|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  **МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**  (Национальный исследовательский университет) | | | | | |
|  | | | | |
|  | Институт №3 «Системы управления, информатика и электроэнергетика»  Кафедра 307 «Цифровые технологии и информационные системы» | | |  |
|  |  | | |  |
|  | ОТЧЁТ о выполнении задания на практическом занятиии «Создание схемы последовательной логики»  по дисциплине «Основы цифровой схемотехники» | | |  |
|  |  | | |  |
|  | Выполнил студент 2-го курса  группы М3О-214Б-22 | Подпись: | Бедретдинов Андрей Маратович |  |
|  | Проверил ст. преподаватель  каф. 307 | Подпись: | Коробков Максим Андреевич |  |
| Москва 2024 | | | | |

# РЕФЕРАТ

Отчёт 22 с., 9 рис., 1 таб..

АСИНХРОННЫЙ RS-ТРИГГЕР, ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ, ПАМЯТЬ, ЦИКЛИЧЕСКИЙ СДВИД, D-ТРИГГЕР, PROTEUS, СИМУЛЯЦИЯ

Цель работы — разработка, реализация и тестирование двух цифровых схем в программе Proteus: асинхронного RS-триггера и схемы циклического сдвига вправо.

В процессе работы выполнены следующие задачи:

1. Изучить теоретические основы работы асинхронного RS-триггера и схемы циклического сдвига вправо.
2. Создать новый проект в программе Proteus и найти все необходимые компоненты в библиотеке.
3. Реализовать асинхронный RS-триггер на логических элементах NOR, обеспечивающий функции ячейки памяти.
4. Разместить логические элементы NOR на рабочем поле и соединить их согласно принципиальной схеме RS-триггера.
5. Реализовать схему циклического сдвига вправо на 4 разряда с 6 входами и 4 выходами.
6. Разместить логические элементы AND, NOT и D-триггеры на рабочем поле и соединить их согласно принципиальной схеме.
7. Настроить параметры компонентов и проверить схему на наличие ошибок с использованием автоматических средств проверки в Proteus.
8. Провести симуляцию работы асинхронного RS-триггера, проверяя правильность установки и сброса состояний.
9. Провести симуляцию работы схемы циклического сдвига, проверяя корректность сдвига битов вправо и циклического повторения значений.
10. Зафиксировать результаты тестирования в виде скриншотов и проанализировать их на соответствие теоретическим ожиданиям.

Отчёт о проведении научно-исследовательской работы оформлен в соответствии с ГОСТ 7.32–2017.

**СОДЕРЖАНИЕ**

[РЕФЕРАТ 2](#_Toc169787324)

[ПОСТАНОВКА ЗАДАЧИ 5](#_Toc169787325)

[КРАТКОЕ ОПИСАНИЕ ТЕОРИИ 7](#_Toc169787326)

[1. Асинхронный RS-триггер 7](#_Toc169787327)

[2. Схема циклического сдвига вправо 8](#_Toc169787328)

[ПРОЦЕСС СОЗДАНИЯ 9](#_Toc169787329)

[1. Асинхронный RS-триггер 9](#_Toc169787330)

[2. Схема циклического сдвига вправо 10](#_Toc169787331)

[ТЕСТИРОВАНИЕ 13](#_Toc169787332)

[1. Тестирование асинхронного RS-триггера 13](#_Toc169787333)

[2. Тестирование схемы циклического сдвига вправо 15](#_Toc169787334)

[3. Анализ результатов 20](#_Toc169787335)

[ЗАКЛЮЧЕНИЕ 21](#_Toc169787336)

# ПОСТАНОВКА ЗАДАЧИ

Целью лабораторной работы является разработка, реализация и тестирование двух цифровых схем в программе Proteus: асинхронного RS-триггера и схемы циклического сдвига вправо. Асинхронный RS-триггер должен быть построен на логических элементах NOR и выполнять функции ячейки памяти, сохраняя свое состояние до поступления сигнала на один из входов, изменяющего это состояние. Схема циклического сдвига вправо должна иметь 4 разряда, 6 входов и 4 выхода, обеспечивая сдвиг битов вправо с циклическим повторением значений.

Для выполнения работы необходимо изучить теоретические основы работы асинхронного RS-триггера и схемы циклического сдвига вправо. В программе Proteus следует создать новый проект, используя встроенные библиотеки для поиска и добавления необходимых компонентов. В первом задании на рабочем поле схемы должны быть размещены логические элементы NOR для создания асинхронного RS-триггера. Входы этих элементов подключаются к внешним источникам сигналов, а выходы одного элемента подключаются ко входам другого, формируя обратную связь.

Во втором задании требуется разместить логические элементы AND, NOT и D-триггеры для создания схемы циклического сдвига вправо. Входные сигналы должны быть подключены к логическим элементам, обеспечивая правильную логическую обработку данных и их сдвиг вправо. Выходы логических элементов подключаются к входам D-триггеров, которые фиксируют состояние сигнала и обеспечивают циклический сдвиг.

