УДК 519.688

# Использование системы компьютерной алгебры Maple для построения логических элементов

*Халидова Оксана Халидшаховна*
*Ставропольский государственный педагогический институт*
*Студент*

*Оленев Александр Анатольевич*
*Ставропольский государственный педагогический институт*
*к. т. н. доцент, доцент кафедры математики и информатики.*

**Аннотация**
В данной работе рассматривается базовая библиотека логических элементов, которые выполняют различные операции в последовательности «от более простых к более сложным». Применение системы компьютерной алгебры (СКА) Maple целесообразно в случае изучения логических элементов с различными операциями, так как ставится задача рассмотреть математические модели вышеперечисленных элементов. Составлены программы в СКА Maple, которые определяют работу логических элементов.
**Ключевые слова:** логический элемент, логические операции, входные и выходные сигналы, логическая схема, таблица истинности.

# Using the Maple computer algebra system to build logical elements

*Khalidova Oksana*
*Stavropol State Pedagogical Institute*
*Student*

*Olenev Aleksandr*
*Stavropol State Pedagogical Institute*
*PhD, associate professor, associate professor at the Department of Mathematics and Computer Science*

**Abstract**
This paper discusses the basic library of logical elements that perform various operations in the sequence "from simpler to more complex." The use of the computer algebra system (SKA) Maple is advisable in the case of studying logical elements with various operations, since the task is to consider mathematical models of the above elements. Compiled programs in SKA Maple, which define the operation of logical elements.
**Keywords:** logical element, logical operations, input and output signals, logic circuit, truth table.

Математическое моделирование – это процесс построения математической модели объекта и исследование его свойств. Модель должна быть представлена в такой форме, которая будет удобна для применения, и она должна воспроизводить последовательность выполняемых логических операций, необходимых для нахождения необходимых значений [1]. Использование метода математического моделирования можно эффективно использовать при изучении свойств логических элементов. Такой подход позволяет обеспечить наглядность работы базовых логических элементов и выполнения основных законов алгебры логики, тем самым помогая понять порядок их работы, синтеза простейших цифровых устройств и достичь высоких результатов при обучении. В качестве инструмента для исследования работы логических элементов предлагается использовать систему компьютерной алгебры (СКА) Maple [2, 3], которая позволяет построить адекватную математическую модель и убедится в правильности работы созданных логических элементов, а также рассмотреть в дальнейшем эквивалентные схемы реализации на основе применения основных законов алгебры логики. Создаваемые модели должны позволять учитывать использование разного количества входных значений логической функции, а также выполнять автоматическое изменение выходных значений логических элементов [4].

Логические элементы или их набор (библиотека) помогают осуществить реализацию простого или сложного логического высказывания и показывают взаимосвязь между входными и выходными значениями сигналов. Результат выходного значения зависит как от входных значений логической функции, так и от их количества. Элементы, реализующие то или иное логическое высказывание, используются, как правило, при построении логических схем ЭВМ и других цифровых устройств.

Такая реализация стала возможной при использовании следующего допущения: замена истинного значения («True») на значение «1», а ложного («False») — нулем «0», то есть формальную логику можно представить, как правила выполнения логических операций с нулями и единицами, а выполнение операций в ПЭВМ свести к выполнению логических операций с двоичными кодами.

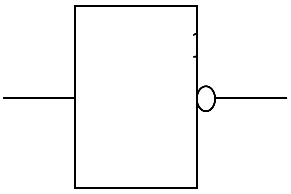
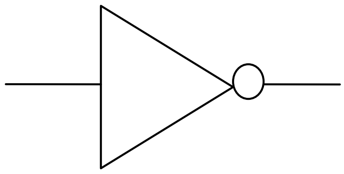
Чаще всего используются логические элементы, реализующие простые логические высказывания: конъюнкция (И), дизъюнкция (ИЛИ), отрицание (НЕ). Это связано в первую очередь с возможностью выражения любого сложного логического высказывания через простые высказывания. Рассмотрим возможные математические модели основных логических элементов.

