

ПОСТРОЕНИЕ СХЕМ ЛОГИЧЕСКИХ ФУНКЦИЙ В ОСНОВНОМ БАЗИСЕ

И.Е. Ягодкина, Е.Р. Каландаров

ФГБОУ ВО «Воронежский государственный лесотехнический университет
имени Г.Ф. Морозова»

Аннотация. В данной работе был проведён комплексный анализ построения схем логических функций в программе Logisim на основе основного базиса. Изучена соответствующая литература по логическим элементам и их применению в цифровых схемах. Были разработаны и смоделированы схемы, которые соответствуют поставленной задаче. Схемы проектировались с учётом возможного использования в реальных технических системах и их оптимизации.

Ключевые слова: Logisim, проектирование, логические функции, основной базис, цифровые схемы

CONSTRUCTION OF LOGICAL FUNCTION CIRCUITS IN THE MAIN BASIS

I.E. Yagodkina, E.R. Kalandarov

Voronezh State University of Forestry and Technologies named after G.F. Morozov

Abstract. In this work, a comprehensive analysis of the construction of logical function circuits in the Logisim program based on the main basis was carried out. The relevant literature on logical elements and their use in digital circuits was studied. Circuits that correspond to the task were developed and modeled. The circuits were designed taking into account possible use in real technical systems and their optimization.

Keywords: Logisim, design, logical functions, main basis, digital circuits

Введение

В основе функционирования всех современных вычислительных машин лежит двоичная система счисления, оперирующая всего двумя символами – 0 и 1. Это не просто удобное соглашение; такой подход напрямую связан с физической реализуемостью вычислений. Транзисторы, фундаментальные элементы современной электроники, работают по принципу "включено/выключено", что естественным образом отображается на двоичные 1 и 0. Этот принцип "включено/выключено" лежит в основе булевой алгебры, математической системы, идеально подходящей для описания работы логических схем. Булева алгебра оперирует логическими операциями:

конъюнкцией (И), дизъюнкцией (ИЛИ) и инверсией (НЕ). Эти три операции образуют *базисный набор*, достаточный для построения любой произвольной булевой функции. Это означает, что любую сложную логическую операцию, какую бы задачу мы ни решали на компьютере, можно разложить на последовательность этих трёх базовых операций.

Важно отметить, что существует не один, а множество базисных наборов. Хотя конъюнкция, дизъюнкция и инверсия образуют наиболее интуитивно понятный и часто используемый базис, другие наборы операций также обладают свойством функциональной полноты. Например, можно построить любой логический элемент, используя только операции Штрих Шеффера ($\neg(x \wedge y)$) или стрелку Пирса ($\neg(x \vee y)$). Эти операции, будучи сами по себе достаточно простыми, позволяют синтезировать всю мощь булевой алгебры. Выбор конкретного базиса часто диктуется технологическими соображениями: определенные логические операции могут быть проще и эффективнее реализованы на физическом уровне с помощью конкретных электронных компонентов.

В рамках моделирования функциональных схем, каждая базовая логическая операция (конъюнкция, дизъюнкция, инверсия) представляется отдельной программной процедурой или функцией. В процессе моделирования, сложная булева функция разбивается на последовательность вызовов этих базовых процедур, соответствующих логической схеме. Например, если нужно смоделировать работу схемы, вычисляющей функцию $F(x, y, z) = (x \wedge y) \vee z$, то программа будет последовательно вызывать процедуру конъюнкции для вычисления $(x \wedge y)$, а затем процедуру дизъюнкции для вычисления результата с z . Такой подход позволяет пошагово проследить выполнение логических операций и отладить работу схемы.