После завершения построения схем необходимо настроить параметры компонентов и проверить их на наличие ошибок с использованием автоматических средств проверки в Proteus. В случае обнаружения ошибок требуется их устранение путем корректировки соединений или параметров компонентов. Завершающим этапом является симуляция и тестирование схем, где активируются входные сигналы и проверяется правильность работы схем. Для асинхронного RS-триггера проверяется корректность установки и сброса состояний, а для схемы циклического сдвига — корректность сдвига битов вправо и циклического повторения значений. Результаты тестирования фиксируются в виде скриншотов и анализируются на соответствие теоретическим ожиданиям.

# КРАТКОЕ ОПИСАНИЕ ТЕОРИИ

### Асинхронный RS-триггер

Асинхронный RS-триггер, также известный как ячейка памяти на логических элементах NOR, является одним из простейших типов триггеров. Он представляет собой базовый блок памяти, который может хранить одноразрядную информацию (бит). RS-триггер имеет два входа: Set (S) и Reset (R), а также два выхода: прямой (Q) и инверсный (Q̅).

Работа RS-триггера основана на использовании двух логических элементов NOR, которые связаны таким образом, что выход одного элемента подается на вход другого. Это создает обратную связь, благодаря которой триггер может сохранять своё состояние.

Таблица 1 — Таблица истинности асинхронного RS-триггера

|  |  |  |  |
| --- | --- | --- | --- |
| **R** | **S** | **Q** | **Q̅** |
| 0 | 0 | Q | Q̅ |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | - | - |

При R = 0 и S = 0 триггер сохраняет свое текущее состояние. При R = 0 и S = 1 триггер устанавливается в состояние Q = 1 (Set). При R = 1 и S = 0 триггер сбрасывается в состояние Q = 0 (Reset). При R = 1 и S = 1 состояние триггера неопределенное и не рекомендуется для нормальной работы.

Схема асинхронного RS-триггера на логических элементах NOR включает два элемента NOR, каждый из которых имеет два входа. Один из входов каждого элемента соединен с выходом другого элемента, создавая обратную связь. Другие входы являются управляющими: входы S (Set) и R (Reset). Состояние триггера определяется комбинацией сигналов на входах S и R.

### Схема циклического сдвига вправо

Схема циклического сдвига вправо используется для сдвига битов двоичного числа вправо, при этом бит, выходящий за пределы младшего разряда, переносится в старший разряд. В данной работе реализована схема циклического сдвига на 4 разряда с 6 входами и 4 выходами, которая включает в себя логические элементы AND, NOT и D-триггеры.

D-триггер является ключевым элементом в данной схеме, поскольку он позволяет фиксировать состояние сигнала на своем входе и сохранять его до следующего тактового сигнала. D-триггер имеет вход данных (D), вход тактового сигнала (CLK) и два выхода: прямой (Q) и инверсный (Q̅).

Принцип работы схемы циклического сдвига заключается в следующем. Входные данные подаются на логические элементы AND и NOT для создания соответствующих логических выражений, которые затем подаются на входы D-триггеров. Каждый D-триггер сохраняет состояние своего входа и передает его на выход при получении тактового сигнала. Таким образом, битовые значения сдвигаются вправо, а выходящий бит возвращается в старший разряд, создавая циклический эффект.

Схема циклического сдвига вправо на 4 разряда включает логические элементы AND, NOT и D-триггеры. Входные сигналы поступают на логические элементы, которые формируют логические выражения для каждого разряда. Эти выражения затем подаются на входы D-триггеров, которые фиксируют состояние сигналов и сдвигают их вправо при каждом тактовом сигнале.

# ПРОЦЕСС СОЗДАНИЯ

### Асинхронный RS-триггер

Процесс создания асинхронного RS-триггера начинается с открытия программы Proteus и создания нового проекта. Важно задать корректное имя проекта и указать директорию для его сохранения, чтобы обеспечить легкий доступ к файлам проекта в дальнейшем.

После создания проекта в программе Proteus начинается поиск и добавление необходимых компонентов. Для реализации асинхронного RS-триггера необходимы логические элементы NOR, входные и выходные порты. Используя встроенную библиотеку компонентов Proteus, в поисковой строке вводится "NOR" для поиска логических элементов NOR. После этого выбранные компоненты добавляются на рабочее поле схемы.

Размещение компонентов на рабочем поле осуществляется таким образом, чтобы обеспечить удобство их соединения. Два логических элемента NOR размещаются рядом друг с другом. Входы каждого элемента NOR подключаются к внешним источникам сигналов: S (Set) и R (Reset). Один из входов каждого элемента подключается к выходу другого элемента, формируя обратную связь.

Соединительные линии создаются с помощью инструмента "Wire". Один выход элемента NOR подключается к входу другого элемента. Это создает обратную связь, необходимую для работы триггера. Входы S и R подключаются к соответствующим портам, которые будут использоваться для подачи управляющих сигналов.

После завершения построения схемы проводится настройка параметров компонентов. Важно убедиться, что все соединения выполнены правильно и что каждый элемент имеет корректные параметры. Для проверки схемы на наличие ошибок используется функция автоматической проверки в Proteus.

