САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

по заключительному заданию

Дисциплина

«Технологии проектирования аппаратных средств компьютерных систем»

выполнил: Курякин Д. А.

группа: 3540901/12001

преподаватель: Антонов А. П.

Санкт-Петербург

2022

Оглавление

[1 Заключительное задание 4](#_Toc106806188)

[1.1 Задание 4](#_Toc106806189)

[1.2 Часть 1. Задача 1 8](#_Toc106806190)

[1.2.1 Описание АЛУ на языке SystemVerilog 8](#_Toc106806191)

[1.2.2 Моделирование 9](#_Toc106806192)

[1.2.3 Интеграция библиотечного компонента в PD 11](#_Toc106806193)

[1.3 Часть 1. Задача 2 13](#_Toc106806194)

[1.3.1 Описание генератора даннх на языке SystemVerilog 13](#_Toc106806195)

[1.3.2 Моделирование 13](#_Toc106806196)

[1.3.3 Интеграция библиотечного компонента в PD 15](#_Toc106806197)

[1.4 Часть 1. Задача 3 17](#_Toc106806198)

[1.4.1 Описание конечного автомата на языке SystemVerilog 17](#_Toc106806199)

[1.4.2 Моделирование 18](#_Toc106806200)

[1.4.3 Интеграция библиотечного компонента в PD 19](#_Toc106806201)

[1.5 Часть 1. Задача 4. 21](#_Toc106806202)

[1.5.1 Создание системы в PD 21](#_Toc106806203)

[1.5.2 Описание на языке SystemVerilog 23](#_Toc106806204)

[1.5.3 Моделирование 24](#_Toc106806205)

[1.6 Часть 1. Задача 5. 25](#_Toc106806206)

[1.6.1 Описание на языке SystemVerilog 25](#_Toc106806207)

[1.6.2 Результат синтеза (RTL) 28](#_Toc106806208)

[1.6.3 Проверка на плате с помощью Signal TapII 28](#_Toc106806209)

[1.7 Часть 2. Задача 2. 29](#_Toc106806210)

[1.7.1 Создание системы в PD 29](#_Toc106806211)

[1.7.2 Программирование Nios 2 31](#_Toc106806212)

[1.7.3 Отладка в Signal Tabe Logic Analyzer 32](#_Toc106806213)

[1.7.4 Результат синтеза (RTL) 33](#_Toc106806214)

[1.7.5 Проверка на плате с помощью Signal TapII 34](#_Toc106806215)

[1.8 Выводы 35](#_Toc106806216)

# Заключительное задание

## Задание

**Часть 1**

**Задача 1**

1. С использованием типов данных и конструкций расширения System Verilog создайте описание параметризированного 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 параметризированное 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**

*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*.*

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

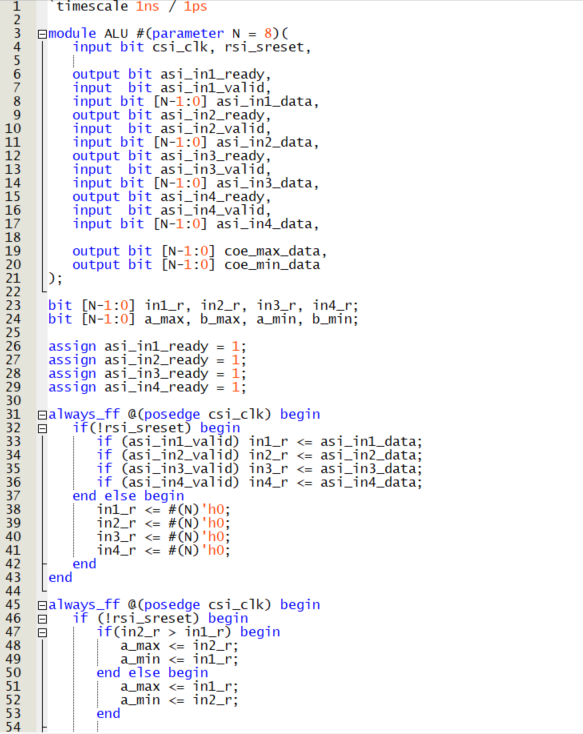
* Каналы входных данных
  + Каналы должны быть ориентированы на использование Avalon ST (Stream) интерфейса в Platform Designer (PD) с поддержкой сигналов Ready и Valid
  + На входах данных д.б использованы регистры.
    - по входному сигналу Valid =1 осуществляется запись данных во входные регистры 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

