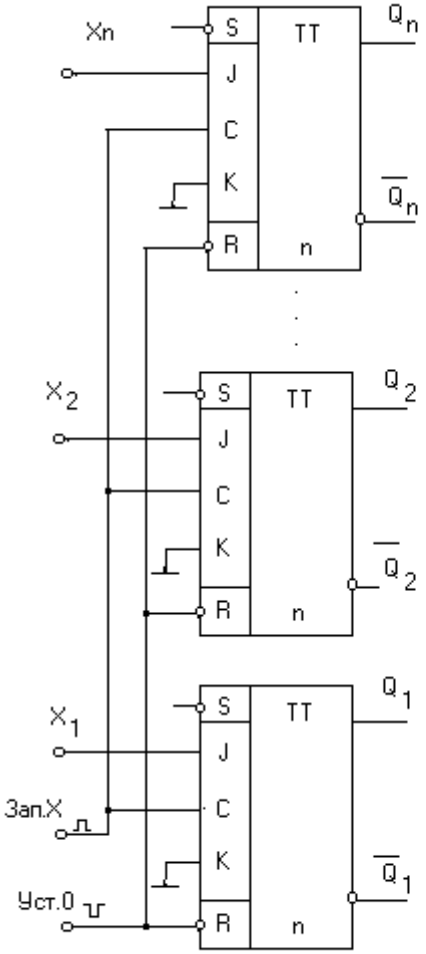


Рисунок 4.2 - Запись информации в регистр параллельного действия с предва-

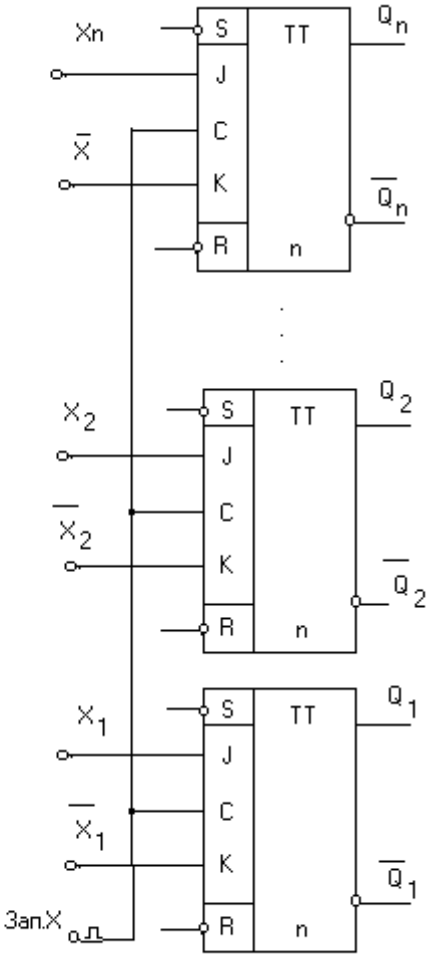


Рисунок 4.3 – запись информации в регистр параллельного действия парафазным

рительной установкой в кодом ноль

На Рисунок 4.4. приведен один из возможных вариантов функциональной электрической схемы, выполняющей операцию поразрядного логического сложения. Схема реализует операцию «ИЛИ» над разрядами кодов чисел X и Y, представленными двоичными цифрами. Результат выполнения операции устанавливается в i- том разряде регистра RGX:

$Q_i = X_i + Y_i$

Вначале управляющим сигналом «Уст.0» триггеры устанавливаются в нулевое состояние, а затем сигналом «запись» осуществляется запись кода числа X в регистр RGX и , наконец, сигналом «Лог.слож» выполняется операция поразрядного логического сложения чисел X и Y.

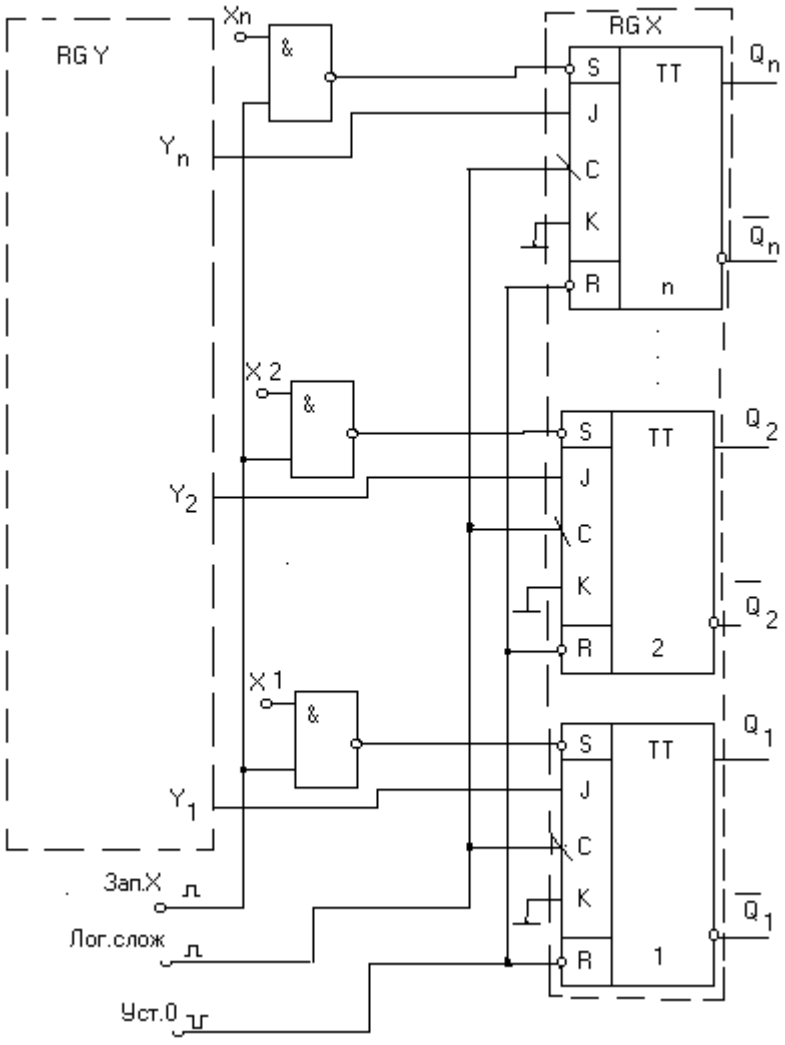


Рисунок 4.4 – Выполнение регистром действия операции поразрядного логического сложения

Поразрядное логическое умножение является логической операцией "И" над i- ми двоичными разрядами кодов чисел X и Y. При этом в i- том разряде регистра GRX устанавливается результат выполнения опера-

ции (Рисунок4.5).

$Q_i = X_i \bullet Y_i$

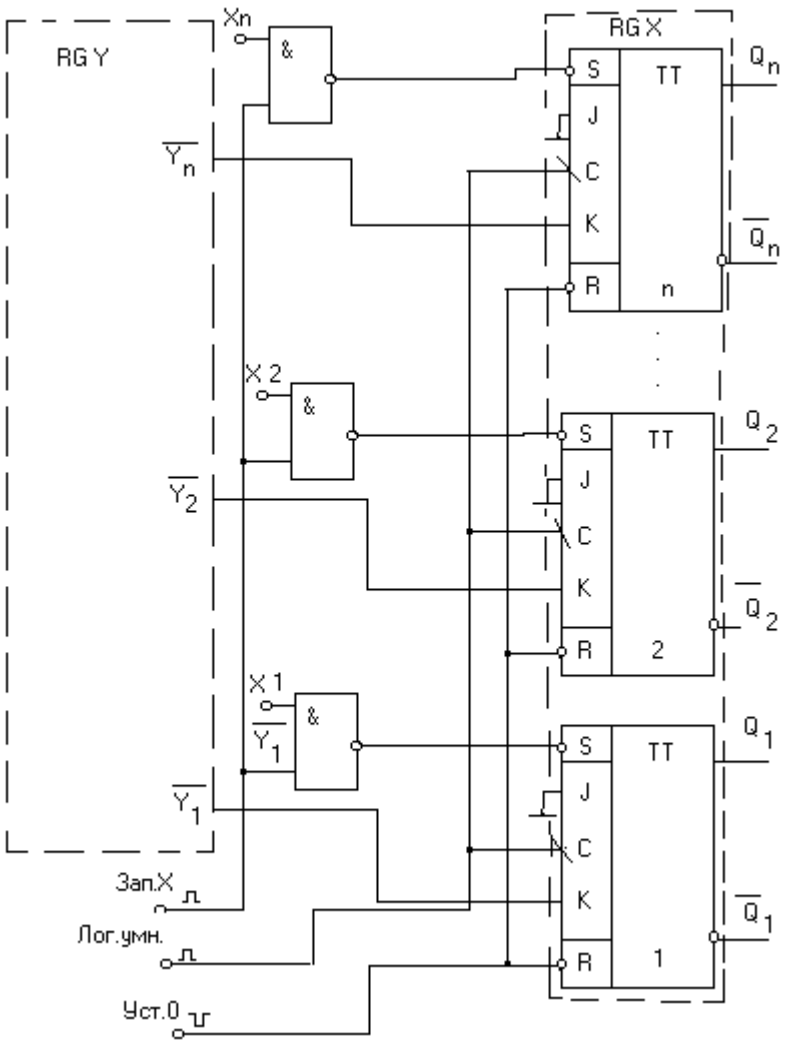


Рисунок 4.5 – Выполнение регистром параллельного действия операции поразрядного логического умножения

Регистры сдвига предназначены для преобразования информации путем ее сдвига под воздействием тактовых импульсов. Такие регистры представляют собой совокупность последовательно соединенных триггеров, как правило, двухступенчатой структуры. Число триггеров определяется разрядностью записываемого слова.

По направлению сдвига информации различают регистры прямого сдвига (вправо, т.е. в сторону младшего разряда), обратного сдвига (влево, т.е. в сторону старшего разряда), реверсивные, допускающие сдвиг в обоих направлениях.

На основе сдвигающего регистра можно построить кольцевой регистр, если входы триггера младшего разряда соединить с выходами триггера старшего разряда. Функциональная электрическая схема такого регистра представлена на Рисунок 4.6

Сдвиг на один разряд выполняется подачей одного импульса на шину сдвига. В момент поступления этого импульса значения сигналов на входах J и K каждого триггера определяют его переход в новое состояние. Это новое состояние каждый триггер принимает по окончании импульса сдвига.

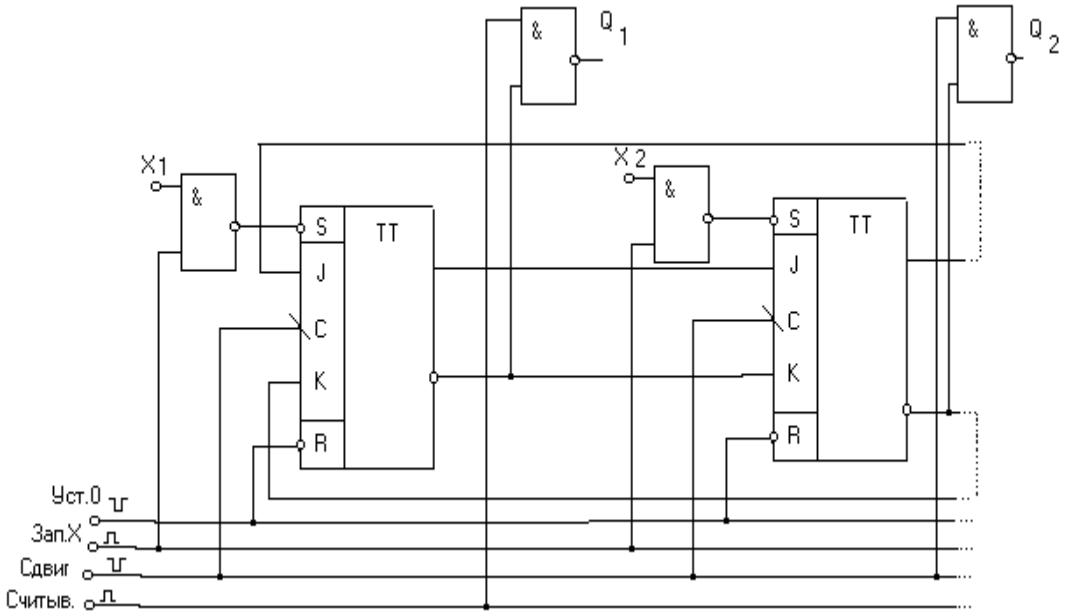


Рисунок 4.6 – Схема регистра сдвига на один разряд вправо

С помощью регистра сдвига, таким образом, можно осуществлять преобразование информации из последовательной формы представления в параллельную. Очевидно, если предусмотрена запись информации параллельным кодом, то можно осуществить преобразование информации из параллельной формы представления в последовательную.

Реверсивные регистры сдвига (многофункциональные) объединяют в себе свойства регистров со сдвигом вправо и со сдвигом влево. Строятся они по тем же схемотехническим принципам, что и рассмотренные регистры, но с использованием дополнительной комбинационной схемы, включающей в себя логические элементы в межразрядных связях.

Обобщенная структурная схема многофункционального регистра сдвига приведена на Рисунок 4.7.

На входы комбинационной части схемы поступают сигналы с выходов триггеров (Q1, Q2, ….. Qn) , а также сигналы управления (X1, X2, ….. Xm). Назначение сигналов управления состоит в том, чтобы из всех возможных микроопераций сдвига, выбрать одну, требуемую в данный момент времени. Таким образом, сигналы, поступающие по шинам управления, организуют определенную цепь сдвига от одних триггеров к другим и запрещают работу всех остальных цепей.

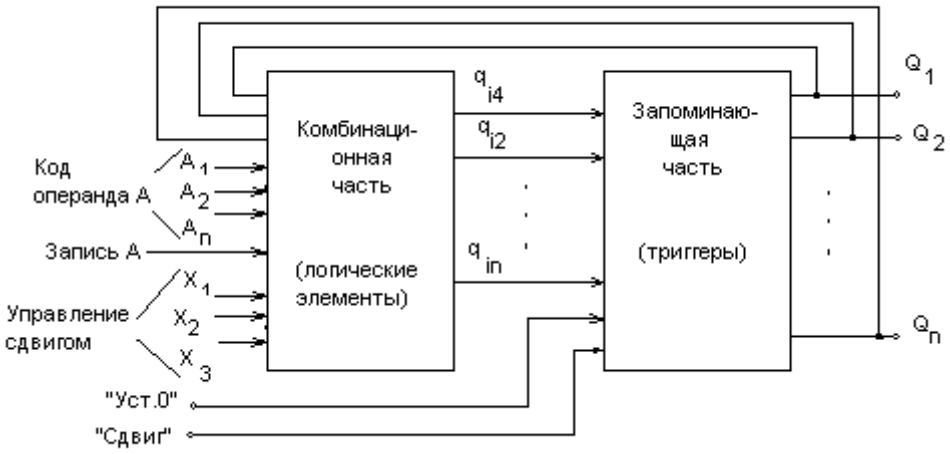


Рисунок 4.7 – Обобщенная схема функционального регистра сдвига

Сигналы с выходов комбинационной части схемы представляют собой функции возбуждения, подаваемые на информационные входы триггеров. Функцию возбуждения 1-го, входа n-го триггера можно написать в следующем виде:

Q (t) = f[X1(t), X2(t),….Xm(t), Q1(t), Q2(t),….Qn(t)].

Таким образом, функции возбуждения будут логическими функциями от (m+n) аргументов. Эти функции могут формироваться комбинационными схемами. Следовательно, если задан тип триггера, то задача логического проектирования многофункционального регистра сдвига сводится составлению функций возбуждения информационных входов каждого триггера, их минимизации и представления в задачном базисе.

Изучим метод синтеза многофункционального регистра сдвига на конкретном примере.

Пусть необходимо синтезировать реверсивный регистр сдвига на один разряд вправо и влево.

Решение.

1. Определение числа шин управления

Это число зависит от числа микроопераций сдвига К, которые должны выполнять проектируемый регистр, и определяется по выражению:

$M = \lceil \log_2 K \rceil$.

выражение в скобках означает округление величины до ближайшего большого целого.

В нашем случае необходимо выполнить две микрооперации: сдвиг вправо и сдвиг влево, следовательно К = 2.

Тогда $m=|\log_2|=1$, т.е. требуется одна шина управления, на которую будет подаваться сигнал управления X. Положим, что при X = 0 осуществляется сдвиг вправо на один разряд, а при X = 1 - влево на один разряд.

2. Построение кодированной, таблицы переходов и функции возбужде-

ния.

Очевидно, достаточно получить выражение для функций возбуждения только i-го триггера постольку структура регистра регулярна (Рисунок 4.8.)

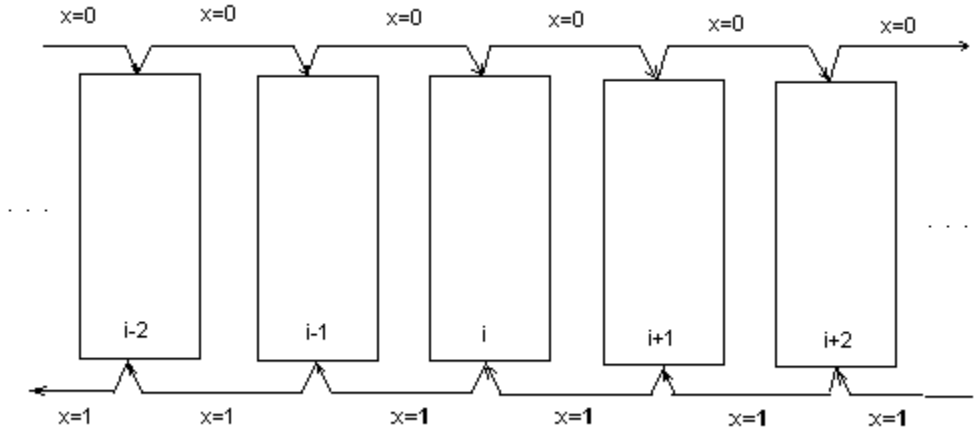


Рисунок 4.8 – Структура регистра сдвига выполняющего две операции сдвига

Следовательно, функцию переходов для 1-го триггера можно представить следующим образом:

$Qi(t+1) = f[X(t), Q_{i-1}(t), Qi(t), Q_{i+1}(t)]$

В качестве элементарного автомата (триггера) в принципе можно использовать любой синхронный триггер. Остановим свой выбор на синхронном JK-триггере. Таблица переходов (табл. 4.1.) и матрица переходов (табл. 4. 2.) имеют вид.

Таблица 4.1.

$q_y(t)$	$Q_k(t)$	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Таблица 4.2.

Q(t) Q(t+1)	q_y (t) q_k (t)
0 - 0	0 -
0 - 1	α_0
1 - 0	1 -
1 - 1	α_1
	α_2 -
	1
	α_3 -
	0

Тогда функции возбуждения для i-го триггера можно представить следующим образом

$$q_{ji}(t) = f[X(t), Q_{i-1}(t), Q_i(t), Q_{i+1}(t)]$$
$$q_{ki}(t) = f[X(t), Q_{i-1}(t), Q_i(t), Q_{i+1}(t)]$$

Составим кодированную таблицу переходов в таблицу функций возбуждения для i-го JK – триггера (табл.4.3.)

Таблица 4.3.

NN наборов	Время t				Время t+1	Время t	
	X	Q_{i-1}	Q_i	Q_{i+1}	Q_i	q_{ji}	q_{ki}
	1	2	3	4	5	6	7
0	0	0	0	0	0	0	α_0
1	0	0	0	1	0	0	α_1
2	0	0	1	0	0	α_2	1
3	0	0	1	1	0	α_3	1
4	0	1	0	0	1	1	α_4
5	0	1	0	1	1	1	α_5
6	0	1	1	0	1	α_6	0
7	0	1	1	1	1	α_7	0
8	1	0	0	0	0	0	α_8
9	1	0	0	1	1	1	α_9
10	1	0	1	0	0	α_{10}	1
11	1	0	1	1	1	α_{11}	0
12	1	1	0	0	0	0	α_{12}
13	1	1	0	1	1	1	α_{13}
14	1	1	1	0	0	α_{14}	1
15	1	1	1	1	1	α_{15}	0

Колонки 1-5 составляют кодированную таблицу переходов 1-го разряда регистра, а 6, 7 - таблицу функций возбуждения i-го JK-триггера, аргументы которых записаны в колонках 1-4.

Рассмотрим порядок заполнения этих таблиц.

Функция переходов Qi(t+l) и функции возбуждения $q_{ni}(t)$ и q_{hi} (t) являются логическими функциями четырех аргументов X, Q_{i-1}, Q_i, Q_{i+1}, поэтому число возможных наборов аргументов равно шестнадцати.

При заполнении пятой колонки Qi(t+l) необходимо учитывать значение сигнала X(t) на шине управления (первая колонка), а также состояния $Qi-_1(t)$(вторая колонка) и Q_{i+1} (t) (четвертая колонка). Согласно принятого условия при X=0 осуществляется сдвиг информации вправо на один разряд, т.е. $Qi(t+1) = Q_{i-1}(t)$, соответственно, при X=1 должен осуществляться сдвиг информации влево на один разряд, т.е. $Qt(t+l) = Q_{i+1}$ (t).

При заполнении таблицы функций возбуждения следует рассмотреть переходы i-го триггера Qi(t) - Qi(t+l) для каждой строки и, воспользовавшись матрицей переходов для JK-триггера (таблица4.2.), вписать значения q_{ji} (t) и $q_{ki}(t)$ для этой строки, чтобы осуществляется заданный переход.

Индексы, стоящие у коэффициентов неопределенности , соответствуют номерам набора аргументов.

3. Минимизация функций возбуждения.

Необходимо отметить, что функции возбуждения q_{ji} (t) и $q_{ki}(t)$ являются неполностью определенными логическими функциями.

Будем полагать, что комбинационная часть многофункционального регистра сдвига будет реализована на элементах универсального базиса "И-НЕ". Поэтому представим функции возбуждения в минимальной дизъюнк-

тивной нормальной форме. Для этого составим карты Карно.

Минимальные дизъюнктивные нормальные формы для функций возбуждения можно получить в том случае, если

$\alpha_0, \alpha_1, \alpha_6, \alpha_7, \alpha_8, \alpha_{11}, \alpha_{12}, \alpha_{15} = 1$, а

$\alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_9, \alpha_{10}, \alpha_{13}, \alpha_{14} = 0$

В результате получим:

$$\text{МДНФ } q_{ji} = X Q_{i+1} + \overline{X} Q_{i-1}$$

$$\text{МДНФ } q_{ki} = X \overline{Q}_{i+1} + \overline{X} \overline{Q}_{i-1}$$

Однако полученные минимальные формы не являются оптимальными для построения схемы регистра. Легко заметить по картам Карно, что функция q_{ki} и q_{ji} являются инверсивными, используя основные законы алгебры логики, докажи это

$q_{ki} = X\overline{Q}_{i+1} + \overline{X}\overline{Q}_{i-1}$ = (дважды применим закон инверсии) = $\overline{\overline{x\overline{Q}_{i+1} + \bar{x}\overline{Q}_{i-1}}}$ =

= $\overline{\overline{X\overline{Q}_{i+1}} \cdot \overline{\overline{X}\overline{Q}_{i-1}}} = \overline{(\overline{X} + Q_{i+1})(X + Q_{i-1})}$ = (1й дистрибутивный закон) =

= $\overline{\overline{X} \cdot X + \overline{X} \cdot Q_{i-1} + X \cdot Q_{i+1} + Q_{i+1} \cdot Q_{i-1}}$ = (правило расширения) =

= $\overline{XQ_{i+1} + \overline{X}Q_{i-1}} = \bar{q}_{ji}, m.e. q_{ki} = \bar{q}_{ji}$

Поэтому в качестве исходных будем использовать следующие выражения:

$q_{ji} = X \cdot Q_{i+1} + \overline{X} \cdot Q_{i-1}, q_{ki} = \bar{q}_{ji}$

Представим функции q_{ji} к q_{ki} в универсальном базисе "И-НЕ" (штрих Шеффера).

$q_{ji} = (X|Q_{i+1}) | [(X|X)|Q_{i-1}]$, $q_{ki} = q_{ji} | q_{ji}$.

4. Построение функциональной электрической схемы многофункционального регистра сдвига

Приведем фрагмент этого регистра на трех триггерах и элементах межразрядных связей (Рисунок4.9.).

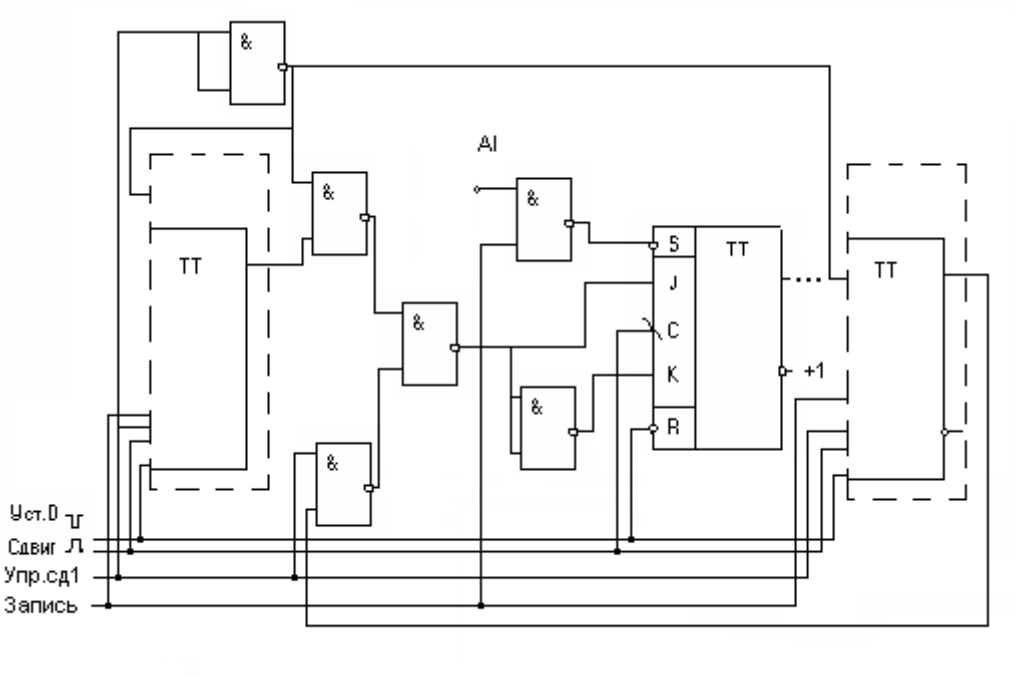


Рисунок 4.9 – Фрагмент регистра сдвига, выполняющего операцию сдвига на 1 разряд влево и 1 разряд вправо

Порядок выполнения работы

Исследовать работу параллельного регистра с приёмом двоичного кода с предварительной установкой нуля во всех разрядах. Информационный код операнда X и сигналы шины управления подавать с ключей. Информационный код операнда X контролировать с использованием счетно-индикаторных ячеек. Контроль записанной информации производится подключением прямых выходов триггеров к счетно-индикаторным ячейкам. Схема испытаний регистра параллельного действия с предварительной установкой в «0» приведена на рисунке 4.10.

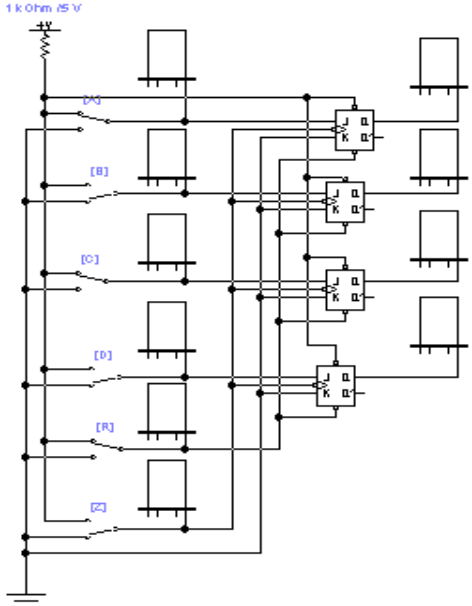


Рисунок 4.10 – Схема испытаний 4^x разрядного регистра параллельного действия с предварительной установкой в ноль

3. Исследовать работу регистра параллельного действия с записью информации парафазным кодом.

Информационный код операнда X и сигнал на шину записи подавать с ключей. Информационный код операнда X контролировать с использованием счетноиндикаторных ячеек. Контроль записанной информации производится подключением прямых выходов триггеров к счетноиндикторным ячейкам.

Схема испытаний регистра параллельного действия с приемом информации парафазным кодом приведена на рисунке 4.4.

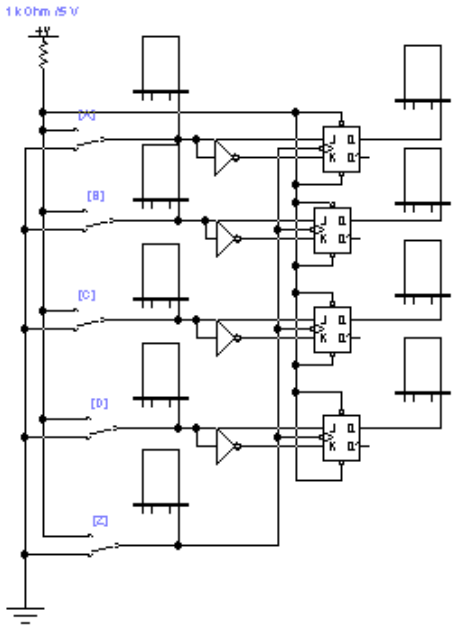


Рисунок 4.11 – Схема испытаний 4^x разрядного регистра параллельного действия с приемом информации парафазным кодом

4. Проверить работу регистров параллельного действия, выполняющие операции поразрядного логического сложения и поразрядного логического умножения.

Схема испытаний регистров приведены на рисунке 4.12 и 4.13 соответственно.