Завершающим этапом является симуляция и тестирование асинхронного RS-триггера. В процессе симуляции подаются сигналы на входы S и R, и наблюдается поведение выходов Q и Q̅. Проверяется правильность установки и сброса состояний триггера. Результаты тестирования фиксируются в виде скриншотов и анализируются на соответствие теоретическим ожиданиям.

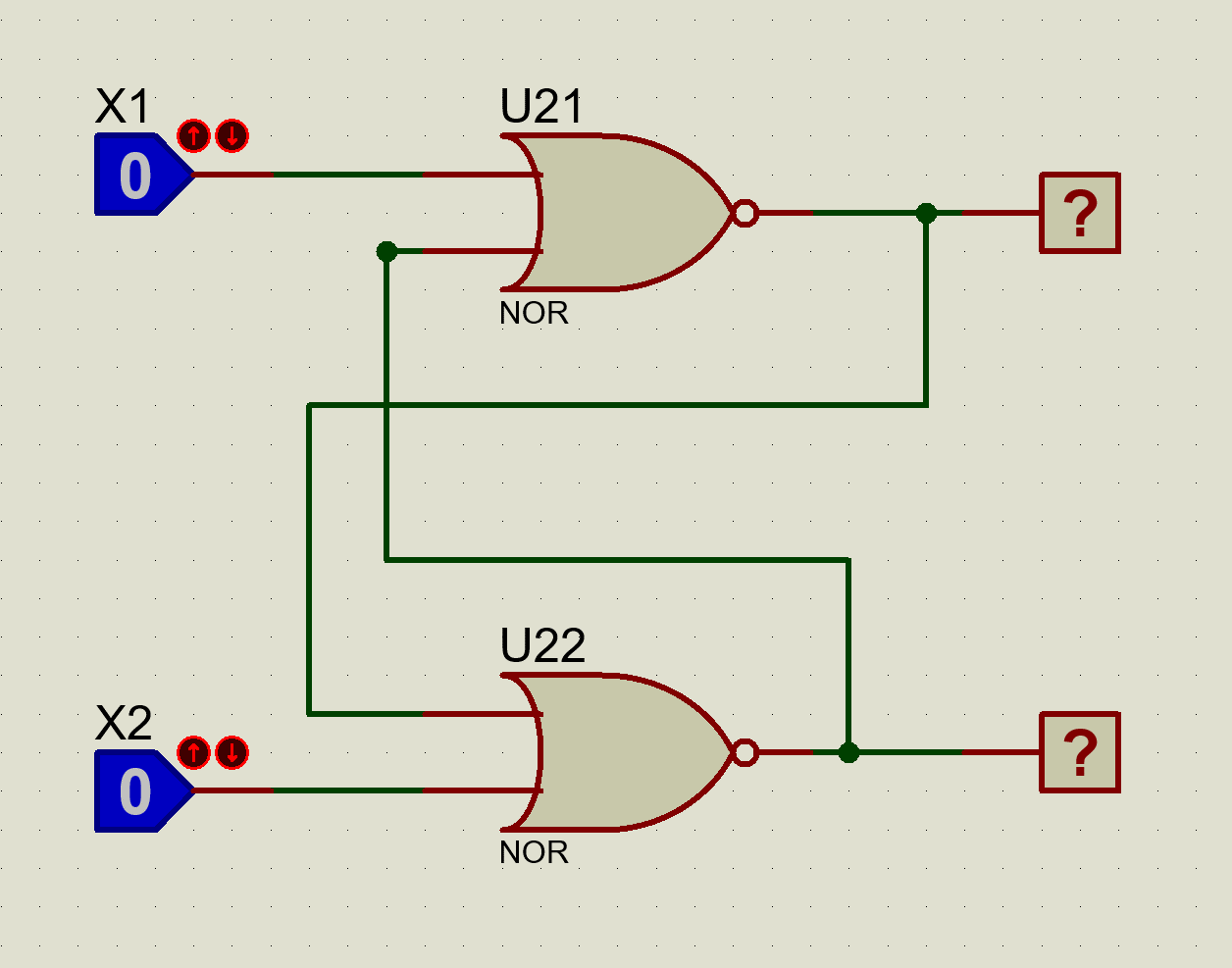


Рисунок 1 — Схема циклического сдвига вправо в Proteus

### Схема циклического сдвига вправо

Процесс создания схемы циклического сдвига вправо начинается аналогично, с открытия программы Proteus и создания нового проекта. В библиотеке компонентов Proteus находятся и добавляются необходимые элементы: логические элементы AND, NOT и D-триггеры.

На рабочем поле схемы размещаются логические элементы AND, NOT и D-триггеры. Входные сигналы подаются на логические элементы AND и NOT для формирования соответствующих логических выражений. Эти выражения затем передаются на входы D-триггеров. Каждый D-триггер имеет вход данных (D), вход тактового сигнала (CLK) и два выхода: прямой (Q) и инверсный (Q̅).

