

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ  
НАПРАВЛЕНИЕ СИСТЕМНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

**курса «Функциональная схемотехника»**

**по теме: «Введение в проектирование цифровых интегральных  
схем»**

**Вариант № 6**

Выполнил студент:

Тюрин Иван Николаевич

группа: Р33102

Преподаватель:

Кустарев П. В.,

Васильев С.Е.

Санкт-Петербург, 2024 г.

# Содержание

Лабораторная работа № 2. Введение в проектирование цифро- вых интегральных схем	2
1. Введение . . . . .	2
2. Задание варианта № 6 . . . . .	2
3. Выполнение задания . . . . .	4
1. Часть 1 . . . . .	4
2. Часть 2 . . . . .	10
4. Вывод . . . . .	12

# Лабораторная работа № 2

## Введение в проектирование цифровых интегральных схем

### 1. Введение

Лабораторная работа №2 посвящена проектированию последовательностной логики на уровне регистровых передач с использованием языка описания аппаратуры Verilog HDL.

В первой части работы предлагается разработать несколько простых блоков цифровой последовательностной логики и объединить их для выполнения заданной функции в одно

функционирующее устройство. Во второй части работы предлагается разработать устройство, управляющее входным потоком данных с помощью одного из указанных алгоритмов обработки.

### 2. Задание варианта № 6

Вариант:

Вариант	Функция 1	FSM	Функция 2	Разрядности	Делитель частоты
6	COUNT_FREE	FSM_1	ARBITER	8 бит	5

Таблица 1.1: Вариант задания

#### Часть 1

1. В соответствии с требованиями вашего варианта, постройте микроархитектурные диаграммы:

- Счётчика;
- Сдвигового регистра;
- Конечного автомата;

- Делителя частоты;
2. Опишите ваши схемы на Verilog HDL;
  3. Подумайте о том, как протестировать каждую из ваших схем отдельно. Напишите тестовый план.
  4. Разработайте эталонные модели и тестовое окружение. Проведите тестирование всех ваших схем, доказывающее их работоспособность.
  5. Разработайте высокоуровневую диаграмму устройства, которое выполняет заданную вариантом функцию, на основе ранее разработанных простейших блоков;
  6. Подумайте о том, как протестировать устройство целиком. Напишите тестовый план.
  7. Разработайте эталонные модели и тестовое окружение. Проведите тестирование разработанного устройства, которое докажет его работоспособность;
  8. Подумайте над тем, как можно оптимизировать разработанную вами схему, при условии отсутствия ограничений;

## **Часть 2**

1. В соответствии с требованиями вашего варианта, постройте микроархитектурную диаграмму устройства.
2. Напишите тестовый план;
3. Разработайте эталонные модели и тестовое окружение. Проведите тестирование разработанного устройства, которое докажет его работоспособность;

## 3. Выполнение задания

### 3.1. Часть 1

#### Счётчик

В соответствии с заданием был разработан счетчик, микроархитектура которого представлена на рис. 1.1.