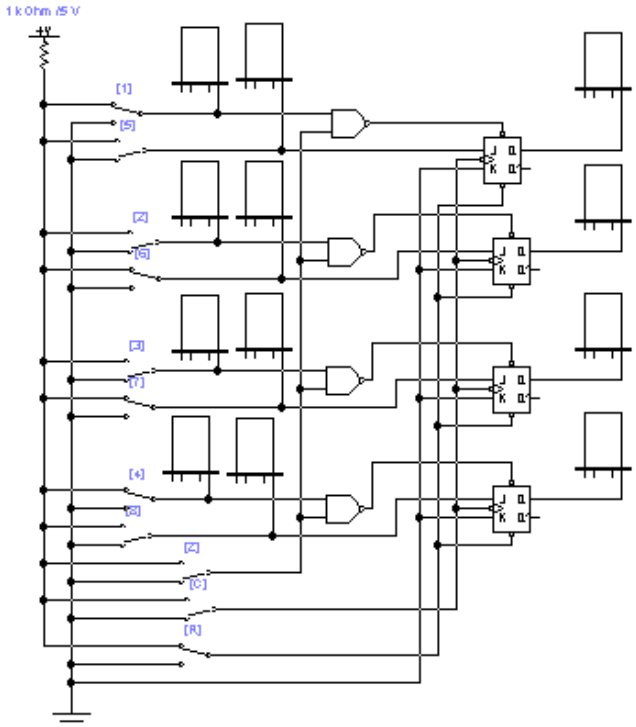


Рисунок 4.12 – Схема испытаний 4^x разрядного регистра параллельного действия выполняющего операцию поразрядного логического сложения

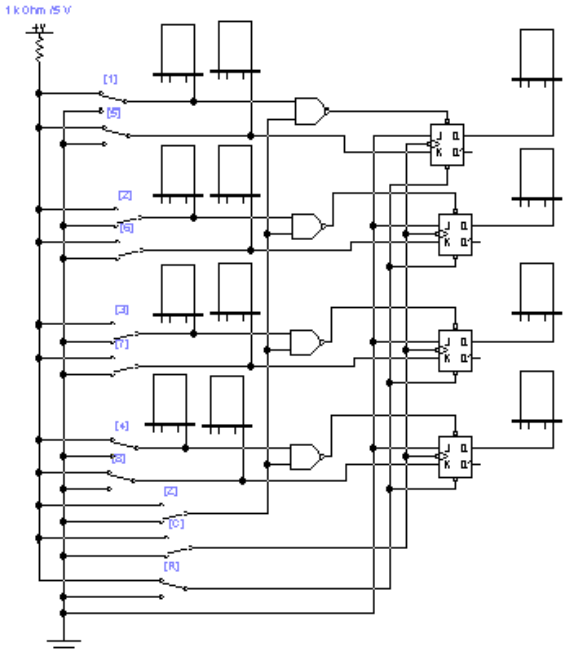


Рисунок 4.13 – Схема испытаний 4^x разрядного регистра параллельного действия выполняющего операцию поразрядного логического умножения

Значение разрядов операнда X (X_1, X_2, X_3, X_4) и операнда У ($У_1$, $У_2$, $У_3$, $У_4$) подавать с ключей.

Сигналы на шины правления «Уст. 0», «Запись X», «Лог.сложение (логическое умножение)» подавать с ключей. Контроль значений разрядов операнда X и операнда У осуществлять с использованием счетно-индикаторных ячеек. Контроль результатов выполняемых операций осуществлять подклю-

чением к прямым выходам триггеров счетно-индикаторных ячеек.

5. Исследовать работу четырехразрядного кольцевого регистра в статическом режиме.

Схема испытаний четырех разрядного кольцевого регистра приведена на рисунке 4.14.

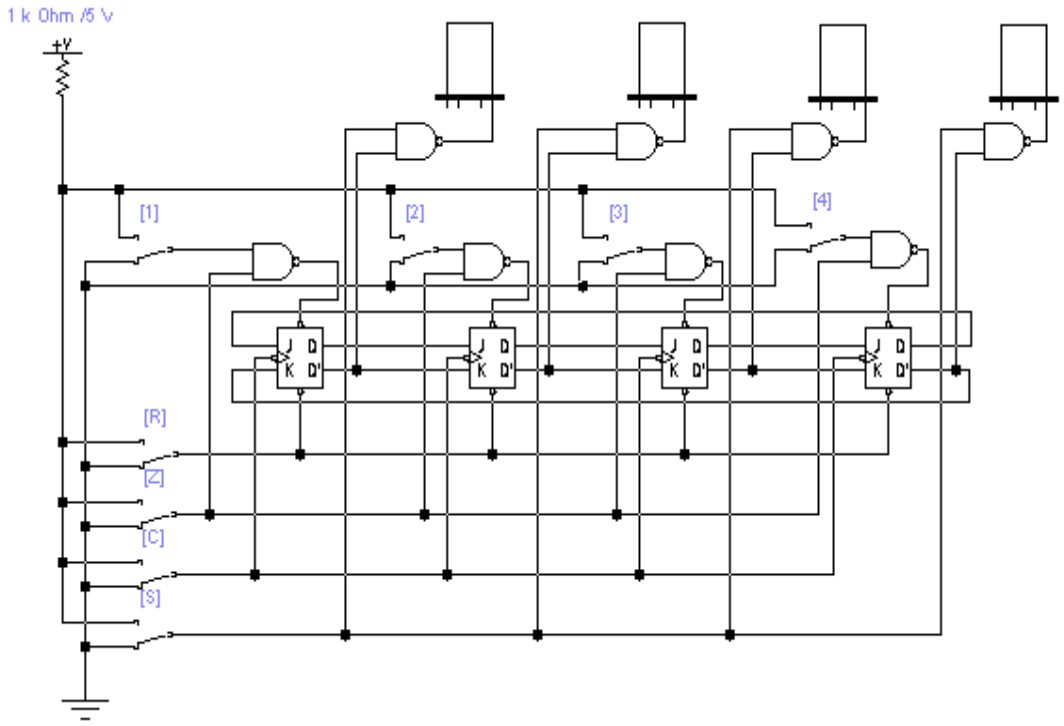


Рисунок4.14 – Схема испытаний 4^x разрядного кольцевого регистра

Управляющие сигналы «Уст.0», «Запись», «Сдвиг», «Считывание» подавать с ключей. Контролировать значения сигналов на шинах управления и значений разрядов кода операнда X с использованием счетно-индикаторных ячеек. Контроль за состоянием триггеров регистра осуществляется подключением прямых выходов триггеров регистра к счетно-индикаторным ячейкам.

Испытание четырёх разрядного кольцевого регистра выполнять в следующей последовательности:

а) кратковременной подачей на шину «Уст. 0» уровня логического «0» перевести все триггеры регистра в «0» состояние;

б) кратковременной подачей на шину «Запись X» уровня логической «1» записать исходный код в регистр.

в) осуществить сдвиг информации на один разряд вправо. (Сдвиг информации осуществляется при изменении потенциала на шине «сдвиг» с «1» на «0») причём информация с последнего четвёртого триггера запишется в первый.

г) подачей уровня логической «1» на шину «считывание» осуществить съем информации в прямом коде с триггером регистра.

6. Исследовать работу синтезированного трех разрядного многофунк-

ционального регистра сдвига выполняющего операцию сдвига на 1 разряд вправо и на один разряд влево приведена на рисунке 4.15.

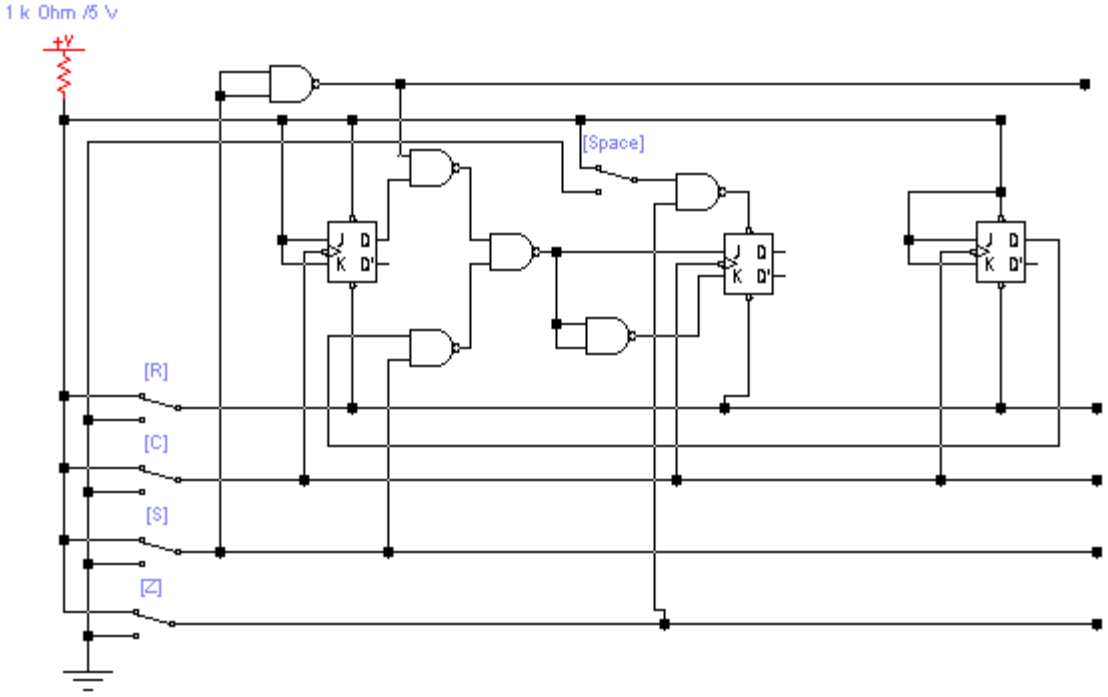


Рисунок 4.15 – Пример схемы испытаний синтезированного регистра сдвига выполняющего операции сдвига на 1 разряд вправо и на один разряд влево

Сигналы на шины управления «уст.0», «сдвиг», «упр.сдвигом», «запись», значений разрядов исходного кода, состояния триггеров регистра контролировать с помощью счетноиндикаторных ячеек.

Испытание синтезируемого многофункционального регистра сдвига осуществлять в следующей последовательности:

а) кратковременной подачей уровня логического «0» состояние;

б) задать значение разряды исходного кода (A_1, A_2, A_3);

в) кратковременной подачей уровня логической «1» на шину «запись» записать исходный код в триггеры регистра;

г) установить сигнал («0» или «1») на шине «упр. сдв.»;

д) осуществить сдвиг информации. Сдвиг на необходимое количество разрядов осуществляется в зависимости от сигнала на шине «упр.сдвигом» при изменении потенциала на шине «сдвиг» с «1» на «0»

е) контроль выполнения операции сдвига осуществляется на счетноиндикаторых ячейках, подключенных к прямым выходам триггеров регистра.

4. Содержание отчета

В результате выполнения работы заполняется бланк отчета, в котором должны быть:

1. Схема испытаний регистра параллельного действия с предварительной установкой в «0»;
2. Схема испытаний регистра параллельного действия с приемом информации парафазным кодом;

3. Схема испытаний регистра параллельного действия, выполняющего операцию поразрядного операцию поразрядного логического сложения;
4. Схема испытаний регистра параллельного действия, выполняющего операцию поразрядного операцию поразрядного логического умножения;
5. Схема испытаний кольцевого регистра сдвига;
6. Модель синтезируемого многофункционального регистра сдвига (подобно модели, изображенной на рисунке 4.8);
7. Таблица и матрица переходов JK-триггера;
8. Кодированная таблица переходов и функций возбуждения i-го триггера регистра;
9. Карты Карно для функций возбуждения q_{yi} и q_{ki};
10. МДНФ функций возбуждения, представленные в базисе И-НЕ;
11. Схема испытаний синтезированного многофункционального регистра сдвига;
12. Выводы по каждому пункту программы работы.

Контрольные вопросы

1. Назначение и классификация регистров.
2. Какие микрооперации можно реализовать с помощью регистра.
3. Способы приема информации в регистр.
4. Изобразить и пояснить схему, осуществляющую преобразование параллельного кода в последовательный базис и наоборот.
5. В чем особенность схемы замкнутого в кольцо регистра сдвига.
6. Порядок синтеза многофункционального регистра сдвига.

Варианты заданий

Вариант	**Микрооперации сдвига, выполняемые регистром**	**Исходный код**
1	Влево на 1 разряд, вправо на 2 разряда	101
2	Вправо на 1 разряд и на 3 разряда	100
3	Вправо на 2 разряда, влево на 3 разряда	011
4	Влево на 2 разряда и на 3 разряда	010
5	Влево на 2 разряда и вправо на 1 разряд	001
6	Влево на 1 разряд и на 2 разряда	110
7	Вправо на 3 разряда, влево на 3 разряда	001
8	Вправо на 2 разряда, влево на 4 разряда	100
9	Вправо на 1 разряд и на 2 разряда	101
10	Вправо на 1 разряд, влево на 2 разряда	011
11	Влево на 1 разряд, вправо на 3 разряда	110
12	Влево на 2 разряд, вправо на 3 разряда	010
13	Вправо на 3 разряда, влево на 2 разряда	001
14	Вправо на 1 разряд, влево на 4 разряда	011
15	Влево на 2 разряд, вправо на 4 разряда	100
16	Вправо на 4 разряд и на 1 разряд	010
17	Вправо на 4 разряда, влево на 3 разряда	110
18	Вправо на 2 разряд и на 3 разряда	101

Вариант	Микрооперации сдвига, выполняемые регистром	Исходный код
19	Вправо на 4 разряда, влево на 2 разряда	001
20	Вправо на 3 разряд, влево на 4 разряда	011
21	Влево на 3 разряд, вправо на 4 разряда	110
22	Вправо на 4 разряд и на 2 разряд	101
23	Вправо на 4 разряда, влево на 4 разряда	010
24	Влево на 1 разряда и на 4 разряда	100
25	Влево на 2 разряда и на 4 разряда	110
26	Влево на 3 разряда и на 4 разряда	001
27	Вправо на 4 разряд и на 3 разряд	101
28	Влево на 1 разряда и на 3 разряда	011
29	Влево на 3 разряд, вправо на 2 разряда	010
30	Вправо на 1 разряда, влево на 2 разряда	100

5 СИНТЕЗ СЧЕТЧИКОВ

Цель работы:

Закрепить знания, полученные на лекциях по принципам построения различных схем счетчиков импульсов, применяемых в дискретных устройствах; приобрести навыки в сборке, наладке и экспериментальном исследовании счетчиков на интегральных микросхемах, получить умения в анализе и синтезе счетчиков импульсов.

Программа работы

1. Провести исследование счетчика с последовательным, сквозным и параллельным переносом.
2. Провести исследование пересчетной схемы.
3. Провести исследование реверсивного счетчика.
4. Провести синтез и исследование двоично-десятичного счетчика.

Теоретическое обоснование

Цифровым счетчиком импульсов называется последовательностный автомат (дискретный автомат с памятью), который осуществляет счет числа поступивших на его выход импульсов и хранение этого числа до сброса счетчика в исходное состояние или до продолжения счета. Счетчики строятся на триггерах. Количество триггеров n должно быть таким, чтобы множество внутренних состояний счетчика 2^n было не меньше максимального числа импульсов, которое должно быть зафиксировано. Если количество счетных импульсов не ограничивать, то счетчик Ксч = 2^n. Через каждые 2^n импульса он будет возвращаться в начальное состояние и снова начинать счет импульсов.

Счетчики можно классифицировать по ряду признаков:

- по основанию системы счисления (двоичные счетчики, у которых $К_{Сч}$ кратно целой степени двойки, десятичные с $К_{Сч} = 10$) и др;
- по направлению счета (суммирующие, вычитающие и реверсивные);
- по способу приема входных сигналов (синхронные и асинхронные);
- по способу построения цепей межразрядных переносов (с последовательным, сквозным, параллельным и групповым переносами) и др.

Конструктивно цифровой счетчик может быть выполнен как в виде совокупности интегральных микросхем – триггеров, определенным образом соединенных на печатной плате, так и в виде одной микросхемы повышенной степени интеграции (Рисунок5.1.), содержащей сформированную на единой подложке схему многоразрядного счетчика.

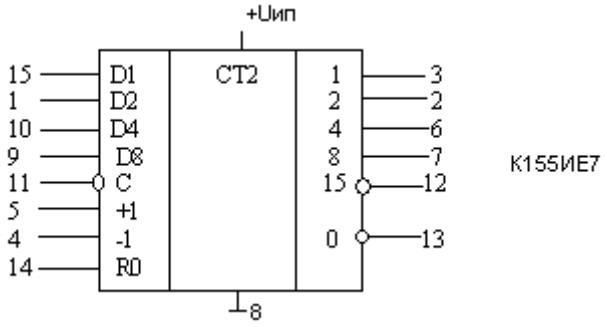


Рисунок 5.1 – Условное графическое обозначение счетчика импульсов

Для определения принципа построения счетчиков с последовательным переносом рассмотрим пример реализации трехразрядного суммирующего счетчика в коде 8421.

Порядок смены состоянбий счетчика задан таблицей 5.1.

Таблица 5.1.

Число импульсов	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	1	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

В качестве исходного принято состояние, которое определяется нулевым уровнем на прямых выходах всех триггеров, т.е. Q1 = Q2 = Q3 = Q4 = 0. Как следует из таблицы, с приходом очередного счетного импульса к содержимому счетчика прибавляется единица. При этом увеличивается на единицу номер состояния, являющийся десятичным эквивалентом соответствующего данному состоянию двоичного числа.

На (Рисунок 5.2, а) приведена схема четырехразрядного асинхронного двоичного счетчика на JK- триггерах с последовательным переносом единицы в старший разряд на (Рисунок5.2,б) – УГО на функциональных схем.

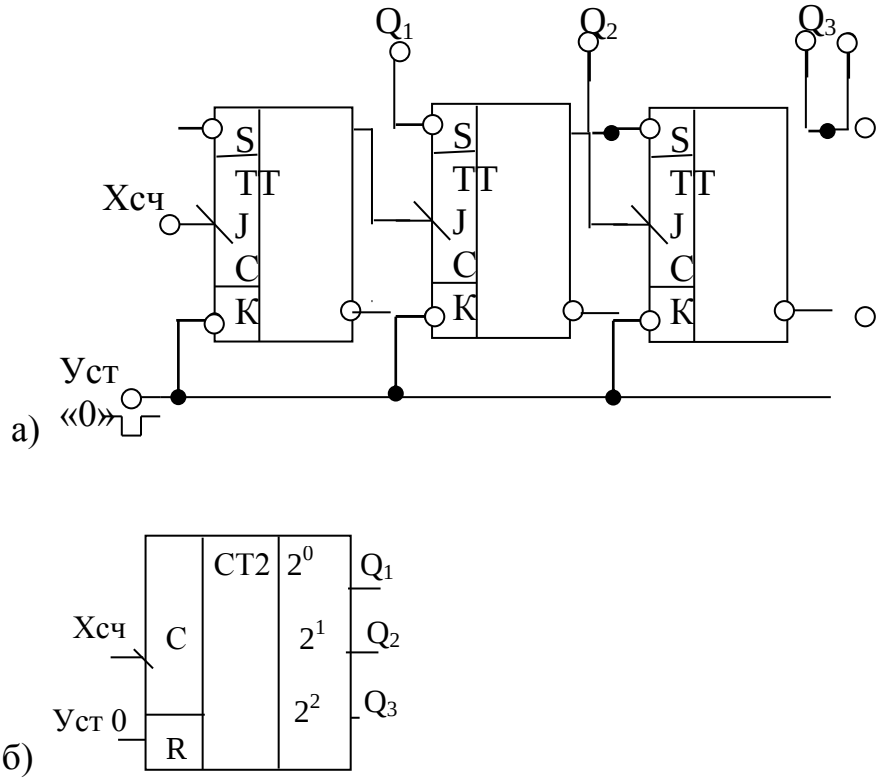


Рисунок 5.2 - а) Схема 4-х разрядного двоичного счетчика с последовательным переносом; б) Условное графическое обозначение

Установка счетчика в исходное состояние (нулевое) осуществляется путем подачи низкого уровня U^0 на шину «Уст.0», подключенную к входам R всех триггеров.

При поступлении на счетчик первого импульса счета триггер Q1перейдет в состояние 1, когда на его входе С потенциал изменится с высокого на низкий (1-0). Изменение состояния каждого последующего разряда происходит при изменении состояния предыдущего разряда с 1 на 0. Это означает, что всякий раз, когда данный триггер в счетчике переходит из состояния в 1 в состояние 0, на его выходе должен формироваться сигнал переноса, опрокидывающий следующий триггер. Если же данный триггер переходит из 0 в 1 , то сигнал переноса на его выходе не должно быть.

Из таблицы 5.1. так же следует, что триггер первого, самого младшего разряда, должен менять свое состояние каждый раз с приходом очередного импульса счета, а триггер каждого последующего разряда – вдвое реже триггера предыдущего разряда.

Работа счетчика поясняется временными диаграммами, изображенными на рис 5.3.

В счетчике с последовательным переносом время переходного процесса (формирования кода на выходных шинах) зависит от его разрядности и числа поступивших импульсов счета

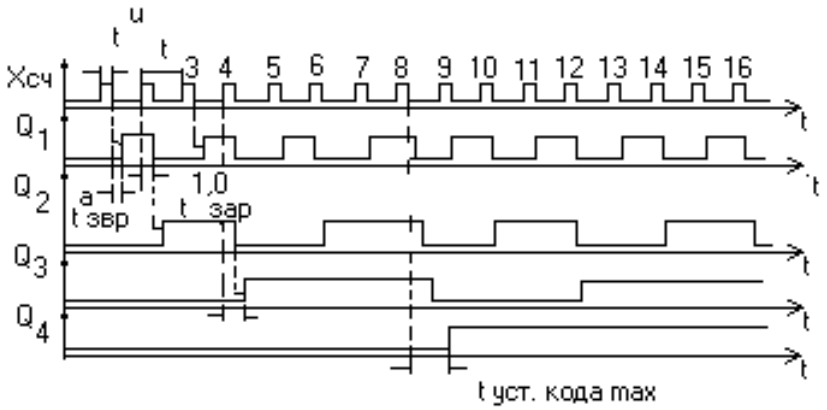


Рисунок 5.3 – Временная диаграмма работы счетчика

Максимальное время установления кода определяется по формуле

$$\text{tуст.кода} = n \cdot t_{ЗД,Р,СР} \qquad (5.1)$$

где n – разрядность счетчика

$$\text{tзд, р, ср} = \frac{t^{0,1}зд,p + t^{1,0}зд,p}{2} \qquad (5.2)$$

Основным недостатком счетчика с последовательным переносом является малое быстродействие.

Для повышения быстродействия счетчиков применяют различные способы ускорения переноса. Один из широко применяемых способов ускорения переноса в счетчиках основан на введении в их схемы логических элементов, с помощью которых обеспечивается возможность одновременного формирования сигналов переноса для последующих разрядов.

На Рисунок 5.4. приведена схема синхронного четырехразрядного счетчика со сквозным переносом. Здесь сигнал Хсч подается одновременно на входы всех триггеров. Переключение любого триггера данного счетчика возможно только, если к объединенному JK входу приложен единичный потенциал, что будет лишь тогда, когда все предшествующие триггеры младших разрядов находятся в состоянии «единица».

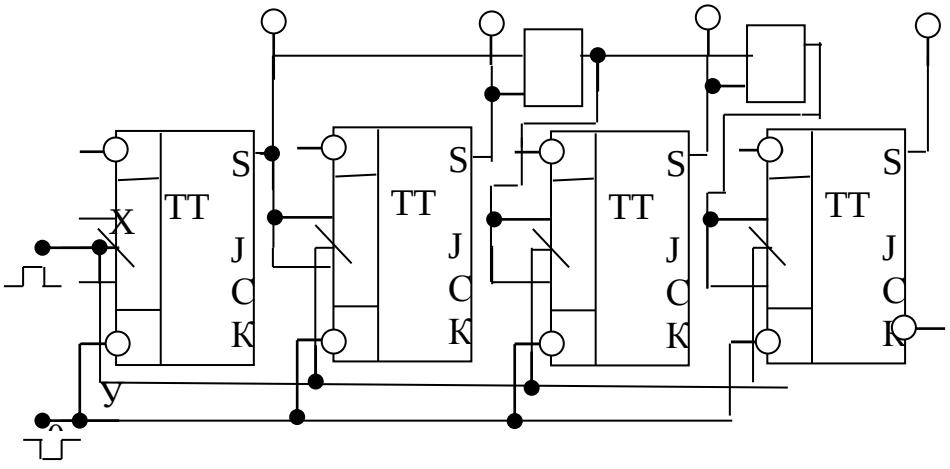


Рисунок 5.4 –Схема синхронного 4-х разрядного счетчика импульсов со

сквозным переносом

Время установления кодов в таком счетчике определяется временем переключения одного триггера и задержки распространения сигнала в последовательной цепи элементов «И».

t уст.кода = $t_{ЗД,Р,СР,ТР} + (n-2)t_{ЗД,Р,СР\ "И"}$ (5.3.)

Где n – разрядность счетчика.

tад, р, ср для элементов «И» и триггера определяется по формуле 5.2.

Еще более высоким быстродействием обладают счетчики с параллельным переносом. Схема счетчика с параллельным переносом приведена на Рисунок5.5. Как видно из схемы, последний триггер должен опрокинуться под воздействием счетного импульса при наличии «И» на выходах всех последующих триггеров. Следовательно, для формирования сигнала переноса в каждый разряд счетчика необходимо включить элемент «И» и соединить его входы с прямыми выходами всех триггеров предыдущих разрядов.

Время установления кода в счетчиках с параллельным переносом постоянно и определяется временем переключения одного триггера и задержкой в одной схеме «И».

t уст.кода = $t_{ЗД,Р,СР,ТР} + t_{ЗД,Р,СР\ "И"}$ (5.4.)

где tзад, р, ср триггера и элемента «И» определяется по формуле 5.2.

С возрастанием номера триггера прогрессивно увеличивается числа входов элемента «И», поэтому разрядность счетчика с параллельным переносом невелика и редко превышает четыре.

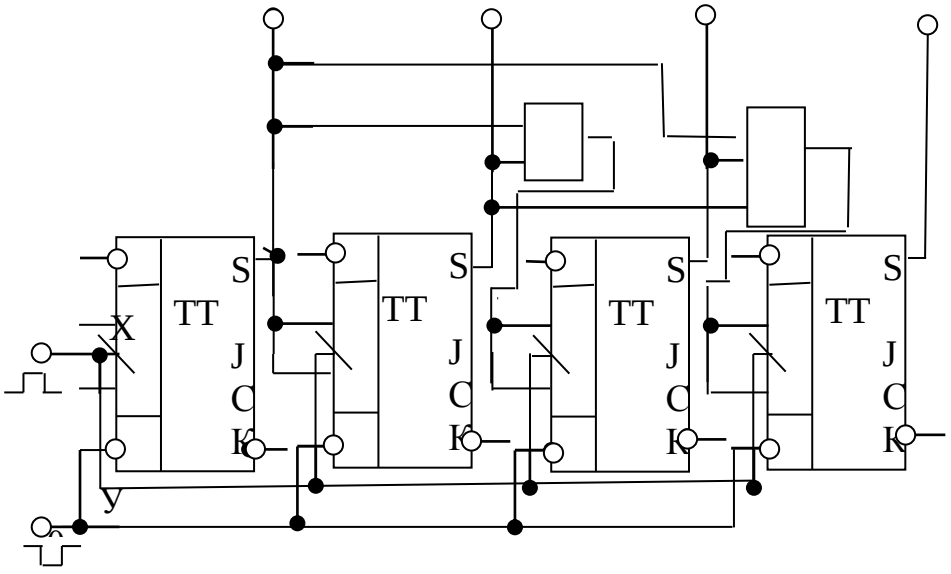


Рисунок 5.5 – Схема 4-х разрядного синхронного счетчика импульсов с параллельным переносом

Рассмотренные счетчики имеют коэффициент пересчета Ксч, кратный степени числа два. Для построения счетчики с Ксч, отличным от 2^n, в его схему вводят обратные связи, позволяющие уменьшить число возможных состояний.

Рассмотрим методику синтеза пересчетной схемы с Ксч = 5.

определяем необходимое число элементов в памяти из выражения

$n = \lceil \log_2 Ксч \rceil$, (5.5.)

$n = \lceil \log_2 5 \rceil = 3$

Определяем число избыточных состояний

$m = 2^n - Ксч$ (5.6.)

$m = 2^3 - 5 = 3$

Представляем число избыточных состояний m разрядным двоичным кодом

$[m]_{10} = 2^{n-1} \ldots 2^2, 2^1, 2^0\ 1_2,$ (5.7.)

$[3]_{10} = [011]_2$

В n- разрядном двоичном коде числа избыточных состояний номер триггера, помеченный «0», указывает, с какого триггера снимается сигнал обратной связи, а номер триггера, помеченный «I», указывает, на S – вход какого триггера подается сигнал обратной связи.

0	1	1
T_0^3	T_2^4	T_4^4

Функциональная электрическая схема синтезированной пересчетной схемы с Ксч = 5 представлена на Рисунок 5.6. Триггер Т4 не входит в пересчетную схему, а лишь регистрирует импульс пересчета Р5 на ее выходе.

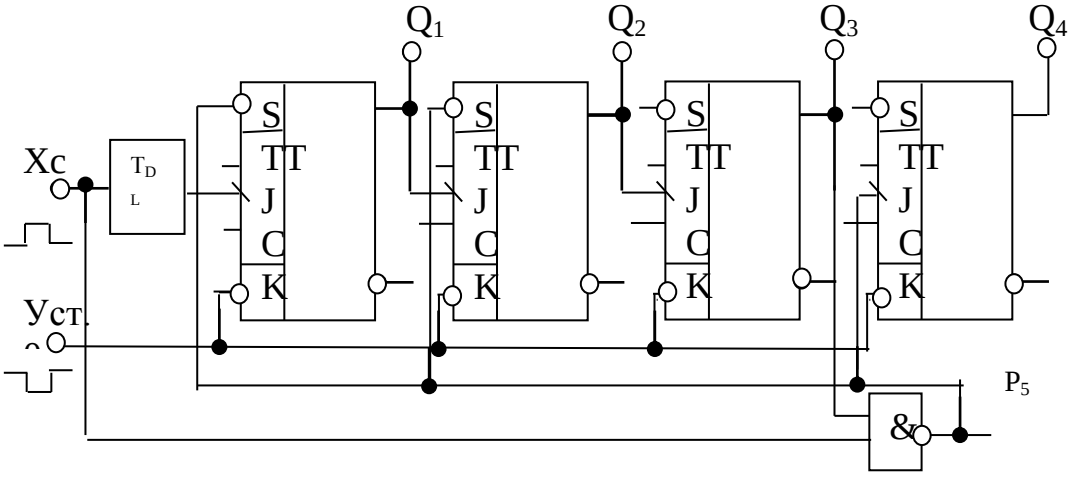


Рисунок 5.6 – Пересчетная схема с Ксч=5

Временные диаграммы, поясняющие работу пересчетной схемы, приведены на Рисунок 5.7.

$\tau_{зд} > t_{зд,\ р,\ «S»}$

Часто требуется, чтобы счетчик осуществлял не суммирование, а вычитание количества поступивших импульсов из некоторого заранее занесенного кода.

Такие счетчики называются вычитающими, или счетчиками обратного счета. Особенность вычитающего счетчика состоит в том, что опрокидывание триггера каждого последующего разряда происходит при изменении уровня на выходе триггера предыдущего разряда от 0 к 1, т.е. при сигнале за-

ема , обратному сигнале переноса в суммирующем счетчике. Строится вычитающий счетчик так же, как суммирующий, но с тем отличием, что входом каждого последующего триггера соединяется инверсный выход предыдущего триггера.

