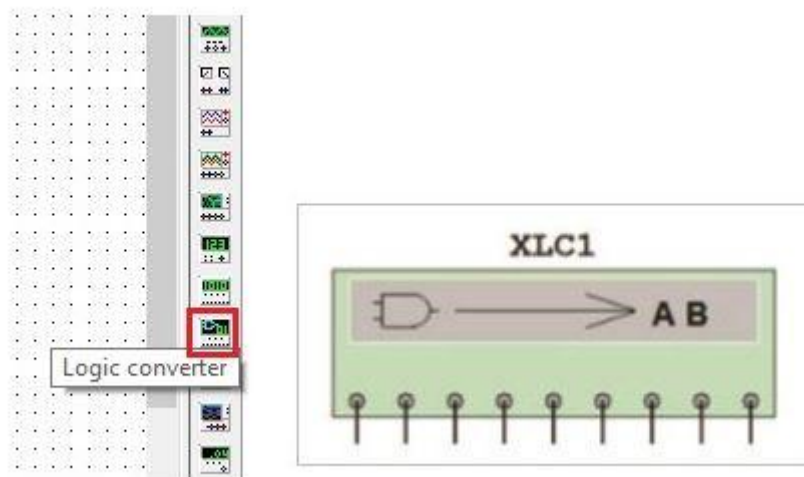


# Руководство к лабораторному практикуму: Введение в цифровую электронику

С использованием платы Digilent Digital Electronics для NI ELVIS III



Лабораторная работа 3:  
Логические элементы в деталях  
и булева алгебра

© 2018 National Instruments

All rights reserved. Данный ресурс и любые его части не могут быть скопированы или в любой форме воспроизведены иным способом без письменного разрешения издателя.

National Instruments относится с уважением к чужой интеллектуальной собственности и призывает к этому же своих читателей. Данный ресурс защищен законами об охране авторских прав и прав на интеллектуальную собственность. Вы имеете право передавать программное обеспечение и прочие материалы, разработанные с помощью описанного в данном ресурсе программного обеспечения, третьим лицам в соответствии с условиями приобретенной вами лицензии и другими законодательными ограничениями.

LabVIEW и National Instruments являются торговыми марками National Instruments.

Названия других упомянутых торговых марок и изделий являются собственностью их правообладателей.

**Дополнительные ограничения ответственности:** Читатель принимает все риски от использования данного ресурса и всей информации, теорий и программ, содержащихся или описанных в нем. Данный ресурс может содержать технические неточности, типографические ошибки, прочие ошибки и упущения, и устаревшую информацию. Ни автор, ни издатель не несут ответственности за любые ошибки или неточности, за обновление любой информации и за любые нарушения патентного права и прочих прав на интеллектуальную собственность.

Ни автор, ни издатель не дают никаких гарантий, включая, но не ограничиваясь, любую гарантию на достаточность ресурса и любой информации, теорий или программ, содержащихся или описанных в нем, и любую гарантию, что использование любой информации, теорий или программ, содержащихся или описанных в ресурсе, не нарушит любое патентное право или иное право на интеллектуальную собственность. РЕСУРС ПОСТАВЛЯЕТСЯ "КАК ЕСТЬ". ИЗДАТЕЛЬ ЗАЯВЛЯЕТ ОБ ОТКАЗЕ ОТ ЛЮБЫХ ГАРАНТИЙ, ЯВНО ВЫРАЖЕННЫЕ ИЛИ ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ, ЛЮБЫЕ ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ ТОВАРНОГО СОСТОЯНИЯ, ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ ЦЕЛИ И НЕНАРУШЕНИЯ ПРАВ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ.

Издатель или автор не предоставляют прав или лицензий под любым патентным правом или иным правом на интеллектуальную собственность прямо, косвенно или лишением права на возражение.

НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ ИЗДАТЕЛЬ ИЛИ АВТОР НЕ НЕСУТ ОТВЕТСТВЕННОСТИ ЗА ПРЯМЫЕ, КОСВЕННЫЕ, ОСОБЫЕ, СЛУЧАЙНЫЕ, ЭКОНОМИЧЕСКИЕ ИЛИ ВТОРИЧНЫЕ УБЫТКИ, ПОНЕСЕННЫЕ ИЗ-ЗА ИСПОЛЬЗОВАНИЕ ЭТОГО РЕСУРСА ИЛИ ЛЮБОЙ ИНФОРМАЦИИ, ТЕОРИЙ ИЛИ ПРОГРАММ, СОДЕРЖАЩИХСЯ ИЛИ ОПИСАННЫХ В НЕМ, ДАЖЕ БУДУЧИ ПРЕДУПРЕЖДЕННЫ О ВОЗМОЖНОСТИ ПОДОБНЫХ УБЫТКОВ, И ДАЖЕ ЕСЛИ УБЫТКИ ВЫЗВАНЫ НЕБРЕЖНОСТЬЮ ИЗДАТЕЛЯ, АВТОРА ИЛИ ИНЫХ ЛИЦ, Применимое законодательство может не разрешить исключение или ограничение случайных или косвенных убытков, поэтому приведенные выше ограничения или исключения могут вас не касаться.

## Лабораторная работа 3: Логические элементы в деталях и булева алгебра

В предыдущей лабораторной работе мы подробно изучили два базовых логических элемента – AND (И) и OR (ИЛИ). Основываясь на этом, мы можем создать ряд других логических элементов. Это: **NOT (НЕ)**, **NAND (И- НЕ)**, **NOR (ИЛИ- НЕ)**, **XOR (исключающее ИЛИ)** и **XNOR (исключающее ИЛИ- НЕ)**. Рассмотрим каждый из них более подробно.

### Цель работы

В этой лабораторной работе студенты должны:

1. Исследовать функции, реализуемые различными логическими элементами
2. Спроектировать теоретически и практически реализовать схемы на различных логических элементах.
3. Рассчитать и реализовать комбинационные логические схемы на основе выражений в ДНФ и КНФ, полученных из таблиц истинности.

## Необходимые инструментальные средства и технологии

Платформа: NI ELVIS III

- ✓ Руководство пользователя:  
<http://www.ni.com/en-us/support/model.ni-elvis-iii.html>
- ✓ Учебные пособия:  
[https://www.youtube.com/playlist?list=PLvcPluVaUMIWm8ziaSxv0gwtshBA2dh\\_M](https://www.youtube.com/playlist?list=PLvcPluVaUMIWm8ziaSxv0gwtshBA2dh_M)

---

Технические средства:  
плата Digilent Digital Electronics Board  
for NI ELVIS III

- ✓ Руководство пользователя NI  
Digital Electronics Board:  
<http://www.ni.com/pdf/manuals/376627b.pdf>

---

Программное обеспечение:  
NI Multisim 14.0.1 Education Version или  
выше

- ✓ Установка Multisim:  
[http://www.ni.com/gate/gb/GB\\_ACAD\\_EMICEVALMULTISIM/US](http://www.ni.com/gate/gb/GB_ACAD_EMICEVALMULTISIM/US)
- ✓ Справка Multisim Help:  
<http://www.ni.com/multisim/technical-resources/>

---

Программное обеспечение:  
NI LabVIEW FPGA Vivado 2014.4

- ✓ Установка:  
<http://www.ni.com/download/labview-fpga-module-2015-sp1/5920/en/>

**Примечание:** драйвер Digilent (Инсталлятор, указанный выше, автоматически загружает на ваш компьютер инсталлятор, указанный ниже)

- ✓ Перейдите по ссылке:  
C:\NIFPGA\programs\Vivado2014\_4\data\xicom\cable\_drivers\nt64\digilent
- ✓ Установите: install\_digilent.exe

## Ожидаемые результаты

В этой лабораторной работе вы должны собрать для отчета:

- Логические выражения в виде дизъюнктивных и конъюнктивных нормальных форм (ДНФ и КНФ)
- Скриншоты, рисунки или эскизы схем
- Таблицы истинности
- Сохраненные схемы
- Ответы на вопросы из раздела *Заключение*

Преподавателю, скорее всего, необходимо предъявить полный отчет о работе. Узнайте у вашего преподавателя, есть ли конкретные требования к отчету или шаблон для его оформления.