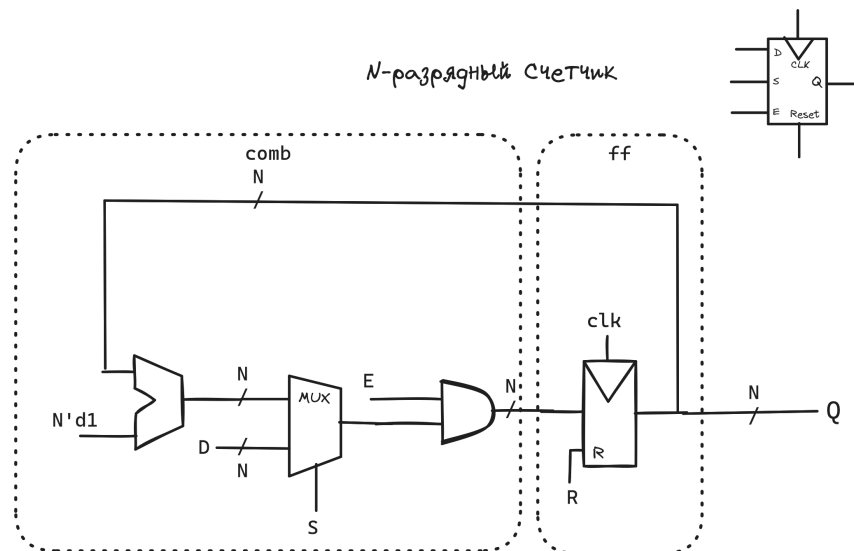


Рис. 1.1: Микроархитектура счетчика

Для тестирования модуля определим основные его свойства.

- С каждым тактом значение в счетчике увеличивается на 1.
- Счетчик корректно загружает данные.
- Счетчик приходит в корректное состояние после сброса.
- Счетчик правильным образом обрабатывает переполнение.

Тестовый план:

1. подключить модуль счетчика к тестирующему модулю,
2. сбросить счетчик до начального состояния,
3. включить счетчик,
4. загрузить в счетчик данные немного меньше чем максимальное значение в нем,

5. выполнить несколько отсчетов не доходя до переполнения с учетом включения и выключения и проверить, что значения увеличиваются верным образом,
6. выполнить еще один отсчет, проверить что произошло переполнение.

## Сдвиговый регистр

В соответствии с заданием был разработан сдвиговый регистр, микроархитектура которого представлена на рис. 1.2.

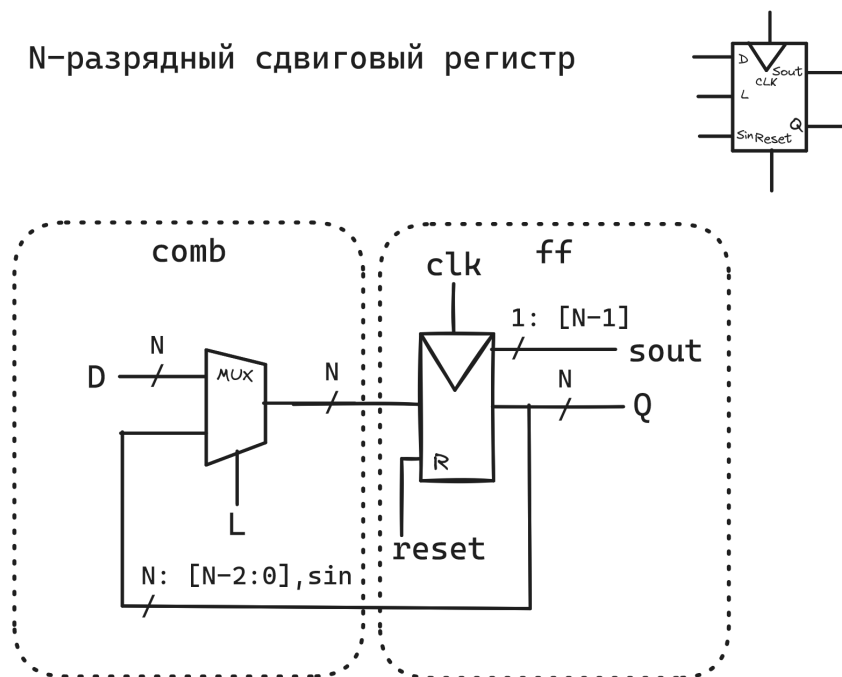


Рис. 1.2: Микроархитектура сдвигового регистра

Для тестирования модуля определим основные его свойства.

- Устройство возвращается к начальному состоянию при получении соответствующего сигнала, тем самым на выходы подаются 0.
- Устройство сохраняет значение со входа данных и подает его на выход при подаче сигнала для записи.
- На каждом такте значение в регистре сдвигается в нужную сторону, с учетом входного значения.

