Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра «Компьютерные системы и программные технологии»

**КУРСОВОЙ ПРОЕКТ**

по курсу «Автоматизация проектирования дискретных устройств»

Выполнил студент гр. 3530901/70203 Иванов И.Д.

(подпись)

Руководитель Антонов А.A. (подпись)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Санкт-Петербург

2020

Санкт-Петербургский государственный политехнический университет

**ЗАДАНИЕ**

**НА ВЫПолнение курсового проекта**

студенту группы 3530901/70203 Иванову Илье Дмитриевичу

*(номер группы) (фамилия, имя, отчество)*

***1. Срок сдачи законченного проекта*** 22.05.2020

***2. Исходные данные к проекту***: Задание для курсового проекта labsv\_7\_8\_9\_10\_11.

***3. Содержание пояснительной записки*** (перечень подлежащих разработке вопросов): labsv\_7-11.

***Дата получения задания***: «23» апреля 2020 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ Антонов А.А.

*(подпись)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ Иванов И.Д.

*(подпись студента)* \_\_\_\_\_\_\_\_\_\_\_\_

*(дата)*

Оглавление

[1. labsv\_7 4](#_Toc40904838)

[2. labsv\_8 9](#_Toc40904842)

[3. labsv\_9 14](#_Toc40904845)

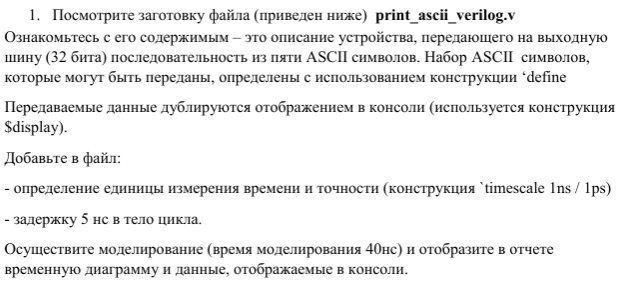
[4. labsv\_10 18](#_Toc40904848)

[5. labsv\_11 20](#_Toc40904851)

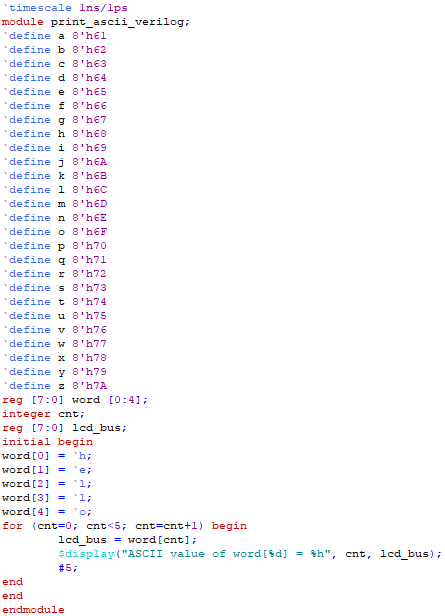
[6. Выводы 33](#_Toc40904858)

# labsv\_7

## 1.1.

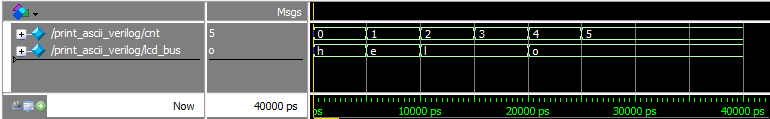


Содержимое файла print\_ascii\_verilog.v после внесения дополнений:

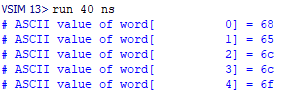


*Рис.1. Содержимое файла print\_ascii\_verilog.v*

Моделирование:

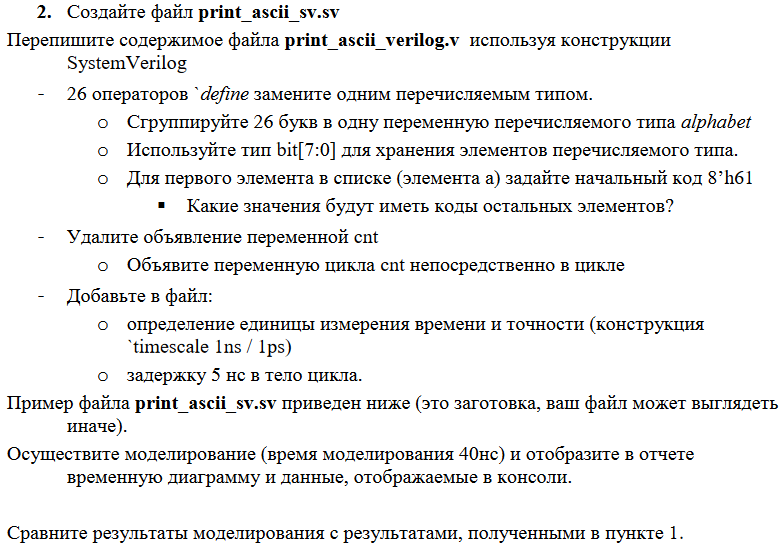


*Рис.2. Полученная временная диаграмма*

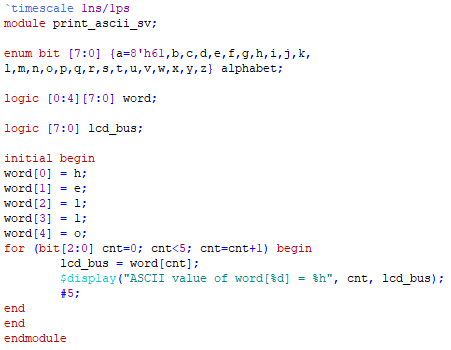
**

*Рис.3. Данные, отображаемые в консоли*

## 1.2.

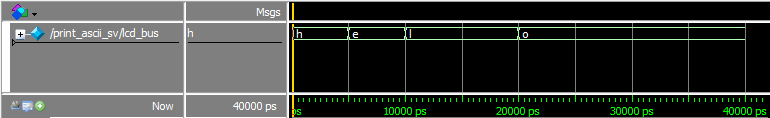


Содержимое файла print\_ascii\_sv.sv:

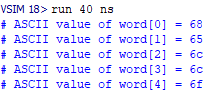


*Рис.4. Содержимое файла print\_ascii\_sv.sv*

Моделирование:



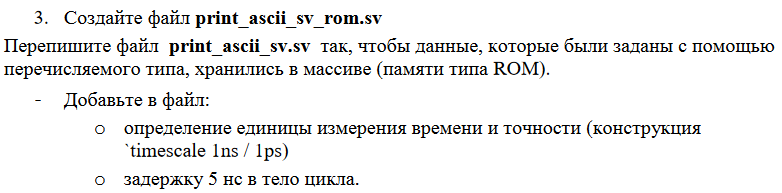
*Рис.5. Полученная временная диаграмма*

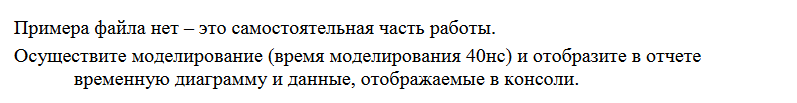
**

*Рис.6. Данные, отображаемые в консоли*

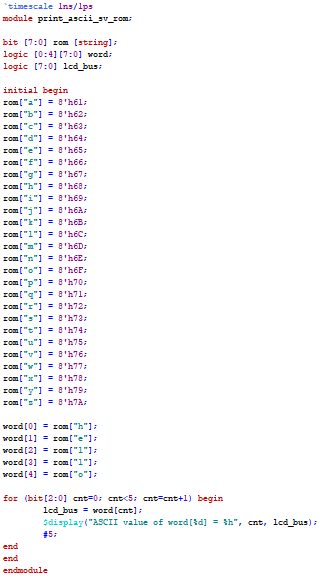
Полученные результаты совпадают с результатами в пункте 1.1.