Соединительные линии создаются с помощью инструмента "Wire". Входные сигналы подключаются к логическим элементам AND и NOT, а выходы этих элементов – к входам D-триггеров. Важно правильно соединить входы и выходы, чтобы обеспечить корректную работу схемы. Выходы D-триггеров подключаются к следующим логическим элементам и формируют циклический сдвиг.

После завершения построения схемы проводится настройка параметров компонентов. Проверяется правильность соединений и соответствие параметров логических элементов и D-триггеров. Для проверки схемы на наличие ошибок используется функция автоматической проверки в Proteus.

Завершающим этапом является симуляция и тестирование схемы циклического сдвига вправо. В процессе симуляции подаются входные сигналы и наблюдается поведение выходов. Проверяется корректность сдвига битов вправо и циклическое повторение значений. Результаты тестирования фиксируются в виде скриншотов и анализируются на соответствие теоретическим ожиданиям.

Эти шаги позволяют успешно создать и протестировать асинхронный RS-триггер и схему циклического сдвига вправо в программе Proteus.

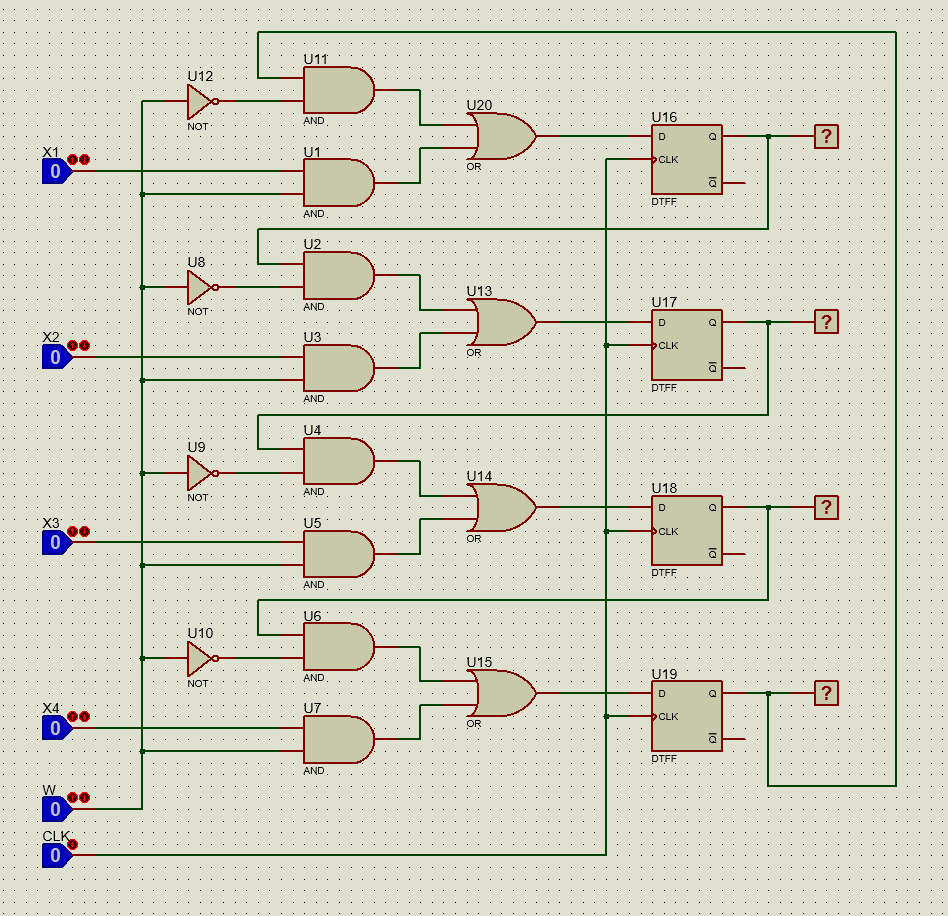


Рисунок 2 — Схема циклического сдвига вправо в Proteus

# ТЕСТИРОВАНИЕ

### Тестирование асинхронного RS-триггера

Для тестирования асинхронного RS-триггера были проведены симуляции с подачей входных сигналов 10 и 01. Основное внимание уделялось проверке правильности установки и сброса состояний триггера.

При подаче входного сигнала 10 (S=1, R=0) триггер должен установить выход Q в состояние 1, а выход Q̅ в состояние 0. На скриншоте видно, что при активном сигнале Set (S=1, R=0) выходные сигналы триггера принимают значения Q=1 и Q̅=0, что соответствует теоретическим ожиданиям.

