

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего  
образования  
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ  
И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра автоматизации обработки информации (АОИ)

**ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ**

Методические указания к лабораторным работам для студентов направления  
“Информатика и вычислительная техника”

(уровень магистратуры)

Замятин Николай Владимирович

Вычислительные системы: методические указания к лабораторным работам для студентов направления подготовки “Информатика и вычислительная техника” (уровень магистратуры/ Н.В. Замятин. – Томск, 2018- с 41.

## Содержание

1. Введение.....	4
2. Операционный автомат.....	5
3. Управляющий автомат.....	15
4. Перцетрон.....	22
5. Исследование канала связи в сетях ЭВМ в виде RLC-цепи.....	26
6. Система моделирования электронных схем Electronics Workbench (EWB).....	31
7. Список рекомендуемой литературы.....	41

## **1.Введение**

Методические указания к лабораторному практикуму по дисциплине " Организация ЭВМ и систем" содержат на выбор описание 4 четырехчасовых лабораторных работ.

Лабораторные работы расположены последовательно, начиная от аппаратных реализаций элементов вычислительных систем, а также представлены современные параллельные вычислительные системы в виде нейронных сетей. Включена работа по сетевым вычислительным системам.

Целью лабораторных работ является закрепление практических навыков управления вычислительными системами на низком уровне и моделирования параллельных систем и систем сетевых технологий.

Для выполнения лабораторных работ студенты группы объединяются в бригады по 2-3 человека, работающие на закрепленном компьютере. Каждая бригада получает индивидуальное задание в соответствии с номером в журнале и оформляет отчет по лабораторной работе.

Выполнение лабораторной работы предполагает предварительное изучение соответствующего раздела курса и методических указаний к очередной работе.

Для допуска к выполнению лабораторной работы студент должен ознакомиться с темами для проработки и предварительно написать текст программы, в соответствии с индивидуальным заданием.

В течение выполнения лабораторной работы студент должен ответить на контрольные вопросы по предыдущей лабораторной работе. К лабораторной работе не допускаются студенты, не сдавшие более двух лабораторных работ.

Пропущенные лабораторные работы выполняются в конце семестра.

В процессе выполнения лабораторных работ следует ограничить перемещения в лаборатории.

## 2. Лабораторная работа №1. Синтез операционного автомата

### Цель работы.

Понять, каким образом выполняются арифметическо - логические операции в микропроцессоре. Научиться синтезировать операционный автомат (ОА) для выполнения вычислительных операций сложения и умножения.

### Основные понятия

Любой универсальный преобразователь информации состоит из следующих операционных устройств (ОУ):

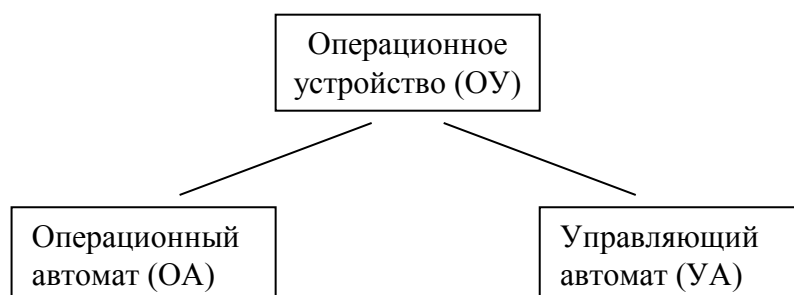
- арифметическо-логического устройства,
- памяти
- устройств ввода-вывода
- шин передачи сигналов.

Согласно концепции академика Глушкова В.М. любое операционное устройство декомпозируется на две части - пассивную исполнительную (ОА) и активную управляющую (УА).

АЛУ часть узлов (сумматор и др. КС) являются исполнителями, а часть узлов (распределитель сигналов) являются управляющими. У этих узлов разные принципы построения, организации. Поскольку принципы организации разные, то и их проектирование тоже раздельное.

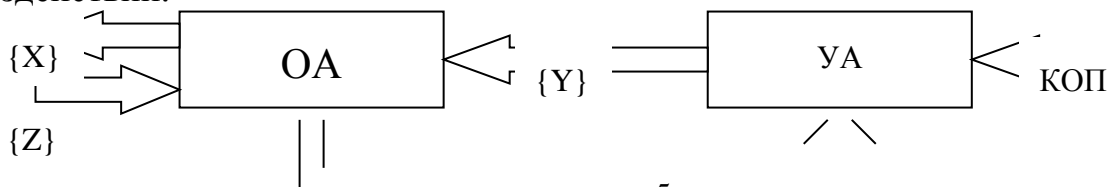
Операционное устройство представляет собой конечный автомат, состоящий из двух частей:

- 1.) операционный автомат;
- 2.) управляющий автомат;



Операционный автомат (ОА) – предназначен для арифметических и логических операций под управлением управляющего автомата.

Управляющий автомат (УА) – предназначен для формирования управляющих воздействий.



\_\_\_\_\_┐┌  
\_\_\_\_\_└└  
{x}

КОП – код операции;

{X} – входные сигналы;

{Y} – управляющие команды;

{Z} – выходные сигналы;

{x} – множество осведомительных сигналов (количество, переполнение и т. п.);

Основа функционирования – принцип микропрограммного управления

1. Любая операция это сложное действие, состоящее из совокупности элементарных действий, называемых микрооперациями (МО). Выполнение каждой МО осуществляется специальной комбинационной схемой (КС) за один такт машинного времени.

2. Порядок выполнения МО определяется алгоритмом операции и зависит от значений логических условий  $x$ . ЛУ принимают значения истина или ложь в зависимости от значений операндов.

3. Алгоритм, представленный, записанный в терминах МО и ЛУ, называется микропрограммой (МП). МП задает порядок выполнения МО и проверки ЛУ во времени.

4. Совокупность микропрограмм  $МП_1, \dots, МП_g$  задает функцию ОУ.

Все команды, формируемые УА называются микрокомандами (запись в триггер, считывание кода, сдвиг).

Rg A, Rg B – входные регистры;

S – сумматор;

Rg C – выходной регистр;

Перечень микроопераций

$y_1$  – запись через управляющую шину во входные регистры A и B;

$y_2$  – проверка знака числа (сдвиг для определения знакового разряда);

$y_3$  – считывание кодов из регистров;

$y_4$  – сложение;

$y_5$  – проверка знака суммы;

$y_6$  – выдача результата на выходную шину;

Совокупность микрокоманд образует микропрограмму действия ОА под воздействием микрокоманды, называемой микрооперацией.

Любая математическая или логическая операция в машинном исполнении состоит из набора простейших микроопераций: сложения и сдвига. Следовательно и ОА предназначен для выполнения двух простейших операций (сложения и сдвига). Поэтому в нём должны присутствовать два основных устройства – сумматор и регистры. Структурная схема операционного автомата приведена ниже.

По управляемой входной шине поступают операнды A и B, которые записываются соответственно в регистры Rg A и Rg B. В этих регистрах над операндами могут выполняться микрооперации сдвига и инвертирования

В определенный момент времени содержимое регистров поступает на сумматор, где выполняется операция суммирования. Результат операции записывается в регистр Rg C и затем поступает на выходную шину. Для формирования условий (признаков) в ОА имеется триггер и счетчик циклов.

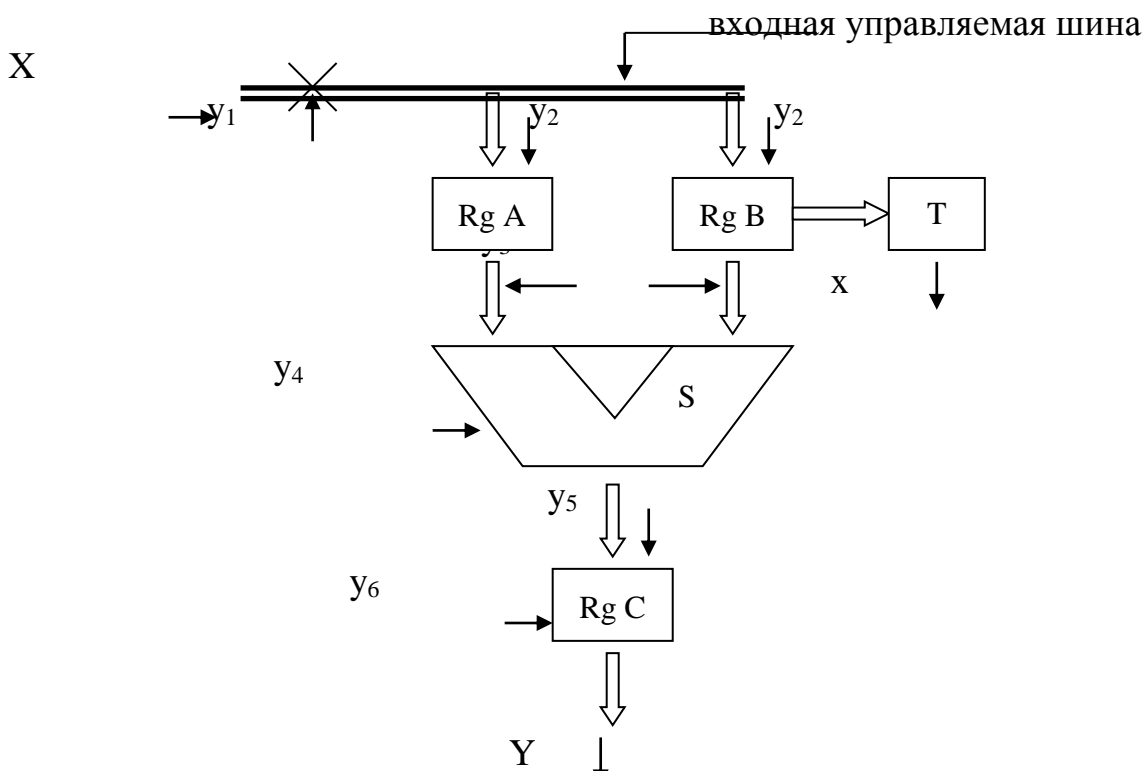
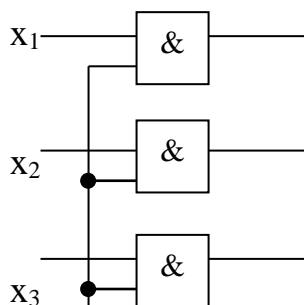


Рис. 1. Структурная схема операционного автомата

Элементы базиса построения операционного автомата:

Управляемая шина



1 – управляющая команда

Микрооперации  $y_1, y_2, y_3, y_4, y_5, y_6$  представляющие одноразрядные сигналы, и поступающие на входы регистров и сумматоров, формирует УА

Обработка информации с помощью ОУ осуществляется путем выполнения операций из списка F в последовательности, которая задается алгоритмом (программой)

решения задачи: ОА, выполняя программу, распределяет выполнение операций, предписанных командами программы, между различными ОУ - АЛУ, контроллерами ПУ.

**Описание входных и выходных слов и массивов.** Слово описывается своим именем и длиной:  $C(n_1:n_2)$ . Здесь  $C$  - имя слова,  $n_1, n_2$  - номера старшего и младшего разрядов слова соответственно. Часть слова называется **полем** и описывается аналогично словам:  $A(0:7)$ ,  $A(0:15)$ ,  $B(15)$ ,  $A(0)$  и т.п.

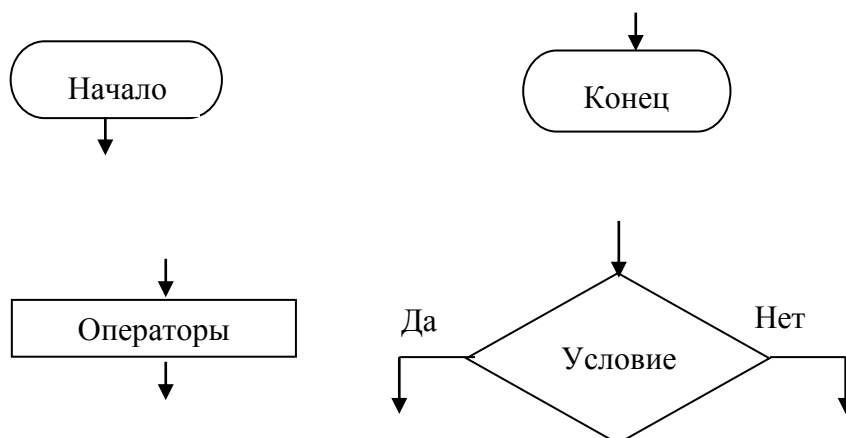
Массивы слов (например, запоминающее устройство) описывается в виде:  $M[m_1:m_2](n_1:n_2)$ . Здесь  $M$  - имя массива,  $m_1, m_2$  - номера первого и последнего элемента (ячейки) массива соответственно,  $n_1, n_2$  - определены выше. Пример описания локальной памяти как массива регистров:  $ЛП[0:15](0:31)$  - 16 тридцатидвухразрядных регистров.