## 1.1 Сведения из теории

### Boolean Algebra

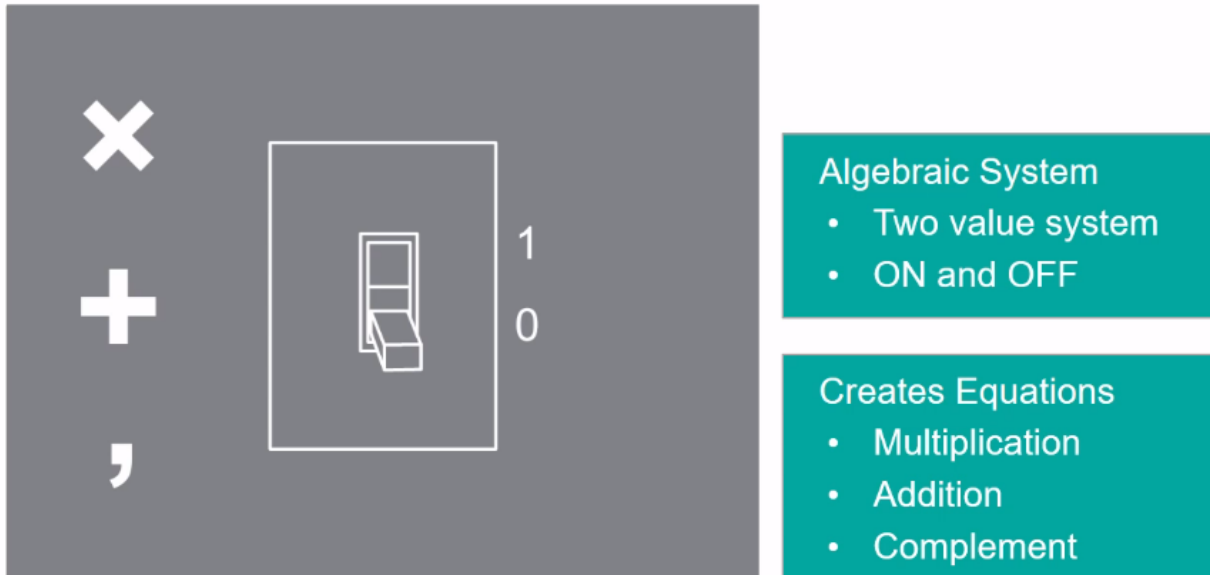


Рисунок 1-1 Видео. Просмотр видео здесь: <https://youtu.be/1wnztS6et0w>



#### Краткое содержание видео

- У логических элементов NAND, NOR, XOR и XNOR как минимум два входа, один выход, и их функция определяется уникальной таблицей истинности.
- Выходная переменная элемента NOT является инверсией входной переменной.
- Булева алгебра - это система, где два значения используются для представления свойств электрических схем с двумя устойчивыми состояниями – ВКЛ и ВЫКЛ.

#### Инверторы

- Инверторы также известны как логические элементы *NOT (НЕ)*
- У них только один вход и один выход.
- Таблица истинности инвертора проста: выходное значение всегда *противоположно* входному.
- Например, если на входе 1, на выходе будет 0, и наоборот. Графически инвертор отображается кружком на входе и/или выходе логического элемента.
- В нашем случае кружок расположен на выходе, то есть этот выход инвертируется. Если бы кружок располагался на входе, то инвертировался бы вход.

- Схемы с более чем одним входом могут использовать логические элементы NAND (И-НЕ) или NOR (ИЛИ-НЕ), которые мы рассмотрим далее.