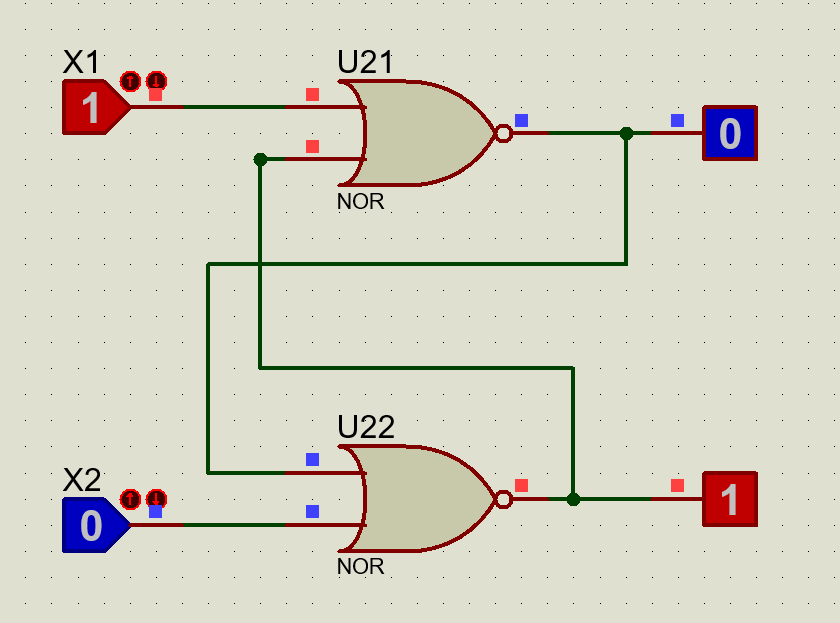


Рисунок 3 — Первый этап тестирования асинхронного RS-триггера

При подаче входного сигнала 01 (S=0, R=1) триггер должен сбросить выход Q в состояние 0, а выход Q̅ в состояние 1. На скриншоте видно, что при активном сигнале Reset (S=0, R=1) выходные сигналы триггера принимают значения Q=0 и Q̅=1, что также соответствует теоретическим ожиданиям.

Изображение выглядит как диаграмма, текст, линия, План

Автоматически созданное описание

Рисунок 4 — Второй этап тестирования асинхронного RS-триггера

Тестирование показало, что асинхронный RS-триггер работает корректно, устанавливая и сбрасывая состояния выходов в соответствии с подаваемыми входными сигналами.

### Тестирование схемы циклического сдвига вправо

Для тестирования схемы циклического сдвига вправо были проведены симуляции с последовательным выполнением операций записи и сдвига. Основное внимание уделялось проверке корректности сдвига битов вправо и циклическому повторению значений.

При зажатой записи и подаче входного сигнала 1100, схема должна записать это значение во внутренние регистры. На скриншоте видно, что при активном сигнале записи входное значение 1100 правильно записано в регистры схемы.

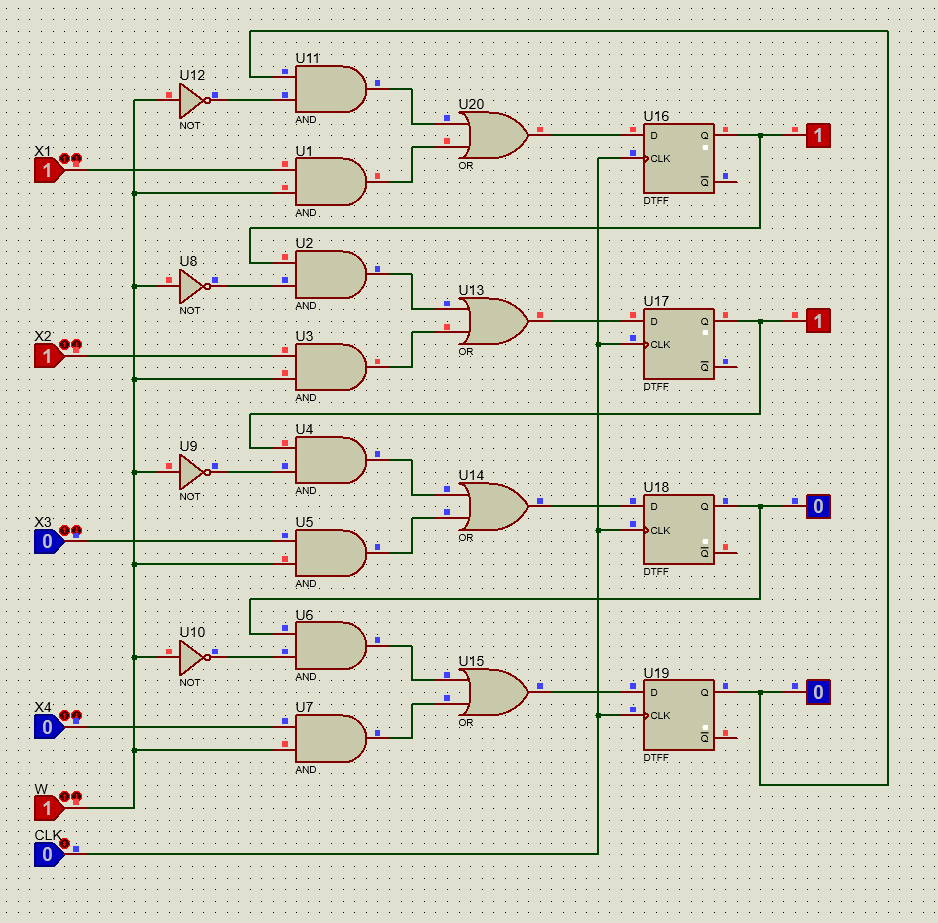


Рисунок 5 — Первый этап тестирования циклического сдвига вправо

После отжатия записи и нажатия на тактовый сигнал (CLK), происходит сдвиг битов вправо на 1 разряд. В результате этого сдвига значение 1100 должно измениться на 0110. На скриншоте видно, что после первого сдвига выходное значение стало 0110, что соответствует теоретическим ожиданиям.