## 1.3.





Содержимое файла print\_ascii\_sv\_rom.sv:

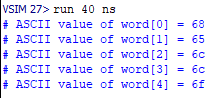


*Рис.7. Содержимое файла print\_ascii\_sv\_rom.sv*

Моделирование:



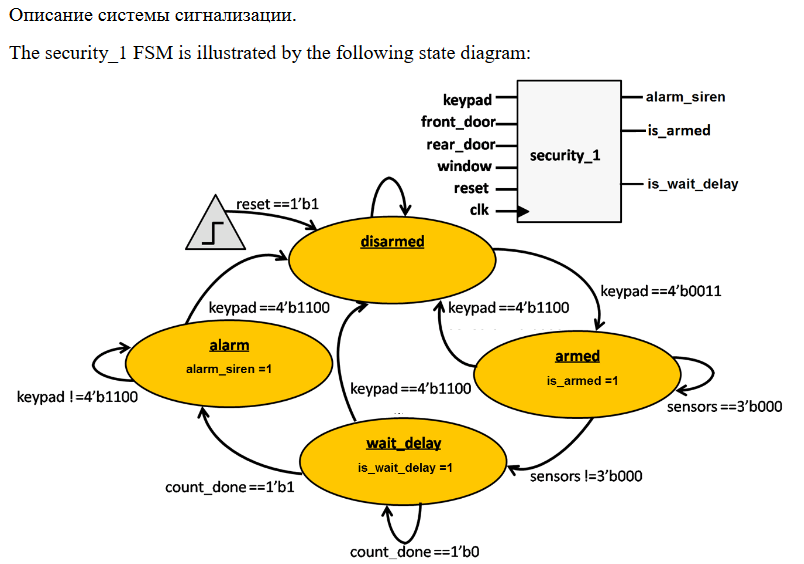
*Рис.8. Полученная временная диаграмма*

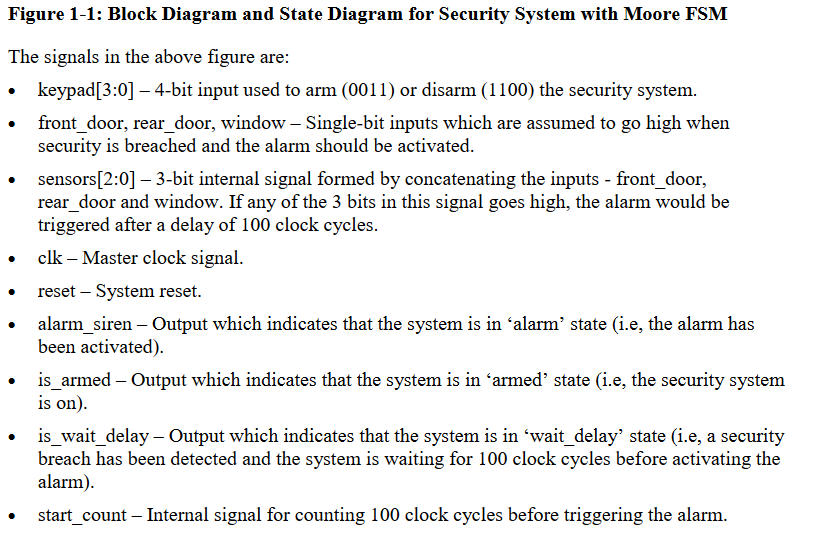
**

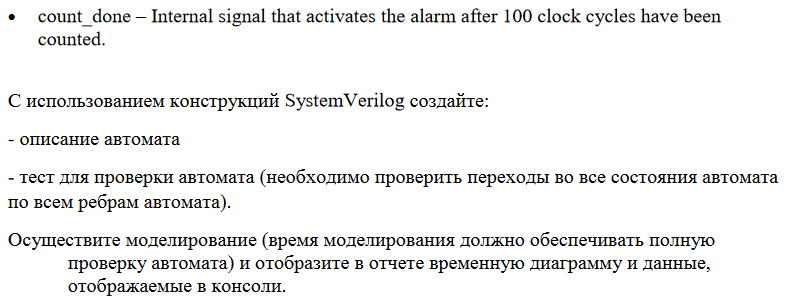
*Рис.9. Данные, отображаемые в консоли*

Полученные результаты совпадают с результатами в пунктах 1.1. и 1.2

# labsv\_8

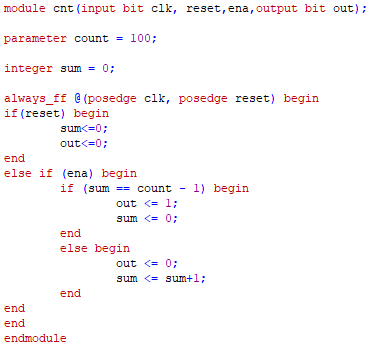
****

****

****

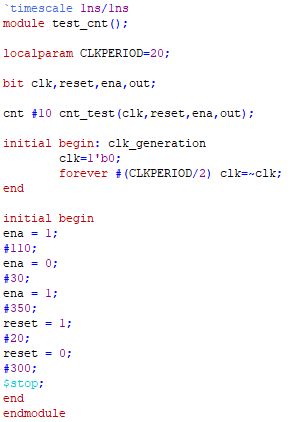
## Описание необходимых модулей и тестов на языке SystemVerilog

Описание счетчика, формирующего задержку в 100 тактов для перехода из состояния wait\_delay в состояние alarm:



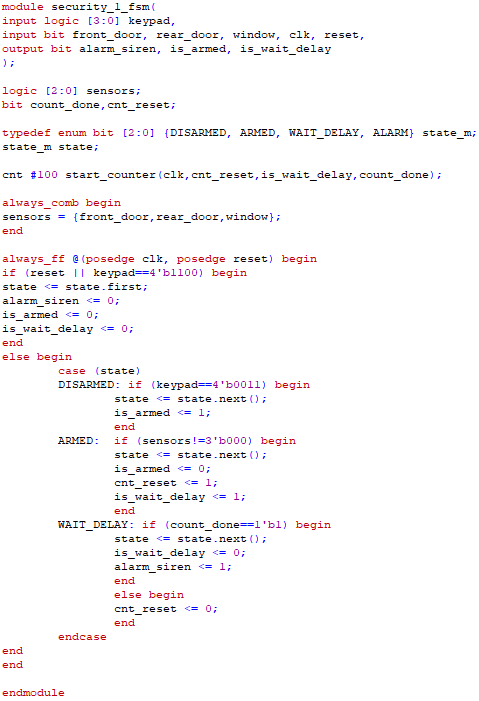
*Рис.10. Описание модуля cnt*

Тест первого класса для проверки работоспособности счетчика с параметром 10:

****

*Рис.11. Описание теста первого класса*

Описание автомата:



*Рис.12. Описание автомата*

Тест второго класса с вычислением результата для проверки работоспособности автомата:

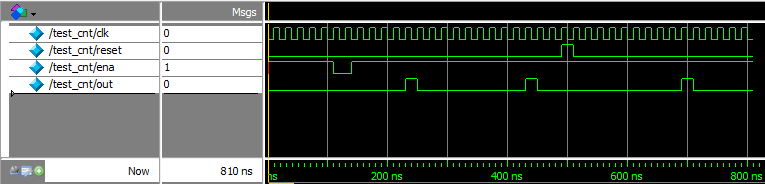


*Рис.13. Описание теста второго класса с вычислением результата*

В task fsm\_test значения, полученные от автомата, сравниваются с ожидаемыми значениями. В случае, если тестирование прошло успешно для всех входных данных, выводится сообщение об успешном прохождении тестирования.

## Тестирование

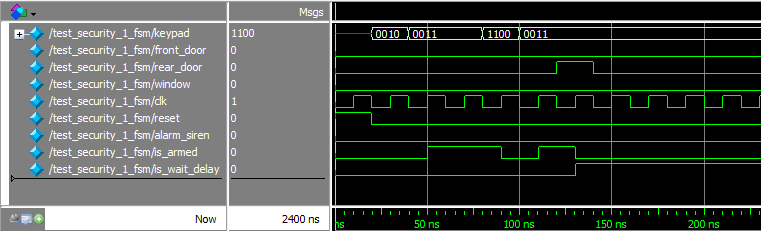
Результаты тестирования счётчика cnt:

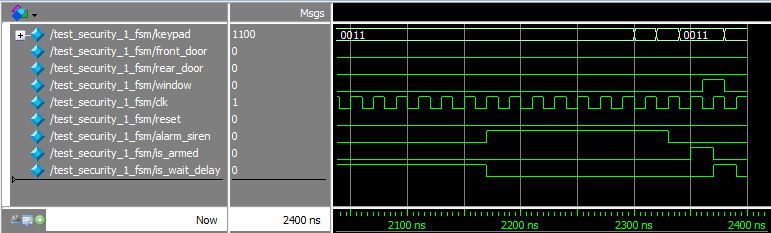


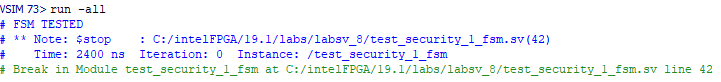
*Рис.14. Результаты тестирования*

Результаты тестирования полностью соответствуют ожидаемым, счётчик работает верно.