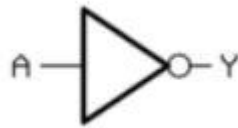


Рисунок 1-2. Инвертор

## Логические элементы NAND (И-НЕ)

- Логические элементы *NAND* инвертируют выход элемента AND.
- Входные значения остаются такими же, как в таблице истинности элемента AND, а выходное значение меняется на противоположное.
- Если любой из входов равен 0, на выходе будет 1.
- Ниже приведены таблица истинности и условное графическое обозначение.

A	B	O
0	0	1
0	1	1
1	0	1
1	1	0



Рисунок 1-3 Таблица истинности и условное графическое обозначение элемента NAND

## Логические элементы NOR (ИЛИ-НЕ)

- Логический элемент *NOR* инвертирует выход элемента OR.
- Уровни входных сигналов остаются такими же, как в таблице истинности элемента OR, а уровень выходного сигнала меняется.
- В общем, если уровень любого из входных сигналов равен 1, уровень выходного сигнала будет равен 0.
- Ниже приведены таблица истинности и условное графическое обозначение.

A	B	O
0	0	1
0	1	0
1	0	0
1	1	0



Рисунок 1-4 Таблица истинности и условное графическое обозначение элемента NOR

## Логические элементы XOR (исключающее ИЛИ)

- Элемент *XOR* также известен как исключающий элемент OR.
- Уровень выходного сигнала будет равен 1, если уровень только одного входного сигнала равен 1. Если на обоих входах 0 или на обоих входах 1, уровень выходного сигнала будет равен 0.
- Ниже приведены таблица истинности и условное графическое обозначение.



A	B	O
0	0	0
0	1	1
1	0	1
1	1	0



Рисунок 1-5 Таблица истинности и условное графическое обозначение элемента XOR

## Логические элементы XNOR (исключающее ИЛИ-НЕ)

- Логический элемент XNOR выполняет действие, противоположное элементу XOR.
- На выходе будет 1, если уровни входных сигналов одинаковы, и 0, если уровни входных сигналов различаются.
- Таблица истинности приведена ниже.

A	B	O
0	0	1
0	1	0
1	0	0
1	1	1

Рисунок 1-6 Таблица истинности элемента XNOR

## Комбинационные логические схемы

Комбинационные логические схемы – это класс схем, выходной сигнал которых зависит только от текущего значения входных сигналов. Комбинационные логические схемы реализуются на логических элементах. Используя различные комбинации логических элементов, можно реализовать различные функции. Для определения функции комбинационных схем часто используется таблица истинности, но иногда, наоборот, комбинационная схема разрабатывается по имеющейся таблице истинности.

## Булева алгебра

*Булева алгебра* – это алгебраическая система, в которой два значения используются для представления свойств электрических схем с двумя устойчивыми состояниями – ВКЛ (ON) и ВЫКЛ (OFF), или 1 и 0. Правила выполнения двух бинарных операций – сложения, умножения и дополнения (') для алгебраических выражений с двумя булевыми переменными приведены в таблицах ниже.

- Можно заметить, что двоичное сложение, умножение и дополнение совпадают с логическими операциями OR, AND и NOT.
- Для дополнения используются несколько условных обозначений: апостроф после переменной, восклицательный знак, тильда или слово NOT (НЕ) перед переменной или чертой над ней.
- Поскольку булева алгебра работает с цифровыми системами, оперирующими только значениями 0 и 1, она называется просто "двоичной логикой".
- Любая логическая функция, какой бы сложной она не была, может быть реализована с использованием только трех базовых логических операций.
- Функция, представленная таблицей истинности, может быть выражена разными способами.
- Зная логическое выражение и функцию, можно реализовать схему.

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Рисунок 1-7 Двоичное умножение (логическая операция AND)

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

Рисунок 1-8 Двоичное сложение (логическая операция OR)

x	x'
0	1
1	0

Рисунок 1-9 Дополнение (логическая операция NOT)

## Дизъюнктивная нормальная форма

Простой метод преобразования таблицы истинности в комбинационную логическую схему – использовать стандартную форму булевого выражения под названием *сумма произведений* (*дизъюнктивная нормальная форма, ДНФ*).