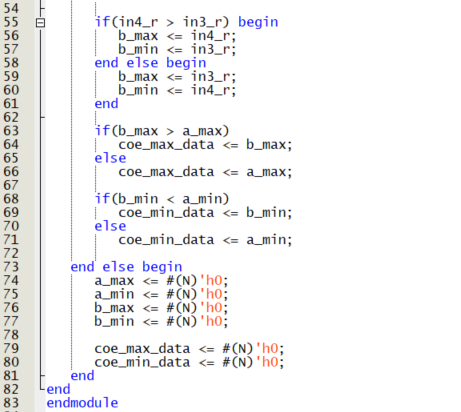
## Часть 1. Задача 1

### Описание АЛУ на языке SystemVerilog

Опишем АЛУ. Описание устройства на языке SystemVerilog приведено ниже на листинге 1-1. В строках 6-20 обявленны входные и выходные значения; в строках 31-43 задана присвоение в регистры; в строках 45-82 заданно нахождение максимального и минимального значения.

Листинг 1-1. ALU.sv

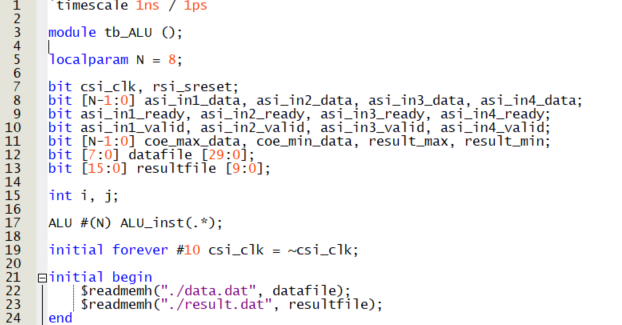


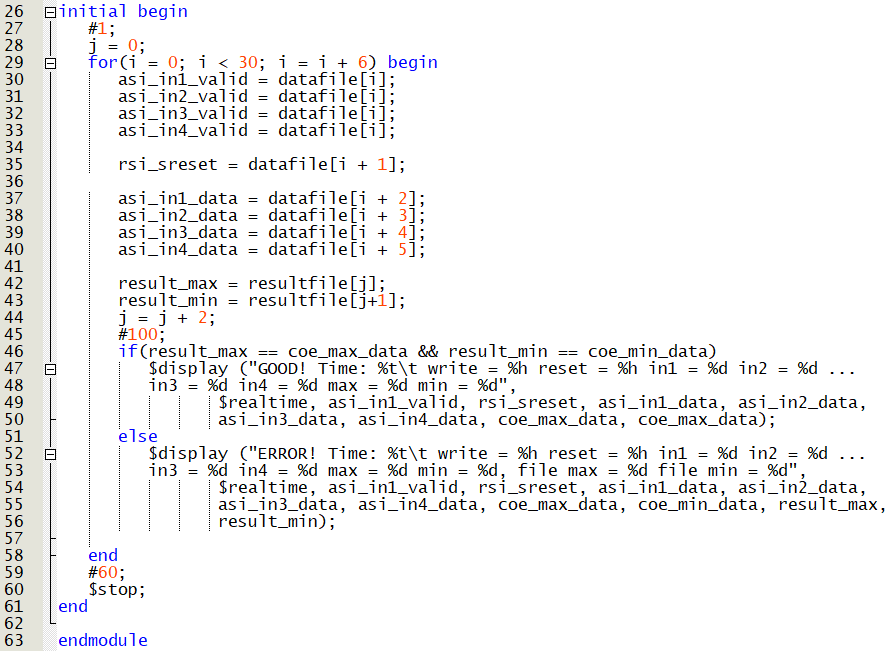