**Логический элемент «НЕ»**

Логический элемент «НЕ» может иметь только один вход и один выход. Выход в таком случае называют инверсным. Возможные вариации названий: НЕ, отрицание, NOT. это переход. Данный логический элемент реализует переход от одного значения к другому, от 1 к 0, от True к False или наоборот.

Условно-графические обозначения (УГО), представляющие логическое отрицание представлены в таблице 1.

Таблица 1. УГО логического элемента «НЕ».

| Международная электротехническая комиссия (IEC) | Американский национальный институт стандартов (ANSI) |
|---|---|
|  НЕ |  |

Процедура для моделирования логического элемента - инвертор, с использованием СКА Maple, выглядит так [5]:

**>with(Logic): # подключение библиотеки логика**

**>элемент_НЕ := proc(a)# процедура для моделирования элемента «НЕ»**

**return &not(a); # возвращение результата**

**end proc: # завершение процедуры**

В качестве проверки правильности работы разработанной процедуры построим таблицу истинности для инвертора с использованием функции СКА Maple - TruthTable:

**>T1:=элемент_НЕ(x); # моделирование элемента НЕ**

$$T1 := \&not(x)$$

**> T11:=TruthTable(T1,[x]); # построение таблицы истинности инвертора**

$$T11 := table([true = false, false = true])$$

**> T11[true]; # проверка работы процедуры при входном истинном значении**

$$false$$

Для полного соответствия с работой логического элемента - инвертор, применяемого в ЭВМ, воспользуемся представлением в форме MOD2. Это позволяет получать выходное значение, зависящее от логической функции выполняемой элемент, в виде значений «0» или «1».

**> T12:= TruthTable(T1,[x],form = MOD2); # построение таблицы истинности инвертора в форме MOD2**

$$T12 := table([0 = 1, 1 = 0])$$

**> T12[0]; # проверка моделирования работы инвертора в форме MOD2**

$$1$$

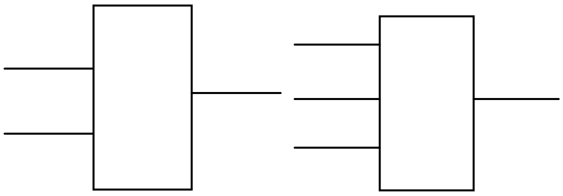
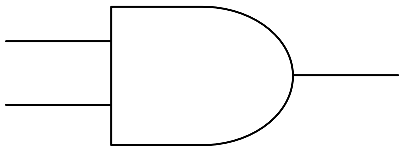
Анализ выходных значений таблицы истинности показывает: полученные результаты моделирования совпадают с теоретическими результатами.

**Логический элемент «И»**

Возможные вариации названий: конъюнкция, логическое умножение, AND. Логический элемент с операцией конъюнкция должен иметь не менее 2 входов и 1 выход. Условно-графические обозначения логического умножения «И» представлены в таблице 2.

Принцип работы простейших логических элементов легче понять и проверить, как сказано выше, используя в качестве регистратора взаимосвязи входных и выходных значений - таблицу истинности. Таблица истинности для логического элемента «И» должна подтверждать следующее утверждение: выходное значение будет истинным (верным) в том и только в том случае, когда все высказывания, истинны одновременно [5].

Таблица 2. УГО логического элемента «И».

| Международная электротехническая комиссия (IEC) | Американский национальный институт стандартов (ANSI) |
|---|---|
|  2И 3И |  |

Разработка процедуры для моделирования логического элемента выполняющего операцию логического умножения должна учитывать возможность построения элемента с количеством входов более двух.