- Сумма произведений – это в буквальном смысле сумма булевских конъюнкций (термов) называемых минитермами.
- Минитерм – это произведение булевых переменных, равное 1.
- Пример ДНФ:  $ABC + BC + DE$ , где  $ABC$ ,  $BC$ , и  $DE$  – минитермы.
- Выражения в ДНФ можно получить из таблицы истинности следующим способом:
  1. Определить, в каких строках таблицы выходное значение равно 1.
  2. Получить для каждой такой строки минитерм из значений входных переменных в этой строке.
  3. Просуммировать минитермы.

Ниже приведен пример преобразования таблицы истинности в булево выражение в ДНФ.

A	B	C	O
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

$A'B'C$   
 $AB'C'$   
 $AB'C$   
 $ABC'$

$O = A'B'C + AB'C' + AB'C + ABC'$

Рисунок 1-10 Преобразование таблицы истинности в ДНФ

## Конъюнктивная нормальная форма

Произведение сумм, или конъюнктивная нормальная форма (КНФ) – другой способ представления таблиц истинности.

- Произведение сумм – это произведение булевых сумм (термов) называемых макстермами.
- Макстерм – это сумма булевых переменных, равная 0.
- Для получения выражения в КНФ из таблицы истинности, необходимо выполнить следующие действия:
  1. Определить, в каких строках таблицы выходное значение равно 0.
  2. Получить для каждой такой строки макстерм из значений входных переменных в этой строке.
  3. Перемножить макстермы.

A	B	C	O
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

$A + B + C$

$A + B' + C$

$A + B' + C'$

$A' + B' + C'$

$O = (A + B + C)(A + B' + C)(A + B' + C')(A' + B' + C')$

Рисунок 1-11 Таблица истинности

Стандартные булевские выражения в дизъюнктивной нормальной форме и конъюнктивной нормальной форме – мощные инструменты для работы с таблицами истинности. Их можно использовать для получения по таблице истинности логического выражения, и, в конечном счете, разработки реальной логической схемы. Схема на основе ДНФ будет состоять из элементов AND, выходные сигналы с которых поступают на элемент OR. Схема на основе КНФ будет состоять из элементов OR, выходные сигналы с которых поступают на элемент AND.



## Самопроверка

*Примечание: Приведенные ниже вопросы должны помочь вам оценить, правильно ли вы поняли изложенную тему. Вы можете посмотреть ответы на вопросы из раздела "Самопроверка" в конце лабораторной работы*

1-1 Запишите выражения в ДНФ и КНФ по таблицам истинности для двух выходных переменных ( $O_1$  и  $O_2$ ).

A	B	C	$O_1$	$O_2$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

Рисунок 1-12 Таблица истинности для двух схем

---

---

## 1.2 Моделирование комбинационной логической схемы – Пример 1

### Схема 1

Используя ДНФ для первой схемы ( $O_1$ ), разработайте схему, реализующую данную таблицу истинности.

- Разместите на поле необходимое количество логических элементов **OR** и **AND** из группы **Misc Digital**.
- Добавьте три константы **INTERACTIVE\_DIGITAL\_CONSTANTS** из группы **Sources**.
- Добавьте один пробник **PROBE\_DIG\_RED** из группы **Indicators**.
- Соедините элементы в соответствии с таблицей истинности.

**Примечание:** Сделайте снимок или зарисуйте вашу схему и включите изображение в отчет.

- Щелкните по кнопке **Run** для запуска моделирования схемы.



Рисунок 1-13 Кнопка Run

- Используя клавиши **A**, **B** и **C** изменяйте входные сигналы схемы.

1-2 Запишите результаты, отображаемые на пробнике, в следующую таблицу истинности.

A	B	C	O
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

- Когда закончите, остановите моделирование щелчком по кнопке **Stop**.



Рисунок 1-14 Кнопка Stop

**Примечание:** Сохраните вашу схему под именем **SOP-1.ms14**.

1-3 Совпадает ли поведение вашей схемы с таблицей истинности из вопроса 1-1? Если нет, в чем может быть ошибка?