### Моделирование

Проверим АЛУ. Для проверки правильности работы использовался тест второго класса (tb\_ALU.sv). В строках 7-13 обявленны входные и выходные значения; в строке 17 задана инициализация моделируемого модуля АЛУ; в строках 21-24 заданно чтение из файлов, в строках 26-58 заданна проверка устройства. На листинге 1-2 приведено описание симуляции:

Листинг 1-2. tb\_alu.sv





Для создания симуляции в настройках проекта в разделе симуляции были добавлены файлы представленные на рис. 1-1.

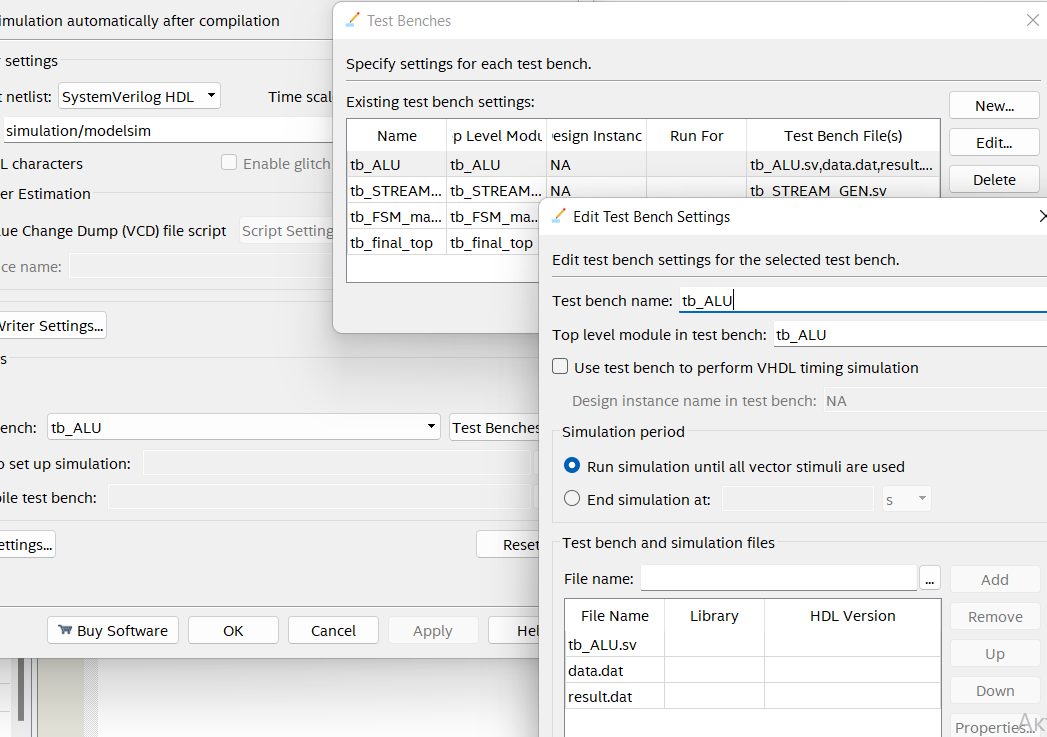


Рис. 1‑1. Настройка симуляции АЛУ.

Результаты моделирования представлены на рисунке 1-2.