Тестовый план:

1. подключить модуль,
2. сбросить до начального состояния,

3. загрузить значение в регистр,
4. в течении нескольких тактов подавать на вход сдвига сигналы и контролировать корректность данных в устройстве,

## Конечный автомат

В соответствии с заданием был разработан конечный автомат, микроархитектура которого представлена на рис. 1.4.

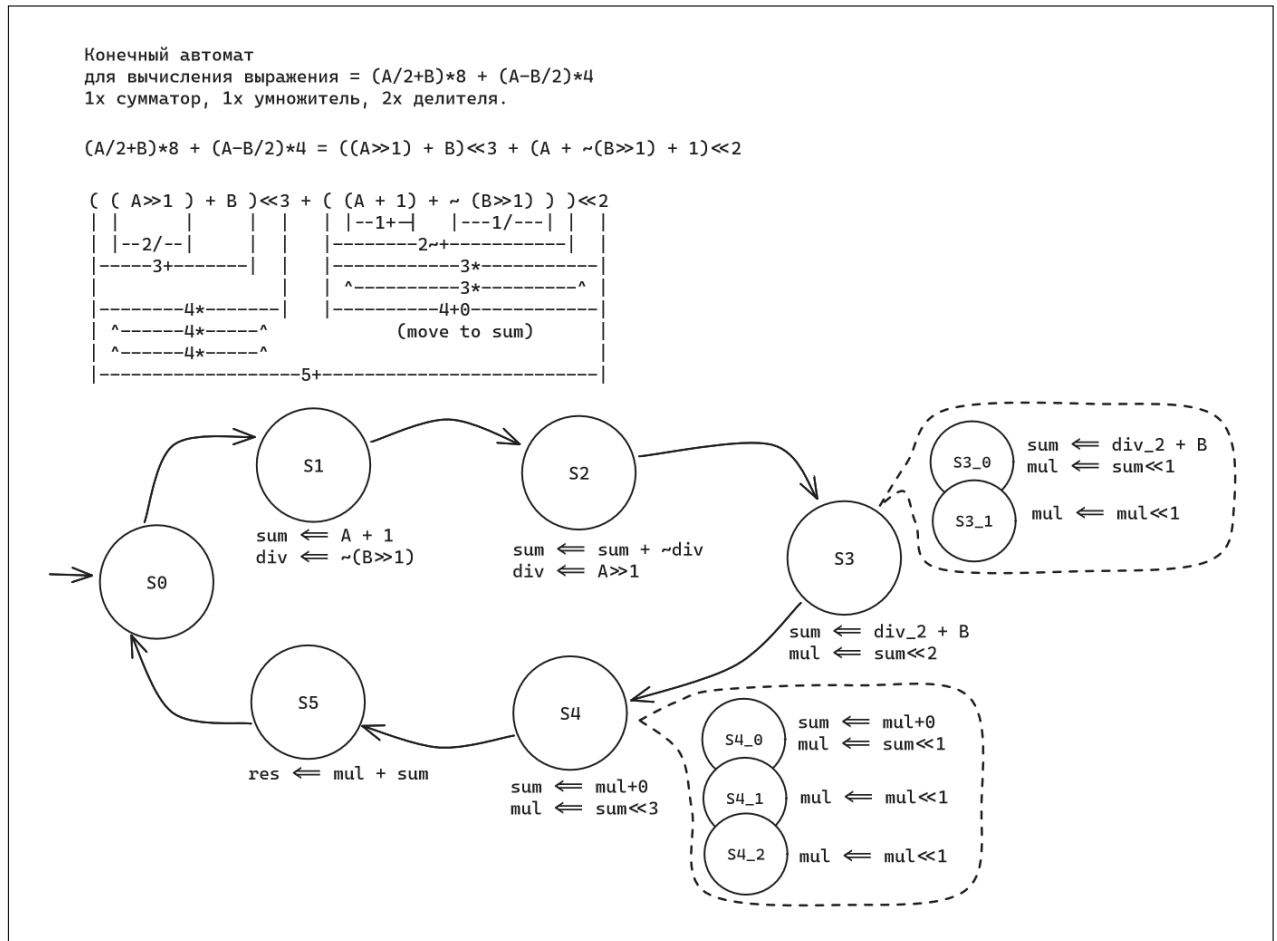


Рис. 1.3: Граф состояний и переходом между ними для FSM-1

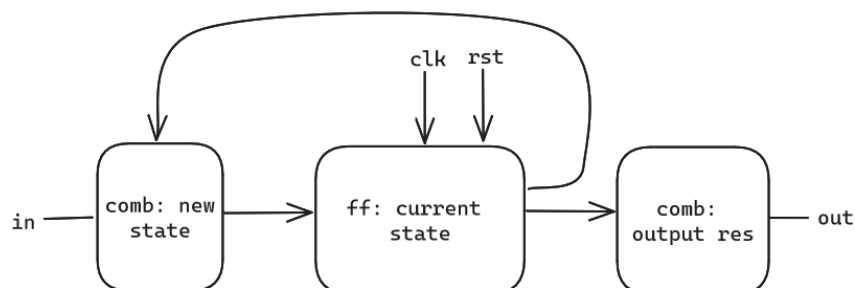


Рис. 1.4: Микроархитектура делителя частоты

Для тестирования модуля определим основные его свойства.

- Устройство приходит в изначальное состояние при получении сигнала асинхронного сброса.
- Устройство начинает вычисления при получении сигнала готовности данных.
- Устройство выдает результат после фиксированного количества отсчетов тактового генератора.