---

---

---

---

## 1.3 Моделирование комбинационной логической схемы – Пример 2

### Схема 2

Создайте новый проект.

- Используя КНФ для второй схемы ( $O_2$ ), разработайте схему, реализующую данную таблицу истинности.
- Разместите на поле необходимое количество логических элементов **OR** и **AND** из группы **Misc Digital**.
- Добавьте три константы **INTERACTIVE\_DIGITAL\_CONSTANTS** из группы **Sources**.
- Добавьте один пробник **PROBE\_DIG\_RED** из группы **Indicators**.
- Соедините элементы в соответствии с таблицей истинности.

**Примечание:** Сделайте скриншот, фотографию или зарисуйте вашу схему и включите изображение в отчет.

- Щелкните по кнопке **Run** для запуска моделирования схемы.



Рисунок 1-15 Кнопка Run

- Используя клавиши **A**, **B** и **C** изменяйте входные сигналы схемы.



1-4 Запишите результаты, отображаемые на пробнике, в следующую таблицу истинности.

A	B	C	O
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

- Когда закончите, остановите моделирование щелчком по кнопке **Stop**.



Рисунок 1-16 Кнопка Stop

**Примечание:** Сохраните вашу схему под именем **POS-2.ms14**.

1-5 Совпадает ли поведение вашей схемы с таблицей истинности из вопроса 1-1? Если нет, в чем может быть ошибка?

---

---

---

---

## 1.4 Проверка выражений ДНФ и КНФ с помощью логического преобразователя

### Логический преобразователь для Схемы 1

*Логический преобразователь* – отличный инструмент для проверки таблиц истинности и логических выражений. Для создания логического преобразователя:

- Поместите на схему логический преобразователь (**Logic Converter**) из панели инструментов на правом экране.
- Дважды щелкните по элементу **Logic Converter**, чтобы открыть его интерфейс.
- В нижней части окна введите в текстовое поле выражение в ДНФ, рассчитанное вами для первой схемы ( $O_1$ ).
- Щелкните по четвертой кнопке, **Expression to Truth Table** (преобразование выражения в таблицу истинности).

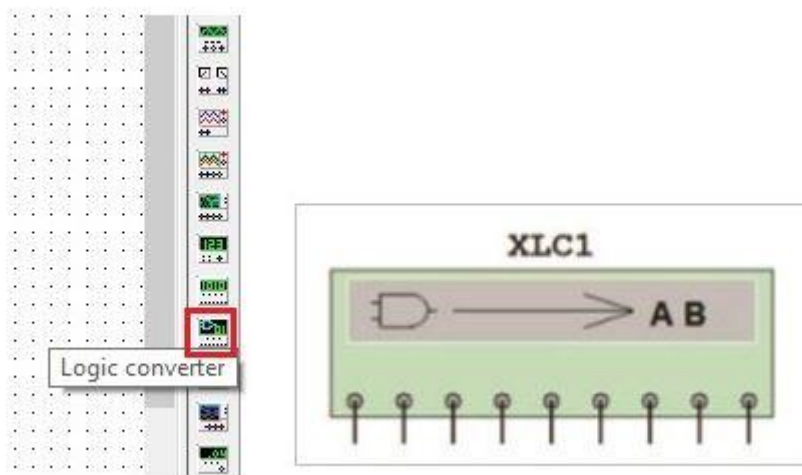


Рисунок 1-17 Логический преобразователь

1-6 Совпадает ли сгенерированная таблица истинности с таблицей из вопроса 1-1? Если да, подтверждает ли это правильность полученного вами выражения в ДНФ?

- А. Да
- В. Нет

## Логический преобразователь для Схемы 2

Для второй схемы ( $O_2$ ):

- В нижней части окна введите в текстовое поле выражение, рассчитанное вами для второй схемы в КНФ.
- Щелкните по четвертой кнопке, **Expression to Truth Table (преобразование выражения в таблицу истинности)**.