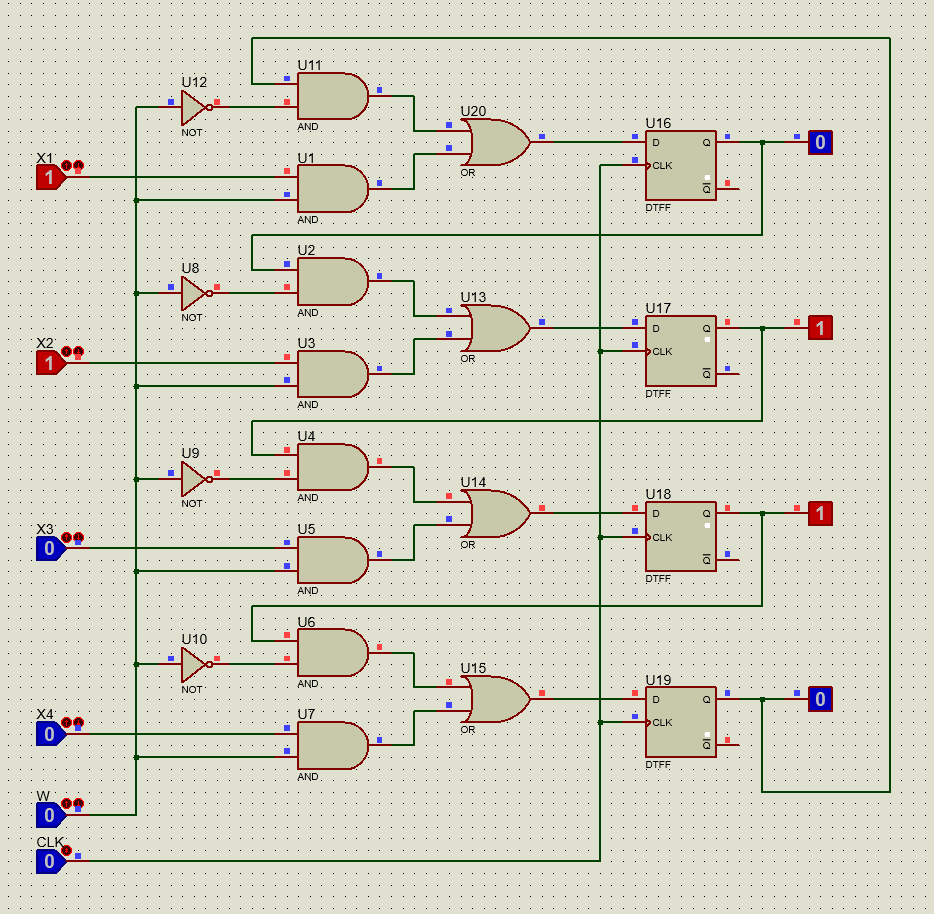


Рисунок 6 — Второй этап тестирования циклического сдвига вправо

После повторного нажатия на тактовый сигнал (CLK), происходит еще один сдвиг битов вправо на 1 разряд. Значение 0110 должно измениться на 0011. На скриншоте видно, что после второго сдвига выходное значение стало 0011, что соответствует теоретическим ожиданиям.