Тестовый план:

1. подключить модуль,
2. сбросить его до начального состояния,
3. выставить данные на линию данных для каждого аргумента,
4. выставить сигнал готовности данных на соответствующую линию,
5. отсчитать 10 тактов (количество состояний до получения результата),
6. проверить соответствие результата вычислений и ожидаемого значения.

## Делитель частоты

В соответствии с заданием был разработан делитель частоты, микроархитектура которого представлена на рис. 1.5.

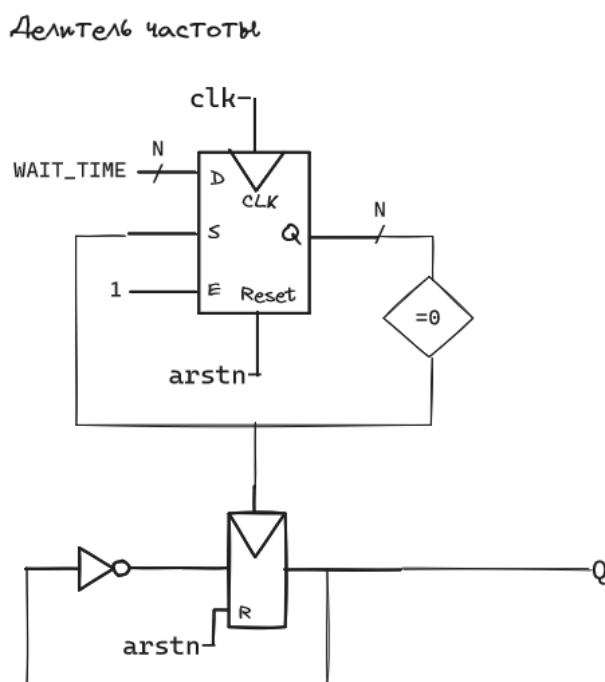


Рис. 1.5: Микроархитектура делителя частоты



Для тестирования модуля определим основные его свойства.

- Устройство приходит в начальное состояние (0 на выходе) при сигнале сброса.
- Устройство изменяет значение на выходе строго каждые N-указанных тактов входного сигнала.