1-7 Совпадает ли сгенерированная таблица истинности с таблицей из вопроса 1-1? Если да, подтверждает ли это правильность полученного вами выражения в КНФ?

- А. Да
- В. Нет

### Функции логического преобразователя

Логический преобразователь может также генерировать схемы на основе выражений в ДНФ и КНФ. Это позволит не тратить время на построение схемы вручную.

- Щелкните по третьей кнопке, **Truth Table to Simplified Expression (преобразование таблицы истинности в упрощенное выражение)**. Эта кнопка позволяет упростить выражение, если это возможно.
- Далее нажмите пятую кнопку, **Expression to Circuit (преобразование выражения в схему)**.
  - Поместите на экран сгенерированную схему.
  - Сделайте скриншот, фотографию или зарисуйте схему и включите изображение в отчет.

1-8 Совпадает ли сгенерированная схема со схемой, которую вы создали в начале лабораторной работы? Если нет, в чем может быть причина?

---

---

---

---

## 1.5 Логический элемент XOR в Multisim

### Схема логического элемента XOR

Соберите следующую схему с логическим элементом XOR:

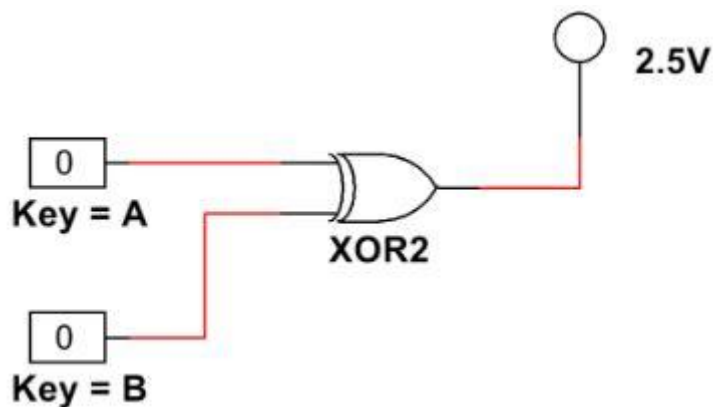


Рисунок 1-18 Схема с элементом XOR

Сконфигурируйте цифровые константы:

- Дважды щелкните по верхней константе (**Digital Constant**).
- В появившемся окне выберите "A" в выпадающем списке 'Key for toggle' (клавиша для переключения).
- Измените переключение второй константы по клавише "B".
- Щелкните по кнопке **Run** для запуска моделирования схемы.



Рисунок 1-19 Кнопка Run

- Нажмите на клавиатуре клавишу 'A', чтобы изменить значение переменной на первом (верхнем) входе на **1**.

1-9 Пробник включается?

- A. Да
- B. Нет

- Еще раз нажмите клавишу 'A', чтобы вернуть значение переменной на соответствующем входе в 0.
- Нажмите клавишу 'B', чтобы изменить значение переменной на втором входе на 1.

1-10 Пробник включается?

- A. Да
- B. Нет

- Нажмите клавишу 'A', чтобы установить переменные на обоих входах в 1.

1-11 Пробник включается?

- A. Да
- B. Нет

1-12 Как бы вы описали поведение этого логического элемента?

---

---

---

---

- Когда закончите, остановите моделирование щелчком по кнопке **Stop**.



Рисунок 1-20 Кнопка Stop

## 1.6 Создание логического элемента NOR на плате Digital Electronics

### Проект PLD

Создайте новый проект PLD.

- Выберите пункт меню **File>>New**.
- В появившемся окне выберите **PLD design** и щелкните по кнопке **Create**.
- Используйте стандартную конфигурацию платы NI Digital Electronics.
- На втором шаге назовите проект PLD '**NOR Gate**' и щелкните по кнопке **Next**.
- На третьем шаге оставьте настройки по умолчанию и щелкните по кнопке **Finish**.