**Описание МО.** Для описания МО используется оператор присваивания «:=» (или «←»). Слева от оператора указывается слово, поле, составное слово или элемент массива. Справа - двоичное выражение, которое описывает правило получения результата МО. Запись двоичного выражения осуществляется при помощи символов, обозначающих различные операции над операндами (словами), представленными в двоичной форме. Например,  $+$  - сложение кодов,  $\vee$  - логическая операция или и т.п. Примеры - в МП умножения.

**Описание ЛУ.** Для описания ЛУ используется различного рода отношения: " $<$ " - меньше, " $>$ " - больше, " $=0$ " - равно нулю, " $\neq 0$ " - не равно нулю и т.п. Примеры:  $A < 0$ ,  $B \geq 0$ ,  $C = 0$  и т.п. (смотри МП умножения).

**Порядок выполнения МО.** Порядок выполнения МО и проверки ЛУ задается в графической форме - в виде так называемой **граф-схемы алгоритма (ГСА)**. ГСА строится с использованием вершин четырех типов: начальной, конечной, операторной и условной и дуг, связывающих эти вершины (рисунок 5.3).

Начальная вершина имеет одну выходящую дугу. Конечная вершина имеет одну входящую дугу. Операторная вершина имеет одну входящую и одну выходящую дугу. В ней записывается один или несколько операторов присваивания, описывающих МО.





Условная вершина имеет одну входящую и две исходящих, отмеченных символами «да» (1) и «нет» (0). Выход по дуге «да» осуществляется в случае, если ЛУ принимает истинное значение (1), и по дуге «нет» - если ложное значение (0).

**Назначение ОА** - выполнение микроопераций из списка  $Y = \{y_1, \dots, y_M\}$  под воздействием управляющих сигналов  $y_m \in Y$  и формирование значений логических условий (осведомительных сигналов)  $X = \{x_1, \dots, x_L\}$ .

С каждым сигналом  $y_m \in Y$  в ОА отождествляется определенная МО. Поступление сигнала  $y_3$  в ОА приводит к выполнению этой МО и записи её результата в С.

С каждым логическим условиям  $x_l \in X$  в ОА отождествляется значение осведомительного сигнала, который принимает значение истина (единица) или ложь (ноль).

### Ход работы

Для синтеза ОА в лабораторной работе необходимо выполнить следующую последовательность действий:

1. Синтезировать RS – триггер,
2. Синтезировать параллельный регистр,
3. Синтезировать сумматор,
4. Собрать триггер, регистр и сумматор в среде Electronics Workbench, промоделировать их работу. Привести процедуры синтеза, схемы и эпюры в отчете,
5. Выполнить процедуру операции сложения двух трехразрядных чисел без знака,
6. Собрать операционный автомат в среде Electronics Workbench, используя модели регистров и сумматора, имеющиеся в библиотеке системы.
7. Промоделировать выполнение арифметических процедур на операционном автомате, путем подачи управляющих воздействий на соответствующие входы регистров и сумматора,
8. Подготовить отчет с описанием результатов выполненной работы и ответы на вопросы.

Пример синтеза RS-триггера.

Синтезировать логическую схему триггера.

последовательность действий:

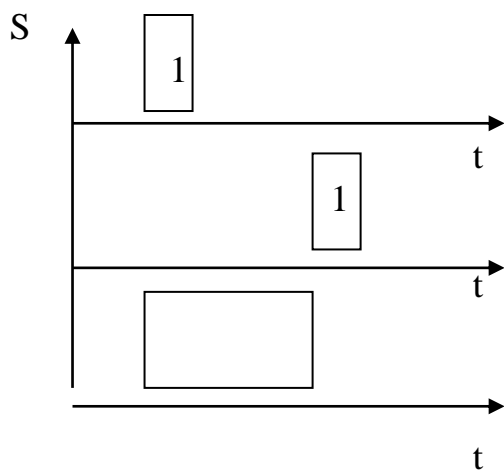
- уяснить задачу
- составить таблицу истинности
- получить булеву функцию
- минимизировать булеву функцию
- взять заданный базис (и - не либо или - не)
- составить логическую схему триггера
- произвести проверку
- изобразить эпюры сигналов

асинхронный RS – триггер.

S – SET (установка);  
 R – RESET (сброс);  
 Q(t) – состояние триггера;  
 Таблица истинности.

R	S	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	-
1	1	1	-

} т. к. на входе ничего нет.  
 } установка единицы.  
 } установка нуля.  
 } не используются.



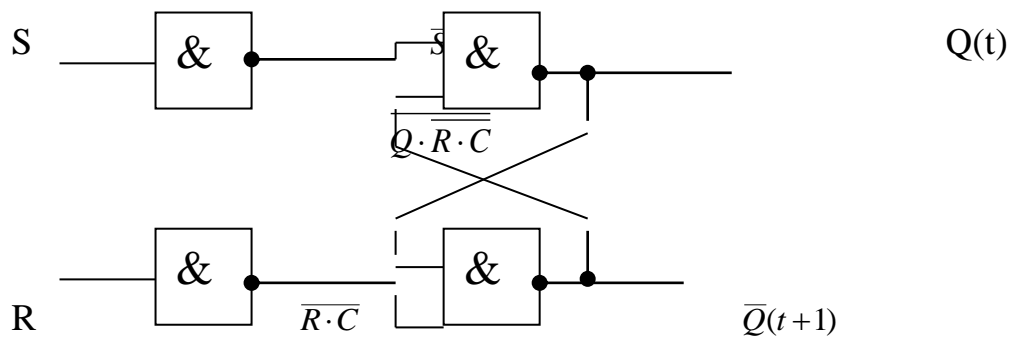
Эпюры выходных сигналов

Методом карт Карно определяем булеву функцию:

	R			
S	-	-	1	1
	6	7	3	2
	0	0	0	1
	4	5	1	0
	Q			

$$Q(t+1) = S + \bar{R} \cdot \bar{Q};$$

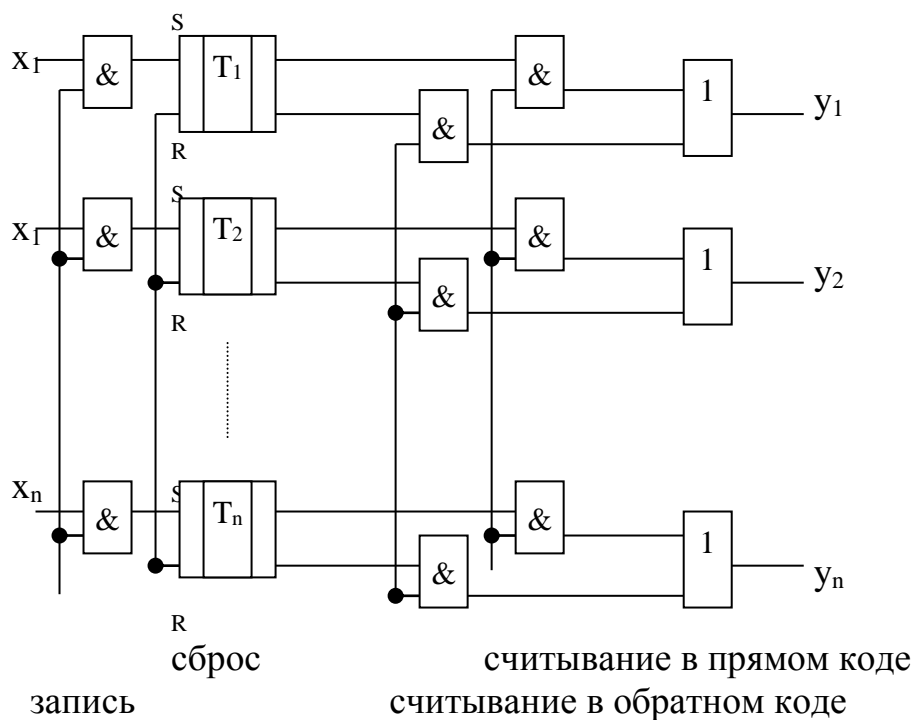
Выберем базис на элементах И – НЕ:



Проверка работы схемы:

$C=0; S=0; R=0; Q(t)=0; Q(t+1)=0.$

Синтезировать параллельный регистр, схема которого представлена на рисунке  
**Параллельный регистр**



$T_n$  – одноклапный RS - триггер.

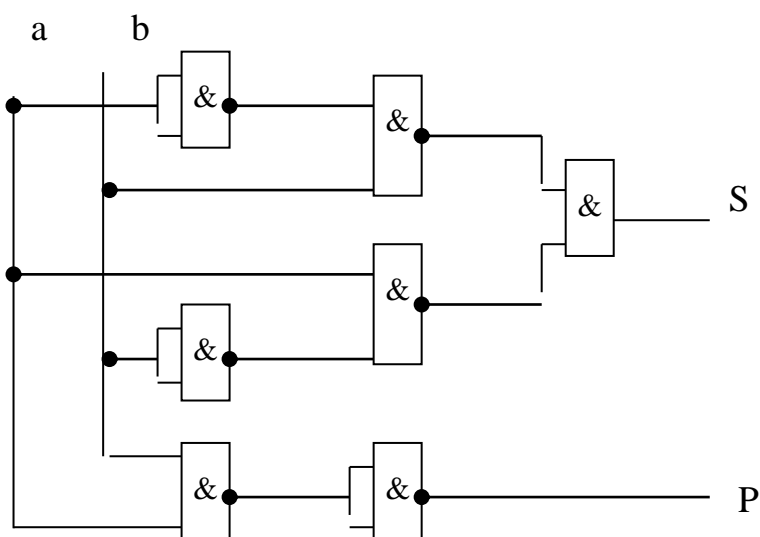
## Сумматор

a	b	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

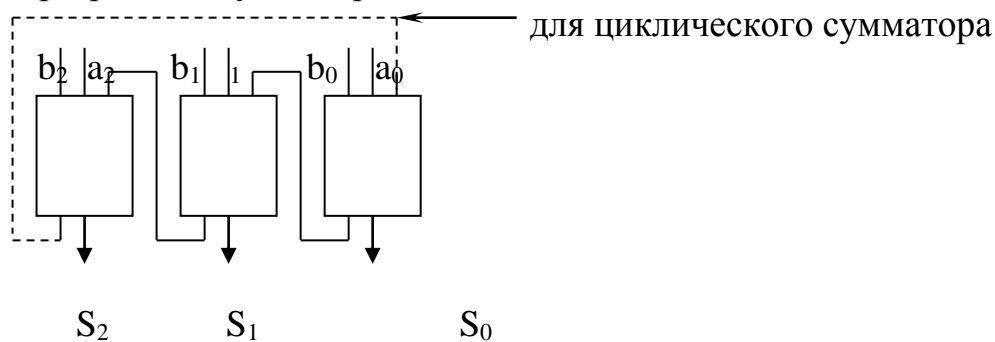
$$S = \overline{\overline{a \cdot b + a \cdot \overline{b}}} = \overline{\overline{a \cdot b} \cdot \overline{a \cdot \overline{b}}};$$

$$P = a \cdot b;$$

## Одноразрядный сумматор



## Трёхразрядный сумматор



## Выполнение арифметических операций

### Двоичная арифметика

Правила выполнения арифметических действий над двоичными числами определяются арифметическими действиями над одноразрядными двоичными числами.

Сложение	Вычитание	Умножение
$0 + 0 = 0$	$0 - 0 = 0$	$0 * 0 = 0$
$0 + 1 = 1$	$1 - 0 = 0$	$1 * 0 = 0$
$1 + 0 = 1$	$1 - 1 = 0$	$0 * 1 = 0$
$1 + 1 = 10$	$10 - 1 = 1$	$1 * 1 = 1$

↑ ↙

перенос  
в  
разряд

старший

Правила выполнения арифметических действий во всех позиционных системах счисления аналогичны.

### Сложение

Как и в десятичной системе счисления, сложение двоичных чисел начинается с правых (младших) разрядов. Если результат сложения цифр МЗР обоих слагаемых не помещается в этом же разряде результата, то происходит перенос. Цифра, переносимая в соседний разряд слева, добавляется к его содержимому. Такая операция выполняется над всеми разрядами слагаемых от МЗР до СЗР.