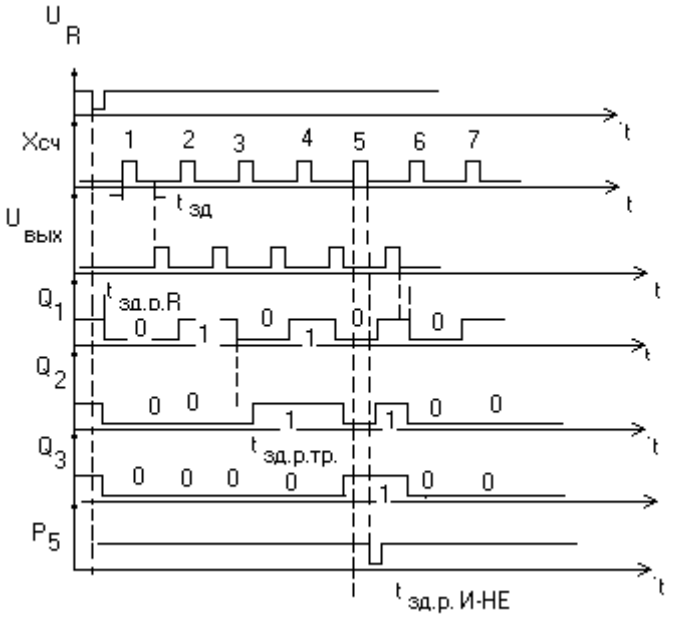


Рисунок 5.7 – Временные диаграммы поясняющие принцип работы пересчетной схемы с Ксч=5

Отсюда вытекает очень простой принцип получения реверсивного счетчика - коммутацией счетного входа последующего триггера на прямой или инверсный выход предыдущего триггера в зависимости от сигнала на шине управления.

На Рисунок5.8. приведена функциональная электрическая схема трехразрядного реверсивного асинхронного счетчика.

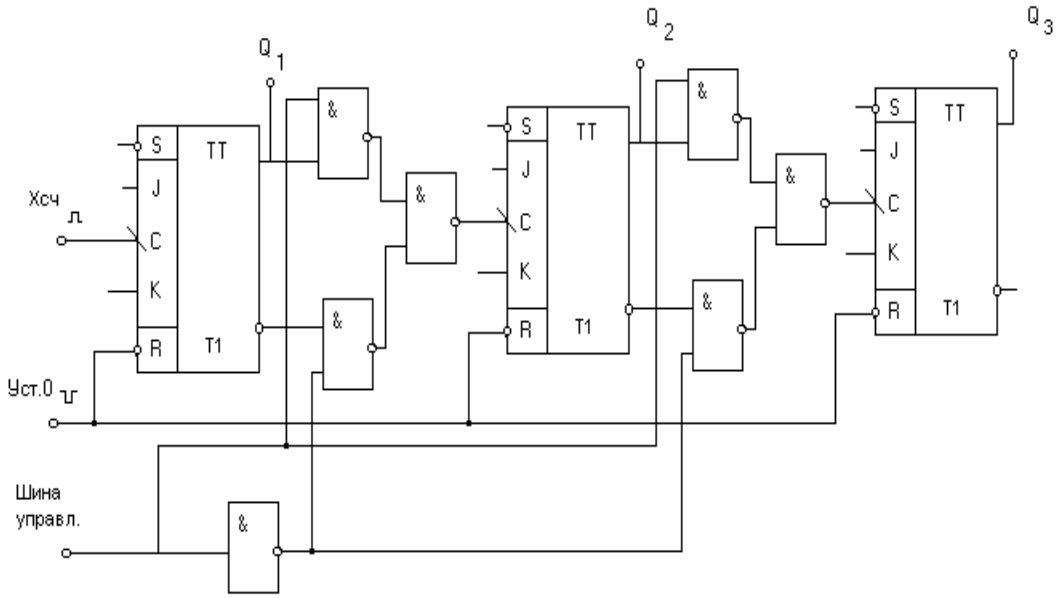


Рисунок 5.8 – Функциональная электрическая схема 3-х разрядного реверсивного счетчика импульсов

На Рисунок5.9. приведена обобщенная схема логической структуры синхронного счетчика. По этой схеме можно уяснить принцип работы любого синхронного счетчика. Сигналы с выходов триггеров поступают на входы комбинационной схемы, которая преобразует поступившую информацию. Сигналы с входов комбинационной схемы подаются на входы триггеров. Преобразованная информация не воспринимается триггером до тех пор, пока на синхронные входы не поступит счетный сигнал Хсч. Информация, находящаяся на выходах каждого триггера, так преобразуется комбинационной схемой, чтобы с приходом очередного Хсч осуществить требуемый переход счетчика из предыдущего состояния в следующее. Таким образом, функции возбуждения входов i-го триггера можно записать в виде:

$$q_{ji} = f_{ji}\,[Q1(t), Q2(t), \ldots\ldots Qn(t)]; \qquad (5.8.)$$

$$q_{ki} = f_{ki}\,[Q1(t), Q2(t), \ldots\ldots Qn(t)]; \qquad (5.9.)$$

Значения всех переменных в этих выражениях определены для одного и того же момента t. Поэтому функции возбуждения триггеров являются переключательными функциями, которым соответствуют комбинированные схемы, формирующие входные сигналы для триггеров. Следовательно, если задан или выбран тип триггера, то задача логического проектирования схемы счетчика заключается в составлении функций возбуждения каждого триггера и минимизации найденных функций в заданном базисе.

Рассмотрим метод синтеза синхронных счетчиков на примере проектирования двоично-десятичного счетчика.

ПРИМЕР. Синтезировать синхронных суммирующий двоично-десятичный счетчик на JK-триггерах с системой кодирования 2421 (2421 – веса двоичных разрядов).

РЕШЕНИЕ 1. определяется число элементарных автоматов (триггеров), обеспечивающих необходимый коэффициент счета

$$Ксч = \lceil \log_2 n \rceil, \qquad (5.10.)$$

где n – число элементарных автоматов.

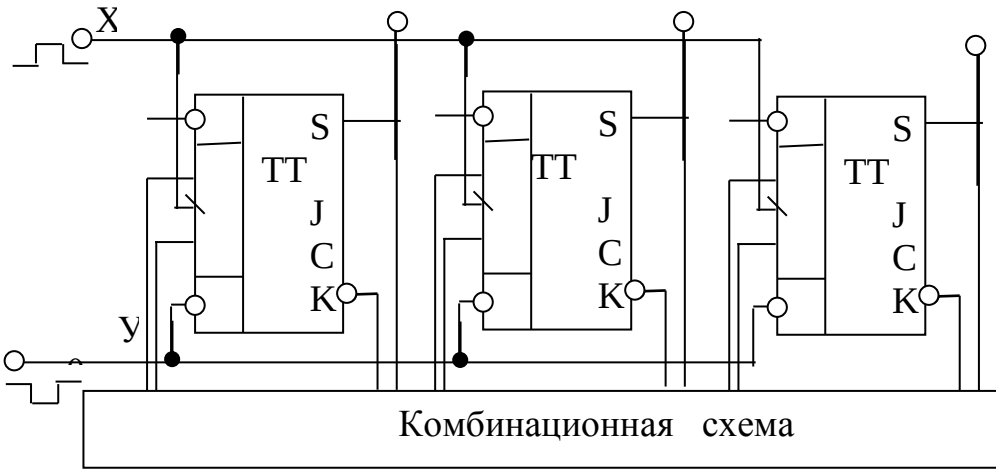


Рисунок 5.9 – Функциональная электрическая схема синтезируемого

двоично-десятичного счетчика

Составляется кодированная таблица состояний элементов памяти (триггеров) в соответствии с заданной системой кодирования 2421 (таблица 5.2.) и матрица переходов заданного JK – триггера (таблица 5.3.)

Таблица 5.2.

Число импульсов	Состояния триггеров и веса разрядов			
	1	2	3	4
	Q4	Q3	Q2	Q1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Таблица 5.3.

Q(t) – Q(t+1)	q_j (t) q_k (t
0 - 0	0 - α_1
0 - 1	1 - α_2
1 - 0	α_3 - 1
1 - 1	α_4 - 0

На основании кодированной таблицы состояний элементов памяти (таблица 5.2.) и матрицы переходов JK – триггера (таблица 5.3) составляется таблица переходов и функций возбуждения синтезируемого счетчика (таблица 5.4.)

Порядок выполнения работы

1. Преподаватель проверяет подготовку к занятию. Дает допуск на выполнение практического занятия.

2. Исследовать работу цифровых счетчиков импульсов. Контроль записанной информации производится подключением прямых выходов триггеров к счетноиндикаторным ячейкам.

Содержание отчета

В результате выполнения практического занятия заполняется бланк отчета, в котором должны быть:

Исследование цифровых счетчиков

а) условное графическое обозначение цифровых счетчиков;

б) таблица истинности цифровых счетчиков;

в) схемы испытаний цифровых счетчиков;

г) временные диаграммы работы цифровых счетчиков.

Синтез и исследование цифровых счетчиков.

а) таблица перевода числа для синтезируемого цифрового счетчика;

в) схема испытаний синтезированного цифрового счетчика;

г) временные диаграммы работы синтезированного цифрового счетчи-

ка.

Выводы по каждому пункту программы работы.

Контрольные вопросы

1. Назначение и классификация регистров.
2. Какие микрооперации можно реализовать с помощью регистра.
3. Способы приема информации в регистр.
4. Изобразить и пояснить схему, осуществляющую преобразование параллельного кода в последовательный базис и наоборот.
5. В чем особенность схемы замкнутого в кольцо регистра сдвига.
6. Порядок синтеза многофункционального регистра сдвига.

6 СИНТЕЗ СУММАТОРОВ И СХЕМ СРАВНЕНИЯ

Цель работы:

Закрепить знания, полученные на лекциях по принципам построения различных схем счетчиков импульсов, применяемых в дискретных устройствах; приобрести навыки в сборке, наладке и экспериментальном исследовании счетчиков на интегральных микросхемах, получить умения в анализе и синтезе счетчиков импульсов.

Методические рекомендаии по выполнению практичесого занятия

Сумматором называется функциональный узел, в котором выполняется операция суммирования цифровых кодов двух чисел, представленных сигналом на его входах.

Сумматоры широко применяются в различных устройствах дискретной обработки данных АСУ и связи и прежде всего в арифметических и управляющих устройствах.

Классификацию сумматоров можно провести по основанию системы счисления чисел, которыми оперирует сумматор. Имеются следующие сумматоры: двоичные, двоично-десятичные и другие. В различных устройствах находят применение как одноразрядные, так и многоразрядные сумматоры.

Одноразрядные сумматоры применяются для обработки последовательных кодов, а их соответствующим образом соединенная совокупность образует многоразрядный сумматор, предназначенный для обработки параллельных кодов.

Важным классификационным признаком многоразрядных сумматоров является способ организации цепей переносов, который может быть последовательным, сквозным и групповым.

Сумматоры бывают комбинационные и накапливающие. Первые выполняются на комбинационных логических элементах и работают при параллельном поступлении входных переменных. Вторые осуществляют сложение переменных, поступающих последовательно.

Накапливающие сумматоры строятся на основе триггеров, которые являются элементами памяти. Поэтому накапливающие сумматоры реализуют не только сложение операндов, но и запоминание (хранение) результата, т.е., помимо прочего, они выполняют функцию регистра. В качестве элементов памяти в сумматорах наиболее часто применяются Т или JK–триггеры.

В отличие от комбинационных сумматоров, в которых одинаковые разряды всех операндов подаются на разные входы одновременно, накапливающие сумматоры для каждого разряда всех операндов имеют один общий вход. На этот вход сначала подается разрядный сигнал одного операнда, а затем второго.

Одноразрядным накапливающим сумматором называется схема, производящая алгебраическое суммирование поочередно поступающих на вход цифр одного разряда слагаемых и переноса из соседнего младшего разряда с

запоминанием результатов суммирования и формированием сигнала переноса в соседний старший разряд.

Основой такого сумматора является триггер со счетным входом (Т-триггер), линия задержки и логический элемент «И». Линия задержки и логический элемент «И» обеспечивают формирование единицы переноса (ФЕП).

Схема сумматора изображена на рисунке 6.1.

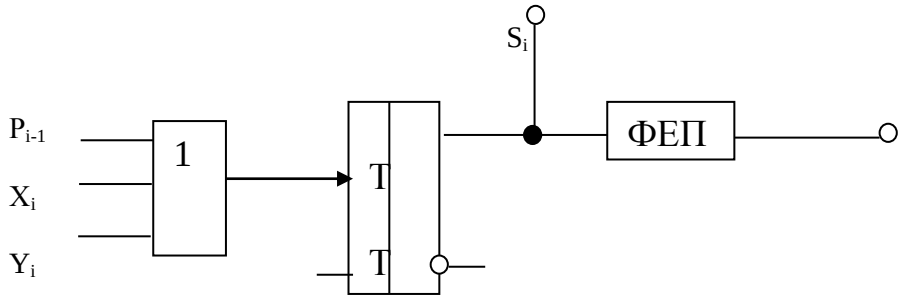


Рисунок 6.1 - Схема сумматора

P_{i-1}-вход для единицы переноса из предыдущего разряда;

P_i -выход единицы переноса i – разряда;

$S_i = x_i + y_i$ – сумма при сложении i – разрядов.

Характер работы сумматора таков, что единица переноса формируется при переходе триггера из единичного в нулевое состояние. Схема формирования единицы переноса должна формировать задержанную относительно момента подачи разряда второго операнда единицу переноса при переходе триггера из единичного состояния. Операция суммирования осуществляется последовательно во времени.

Задача 1

Провести синтез двоичного сумматора последовательного действия, закон функционирования которого задан таблицей переходов и выходов (табл. 1 и 2), построить функциональную схему на JK и T–триггерах.

Таблица 1

Состояния Входы	a_0	a_1
x_0	a_0	a_0
x_1	a_0	a_1
x_2	a_0	a_1
x_3	a_1	a_1

Таблица 2

Состояния Входы	a_0	a_1
x_0	y_0	y_1
x_1	y_1	y_0
x_2	y_1	y_0
x_3	y_0	y_1

Задача 2

Провести синтез сумматора, производящего поразрядное суммирование букв n– разрядного слова:

а) по модулю 3;

б) по модулю 4;
в) по модулю 7.

7 ЭЛЕМЕНТНАЯ БАЗА ЦИФРОВЫХ УСТРОЙСТВ

Цель работы: закрепить знания, полученные на лекции, посвященной т элементной базе цифровых устройств.

Программа работы:

1. Основные обозначения на схемах.
2. Серии цифровых микросхем.
3. Корпуса цифровых микросхем.
4. Функции цифровых устройств.

Методические указания по выполения практического занятия

Для изображения электронных устройств и их узлов применяется три основных типа схем:

- принципиальная схема ;
- структурная схема ;
- функциональная схема.

Различаются они своим назначением и, самое главное, степенью детализации изображения устройств.

Принципиальная схема — наиболее подробная. Она обязательно показывает все использованные в устройстве элементы и все связи между ними. Если схема строится на основе микросхем, то должны быть показаны номера выводов всех входов и выходов этих микросхем. Принципиальная схема должна позволять полностью воспроизвести устройство. Обозначения принципиальной схемы наиболее жестко стандартизованы, отклонения от стандартов не рекомендуются.

Структурная схема — наименее подробная. Она предназначена для отображения общей структуры устройства, то есть его основных блоков, узлов, частей и главных связей между ними. Из структурной схемы должно быть понятно, зачем нужно данное устройство и что оно делает в основных режимах работы, как взаимодействуют его части. Обозначения структурной схемы могут быть довольно произвольными, хотя некоторые общепринятые правила все-таки лучше выполнять.

Функциональная схема представляет собой гибрид структурной и принципиальной. Некоторые наиболее простые блоки, узлы, части устройства отображаются на ней, как на структурной схеме, а остальные — как на принципиальной схеме. Функциональная схема дает возможность понять всю логику работы устройства, все его отличия от других подобных устройств, но не позволяет без дополнительной самостоятельной работы воспроизвести это устройство. Что касается обозначений, используемых на функциональных схемах, то в части, показанной как структура, они не стандартизованы, а в части, показанной как принципиальная схема, — стандартизованы.

В технической документации обязательно приводятся структурная или функциональная схема, а также обязательно принципиальная схема. В

научных статьях и книгах чаще всего ограничиваются структурной или функциональной схемой, приводя принципиальные схемы только некоторых узлов.

А теперь рассмотрим основные обозначения, используемые на схемах.

Все узлы, блоки, части, элементы, микросхемы показываются в виде прямоугольников с соответствующими надписями. Все связи между ними, все передаваемые сигналы изображаются в виде линий, соединяющих эти прямоугольники. Входы и входы/выходы должны быть расположены на левой стороне прямоугольника, выходы — на правой стороне, хотя это правило часто нарушают, когда необходимо упростить рисунок схемы. Выводы и связи питания, как правило, не прорисовывают, если, конечно, не используются нестандартные включения элементов схемы. Это самые общие правила, касающиеся любых схем.

Прежде чем перейти к более частным правилам, дадим несколько определений.

Положительный сигнал (сигнал положительной полярности) — это сигнал, активный уровень которого — логическая единица. То есть нуль — это отсутствие сигнала, единица — сигнал пришел (рисунок 7.1).

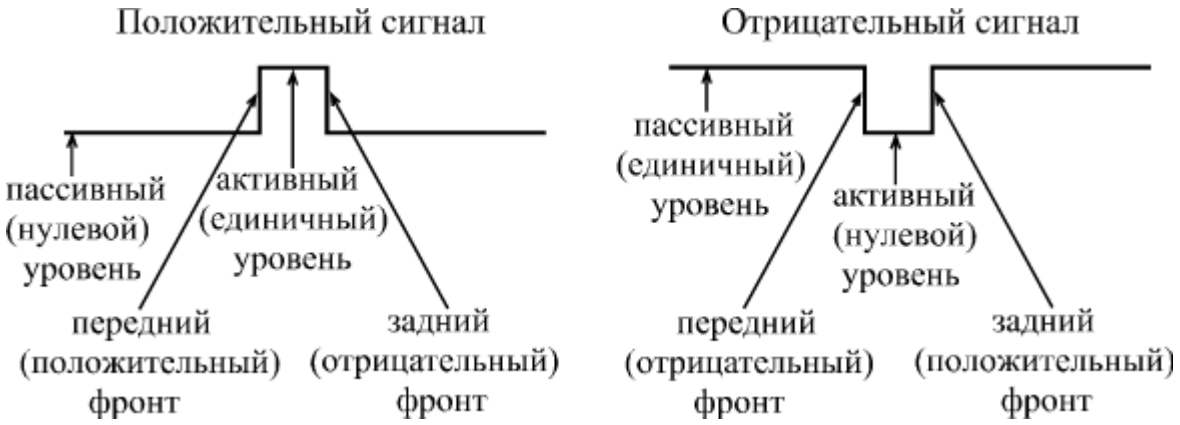


Рисунок 7.1. Элементы цифрового сигнала

Отрицательный сигнал (сигнал отрицательной полярности) — это сигнал, активный уровень которого — логический нуль. То есть единица — это отсутствие сигнала, нуль — сигнал пришел (Рисунок 7.1).

Активный уровень сигнала — это уровень, соответствующий приходу сигнала, то есть выполнению этим сигналом соответствующей ему функции.

Пассивный уровень сигнала — это уровень, в котором сигнал не выполняет никакой функции.

Инвертирование или инверсия сигнала — это изменение его полярности.

Инверсный выход — это выход, выдающий сигнал инверсной полярности по сравнению с входным сигналом.

Прямой выход — это выход, выдающий сигнал такой же полярности, какую имеет входной сигнал.

Положительный фронт сигнала — это переход сигнала из нуля в единицу.

Отрицательный фронт сигнала (спад) — это переход сигнала из единицы в нуль.

Передний фронт сигнала — это переход сигнала из пассивного уровня в активный.

Задний фронт сигнала — это переход сигнала из активного уровня в пассивный.

Тактовый сигнал (или строб) — управляющий сигнал, который определяет момент выполнения элементом или узлом его функции.

Шина — группа сигналов, объединенных по какому-то принципу, например, шиной называют сигналы, соответствующие всем разрядам какого-то двоичного кода.

Рисунок 7.2 - Обозначение входов и выходов

Для обозначения полярности сигнала на схемах используется простое правило: если сигнал отрицательный, то перед его названием ставится знак минус, например, -WR или -OE, или же над названием сигнала ставится черта. Если таких знаков нет, то сигнал считается положительным. Для названий сигналов обычно используются латинские буквы, представляющие собой сокращения английских слов, например, WR — сигнал записи (от "*write*" — "писать").

Инверсия сигнала обозначается кружочком на месте входа или выхода. Существуют инверсные входы и инверсные выходы (Рисунок 7.2).

Если какая-то микросхема выполняет функцию *по* фронту входного сигнала, то на месте входа ставится косая черта (под углом 45°), причем наклон вправо или влево определяется тем, положительный или отрицательный фронт используется в данном случае (Рисунок 7.2).

Тип выхода микросхемы помечается специальным значком: выход 3С — перечеркнутым ромбом, а выход ОК — подчеркнутым ромбом (Рисунок 7.2). Стандартный выход (2С) никак не помечается.

Наконец, если у микросхемы необходимо показать неинформационные

выводы, то есть выводы, не являющиеся ни логическими входами, ни логическими выходами, то такой *вывод* помечается косым крестом (две перпендикулярные линии под углом 45°). Это могут быть, например, выводы для подключения внешних элементов (резисторов, конденсаторов) или выводы питания (Рисунок 7.3).

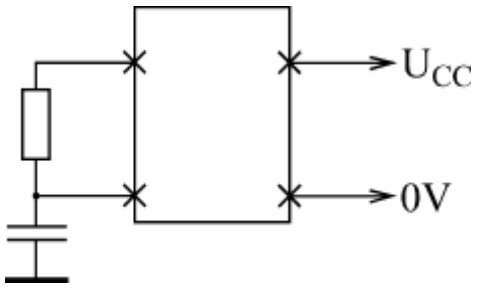


Рисунок 7.3. Обозначение неинформационных выводов

В схемах также предусматриваются специальные обозначения для шин (Рисунок 7.4). На структурных и функциональных схемах шины обозначаются толстыми линиями или двойными стрелками, причем количество сигналов, входящих в шину, указывается рядом с косой чертой, пересекающей шину. На принципиальных схемах шина тоже обозначается толстой линией, а входящие в шину и выходящие из шины сигналы изображаются в виде перпендикулярных к шине тонких линий с указанием их номера или названия (Рисунок 7.4). При передаче по шине двоичного кода нумерация начинается с младшего разряда кода.

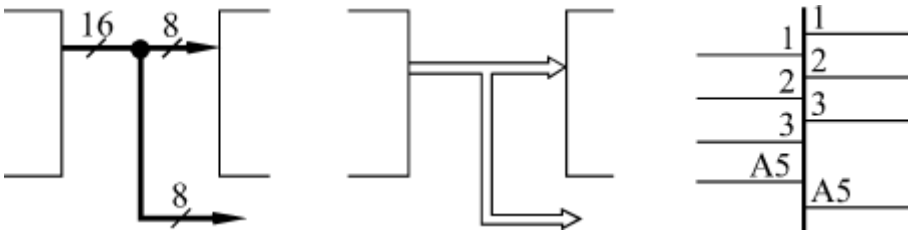


Рисунок 7.4. Обозначение шин

При изображении микросхем используются сокращенные названия входных и выходных сигналов, отражающие их функцию. Эти названия располагаются на рисунке рядом с соответствующим выводом. Также на изображении микросхем указывается выполняемая ими *функция* (обычно в центре вверху). Изображение микросхемы иногда делят на три *вертикальные поля*. Левое *поле* относится к входным сигналам, правое — к выходным сигналам. В центральном *поле* помещается название микросхемы и символы ее особенностей. Неинформационные выводы могут указываться как на левом, так и на правом *поле*; иногда их показывают на верхней или нижней стороне прямоугольника, изображающего микросхему.

В табл. 7.1 приведены некоторые наиболее часто встречающиеся обо-

значения сигналов и функций микросхем. Микросхема в целом обозначается на схемах буквами DD (от английского "digital" — "цифровой") с соответствующим номером, например, DD1, DD20.1, DD38.2 (после точки указывается номер элемента или узла внутри микросхемы).

Таблица 7.1. Некоторые обозначения сигналов и микросхем

Обозначение	**Название**	**Назначение**
&	And	Элемент И
=1	Exclusive Or	Элемент Исключающее ИЛИ
1	Or	Элемент ИЛИ
A	Address	Адресные разряды
BF	Buffer	Буфер
C	*Clock*	*Тактовый сигнал* (строб)
CE	*Clock* Enable	Разрешение *тактового сигнала*
CT	Counter	Счетчик
CS	*Chip* Select	Выбор микросхемы
D	Data	Разряды данных, данные
DC	*Decoder*	*Дешифратор*
EZ	Enable Z-state	Разрешение третьего состояния
G	Generator	Генератор
I	Input	Вход
I/O	Input/Output	Вход/Выход
OE	*Output Enable*	Разрешение выхода
MS	*Multiplexer*	*Мультиплексор*
Q	Quit	Выход
R	Reset	Сброс (установка в нуль)
RG	Register	Регистр
S	Set	Установка в единицу
SUM	*Summator*	*Сумматор*
T	*Trigger*	Тригер
TC	Terminal Count	Окончание счета
Z	Z-state	Третье состояние выхода

Серии цифровых микросхем

В настоящее время выпускается огромное количество разнообразных цифровых микросхем: от простейших логических элементов до сложнейших

процессоров, микроконтроллеров и специализированных БИС (Больших Интегральных Микросхем). Производством цифровых микросхем занимается множество фирм — как у нас в стране, так и за рубежом. Поэтому даже классификация этих микросхем представляет собой довольно трудную задачу.

Однако в качестве базиса в цифровой схемотехнике принято рассматривать классический набор микросхем малой и средней степени интеграции, в основе которого лежат ТТЛ серии семейства 74, выпускаемые уже несколько десятилетий рядом фирм, например, американской фирмой *Texas Instruments* (TI). Эти серии включают в себя функционально полный комплект микросхем, используя который, можно создавать самые разные цифровые устройства. Даже при компьютерном проектировании современных сложных микросхем с программируемой логикой (ПЛИС) применяются модели простейших микросхем этих серий семейства 74. При этом разработчик рисует на экране компьютера схему в привычном для него элементном базисе, а затем *программа* создает прошивку ПЛИС, выполняющую требуемую функцию.

Рисунок 7.5. Система обозначений фирмы Texas Instruments

Каждая микросхема серий семейства 74 имеет свое обозначение, и система обозначений отечественных серий существенно отличается от принятой за рубежом.

В качестве примера рассмотрим систему обозначений фирмы Texas Instruments (Рисунок 7.5). Полное обозначение состоит из шести элементов:

1. Идентификатор фирмы SN (для серий AC и ACT отсутствует).
2. Температурный диапазон (тип семейства):
   - 74 — коммерческие микросхемы (температура окружающей среды для биполярных микросхем — 0...70°C, для КМОП микросхем — –40...+85°C),
   - 54 — микросхемы военного назначения (температура — –55...+125°C).
3. Код серии (до трех символов):
   - Отсутствует — стандартная ТТЛ–серия.
   - LS (*Low Power* Schottky) — маломощная серия ТТЛШ.
   - S (Schottky) — серия ТТЛШ.
   - *ALS* (Advanced Schottky) — улучшенная серия ТТЛШ.
   - F (FAST) — быстрая серия.

- HC (High Speed *cmos*) — высокоскоростная КМОП–серия.
- HCT (High Speed *cmos* with *ttl* inputs) — серия HC, совместимая по входу с ТТЛ.
- AC (Advanced *cmos*) — улучшенная серия КМОП.
- ACT (Advanced *cmos* with *ttl* inputs) — серия AC, совместимая по входу с ТТЛ.
- BCT (BiCMOS Technology) — серия с БиКМОП–технологией.
- ABT (Advanced BiCMOS Technology) — улучшенная серия с БиКМОП–технологией.
- LVT (Low Voltage Technology) — серия с низким напряжением питания.

4. Идентификатор специального типа (2 символа) — может отсутствовать.
5. Тип микросхемы (от двух до шести цифр). Перечень некоторых типов микросхем приведен в приложении.
6. Код типа корпуса (от одного до двух символов) — может отсутствовать. Например, N — пластмассовый корпус DIL (DIP), J — керамический DIL (DIC), T — плоский металлический.

   Примеры обозначений: SN74ALS373, SN74ACT7801, SN7400.

Рисунок 7.6. Обозначения отечественных микросхем

Отечественная система обозначений микросхем отличается от рассмотренной довольно существенно (Рисунок 7.6). Основные элементы обозначения следующие:

Буква К обозначает микросхемы широкого применения, для микросхем военного назначения буква отсутствует.

Тип корпуса микросхемы (один символ) — может отсутствовать. Например, Р — пластмассовый корпус, М — керамический, Б — бескорпусная микросхема.

Номер серии микросхем (от трех до четырех цифр).

Функция микросхемы (две буквы).

Номер микросхемы (от одной до трех цифр). Таблица функций и номеров микросхем, а также таблица их соответствия зарубежным аналогам приведены в приложении.

Например, КР1533ЛА3, КР531ИЕ17, КР1554ИР47.

Главное достоинство отечественной системы обозначений состоит в том, что *по* обозначению микросхемы можно легко понять ее функцию. Зато в системе обозначений Texas Instruments виден тип серии с его особенностями.

Таблица 2.2. Сравнение параметров одинаковых микросхем в разных стандартных сериях

	К155ЛА3 (SN7400N)	К555ЛА3 (SN74LS00N)	КР1533ЛА3 (SN74ALS00N)	КР1554ЛА3 (SN74AC00N)
t_{PLH}, нс не более	22	15	11	8,5
t_{PHL}, нс не более	15	15	8	7,0
I_{IL}, мА не более	-1,6	-0,45	-0,1	-0,001
I_{IH}, мА не более	0,04	0,02	0,02	0,001
I_{OL}, мА не менее	16	8	15	86
I_{OH}, мА не менее	-0,4	-0,4	-0,4	-75
U_{OL}, В не более	0,4	0,5	0,5	0,3
U_{OH}, В не менее	2,4	2,7	2,5	4,4
I_{CC}, мА не более	12	4,4	3	0,04

На первом уровне представления (логическая модель) серии не различаются ничем. То есть одинаковые микросхемы разных серий работают *по* одним и тем же таблицам истинности, по одним и тем же алгоритмам. Правда, надо учитывать, что некоторые микросхемы имеются только в одной из серий, а некоторых нет в нескольких сериях.

На втором уровне представления (модель с учетом задержек) серии отличаются величиной задержки распространения сигнала. Это различие может быть довольно существенным. Поэтому в тех схемах, где величина задержки принципиальна, надо использовать микросхемы более быстрых серий (Рисунок 7.3).

На третьем уровне представления (электрическая модель) серии различаются величинами входных и выходных токов и напряжений, а также, что не менее важно, токами потребления (Рисунок 7.3). Поэтому в тех устройствах, где эти параметры принципиальны, надо применять микросхемы, обеспечивающие, к примеру, низкие входные токи, высокие выходные токи и малое потребление.

Серия К155 (SN74) — это наиболее старая серия, которая постепенно снимется с производства. Она отличается не слишком хорошими параметрами *по* сравнению с другими сериями. С этой классической серией принято сравнивать все остальные.