Процедуру для моделирования элемента «И» с использованием СКА Maple, можно представить следующим образом [5]:

**>with(Logic):# подключение библиотеки логика**

**>элемент_И := proc(a::seq(anything))# объявление процедуры для элемента «И»**

**local result, i; # объявление переменных**

**if nops([a]) < 2 then # если элемент будет иметь меньше двух входов, то …**

**error "Элемент И должен иметь не менее двух входов."; #вывод предупреждения в случае неправильного выбора количества входов**

**end if;# завершение команды «если»**

**result := a[1] &and a[2]; # присвоение команды переменной «результат»**

**fori from 3 to nops([a]) do #для 3 входов**

**result := result &and a[i]; # присвоение команды переменной «результат» для более чем 2-х входового логического элемента**

**end do;**

**return result; # возвращение результата**

**end proc: # завершение процедуры**

В качестве проверки правильности разработанной процедуры используем построение таблицы истинности. Таблицу истинности построим, используя возможности системы компьютерной алгебры (СКА) Maple, для этого воспользуемся командой библиотеки Logic - TruthTable.

**> T1:=элемент_И(x,y);# моделирование логического элемента «И» с двумя входами**

*T1* := *x* &and *y*

**>T11:=TruthTable(T1,[x,y]); # построение таблицы истинности для логического элемента «И»**

*T11* := *table*([(*false*, *false*) = *false*, (*false*, *true*) = *false*, (*true*, *true*) = *true*, (*true*, *false*) = *false*])

При попытке построения логического элемента «И» с одним входом, созданная процедура выдает ошибку:

**>T1:=элемент_И(x);# ошибка при построении одновходового элемента «И»**

Error, (in элемент_И) Элемент И должен иметь не менее двух входов.

Построение элемента «И» на три входа:

**>T1:=элемент_И(x, y, z); моделирование логического элемента «И» с тремя входами**

*T1* := (*x* &and *y*) &and *z*

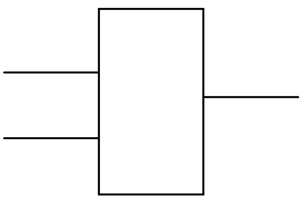
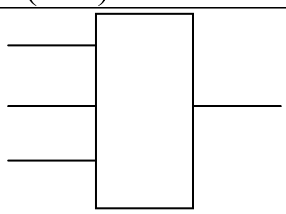
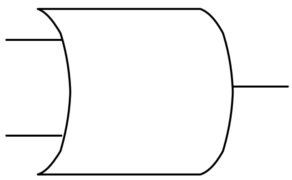
Анализ выходных значений таблицы истинности показывает: полученные результаты моделирования совпадают с теоретическими результатами.

**Логический элемент «ИЛИ»**

Возможные вариации названий: логическое сложение, дизъюнкция, OR.

Условно-графические обозначения логического сложения представлены в таблице 3.

Таблица 3. УГО логического элемента «ИЛИ».

| Международная электротехническая комиссия (IEC) | Американский национальный институт стандартов (ANSI) |
| --- | --- |
|  2ИЛИ  3ИЛИ |  |

Дизъюнктор можно представить с 2, 3, 4, 5, 6, 7, 8 входами и 1 выходом.

Процедура для математического моделирования логического элемента «ИЛИ» с операцией дизъюнкция, подобна процедуре логического элемента конъюнкции.

```
>with(Logic): # подключение библиотеки логика
>элемент_ИЛИ := proc(a::seq(anything)) #объявление процедуры для элемента «И»
local result, i;
if nops([a]) < 2 then
error "Элемент И должен иметь не менее двух входов."; #вывод предупреждения в случае неправильного выбора количества входов
end if;
result := a[1] &or a[2];
fori from 3 to nops([a]) do
result := result &or a[i];
end do;
return result; # возвращение результата
end proc: # завершение процедуры
```

Полученная таблица истинности для элемента «ИЛИ» с двумя входными сигналами должна показывать истинное значение тогда, когда истинно хотя бы одно из входящих в него высказываний. Используя систему компьютерной алгебры Maple, докажем это:

```
> T1:=элемент_ИЛИ(x,y,z); # моделирование логического элемента «ИЛИ» с тремя входами
```