Используя переключатели, светодиоды и логические элементы, создайте следующую схему:

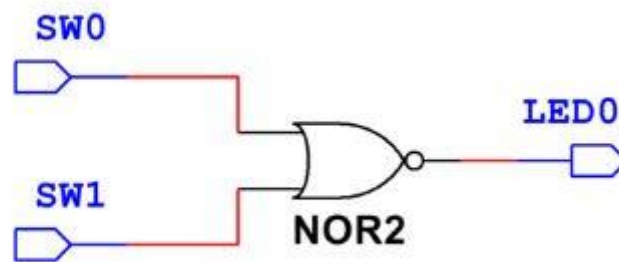


Рисунок 1-21 Проект PLD

### Развертывание схемы на плате

Для запуска этой схемы необходимо развернуть ее на плате.

- Выберите пункт меню **Transfer>>Export to PLD**.
- В появившемся окне оставьте настройки по умолчанию и щелкните по кнопке **Next**.
- На втором шаге укажите следующие настройки:
  - Выберите инструмент **Xilinx Vivado Design Suite**.
  - Выберите место для загрузки экспортируемого файла.
  - Выберите **XC7Z020**.
  - Щелкните по кнопке **Refresh**, чтобы убедиться, что ваша микросхема PLD обнаружена.
  - Щелкните по кнопке **Finish** для развертывания схемы на плате Digital Electronics.

**Примечание:** Теперь плата Digital Electronics запрограммирована, и можно изменять состояния переключателей **SW0** и **SW1**. В зависимости от их значений светодиод **LED0** будет загораться и гаснуть.

1-13 Изменяйте уровни на входах логического элемента для заполнения приведенной ниже таблицы истинности.

SW1	SW0	O
0	0	
0	1	
1	0	
1	1	

1-14 Совпадает ли полученная таблица истинности с таблицей для логического элемента NOR из раздела "Сведения из теории"?

---

---

## 1.7 Заключение

1-15 Чем отличается ДНФ от КНФ? Рассмотрите таблицы истинности и логические элементы.

---

---

---

---

1-16 Может ли одна таблица истинности соответствовать нескольким логическим схемам? Если да, приведите пример.

---

---

---

---

1-17 Почему можно упростить некоторые выражения КНФ и ДНФ?

---

---

---

---



1-18 Что такое комбинационная логическая схема?

- A. Схема, состоящая из двух или более логических элементов
- B. Схема, выходной сигнал которой зависит только от текущих значений входных переменных, и которая подчиняется законам булевой алгебры
- C. Схема, которая может работать только в Multisim
- D. Схема с двумя или более входами

1-19 Минтерм используется для определения:

- A. Таблиц истинности
- B. Дизъюнктивной нормальной формы
- C. Конъюнктивной нормальной формы
- D. Количества элементов NAND в схеме

1-20 Схема, создаваемая по КНФ, состоит из:

- A. Элементов NOR, выходные сигналы с которых поступают на элемент NAND
- B. Элементов NAND, выходные сигналы с которых поступают на элемент NOR
- C. Элементов OR, выходные сигналы с которых поступают на элемент AND
- D. Элементов AND, выходные сигналы с которых поступают на элемент OR

1-21 В Multisim логические элементы находятся в следующем семействе раздела Misc Digital group:

- A. TIL
- B. Microcontrollers
- C. Memory
- D. Line driver

1-12 Чем заменяются интерактивные цифровые константы при создании схемы в программируемой логической интегральной схеме (PLD)?

- A. Светодиодами
- B. FPGA
- C. Логическими элементами
- D. Переключателями

Ответы – только на вопросы из раздела *Самопроверка*



Проверьте себя

1-1

**O1 – ДНФ:**  $A'B'C + A'BC' + ABC'$

**O1 – КНФ:**  $(A + B + C)(A + B' + C')(A' + B + C)(A' + B + C')(A' + B' + C')$

**O2 – ДНФ:**  $A'B'C + A'BC' + A'BC + AB'C + ABC' + ABC$

**O2 – КНФ:**  $(A + B + C)(A' + B + C)$