Результаты тестирования автомата:



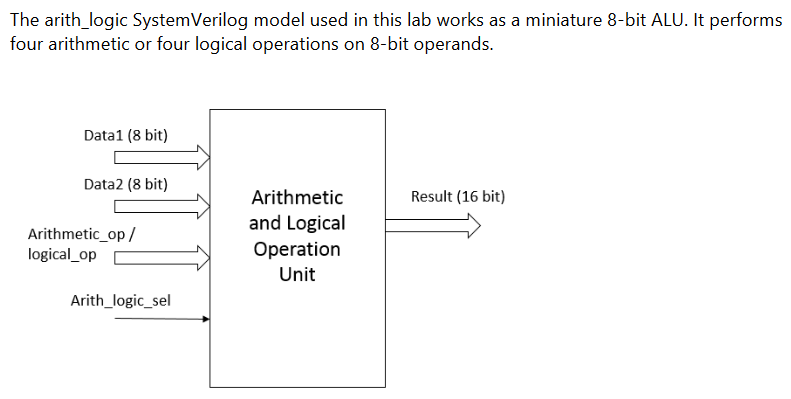
*Рис.15. Результаты тестирования*

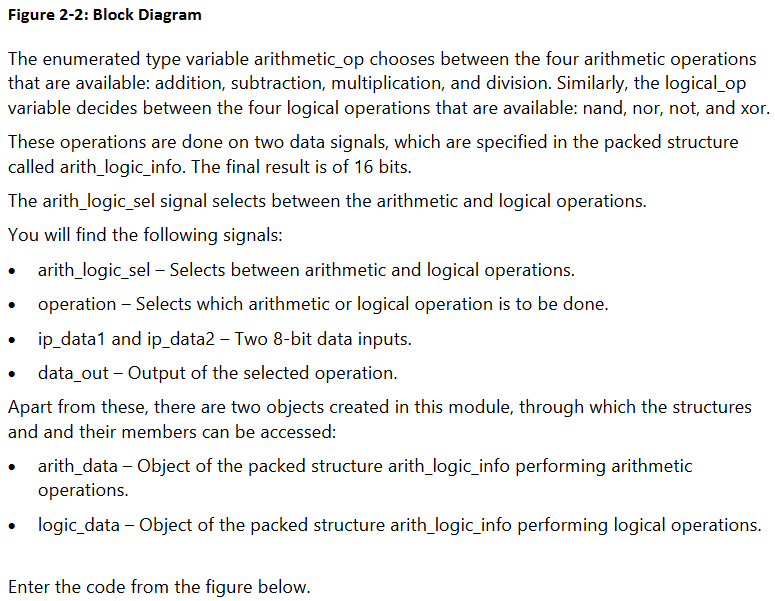


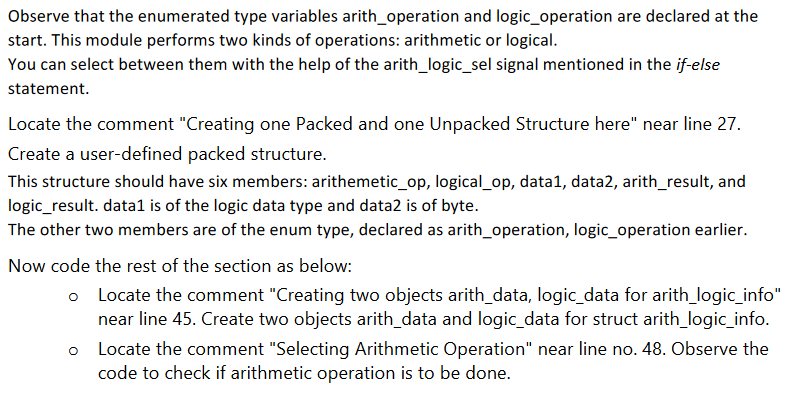
*Рис.16. Сообщение об успешном прохождении тестирования*

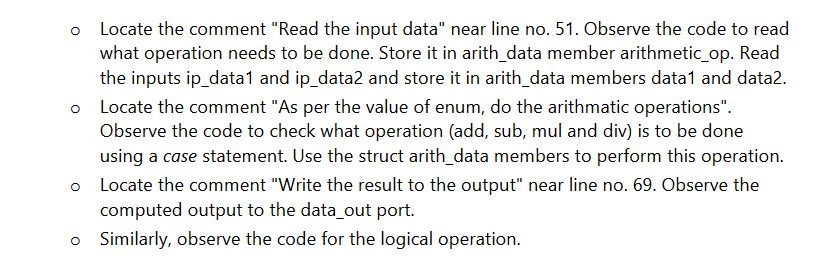
Все тесты успешно пройдены. Были проверены переходы во все состояния автомата по всем ребрам.

# labsv\_9





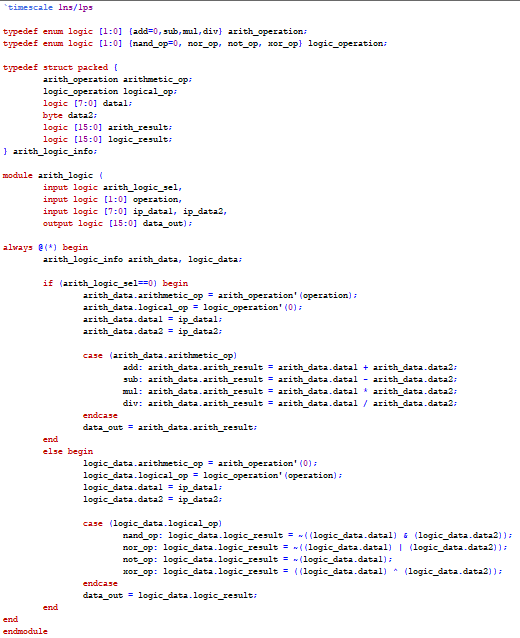






## Описание необходимых модулей и тестов на языке SystemVerilog

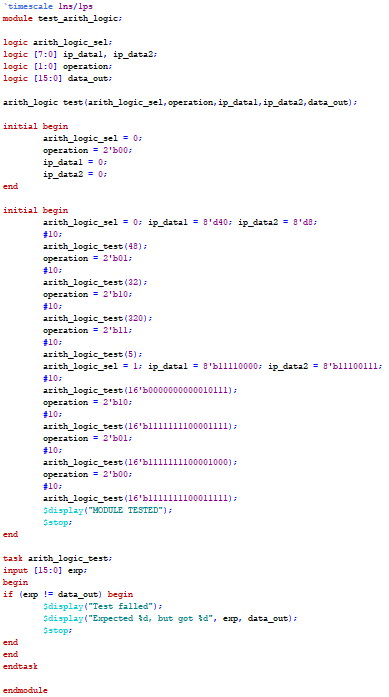
Содержимое файла arith\_logic.sv после внесения дополнений:



*Рис.17. Содержимое файла arith\_logic.sv*

Была создана требуемая структура arith\_logic\_info. В зависимости от значения сигнала arith\_logic\_sel выполняются либо арифметические, либо логические операции над входными данными.