Более того, моделирование позволяет экспериментировать с различными базисными наборами, сравнивая эффективность и сложность реализации той же функции с использованием разных логических элементов. Например, можно сравнить количество необходимых логических вентилях (физических реализаций логических операций) для реализации данной функции при использовании стандартного базиса (конъюнкция, дизъюнкция, инверсия) и базиса, состоящего только из элементов Штриха Шеффера. Это помогает оптимизировать проектирование цифровых схем, минимизируя затраты на компоненты и энергопотребление. Компьютерные эксперименты позволяют провести такое сравнение для большого числа функций и базисов, что невозможно сделать ручным способом. Результаты таких экспериментов позволяют выбирать наиболее эффективный вариант реализации логической схемы для конкретной задачи. Современные инструменты автоматизированного проектирования (EDA) широко используют такие методы для оптимизации сложных цифровых систем.

1. Библиотека базовых логических элементов

1.1. Моделирование элементов основного базиса

Цифровые схемы, лежащие в основе современных компьютеров и электронных устройств, построены на комбинации нескольких базовых логических элементов. К их числу относятся инвертор, элемент "И" и элемент "ИЛИ". Каждый из этих элементов выполняет специфическую логическую операцию над входными сигналами и генерирует соответствующий выходной сигнал. Давайте рассмотрим каждый из них более детально.

1.2. Инвертор (элемент "НЕ")

Инвертор – это самый простой логический элемент, выполняющий операцию логического отрицания. Его функциональность заключается в том, что он меняет логическое состояние входного сигнала на противоположное. То есть, если на вход инвертора подается логическая единица (1), на выходе будет логический ноль (0), и наоборот.

Функциональные особенности:

Один вход и один выход: это характерная черта инвертора, отличающая его от других логических элементов.

Инверсия сигнала: Выходной сигнал всегда противоположен входному.

Таблица истинности:

Вход	Выход
0	1
1	0

Применение:

Инверторы широко используются для изменения логического уровня сигнала, формирования инверсных сигналов и реализации других логических функций в сочетании с другими элементами.

Анализ работы инвертора

Работа инвертора базируется на зависимости между входным и выходным напряжением. Рассмотрим три режима работы:

1. **Режим насыщения (логическая единица на выходе):** если на входе инвертора V_{in} низкое, PMOS-транзистор включен, а NMOS-транзистор выключен. В этом случае выход V_{out} будет высоким (логическая единица).

2. **Режим переключения:** когда входной сигнал V_{in} находится в переходной области, оба транзистора частично проводимы, и выходное напряжение начинает изменяться в зависимости от сопротивлений обоих транзисторов. В этом режиме происходит плавный переход от низкого уровня к высокому.

3. **Режим отсечки (логический ноль на выходе):** если на входе V_{in} высокое, PMOS-транзистор выключен, а NMOS-транзистор включен, вытягивая выход V_{out} к логическому нулю (земля).

1.3. Элемент "И" (конъюнкция)

Элемент "И" выполняет логическую операцию конъюнкции, которую можно интерпретировать как логическое умножение. Выходной сигнал элемента "И" равен логической единице только в том случае, если все его входные сигналы равны логической единице.

Функциональные особенности:

Два или более входов, один выход: Число входов может варьироваться в зависимости от конкретной реализации.

Логическое умножение: Выходной сигнал равен 1 только при всех входных сигналах, равных 1.

Таблица истинности для двухвходового элемента "И":

Вход 1	Вход 2	Выход
0	0	0
0	1	0
1	0	0
1	1	1

Применение:

Элементы "И" используются для проверки одновременного выполнения нескольких условий, реализации логических функций, таких как сравнение чисел, формирование управляющих сигналов и т.д.

Анализ работы конъюнкции в цифровых схемах

Конъюнкция реализуется в цифровой логике с помощью AND-элемента (логический элемент И). Этот элемент имеет два или более входов и один выход. Выходной сигнал будет высоким (логическая 1) только в том случае, если все входные сигналы высокие (логическая 1). В любом другом случае выход будет низким (логический 0).

Логический элемент AND:

Входы: Один или более логических сигналов A, B, C, и так далее.