Серия К555 (SN74LS) отличается от серии К155 малыми входными токами и меньшей потребляемой мощностью (ток потребления — почти втрое меньше, чем у К155). По быстродействию (по временам задержек) она близ-

ка к К155.

Серия КР531 (SN74S) отличается высоким быстродействием (ее задержки примерно в 3–4 раза меньше, чем у серии К155), но большими входными токами (на 25% больше, чем у К155) и большой потребляемой мощностью (ток потребления — больше в полтора раза по сравнению с К155).

Серия КР1533 (SN74ALS) отличается повышенным примерно вдвое по сравнению с К155 быстродействием и малой потребляемой мощностью (в четыре раза меньше, чем у К155). Входные токи еще меньше, чем у К555.

Серия КР1531 (SN74F) отличается высоким быстродействием (на уровне КР531), но малой потребляемой мощностью. Входные токи и ток потребления примерно вдвое меньше, чем у К155.

Серия КР1554 (SN74AC) отличается от всех предыдущих тем, что она выполнена по КМОП-технологии. Поэтому она имеет сверхмалые входные токи и сверхмалое потребление при малых рабочих частотах. Задержки примерно вдвое меньше, чем у К155.

Наибольшим разнообразием имеющихся микросхем отличаются серии К155 и КР1533, наименьшим — КР1531 и КР1554.

Надо отметить, что приведенные здесь соотношения *по* быстродействию стандартных серий довольно приблизительны и верны не для всех разновидностей микросхем, имеющихся в разных сериях. Точные значения задержек необходимо смотреть в справочниках, причем желательно в фирменных справочных материалах.

Микросхемы разных серий обычно легко сопрягаются между собой, то есть сигналы с выходов микросхем одной серии можно смело подавать на входы микросхем другой серии. Одно из исключений — соединение выходов ТТЛ-микросхем со входами КМОП-микросхем серии КР1554 (74AC). При таком соединении необходимо применение *резистора* номиналом 560 Ом между сигналом и напряжением питания (Рисунок 7.7).

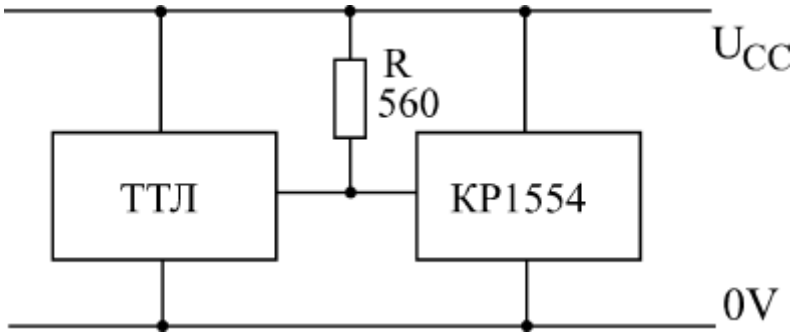


Рисунок 7.7. Сопряжение TTL с CMOS

При выборе той или иной серии микросхем следует также учитывать, что микросхемы мощной и быстрой серии КР531 создают высокий уровень помех *по* шинам питания, а микросхемы маломощной серии К555 очень чувствительны к таким помехам. Поэтому серию КР531 рекомендуется использо-

вать только в крайних случаях, при необходимости получения очень высокого быстродействия. Не рекомендуется также применять в одном устройстве мощные быстродействующие микросхемы и маломощные микросхемы.

Корпуса цифровых микросхем

Большинство микросхем имеют корпус, то есть прямоугольный *контейнер* (пластмассовый, керамический, металлокерамический) с металлическими выводами (ножками). Предложено множество различных типов корпусов, но наибольшее распространение получили два основных типа:

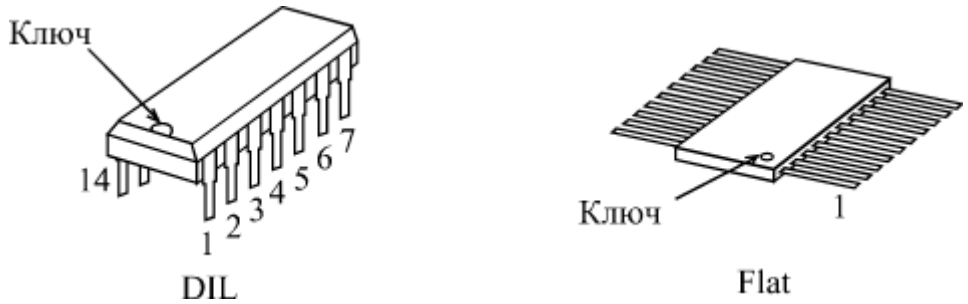


Рисунок 7.8. Примеры корпусов DIL и Flat

Корпус с двухрядным вертикальным расположением выводов, например, DIP (Dual In Line Package, Plastic) — пластмассовый корпус, DIC (Dual In Line Package, Ceramic) — керамический корпус. Общее название для таких корпусов — DIL (Рисунок 8.8). Расстояние между выводами составляет 0,1 дюйма (2,54 мм). Расстояние между рядами выводов зависит от количества выводов.

Корпус с двухрядным плоскостным расположением выводов, например, FP (Flat-Package, Plastic) — пластмассовый плоский корпус, FPC (Flat-Package, Ceramic) — керамический плоский корпус. Общее название для таких корпусов — Flat (Рисунок 2.8). Расстояние между выводами составляет 0,05 дюйма (1,27 мм) или 0,025 дюйма (0,628 мм).

Номера выводов всех корпусов отсчитываются начиная с вывода, помеченного ключом, *по* направлению против часовой стрелки (если смотреть на микросхему сверху). Ключом может служить вырез на одной из сторон микросхемы, точка около первого вывода или утолщение первого вывода (Рисунок 2.8). Первый вывод может находиться в левом верхнем или в правом нижнем углу (в зависимости от того, как повернут корпус). Микросхемы обычно имеют стандартное число выводов из ряда: 4, 8, 14, 16, 20, 24, 28,.… Для микросхем стандартных цифровых серий используются корпуса с количеством выводов начиная с 14.

Назначение каждого из выводов микросхемы приводится в справочниках *по* микросхемам, которых сейчас имеется множество. Правда, лучше ориентироваться на справочники, издаваемые непосредственно фирмами-изготовителями. В данном курсе назначение выводов не приводится.

Отечественные микросхемы выпускаются в корпусах, очень похожих на *DIL* и *Flat*, но расстояния между их выводами вычисляются *по* метрической

шкале и поэтому чуть-чуть отличаются от принятых за рубежом. Например, 2,5 мм вместо 2,54 мм, 1,25 мм вместо 1,27 мм и т.д. Для корпусов с малым числом выводов (до 20) это не слишком существенно, но для больших корпусов расхождение в расстоянии может стать существенным. В результате на плату, рассчитанную на зарубежные микросхемы, нельзя поставить отечественные микросхемы, и наоборот.

Двоичное кодирование

Одиночный цифровой сигнал не слишком информативен, ведь он может принимать только два значения: нуль и единица. Поэтому в тех случаях, когда необходимо передавать, обрабатывать или хранить большие объемы информации, обычно применяют несколько параллельных цифровых сигналов. При этом все эти сигналы должны рассматриваться только одновременно, каждый из них *по* отдельности не имеет смысла. В таких случаях говорят о двоичных кодах, то есть о кодах, образованных цифровыми (логическими, двоичными) сигналами. Каждый из логических сигналов, входящих в код, называется *разрядом* . Чем больше разрядов входит в код, тем больше значений может принимать данный код.

В отличие от привычного для нас десятичного кодирования чисел, то есть кода с основанием десять, при двоичном кодировании в основании кода лежит число два (Рисунок 2.9). То есть каждая цифра кода (каждый разряд) двоичного кода может принимать не десять значений (как в десятичном коде: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9), а всего лишь два — 0 и 1. Система позиционной записи остается такой же, то есть справа пишется самый младший разряд, а слева — самый старший. Но если в десятичной системе *вес* каждого следующего разряда больше веса предыдущего в десять раз, то в двоичной системе (при двоичном кодировании) — в два раза. Каждый разряд двоичного кода называется бит (от английского "Binary Digit" — "двоичное число").

Рисунок 7.9. Десятичное и двоичное кодирование

В табл. 2.3 показано соответствие первых двадцати чисел в десятичной и двоичной системах.

Из таблицы видно, что требуемое количество разрядов двоичного кода значительно больше, чем требуемое количество разрядов десятичного кода. Максимально возможное число при количестве разрядов, равном трем, составляет при десятичной системе 999, а при двоичной — всего лишь 7 (то есть 111 в двоичном коде). В общем случае n-разрядное двоичное число мо-

жет принимать 2^n различных значений, а n-разрядное десятичное число — 10^n значений. То есть запись больших двоичных чисел (с количеством разрядов больше десяти) становится не слишком удобной.

Таблица 7.3. Соответствие чисел в десятичной и двоичной системах

Десятичная система	Двоичная система	Десятичная система	Двоичная система
0	0	10	1010
1	1	11	1011
2	10	12	1100
3	11	13	1101
4	100	14	1110
5	101	15	1111
6	110	16	10000
7	111	17	10001
8	1000	18	10010
9	1001	19	10011

Для того чтобы упростить *запись* двоичных чисел, была предложена так называемая шестнадцатиричная система (16-ричное *кодирование*). В этом случае все двоичные разряды разбиваются на группы *по* четыре разряда (начиная с младшего), а затем уже каждая *группа* кодируется одним символом. Каждая такая *группа* называется **полубайтом** (или **нибблом, тетрадой**), а две группы (8 разрядов) — байтом. Из табл. 2.3 видно, что 4-разрядное двоичное число может принимать 16 разных значений (от 0 до 15). Поэтому требуемое число символов для шестнадцатиричного кода тоже равно 16, откуда и происходит название кода. В качестве первых 10 символов берутся цифры от 0 до 9, а затем используются 6 начальных заглавных букв латинского алфавита: A, B, C, D, E, F.

Рисунок 7.10. Двоичная и 16-ричная запись числа

В табл. 7.4 приведены примеры 16-ричного кодирования первых 20 чисел (в скобках приведены двоичные числа), а на Рисунок 2.10 показан пример записи двоичного числа в 16-ричном виде. Для обозначения 16-ричного кодирования иногда применяют букву "h" или "H" (от английского Hexadecimal) в конце числа, например, запись A17F h обозначает 16-

ричное число A17F. Здесь A1 представляет собой старший байт числа, а 7F — младший байт числа. Все число (в нашем случае — двухбайтовое) называется словом.

Таблица 7.4. 16-ричная система кодирования

Десятичная система	16-ричная система	Десятичная система	16-ричная система
0	0 (0000)	10	A (1010)
1	1(0001)	11	B (1011)
2	2 (0010)	12	C (1100)
3	3 (0011)	13	D (1101)
4	4 (0100)	14	E (1110)
5	5 (0101)	15	F (1111)
6	6 (0110)	16	10 (00010000)
7	7 (0111)	17	11 (00010001)
8	8 (1000)	18	12 (00010010)
9	9 (1001)	19	13 (00010011)

Для перевода 16-ричного числа в десятичное необходимо умножить значение младшего (нулевого) разряда на единицу, значение следующего (первого) разряда на 16, второго разряда на 256 (16^2) и т.д., а затем сложить все произведения. Например, возьмем число A17F:

$$A17F=F*16^0 + 7*16^1 + 1*16^2 + A*16^3 = 15*1 + 7*16+1*256+10*4096=41343$$

Таблица 2.5. 8-ричная система кодирования

Десятичная система	8-ричная система	Десятичная система	8-ричная система
0	0 (000)	10	12 (001010)
1	1(001)	11	13 (001011)
2	2 (010)	12	14 (001100)
3	3 (011)	13	15 (001101)
4	4 (100)	14	16 (001110)
5	5 (101)	15	17 (001111)
6	6 (110)	16	20 (010000)
7	7 (111)	17	21 (010001)
8	10 (001000)	18	22 (010010)
9	11 (001001)	19	23 (010011)

Но каждому специалисту по цифровой аппаратуре (разработчику, оператору, ремонтнику, программисту и т.д.) необходимо научиться так же свободно обращаться с 16-ричной и двоичной системами, как и с обычной деся-

тичной, чтобы никаких переводов из системы в систему не требовалось.

Значительно реже, чем 16-ричное, используется восьмеричное *кодирование*, которое строится *по* такому же принципу, что и 16-ричное, но двоичные разряды разбиваются на группы *по* три разряда. Каждая *группа* (разряд кода) затем обозначается одним символом. Каждый разряд 8-ричного кода может принимать восемь значений: 0, 1, 2, 3, 4, 5, 6, 7 (табл.7.5).

Помимо рассмотренных кодов, существует также и так называемое двоично-десятичное *представление* чисел. Как и в 16-ричном коде, в двоично-десятичном коде каждому разряду кода соответствует четыре двоичных разряда, однако каждая *группа* из четырех двоичных разрядов может принимать не шестнадцать, а только десять значений, кодируемых символами 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. То есть одному десятичному разряду соответствует четыре двоичных. В результате получается, что написание чисел в двоично-десятичном коде ничем не отличается от написания в обычном десятичном коде (табл. 7.6), но в реальности это всего лишь специальный двоичный код, каждый разряд которого может принимать только два значения: 0 и 1. Двоично-десятичный код иногда очень удобен для организации десятичных цифровых индикаторов и табло.

Таблица 7.6. Двоично-десятичная система кодирования

Десятичная система	**Двоично-десятичная система**	**Десятичная система**	**Двоично-десятичная система**
0	0 (0000)	10	10 (00010000)
1	1(0001)	11	11 (00010001)
2	2 (0010)	12	12 (00010010)
3	3 (0011)	13	13 (00010011)
4	4 (0100)	14	14 (00010100)
5	5 (0101)	15	15 (00010101)
6	6 (0110)	16	16 (00010110)
7	7 (0111)	17	17 (00010111)
8	8 (1000)	18	18 (00011000)
9	9 (1001)	19	19 (00011001)

В двоичном коде над числами можно проделывать любые арифметические операции: сложение, вычитание, умножение, деление.

Рассмотрим, например, *сложение* двух 4-разрядных двоичных чисел. Пусть надо сложить число 0111 (десятичное 7) и 1011 (десятичное 11). Сложение этих чисел не сложнее, чем в десятичном представлении.

При сложении 0 и 0 получаем 0, при сложении 1 и 0 получаем 1, при сложении 1 и 1 получаем 0 и перенос в следующий разряд 1. Результат — 10010 (десятичное 18). При сложении любых двух n-разрядных двоичных чи-

сел может получиться n-разрядное или (n+1) -разрядное число.

Точно так же производится вычитание. Пусть из числа 10010 (18) надо вычесть число 0111 (7). Записываем числа с выравниванием по младшему разряду и вычитаем точно так же, как в случае десятичной системы:

При вычитании 0 из 0 получаем 0, при вычитании 0 из 1 получаем 1, при вычитании 1 из 1 получаем 0, при вычитании 1 из 0 получаем 1 и заем 1 в следующем разряде. Результат — 1011 (десятичное 11).

При вычитании возможно получение отрицательных чисел, поэтому необходимо использовать двоичное представление отрицательных чисел.

Для одновременного представления как двоичных положительных, так и двоичных отрицательных чисел чаще всего используется так называемый дополнительный код. Отрицательные числа в этом коде выражаются таким числом, которое, будучи сложено с положительным числом такой же величины, даст в результате нуль. Для того чтобы получить отрицательное число, надо поменять все биты такого же положительного числа на противоположные (0 на 1, 1 на 0) и прибавить к результату 1. Например, запишем число –5. Число 5 в двоичном коде выглядит 0101. Заменяем биты на противоположные: 1010 и прибавляем единицу: 1011. Суммируем результат с исходным числом: 1011 + 0101 = 0000 (перенос в пятый разряд игнорируем).

Помимо стандартных арифметических операций, в двоичной системе счисления используются и некоторые специфические операции, например, сложение по модулю 2. Эта операция (обозначается А) является побитовой, то есть никаких переносов из разряда в разряд и заемов в старших разрядах здесь не существует.

Среди других побитовых операций над двоичными числами можно отметить функцию И и функцию ИЛИ. *Функция* И дает в результате единицу только тогда, когда в соответствующих битах двух исходных чисел обе единицы, в противном случае результат —0. *Функция* ИЛИ дает в результате единицу тогда, когда хотя бы один из соответствующих битов исходных чисел равен 1, в противном случае результат 0.

Функции цифровых устройств

Любое цифровое устройство от самого простейшего до самого сложного всегда действует *по* одному и тому же принципу (Рисунок 2.11). Оно принимает входные сигналы, выполняет их обработку, передачу, хранение и выдает выходные сигналы. При этом совсем не обязательно любое изменение входных сигналов приводит к немедленному и однозначному изменению выходных сигналов. Реакция устройства может быть очень сложной, отложенной по времени, неочевидной, но суть от этого не меняется.

В качестве входных сигналов нашего устройства могут выступать сигналы с выходов других цифровых устройств, с тумблеров и клавиш или с датчиков физических величин. Причем в последнем случае, как правило, необходимо преобразование аналоговых сигналов с датчиков в потоки цифровых кодов (Рисунок 7.12) с помощью аналого-цифровых преобразователей

(АЦП). Например, в случае персонального компьютера входными сигналами являются сигналы с клавиатуры, с датчиков перемещения мыши, с микрофона (давление воздуха, то есть звук, преобразуется в аналоговый электрический сигнал, а затем — в цифровые коды), из кабеля локальной сети и т.д.

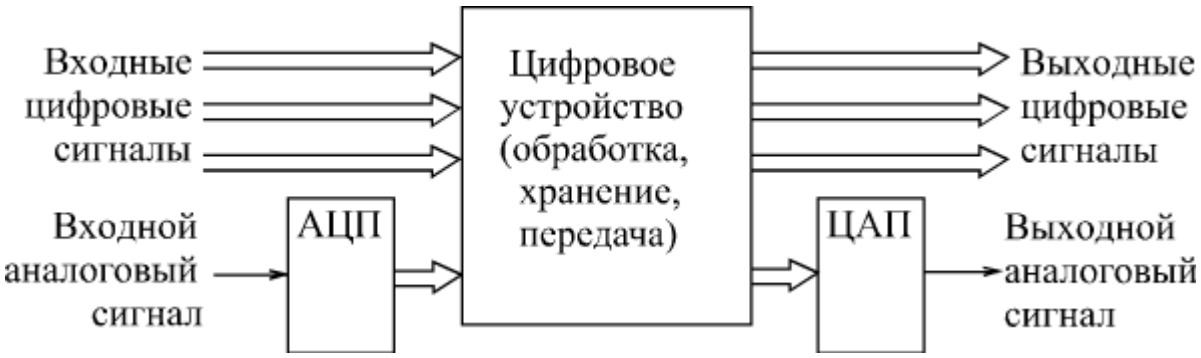


Рисунок 7.11. Включение цифрового устройства

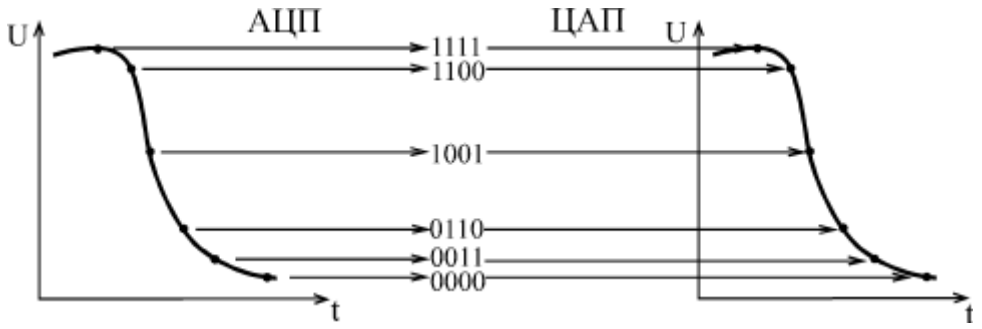


Рисунок 7.12. Аналого-цифровое и цифро-аналоговое преобразование

Выходные сигналы цифрового устройства могут предназначаться для подачи на другие цифровые устройства, для индикации (на экране монитора, на цифровом индикаторе и т.д.), а также для формирования физических величин. Причем в последнем случае необходимо преобразовывать потоки кодов с цифрового устройства в непрерывные (аналоговые) сигналы (Рисунок 7.12) с помощью цифро-аналоговых преобразователей (ЦАП) и в физические величины. Например, в случае персонального компьютера выходными сигналами будут: сигналы, подаваемые компьютером на принтер; сигналы, идущие на видеомонитор (аналоговые или цифровые); звук, воспроизводимый динамиками компьютера (потоки кодов с компьютера преобразуются в аналоговый *электрический сигнал*, который затем преобразуется в давление воздуха — звук).

Одно цифровое устройство может состоять из нескольких более простых цифровых устройств. Часто эти составные элементы называют блоками, модулями, узлами, частями. Если объединяются несколько сложных цифровых устройств, то говорят уже о цифровых системах, комплексах, установках. Мы в основном будем использовать термин "устройство", как занимаю-

щий промежуточное положение.

Связь между входными и выходными сигналами может быть жесткой, неизменной или гибко изменяемой (программируемой). То есть цифровое устройство может работать *по* жесткому, раз и навсегда установленному алгоритму или *по* алгоритму программируемому. Как правило, при этом выполняется один очень простой принцип: чем больше возможностей для изменения связи входных и выходных сигналов, чем больше возможностей изменения алгоритма работы, тем цифровое устройство будет медленнее. Речь в данном случае, конечно же, идет о предельно достижимом быстродействии.

Иначе говоря, простые устройства с жесткой логикой работы всегда могут быть сконструированы с более высоким быстродействием *по* сравнению с программируемыми, гибкими устройствами со сложным алгоритмом работы. Жесткая логика также обеспечивает малый объем аппаратуры (малые аппаратурные затраты) для реализации простых функций. Зато программируемые, интеллектуальные устройства обеспечивают более высокую гибкость и меньшую стоимость при необходимости сложной обработки. А для реализации простых функций они часто оказываются избыточно сложными. Так что выбор между двумя этими типами цифровых устройств зависит от конкретной решаемой задачи.

Значительное число задач может быть решено как чисто аппаратным путем (с помощью устройств на жесткой логике), так и программно-аппаратным путем (с помощью программируемых устройств). В таких случаях надо смотреть, какие характеристики устройства являются самыми важными: скорость работы, стоимость, гибкость, простота проектирования и т.д. — и в зависимости от этого выбирать то или иное решение, так или иначе перераспределять функции между программным обеспечением и аппаратурой.

8 СИНТЕЗ ФУНКЦИОНАЛЬНЫХ СХЕМ КОМБИНАЦИОННОГО ТИПА

Цель занятия:

Закрепить знания, полученные на лекциях и практических занятиях по методике синтеза логических схем и типовых дискретных устройств без памяти, получить навыки в сборке, отладке и экспериментальном исследовании логических схем на ИМС, сумматоров выполненных на элементах универсальных базисов ИЛИ-НЕ, И - НЕ.

Методические указания по выполенению практического занятия

Дискретным автоматом называется устройство, в котором осуществляется преобразование дискретной информации. Для представления информации в дискретном автомате обычно используется двоичный алфавит 0 и 1.

В дискретном автомате без памяти значения двоичных переменных на выходах автомата в данный момент времени определяются значениями двоичных переменных на входах автомата в тот же момент времени и не зависят от значений входных переменных в предыдущие моменты времени.

Дискретные автоматы без памяти, иначе называемые комбинационными или логическими схемами, используются в цифровых вычислительных машинах, автоматизированных системах управления в качестве устройств для преобразования информации.

Синтез дискретного автомата без памяти заключается в построении функциональной электрической схемы автомата, реализующей заданную функциональную зависимость между входными и выходными двоичными переменными и отвечающей предъявленным требованиям в отношении типов элементов, используемых в схеме.

В качестве логических элементов для построения функциональной схемы автомата без памяти используются элементы, реализующие простейшие функции алгебры логики. Для реализации автоматом без памяти любой сколь угодно сложной функциональной зависимости между входными и выходными двоичными переменными необходимо, чтобы простейшие функции, реализуемые используемыми элементами, образовывали функционально полную систему логических функций.

Функциональная зависимость между входными и выходными двоичными переменными первоначально может быть описана с использованием слов обычного разговорного языка. Такое описание функциональной зависимости называют содержательным описанием. В процессе синтеза автомата разработчик от содержательного описания переходит к формализованным описаниям, имеющим вид таблиц или формул алгебры логики. Таким образом, исходными данными для синтеза дискретного автомата без памяти, как правило, являются:

1) описание функциональной зависимости между входными и выходными двоичными переменными;

2) требования к функциональной электрической схеме автомата в отношении типов используемых логических элементов.

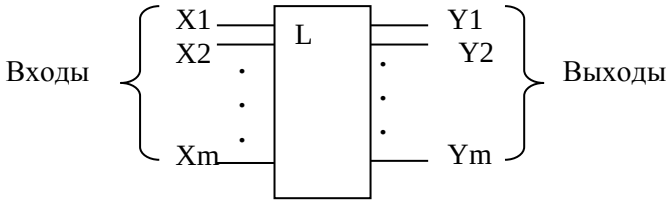


Рисунок 8.1. – Комбинационое цифровое усторйство

В общем случае математическая модель дискретного автомата без памяти может быть представлена (n, m) - полюсником (рис. 8.1), который реализует логическую операцию однозначного преобразования двоичного n-разрядного кода XI, X2, ..., Xn в другой вид Y1, Y2, ..., Ym по заданному закону, где X1, X2,..., Xn - элементы водного кода, принимающие значения 0 или 1; Yl, Y2, ... , Ym - элементы выходного кода.

Интегральные логические элементы завоевывают в последние годы основные позиции в электронном машиностроении в силу тех достоинств, которыми они обладают.

Электронной промышленностью выпускается большое число серий интегральных микросхем для цифровых устройств. Микросхемы одной серии изготавливаются по единой технологии и согласованы по своим электрическим и конструктивным параметрам для совместного использования в электронной аппаратуре.

Номенклатура (набор) микросхем в одной серии обеспечивает выполнение любых логических операций.

В данной практическому занятию предполагается синтез дискретных автоматов без памяти проводить на элементах ТТЛ серии К155.

Задача синтеза одновыходной логической схемы сводится к задаче синтеза (n,1) - полюсника. Сначала одним из способов задается некоторая логическая функция Y = f (X1, X2,.., Xn), а затем производится ее минимизация и представление в требуемом базисе.

После этого переходят к составлению функциональной электрической схемы, закон функционирования которой задан логической функцией Y = f (X1, X2,..., Xn).

Пример. Синтезировать одновыходную логическую схему, закон функционирования которой задан в числовой форме

$$f1(A, D, C, Д) = \sum_0 (5,6,8,9,10,11,13).$$

Решение

1. Зададим закон функционирования одновыходной, логической схемы таблицей истинности (таблица 9.1).

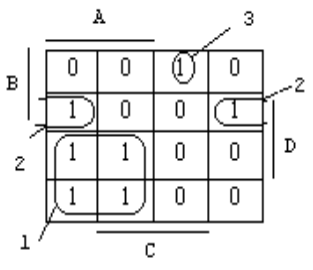
2. Представим закон функционирования логической схемы в канонической форме (СДНФ f1 и СКНФ f1).

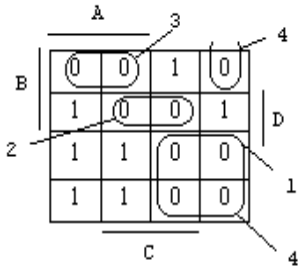
Таблица 8.1

№ наборов	Входные переменные и их веса				Выходная функция
	8	4	2	1	f1(A,B,C,Д)
	A	B	C	Д	
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

$$СДНФf1(A,B,C,Д) = \overline{A}B\overline{C}Д + \overline{A}BC\overline{Д} + A\overline{B}\overline{C}\overline{Д} + A\overline{B}\overline{C}Д + A\overline{B}C\overline{Д} + A\overline{B}CД +$$
$$+ AB\overline{C}Д$$
$$СКНФf1(A,B,C,Д) = (A + B + C + Д)\cdot(A + B + C + \overline{Д})\cdot A + B + \overline{C} + Д)\cdot(A +$$
$$+ B + \overline{C} + \overline{Д})(A + \overline{B} + C + Д)\cdot(A + \overline{B} + \overline{C} + \overline{Д})\cdot(\overline{A} + \overline{B} + C + Д)\cdot(\overline{A} + \overline{B} + \overline{C} + Д)\cdot$$
$$\cdot(\overline{A} + \overline{B} + \overline{C} + \overline{Д}).$$

3. Минимизируем логическую функцию F1(A,B,C,Д), используя метод карт Карно (таблицы 8.2 и 89.3).





$$МДНФf1(A,B,C,Д) = \frac{A\overline{B}}{1} + \frac{B\overline{C}Д}{2} + \frac{\overline{A}BC\overline{Д}}{3} - 9букв \quad (9.1)$$

$$МКНФf1(A,B,C,Д) = \frac{(A+B)}{1} \cdot \frac{(\overline{B}+\overline{C}+\overline{Д})}{2} \cdot \frac{(\overline{A}+\overline{B}+Д)}{3} \cdot \frac{(A+C+Д)}{4} - 11букв \quad (9.2)$$

4. Представим МДНФ f1(A,B,C,Д) в универсальном базисе И-НЕ, считая что располагаем как переменными, так и их инверсиями.

$$f_1(A,B,C,Д) = (A|\overline{B})|(B|\overline{C}|Д)|(\overline{A}|B|C|\overline{Д}) \quad (9.3)$$

5. Представим МКНФ f1(A,B,C,Д) в универсальном базисе ИЛИ-НЕ.

$$f_1(A,B,C,Д) = (A \downarrow B) \downarrow (\overline{B} \downarrow \overline{C} \downarrow \overline{Д}) \downarrow (\overline{A} \downarrow \overline{B} \downarrow Д) \downarrow (A \downarrow C \downarrow Д) \quad (9.4)$$

6. Составим функциональную электрическую схему одновыходной логической схемы, закон функционирования которой задан функцией f1 (A,B,C,Д).

В принципе ее можно составить на элементах основною базиса И, ИЛИ, НЕ по выражениям (9.1) или (9.2), на элементах универсального базиса И - НЕ по выражению (9.3), либо на элементах другого универсального базиса ИЛИ - НЕ по выражению (9.4).

Однако операции И, ИЛИ, НЕ, через которые выражается логическая функция в выражениях (9.1) и (9.2), не лежат в основе ни одной известной серии ИМС. Поэтому возможно реализовать логическую функцию с помощью одной из схем, построенной в базисе И-НЕ либо ИЛИ - НЕ.

Выражение (9.3) короче (9.4); поэтому и функциональная схема, построенная на элементах И - НЕ, будет экономичнее (суммарное число входов всех логических элементов меньше).

Функциональная электрическая схема будет иметь вид рис. 8.2).