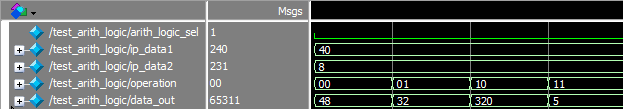
Тест второго класса с вычислением результата для проверки работоспособности модуля:

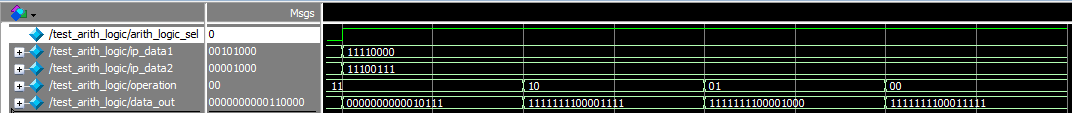


*Рис.18. Описание теста второго класса с вычислением результата*

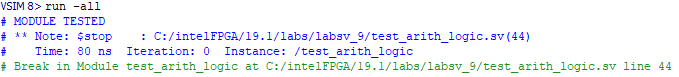
В task arith\_logic\_test значения, полученные из модуля, сравниваются с ожидаемыми значениями. В случае, если тестирование прошло успешно для всех входных данных, выводится сообщение об успешном прохождении тестирования.

## Моделирование





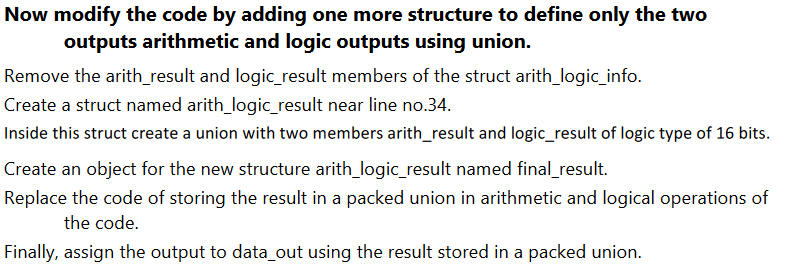
*Рис.19. Полученная временная диаграмма*

**

*Рис.20. Сообщение об успешном прохождении тестирования*

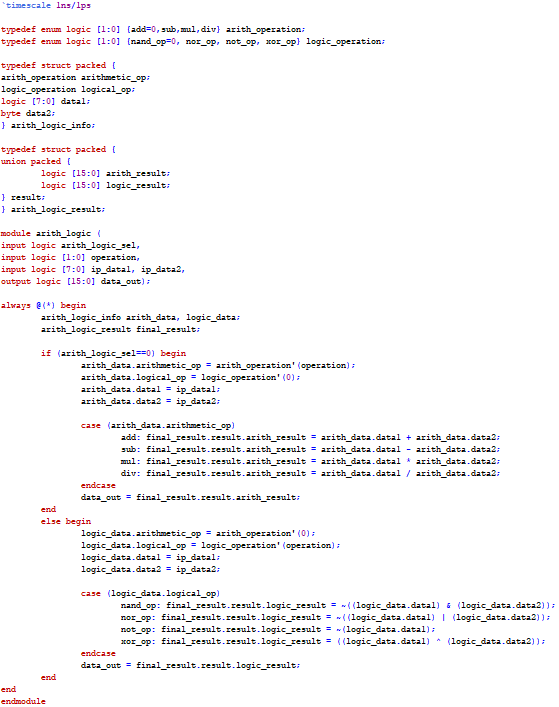
Все тесты успешно пройдены, модуль работает верно.

# labsv\_10



## Описание необходимых модулей и тестов на языке SystemVerilog

Содержимое файла arith\_logic.sv после внесения изменений:

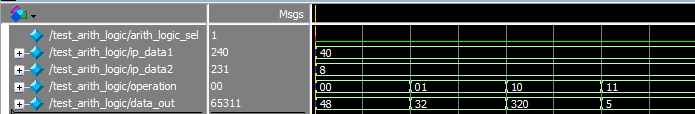


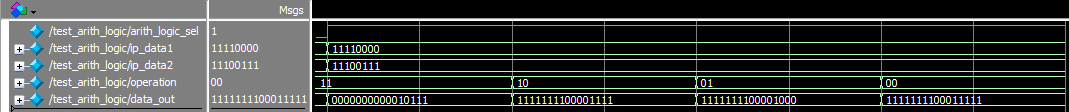
*Рис.21. Содержимое файла arith\_logic.sv*

Была добавлена ещё одна структура для определения двух выходных сигналов (арифметического и логического).

Модуль test\_arith\_logic оставим без изменений и снова используем для тестирования.

## Моделирование





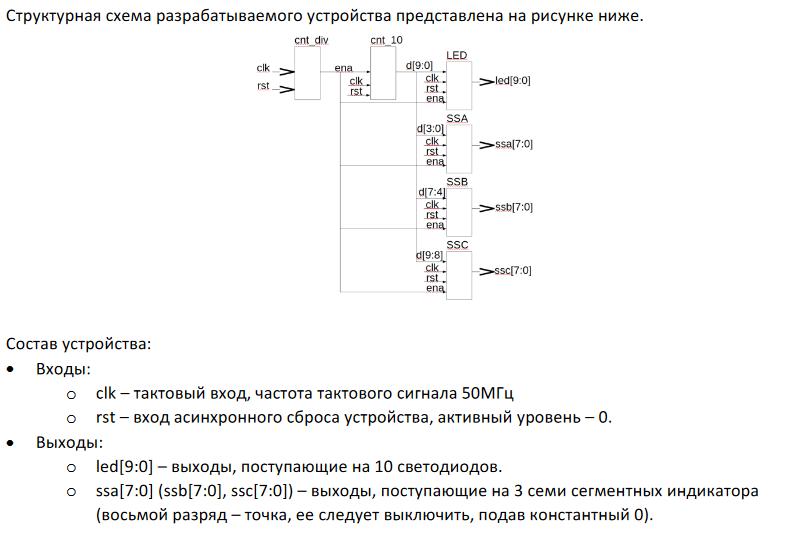
*Рис.22. Полученная временная диаграмма*

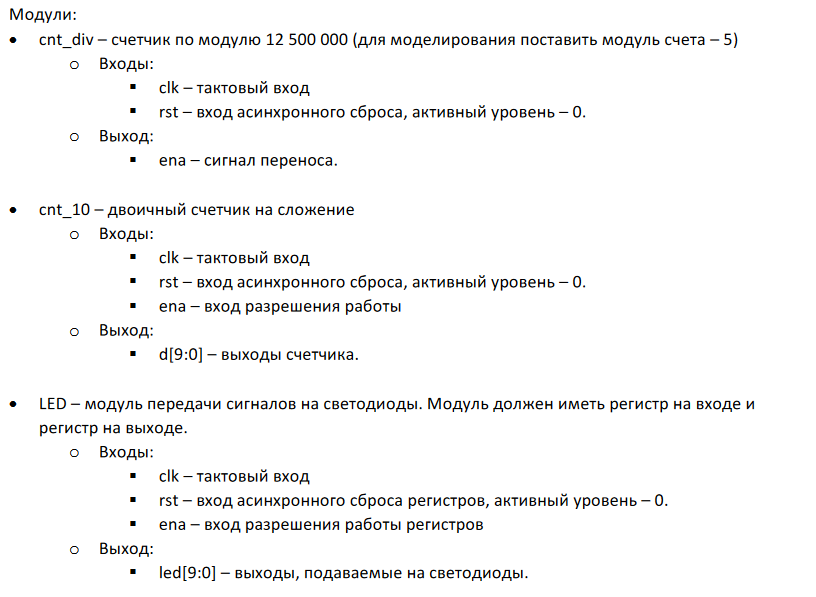
**

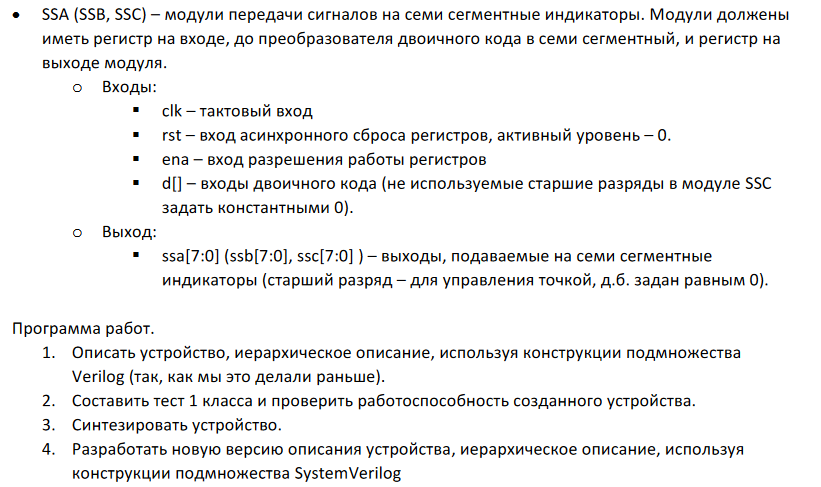
*Рис.23. Сообщение об успешном прохождении тестирования*