### Вычитание

Операция вычитания двоичных чисел аналогична операции в десятичной системе счисления. Операция вычитания начинается, как и сложение, с МЗР. Если содержимое разряда уменьшаемого меньше содержимого одноименного разряда вычитаемого, то происходит заем 1 из соседнего старшего разряда. Операция повторяется над всеми разрядами операндов от МЗР до СЗР.

Второй вариант операции вычитания - когда уменьшаемое меньше вычитаемого. В этом случае отрицательное число представляется в дополнительном коде.

### Умножение

Как и в десятичной системе счисления, операция перемножения двоичных многоразрядных чисел производится путем образования частичных произведений и последующего их суммирования. Частичные произведения формируются в результате умножения множимого на каждый разряд множителя, начиная с МЗР. Каждое частичное произведение смещено относительно предыдущего на один разряд. Поскольку умножение идет в двоичной системе счисления, каждое частичное произведение равно либо 0 (если в соответствующем разряде множителя стоит 0), либо является копией множимого, смещенного на соответствующее число разрядов влево (если в разряде множителя стоит 1). Поэтому умножение двоичных чисел идет путем сдвига и сложения. Таким образом, количество частичных произведений определяется количеством единиц в множителе, а их сдвиг - положением единиц (МЗР частичного произведения совпадает с положением соответствующей единицы в множителе). Положение точки в дробном числе определяется так же, как и при умножении десятичных чисел.

Операция умножения состоит в формировании суммы частичных произведений, которые суммируются с соответствующими сдвигами друг относительно друга. Этот

процесс суммирования можно начинать либо с младшего, либо со старшего частичного произведения. В ЭВМ процессу суммирования придают последовательный характер, т.е. формируют одно частичное произведение, к нему с соответствующим сдвигом прибавляют следующее и т.д. (т.е. не формируют все частичные произведения, а потом их складывают). В зависимости от того, с какого частичного произведения начинается суммирование (старшего или младшего), сдвиг текущей суммы осуществляется влево или вправо. При умножении целых чисел для фиксации результата в разрядной сетке число разрядов должно равняться сумме числа разрядов в  $X$  и  $Y$ .

Содержание отчета:

1. Титульный лист
  2. Цель работы
  3. Ход работы
- RS-триггер  
Многоразрядный сумматор  
Регистр  
Описание арифметической операции, согласно задания,  
Схема операционного автомата.
4. Ответы на вопросы

Вопросы к лабораторной работе

1. Принцип академика Глушкова В.М.
2. Какова разрядность управляющих сигналов, поступающих на ОА
3. Основные функции ОА
4. Какие основные микрооперации реализуются в ОА
5. В чем заключается принцип микропрограммирования
6. Каким образом меняется структура ОА для выполнения различных арифметическо-логических операций.

### 3.Лабораторная работа № 2 Синтез управляющего автомата

#### Цель работы:

Понять, каким образом выполняются арифметическо - логические операции в микропроцессоре. Научиться синтезировать управляющий автомат (УА) для выполнения вычислительных операций сложения и умножения.

#### Основные понятия

Типы УА:

- 1.) УА с жёсткой логикой;
- 2.) УА с программируемой логикой;

УА с жёсткой логикой вырабатывает управляющие воздействия, путем срабатывания триггеров, число которых определяется числом устойчивых состояний в граф-схеме алгоритма, и входными условиями.

В УА с программируемой логикой коды микроопераций выбирается из ячеек ПЗУ, исходя из входных условий,

Пример схемы алгоритма (ГСА) – представлен на рис.2. Здесь Сч - счетчик циклов.

Операция умножения разделяется на 7 МО, основные из которых сложение (реализуется за один такт комбинационным двоичным сумматором) и сдвиг (реализуется регистром сдвига). Порядок выполнения МО зависит от значений двух ЛУ (осведомительных сигналов):  $V(0)$ ,  $СЦ=0$ . ГСА умножения задает порядок выполнения МО и проверки ЛУ во времени. Например, в зависимости от  $V(0)$  в следующий момент времени, в следующем такте будет выполняться либо МО сложения  $C:=C+A$  (если  $V(0)=1$ ), либо МО сдвига, если  $V(0)=0$ .

Обработка информации с помощью ОУ осуществляется путем выполнения операций из списка  $F$  в последовательности, которая задается алгоритмом (программой) решения задачи: ОА, выполняя программу, распределяет выполнение операций, предписанных командами программы, между различными ОУ - АЛУ, контроллерами ПУ.

Запуск (инициализация) операции  $f_g \in F$  осуществляется путем подачи кода операции в ОУ из УУ. Реализация операции  $f_g$  осуществляется путем выполнения МО в порядке, заданном микропрограммой, хранимой внутри ОУ (т.е. без участия УУ). Работа (функционирование) ОУ во времени осуществляется тактами. Реализация МПг в общем случае занимает различное количество тактов  $n$ , т.е. время выполнения операции  $t_g = nT$ , где  $T$ -продолжительность такта.

Принцип микропрограммного управления является основой организации (построения) ОУ. Эти процедуры достаточно схожи с фон Неймановскими принципами функционирования. В основе управления лежит алгоритм. Только у Неймана он представляется в виде макропрограммы и поступает в процессор извне (из ОЗУ извлекается процессором). Здесь алгоритм в виде МП уже находится внутри ОУ. При выполнении программы ЦП генерирует определенную последовательность

операций, реализуемых ОУ. При выполнении операции  $f_g$  ОУ генерирует последовательность МО, реализуемых комбинационными схемами КС.

Отличия между этими принципами: 1) операция - сложное действие, для реализации которого необходимо ОУ. МО - элементарное действие, для реализации которого достаточно КС. 2) Операция выполняется за  $n$  тактов:  $t_{\text{опер}}=nT$ . МО выполняется за один такт (алгоритм – в структуре КС). КС управляется данными на ее входах.

Функция ОУ определяется совокупностью микропрограмм  $МП_1, \dots, МП_G$ , описывающих алгоритмы операций  $f_1, \dots, f_g$ . Для описания МП в языке используются различные средства, обеспечивающие описание слов, МО, ЛУ, а также средства, описывающие порядок их выполнения во времени.

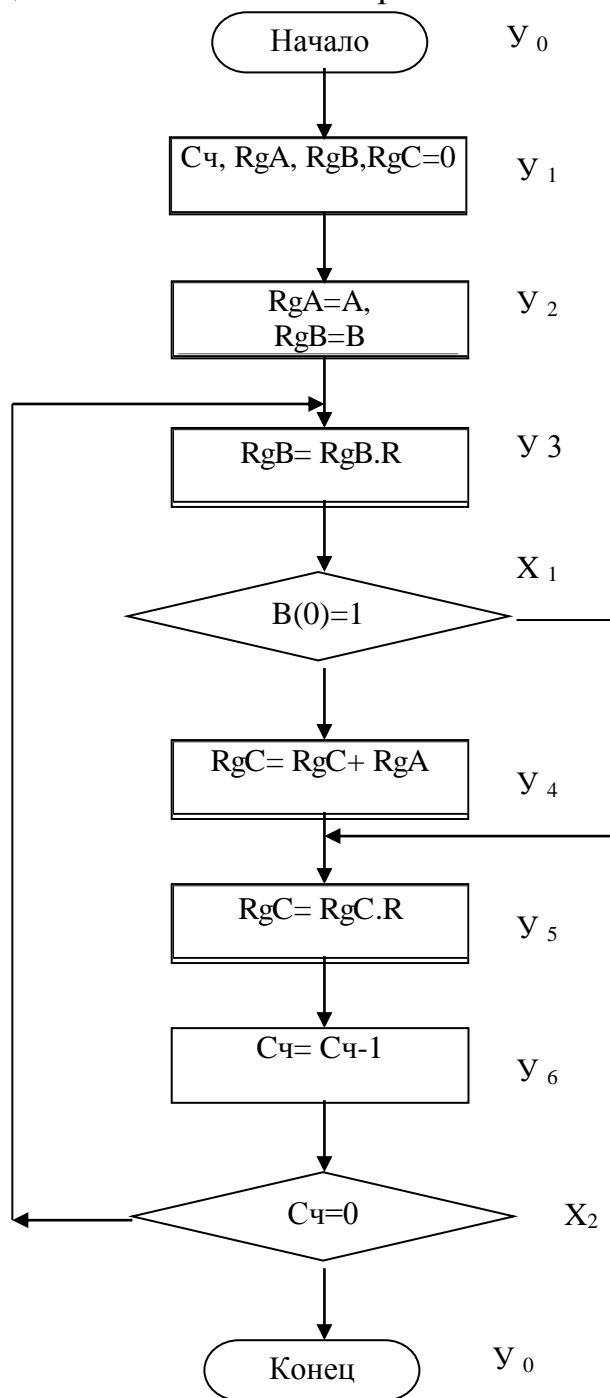




Рис.2. Граф-схема алгоритма

Пример синтеза регистра со сдвигом.

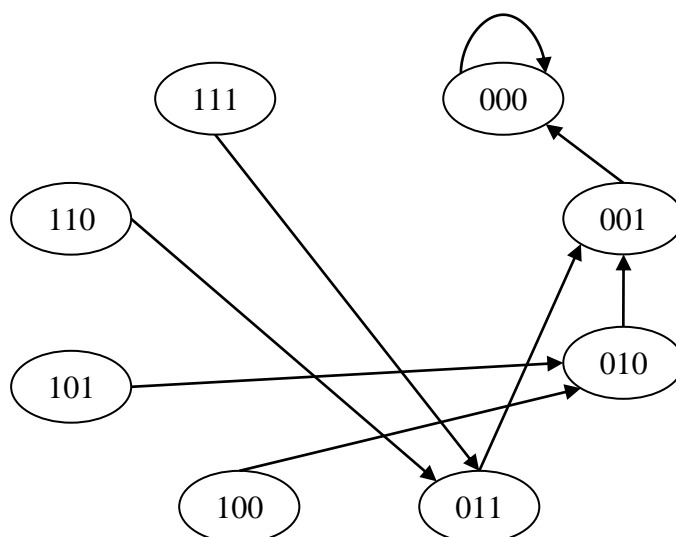
синтезировать логическую схему регистра;

- уяснить задачу;
- составить граф переходов;
- составить таблицу истинности;
- получить булевы функции;
- минимизировать булевы функции;
- взять заданный базис в виде заданных триггеров;
- составить логическую схему;
- произвести проверку.

Наименование регистра	Вид сдвига
D-триггер	Сдвиг вправо

Регистр – логический узел с памятью, состоящий из группы триггеров. Число триггеров соответствует количеству разрядов в слове, которое обрабатывается или запоминается в регистре. Рассмотрим 3-рядный регистр со сдвигом вправо. Он состоит из трёх D-триггеров, соединенных таким образом, что каждый тактовый импульс вызывает перемещение содержимого триггера в следующий за ним триггер.

Составляем граф переходов.