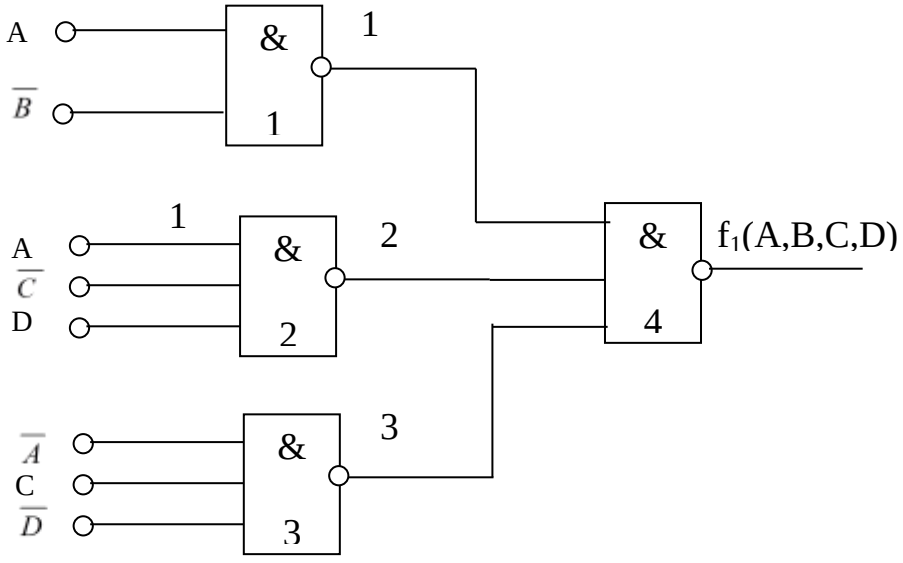


Рисунок 8.2 – Одновыходная комбинационная схема в базисе Шеффера (И-

НЕ)

7. Построим временные диаграммы работы синтезированной схемы (рис. 8.3)

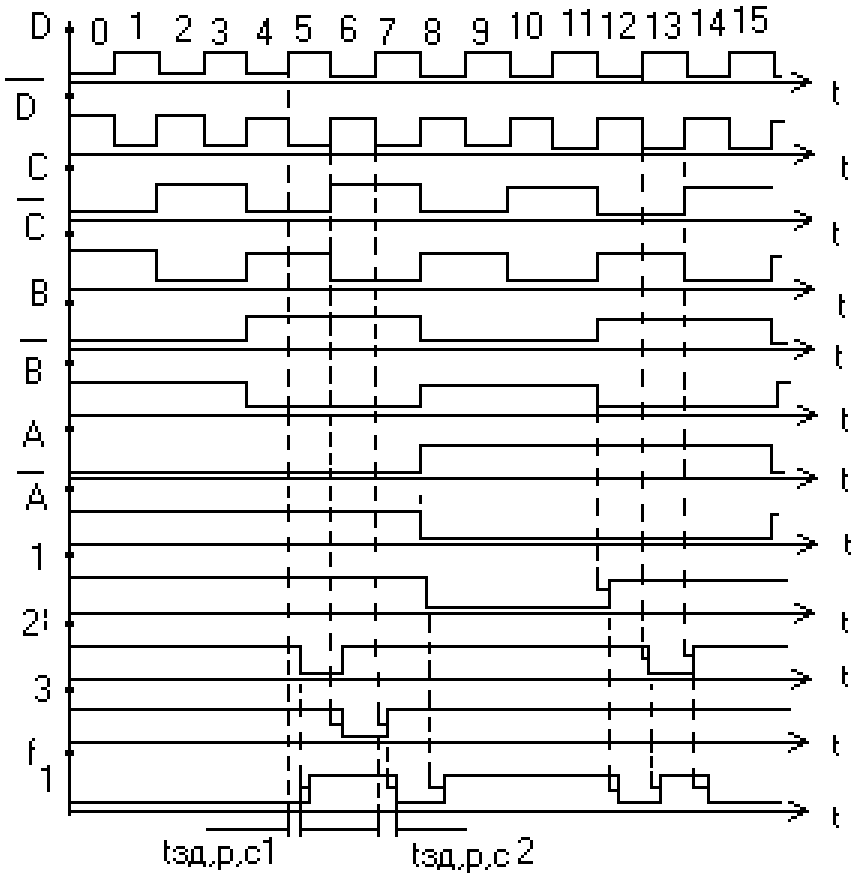


Рисунок 8.3 – Временные диаграммы работы одновыходной комбинационной схемы

Анализ временных диаграмм показывает, что логическая схема синтезирована верно, так как выходной сигнал f1 принимает значение единицы на наборах 5,6,8,9,10,11,13,что соответствует условию примера.

9 СИНТЕЗ И АНАЛИЗ АВТОМАТОВ С ПАМЯТЬЮ

Цель занятия:

Закрепить знания, полученные на лекциях по синтезу автоматов с памятью.

Методические указания по выполенению практического занятия

Конечными автоматами (автоматами с памятью или последовательными схемами) называются цифровые автоматы с памятью, имеющие конечное число состояний, которые отличаются от автоматов без памяти (комбинационных или логических) схем тем, что выходной сигнал зависит не только от входного сигнала, но и от значений входных сигналов, поданных на автомат в предыдущие моменты времени.

Конечный автомат способен «запоминать» состояние, в которое он был поставлен в предыдущем такте его работы, и определять состояние автомата в последующие такты. Наличие памяти у автомата значительно расширяет его логические возможности, что позволяет синтезировать практически любые алгоритмы переработки информации, как бы сложны они не были.

Символы 0 и 1 двухзначной логики называются буквами и образуют алфавит. Совокупность всех слов, поступивших на вход цифрового автомата, образует множество слов входного алфавита. Связь множества слов входного алфавита со множеством слов выходного алфавита называют алфавитными операторами. Слова могут обозначаться или набором букв выбранного алфавита, или какой-либо одной буквой другого алфавита. Как правило, наборы букв располагаются в определенной последовательности, соответствующей их циркуляции в автомате.

Алфавитные операторы, отражающие функционирование цифровых автоматов, т.е. соответствие между словами входного и выходного алфавита, могут выражаться в виде уравнений, таблиц и граф-схем.

Алфавитные операторы, записанные в виде канонических уравнений, представляют систему рекуррентных соотношений. Автоматы Мили задаются операторами:

$$y(t) = \varphi[a(t), x(t)];$$
$$a(t+1) = f[a(t), x(t)], \qquad (2.2.1)$$

где t – такты автомата;

x – входные сигналы;

y – выходные сигналы;

a – сигналы внутреннего состояния автомата.

Выражение $y(t) = \varphi[a(t), x(t)]$, определяющее выходной сигнал автомата, называют функцией выходов, а выражение $a(t+1) = f[a(t), x(t)]$, определяющее состояние автомата в момент дискретного времени $t+1$, называется функцией переходов. Зная эти функции и начальное состояние автомата, можно определить выходное слово и внутреннее состояние в зависимости от

входного.

Вместо канонических уравнений алфавитные операторы можно записать в виде таблиц переходов и выходов. Таблица переходов (табл. 9.1) отражает состояние автомата, соответствующее такту t +1. Таблица выходов (табл. 9.2) позволяет определить значение выходного слова.

Таблица 9.1

xi \ aj	a0	a1	a2
x0	a0	a0	a0
x1	a1	a1	a1

Таблица 9.2

xi \ aj	a0	a1	a2
x0	y0	y0	y0
x1	y0	y1	y0

Входами этих таблиц являются значения x_i входных сигналов и a_j состояний автомата в момент времени t. В клетках табл. 2.2.1, находящиеся на пересечении соответствующих столбцов a_j и строк x_i помещаются значения внутренних состояний $a(t+1) = f[a(t), x(t)]$, а в аналогичных клетках табл. 9.2– выходные значения $y(t) = \varphi[a(t), x(t)]$.

Состояние a_0 всегда берется в качестве начального и называется исходным состоянием. Возможно совмещение обеих таблиц в одной (табл.9.3), что улучшает наглядность и облегчает работу.

Таблица 9.3

xi \ aj	a0	a1	a2
x0	a0 / y0	a0 / y0	a0 / y0
x1	a1 / y0	a2 / y1	a2 / y0

По таблицам переходов и выходов легко определяется последовательность выходных сигналов и переходов из одного состояния в другое, как реакция на любое входное слово. Для наглядности и удобства при анализе и синтезе автоматов используются графы. На рис. 9.1 приведен граф автомата, заданного табл. 9.3.

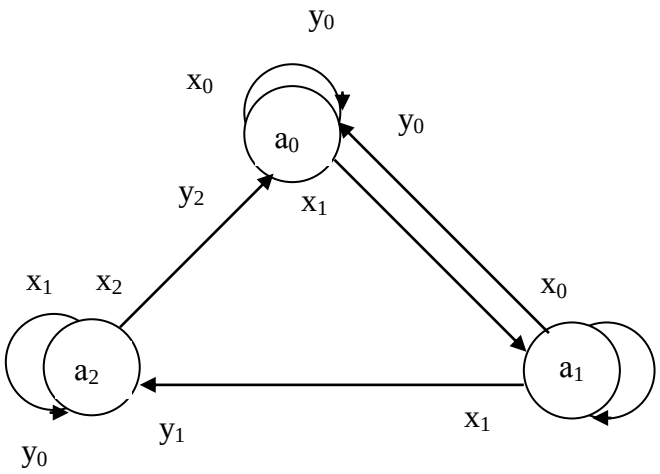


Рисунок. 9.1

Вершины графа соответствуют состояниям автомата, стрелки (ребра графа) указывают направления переходов, а отметка ребер графа – входным и выходным сигналам x_i и y_i , которые указывают значения входного сигнала, под действием которого происходит данный переход, и выходного сигнала, который при этом выдается. Направленность графа, которая отображается стрелками, позволяет проследить переходы состояний автомата при подаче на его вход слов. Если ребро графа упирается в то же состояние, из которого оно выходит, то это означает, что состояние автомата не изменится при подаче входного слова. Переход от графа к таблице переходов и выходов не представляет затруднений.

Часто встречаются так называемые частичные автоматы, у которых некоторые последовательности входных сигналов никогда не подаются. Последние называются запрещенными входными словами данного автомата. В этом случае функции переходов и выходов определены не для всех пар значений **a** и **x**. При синтезе обычно производят доопределение частичного автомата так, чтобы соответствующая схема содержала минимальное число элементов.

В таблицах переходов и выходов частичного автомата для запрещенных входных слов состояния автомата не определены и поэтому в соответствующей клетке ставится прочерк. На практике встречаются автоматы Мили и автоматы Мура. Автоматы Мили задаются операторами (2.2.1) и отличаются тем, что их выходные сигналы зависят как от состояния автомата, так и от значения входного сигнала.

Автоматы Мура задаются операторами

$$y(t) = \varphi[a(t)];$$

$$a(t+1) = f[x(t), a(t)] \qquad (2.2.2)$$

и отличаются тем, что выходные сигналы в данном такте не зависят от входного слова в том же такте и определяются только состоянием a (t).

Задача структурного синтеза конечных автоматов состоит в выборе типов и качества элементарных автоматов и способа соединения их между со-

бой с помощью логических элементов, при котором получается оптимальная в некотором смысле (минимальная) схема, функционирующая в соответствии с заданными алфавитными операторами.

Обобщенная структурная схема конечного автомата может быть представлена так, как показано на рис. 2.2.3, где Q_1, Q_1, . . . , Q_m – элементарные автоматы (элемент памяти); X, Y, A – алфавиты входов, выходов и состояний соответственно.

Сигналы q_1, q_2, . . . , q_m являются логическими функциями физических входов и состояний конечных автоматов и называются функциями возбуждения элементарных автоматов Q_1, Q_1, . . . , Q_m соответственно.

Функции возбуждения формируются с помощью КС1, а функции выходов – с помощью КС2.

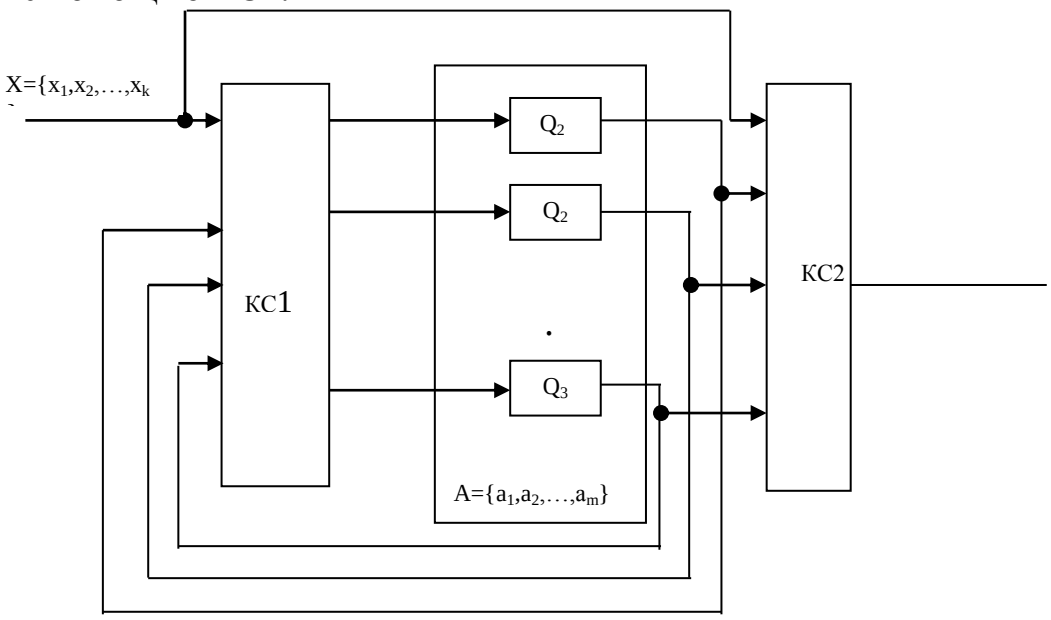


Рисунок. 9.3

Конечная цель структурного синтеза конечного автомата состоит в определении выражений для функции возбуждения q_i всех элементарных автоматов Q_i , а также в определении функций физических выходов и построения на их основе функциональной схемы.

Порядок синтеза:

1. Задание автомата в канонической форме.

2. Выбор структуры сигналов на физических входах и выходах.

3.Кодирование входных и выходных сигналов конечного автомата в соответствии с выбранной (заданной) структурой сигналов на физических входах и выходах, а также кодирование внутренних состояний автомата состояниями элементарных автоматов.

4. Составление кодированных таблиц переходов и выходов (кодированного графа) автомата.

5. Составление таблиц функций возбуждения элементарных автоматов, входящих в синтезируемый конечный автомат. Получение минимальных форм этих функций и функций выходов.

6. Составление структурной схемы конечного автомата на основании минимальных форм функций возбуждения и функций выходов.

Задача 1

Рисунок 9.4.

Провести синтез устройства контроля робота-пылесоса (рисунок 9.4):

- Входной сигнал ОВ поступает с датчика препятствия и равен 1, если обнаружено препятствие на пути движения робота-пылесоса.

- Возможны три типа передвижения: движение вперед, поворот на 90° вправо и поворот на 90° влево.

Функционирование робота-пылесоса описывается следующими правилами.

1. Если обнаружено препятствие, то поворачивать направо до тех пор, пока путь не будет свободен.

2. После этого продолжить движение вперед.

3. При обнаружении следующего препятствия поворачивать налево до тех пор, пока путь не будет свободен.

4. После этого продолжить движение вперед.

И т. д.

Задача 2

Произвести синтез схемы конечного автомата, подсчитывающего число единиц во входных словах и отмечающего факт четности единиц этого числа. Закон функционирования автомата задан графом (рис. 2.2.21). В качестве элементов памяти использовать JK - триггеры.

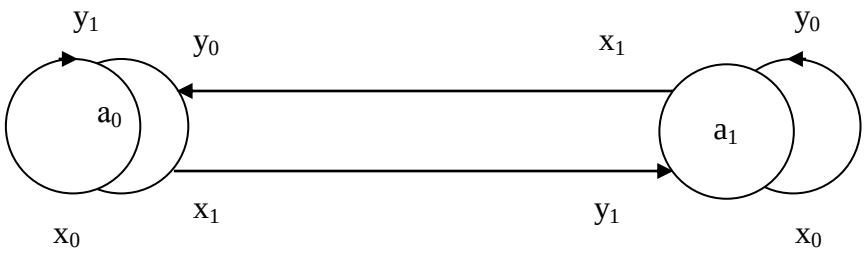


Рисунок 9.5

Задача 3

Спроектировать устройство управления автоматическими воротами. Устройство получает 4 входных сигнала.

1. «Запрос» поступает при нажатии водителем кнопки открытия ворот.

2. «Верх» поступает с датчика состояния ворот. Принимает значение 1 при полностью открытых воротах.

3. «Низ» поступает с датчика состояния ворот. Принимает значение 1 при полностью закрытых воротах.

4. «Занято» поступает с датчика внутри помещения. Принимает значение 0 при пустом помещении.

Устройство генерирует 2 выходных сигнала.

1. ON/OFF ворота неподвижны (ON/OFF=0) или приведены в движение (ON/OFF=1).

2. UP/DOWN показывает направление движения ворот.

2.а Ворота закрываются ON/OFF=1, UP/DOWN=0.

2.б Ворота открываются ON/OFF=1, UP/DOWN=1.

Устройство управления должно реализовывать следующую последовательность операций.

1. Ждать поступления сигнала «Запрос» (Запрос=1).

2. Открыть ворота (ON/OFF=1, UP/DOWN=0).

3. Дождаться, пока ворота откроются полностью (Верх=1).

4. Дождаться, пока автомобиль выедет (Занято=0) и начать закрывать ворота (ON/OFF=1, UP/DOWN=1).

5. Дождаться, пока ворота закроются полностью (Низ=1) и возвратиться в начальное состояние, ожидая нового сигнала «Запрос».

Задача 4

Спроектировать устройство с одним входом X и одним выходом Z, функционирующее согласно графу, изображенному на рисунке 9.6.

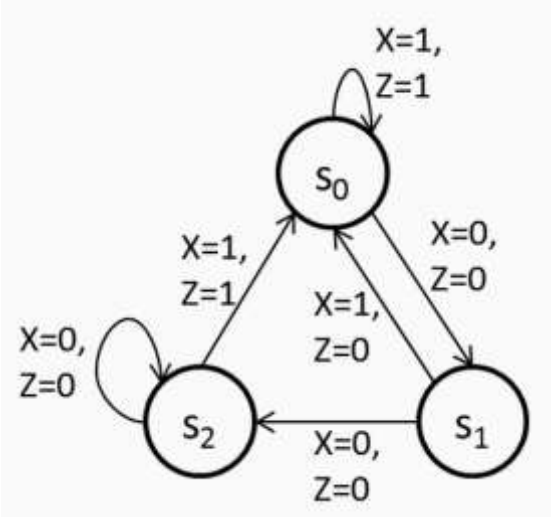


Рисунок 9.6.

Задача 5

Спроектировать устройство с одним входом X и одним выходом Y, которое обнаруживает поступление на вход нечетного количества единиц после двух подряд идущих нулей. Граф устройства изображен на рисунке 9.7.

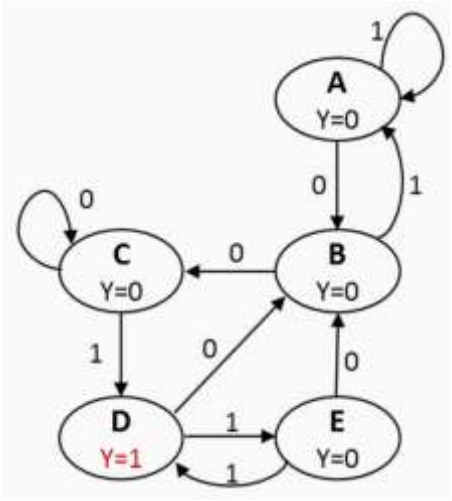


Рисунок 9.7.

10 СТРУКТУРНЫЙ СИНТЕЗ ЛОГИЧЕСКИХ СХЕМ

Цель занятия

Уяснить постановку задачи синтеза комбинационных схем в функционально полных наборах элементов. Закрепить методику и основные этапы процесса структурного синтеза, приобрести первичные умения и навыки построения комбинационных схем в различных функционально полных наборах элементов.

Программа работы:

1. Последовательные и параллельные соединения вентилей.
2. Основные этапы построения комбинационной схемы в выбранном функционально полном наборе.

Методические указания по выполеннению работы

1. Последовательное и параллельное соединение вентилей

Синтез логической схемы – это проектирование этой схемы, если задан закон ее функционирования. В самой общей постановке задача структурного синтеза состоит в построении комбинационных схем, реализующих заданные алгоритмы с помощью простых вентилей – *конъюнктора*, *дизъюнктора*, *инвертора* и др.

В результате структурного синтеза должен быть определен перечень логических элементов, входящих в комбинационную схему, и порядок их соединения между собой. В качестве функционально полной системы функций первоначально можно взять избыточную систему $\{\neg, \&, \vee\}$.

Данное практическое занятие является итоговым по циклу лекций и практических занятий, на которых изучалась алгебра высказываний.

Задачи

Изобразить схемы дискретных сигналов, состоящие из последовательно и параллельно включенных контактов.

Изобразить электрические и стандартные графические изображения (схемы) инвертора, дизъюнктора, конъюнктора, стрелки Пирса и штриха Шеффера.

Привести примеры функционально полных систем функций (связок) алгебры логики. Привести примеры базисов.

Построить комбинационные схемы, заданные функциями:

1. $f(x,y,z) = xyz \vee \overline{xyz} \vee \bar{x}y$;
2. $f(x,y,z) = x(yz \vee \overline{yz}) \vee \bar{x}(\bar{y}z \vee y\bar{z})$;
3. $f(x,y,z,u) = (\bar{x} \vee y) \vee (zy \vee x) \vee u$;
4. $f(x,y,z) = (x \to y)(y \to z)$.

2. Основные этапы построения комбинационной схемы в выбранном функционально полном наборе

Процесс структурного синтеза логических схем состоит из следующих основных этапов:

1. Формализация словесного задания схемы.
2. Составление таблицы истинности.
3. Запись логической функции в СДНФ или СКНФ.
4. Выбор функционально полной системы логических элементов.
5. Минимизация логической функции.
6. Построение комбинационной схемы, соответствующей логической функции.
7. Проверка правильности работы комбинационной схемы.

Хотя при синтезе комбинационной схемы формализация словесного задания обычно не вызывает особых затруднений и сводится к выявлению наборов переменных, на которых функция равна 0 или 1, на занятии рекомендуется решить одну из задач с использованием первого этапа.

Задачи

1. По установленному сигналу каждый игрок **A** и **B** замыкает или размыкает выключатель, находящийся под его управлением. Если оба делают одно и то же, то выигрывает игрок **A**, в противном случае – выигрывает игрок **B**. Построить схему так, чтобы в случае выигрыша игрока **A** зажглась лампочка.

2. Имеется одна лампочка в лестничном пролете двухэтажного дома. Построить схему так, чтобы на каждом этаже своим выключателем можно было гасить и зажигать лампочку независимо от положения другого выключателя.

Для данных булевых функций построить соответствующие им комбинационные схемы в избыточном функционально полном наборе $\{\&, \vee, \bar{}\}$, а также в базисах стрелка Пирса $\{\downarrow\}$и штрих Шеффера $\{|\}$.

11 ИССЛЕДОВАНИЕ КОДОПРЕОБРАЗОВАТЕЛЕЙ

Цель работы

Закрепить знания, полученные на лекции, посвященной типовым комбинационным устройствам и исследовать работу преобразователей кодов на основе их компьютерных моделей.

Методические указания по выполнению практического занятия

Рассмотрим преобразование двоичного кода в код Грея, у которого переход к соседнему числу сопровождается изменением только в одном разряде. Так, в технике аналого-цифрового преобразования и пересчетных устройствах широко используется код Грея. Он позволяет существенно сократить время преобразования и повысить эффективность защиты от нежелательных сбоев при переходах выходного кода. Недостатком кода Грея является то, что в нем затруднено выполнение арифметических операций и цифро-аналоговое преобразование. Поэтому при необходимости код Грея преобразуется в обычный двоичный код. Переход от двоичного кода к коду Грея осуществляется следующим образом: старшие разряды совпадают, а любой следующий разряд Y_k кода Грея равен сумме по модулю два соответствующего $X_к$ и предыдущего $X_{к+1}$ разрядов двоичного кода, т.е. $Y_k = X_k + X_{k-1}$. При обратном переходе старшие разряды также совпадают, но каждый следующий разряд получается в результате суммирования по модулю два полученного разряда двоичного кода и соответствующего разряда кода Грея, т.е. $X_{к-1} = Y_{k-1} + X_к$.

Эту процедуру можно также свести к последующему просмотру и преобразованию цифр кода Грея, начиная со старшего разряда, цифра остается без изменения, если число предшествующих единиц четно (нуль считается четным числом) и инвертируется, если число предшествующих единиц нечетно. Преобразование двоичного 4-х разрядного кода в код Грея приведено в таблице истинности (таблица 11.2).

Таблица 11.2 – Код Грея

Позиционный двоичный код					Код Грея				
N	X_1	X_2	X_3	X_4	N	Y_1	Y_2	Y_3	Y_4
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1
2	0	0	1	0	3	0	0	1	1
3	0	0	1	1	2	0	0	1	0
4	0	1	0	0	6	0	1	1	0
5	0	1	0	1	7	0	1	1	1
6	0	1	1	0	5	0	1	0	1
7	0	1	1	1	4	0	1	0	0
8	1	0	0	0	12	1	1	0	0
9	1	0	0	1	13	1	1	0	1

10	1	0	1	0	15	1	1	1	1
11	1	0	1	1	14	1	1	1	0
12	1	1	0	0	10	1	0	1	0
13	1	1	0	1	11	1	0	1	1
14	1	1	1	0	9	1	0	0	1
15	1	1	1	1	8	1	0	0	0

На рисунке 11.1 представлена схема преобразователя двоичного кода в код Грея на основе шифратора и дешифратора.

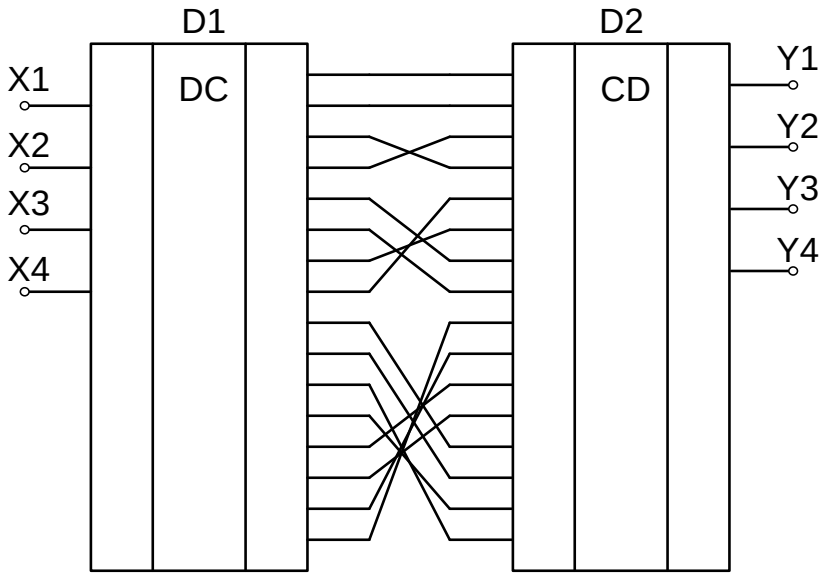


Рисунок 11.1 – Схема кодопреобразователя

На рисунке 11.2 приведена компьютерная модель преобразователя двоичного кода в код Грея на основе шифратора и дешифратора.

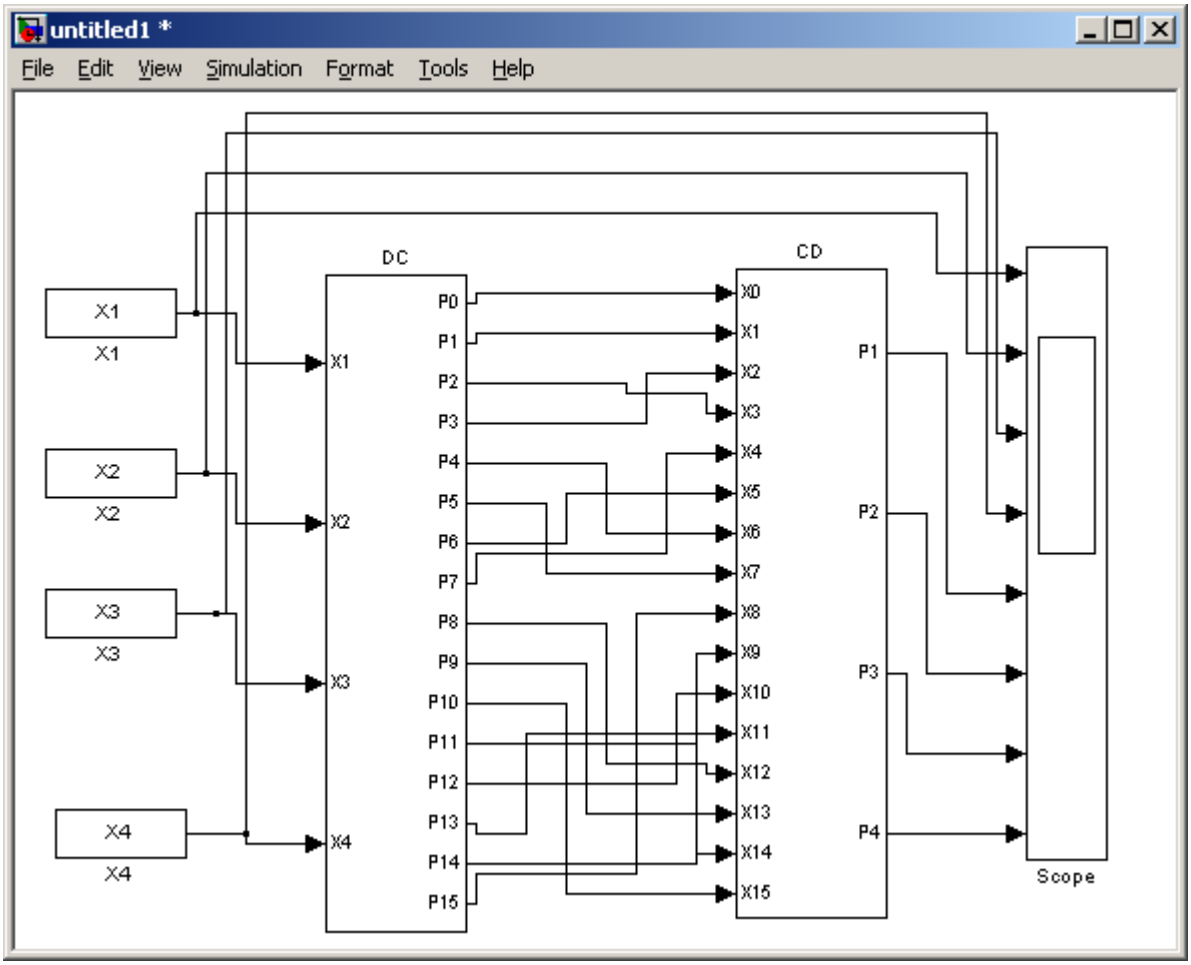


Рисунок 11.2 – Компьютерная модель кодопреобразователя

На рисунке 11.3 приведены временные диаграммы функционирования кодопреобразователя. Анализ временных диаграмм позволяет сделать вывод о том, что кодопреобразователь функционирует правильно.