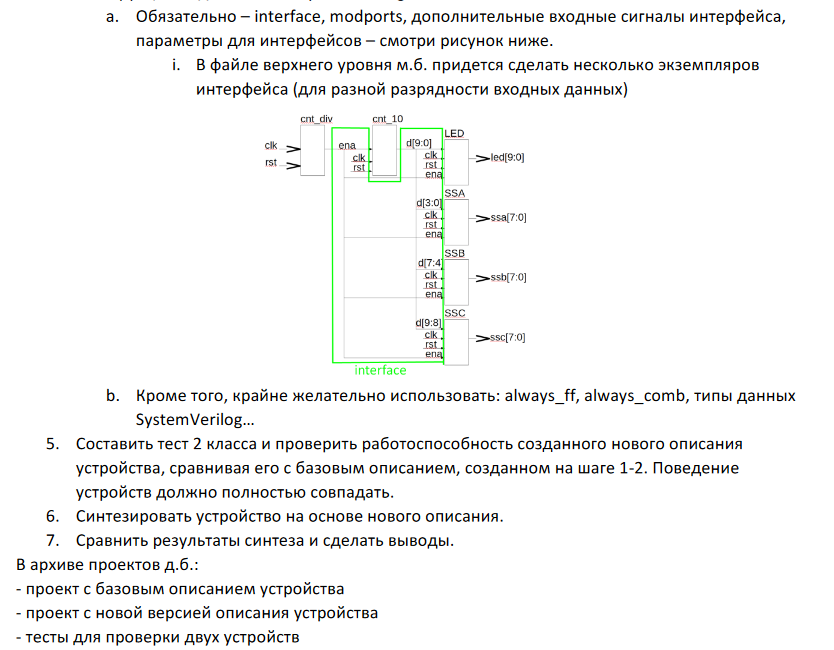
Все тесты успешно пройдены, модуль работает верно.

# labsv\_11



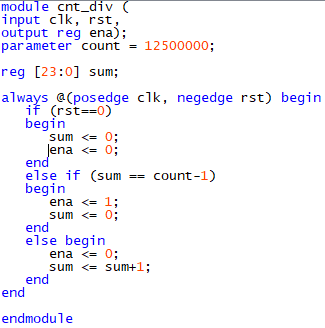






## Описание необходимых модулей и тестов на языке Verilog

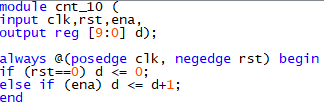
Описание модуля cnt\_div – счётчика по модулю 12 500 000:



*Рис.24. Описание модуля cnt\_div*

Счетчик-делитель cnt\_div формирует сигнал ena по модулю count.

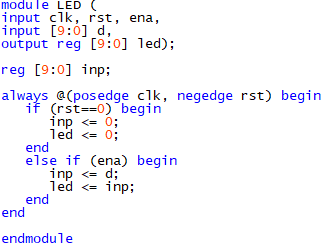
Описание модуля cnt\_10 – двоичного счётчика на сложение:



*Рис.25. Описание модуля cnt\_10*

Счетчик cnt\_10 формирует выходной 10-разрядный сигнал, увеличивая своё значение на единицу при ena = 1.

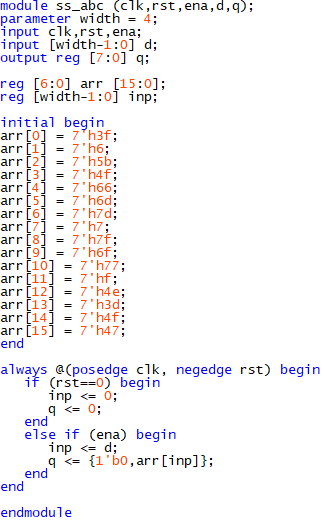
Описание модуля LED – модуля передачи сигналов на светодиоды:



*Рис.26. Описание модуля LED*

Модуль LED передает сигнал от счетчика cnt\_10 на светодиоды. На входе и выходе модуля установлены регистры.

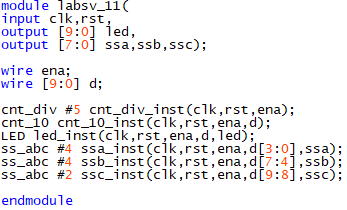
Параметризированное описание модулей ssa, ssb, ssc – модулей передачи сигналов на семисегментные индикаторы:



*Рис.27. Описание модуля ss\_abc*

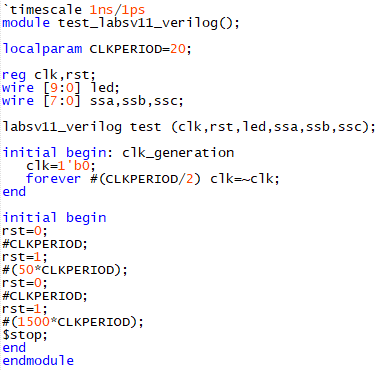
Модули имеют регистр на входе, до преобразователя двоичного кода в семисегментный, а также регистр на выходе.

Описание модуля верхнего уровня:



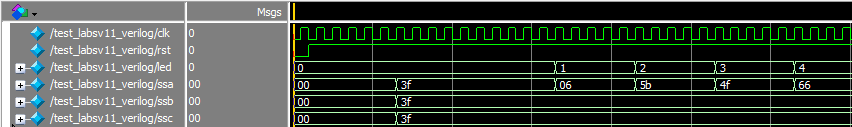
*Рис.28. Описание модуля верхнего уровня*

Тест первого класса:



*Рис.29. Описание теста первого класса*

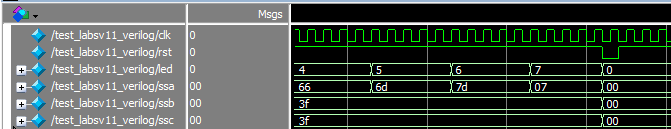
## Тестирование

**

*Рис.30. Полученная временная диаграмма*

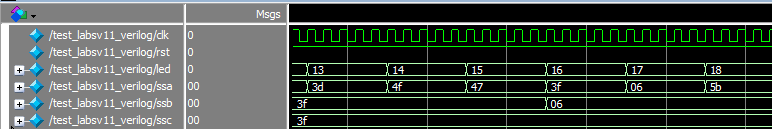
После 5-ти тактов clk происходит запись нулей (0x3f для семисегментного кода) в led, ssa, ssb и ssc, ещё через 5 тактов на вход подаётся единица, а ещё через 5 тактов на вход подаётся 2, а на выходах led и ssa оказывается предыдущее значение - 1 (0x06 для семисегментного кода), после чего подаётся 3, а на выходах оказывается 2 и т.д.