Граф переходов

Составляем таблицу истинности.

	x	Q <sub>2</sub> (t)	Q <sub>1</sub> (t)	Q <sub>0</sub> (t)	Q <sub>2</sub> (t+1)	Q <sub>1</sub> (t+1)	Q <sub>0</sub> (t+1)	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
8	1	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	0	0
10	1	0	1	0	0	0	1	0	0	1
11	1	0	1	1	0	0	1	0	0	1
12	1	1	0	0	0	1	0	0	1	0
13	1	1	0	1	0	1	0	0	1	0
14	1	1	1	0	0	1	1	0	1	1
15	1	1	1	1	0	1	1	0	1	1

По таблице истинности получим булевы функции

$$D_2 = 0$$

$$D_1 = XQ_2\bar{Q}_1\bar{Q}_0 + XQ_2\bar{Q}_1Q_0 + XQ_2Q_1\bar{Q}_0 + XQ_2Q_1Q_0$$

$$D_0 = X\bar{Q}_2Q_1\bar{Q}_0 + X\bar{Q}_2Q_1Q_0 + XQ_2\bar{Q}_1\bar{Q}_0 + XQ_2\bar{Q}_1Q_0$$

С помощью карт Карно минимизируем булевы функции

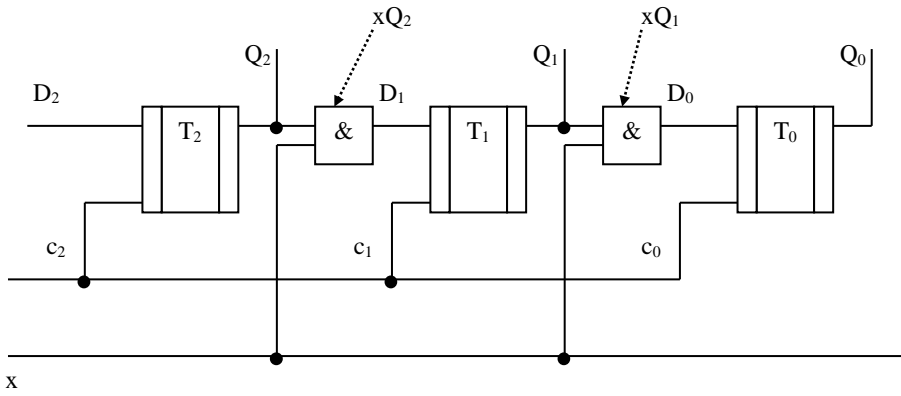
	x				
Q <sub>2</sub>	1 <sub>12</sub>	1 <sub>14</sub>	0 <sub>6</sub>	0 <sub>4</sub>	Q <sub>0</sub>
	1 <sub>13</sub>	1 <sub>15</sub>	0 <sub>7</sub>	0 <sub>5</sub>	
	0 <sub>9</sub>	0 <sub>11</sub>	0 <sub>3</sub>	0 <sub>1</sub>	
	0 <sub>8</sub>	0 <sub>10</sub>	0 <sub>2</sub>	0 <sub>0</sub>	
		Q <sub>1</sub>			

$$D_1 = XQ_2$$

	x				
Q <sub>2</sub>	0 <sub>12</sub>	1 <sub>14</sub>	0 <sub>6</sub>	0 <sub>4</sub>	Q <sub>0</sub>
	0 <sub>13</sub>	1 <sub>15</sub>	0 <sub>7</sub>	0 <sub>5</sub>	
	0 <sub>9</sub>	1 <sub>11</sub>	0 <sub>3</sub>	0 <sub>1</sub>	
	0 <sub>8</sub>	1 <sub>10</sub>	0 <sub>2</sub>	0 <sub>0</sub>	
		Q <sub>1</sub>			

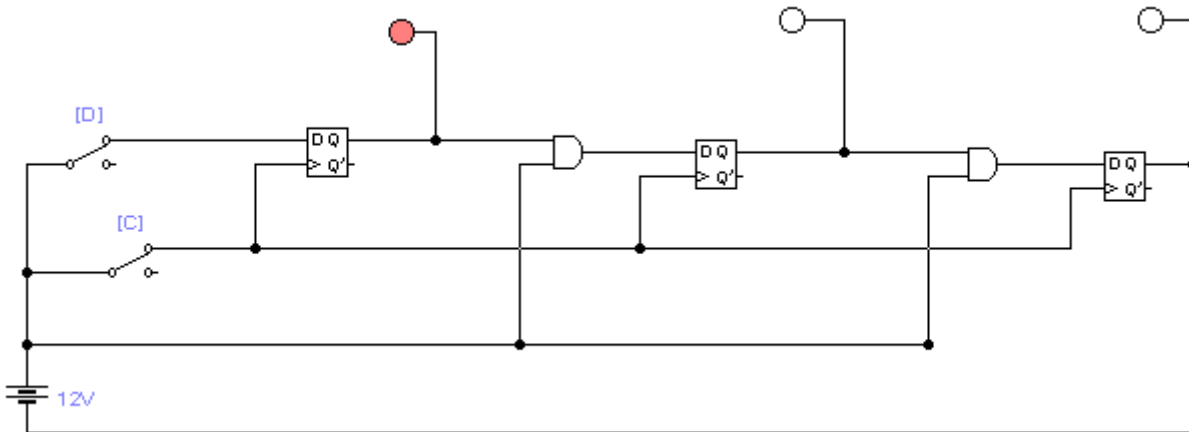
$$D_0 = XQ_1$$

Составим логическую схему



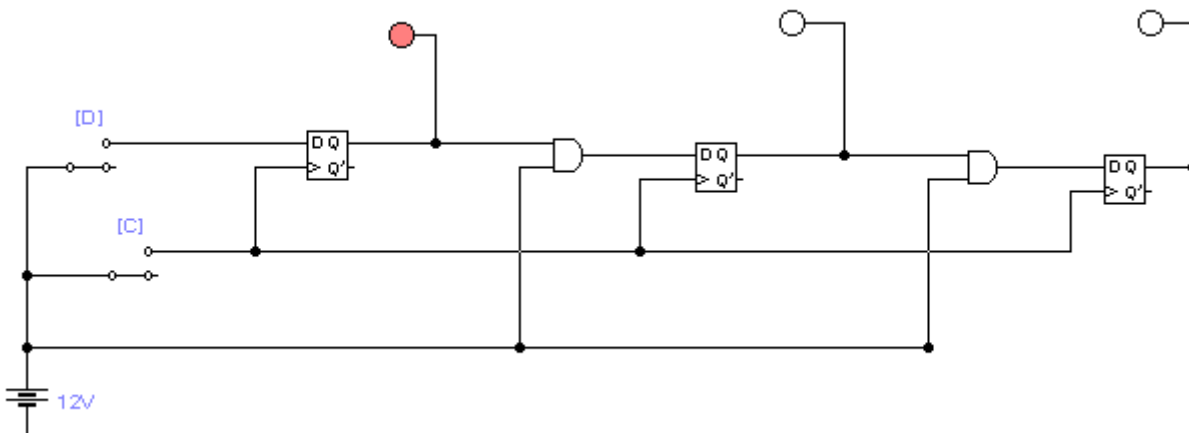
Реализуем логический узел в среде EWB и докажем его работоспособность записыванием и считыванием трехразрядных кодов.

Шаг 1.  $Q_2=Q_1=Q_0$ . Запишем в регистр трехразрядный код – 100, включив оба ключа D и C ( $D_2=1$ ,  $D_1=x \& Q_2=1 \& 0=0$ ,  $D_0=x \& Q_1=1 \& 0=0$ ).



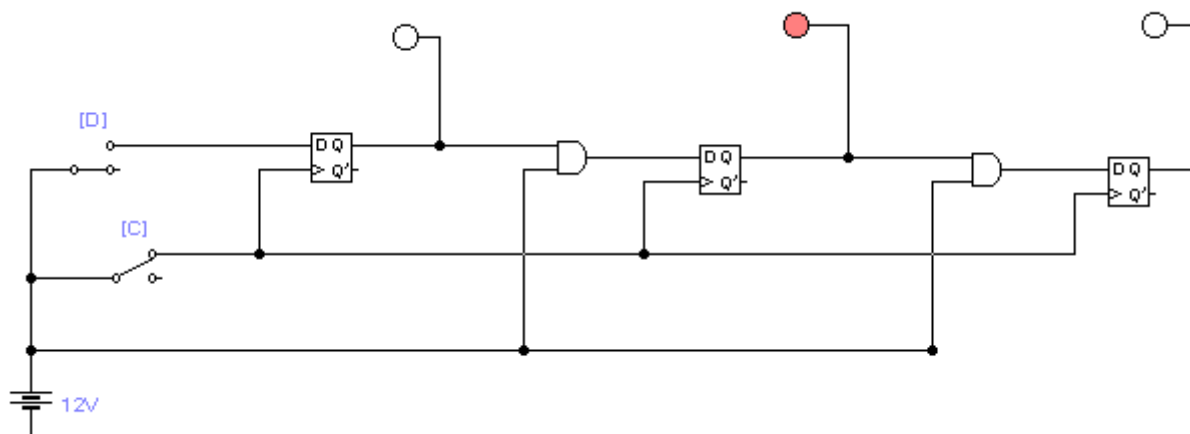
$Q_2(t+1)=1$ , при  $D_2=1$  и  $C_2=1$ ;  $Q_1(t+1)=0$ , при  $D_1=0$  и  $C_1=1$ ;  $Q_0(t+1)=0$ , при  $D_0=0$  и  $C_0=1$ .

Шаг 2. Выключим оба ключа (D и C).



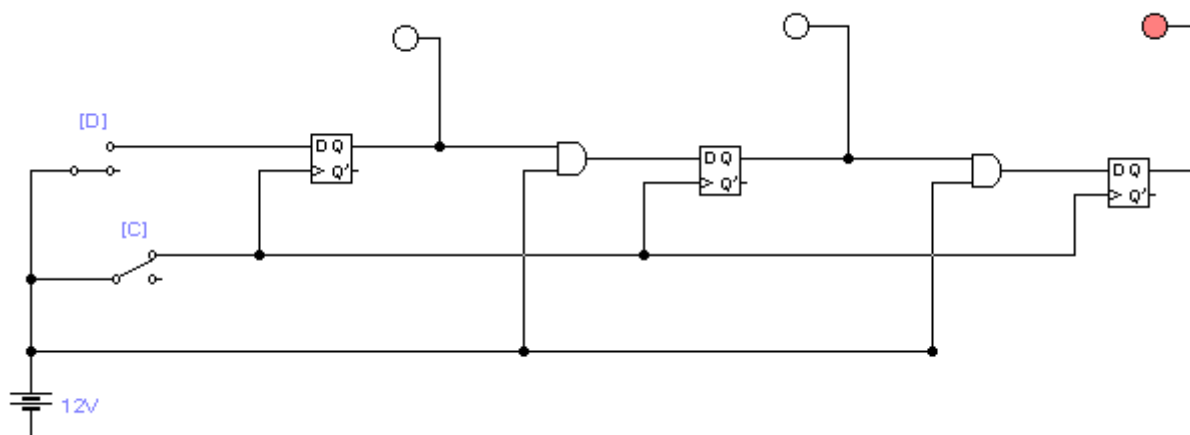
Произошла задержка сигнала, т.е. в регистре сохранился трехразрядный код – 100.

Шаг 3. Включим ключ С, что соответствует тактовому импульсу, вызывающему перемещение содержимого триггера в следующий за ним триггер, при выключенном ключе D ( $D_2=0$ ,  $D_1=x \& Q_2=1 \& 1=1$ ,  $D_0=x \& Q_1=1 \& 0=0$ ).



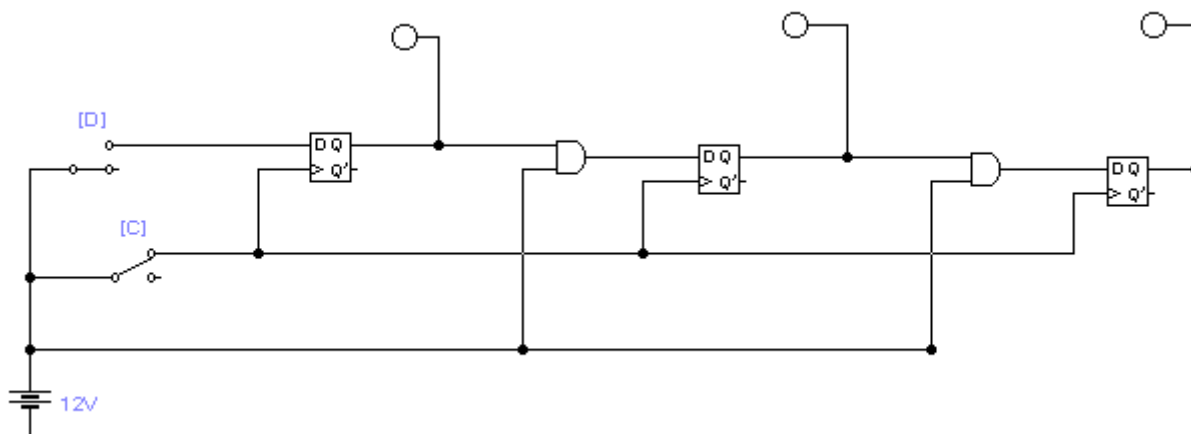
Получили трехразрядный код – 010.  $Q_2(t+1)=0$ , при  $D_2=0$  и  $C_2=1$ ;  $Q_1(t+1)=1$ , при  $D_1=1$  и  $C_1=1$ ;  $Q_0(t+1)=0$ , при  $D_0=0$  и  $C_0=1$ .

Шаг 4. Выключим и снова включим ключ С, тем самым подав следующий тактовый импульс ( $D_2=0$ ,  $D_1=x \& Q_2=1 \& 0=0$ ,  $D_0=x \& Q_1=1 \& 1=1$ ).