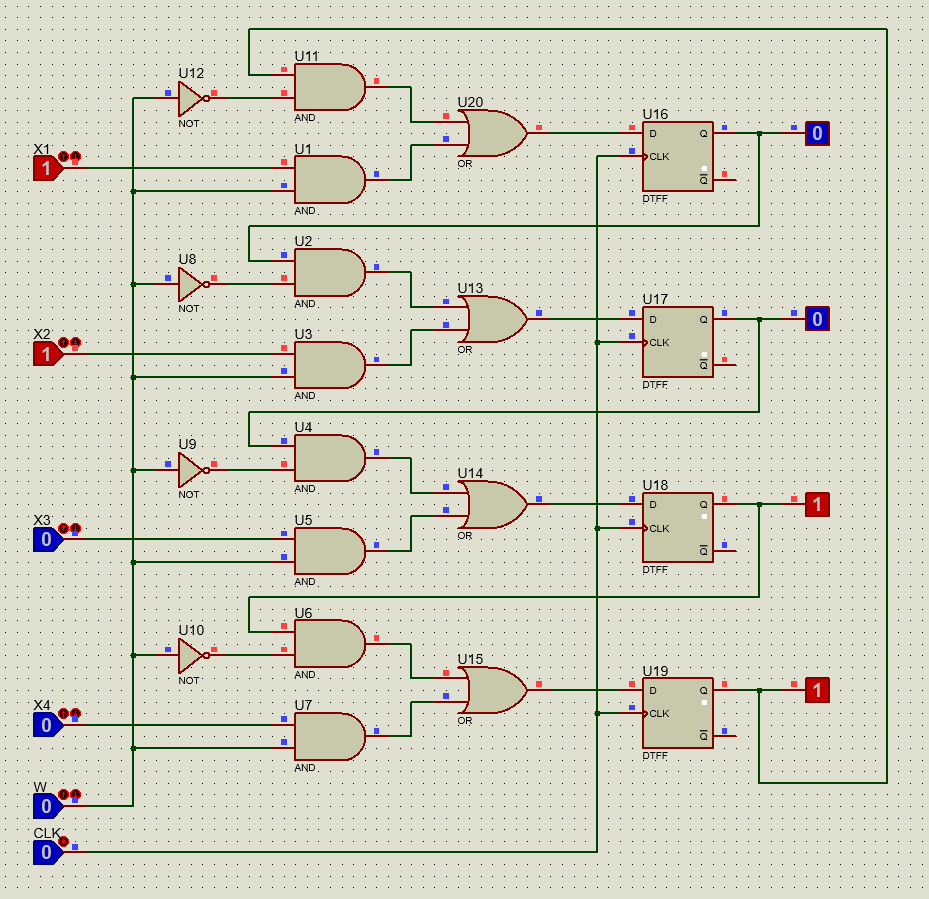


Рисунок 7 — Третий этап тестирования циклического сдвига вправо

После еще одного нажатия на тактовый сигнал (CLK), происходит сдвиг битов вправо на 1 разряд. Значение 0011 должно измениться на 1001. На скриншоте видно, что после третьего сдвига выходное значение стало 1001, что соответствует теоретическим ожиданиям.

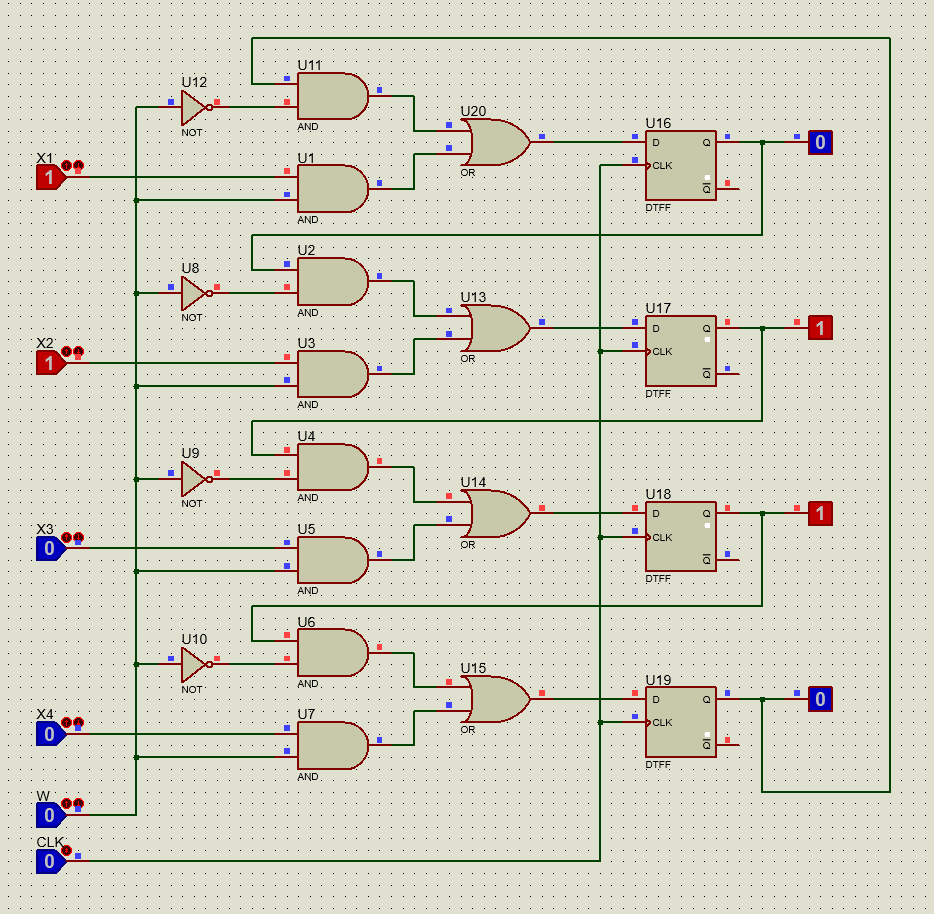


Рисунок 8 — Четвертый этап тестирования циклического сдвига вправо

После повторного нажатия на тактовый сигнал (CLK), происходит окончательный сдвиг битов вправо на 1 разряд, возвращая значение к исходному 1100. На скриншоте видно, что после четвертого сдвига выходное значение стало 1100, как и было изначально, что подтверждает корректность циклического сдвига.