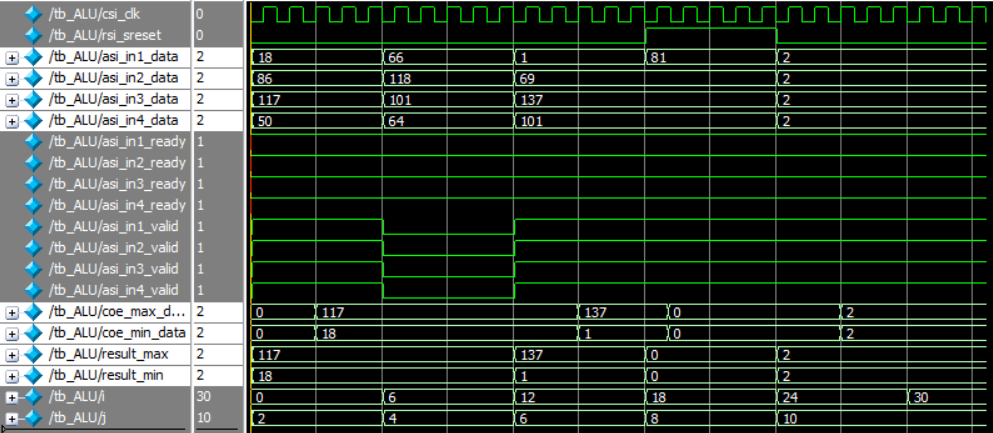


Рис. 1‑2 Результаты моделирования тестового файла tb\_ALU.sv (тип отображения - Decimal)

Для моделирования были инициализированы входные и выходные данные. На рис. 1-3 и 1-4 показаны файлы со входными и выходными данными.

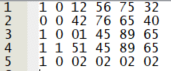


Рис. 1‑3 Файл входных данных data.dat



Рис. 1‑4 Файл выходных данных result.dat

При тестировании в консоль выводились результаты проверки АЛУ. Автоматическая проверка результатов моделирования приведена на рис 1-5.

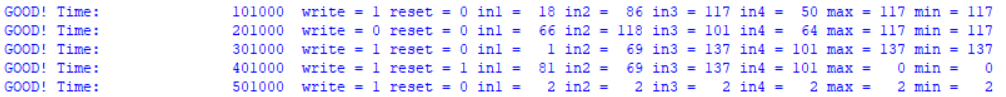


Рис. 1‑5 Автоматическая проверка

В результате симуляции доказанна работоспособность АЛУ.

### Интеграция библиотечного компонента в PD

Созданное устройство было интегрировано как библиотечный компонент в Platform Designer (библиотечная папка - kurs). На рис. 1-6 показана интеграция

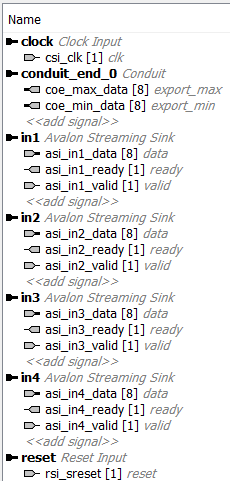


Рис. 1‑6 Настройки Signals & Interfaces в Platform Designer для АЛУ

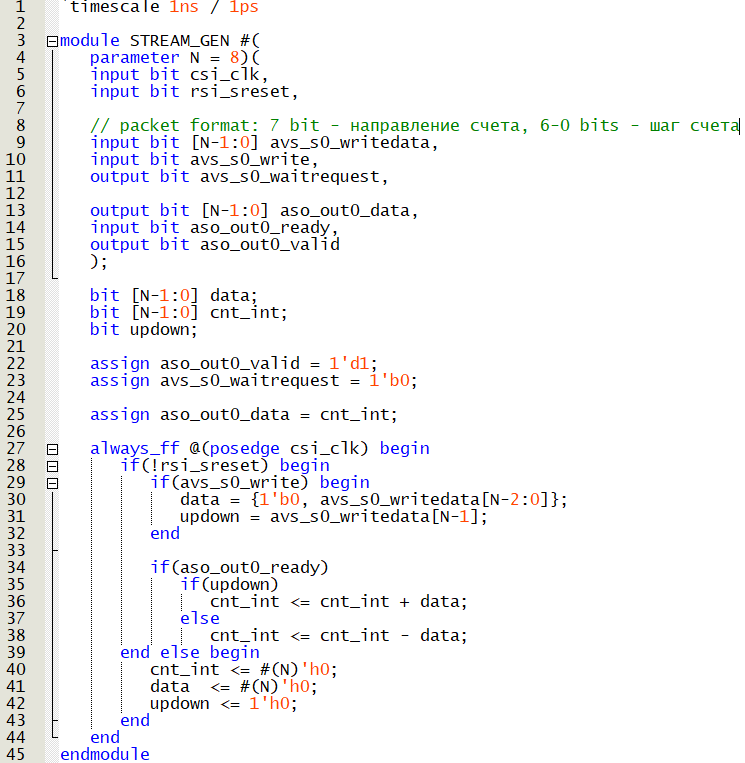
В результате было получено новое ip-ядро в Platform Designer.