Тестовый план:

1. подключить тестируемый модуль,
2. сбросить состояние модуля до начального,
3. проверить, что строго через указанные N тактов состояние на выходе модуля изменяется на противоположное.

### Сложно функциональный блок COUNT\_FREE

В соответствии с заданием был разработан сложно функциональный блок выполняющий функцию COUNT\_FREE, в основе которой лежит конечный автомат. Диаграмма состояний этого конечного автомата представлена на рис. 1.7. Микроархитектура модуля с условными обозначениями представлена на рис. 1.6.

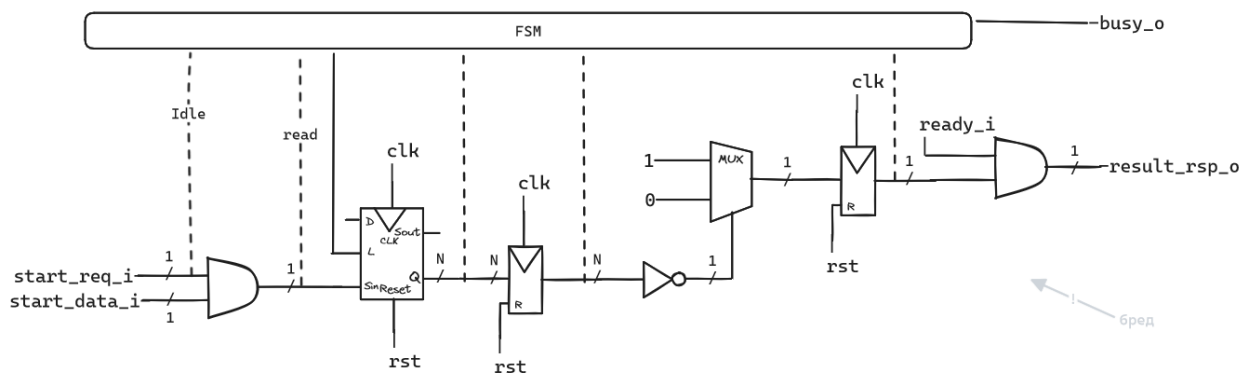


Рис. 1.6: Микроархитектура модуля реализующего функцию COUNT\_FREE.