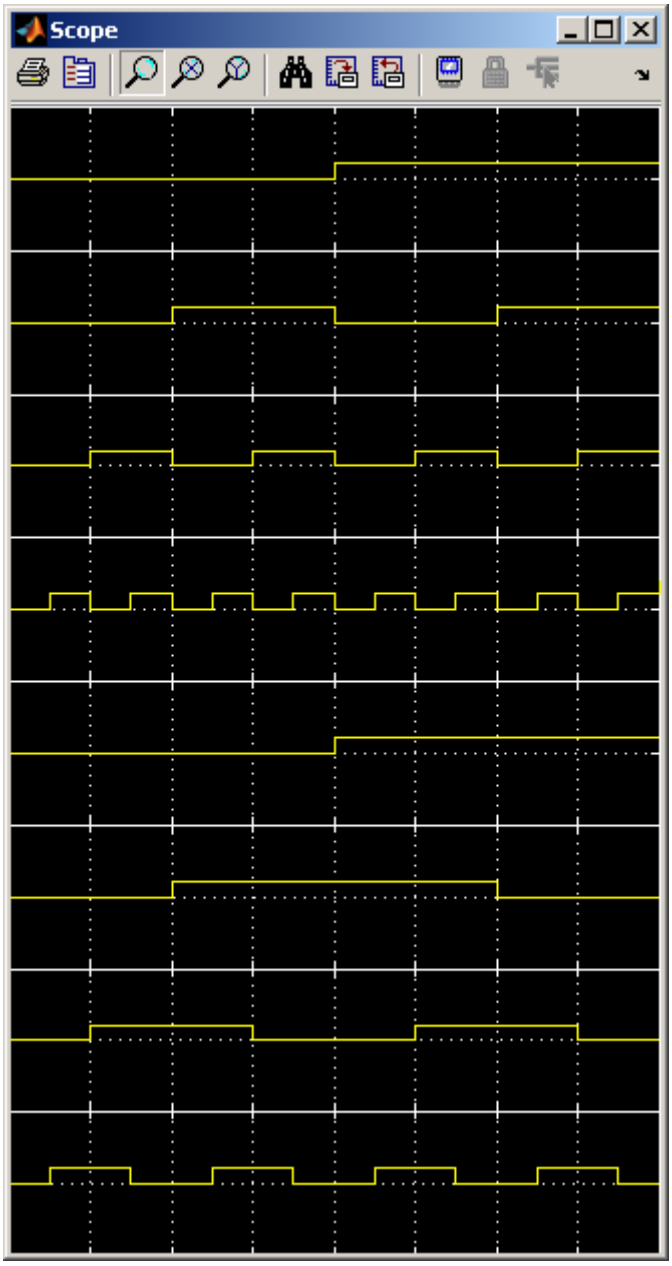


Рисунок 11.3 – Временные диаграммы функционирования кодопреобразователя.

Задание дляпрактического занятия:

Создать компьютерную модель преобразователей кодов (согласно таблице 1) и исследовать его работу.

Таблица 11.1 – Варианты заданий на практическое занятие

Вариант / X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	7	1	5	9	1	0	4	1	7	7	13	11	0	6	7
1	1	8	2	6	10	2	1	5	2	8	8	14	12	1	7	8
2	2	9	0	7	0	0	2	6	0	9	9	15	13	2	8	9
3	3	10	6	8	1	6	3	7	6	10	10	7	14	3	9	10
4	4	11	15	0	2	8	4	8	12	11	11	8	0	4	10	0
5	5	12	7	1	11	9	5	9	13	12	0	1	1	7	0	1
6	6	13	8	2	3	3	6	12	9	0	1	2	3	8	1	5
7	9	0	9	3	4	12	7	13	3	1	2	0	4	9	2	6
8	10	1	3	4	5	13	8	14	4	2	3	4	5	10	3	11
9	11	2	4	9	6	14	9	15	5	3	6	5	6	11	4	12
10	12	3	12	10	7	15	10	0	10	4	12	10	2	12	5	13
11	13	4	13	11	8	7	11	1	11	5	13	11	9	13	11	14
12	14	5	14	12	12	4	12	2	14	6	14	6	10	14	12	2
13	15	6	5	13	13	5	13	3	15	13	4	12	15	15	13	3
14	7	14	10	14	14	10	14	10	7	14	5	9	7	5	14	4
15	8	15	11	15	15	11	15	11	8	15	15	3	8	6	15	15

Содержание отчета по практическому занятию:

1. Задание на практическое занятие.
2. Схема кодопреобразователя.
3. Компьютерная модель исследуемого устройства и временные диаграммы его функционирования.

Контрольные вопросы:

1. Приведите условно-графическое обозначение кодопреобразователя.
2. Дайте определение преобразователя кодов.
3. В чем сложность выполнения арифметических операций в коде Грея?

12 ИССЛЕДОВАНИЕ ШИФРАТОРОВ

Цель работы

Закрепить знания, полученные на лекциях, посвященной типовым комбинационным устройствам и исследовать работу шифраторов.

Методические указания по выполнению практического занятия

Шифраторы (кодером) называется логическое устройство, преобразующее m - разрядный унитарный код в n-разрядный двоичный позиционный код.

Унитарный код - код. в каждом наборе которого единичное значение принимает лишь один из разрядов.

Между количеством входных (m) и выходных (n) шин существует следующая связь:

$$m < 2^n$$

Если $m = 2^n$, то такой шифратор называется полным.

В цифровой технике шифраторы применяются как преобразователи некоторой позиции в соответствующий ей код.

1) в преобразователях кодов;

2) устройствах телеобработки при посылке различных символов в линии связи;

3) во входных устройствах ЭВТ и т.д.

УГО шифратора 1 на функциональных электрических схема приведено на рис.12.1.

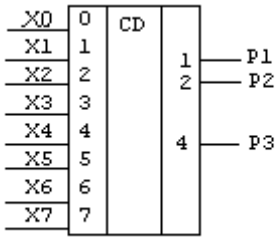


Рисунок 12.1 – Условное графическое обозначение шифратора

CD – кодер (шифратор) – обозначение функции, реализуемой элементом. Пример маркировки шифратора.

К 120 И В 1 – полный шифратор на 8 входов. Или например,

К 165 И В 1 – шифратор приоритета, в котором трехразрядный двоичные позиционный код на выходе ИМС показывает номер самого старшего разрядного входного кода, в котором имеется единица.

На УГО в дополнительном поле слева входы помечают десятичными обозначениями кодовых комбинаций, соответствующих им. В дополнитель-

ном поле справа выходы помечают десятичными числами, указывающими двоичные веса позиции выходного кода.

Рассмотрим принцип построения полного шифратора на 4 входа.

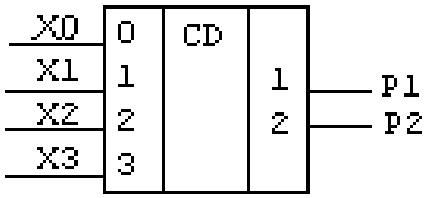


Рисунок 12.2 – Условное графическое обозначение шифратора на 4 входа

Составим таблицу истинности (таблица 12.3), описывающую закон функционирования интезируемого шифратора.

Таблица 12.3.

№ наб.	X0	X1	X2	X3	P2	P3
8	1	0	0	0	0	0
4	0	1	0	0	0	1
2	0	0	1	0	1	0
1	0	0	0	1	1	1

Столбец N наборов наполняется последним.

Из таблицы видно, что выходные сигналы P1 и P2 описываются полностью определенными логическими функциями. Составим карты и произведем минимизацию функций P1 и P2.

МДНФ $P1 = X1 + X3$ МДНФ $P2 = X2 + X3$

Синтезируемый шифратор, таким образом, можно реализовать на двухвходовых элементах ИЛИ (рис. 12.4).

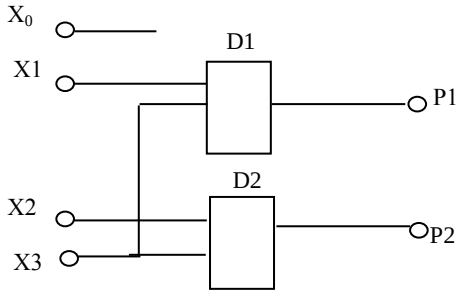


Рисунок 12.4 - Схема шифратора на 4 входа в основном базисе

Таким образом, выходные сигналы могут формироваться как дизъюнкции входных сигналов, принимающих значение "1" в строках таблицы, в которых равны единице входные сигналы.

Это положение распространяется на шифраторы с любым числом вхо-

дов, так для полного шифратора на восемь входов таблица истинности приведена в таблице 11.8. МДНФ функций будут следующими:

МДНФ P1 = XI + X3 + X5 + X7

МДНФ P2 = X2 + X3 + X5 + X7

МЛНФ P3 = X4 + X3 + X5 + X7

Таблица 12.5

X0	X1	X2	X3	X4	X5	X6	X7	P3	P2	P1
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Необходимо отметить, что кодирование позиции Xo осуществляется тривиально: она не соединяется ни с одной из входных шин. Ее код будет существовать на выходе не только при возбуждении шины Xo, но и в состоянии покоя, и при выключенной схеме, и во многих случаях даже при неисправности шифратора. Поэтому нулевой выходной код (00 ...0) и соответствующую ему позицию (X_0) часто не используют.

Рассмотрим пример исследования дешифратора на 4 выхода. Структурная схема дешифратора приведена на рисунке 12.6.

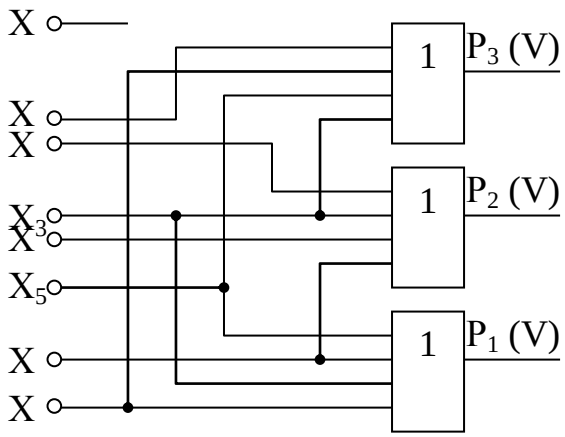


Рисунок 12.6 − Структурная схема шифратора

Компьютерная модель шифратора библиотека элементов: шифратора и дешифратора в программе Simulink приведена на рисунке 12.7.

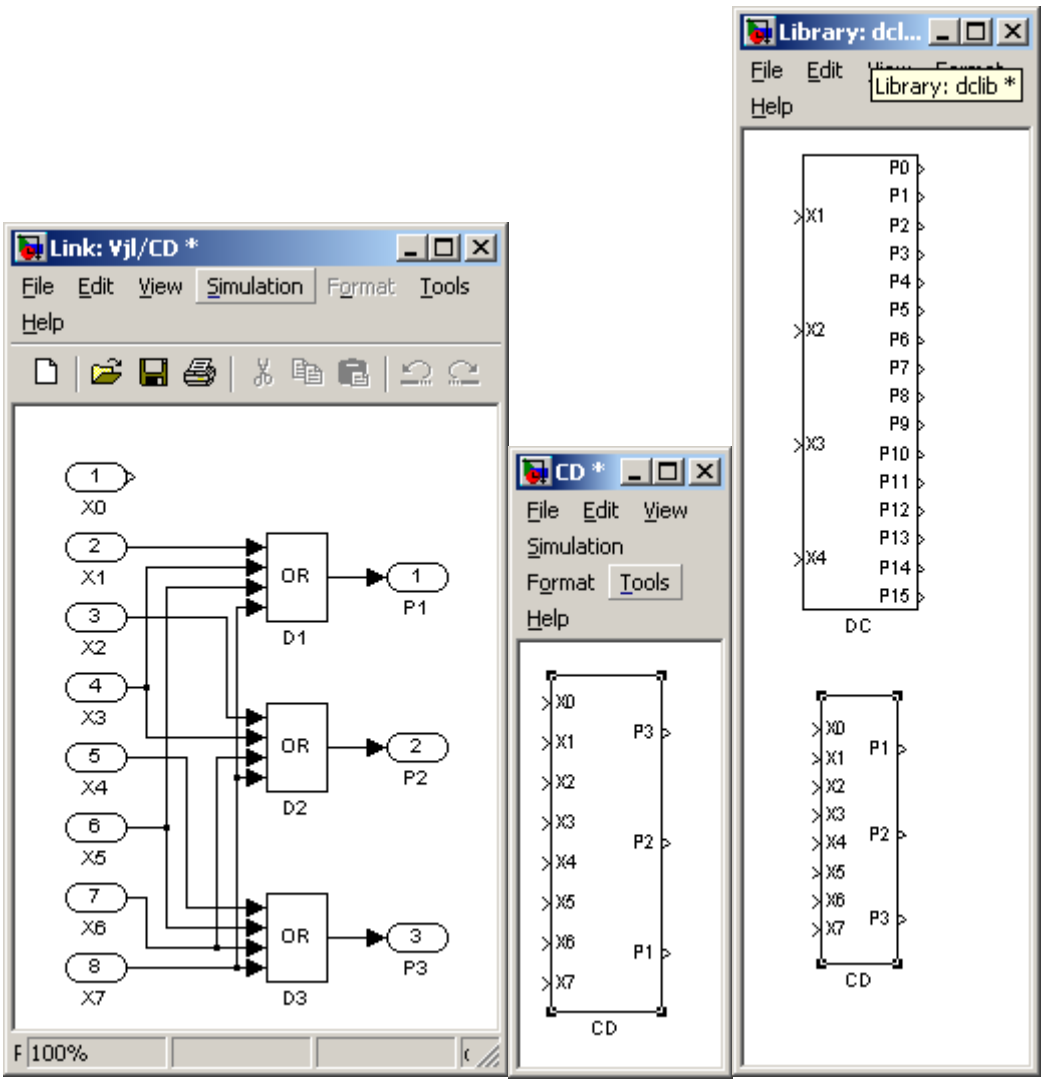


Рисунок 12.7 − Компьютерная модель шифратора

На рисунке 12.8 представлена схема исследования шифратора в динамическом режиме.

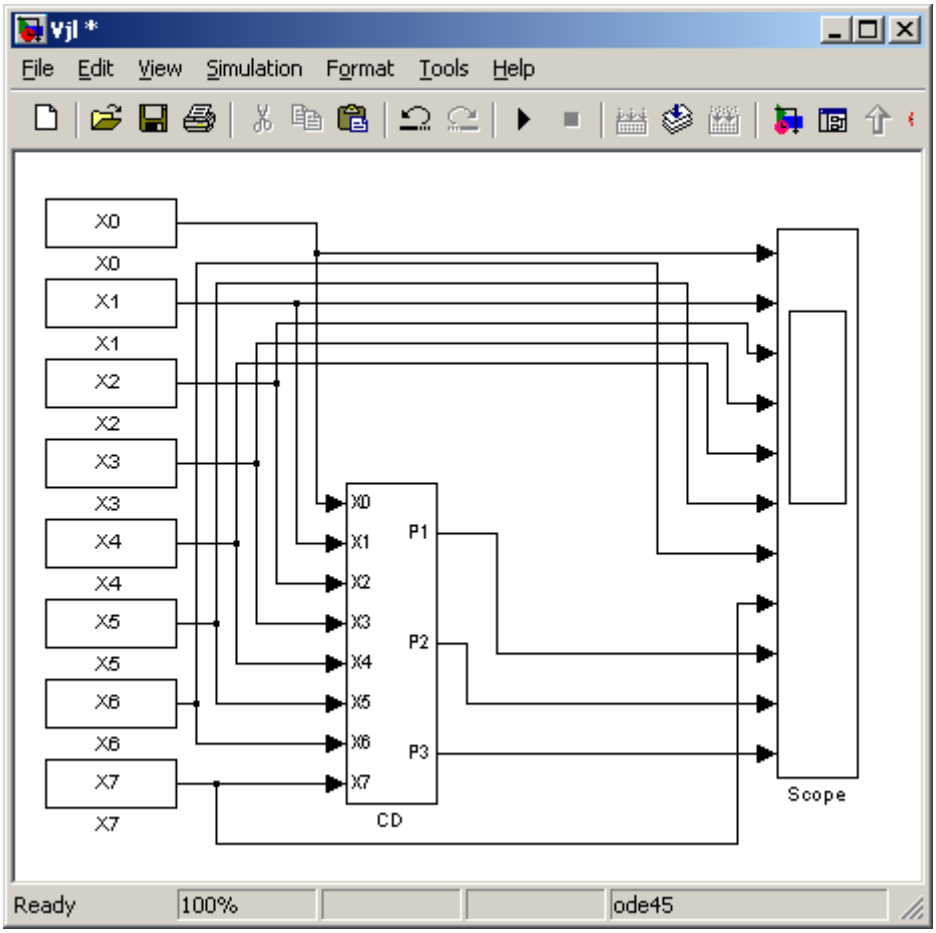


Рисунок 12.8 − Схема исследования шифратора в динамическом режиме

Временные диаграммы, представленные на рисунке 10, позволяют сделать вывод о правильности функционирования шифратора.

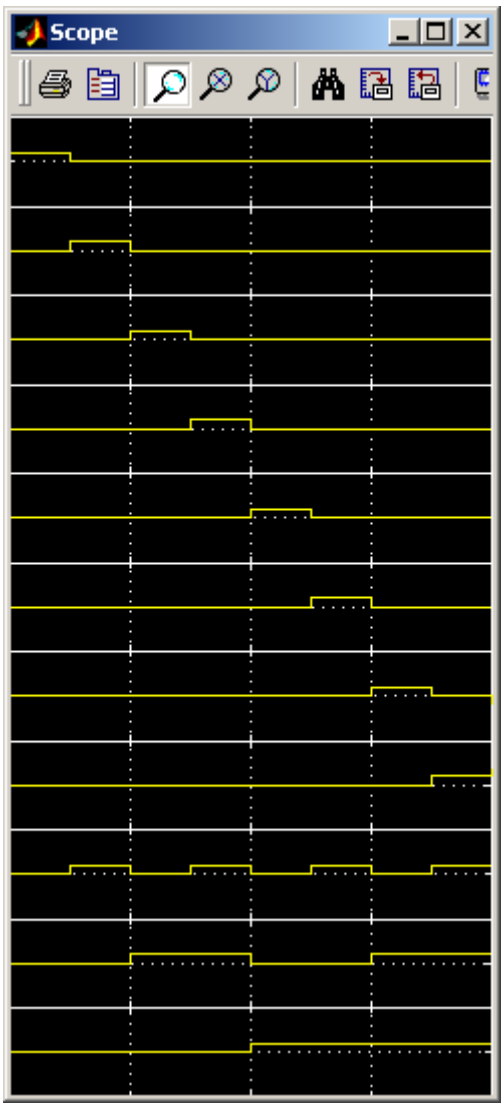


Рисунок 12.8 − Временные диаграммы функционирования шифратора

Задание дляпрактического занятия:

Создать компьютерную модель шифратора на 4-е выхода и исследовать их работу.

Содержание отчета по практическому занятию:

1. Задание на практическое занятие.
2. Схемы шифратора.
3. Компьютерные модели исследуемых устройств и временные диаграммы их функционирования.

13 ИССЛЕДОВАНИЕ ДЕШИФРАТОРОВ

Цель работы

Закрепить знания, полученные на лекциях, посвященной типовым комбинационным устройствам и исследовать работу дешифраторов.

Методические рекомендации по выполнению практического занятия

Схема матричного дешифратора на 4-е входа приведена на рисунке 13.1.

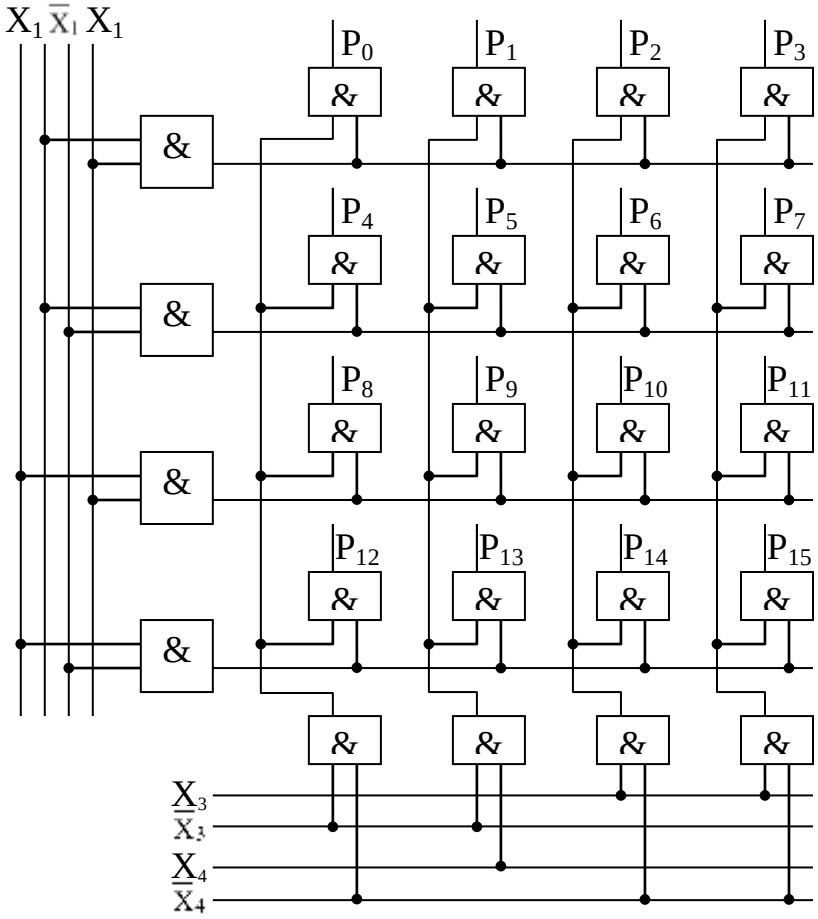


Рисунок 13.1 - Схема матричного дешифратора

Компьютерная модель исследования матричного дешифратора в статическом режиме приведена на рисунке 2. На рисунке приведен пример исследования функционирования дешифратора на 0-м наборе, когда все входные переменные приминают значении 0, при этом единица появляется только на выходе P0.

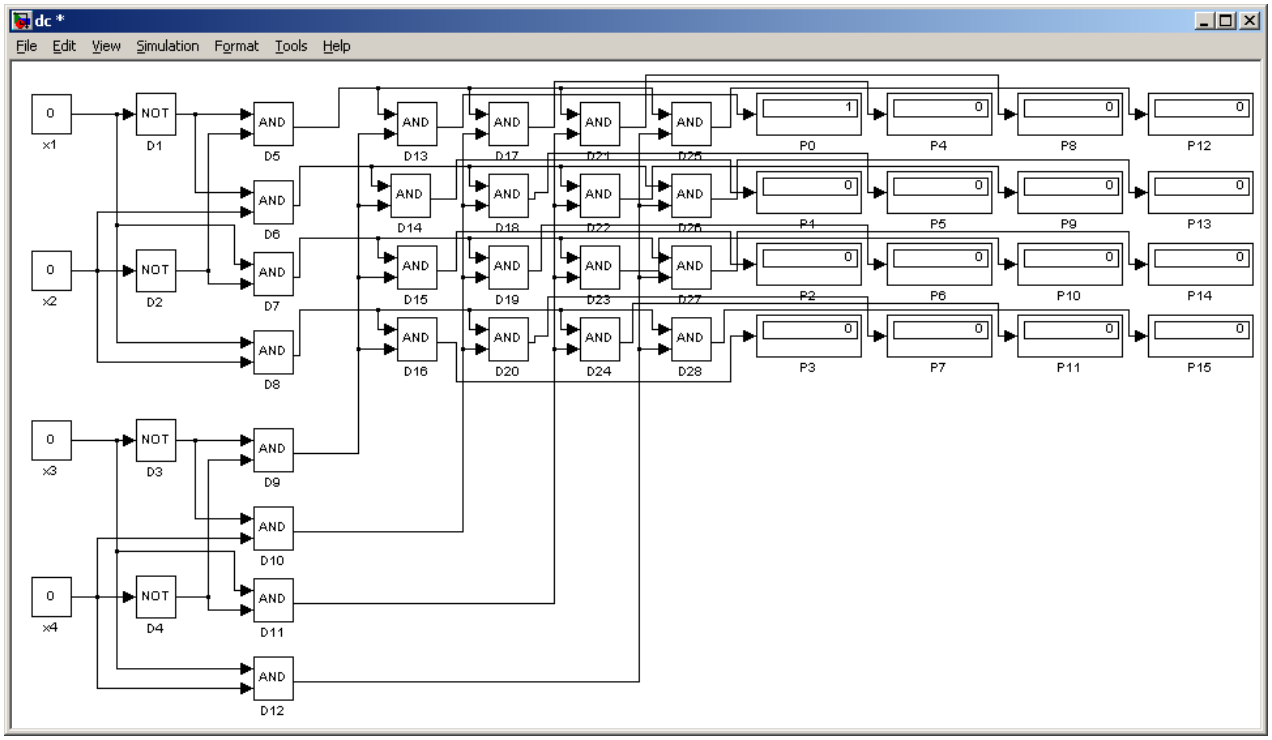


Рисунок 13.2 − Компьютерная модель исследования матричного дешифратора в статическом режиме

Для исследования 4-х входового дешифратора в динамическом режиме можно объединить блоки, входящие в модель, командой Crate Subsystem, добавив разработанную модель (рисунок 13.3) к новой библиотеке (рисунок 13.4).

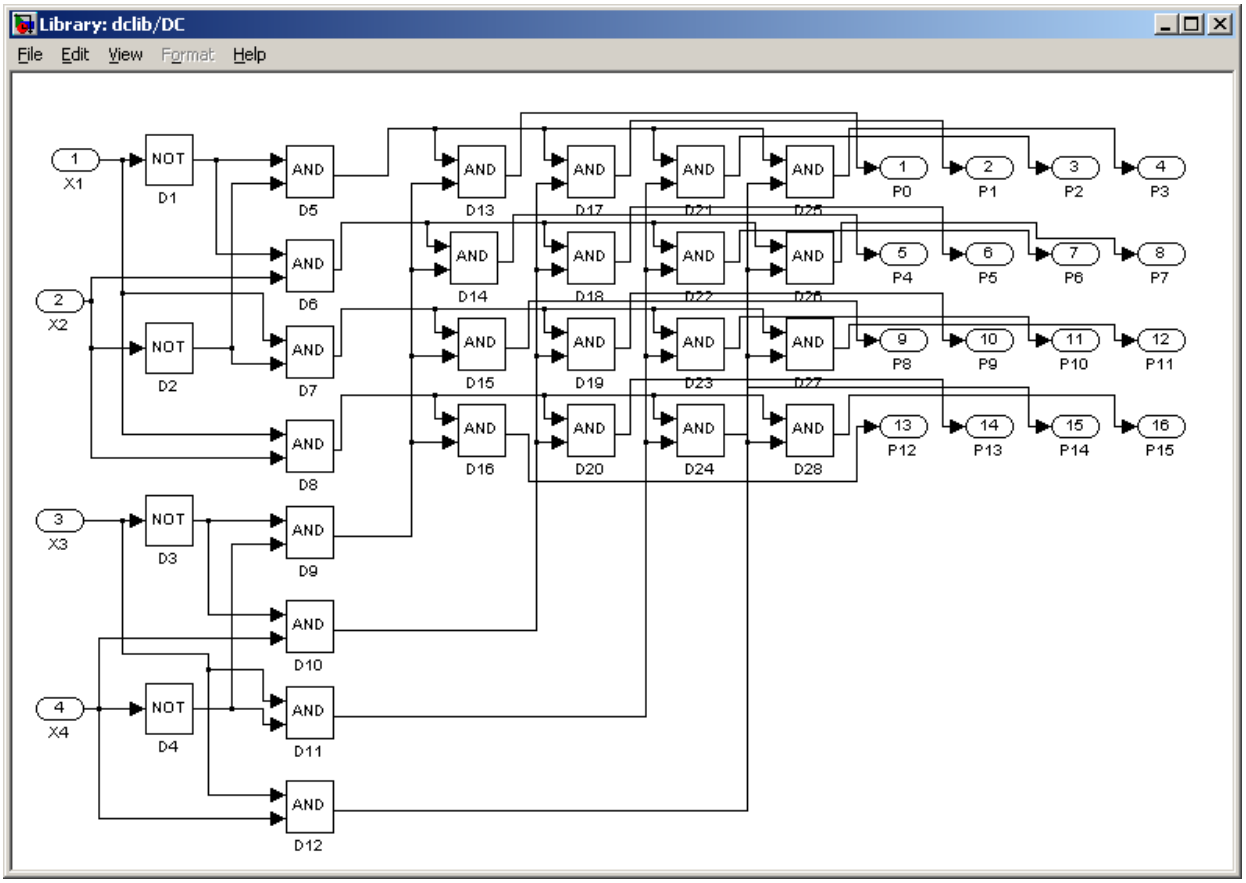


Рисунок 13.3 − Модель дешифратора

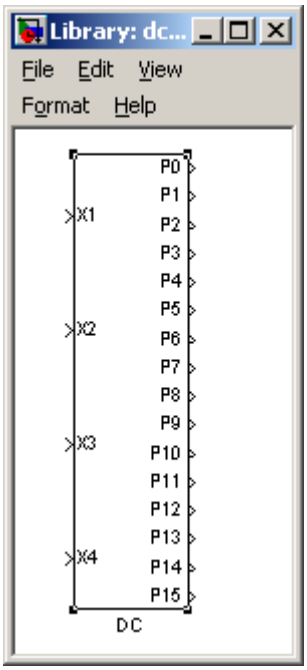


Рисунок 13.4 − Библиотека с дешифратором

Компьютерная модель исследования дешифратора в динамическом режиме (рисунок 5) кроме дешифратора должна содержать элементы (From File), задающие входные переменные и регистрирующий элемент − осциллограф (Scope). Порядок добавления описанных элементов к компьютерной модели и задания их параметров детально описан в предыдущей практиче-

скому занятию.