Выход: $A \wedge B$, $A \wedge B \wedge C$, и так далее.

Пример:

Для двух входов A и B:

Если $A=1$ и $B=1$, то $A \wedge B=1$

Если хотя бы один из входов $A=0$ или $B=0$, то $A \wedge B=0$.

1.4. Элемент "ИЛИ" (дизъюнкция)

Элемент "ИЛИ" выполняет логическую операцию дизъюнкции, которую можно интерпретировать как логическое сложение. Выходной сигнал элемента "ИЛИ" равен логической единице, если хотя бы один из его входных сигналов равен логической единице.

Функциональные особенности:

Два или более входов, один выход: как и элемент "И", элемент "ИЛИ" может иметь различное число входов.

Логическое сложение: Выходной сигнал равен 1, если хотя бы один входной сигнал равен 1.

Таблица истинности для двухвходового элемента "ИЛИ":

Вход 1	Вход 2	Выход
0	0	0
0	1	1

| 1 | 0 | 1 |
| 1 | 1 | 1 |

Применение:

Элементы "ИЛИ" используются для реализации функций выбора, обнаружения ошибок, формирования управляющих сигналов и т.д.

Анализ работы дизъюнкции в цифровых схемах

Дизъюнкция в цифровой логике реализуется с помощью OR-элемента. Этот элемент имеет два или более входов и один выход. Выходной сигнал будет высоким (логическая 1) при наличии хотя бы одного высокого сигнала на входе.

Логический элемент OR:

Два входа: A и B.

Выход: $A \vee B$ — результат логической дизъюнкции.

Пример:

Если к элементу OR поданы следующие сигналы:

$A=0$, $B=1$, тогда выходной сигнал будет $A \vee B=1$.

Если в цепь добавить больше входов, например, три переменные AAA, BBB, и CCC, дизъюнкция этих переменных запишется как:

$A \vee B \vee C$.

Результат будет «1», если хотя бы одна из переменных равно 1.

Инвертор, элемент "И" и элемент "ИЛИ" являются фундаментальными элементами цифровой электроники. Понимание их принципов работы и взаимодействия позволяет создавать сложные цифровые системы, способные выполнять разнообразные задачи.

Построение схемы логической функции в Logisim можно выполнить пошагово следующим образом:

2. Построение функциональной схемы в основном базисе

Давайте разберем процедуру моделирования функциональной схемы, представленной на рисунке 1, используем программное обеспечение Logisim и ключевые логические элементы, такие как AND (И), OR (ИЛИ) и NOT (НЕТ). Этот операция содержит пару тройку основных действий, которые мы рассмотрим по пунктам.