*T1* := (*x* &or *y*) &or *z*

```
> T11:=TruthTable(T1,[x,y,z]); # построение таблицы истинности для логического элемента «ИЛИ» на три входа
```

*T11* := *table*([(*true*, *false*, *false*) = *true*, (*true*, *true*, *true*) = *true*, (*false*, *false*, *true*) = *true*, (*false*, *false*, *false*) = *false*, (*true*, *false*, *true*) = *true*, (*false*, *true*, *true*) = *true*, (*false*, *true*, *false*) = *true*, (*true*, *true*, *false*) = *true*])

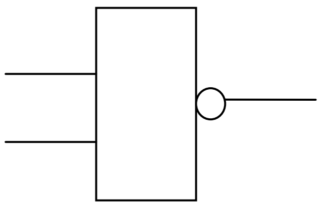
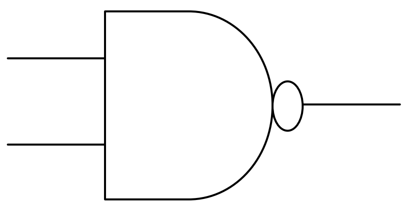
Анализ выходных значений таблицы истинности показывает: полученные результаты моделирования совпадают с теоретическими результатами.

**Логический элемент Штрих Шеффера**

Возможные вариации названий: логическое умножение с отрицанием, И-НЕ, NAND.

Условно-графические обозначения логического умножения с отрицанием представлены в таблице 4.

Таблица 4. УГО логического элемента «И-НЕ».

| Международная электротехническая комиссия (IEC) | Американский национальный институт стандартов (ANSI) |
|---|---|
| <br>2И-НЕ |  |

Такой логический элемент сначала выполняет операцию конъюнкции, а затем логическое отрицание.

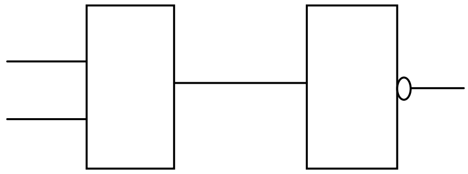


Рисунок 1. Последовательное соединение элемента «И» и элемента «НЕ».

Таблица истинности для выполнения операции конъюнкции с последующим инвертированием на два входа инверсна (противоположна) таблице для элемента «И».

> **G1:=элемент_И(a,b);# моделирование логического элемента «И» на два входа**

$$G1 := a \ \&\text{and} \ b$$

> **TruthTable(G1,[a,b]); # построение таблицы истинности для логического элемента «И» на два входа**

table([(*true*, *true*) = *true*, (*false*, *true*) = *false*, (*false*, *false*) = *false*,
(*true*, *false*) = *false*
])

> **G2:=элемент_НЕ(G1); моделирование логического элемента «НЕ»(«И») на два входа**

$$G2 := \&\text{not}(a \ \&\text{and} \ b)$$

> **TruthTable(G2,[a,b]); # построение таблицы истинности для логического элемента «НЕ»(«И») на два входа #(1)**

table([(*true*, *true*) = *false*, (*false*, *true*) = *true*, (*false*, *false*) = *true*,
(*true*, *false*) = *true*
])

> **TruthTable(G2,[a,b],form = MOD2); # построение таблицы истинности для логического элемента «НЕ»(«И») на два входа в форме MOD2 #(2)**

table([(0, 1) = 1, (0, 0) = 1, (1, 0) = 1, (1, 1) = 0])

Для создания общей процедуры математического моделирования работы логического элемента «2И-НЕ» с помощью СКА Maple воспользуемся одним из основных законов алгебры логики - законом де Моргана.

$$\overline{(a \wedge b)} = \bar{a} \vee \bar{b}$$

Описать логический элемент «3И-НЕ» можно по тому же принципу:

$$\overline{(a \wedge b \wedge c)} = \bar{a} \vee \bar{b} \vee \bar{c}$$

Тогда для построения многовходового логического элемента, выполняющего операцию операцией «И-НЕ», процедура будет выглядеть следующим образом:

