**(10 февраля) Вводная работа.**

**Знакомство с библиотекой** **SystemC. Создание первого проекта Hello world.**

Программа:

1. В среде NetBeans создать новый проект.
2. В настройках компоновщика (Linker) для проекта подключить библиотеку SystemC (через PkgConfig)
3. Используя исходный код программы из примера /usr/share/doc/systemc/examples/sysc/simple\_fifo, создать программу, моделирующую работу очереди.

<описание архитектуры проекта>

1. Выполнить компиляцию проекта, убедиться в правильности подключения библиотеки SystemC.
2. Запустить созданное приложение, наблюдая результаты моделирования устройства в консоли.
3. Модифицировать созданное устройство, изменив тип данных, с которыми работает очередь, на int. Соответствующим образом модифицировать тест.

**(17 февраля) Первая работа.**

**Знакомство с описанием синхронных устройств на RTL уровне на языке SystemC.**

Программа:

1. Скопировать в локальную папку проект с примером регистра.
2. Выполнить компиляцию проекта. Запустить созданное приложение, наблюдая результаты моделирования устройства в консоли. Проверить правильность работы устройства, открыв сгенерированный vcd файл в GTKWave.
3. Разработать собственные устройства: счетчик и сдвигающий регистр, - с дополнительными функциями по индивидуальному заданию.
4. Создать тесты с самопроверкой для всех основных и дополнительных функций разработанных устройств.

Основные входы/выходы счетчика:

* clk
* areset
* sreset\_n
* dout

Основные входы/выходы сдвигающего регистра:

* clk
* areset
* sreset
* cin
* cout
* dout

Варианты индивидуальных заданий

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № п/п | counter | | | shift\_register | | |
| ena | load | up/down | ena | load | left/right |
| 1 | + |  |  |  | + |  |
| 2 | + |  |  |  |  | + |
| 3 |  | + |  | + |  |  |
| 4 |  | + |  |  |  | + |
| 5 |  |  | + | + |  |  |
| 6 |  |  | + |  | + |  |

**(3 марта) Вторая работа.**

**Знакомство с описанием комбинаторных устройств на языке SystemC.**

Программа:

1. Скопировать в локальную папку проект с примером регистра (с подсчетом четности).
2. Выполнить компиляцию проекта. Запустить созданное приложение, наблюдая результаты моделирования устройства в консоли. Проверить правильность работы устройства, открыв сгенерированный vcd файл в GTKWave.
3. Разработать конечный автомат Мура по индивидуальному заданию.

<схема КА с делением на синхронную и комбинаторную части>

1. Создать тесты с самопроверкой для всех переходов конечного автомата.

<таблица с вариантами (таблица истинности для переходов и выходов КА)>

**(10 марта, 17 марта) Третья работа.**

**Знакомство с описанием параметризованных устройств на языке SystemC.**

Задачи:

1. Создать потактовое описание FIFO, работающего с 8-разрядными словами. Глубина FIFO должна задаваться в конструкторе.

Обязательные входы:

- clk

- sreset\_n

- data\_in

- push

- pop

Обязательные выходы:

- data\_out

- empty

- full

2. Создать на основе класса FIFO шаблон класса FIFOParam, в котором параметром выступает тип данных.

3. Унаследовать от созданного шаблона класса FIFOParam, шаблон класса FIFOParamExtended с дополнительными выходами:

- almost\_empty

- almost\_full

Для всех трёх описаний необходимо написать соответствующие тесты.

**Часть 2. Курсовое проектирование. Разработка потактовой модели процессора**

**Раздел 1.** Спецификация компонентов процессорной системы.

Обязательные компоненты системы: АЛУ, контроллер памяти, регистровый файл.

Дополнительные компоненты системы по индивидуальному заданию:

- контроллер прямого доступа к памяти (DMA)

- поддержка виртуальной памяти (MMU)

- поддержка защиты памяти (MPU)

- кэш-память команд/данных ($I/D)

- интерфейс ввода/вывода (GPIO)

Варианты индивидуальных заданий

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | DMA | MMU | MPU | $I/D | GPIO |
| 1 | + | + |  |  | + |
| 2 | + |  | + |  | + |
| 3 | + |  |  | + | + |
| 4 |  | + | + |  | + |
| 5 |  | + |  | + | + |
| 6 |  |  | + | + | + |

Предусмотреть отладочный режим работы модулей.

**Раздел 2.** Спецификация системного окружения для отладки и тестирования процессора.

Допущения:

Время доступа к внешней памяти — от 7 процессорных тактов.

**Раздел 3.** Спецификация системы команд процессора типа RISC, исходя из выбранного набора компонентов процессорной системы и целевых алгоритмов.

**Раздел 4.** Разработка описания ядра процессора, компонентов процессорной системы и элементов системного окружения на языке SystemC.

**Раздел 5.** Модульное тестирование разработанной системы.

**Раздел 6.** Создание программы в машинных кодах для реализации заданных алгоритмов. Демонстрация работоспособности процессора. Оценка эффективности выполнения алгоритмов.

Алгоритмы:

1. Сортировка (для массивов размером до 1024 слов)

a[n]

FOR j=0 TO n-2 STEP 1

f=0

FOR i=0 TO n-1-j STEP 1

IF a[i] > a[i+1] THEN

SWAP A[i], A[i+1]

f=1

IF f=0 THEN EXIT FOR

2. Умножение матриц (для матриц с каждой размерностью до 1024 слов)

a[n][m]

b[m][p]

q[n][p] = 0

FOR i = 0 TO n-1 STEP 1

FOR j = 0 TO p-1 STEP 1

FOR k = 0 TO m-1 STEP 1

q[i][j] = q[i][j] + a[i][k]\*b[k][j]

3. Медианный фильтр для двумерного массива (программу разработать самостоятельно)