Получили трехразрядный код – 001.  $Q_2(t+1)=0$ , при  $D_2=0$  и  $C_2=1$ ;  $Q_1(t+1)=0$ , при  $D_1=0$  и  $C_1=1$ ;  $Q_0(t+1)=1$ , при  $D_0=1$  и  $C_0=1$ .

Шаг 5. Повторим предыдущую операцию ( $D_2=0$ ,  $D_1=x \& Q_2=1 \& 0=0$ ,  $D_0=x \& Q_1=1 \& 0=0$ ).



Получили трехразрядный код – 000.  $Q_2(t+1)=0$ , при  $D_2=0$  и  $C_2=1$ ;  $Q_1(t+1)=0$ , при  $D_1=0$  и  $C_1=1$ ;  $Q_0(t+1)=0$ , при  $D_0=0$  и  $C_0=1$ .

В результате выполнения работы должна быть синтезирована логическая схема последовательного регистра со сдвигом вправо на D-триггерах. Функционирование синтезированной схемы подтверждает правило изменения трехразрядного кода занесенного в регистр, при подачи тактовых импульсов (100 → 010 → 001 → 000), что соответствует построенному графу переходов.

Аналогичным образом необходимо синтезировать управляющий автомат.

### Последовательность действий при синтезе УА

1. Уяснить задачу
2. Ознакомиться со способом выполнения арифметической операции
3. Составить ГСА
4. Разметить ГСА
5. Составить граф переходов
6. Выбрать триггер
7. Составить таблицу истинности
8. Получить булевы функции;
9. Минимизировать булевы функции;
10. Взять заданный базис в виде заданных триггеров;
11. Составить логическую схему;
12. Произвести проверку.

Содержание отчета:

1. Титульный лист
2. Цель работы
3. Ход работы
  - триггер со сдвигом
  - описание арифметической операции, согласно задания,
  - ГСА арифметической операции
  - граф переходов
  - схема управляющего автомата.
  - результаты проверки

#### 4. Ответы на вопросы

##### Вопросы к лабораторной работе

1. В чем отличие УА с жесткой логикой от УА с программируемой логикой.
2. В какой форме представляются сигналы из УА для управления в ОА.
3. Какую роль играет разметка в ГСА
4. Почему D- триггер предпочтительнее применять в УА с жесткой логикой.
5. Чем конечный автомат Мили отличается от автомата Мура.
6. Чем определяется количество триггеров в УА с жесткой логикой.

#### 4. Лабораторная работа № 3 Перцептрон

##### Цель работы

Целью данной лабораторной работы является изучение работы перцептрона и подготовка программы, позволяющей определить сдвиг одного восьмиразрядного двоичного числа относительно другого. (Возможно выполнение усложненного задания) Разпознавание образа знака на основе модели однослойного перцептрона.

##### Общие сведения

Перцептрон - простая нейронная модель, показанная на рис. 1 состоит элемента  $\Sigma$ , и пороговой функции. Элемент  $\Sigma$  умножает каждый вход  $x$  на вес  $w$  и суммирует взвешенные входы. Если эта сумма больше заданного порогового значения, выход равен единице, в противном случае – нулю. В зависимости от выходного сигнала принимают решение: 1 – входной сигнал принадлежит классу №1, 0 – классу №2. Такие модели, а также модели, подобные им, применяют в распознавании образов, и в задачах классификации. Перцептронные сети обладают простотой обучения и программной реализации, а также служат основой для изучения более сложных нейронных сетей. Ёмкость данной сети совпадает с числом нейронов в сети.

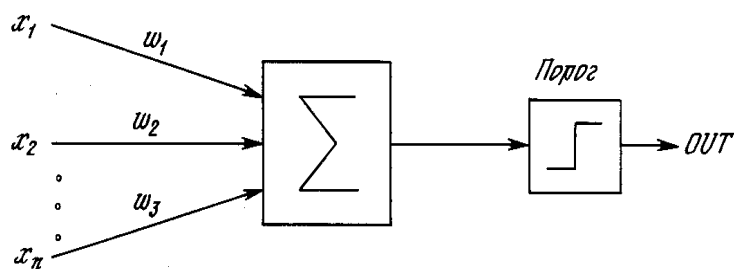


Рис.1. Математическая модель нейрона

1. Алгоритм обучения

1. Инициализация сети (задаём параметры: скорость обучения, величина порога, число обучений, веса  $w$  задаём случайным образом).
2. Подача входного образа (вектор  $X_i$ ).
3. Подача желаемого выходного результата.  $Y$ .
4. Вычисление суммы  $S_j = \sum X_i(t) * W_i(t)$ .
5. Вычисление выходного сигнала  $Y_{\text{выч.}} = F(S_j)$ , где  $F(S_j)$  – пороговая функция.
6. Проверка достижимости действительного выходного сигнала желаемому. Если достигли, переход на шаг 2, иначе, подстраиваем весовые коэффициенты  $W_i(t)$ .
7. Проверка окончания входных образов. Если закончились, то конец, иначе, переход на шаг 2.

### Ход работы

1. Ознакомится с теоретической частью по НС.
2. Составить алгоритм обучения НС методом обратного распространения ошибки.
3. Написать программу моделирования химической системы, реализующую этот алгоритм. (данные получить у преподавателя).
4. Ответить на контрольные вопросы.
5. Подготовить отчет и защитить лабораторную работу.

Пример реализации интерфейса для программного модуля персептрон

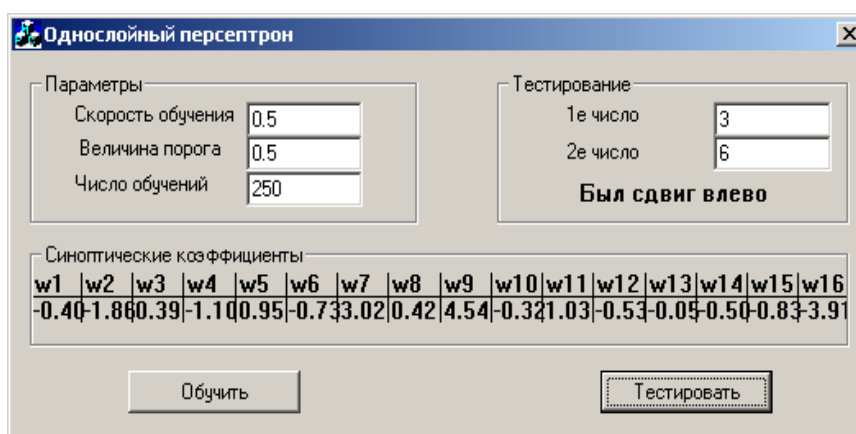


Рис. 2. Сдвиг влево

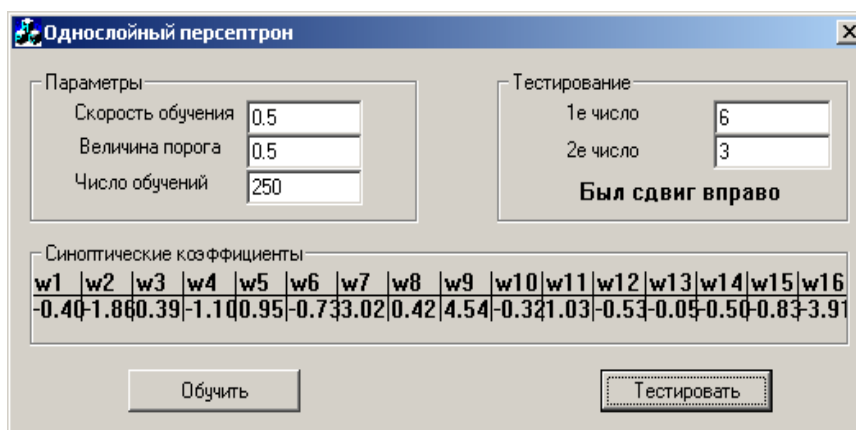


Рис. 3. Сдвиг вправо.

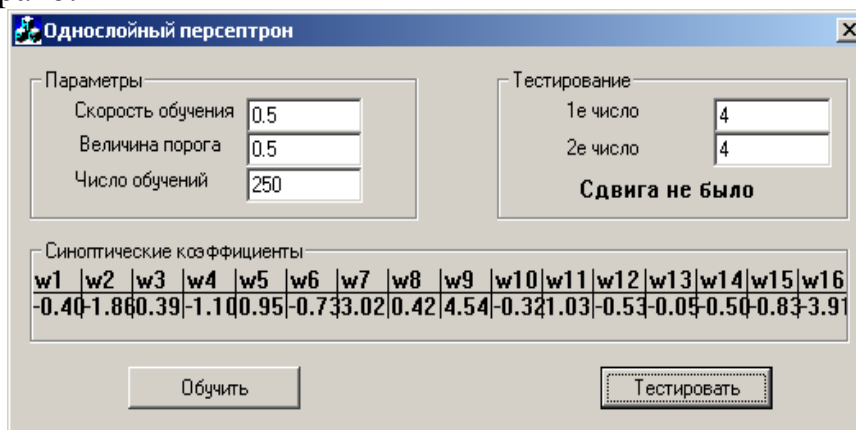


Рис. 4. Сдвига не было.

Также возможно выполнение усложненного задания с реализацией перцептрона для распознавания образов цифр перцептроном. Обучение можно выполнить на 3-х наборах цифр (3 шрифта).

0

3

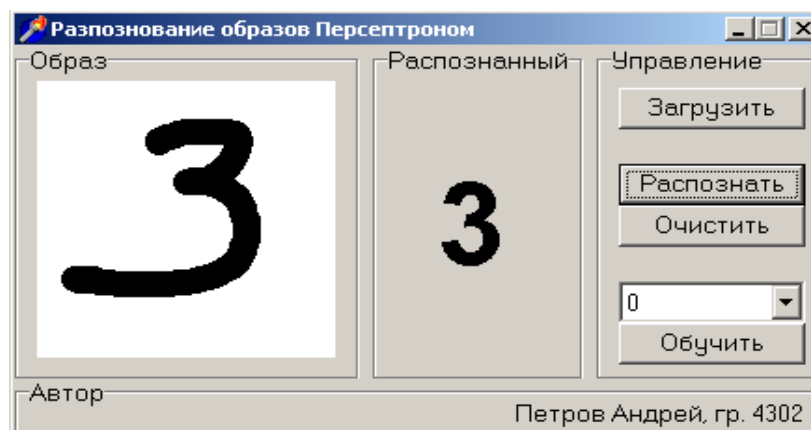
7

Arial

Comic Sans

Times New Roman

Затем провести тестирование на рукописных цифрах с определением процентов правильных распознаваний. Пример реализации интерфейса приведен на рис. 5.



## 2. Содержание отчета

1. Цель работы
2. Ход работы
3. Описание работы перцептрона
4. Задание на лабораторную работу



5. Схема алгоритма
6. Результаты работы
7. Выводы
8. Листинг программы

#### Контрольные вопросы

1. Чем отличаются перцептроны с прямыми и обратными связями,
2. Чем отличается обучение с учителем от обучения без учителя.
3. Какие существуют методы обучения нейронных сетей.
4. какую функцию активации имеет однослойный перцептрон.
5. Как влияет скорость обучения на качество обучения перцептрона .
6. Почему порядок предъявления примеров в обучающей выборке может влиять на качество обучения.
7. В чем преимущества метода обратного распространения ошибки.
8. В чем его недостатки.
9. Как влияет уменьшение количества входных нейронов на функционирование сети.
10. Какими свойствами должна обладать функция активации при использовании алгоритма обратного распространения ошибки.

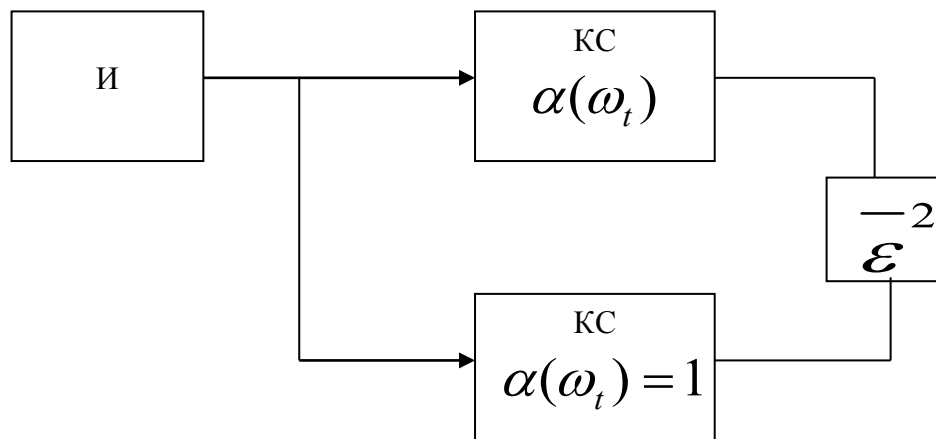
## 5. Лабораторная работа 4. Исследование канала связи в сетях ЭВМ в виде RLC-цепи

### Цель работы.

Цель работы состоит в изучении метода оценки качества канала связи на основе потерь энергии при передаче сигналов определенной формы по этому каналу. В качестве канала принять RLC-цепь.