**>with(Logic): # подключение библиотеки логика**

**>Элемент_И_НЕ:= proc(a::seq(anything))**

```
local result, i;
ifnops([a]) < 2 then
error "Элемент И-НЕ должен иметь не менее двух входов.";
end if;
result := &not a[1] &or (&not(a[2]));
fori from 3 to nops([a]) do
result := result &or (&not(a[i]));
end do;
return result;
end proc:
```

Для проверки работоспособности построим таблицу истинности:

**> T1:=Элемент_И_НЕ(x,y); моделирование логического элемента «И-НЕ» на два входа с использованием законов алгебры логики**

$$T1 := \&not(x)\ \&or\ \&not(y)$$

**> T11:=TruthTable(T1,[x,y]); # построение таблицы истинности для логического элемента «И-НЕ» на два входа**

$$T11 := table([(false, true) = true, (true, true) = false, (false, false) = true, (true, false) = true])$$

Анализ выходных значений таблицы истинности показывает: полученные результаты моделирования совпадают с теоретическими результатами, и итогами, полученными в ходе раздельного моделирования.

**Логический элемент Стрелка Пирса**

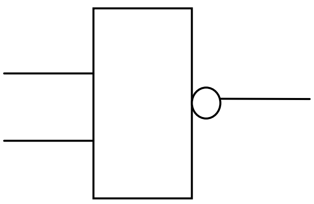
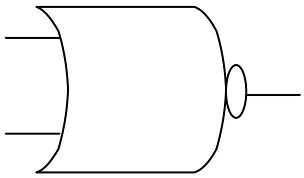
Возможные вариации названий: логическое сложение с отрицанием, NOR.

Логический элемент с операцией «ИЛИ-НЕ» выполняет сначала операцию логического сложения, затем логическое отрицание.

Условно-графическое обозначение для логического элемента «ИЛИ-НЕ» представлены в таблице 5.

Таблица истинности для элемента с двумя входами и выходом должна показывать истинное значение только в одном случае - на все входы логического элемента подаются ложные значения.

Таблица 5. УГО логического элемента «ИЛИ-НЕ».

| Международная электротехническая комиссия (IEC) | Американский национальный институт стандартов (ANSI) |
|---|---|
|  2ИЛИ-НЕ |  |

Покажем, что это именно так, применяя СКА Maple.

Раздельное моделирование: представление схемы в виде последовательного соединения логического элемента «ИЛИ» и элемента «НЕ»:

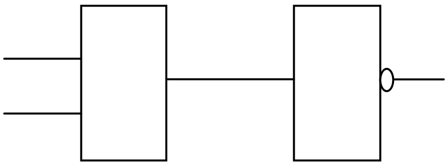


Рисунок 2. Последовательное соединение элемента «ИЛИ» и элемента «НЕ».

> **G1:=элемент_ИЛИ(a,b,c);# моделирование логического элемента «ИЛИ» на три входа**

$$G1 := (a \ \&\text{or} \ b) \ \&\text{or} \ c$$

> **TruthTable(G1,[a,b,c]); # построение таблицы истинности для логического элемента «ИЛИ» на три входа**

table([(*false*, *true*, *false*) = *true*, (*false*, *true*, *true*) = *true*, (*false*, *false*, *true*) = *true*,
(*true*, *true*, *true*) = *true*, (*true*, *true*, *false*) = *true*, (*true*, *false*, *true*) = *true*,
(*true*, *false*, *false*) = *true*,
(*false*, *false*, *false*) = *false*
])

