

Часть 1

Задача 1

1. С использованием типов данных и конструкций расширения System Verilog создайте описание параметризованного устройства (ALU) – *описание и варианты приведены в приложении А.*
2. С использованием типов данных и конструкций расширения System Verilog для созданного устройства разработайте тест класса 2 (с самопроверкой).
 - a. Исходные данные и ожидаемые данные для проверки должны считываться из файлов. Типы команд считывания из файлов – любые.
 - b. Тест должен обеспечивать проверку всех режимов работы устройства (сброс, сигнал Valid =0 =1) и доказывать правильность работы разработанного устройства (работает в соответствии с алгоритмом)
3. В пакете ModelSim проведите моделирование созданного устройства.
4. Интегрируйте созданное устройство как библиотечный компонент в PD (библиотечная папка–kurs).

Задача 2

1. С использованием типов данных и конструкций расширения System Verilog создайте описание устройства (STREAM_GEN), которое будет формировать потоковые данные по Avalon ST (Stream) интерфейсу. Тип формируемых данных – выход счетчика на сложение/вычитание с переменным шагом счета. Устройство должно позволять настраивать режим его работы (направление счета - на сложение/вычитание, шаг счета) по интерфейсу Avalon MM slave.
 - a. Выводы устройства (имена выводов модуля м.б. выбраны любыми, **рекомендуется** выбрать имена, облегчающие интеграцию с Platform Designer (PD))
 - b. Канал выходных потоковых данных:
 - i. Data[N:1] - информационные выходы.
 - ii. Должен быть ориентирован на использование Avalon ST (Stream) интерфейса в Platform Designer (PD).
 1. Выход Valid постоянно =1 (т.к. выходные потоковые данные формируются непрерывно)
 2. Вход Ready: если =1 – идет формирование выходных потоковых данных; если = 0 – счетчик, формирующий потоковые данные, останавливается.
 - iii. Вход: тактового сигнала – clk.
 - iv. Вход: сигнала синхронного сброса – sreset.
 - c. Выводы интерфейса Avalon MM slave.
2. С использованием типов данных и конструкций расширения System Verilog для созданного устройства разработайте тест класса 1.
3. В пакете ModelSim проведите моделирование созданного устройства. Тест должен обеспечивать проверку всех режимов работы устройства (сброс, сигнал Ready =0 =1, запись конфигурационных данных по Avalon MM slave интерфейсу) и доказывать правильность работы разработанного устройства (работает в соответствии с алгоритмом)
4. Интегрируйте созданное устройство как библиотечный компонент в PD (библиотечная папка–kurs).

Задача 3

1. С использованием типов данных и конструкций расширения System Verilog создайте **параметризованное** описание устройства (FSM_master), которое будет передавать по интерфейсу Avalon MM Master настройки для компонентов STREAM_GEN
 - a. Количество компонентов STREAM_GEN в системе, для всех вариантов, = 4
 - i. Адрес **каждого** компонента STREAM_GEN в системе задается параметром
 - ii. Режим работы **каждого** компонента STREAM_GEN в системе (счет на сложение/вычитание, шаг счета) задается параметром.

- b. Выводы устройства (имена выводов модуля м.б. выбраны любыми, **рекомендуется** выбрать имена, облегчающие интеграцию с Platform Designer (PD))
 - i. Выводы интерфейса Avalon MM Master.
 - ii. Вход: тактового сигнала – clk.
 - iii. Вход: сигнала синхронного сброса – sreset.
 2. С использованием типов данных и конструкций расширения System Verilog для созданного устройства разработайте тест класса 1.
 3. В пакете ModelSim проведите моделирование созданного устройства. Тест должен обеспечивать проверку всех режимов работы устройства (сброс) и доказывать правильность работы разработанного устройства (работает в соответствии с алгоритмом – передает по шине Avalon MM по адресам, заданным параметрами, данные, заданные параметрами.)
 4. Интегрируйте созданное устройство как библиотечный компонент в PD (библиотечная папка–kurs).