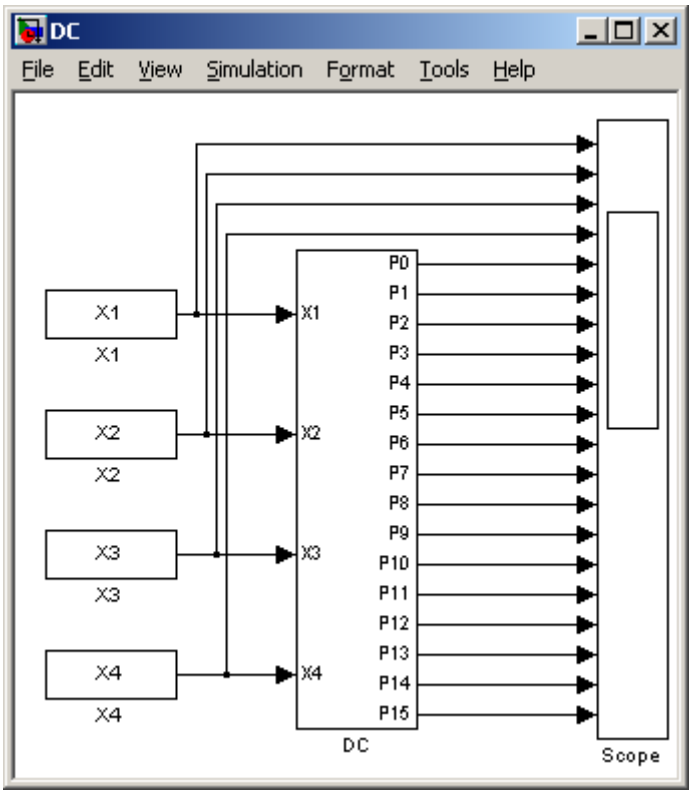


Рисунок 13.5 − Компьютерная модель исследования дешифратора в динамическом режиме

Временные диаграммы осциллографа приведены на рисунке 13.6. Анализ временных диаграмм позволяет сделать вывод о правильности функционирования компьютерной модели.

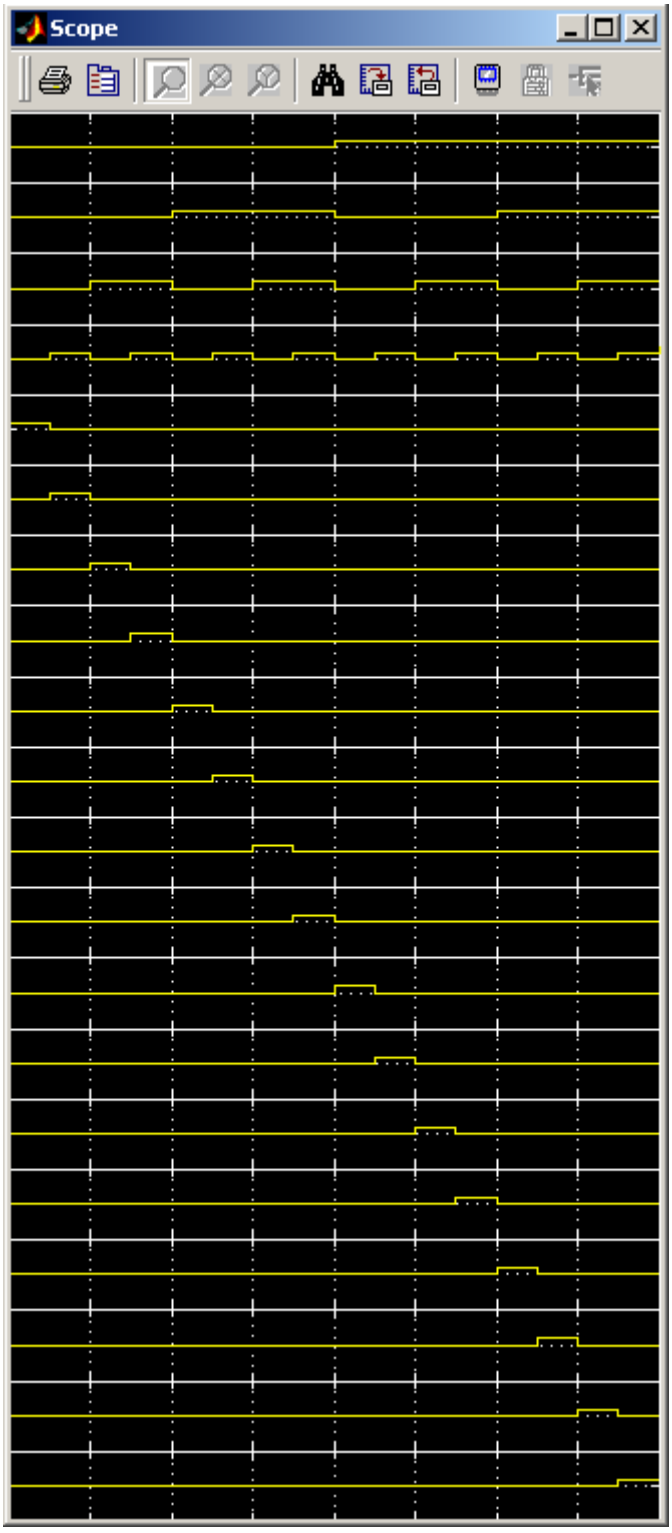


Рисунок 13.6 − Временные диаграммы осциллографа

Задание дляпрактического занятия:

Создать компьютерную модель дешифратора на 4-е входа (согласно таблице 1), шифратора на 4-е выхода и исследовать их работу.

Таблица 13.1 − Варианты заданий на синтез дешифратора

Вариант	Тип дешифратора
1	Линейный
2	Пирамидальный
3	Матричный

Содержание отчета по практическому занятию:

1. Задание на практическое занятие.
2. Схемы дешифратора.
3. Компьютерные модели исследуемых устройств и временные диаграммы их функционирования.

14 ИССЛЕДОВАНИЕ МУЛЬТИПЛЕКСОРОВ

Цель работы

Закрепить знания, полученные на лекции, посвященной типовым комбинационным устройствам и исследовать функционирование мультиплексоров на основе их компьютерных моделей.

Методические рекомендации по выполнению практического занятия

Мультиплексором является цифровое комбинационное устройство, коммутирующее один из информационных входов на выход в зависимости от комбинации сигналов на управляющих входах.

Схема демультиплексора с двумя управляющими и 4-мя информационными входами приведена на рисунке 14.1.

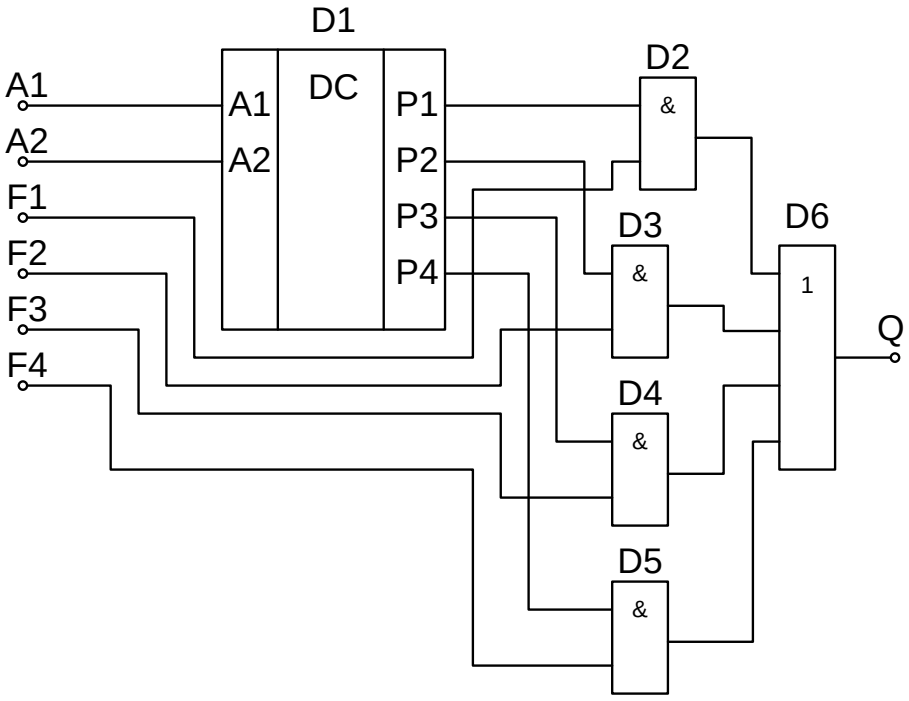


Рисунок 14.1 − Схема мультиплексора

На приведенной схеме основным блоком является дешифратор на два входа. Схема дешифратора на два входа приведена на рисунке 14.2.

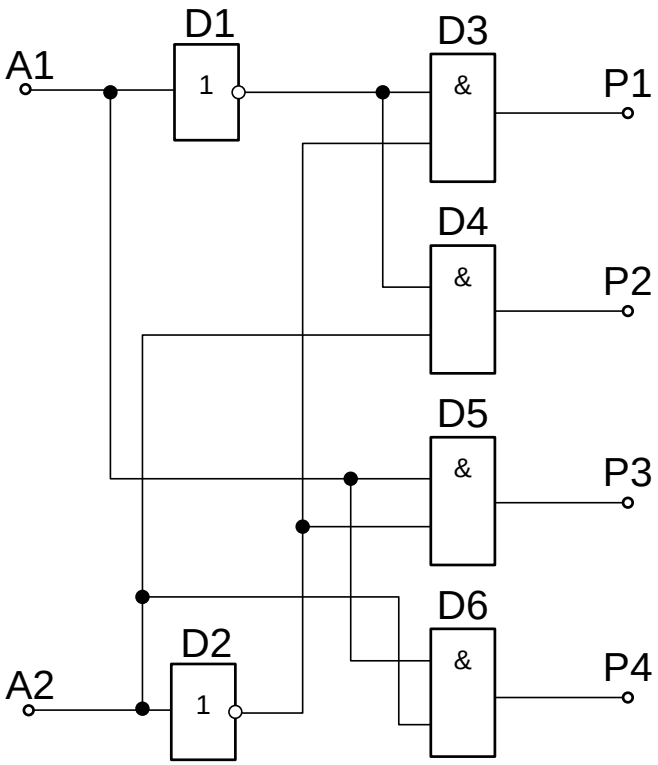


Рисунок 14.2 – Схема дешифратора на два входа

Для создания модели мультиплексора необходимо создать компьютерную модель дешифратора на два входа. Модель дешифратора в программе Simulink приведена на рисунке 14.3.

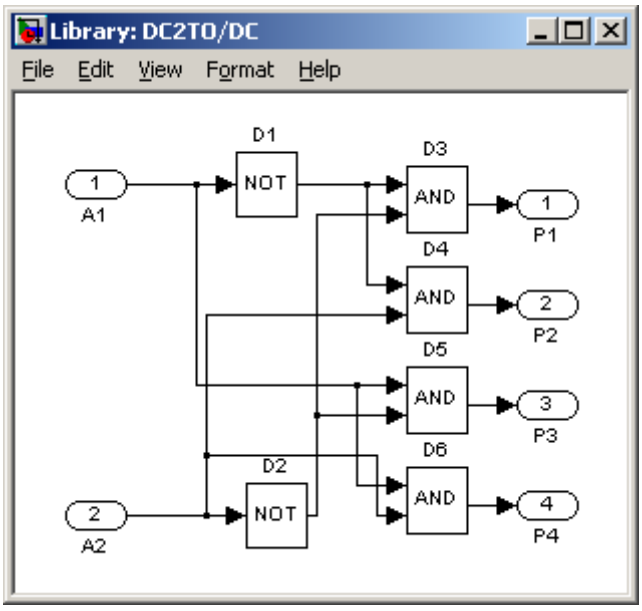


Рисунок 14.3 − Модель дешифратора

Для создания модели дешифратора отдельным блоком необходимо выделить все блоки на модели рисунка 3 и применить команду Create Subsystem контекстного меню. Модель дешифратора и содержащая его библиотека приведены на рисунке 14.4.

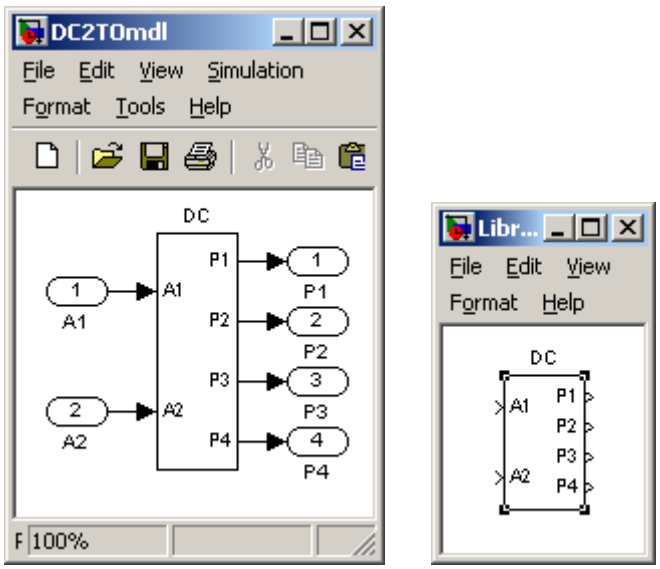


Рисунок 14.4 – Библиотека дешифратора и его модель

Для проверки правильности функционирования разработанного блока необходимо собрать схему, представленную на рисунке 14.5.

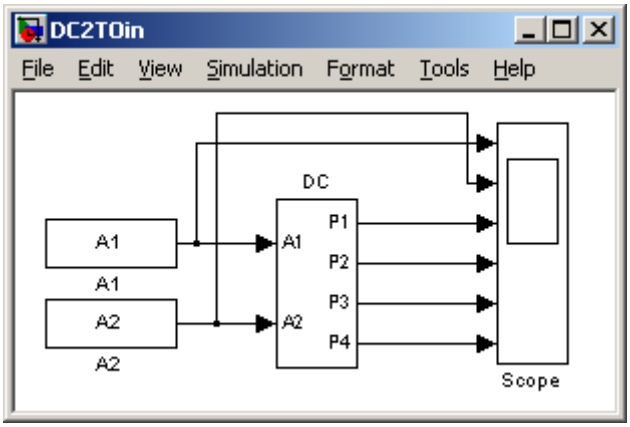


Рисунок 14.5 – Схема исследования дешифратора

Временные диаграммы, полученные при запуске модели, приведены на рисунке 14.6.

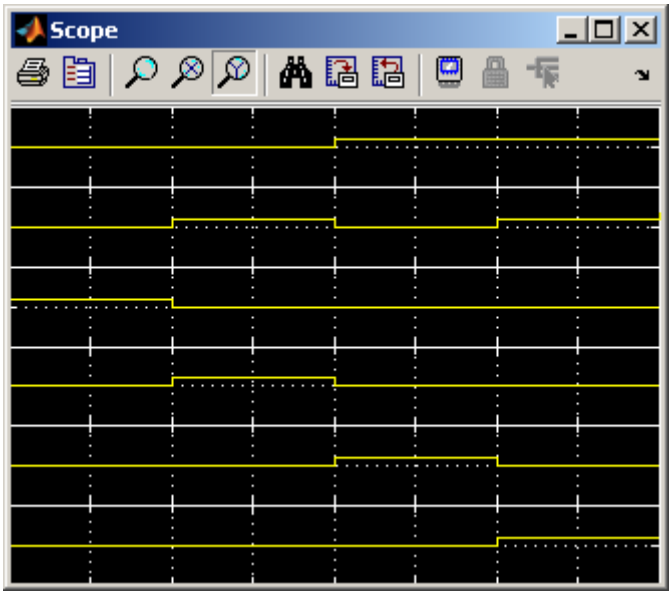


Рисунок 14.6 – Временные диаграммы функционирования дешифратора

Для создания модели мультиплексора, приведенной на рисунке 14.1, необходимо использовать блок дешифратора и блоки логических операций. Модель мультиплексора в программе Simulink представлена на рисунке 14.7.

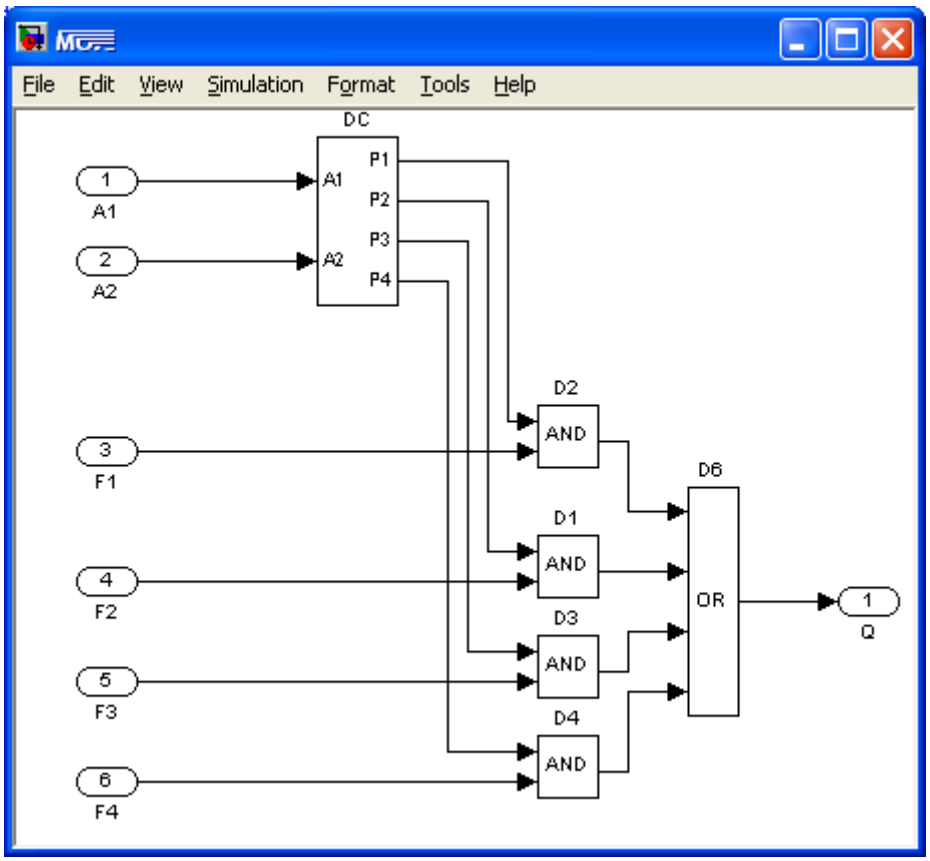


Рисунок 14.7 – Модель мультиплексора

Для создания модели мультиплексора в виде отдельного блока необходимо выделить все блоки на модели рисунка 6 и применить команду Create System контекстного меню. Модель мультиплексора и библиотека, содержа-

щая мультиплексор, приведены на рисунке 14.8.

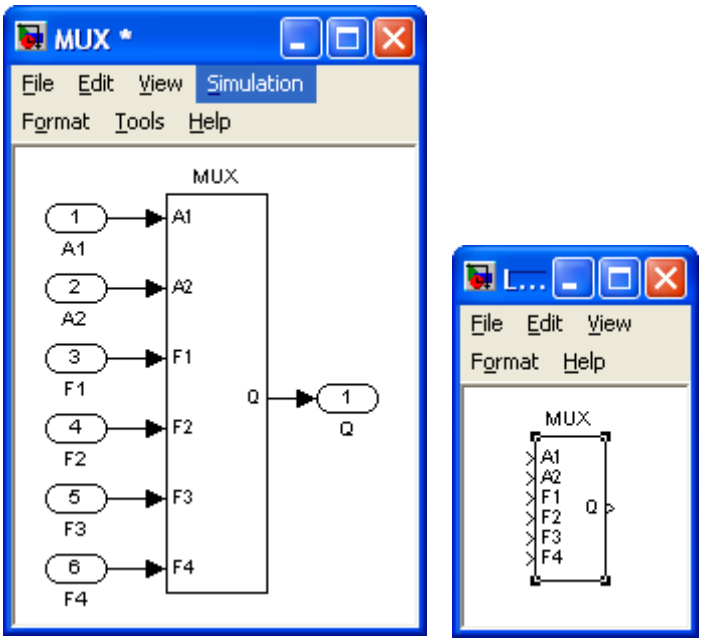


Рисунок 14.8 – Блок демультиплекосра и его библиотека

Для исследования функционирования мультиплексора в динамическом режиме необходимо собрать модель, представленную на рисунке 14.9.

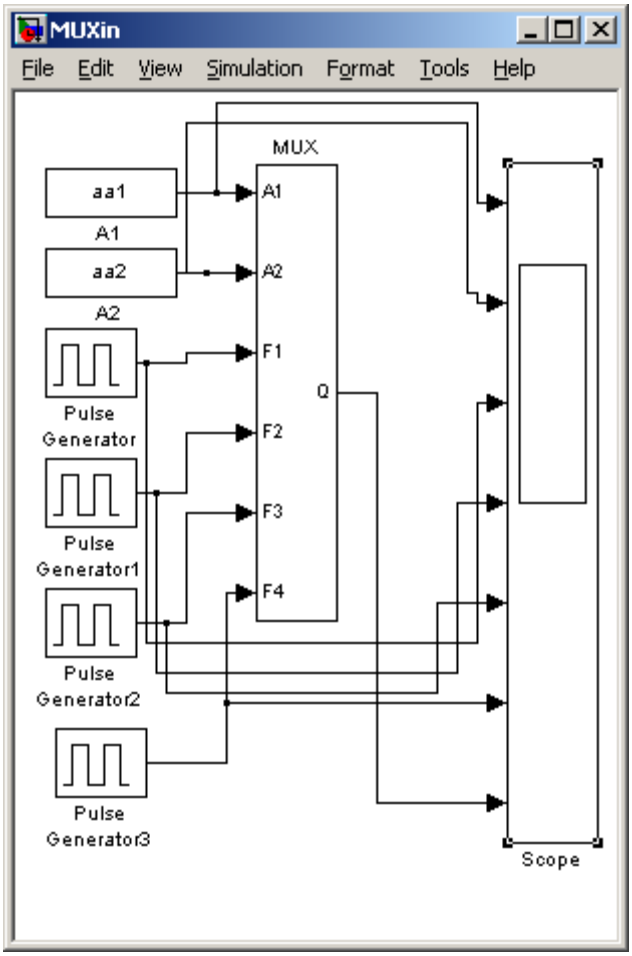


Рисунок 14.9 – Модель исследования мультиплексора в динамическом режиме

На управляющие входы мультиплексора поданы сигналы блоков From File, определяемые созданными переменными, а на информационные сигналы блоков Pulse Generator с различным периодом импульсов.

Временные диаграммы функционирования мультиплексора, полученные в результате запуска схемы, представлены на рисунке 14.10.

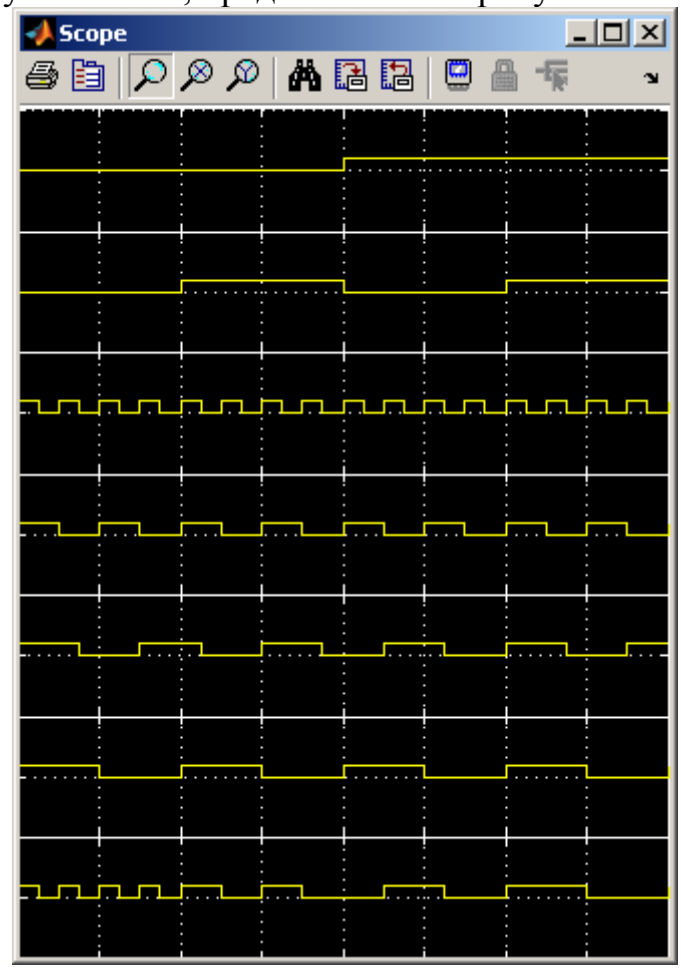


Рисунок 14.10 – Временные диаграммы функционирования мультиплексора

Анализ полученных временных диаграмм позволяет сделать вывод о правильности функционирования мультиплексора. На рисунке 9 временными диаграмма являются (сверху вниз):

1 – управляющий вход A1;

2 – управляющий вход A2;

3, 4, 5, 6 – сигналы генераторов с различным периодом импульсов;

7 – выход мультиплексора.

Задание на практическое занятие:

Создать компьютерные модели мультиплексоров и демультиплексоров и исследовать их работу.

Содержание отчета по практическому занятию:

1. Задание на практическое занятие.
2. Схема и мультиплексора.
3. Компьютерные модели исследуемых устройств и временные диаграммы их функционирования.

Контрольные вопросы:

1. Приведите условно-графическое обозначение мультиплексора.
2. Дайте определение мультиплексора.
3. Каким соотношением связаны между собой количество информационных и управляющих входов мультиплексора?

15 ИССЛЕДОВАНИЕ ДЕМУЛЬТПЛЕКСОРОВ

Цель работы: закрепить знания, полученные на лекции, посвященной типовым комбинационным устройствам и исследовать функционирование мультиплексоров и демультиплексоров на основе их компьютерных моделей.

Методические рекомендации по выполнению практического занятия

Демультиплексором является цифровое комбинационное устройство, коммутирующее информационный вход на один из выходов в зависимости от комбинации сигналов на управляющих входах.

Схема демультиплексора с двумя управляющими входами приведена на рисунке 15.1.

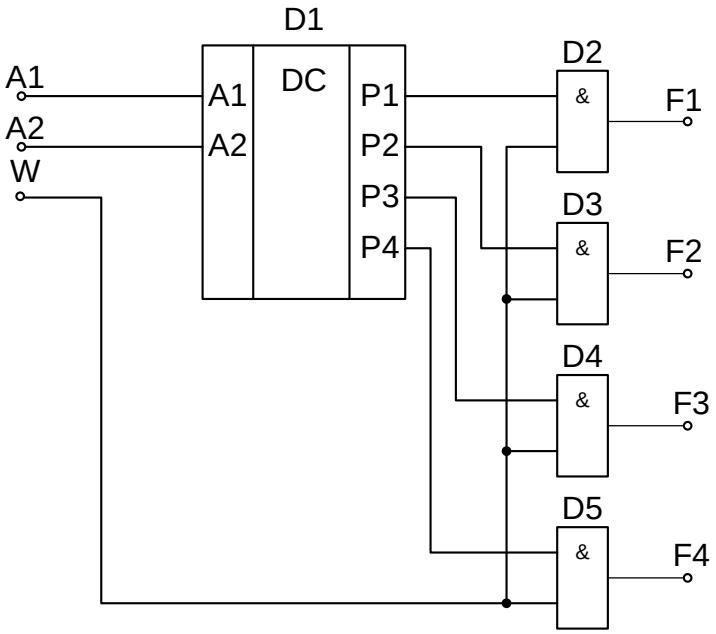


Рисунок 15.1 – Схема демультиплексора

Для создания модели демультиплексора, приведенной на рисунке 15.1, необходимо использовать блок дешифратора и блоки логических операций. Модель демультиплексора в программе Simulink представлена на рисунке 15.2.

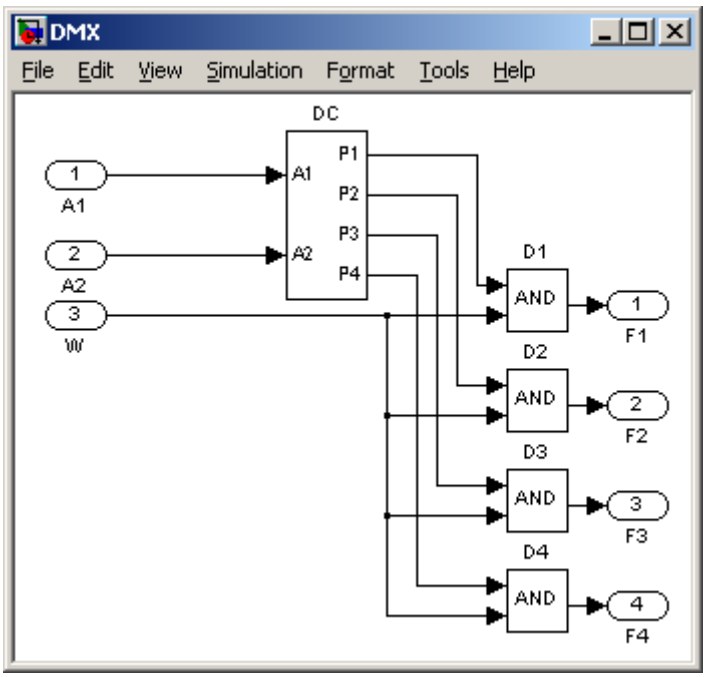


Рисунок 15.2 – Модель демультиплексора

Для создания модели демультиплексора в виде отдельного блока необходимо выделить все блоки на модели рисунка 15.2 и применить команду Create System контекстного меню. Модель мультиплексора и библиотека, содержащая демультиплексор, приведены на рисунке 15.3.

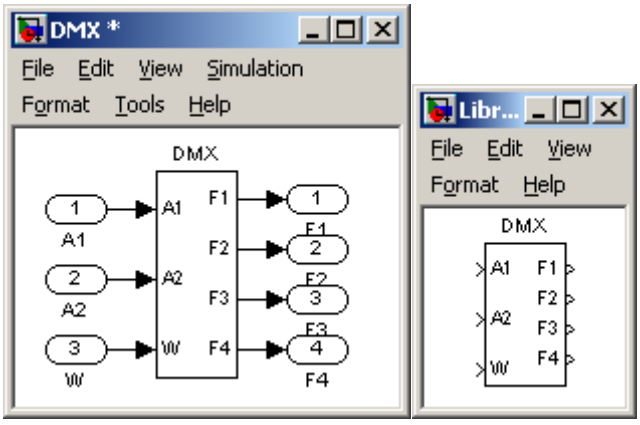


Рисунок 13 – Блок демультиплексора и его библиотека

Для исследования функционирования демультиплексора в динамическом режиме необходимо собрать модель, представленную на рисунке 15.4.