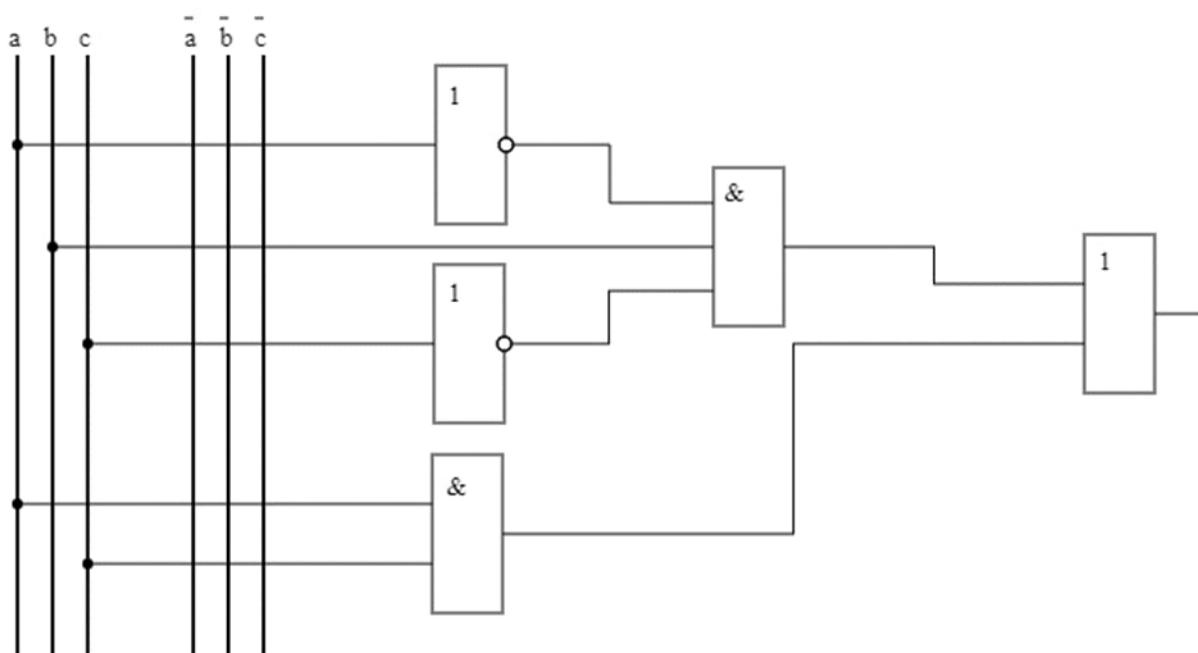


Рисунок 1 – Схема для исследования в нормальной дизъюнктивной формы

Шаг 1: Запуск Logisim

Сначала находим программу Logisim у себя на компьютере, если же ее нету, то установите. Вторым шагом будет запуск данного программного обеспечения. Далее спроектирует новый проект в главном меню Файл → Создать. Эта операция даст вам создать новую свою схему с полного 0.

Шаг 2: Подготовка входных переменных

Сейчас нужно подобрать входящие переменные (input), которые будут применяться в вашей созданной схеме. Для этого сделайте следующие шаги:

1. На панели инструментов (в левом верхнем углу) выберите компонент Input Pin (входные пины) и перетащите их на рабочее поле и назовите их a, b, c для идентификации значений, которые вводятся в схему. Эти переменные могут представлять любые бинарные значения, которые вы будете использовать в логических операциях.

2. К тому же, вам нужны будут вспомогательные входы для инверсий переменных a, b, и c. С этой целью примените элемент NOT Gate из панели компонентов. Инверсия сигналов даст вам возможность получать значения, противоположные исходным.

Шаг 3: Размещение логических элементов

На очередном шаге вам предстоит размещать необходимые логические элементы, которые будут реализовывать операции над переменными. Эти элементы доступны на панели слева в разделе Gates:

1. NOT (НЕ): поместите три элемента NOT около входов a, b, и c. Эти элементы будут инвертировать значения входных переменных, создавая выходы $\bar{a}$, $\bar{b}$, и $\bar{c}$ (то есть не a, не b и не c).

2. AND (И): Вам необходимо будет три элемента AND (элементы И), так как на вашей схеме изображены три группы логических операций:

- Начальный элемент AND (И) должен соединять a , b , и $\neg c$ (a , b и не c).
- Второй элемент AND связывает $\neg a$, b , и c (не a , b , c).
- Третий элемент AND связывает a , $\neg b$, и c (a , не b , c).

Данные логические операции будут определять, комбинации входных значений будут приводить к определенным выходным результатам.

Шаг 4: Построение соединений

Сейчас наступило время соединить все элементы схемы. Корректное подключение будет ключевым для правильной работы вашей схемы. Вот основные шаги:

1. Подключите входные пины a , b , и c к соответствующим элементам AND. Убедитесь, что каждый вход подключен правильно.
2. Далее, подключите выходы элементов NOT к надлежащим входам элементов AND, чтобы передать инвертированные значения $\neg a$, $\neg b$, и $\neg c$.
3. Соедините выходы всех трех элементов AND с входами элемента OR (ИЛИ), если он присутствует в вашей схеме. Это обеспечит получение окончательного результата на выходе.