Задача 4

1. В PD создайте описание системы, включающей:
 - a. модуль тактового сигнала
 - b. ALU
 - c. Четыре устройства STREAM_GEN .
 - d. FSM_master
2. С использованием типов данных и конструкций расширения System Verilog для созданного устройства разработайте тест класса 1.
3. В пакете ModelSim проведите моделирование созданного устройства. Тест должен обеспечивать проверку всех режимов работы устройства (сброс) и доказывать правильность работы разработанного устройства (работает в соответствии с алгоритмом: настраивает устройства STREAM_GEN, обеспечивает обработку данных в соответствии с алгоритмом ALU)

Задача 5

1. Настройте In-System Sources & Probes (для формирования сигнала sreset и визуализации выходных данных устройства)
2. Настройте Signal TapII для демонстрации временных диаграмм аналогичных временным диаграммам полученным в задаче 4(этап 3).
3. На плате MAX10NEEK, с помощью In-System Sources & Probes и Signal TapII покажите и докажите правильность работы разработанного устройства (работает в соответствии с алгоритмом: настраивает устройства STREAM_GEN, обеспечивает обработку данных в соответствии с алгоритмом ALU)

Часть 2

Задача 1

1. Скопируйте систему, разработанную в Части 1, в новую папку.
2. В скопированной системе замените устройство FSM master процессором NIOSII (ОБРАТИТЕ ВНИМАНИЕ - потребуется добавить в систему модуль памяти для хранения программ и данных). Настройка процессора: с JTAG Debug или без него может быть выбрана любой (целесообразно на первом этапе включить с JTAG Debug для возможности отладки на плате).
3. Создайте два варианта программы настройки STREAM_GEN
 - a. с использованием указателей;
 - b. с использованием драйверов (макрофункций) или символических имен из system.h.
4. Настройте In-System Sources & Probes (для формирования сигнала sreset и визуализации выходных данных устройства)
5. Настройте Signal TapII для демонстрации временных диаграмм аналогичных временным диаграммам полученным в задаче 4 (этап 3) части 1.
6. На плате MAX10NEEK (для каждого варианта программы настройки STREAM_GEN), с помощью In-System Sources & Probes и Signal TapII покажите и докажите правильность работы разработанного устройства (работает в соответствии с алгоритмом: настраивает устройства STREAM_GEN, обеспечивает обработку данных в соответствии с алгоритмом ALU)

ПРИЛОЖЕНИЕ А

(варианты устройств ALU)

Вариант 1 – поиск, на каждом такте, максимума и минимума среди 4 входных потоков данных. Устройство имеет 4 входных потока данных и два выходных потока данных (минимальные и максимальные значения). Разрядность данных – 8 бит.

Вариант 2 – поиск, на каждом такте, медианы среди 4 входных потоков данных (медиана вычисляется как среднее между элементами 2 и 3 в отсортированном массиве из 4 элементов). Устройство имеет 4 входных потока данных и один выходной поток данных. Разрядность данных – 8 бит.

Вариант 3 – поиск, на каждом такте, максимума среди 4 входных потоков данных и формирование 4 выходных потоков данных по алгоритму: входные данные – (минус) найденное максимальное значение (обратите внимание – один из каналов, где на входе было максимальное значение, на выходе даст 0). Устройство имеет 4 входных потока данных и четыре выходных потока данных. Разрядность данных – 8 бит.

Вариант 4 – поиск, на каждом такте, минимума среди 4 входных потоков данных и формирование 4 выходных потоков данных по алгоритму: входные данные – (минус) найденное минимальное значение (обратите внимание – один из каналов, где на входе было минимальное значение, на выходе даст 0). Устройство имеет 4 входных потока данных и четыре выходных потока данных. Разрядность данных – 8 бит.

Вариант 5 – на каждом такте, сумма квадратов среди 1 и 2 входных потоков данных – (минус) сумма квадратов среди 3 и 4 входных потоков данных. Устройство имеет 4 входных потока данных и один выходной поток данных. Разрядность данных – 8 бит.

Выводы устройства (имена выводов модуля м.б. выбраны любыми, **рекомендуется** выбрать имена, облегчающие интеграцию с Platform Designer (PD)):

- Каналы входных данных
 - Каналы должны быть ориентированы на использование Avalon ST (Stream) интерфейса в Platform Designer (PD) с поддержкой сигналов Ready и Valid
 - На входах данных д.б использованы регистры.
 - по входному сигналу Valid =1 осуществляется запись данных во входные регистры
 - Выходной сигнал Ready постоянно равен 1
- Каналы выходных данных:
 - Должны быть ориентированы на использование Avalon Conduit интерфейса в Platform Designer (PD).
 - На выходе д.б использован регистр
- Вход: тактового сигнала – clk.
- Вход: сигнала синхронного сброса – sreset.