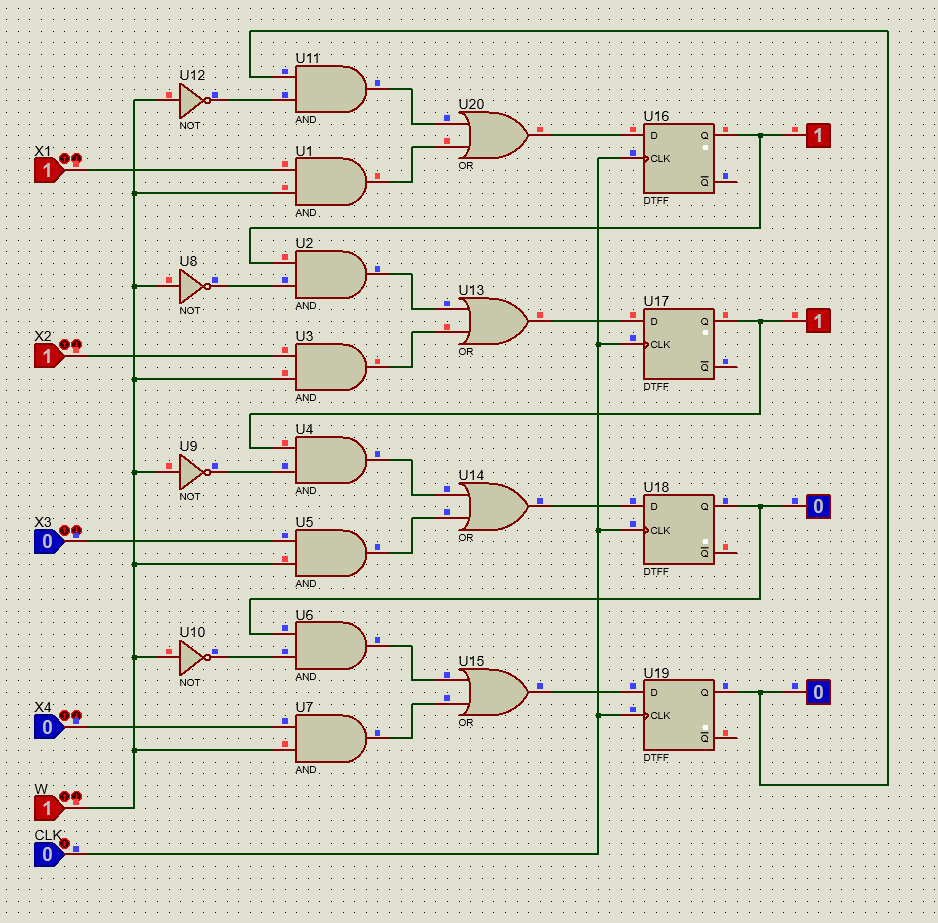


Рисунок 9 — Пятый этап тестирования циклического сдвига вправо

### Анализ результатов

Тестирование показало, что схема циклического сдвига вправо работает корректно, обеспечивая правильный сдвиг битов и циклическое повторение значений при каждом тактовом сигнале.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы успешно разработаны, реализованы и протестированы две цифровые схемы: асинхронный RS-триггер на логических элементах NOR и схема циклического сдвига вправо на 4 разряда с 6 входами и 4 выходами. Основной целью работы было изучение принципов работы этих схем и их реализация в программе Proteus.

Первой задачей было создание асинхронного RS-триггера. Эта схема была реализована с использованием двух логических элементов NOR. Триггер успешно прошел тестирование, продемонстрировав корректное поведение при подаче различных комбинаций входных сигналов. При подаче сигнала 10 триггер правильно устанавливал выходное значение в 1, а при подаче сигнала 01 — сбрасывал его в 0. Эти результаты подтвердили правильность работы триггера и соответствие теоретическим ожиданиям.

Второй задачей была реализация схемы циклического сдвига вправо. Для этой схемы использовались логические элементы AND, NOT и D-триггеры. В процессе тестирования схема успешно выполняла сдвиг битов вправо и демонстрировала циклическое повторение значений. Входное значение 1100 после четырех сдвигов вернулось к исходному значению, что подтвердило корректность работы схемы.

Процесс создания схем включал изучение теоретических основ, проектирование схем, настройку параметров компонентов и проведение симуляций для проверки их работоспособности. Важной частью работы было тестирование схем и анализ полученных результатов. Скриншоты, сделанные в процессе тестирования, наглядно продемонстрировали правильность работы схем.

Лабораторная работа позволила закрепить теоретические знания по схемотехнике и получить практические навыки в разработке и тестировании цифровых схем. Выполненная работа показала важность правильного проектирования и настройки схем, а также значимость этапа тестирования для подтверждения их корректной работы.

Достигнутые результаты подтверждают успешное выполнение поставленных задач и позволяют сделать вывод о готовности использовать приобретенные знания и навыки в дальнейшей учебной и профессиональной деятельности.