

Лабораторная работа №2. Последовательностная логика

2.1 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.

2.2 Описание лабораторной работы

2.2.1 Указания к выполнению

Лабораторная работа посвящена знакомству с техниками описания последовательностных схем и блоков на RTL-уровне с использованием языка Verilog HDL. Работа выполняется в Vivado Design Suite.

2.2.2 Порядок выполнения работы

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;

Отчет должен включать:

1. Титульный лист.
2. Задание в соответствии с вариантом.
3. Все разработанные микроархитектурные диаграммы для ваших схем.
4. Описание работы всех разработанных блоков, начиная с подачи входных данных и заканчивая получением результата.
5. Тестовые планы для каждого разработанного блока.
6. Выводы по работе. В выводе опишите подробно, как вы подходили к решению каждой задачи, какие у вас возникли вопросы и трудности и как вы разобрались со всеми сложностями.

3. Варианты заданий

3.1 Счётчик

Необходимо описать простейший последовательностный счетчик. Значение этого счетчика должно инкрементироваться на 1 каждый фронт тактового сигнала, если сигнал разрешения активен. Необходимо также предусмотреть возможность асинхронного сброса. Разрядность счетчика устанавливается в соответствии с вариантом.

3.2 Сдвиговый регистр

Необходимо описать сдвиговый регистр с параллельным входом и последовательным выходом. Каждый фронт тактового сигнала, при наличии активного сигнала разрешения, выполняется операция сдвига вправо. Крайний правый выдвинутый разряд должен подаваться на выход схемы. Необходимо также предусмотреть возможность асинхронного сброса. Разрядность регистра устанавливается в соответствии с вариантом.

3.3 Конечные автоматы

У вас имеется ограниченное число блоков сумматора, умножителя на 2 и делителя на 2. На их основе вам необходимо реализовать заданные функции. Разрядность операндов указана в варианте. Разрядность результата определяется студентом. В рамках этого задания, запрещено пользоваться операторами -, * и /, а так же соответствующим им встроенным функциям-примитивам языка Verliog

3.3.1 FSM_1

Один сумматор, один умножитель, два делителя.

$$(A/2+B)*8 + (A-B/2)*4$$

3.3.2 FSM_2

Один сумматор, один умножитель.

$$((A+B)*2 + B)$$

3.3.3 FSM_3

Один сумматор.

$$A*4 + B*2$$

3.3.4 FSM_4

Два сумматора, один умножитель, один делитель.

$$((A + B)*4 + B)/2 + (B/2 + A*4)$$

3.4 Делитель частоты

Необходимо описать схему делителя частоты, который уменьшает входную частоту в то количество раз, которое указано у вас в варианте.

3.5 Первая функция

В рамках первой реализуемой функции необходимо описать модуль, выполняющий заданную функцию. Все внутренние элементы памяти вашего модуля должны тактироваться от делителя частоты и обладать сигналами разрешения и асинхронного сброса. Значение уменьшения частоты указано в варианте. По-возможности старайтесь использовать простейшие модули из предыдущей части лабораторной работы. Некоторые модули потребуют модификации.

3.5.1 Функция “COUNT_ONES”

За заданный промежуток времени вам необходимо рассчитать количество фронтов тактового сигнала из делителя частоты. После расчёта на выходные порты необходимо подать число единиц в двоичном представлении рассчитанного числа. Необходимые сигналы в интерфейсе вашего модуля:

Порт	Тип	Описание
clk	Вх.	Сигнал тактовой частоты.
rst	Вх.	Асинхронный сигнал сброса.
measure_req_i	Вх.	Сигнал регулирующий временной промежуток. При подаче в первый раз запускает временной отсчёт. При подаче во второй раз останавливает временной отсчёт. Именно таким образом регулируется требуемый промежуток времени.
result_rsp_o	Вых.	Сигнал готовности результата. Выставляется в высокий уровень тогда и только тогда, когда на шине данных установлен корректный результат. Держится в высоком состоянии ровно один период тактового сигнала.
busy_o	Вых.	Сигнал занятости устройства.
result_data_o	Вых.	Шина данных результата.

3.5.2 Функция “COUNT_ZEROES”

За заданный промежуток времени вам необходимо рассчитать количество спадов тактового сигнала из делителя частоты. После расчёта на выходные порты необходимо подать число нулей в двоичном представлении рассчитанного числа. Необходимые сигналы в интерфейсе вашего модуля:

Порт	Тип	Описание
clk	Вх.	Сигнал тактовой частоты.
rst	Вх.	Асинхронный сигнал сброса.

measure_req_i	Вх.	Сигнал регулирующий временной промежуток. При подаче в первый раз запускает временной отсчёт. При подаче во второй раз останавливает временной отсчёт.
ready_i	Вх.	Сигнал готовности внешнего устройства принять результат.
result_rsp_o	Вых.	Сигнал готовности результата. Выставляется в высокий уровень тогда и только тогда, когда на шине данных установлен корректный результат. Держится в высоком состоянии ровно один период тактового сигнала, когда внешнее устройство готово принять результат; в противном случае держится в высоком состоянии до тех пор, пока устройство не будет готово принять результат.
busy_o	Вых.	Сигнал занятости устройства.
result_data_o	Вых.	Шина данных результата.