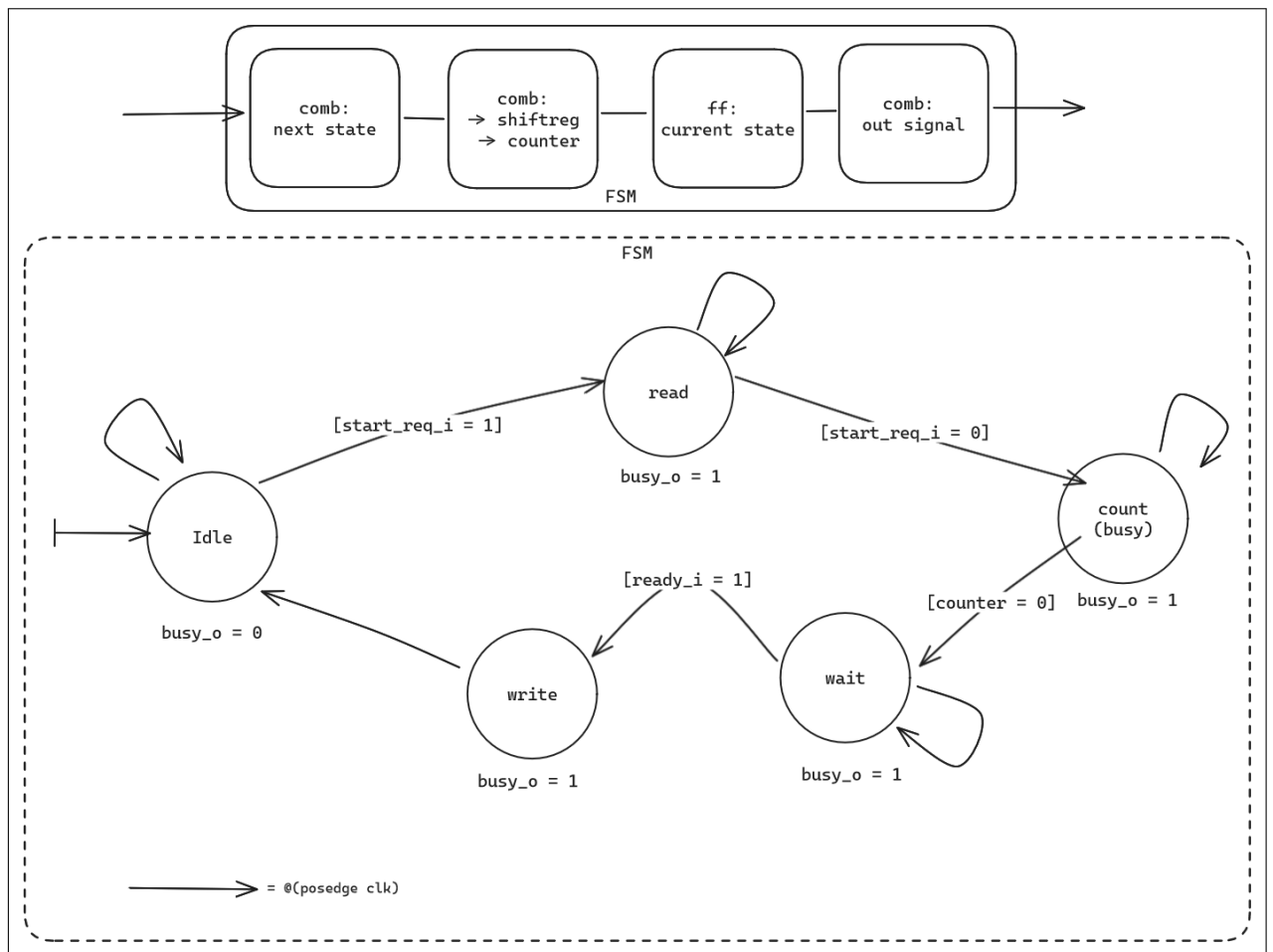


Рис. 1.7: Граф состояний и переходов между ними для конечного автомата реализующего функцию COUNT\_FREE.

Для тестирования модуля определим основные его свойства.

- Устройство приходит в начальное состояние после получения сигнала сброса.
- Устройство начинает считывать данные при высоком уровне сигнала на начало записи и прекращает при низком уровне этого сигнала.
- Устройство сообщает о своей занятости во время считывания данных и ожидания требуемого количества тактов.
- Устройство сообщает о готовности после прошествия требуемого количества тактов, считанного ранее.

Тестовый план:

1. подключить тестируемый модуль,
2. сбросить состояние модуля до начального,

3. выставить на линию сигнала начала записи логическую 1 и проверить состояние занятости на выходе,
4. произвести запись данных в устройство и проверить состояние занятости на выходе,
5. выставить на линию сигнала начала записи логический 0 и проверить состояние занятости на выходе,
6. отсчитать нужное количество тактов и проверить состояние готовности на линии выхода.

## 3.2. Часть 2

Для реализации сложнофункционального блока были рассмотрены два подхода: конечный автомат с состояниями, соответствующими обработке конкретного транзакта (см. рис. 1.9), и схема без выраженных состояний и вектором запросов от представленных транзактов, микроархитектура которого представлена на рис. 1.8.