Сигнал асинхронного сброса:



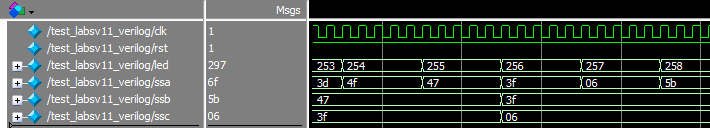
*Рис.31. Демонстрация работы сигнала асинхронного сброса*

Когда ssa переполняется, к выходу ssb добавляется единица, а сам ssa обнуляется:



*Рис.32. Демонстрация переполнения ssa*

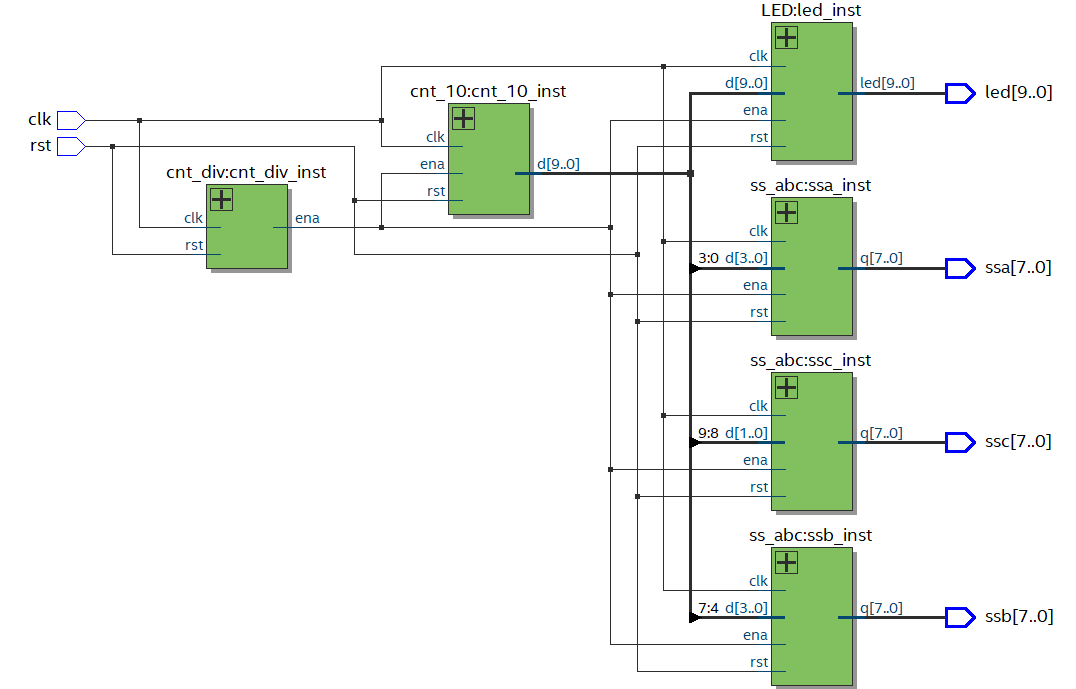
Когда ssb переполняется, к выходу ssc добавляется единица, а сам ssb обнуляется:



*Рис.33. Демонстрация переполнения ssb*

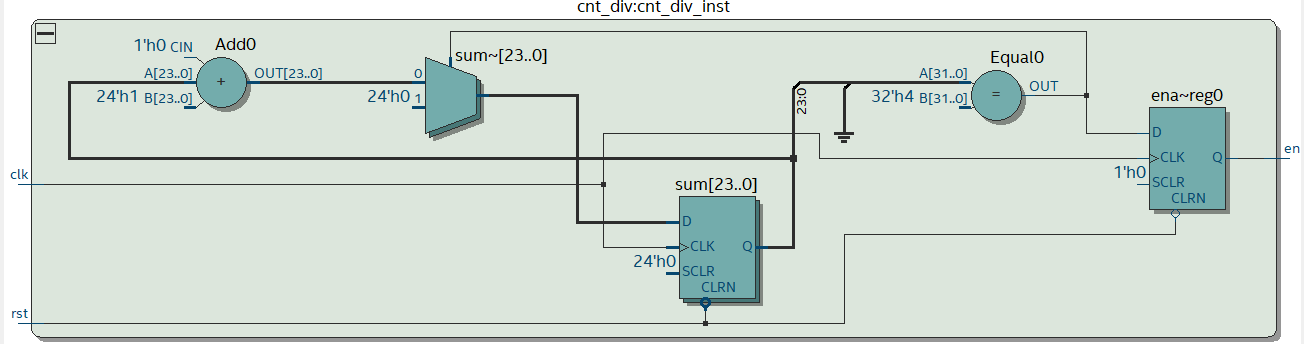
Результаты тестирования полностью соответствуют ожидаемым.

## Синтез устройства

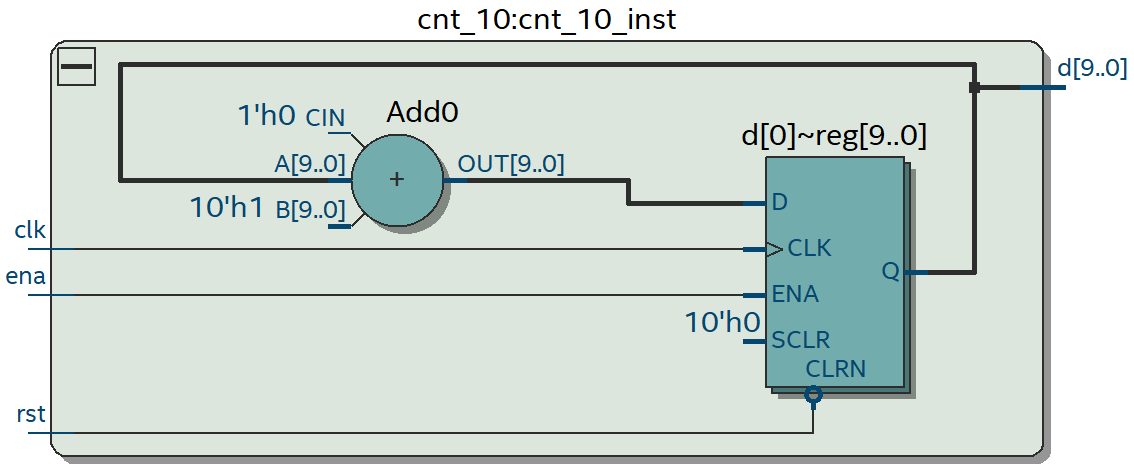


*Рис.34. Схема устройства в RTL-Viewer*

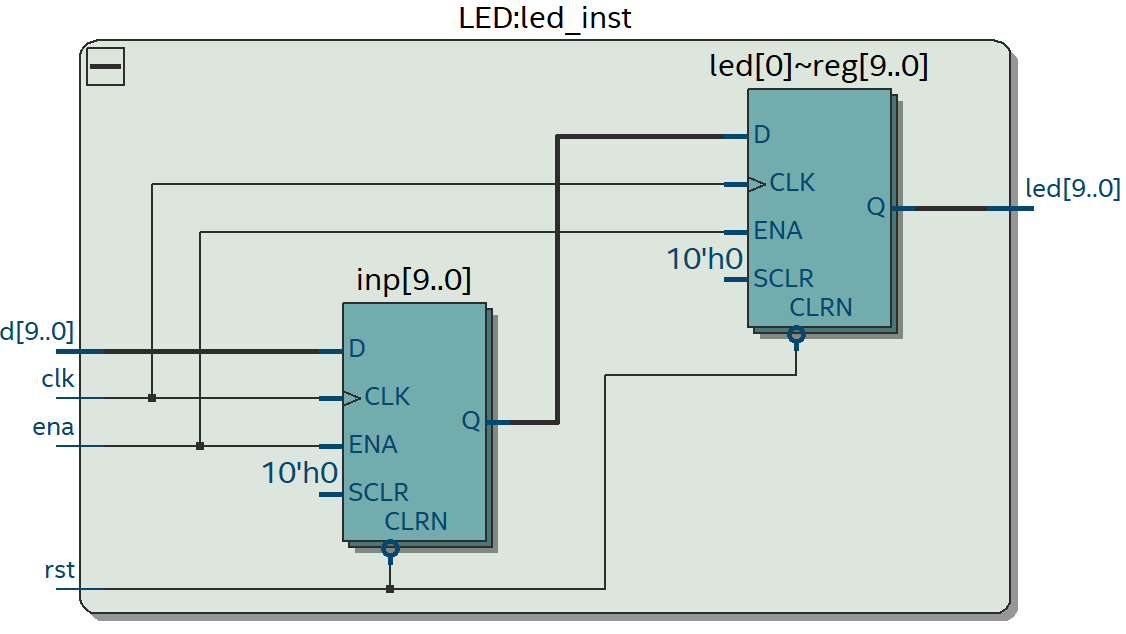
Полученная схема совпадает с заданной структурной схемой.