## Часть 1. Задача 2

### Описание генератора данных на языке SystemVerilog

Опишем генератор данных. Описание устройства на языке SystemVerilog приведено ниже на листинге 1-3. В строках 5-16 обвялены входные и выходные значения; в строках 27-44 описано устройство генерации данных.

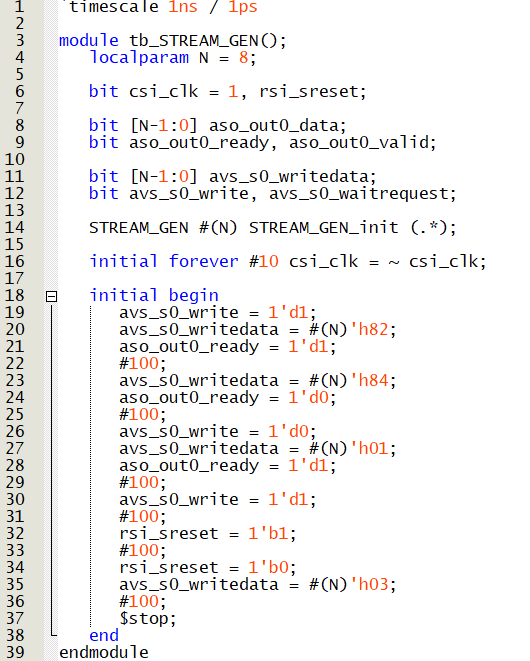
Листинг 1-3. STREAM\_GEN.sv



### Моделирование

Проверим генератор данных. Для проверки правильности работы использовался тест первого класса (tb\_ STREAM\_GEN.sv). В строках 6-12 обвялены входные и выходные значения; в строке 14 задана инициализация моделируемого модуля генератора данных; в строках 18-38 задано создание данных для проверки устройства. На листинге 1-4 приведено описание симуляции:

Листинг 1-4. tb\_STREAM\_GEN.sv



Для создания симуляции в настройках проекта в разделе симуляции были добавлены файлы представленные на рис. 1-7.