### Ход работы.

Для изучения качества передачи канала связи наиболее целесообразно использовать критерий в виде потерь энергии сигнала. В нашем случае был выбран прямоугольный сигнал. Для анализа будем использовать следующую модель:



где И - источник, формирующий сигнал  $U(t)$ ,

$\alpha(\omega_t)$  – передаточная характеристика канала,

$\varepsilon^2$  – потери энергии сигнала связи.

Для оценки потерь энергии сигнала необходимо найти разность энергий сигналов, прошедших реальным каналом с передаточной характеристикой  $\alpha(\omega_t)$  и идеальным каналом с  $\alpha(\omega_t) = 1$ . Тогда энергия потерь  $\varepsilon^2$  будет иметь вид:

$$\begin{aligned}\varepsilon^2 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} |W(\omega_t)|^2 * d\omega_t - \frac{1}{2\pi} \int_{-\infty}^{\infty} |W(\omega_t)|^2 * |\alpha(\omega_t)|^2 d\omega_t = \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} |W(\omega_t)|^2 * (1 - |\alpha(\omega_t)|^2) d\omega_t,\end{aligned}$$

где  $W(\omega_t)$  – спектральная функция сигнала,

$\alpha(\omega_t)$  – передаточная характеристика канала,

$\overline{\varepsilon}^2$  - потери энергии, усредненные по всем частотам

Прямоугольный сигнал описывается функцией:

$$U(t) := \begin{cases} U_0 & \text{if } 0 < t < T_i \\ 0 & \text{otherwise} \end{cases},$$

и имеет вид, представленный на рис. 1.  $U_0$  и  $T$  были взяты в интервале от 0 до 1, так как при таких значениях потери энергии будут незначительными.  $T=0,8$   $U_0=0.9$

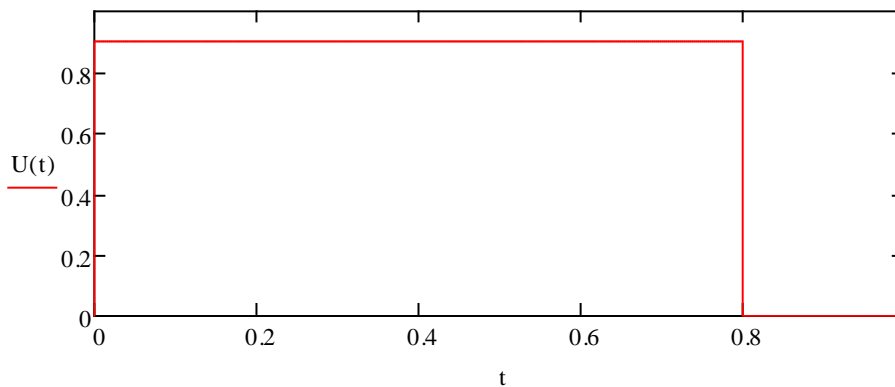


Рис. 1 – Функция прямоугольного сигнала.

Спектральная функция сигнала и зависимость  $W(\omega_t)$  приведены рис.2:

$$W(\omega_t) = U_0 T_i * \frac{\sin \frac{\omega_t T}{2}}{\frac{\omega_w T}{2}}$$

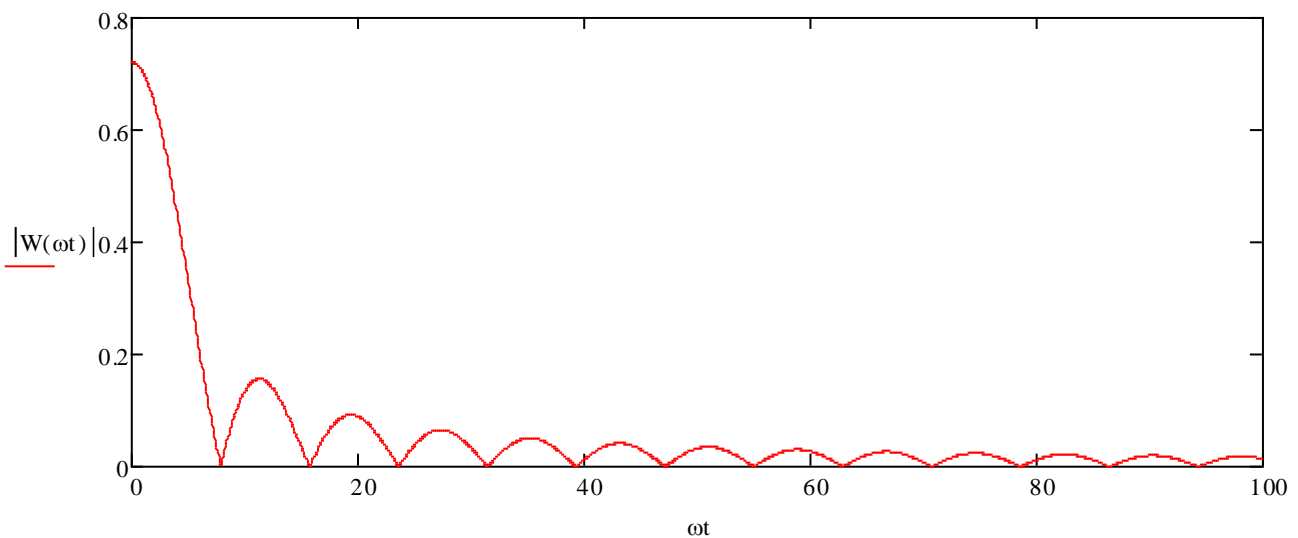


Рис. 2 – Спектральная функция сигнала

Для описания функции пропускания канала была использована интегральная RLC-цепь. В этом случае передаточная характеристика  $\alpha(\omega_i)$  определяется соотношением

$$\alpha(\omega_i) = \frac{Q}{\sqrt{1 + (2 \cdot \frac{Q \cdot (\omega_i - \omega_0)}{\omega_0})^2}}, \text{ где } Q = \frac{\rho}{R}, \rho = \sqrt{\frac{L}{C}}, \omega_0 = \frac{1}{\sqrt{L \cdot C}}$$

В нашем случае  $R=30$  Ом,  $C=0.4$  мкФ,  $L=10^{-3}$  (значения подобрать самостоятельно)

На рис. 3 представлена передаточная характеристика канала.

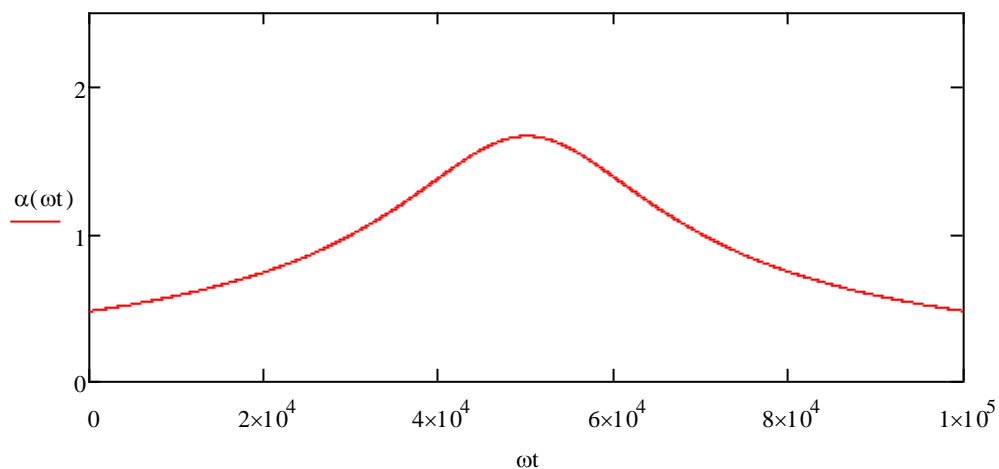


Рис.3 – Передаточная характеристика канала.

Для наглядного выявления частот сигнала, которые не пройдут при использовании канала связи, зависимости пропускных способностей реального и идеального каналов представлены на одном графике (рис. 4).

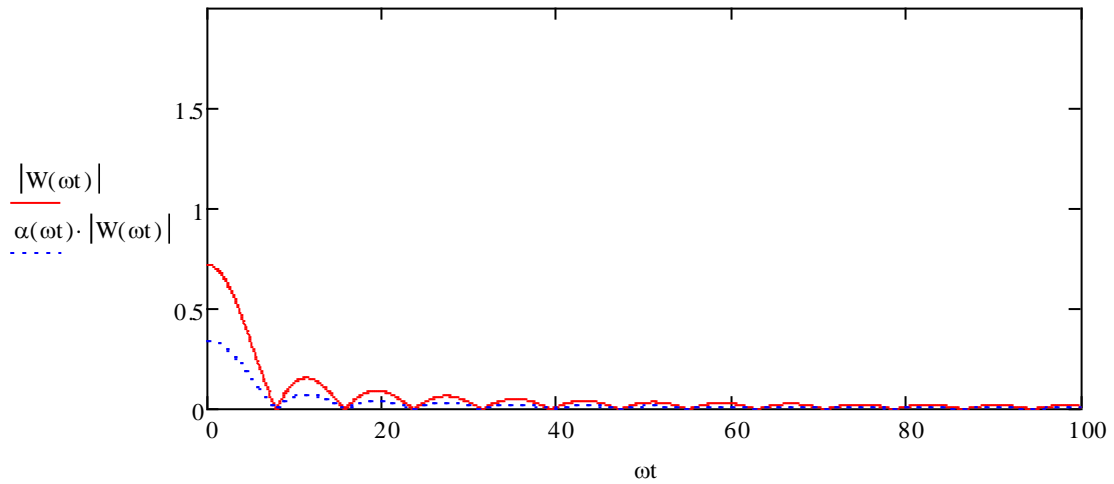


Рис.4 – пропускные способности реального и идеального каналов.

Энергии для идеального и реального каналов определяются по следующим формулам:

$$E1 := \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} (|W(\omega t)|)^2 d\omega t$$

$$E2 := \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} (|W(\omega t)|)^2 \cdot (|\alpha(\omega t)|)^2 d\omega t$$

$$E1 = 0.648$$

$$E2 = 0.149$$

Количество энергии, которое теряется при несовпадении полосы частот сигнала и канала определяется формулой:

$$\varepsilon_{\omega t} := (|W(\omega t)|)^2 \cdot [1 - (|\alpha(\omega t)|)^2]$$

График зависимости количества энергии, которая теряется при несовпадении полосы частот и канала приведен на рис.5.

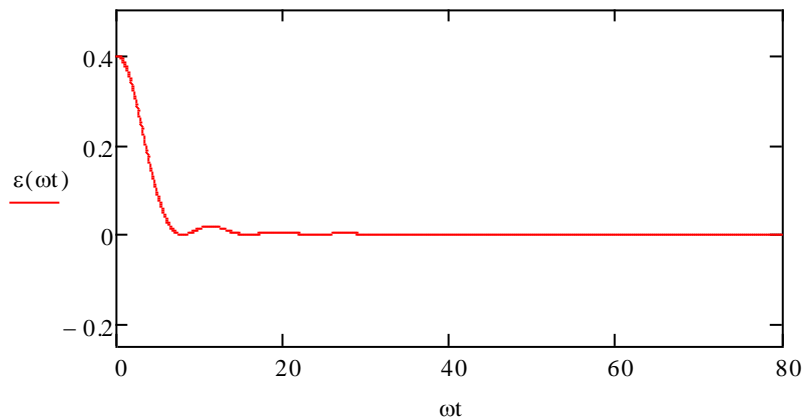


Рисунок 5 - График зависимости  $\varepsilon(\omega t) := (|W(\omega t, t)|)^2 \cdot [1 - (|\alpha(\omega t)|)^2]$

### **Результаты работы.**

В результате проделанной работы был изучен метод оценки качества канала связи на основе потерь энергии при передаче сигналов определенной формы по этому каналу.