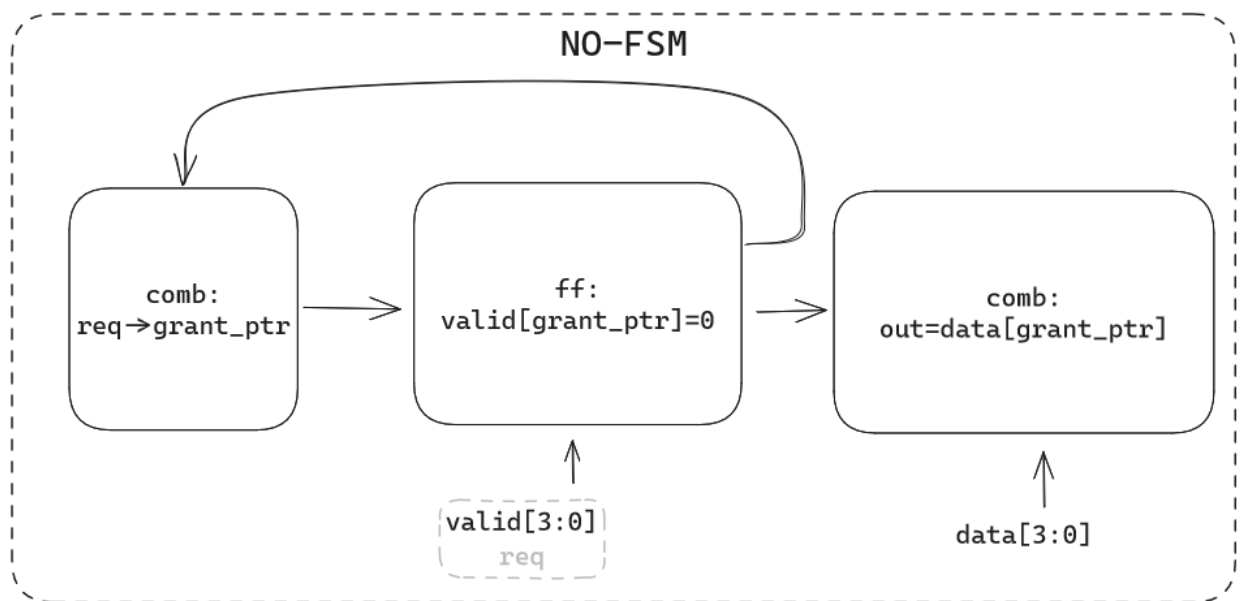


Рис. 1.8: Микроархитектура сложнофункционального блока "Арбитер выполненная без использования FSM