*Рис.35. cnt\_div в RTL-Viewer*

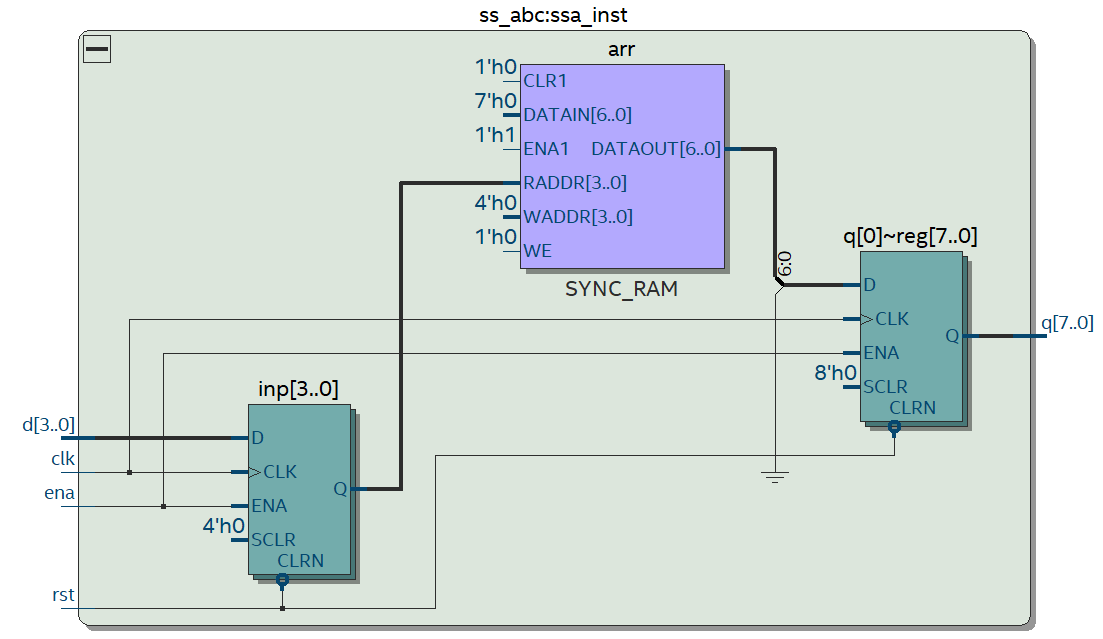
**

*Рис.36. cnt\_10 в RTL-Viewer*

**

*Рис.37. LED в RTL-Viewer*

Модуль LED имеет регистры на входе и выходе.



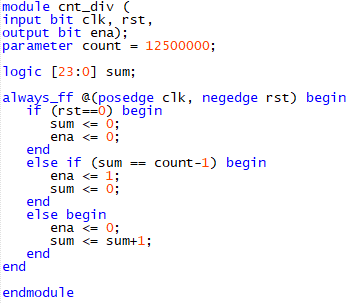
*Рис.38. ss\_abc(ssa) в RTL-Viewer*

Модули ssa, ssb и ssc имеют регистр на входе, до преобразователя двоичного кода в семисегментный, а также регистр на выходе.

## Описание необходимых модулей и тестов на языке SystemVerilog

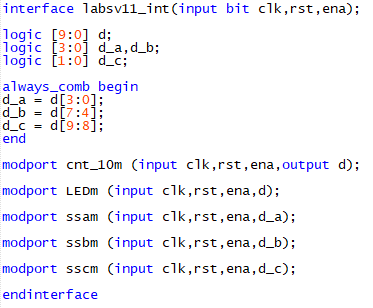
Был создан интерфейс, имеющий 5 modport-ов, по одному на каждый модуль (кроме cnt\_div). Кроме того, ssa, ssb и ssc имеют разную разрядность, поэтому для подключения к интерфейсу модуль ss\_abc был разбит на три отдельных ss\_a, ss\_b и ss\_c.

Описание модуля cnt\_div:



*Рис.39. Описание модуля cnt\_div*

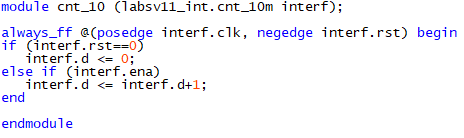
Описание интерфейса:



*Рис.40. Описание интерфейса*

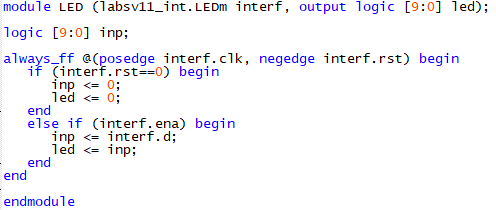
Сигнал, формирующийся на выходе счетчика cnt\_10, был разделен на 3 входных сигнала соответствующих разрядностей для модулей ss\_a, ss\_b и ss\_c.

Описание модуля cnt\_10:



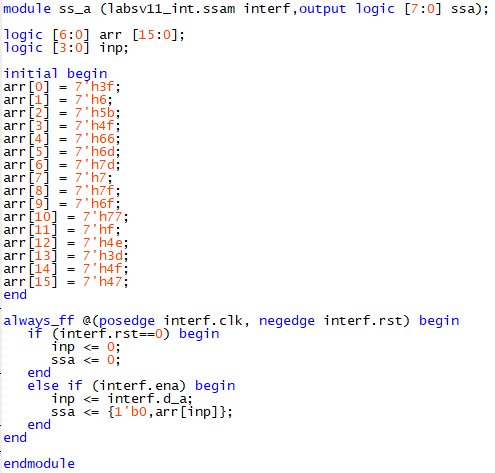
*Рис.41. Описание модуля cnt\_10*

Описание модуля LED:



*Рис.42. Описание модуля LED*

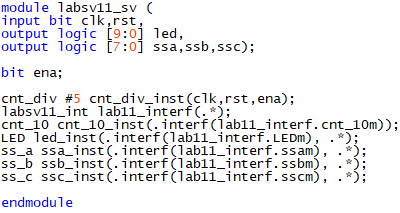
Описание модуля ss\_a:



*Рис.43. Описание модуля ss\_a*

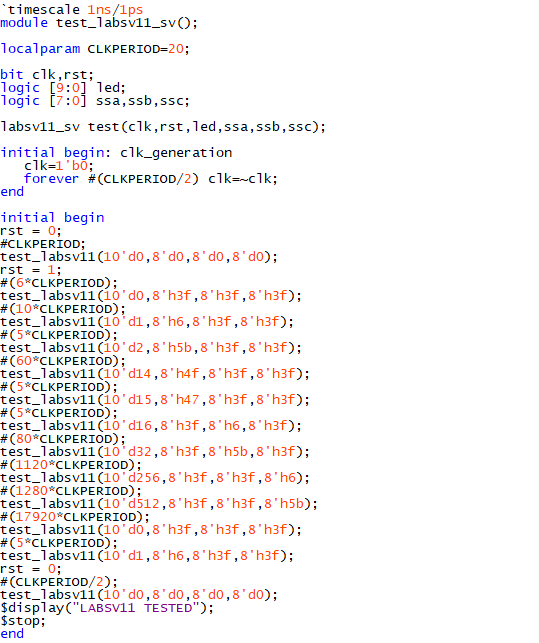
Описание модулей ss\_b и ss\_c аналогично (с заменой modport-а на ssbm, выхода на ssb и входных данных на d\_b для ss\_b и заменой modport-a, выхода, входного сигнала и разрядности переменной inp для ss\_c).

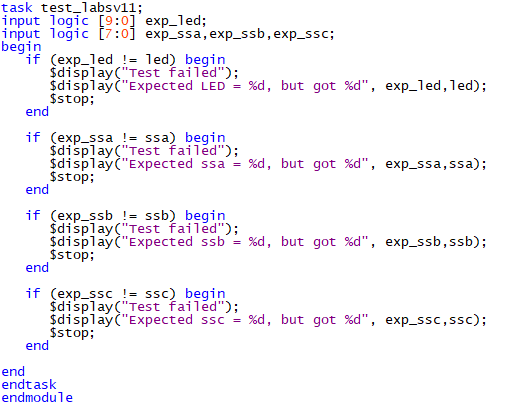
Описание модуля верхнего уровня:



*Рис.44. Описание модуля верхнего уровня*

Тест второго класса с вычислением результата:





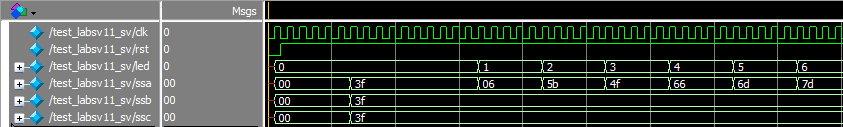
*Рис.45. Описание теста второго класса с вычислением результата*

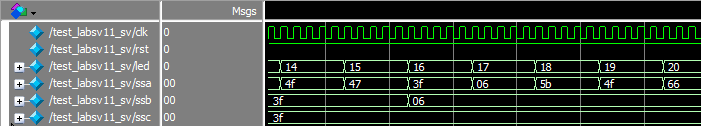
В task test\_labsv11 получаемые значения сравниваются с ожидаемыми. В случае, если тестирование прошло успешно для всех входных данных, выводится сообщение об успешном прохождении тестирования.