### **Контрольные вопросы**

1. Для какой цели предназначен канал связи в сетях ЭВМ
2. Что такое спектральная функция сигнала
3. Каким образом представляется спектральная функция сигнала
4. Каким образом представляется передаточная функция канала
5. Почему в цифровых сетях эвм используют цифровое представление сигналов.

### **Содержание отчета**

1. Цель работы
2. Краткое содержание хода работы
3. Формула и график спектральной функции сигнала
4. Формула и график передаточной функции канала
5. Графики пропускной способности реального и идеального каналов
6. График зависимости потерь энергии сигнала
7. Выводы по работе
8. Ответы на контрольные вопросы.

## **6. Методические указания по работе с системой моделирования электронных схем Electronics Workbench (EWB)**

Программный комплекс EWB разработан фирмой Interactive Image Technologies (Канада) для схемотехнического моделирования цифровых и аналоговых радиоэлектронных устройств.

Предварительное исследование электронной схемы с применением компьютерного моделирования позволяет найти оптимальные параметры для работы исследуемого устройства, не прибегая к его практической реализации. Исследование на программной модели позволяет ознакомиться с возможностями проверки правильности построения схем. При разработке сложных схем физическое моделирование бывает просто невозможно из-за чрезвычайной сложности устройства.

Особенность программы EWB в наличии в ней контрольно-измерительных приборов, по внешнему виду, органам управления и характеристикам максимально приближенных к их промышленным аналогам.

Программа EWB 4.1 рассчитана для работы в среде Windows 3.xx или 95.98 и занимает около 5 Мбайт дисковой памяти, EWB 5.0 - в среде Windows 95/98 и NT 3.51, требуемый объём дисковой памяти - около 16 Мбайт. Для размещения временных файлов требуется дополнительно 10.....20 Мбайт свободного пространства.

В данном руководстве рассматривается версия EWB5PRO и EWB v.5.12.

### **Структура окна и система меню.**

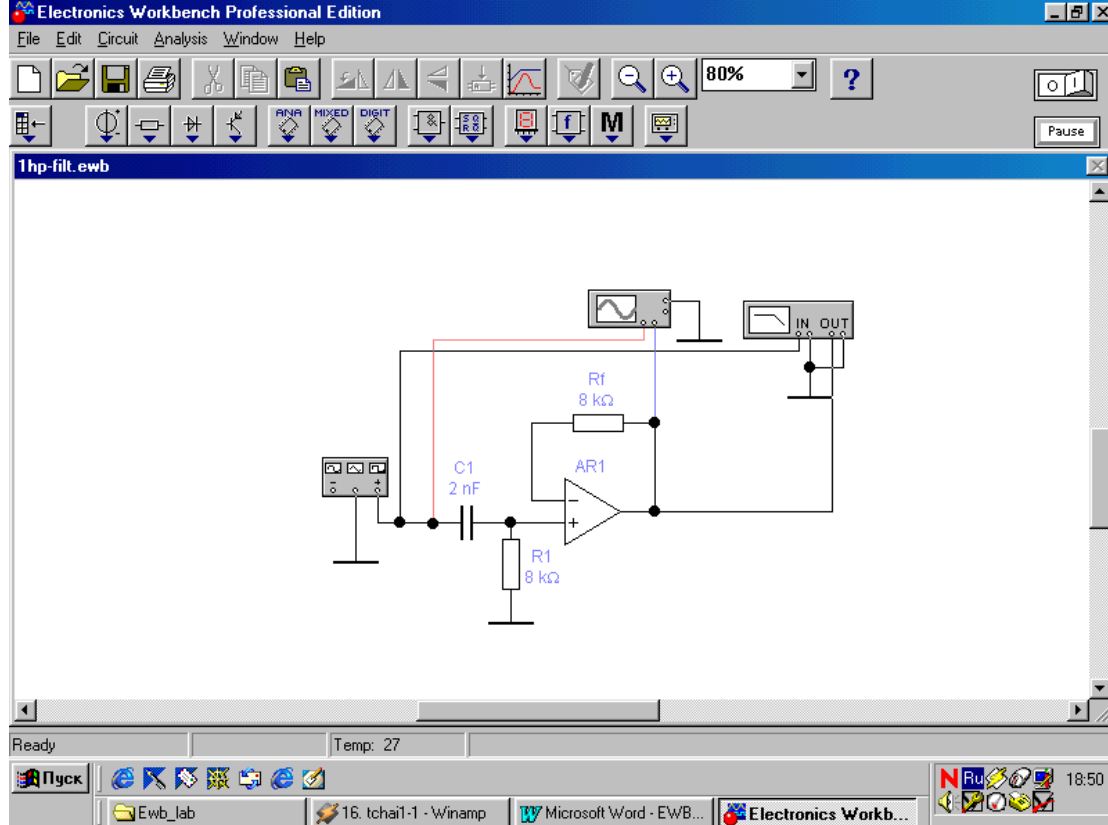
Окно содержит строку команд меню, строку основных типовых электронных устройств, поле для составления исследуемой схемы и полосы управления прокруткой.

Основные команды меню:

#### ***Меню File:***

первые четыре команды меню типовые и пояснений не требуют.

- **Revert to Saved** -стирание всех изменений, внесенных в текущем сеансе редактирования, и восстановление схемы в первоначальном виде.
- **Install** - установка дополнительных программ с жёстких дисков.
- **Import** - импорт текстовых файлов описания схемы .
- **Export** - составление текстового описания схемы и задания на моделирование в формате SPICE.



Окно программы EWB 5.x.

### **Меню Edit:**

- **CUT** - стирание (вырезание) выделенной части схемы с сохранением в буфере обмена. Выделение одного компонента производится щелчком мыши на изображении компонента. Для выделения части схемы или нескольких компонентов курсор мыши в левый угол воображаемого прямоугольника, охватывающего выделяемую часть, нажать левую кнопку мыши и, не отпуская её, протянуть курсор по диагонали этого прямоугольника, контуры которого появляются уже в начале движения мыши, и затем отпустить кнопку. Выделенные компоненты окрашиваются в красный цвет.
- **COPY** - копирование выделенной части схемы в буфер обмена.
- **PAST**- вставка содержимого буфера обмена на рабочее поле программы. Фрагмент затем ещё будучи отмеченным перетаскивается с помощью мыши в нужное место.
- **DELETE** - стирание выделенной части схемы.
- **SELECT ALL** - выделение всей схемы.
- **COPYBITS** - команда превращает курсор мыши в крестик, которым по правилу прямоугольника можно выделить нужную часть экрана, после отпущения левой кнопки мыши выделенная часть копируется в буфер обмена, после чего его содержимое может быть импортировано в любое приложение Windows. Копирование всего экрана производится нажатием клавиш Print Screen; копирование активной в данный момент части экрана, например, диалогового окна - комбинацией Alt+Print Screen.
- **Show Clipboard**- показать содержимое буфера обмена.
- **Copy as Bitmap** - копирует выделенный участок в буфер обмена.

**Меню Circuit** - используется при подготовке схем, а также для задания параметров моделирования.



- **Activat** - запуск моделирования.
  - **Stop** - остановка моделирования. Эти две команды дублируются нажатием кнопки выключателя, расположенного в правом верхнем углу экрана.
  - **Pause** - прерывание моделирования.
  - **Label** - ввод позиционного обозначения выделенного компонента с помощью диалогового окна.
  - **Value** - изменение номинального значения параметра компонента с помощью диалогового окна.
  - **Model** - выбор модели компонента, команда выполняется также двойным щелчком по компоненту. Работа с меню, как и во всех других подобных случаях, заканчивается нажатием кнопок **Accept** или **Cancel** - с сохранением или без сохранения введённых изменений.
  - **Zoom** - раскрытие (развёртывание) выделенной подсхемы или контрольно-измерительного прибора, команда выполняется также двойным щелчком мыши по иконке компонента или прибора.
  - **Rotate**- вращение выделенного компонента.
  - **Fault** - имитация неисправности выделенного компонента путём введения :
    - leakage- сопротивления утечки,
    - short - короткого замыкания,
    - open - обрыва,
    - none - отсутствие неисправности (включено по умолчанию).
  - **Subcircuit** - преобразование предварительно выделенной части схемы в подсхему.
  - **Wire Color** - изменение цвета предварительно выделенного проводника. Расцветка проводников важна в случае применения логического анализатора, - в этом случае цвет проводника определяет цвет временной диаграммы.
  - **Preferences**- выбор элементов оформления схемы в соответствии с меню.
- Для создания схем, рассматриваемых в рамках лабораторных работ по дисциплине "организация ЭВМ и систем" достаточно воспользоваться имеющимися типовыми компонентами.

Для открытия нужной библиотеки компонентов нужно подвести курсор мыши к соответствующей иконке и нажать один раз её левую кнопку. В выпадающем множестве выбирается необходимый значок, и передвигается при удержании левой клавиши мыши на рабочее поле программы. Для установки параметров необходимо двойным нажатием левой кнопкой мыши раскрыть меню настройки параметров компонента. Выбор подтверждается нажатием кнопкой

**Accept** и клавишей **Enter**.

После размещения компонентов производится соединение их выводов проводниками. При этом необходимо учитывать, что к выводу компонента можно подключить только один проводник.

Для выполнения подключения курсор мыши подводится к выводу компонента и после появления прямоугольной площадки синего цвета, нажимается левая кнопка и появляющийся при этом проводник протягивается к выводу другого компонента до

появления на нём такой же прямоугольной площадки, после чего кнопка мыши отпускается и соединение готово. При необходимости подключения к этим выводам других проводников в библиотеке Passive выбирается точка (символ соединения) и переносится на ранее установленный проводник. После удачной постановки точки

к проводнику подсоединяется ещё два проводника.

Точка соединения может быть использована не только для подключения проводников, но и для введения надписей.

Если необходимо переместить отдельный сегмент проводника, к нему подводится курсор, нажимается левая кнопка и после появления в вертикальной или горизонтальной плоскости двойного курсора производятся нужные перемещения.

Подключение к схеме контрольно-измерительных приборов производится аналогично. Причём для таких приборов, как осциллограф или логический анализатор, соединения целесообразно проводить цветными проводниками, поскольку их цвет определяет соответствующую осциллограмму.

## Основные компоненты EWB.

### Компонент Выход из EWB.



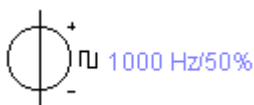
### Вспомогательные компоненты - группа SOURCES:



- заземление (метка) . точка нулевого потенциала в схеме.



- источник фиксированного напряжения +5 вольт



- генератор однополярных прямоугольных импульсов (амплитуда, частота, коэффициент заполнения).

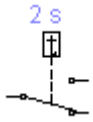
### Основные пассивные элементы - группа BASIC:



- точка соединения проводников, используется также для введения на схему надписей длиной не более 14 символов (других способов введения текста в EWB не существует).



- переключатель, управляемый нажатием задаваемой клавишей клавиатуры (в квадратных скобках), по умолчанию- клавиша пробела.
- переключатель, автоматически срабатывающий через заданное время на включение

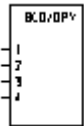


и выключение (время в секундах).

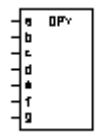
### Индикаторные приборы - группа **INDICATORS**.



- светоиндикатор (свет свечения может быть настроен красным, зелёным и синим)



- семисегментный индикатор с дешифратором .



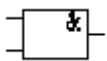
- семисегментный индикатор .

10 W/12 V

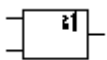


- лампа накаливания.

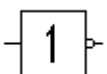
### Логические элементы - группа **LOGIC GATES**



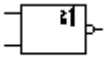
- логический элемент "И"



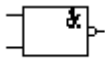
- логический элемент "ИЛИ"



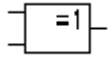
- логический элемент "НЕ"



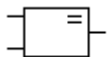
- логический элемент "ИЛИ-НЕ"



- логический элемент "И-НЕ"

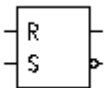


- логический элемент исключающее "ИЛИ"

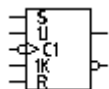


- логический элемент импликация

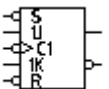
#### 4.6. Комбинированные цифровые компоненты.