> **G2:=элемент_НЕ(G1); # моделирование логического элемента «НЕ»(«ИЛИ») на три входа**

$$G2 := \&\text{not}((a \ \&\text{or} \ b) \ \&\text{or} \ c)$$

> **TruthTable(G2,[a,b,c]); # построение таблицы истинности для логического элемента «НЕ»(«ИЛИ») на три входа #(3)**

table([(*false*, *true*, *false*) = *false*, (*false*, *true*, *true*) = *false*, (*false*, *false*, *true*) = *false*,
(*true*, *true*, *true*) = *false*, (*true*, *true*, *false*) = *false*, (*true*, *false*, *true*) = *false*,
(*true*, *false*, *false*) = *false*,
(*false*, *false*, *false*) = *true*
])

> **TruthTable(G2,[a,b,c],form = MOD2); # построение таблицы истинности для логического элемента «НЕ»(«ИЛИ») на три входа в форме MOD2 #(4)**

table([(0, 1, 0) = 0, (0, 1, 1) = 0, (1, 0, 0) = 0, (1, 0, 1) = 0, (1, 1, 0) = 0, (0, 0, 1) = 0,
(1, 1, 1) = 0,
(0, 0, 0) = 1
])

Для создания общей процедуры математического моделирования работы логического элемента «2ИЛИ-НЕ» с помощью СКА Maple воспользуемся одним из основных законов алгебры логики - законом де Моргана.

$$\overline{(a \vee b)} = \overline{a} \wedge \overline{b}$$

Описать логический элемент «3ИЛИ-НЕ» и с большим количеством входов, можно по тому же принципу:

$$\overline{(a \lor b \lor c)} = \overline{a} \land \overline{b} \land \overline{c}$$

Процедура для построения многовходового логического элемента, выполняющего операцию операцией «ИЛИ-НЕ», будет выглядеть следующим образом:

```
>with(Logic):
>Элемент_ИЛИ_НЕ:= proc(a::seq(anything))
local result, i;
ifnops([a]) < 2 then
error "Элемент ИЛИ-НЕ должен иметь не менее двух входов.";
end if;
result := a[1] &nor a[2];
fori from 3 to nops([a]) do
result := result &and (&not(a[i]));
end do;
return result;
endproc:
```

Для проверки правильности разработанной процедуры построим таблицу истинности:

**> T1:=Элемент_ИЛИ_НЕ (x,r,t); # моделирование логического элемента «ИЛИ-НЕ» на три входа**

$$T1 := (x \text{ \&nor } r) \text{ \&and } \text{\&not}(t)$$

**>TruthTable(T1,[x,r,t]); # построение таблицы истинности для логического элемента «ИЛИ-НЕ» на три входа**

$$table([(false, true, true) = false, (true, true, true) = false, (true, false, false) = false, (false, false, true) = false, (false, false, false) = true, (true, true, false) = false, (false, true, false) = false, (true, false, true) = false])$$

Анализ выходных значений таблицы истинности показывает: полученные результаты моделирования совпадают с теоретическими результатами и результатами, полученными в ходе раздельного моделирования**,** что доказывает правильность и работоспособность разработанных процедур.

**Логический элемент Исключающее ИЛИ**.

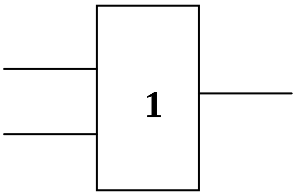
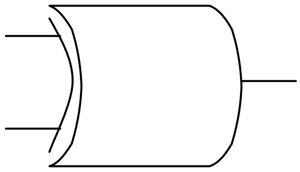
Возможные вариации названий: сложение по модулю 2, XOR, исключающее ИЛИ, MOD 2.

Условно графическое обозначение логического элемента «исключающее ИЛИ» представлены в таблице 6.

Сложение по модулю 2 – логический элемент, который имеет не менее двух входов и один выход.

Таблица истинности должна показывать результат — истина, в том и только в том случае, когда два значения не равны. Построим таблицу истинности с использованием системы компьютерной алгебры Maple.