Дополнительные рекомендации

1. Тестирование схемы: после завершения подключения всех элементов, вы можете протестировать свою схему, изменяя значения переменных a , b , и c . Это поможет вам убедиться, что логические операции выполняются корректно.
2. Документация: рекомендуется вести документацию по вашей схеме, включая описание каждой логической операции и ее назначения. Это упростит дальнейшую работу и позволит другим пользователям понять вашу логику.
3. Экспорт схемы: Logisim позволяет экспортировать вашу схему в различные форматы. Это может быть полезно для презентаций или дальнейшего анализа.

Таким образом, процесс моделирования функциональной схемы в Logisim включает в себя последовательное выполнение шагов по подготовке входов, размещению логических элементов и построению соединений. Важно тщательно следить за каждым этапом, чтобы ваша схема работала правильно и эффективно.

Список литературы

1. Абрамович А. И., Бояринцева М. А. Логические основы цифровых устройств. – М.: Наука и техника, 2019. – 256 с.
2. Каминский, А. Г. Основы цифровой электроники: учебное пособие. – СПб.: Питер, 2018. – 320 с.
3. Микруков, П. В. Цифровые устройства и микропроцессоры: основы проектирования логических схем. – М.: Издательство МГУ, 2021. – 400 с.
4. Барков С. А., Степанов Е. М. Теория и проектирование цифровых систем на основе логических схем. – М.: Академия, 2020. – 350 с.
5. Хидиров, В. М. Основы теории логических функций. – СПб.: СПбГЭТУ "ЛЭТИ", 2022. – 200 с.

6. Манилов, А. Д. Логика и архитектура цифровых схем. – М.: Физматлит, 2017. – 280 с.
7. Боуман П., Макмартин Л. Проектирование цифровых устройств с использованием основных логических элементов. – М.: Мир, 2016. – 240 с.
8. Яблоков Н. А., Сидоров А. А. Алгоритмы синтеза и оптимизации логических схем. – СПб.: Издательство Политехнического университета, 2019. – 300 с.
9. Гейл Дж., Макинтайр К. Основы проектирования цифровых схем. – М.: Диалектика, 2018. – 350 с.
10. Николенко С.Д., Сазонова С.А., Мальцева Е.И. Процесс технического обследования кукольного театра // Моделирование систем и процессов. – 2022. – Т. 15, № 3. – С. 44-52.

References

1. Abramovich A. I., Boyarintseva M. A. Logical foundations of digital devices. - Moscow: Science and Technology, 2019. – 256 p.
2. Kaminsky, A. G. Fundamentals of digital electronics: textbook. – SPb.: Piter, 2018. – 320 p.
3. Mikrukov, P. V. Digital devices and microprocessors: fundamentals of logic circuit design. – Moscow: MSU Publishing House, 2021. - 400 p.
4. Barkov S. A., Stepanov E. M. Theory and design of digital systems based on logic circuits. – Moscow: Academy, 2020. – 350 p.
5. Khidirov V. M. Fundamentals of the theory of logic functions. – SPb.: SPbGETU “LETI”, 2022. – 200 p.
6. Manilov A. D. Logic and architecture of digital circuits. – Moscow: Fizmatlit, 2017. – 280 p.
7. Bowman P., McMartin L. Designing digital devices using basic logic elements. – M.: Mir, 2016. – 240 p.
8. Yablokov N. A., Sidorov A. A. Algorithms of synthesis and optimization of logic circuits. - St. Petersburg: Publishing house of Polytechnic University, 2019. – 300 p.
9. Gale J., McIntyre K. Fundamentals of digital circuit design. – Moscow: Dialectics, 2018. – 350 p.
10. Nikolenko S.D., Sazonova S.A., Maltseva E.I. Process of technical inspection of the puppet theater // Modeling of systems and processes. – 2022. – Vol. 15, № 3. – P. 44-52.