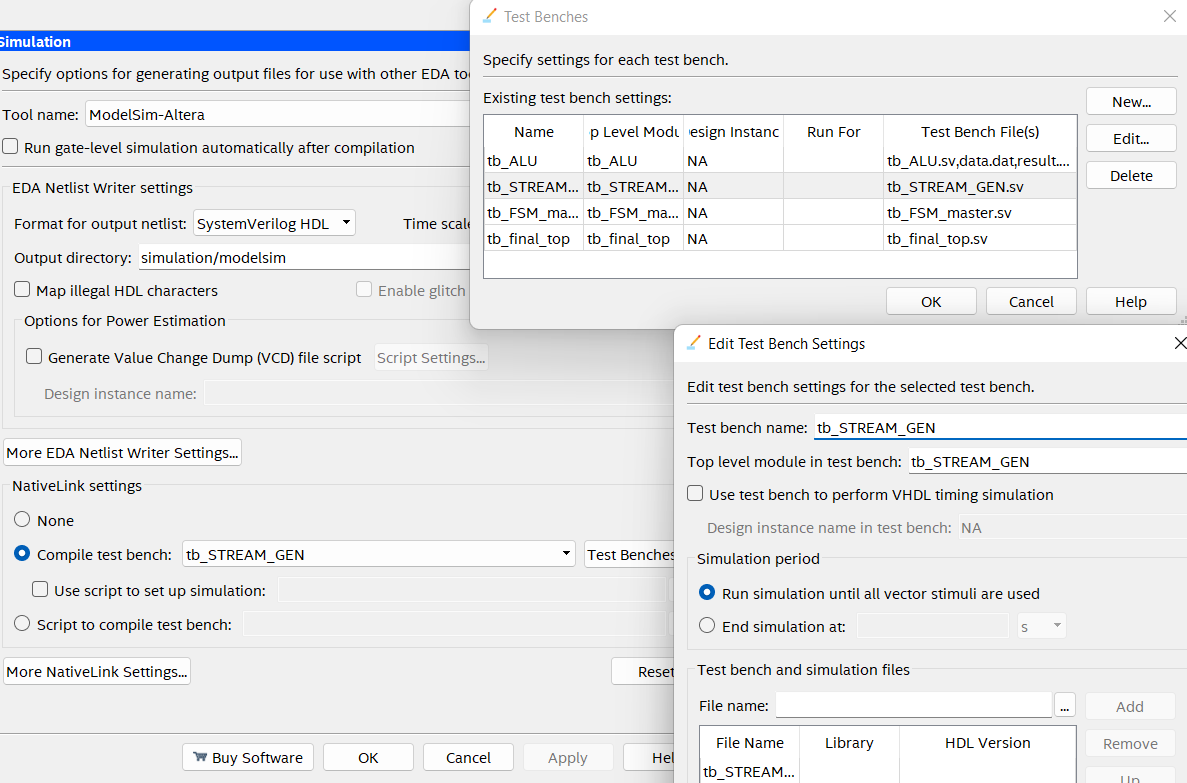


Рис. 1‑7. Настройка симуляции генератора данных.

Результаты моделирования представлены на рисунке 1-8 ниже:

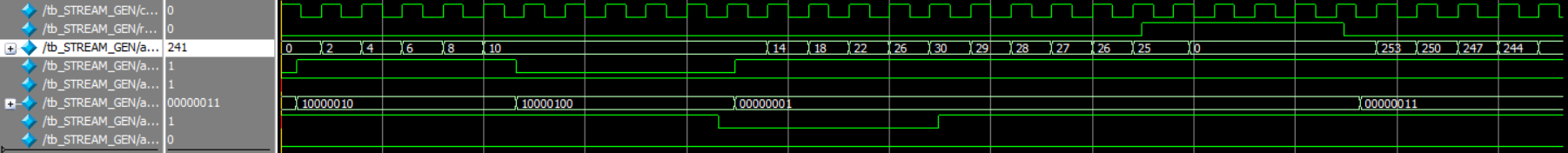


Рис. 1‑8. Результаты моделирования тестового файла tb\_ STREAM\_GEN.sv (тип отображения - Decimal)

В результате симуляции доказанна работоспособность генератора данных.

### Интеграция библиотечного компонента в PD

Созданное устройство было интегрировано как библиотечный компонент в Platform Designer (библиотечная папка - kurs):