Таблица 6. УГО логического элемента «исключающее ИЛИ».

| Международная электротехническая комиссия (IEC) | Американский национальный институт стандартов (ANSI) |
|---|---|
|  исключающее ИЛИ |  |

Процедура моделирования логического элемента выполняющего операцию «исключающее ИЛИ» выглядит так:

```
>with(Logic):
>Элемент_MOD2 := proc(a::seq(anything))
local result, i;
ifnops([a]) < 2 then
error "Элемент MOD2 должен иметь не менее двух входов.";
endif;
result := a[1] &xor a[2];
fori from 3 to nops([a]) do
result := result &xor a[i];
end do;
return result;
endproc:
```

Для проверки построим таблицу истинности логического элемента с операцией «исключающее ИЛИ»:

**> Элемент_MOD2(x,y,z,w); # моделирование логического элемента «MOD2» на четыре входа**

$$((x \,\&\mathrm{xor}\, y) \,\&\mathrm{xor}\, z) \,\&\mathrm{xor}\, w$$

**>TruthTable(%,[x,y,z,w]); # построение таблицы истинности для логического элемента «MOD2» на четыре входа**

$$table([(true, true, false, true) = true, (false, true, false, false) = true, (false, false, true, true) = false, (true, true, true, false) = true, (false, false, false, true) = true, (false, false, false, false) = false, (true, true, false, false) = false, (true, false, true, true) = true, (false, true, true, false) = false, (false, true, false, true) = false, (true, false, true, false) = false, (true, true, true, true) = false, (true, false, false, true) = false, (false, false, true, false) = true, (false, true, true, true) = true, (true, false, false, false) = true])$$

Анализ выходных значений таблицы истинности показывает: полученные результаты моделирования совпадают с теоретическими результатами.

Современные цифровые устройства состоят из интегральных микросхем, на которые возложено выполнение определенных, в том числе и логических функций. Для выполнения одной сложной логической функции требуется построение нескольких простых и рассмотрение различных вариантов построения логических схем является актуальной задачей, требующей правильного понимания порядка функционирования простейших логических устройств [6, 7].

Все вышеприведенные модели логических элементов можно использовать при моделировании построения цифровых устройств – шифраторов, сумматоров, мультиплексоров, дешифраторов и так далее. Номенклатура разработанной библиотеки позволяет производить построение и проверку правильности их реализации при использовании различных вариантов построения, а также при изучении и доказательстве основных законов алгебры логики.

**Библиографический список**

1. Игнатьев Ю.Г. Математическое моделирование фундаментальных объектов и явлений в системе компьютерной математики Maple. Лекции для школы по математическому моделированию. Казань: Казанский университет, 2014. 298 с.
2. Васильев А. Н. Maple 8. Самоучитель, М.: Диалектика, Вильямс, 2003. 352 с.
3. Говорухин В.Н., Цибулин В.Г. Введение в Maple. Математический пакет для всех. Мир. 1997. 208 с.
4. Оленев А.А., Малиатаки В.В. Моделирование логических элементов и простейших узлов ЭВМ в системе компьютерной математики Maple // Информатика в школе. 2017. №8(131). С. 58-63.
5. Оленев А.А., Малиатаки В.В. Логические элементы и схемы в СКА Maple. // Современные технологии в нефтегазовом деле. 2017. Уфа: Изд-во УГНТУ. 2017. С. 280-282.
6. Красильников В.В., Оленев А. А., Тоискин В.С., Тынчеров К.Т. Использование системы компьютерной алгебры Maple в булевой алгебре. // Актуальные вопросы инженерного образования -2016. Сборник научных трудов международной научно-методической конференции, посвященной 60-летию филиала УГНТУ в г. Октябрьском. Изд-во УГНТУ. 2016. С.303-310.
7. Красильников В.В., Оленев А. А., Тоискин В.С., Тынчеров К.Т. Использование системы компьютерной алгебры Maple при изучении дискретной математики. // Актуальные вопросы инженерного образования -2016. Сборник научных трудов международной научно-методической конференции, посвященной 60-летию филиала УГНТУ в г. Октябрьском. Изд-во УГНТУ. 2016. С.303-310.