3.5.3 Функция “COUNT_POSEDGE”

За заданный промежуток времени вам необходимо рассчитать количество фронтов тактового сигнала из делителя частоты. После расчета необходимо выполнить процедуру последовательной выгрузки полученного значения через однобитный порт данных. Необходимые сигналы в интерфейсе вашего модуля:

Порт	Тип	Описание
clk	Вх.	Сигнал тактовой частоты.
rst	Вх.	Асинхронный сигнал сброса.
measure_req_i	Вх.	Сигнал регулирующий временной промежуток. При подаче в первый раз запускает временной отсчёт. При подаче во второй раз останавливает временной отсчёт.
ready_i	Вх.	Сигнал готовности внешнего устройства принять результат.
result_rsp_o	Вых.	Сигнал готовности результата. Выставляется в высокий уровень тогда и только тогда, когда на шине данных установлен корректный результат. Держится в высоком состоянии ровно один период тактового сигнала, когда внешнее устройство готово принять результат; в противном случае держится в высоком состоянии до тех пор, пока устройство не будет готово принять результат.
busy_o	Вых.	Сигнал занятости устройства.
result_data_o	Вых.	Однобитный порт данных результата. Каждый новый фронт

		тактового сигнала на этом выходе должен быть следующий разряд результата.
--	--	---

3.5.4 Функция “COUNT_FREE”

Необходимо после прошествия определенного количества времени сформировать однобитный сигнал запроса. Информацию о количестве тактов, после которого необходимо формировать сигнал, устройство получает по однобитному последовательностному порту запроса. Необходимые сигналы в интерфейсе вашего модуля:

Порт	Тип	Описание
clk	Вх.	Сигнал тактовой частоты.
rst	Вх.	Асинхронный сигнал сброса.
start_req_i	Вх.	Сигнал запроса. Представляет собой сигнал валидности для однобитного сигнала входных данных.
start_data_i	Вх.	Однобитный сигнал данных, представляющий собой один разряд исходного числа.
ready_i	Вх.	Сигнал готовности внешнего устройства принять результат.
result_rsp_o	Вых.	Сигнал готовности результата. Выставляется в высокий уровень тогда и только тогда, когда на шине данных установлен корректный результат. Держится в высоком состоянии ровно один период тактового сигнала, когда внешнее устройство готово принять результат; в противном случае держится в высоком состоянии до тех пор, пока устройство не будет готово принять результат.
busy_o	Вых.	Сигнал занятости устройства.

3.6 Сложно-функциональный блок

В рамках второй части задания необходимо реализовать модуль, выполняющий определенную процедуру обработки входного потока данных. Все внутренние элементы памяти вашего модуля должны тактироваться от делителя частоты и обладать сигналами разрешения и асинхронного сброса.

3.6.1 Модуль “ARBITER”

На входе у модуля имеются четыре независимых интерфейса подачи транзакций. По каждому из этих интерфейсов в любой момент времени может поступить транзакция, которая должна быть передана на единственный выходной порт. В случае, если в модуль одновременно приходят несколько транзакций, то они должны быть обработаны по алгоритму Round-Robin. Интерфейс подачи транзакции должен включать в себя шину данных и соответствующий ей сигнал валидности данных. Выходной интерфейс также включает в себя шину данных и сигнал готовности результата. Помимо этого в вашем модуле необходимо реализовать сигнал готовности устройства принять данные на обработку. Разрядность данных устанавливается в соответствии с вариантом.

3.6.2 Модуль “BUFFER_LRU”

Модуль-буффер из восьми элементов, в котором вытеснение происходит по алгоритму LRU. Интерфейс подачи транзакции должен включать в себя шину данных и соответствующий ей сигнал валидности данных. Выходной интерфейс представляет собой текущее состояние всех восьми элементов буффера.

3.6.3 Модуль “BUFFER_MRU”

Модуль-буффер из восьми элементов, в котором вытеснение происходит по алгоритму MRU. Интерфейс подачи транзакции должен включать в себя шину данных и соответствующий ей сигнал валидности данных. Выходной интерфейс представляет собой текущее состояние всех восьми элементов буффера.

3.6.4 Модуль “FIFO”

Модуль очереди из восьми элементов, работающей по алгоритму FIFO. Интерфейс записи должен включать в себя шину данных и соответствующий ей сигнал валидности данных. Интерфейс чтения также представляет собой комбинацию сигнала подачи запроса и шину данных результата. Вам необходимо учесть и корректно обработать ситуацию конфликта сигналов чтения и записи.

3.6.5 Модуль “LIFO”

Модуль очереди из восьми элементов, работающей по алгоритму LIFO. Интерфейс записи должен включать в себя шину данных и соответствующий ей сигнал валидности данных. Интерфейс чтения также представляет собой комбинацию сигнала подачи запроса и шину данных результата. Вам необходимо учесть и корректно обработать ситуацию конфликта сигналов чтения и записи.

3.7 Таблица с вариантами

Вариант	Функция 1	FSM	Функция 2	Разрядности	Делитель частоты
1	COUNT_ONES	FSM_4	ARBITER	16 бит	2
2	COUNT_ZEROES	FSM_3	LRU	8 бит	4
3	COUNT_POSEDGE	FSM_2	MRU	40 бит	5
4	COUNT_FREE	FSM_1	FIFO	32 бит	10
5	COUNT_ZEROES	FSM_2	LIFO	24 бита	4
6	COUNT_FREE	FSM_1	ARBITER	8 бит	5
7	COUNT_POSEDGE	FSM_3	LIFO	16 бит	10
8	COUNT_ONES	FSM_3	FIFO	24 бита	2
9	COUNT_FREE	FSM_4	LRU	32 бита	5
10	COUNT_FREE	FSM_2	MRU	40 бит	5