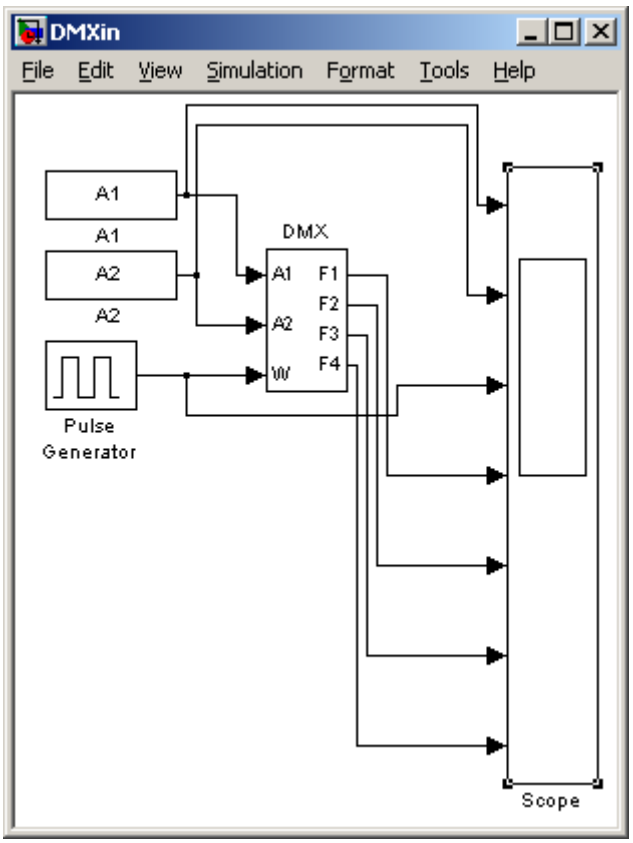


Рисунок 15.4 – Модель исследования демультиплексора в динамическом режиме

На управляющие входы демультиплексора поданы сигналы блоков From File, определяемые созданными переменными, а на информационный сигналы блока Pulse Generator.

Временные диаграммы функционирования демультиплексора, полученные в результате запуска схемы представлены на рисунке 15.5.

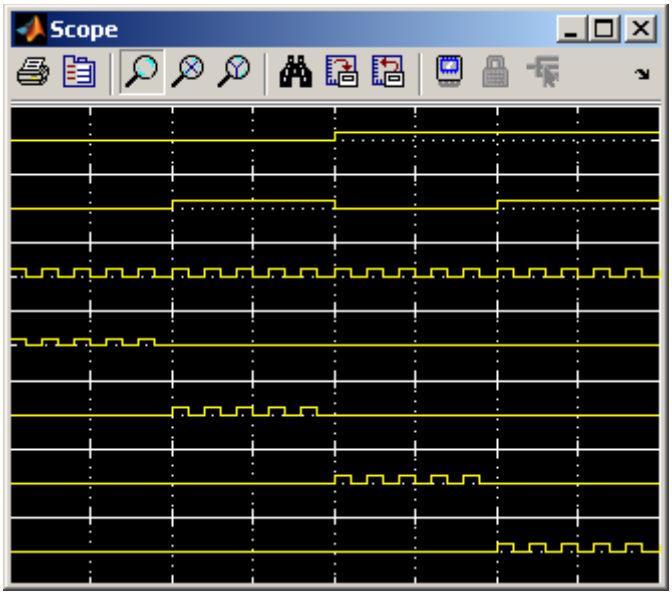


Рисунок 15.5 – Временные диаграммы функционирования демультиплексора

Анализ полученных временных диаграмм позволяет сделать вывод о правильности функционирования демультиплексора. На рисунке 9 временными диаграмма являются (сверху вниз):

1 – управляющий вход A1;
2 – управляющий вход A2;
3 – сигнал генератора импульсов;
4, 5, 6, 7 – выходы демультиплексора.

Задание на практическое занятие:

Создать компьютерные модели мультиплексоров и демультиплексоров и исследовать их работу.

Содержание отчета по практическому занятию:

1. Задание на практическое занятие.
2. Схема демультиплексора.
3. Компьютерные модели исследуемых устройств и временные диаграммы их функционирования.

Контрольные вопросы:

1. Приведите условно-графическое обозначение демультиплексора.
2. Дайте определение демультиплексора.
3. Каким соотношением связаны между собой количество информационных выходов и управляющих входов демультиплексора?

16 СИНТЕЗ СУММАТОРОВ И СХЕМ СРАВНЕНИЯ

Цель работы: закрепить знания, полученные на лекции, посвященной типовым комбинационным устройствам и исследовать функционирование сумматоров на основе их компьютерных моделей.

Программа работы:

Исследование сумматоров сложения чисел, представленных последовательным кодом.

Исследование сумматоров сложения чисел, представленных в параллельном коде.

Задание для практического занятия:

Создать компьютерную модель сумматора и на ее основе составить схемы 4-х разрядных сумматоров, предназначенных для сложения двух аргументов X и Y (согласно таблице 16.1), в последовательном и параллельном коде.

Таблица 16.1 − Варианты заданий на исследование сумматоров

Вариант	Аргумент X	Аргумент Y
1.	1101	1001
2.	1001	1011
3.	1111	1001
4.	1101	1001
5.	1001	1011
6.	1111	1001
7.	1001	1101
8.	1011	1001
9.	1001	1111
10.	1001	1101
11.	1010	1001
12.	1001	1111
13.	1001	1101
14.	1011	1001
15.	1001	1111
16.	1000	1101
17.	1011	1001
18.	1001	1111
19.	1101	1000
20.	1001	1011
21.	1111	1001
22.	1101	1001
23.	1000	1010

24.	1111	1001
25.	1010	1001
26.	1000	1001
27.	1111	0111
28.	1001	1001

Содержание отчета по практическому занятию:

Задание на практическое занятие.

1. Схемы сумматоров для сложения чисел в последовательном и параллельном коде.
2. Компьютерные модели исследуемых устройств и временные диаграммы их функционирования.

Пример выполнения практического занятия:

Схема одноразрядного 3-х входового сумматора представлена на рисунке 16.1.

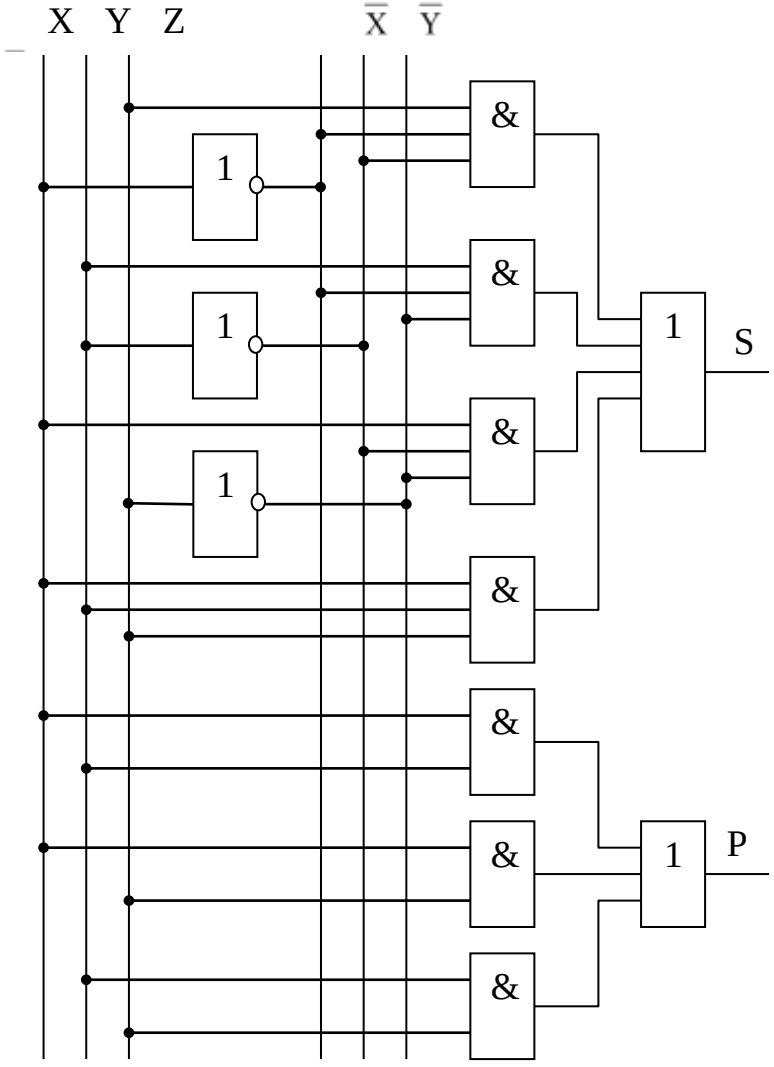


Рисунок 16.1 − Схема сумматора

На рисунке 16.2 представлена компьютерная модель одноразрядного

сумматора.

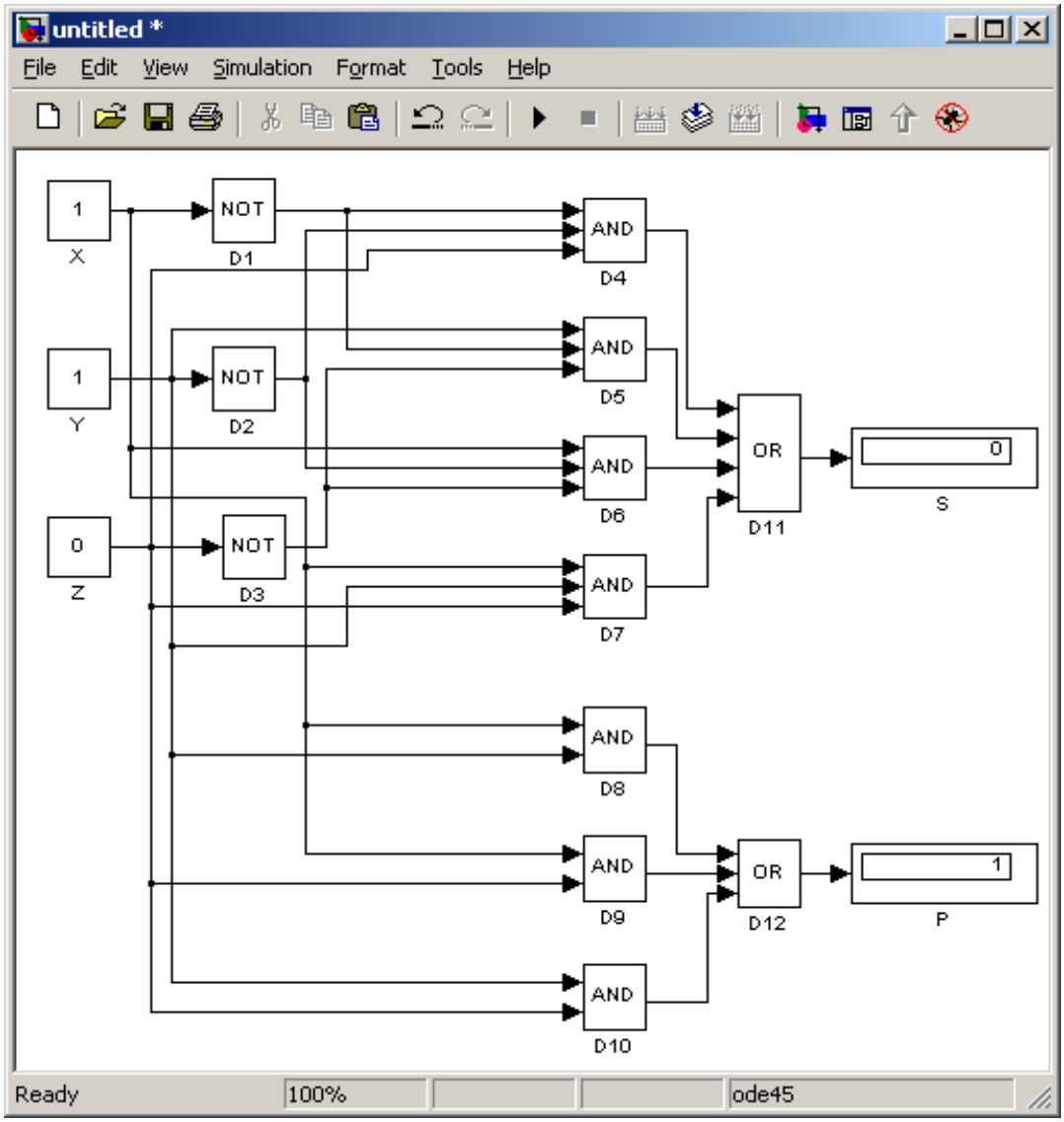


Рисунок 16.2 – Компьютерная модель одноразрядного сумматора

Для многократного использования сумматора необходимо создать библиотеку, содержащую одноразрядный сумматор. На рисунке 3 представлена модель сумматора на основе которой можно создать библиотеку. В этой модели блоки Constant заменены на In, а блоки Display на Out. Автоматического добавления указанных блоков можно добиться при помощи выделения всех блоков, которые необходимо включить в модель и выполнением команды Create Subsystem (рисунок 16.3).

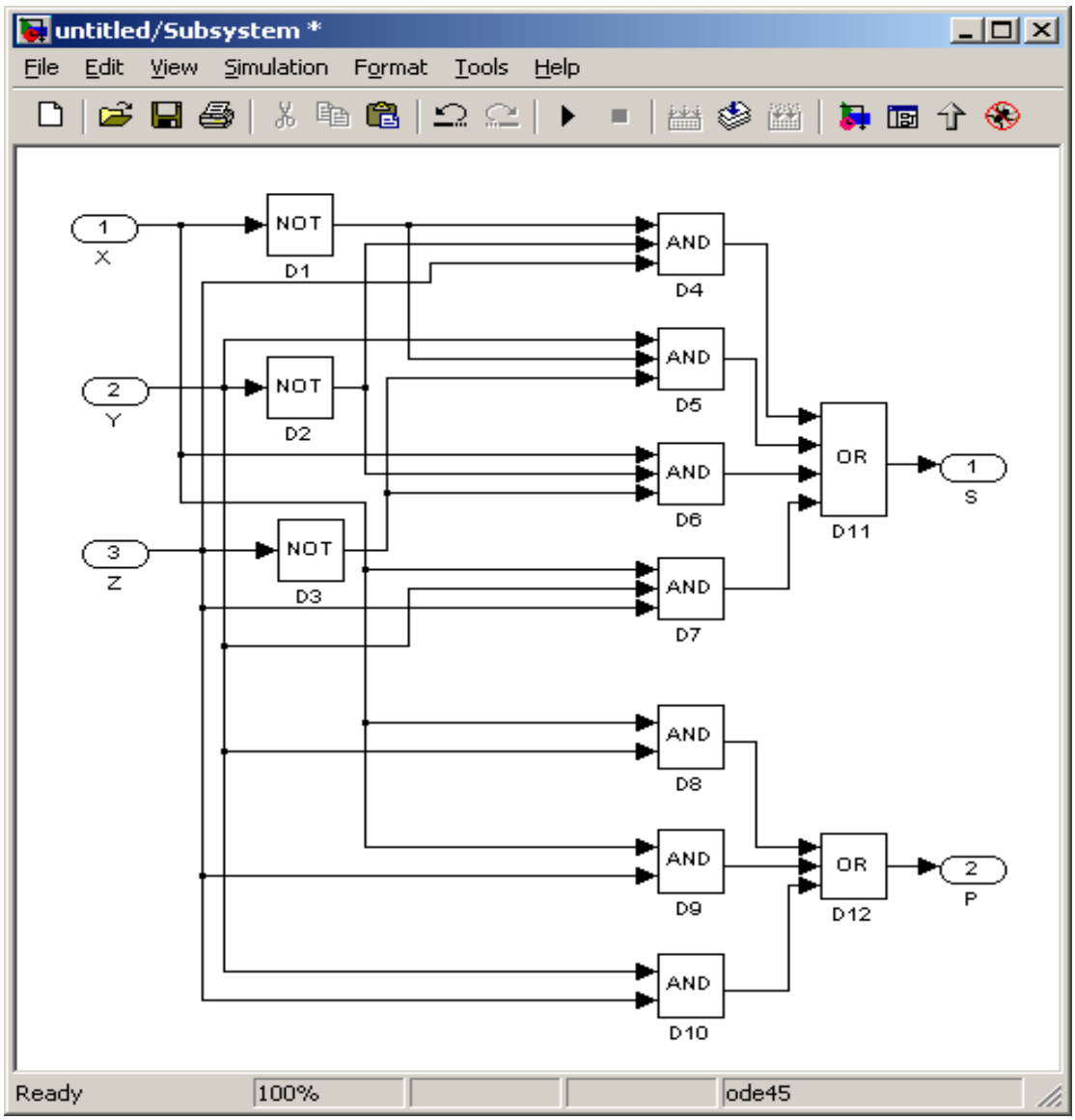


Рисунок 16.3 − Компьютерная модель сумматора после замены блоков

Библиотека, содержащая одноразрядный сумматор представлена на рисунке 16.4.

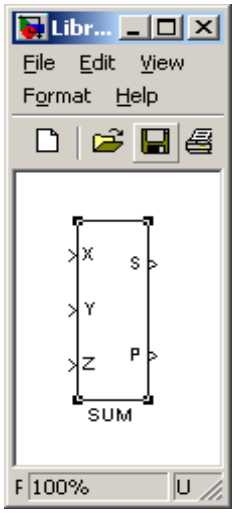


Рисунок 16.4 − Библиотека, содержащая одноразрядный сумматор

На рисунке 16.5 представлена компьютерная модель исследования одноразрядного сумматора в статическом режиме.

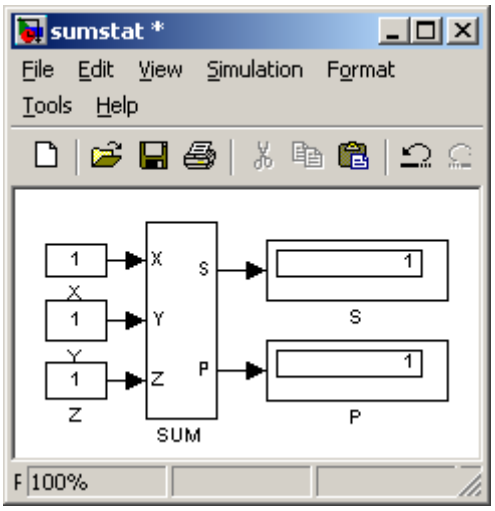


Рисунок 16.5 − Компьютерная модель исследования одноразрядного сумматора в статическом режиме

Для исследования функционирования одноразрядного сумматора в динамическом режиме необходимо блоки Constant заменить на блоки From File, а блоки Display на Scope. На рисунке 6 представлена компьютерная модель исследования одноразрядного сумматора в динамическом режиме.

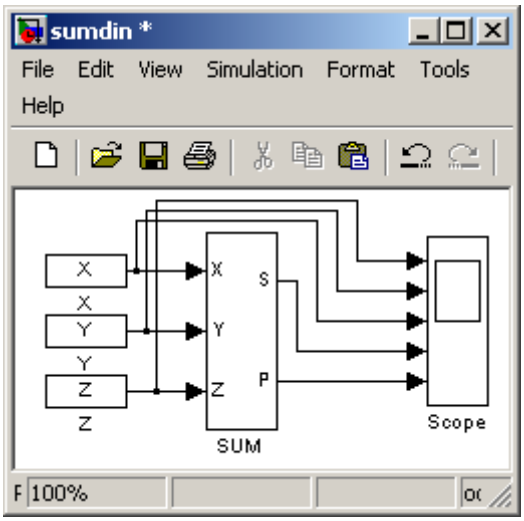


Рисунок 16.6 − Компьютерная модель исследования одноразрядного сумматора в статическом режиме

Для задания переменных необходимо в окне команд Matlab выполнить команды:

X=[0 1 2 3 4 5 6 7;0 0 0 0 1 1 1 1]
Y=[0 1 2 3 4 5 6 7;0 0 1 1 0 0 1 1]
Z=[0 1 2 3 4 5 6 7;0 1 0 1 0 1 0 1]

В результате в рабочей области будут сформированы 3 переменные

X,Y и Z (рисунок 7).

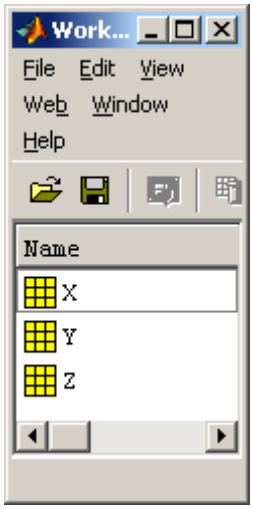


Рисунок 16.7 − Рабочая область Matlab

После этого каждую сформированную переменную необходимо сохранить в файл с расширением *.mat и указать эти файлы в качестве параметров блоков From File. Анализ временных диаграмм (рисунок 8), полученных путем запуска модели, позволяет сделать вывод о правильности функционирования сумматора.

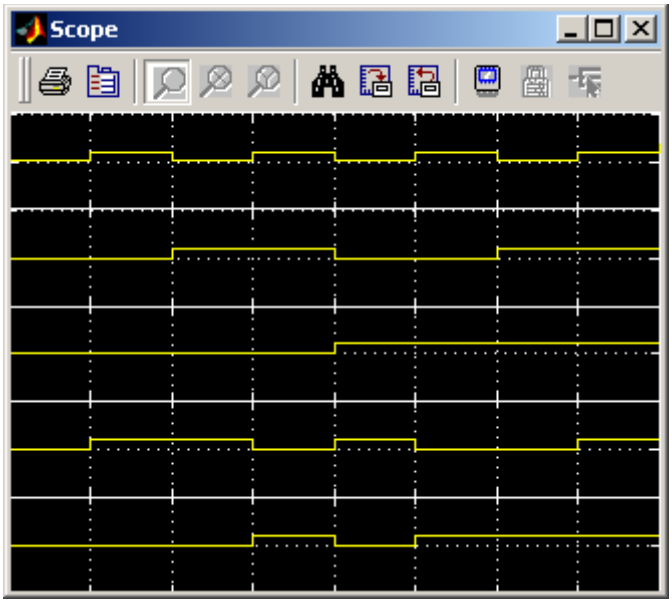


Рисунок 16.8 − Временные диаграммы исследования одноразрядного сумматора

Схема включения одноразрядного сумматора для сложения двух чисел, представленных в последовательном коде представлена на рисунке 9.

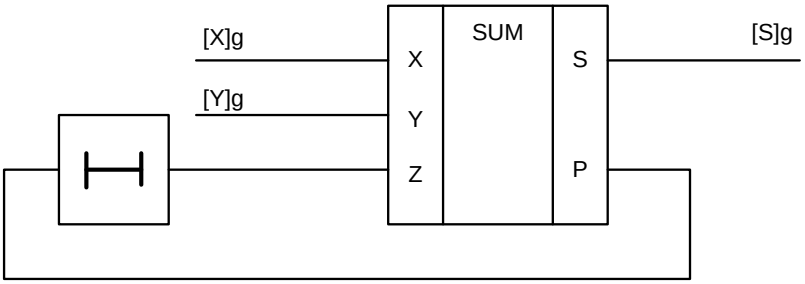


Рисунок 16.9 − Схема последовательного сумматора

Компьютерная модель исследования функционирования сумматора сложения чисел, представленных в последовательном коде изображена на рисунке 16.10.

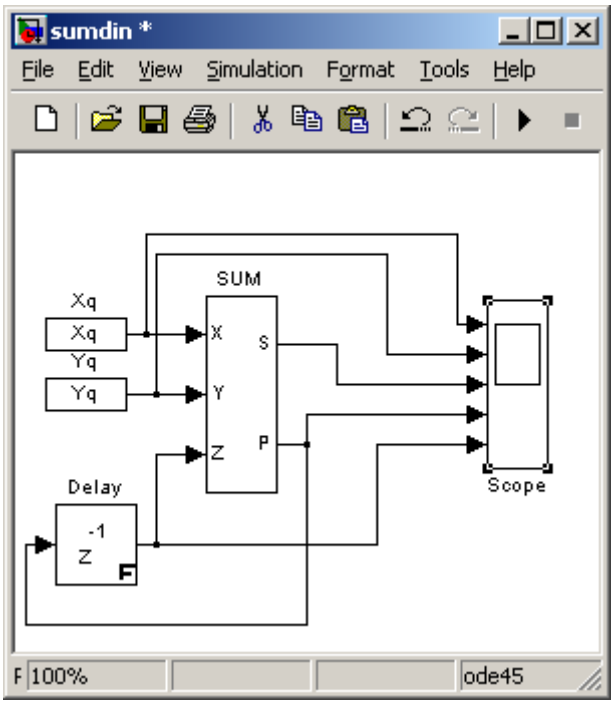


Рисунок 16.10 − Компьютерная модель сумматора сложения чисел, представленных в последовательном коде

Пусть необходимо сложить два числа X=11001 и Y=01101:

$$\begin{array}{r} 11001 \\ +\ 01101 \\ \hline 100110 \end{array}$$

Для этого в окне команд необходимо сформировать две матрицы:

Xq=[0 1 2 3 4;1 0 0 1 1]

Yq=[0 1 2 3 4;1 0 1 1 0]

сохранить их в виде файлов и указать в качестве параметра блоков From File. Анализ временных диаграмм (рисунок 16.11), полученных при запуске модели, представленной на рисунке 16.11, позволяет сделать вывод о правильности функционирования составленной модели.

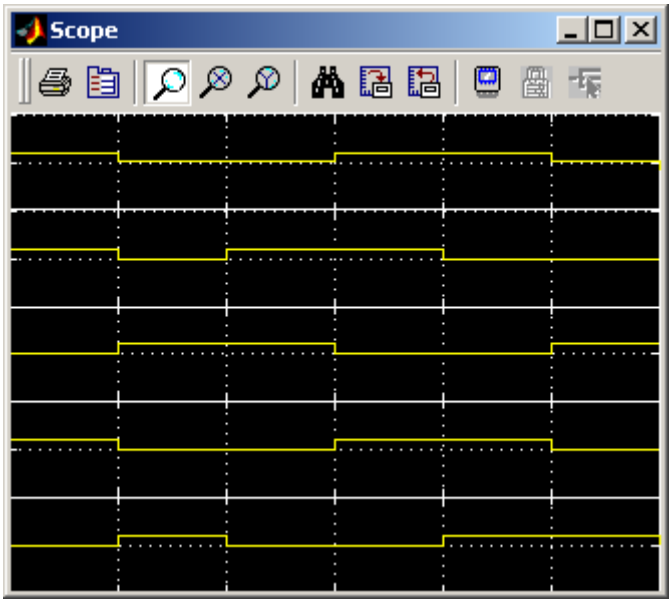


Рисунок 16.11 − Временные диаграммы исследования сумматора сложения чисел, представленных в последовательном коде

2. Исследование сумматоров сложения чисел, представленных в параллельном коде

Фрагмент схемы включения сумматора для сложения чисел, представленных в параллельном коде представлен на рисунке 16.12.

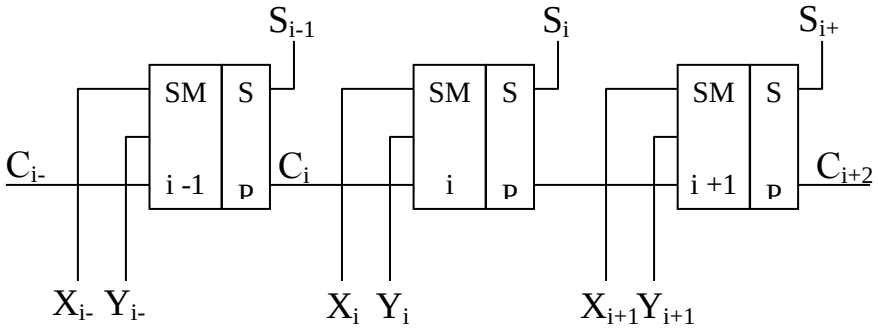


Рисунок 16.12 - Схема параллельного сумматора

Компьютерная модель сложения 5-ти разрядных чисел будет содержать 5 одноразрядных сумматоров (рисунок 16.13).

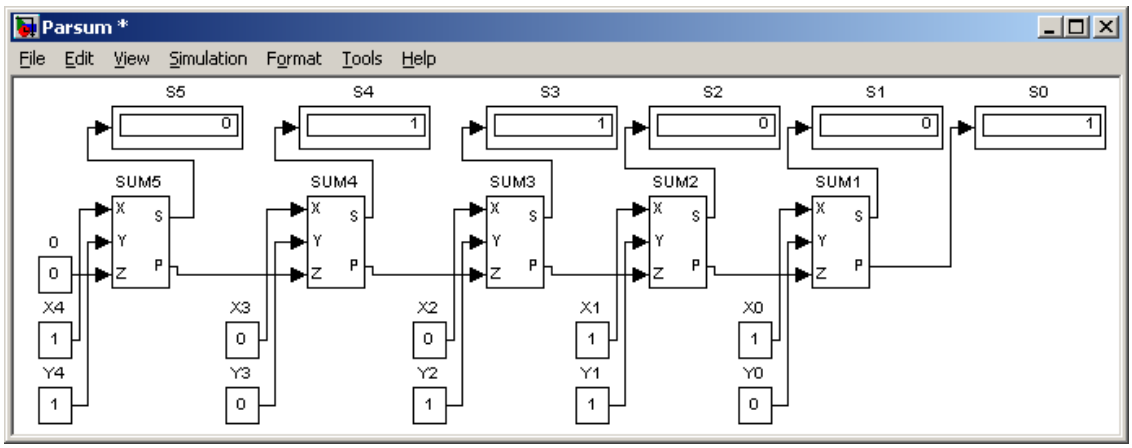


Рисунок 16.13 − Компьютерная модель параллельного сумматора

Запуск компьютерной модели, представленной на рисунке 16.13, позволяет сделать вывод о ее правильном функционировании.

Контрольные вопросы:

1. Приведите условно-графическое обозначение сумматора.
2. Приведите условно-графическое обозначение полусумматора.
3. Дайте определение сумматора.
4. Каково назначение линии задержки в схеме последовательного сумматора.

ЛИТЕРАТУРА

1. Васильев, С. А. Основы цифровой схемотехники в информационных система: учебное пособие / С. А. Васильев, И. Л. Коробова. — Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2021. — 81 с. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/122974.html
2. Кожухов, В. В. Электронные цепи и микросхемотехника. Импульсные и цифровые устройства. Конспект лекций: учебное пособие / В. В. Кожухов. — Новосибирск: Новосибирский государственный технический университет, 2021. — 166 с. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/126611.html
3. Митрошин, В. Н. Схемотехника цифровых устройств: учебное пособие / В. Н. Митрошин, А. Г. Мандра, Г. Н. Рогачев. — Самара: Самарский государственный технический университет, ЭБС АСВ, 2019. — 118 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/111423.html
4. Новиков, Ю. В. Введение в цифровую схемотехнику : учебное пособие / Ю. В. Новиков. — 3-е изд. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 392 с. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/89431.html
5. Фролов, А. В. Схемотехника цифровых устройств: лабораторный практикум / А. В. Фролов. — Комсомольск-на-Амуре: Комсомольский-на-Амуре государственный университет, 2022. — 129 с. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/122769.html
6. Червяков, Н.И. Цифровая схемотехника. От логического элемента до системы управления: Практикум / Н. И. Червяков, П. А. Ляхов, А. И Галушкин. – Ставрополь: СКФУ, 2015. – 191 с.
7. Червяков, Н.И. Цифровая схемотехника: Учебное пособие / Н. И. Червяков, А. И Галушкин. – Ставрополь: СКФУ, 2015. – 198 с.