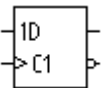
- асинхронный RS-триггер



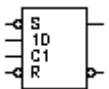
- универсальный JK-триггер с прямым тактовым входом и входами предустановки



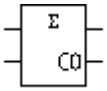
- универсальный JK-триггер с инверсным тактовым входом и инверсными входами предустановки



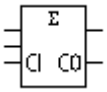
- D-триггер без предустановки



- D- со входами предустановки



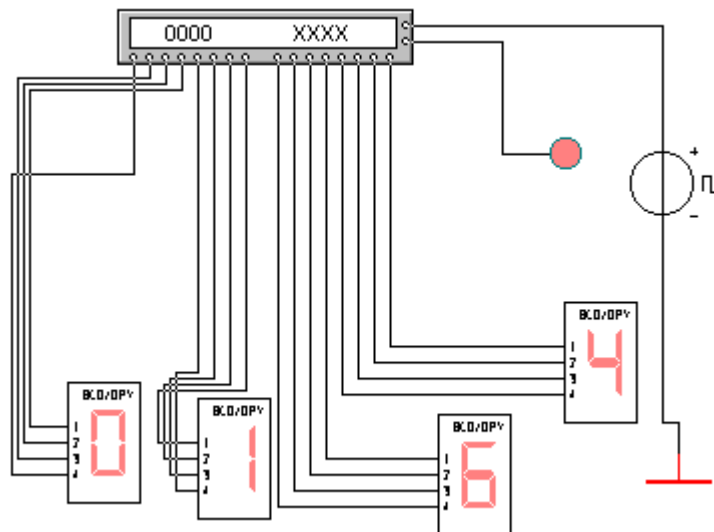
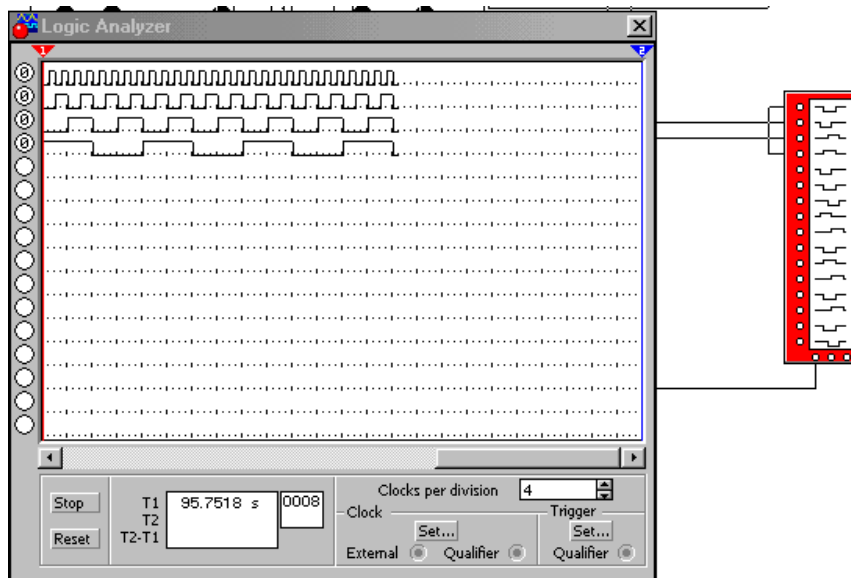
- полусумматор



- полный сумматор

Приборы, группа **INSTRUMENTS**.

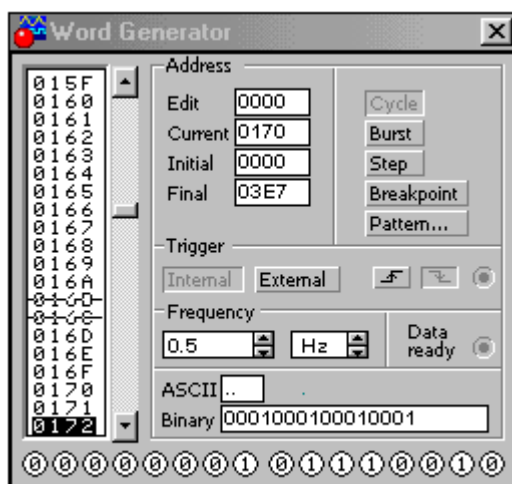
- логический анализатор



- генератор слова - **Word Generator**

На первом рисунке показан генератор слова с подключенными семисегментными индикаторами и внешним генератором синхроимпульсов.

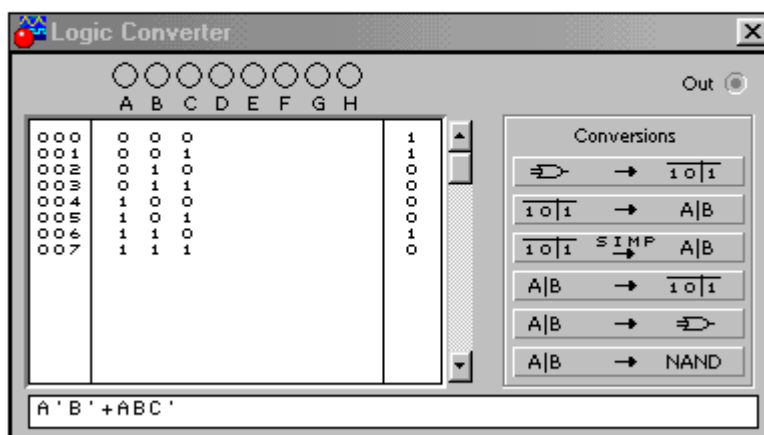
На втором рисунке генератор слова показан в развёрнутом виде. Генератор (или кодовый генератор) предназначен для генерации 16-ти 16-ти разрядных двоичных слов, которые набираются пользователем на экране, расположенным в левой части лицевой панели. Для набора двоичных комбинаций



необходимо щёлкнуть мышью на соответствующем разряде и затем ввести с клавиатуры число в десятичном коде.

Сформированные слова выдаются на шестнадцать расположенных в нижней части прибора выходных клемм-индикаторов:

- с индикацией в двоичном коде в строке окна binary;
- в пошаговом (step), циклическом (cycle) или с выбранного слова до конца (при нажатии кнопки BURST) при заданной частоте посылок (установка- заданием частоты в окнах FREQUENCY);
- при внутреннем или внешнем запуске ( при нажатии кнопки EXTERNAL, справа верхняя клемма служит для подключения сигнала синхронизации);
- при запуске по переднему или заднему фронту сигнала синхронизации служит кнопка
- на правую нижнюю клемму выдается выходной синхронизирующий импульс.



## Логический преобразователь- **Logic Converter**.

На лицевой панели преобразователя показаны клеммы-индикаторы входов А,В,.....Н и одного выхода OUT, экран для отображения таблицы истинности исследуемой схемы, экран-строка для отображения её булева выражения (в нижней части).

Логический анализ n-входного устройства с одним выходом может осуществлять следующие действия, используя кнопки управления:

1.



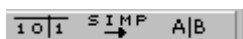
- таблицу истинности исследуемого устройства;

2.



- булево выражение, реализуемое устройством;

3.



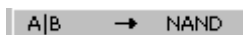
- минимизированное булево выражение;

4.



- схему устройства на логических элементах без ограничения их типа;

5.

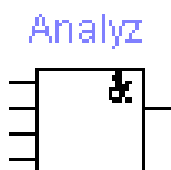


- схему устройства только на логических элементах И-НЕ.

## Пример составления исследуемой схемы.

Собрать схему логического элемента "И".

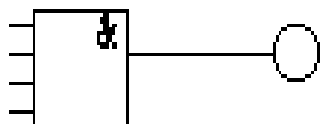
В группе Logic Gates, выбирается логический элемент "И".



Двумя щелчками мыши на изображении логического элемента переходим к настройкам параметров логического элемента "И". Выбираем количество входов, например 4.

Можно присвоить название логическому элементу.

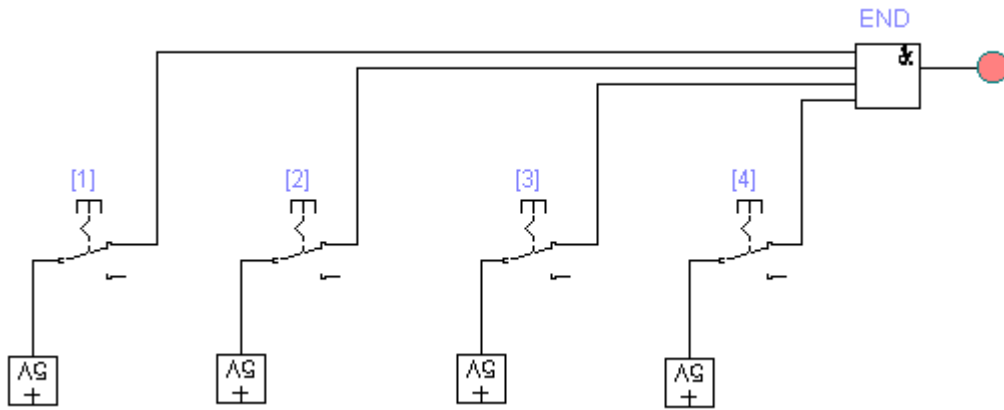
К выходу логического элемента присоединяем из группы **INDICATORS** красный светодиод.



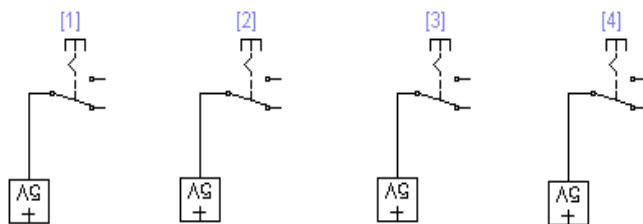
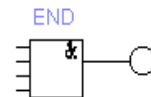
Для получения логического сигнала (0 или 1) удобно воспользоваться источником напряжения



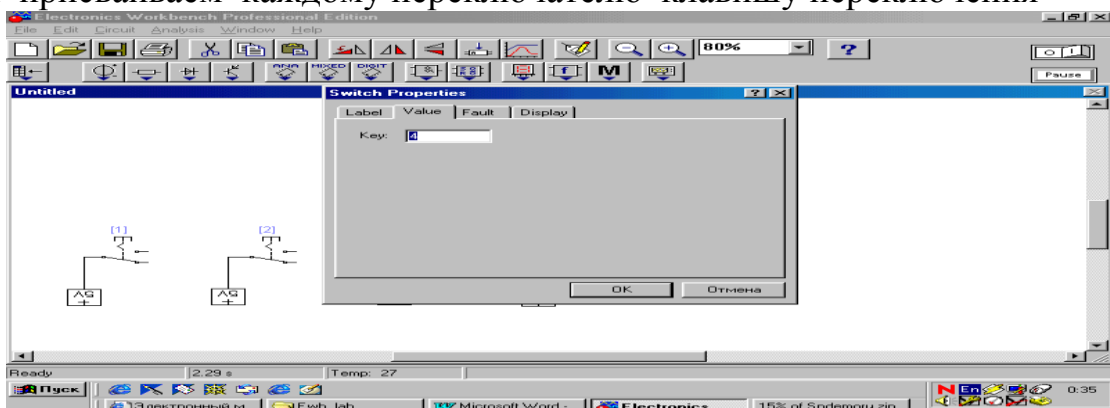
и переключателем



Затем набираем 4 источника и 4 переключателя



При этом присваиваем каждому переключателю клавишу переключения





Затем соединяем входы логической схемы "И" с каждым из переключателей.

Проверка состоит в подаче различных кодовых комбинаций на вход логической схемы.

На выходе логической схемы "И" появляется логическая 1 (горит светодиод) только при подаче логических 1 (потенциал 5 вольт) на все четыре входа логической схемы "И".

### 7. Список рекомендуемой литературы

1. Орлов С.А. Организация ЭВМ и систем: учебник для вузов / С. А. Орлов, Б. Я. Цилькер. - 3-е изд. - СПб. : ПИТЕР, 2014. - 688 с. : <https://ibooks.ru/reading.php?productid=340894>
2. Организация ЭВМ и систем: учебное пособие / Н. В. Замятин - 2018. 214 с.: Научно-образовательный портал ТУСУР, <https://edu.tusur.ru/publications/8499>
3. Колдаев В.Д. Архитектура ЭВМ: Учебное пособие / В.Д. Колдаев, С.А. Лупин. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 384 с. - <http://znanium.com/bookread2.php?book=424016>
4. [Замятин Н. В.](#) Нечеткая логика и нейронные сети: учебное пособие. - Томск: Эль Контент, 2014. - 146 с.
11. Таненбаум Э. Архитектура компьютера. 4-е изд. – СПб.: Питер, 2013.
5. Электронные ресурсы: Гуров В.В. Архитектура микропроцессоров. // Интернет Университет // <http://old.intuit.ru/department/hardware/microarch/11/> Ланина, дистанционный курс по организации ЭВМ. // <http://paralichka85.px6.ru/index.htm>