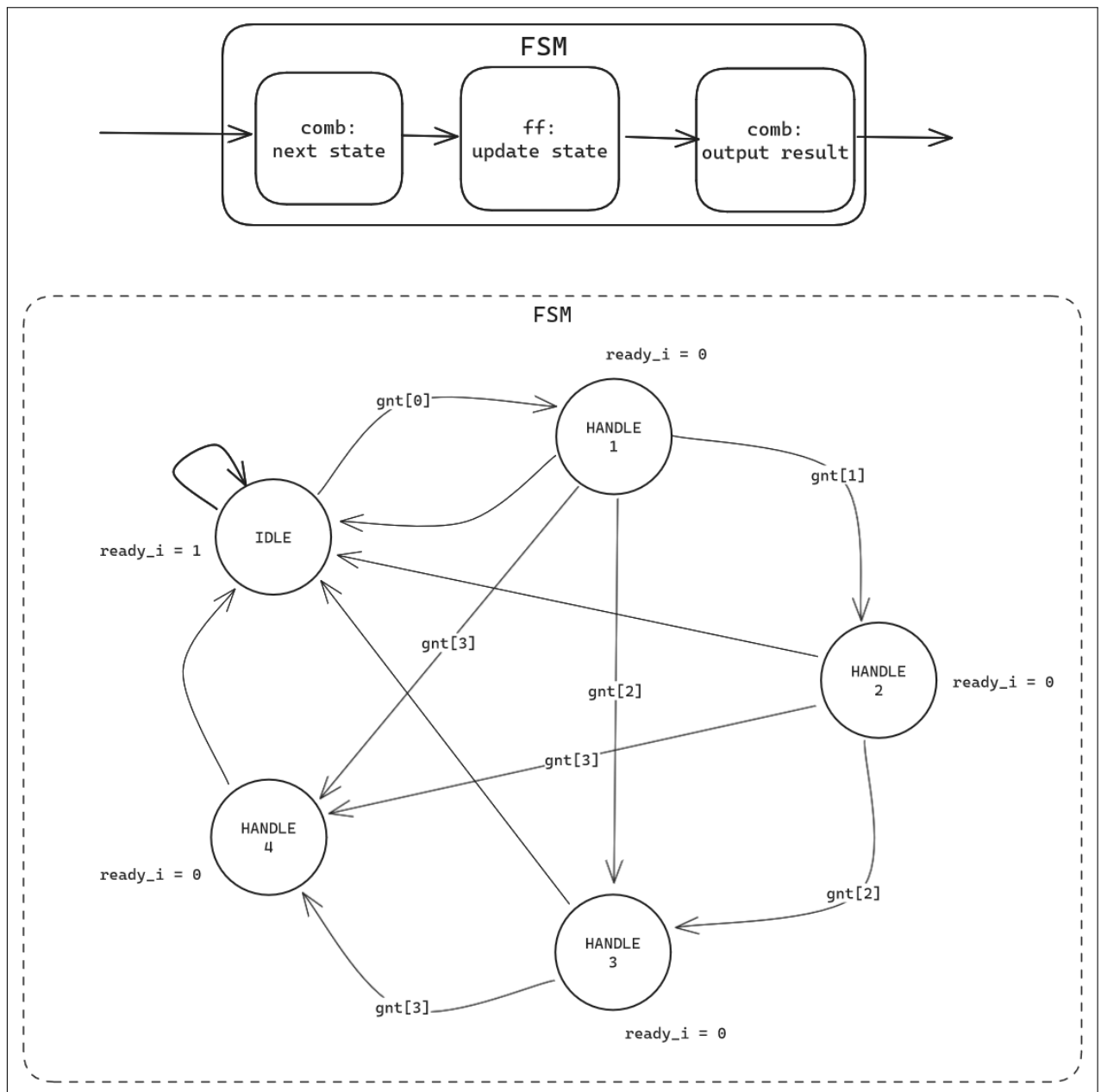


Рис. 1.9: Граф состояний и переходов между ними для возможной реализации сложнофункционального блока "Арбитер"

Для тестирования модуля определим основные его свойства.

- Устройство приходит в начальное состояние после получения сигнала сброса.
- Устройство получает на входную шину данных числа.
- Устройство получает сигналы валидности на соответствующую шину.
- устройство начинает выдавать данные с шины данных в соответствии с определенным порядком транзактов.
- Устройство находится в состоянии Ready, если не передает данные.

- Устройство выдает сигнал Valid на выходную линию при выполнении передачи данных.

Тестовый план:

1. подключить тестируемый модуль,
2. сбросить состояние модуля до начального,
3. для каждого транзакта выставить необходимые данные и сигнал готовности на входные шины,
4. начиная со следующего такта регистрировать данные на выходной шине данных и проверить выходной сигнал валидности.

## 4. Вывод

В результате выполнения работы были реализованы на языке SystemVerilog все требуемые модули. При этом для корректного их описания на языке пришлось разработать микроархитектурные диаграммы и обратиться к учебным материалам с целью узнать принципы описания конечных автоматов на языке описания аппаратуры.

На примере реализации модуля «Арбитер» выяснили, что можно использовать разные подходы к реализации алгоритмов в аппаратуре, не все они одинаково очевидны и просты в реализации.

Во время разработки тестовых сценариев было выяснено, что задача составления хороших тестов довольно сложная, но очень важная для валидации корректности работы цифровой схемы.