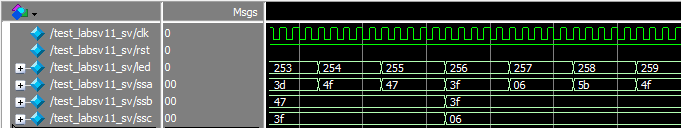
В тесте проверяется правильность результатов на выходах led, ssa, ssb и ssc при счёте и различных переполнениях, а также работа сигнала асинхронного сброса.

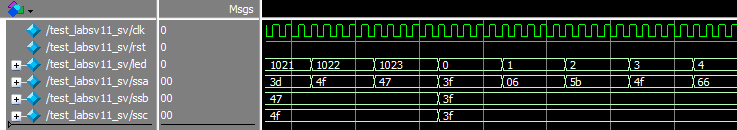
## Тестирование

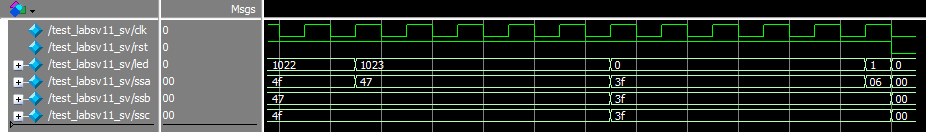
Результаты тестирования:



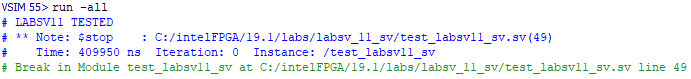








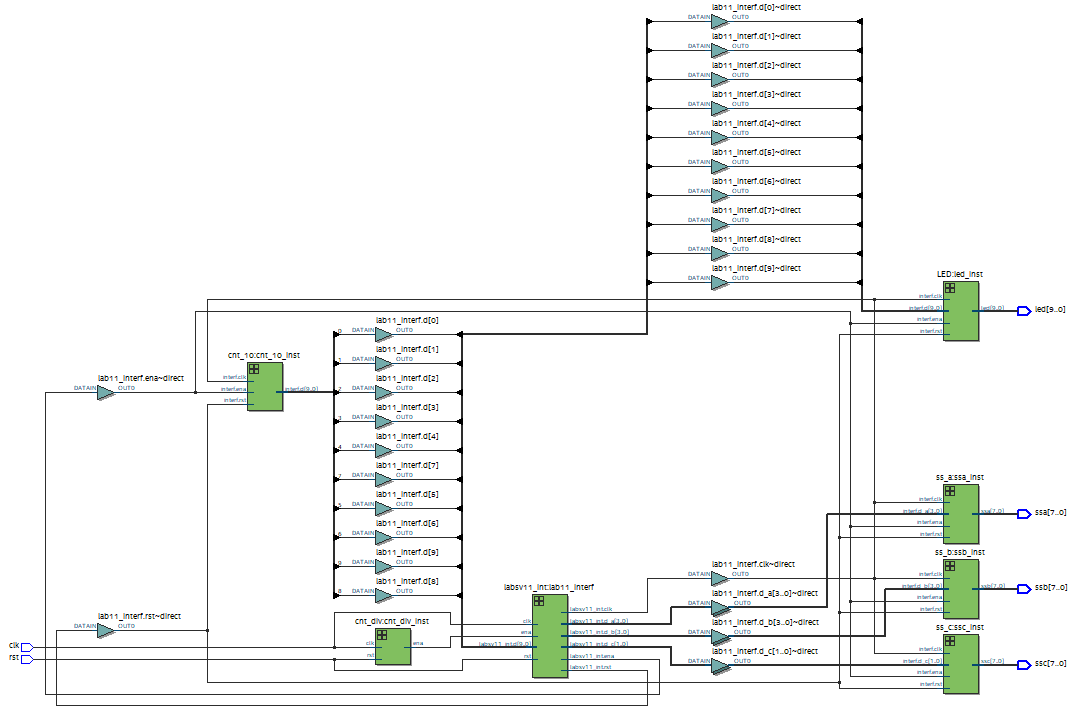
*Рис.46. Полученная временная диаграмма*



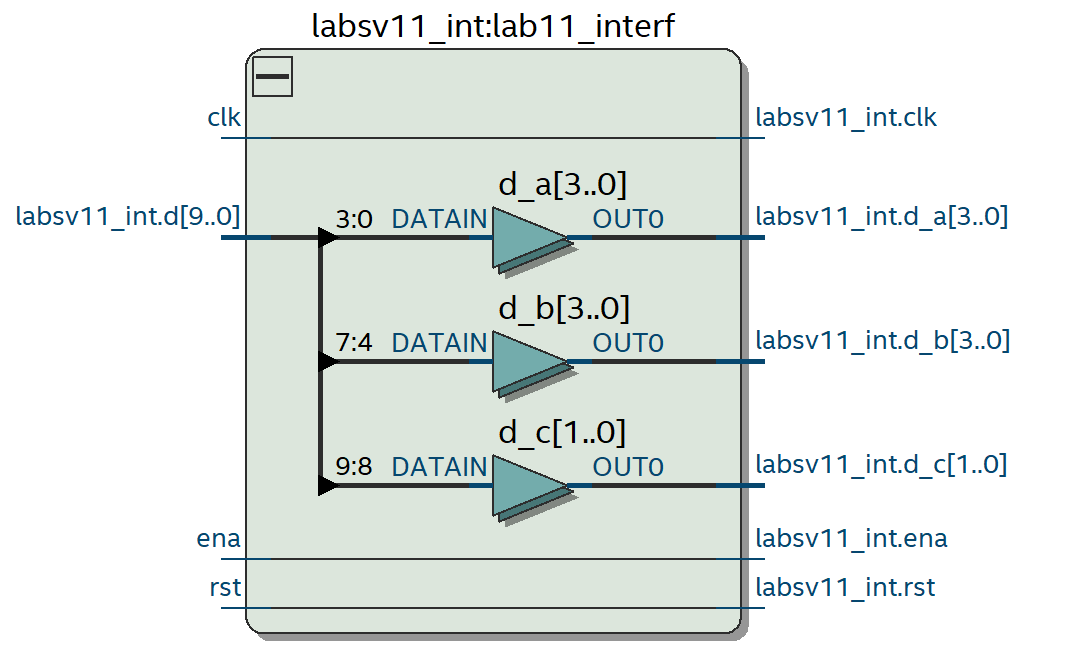
*Рис.47. Сообщение об успешном прохождении тестирования*

Все тесты успешно пройдены. Поведение устройства совпадает с результатом, полученным в пункте 5.2.

## Синтез устройства



*Рис.48. Схема устройства в RTL-Viewer*

**

*Рис.49. Схема интерфейса в RTL-Viewer*

Логическая реализация модулей cnt\_div, cnt\_10, LED, ss\_a, ss\_b и ss\_c не изменилась.

Полученная схема соответствует схеме, указанной в задании. Разница между ней и схемой, полученной в пункте 5.3, заключается в появлении на схеме интерфейса, добавленного для того, чтобы сгруппировать порты и избежать их повторного определения.

# Выводы

В ходе выполнения данного курсового проекта были выполнены все 5 поставленных задач, а также закреплены знания конструкций языков Verilog и SystemVerilog. В частности, был получен опыт работы с struct и union, описан интерфейс с использованием modport-ов и написаны тесты для проверки заданных устройств.