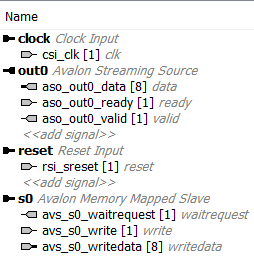


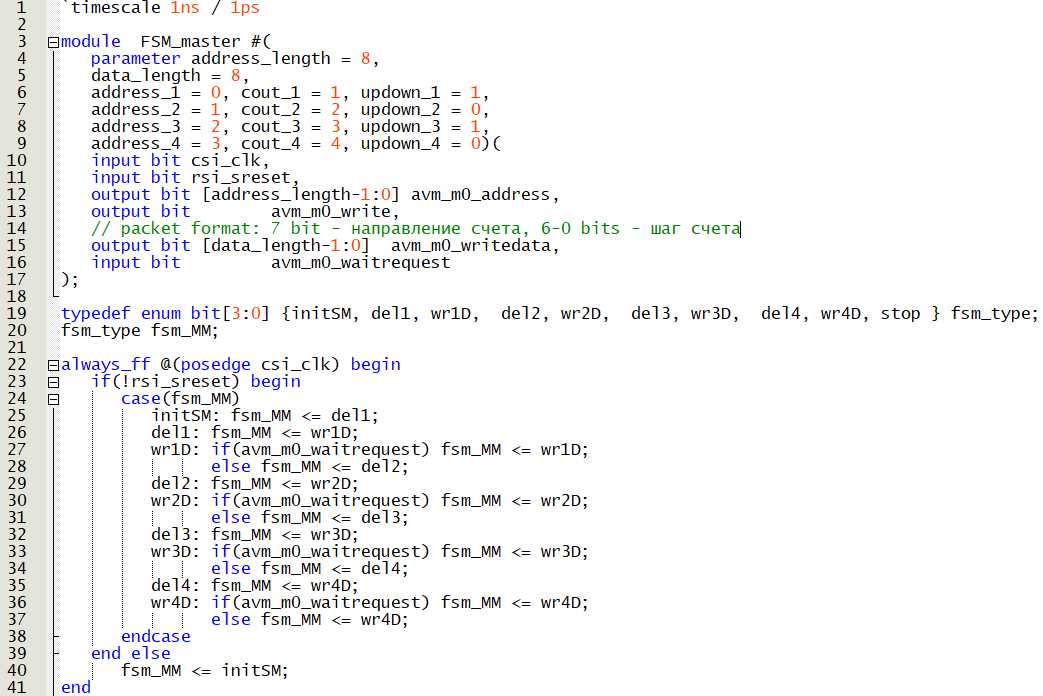
Рис. 1‑9. Настройки Signals & Interfaces в Platform Designer для STREAM\_GEN

## Часть 1. Задача 3

### Описание конечного автомата на языке SystemVerilog

Опишем конечный автомат. Описание устройства на языке SystemVerilog приведено ниже на листинге 1-5. В строках 4-9 обвялены параметры; в строках 10-16 обвялены входные и выходные значения; в строках 19-41 описан конечный автомат; а строках 43-76 описана генерация данных опираясь на состояние конечного автомата.

Листинг 1-5. FSM\_master.sv

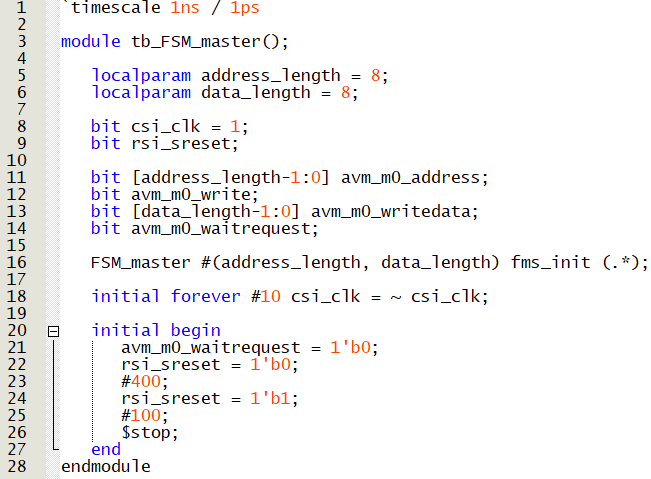




### Моделирование

Проверим конечный автомат. Для проверки правильности работы использовался тест первого класса (tb\_FSM\_master.sv). В строках 5-14 обвялены входные и выходные значения; в строке 16 задана инициализация моделируемого модуля генератора данных; в строках 18-27 задано создание данных для проверки устройства. На листинге 1-6 приведено описание симуляции:

Листинг 1-6. tb\_FSM\_master.sv



Для создания симуляции в настройках проекта в разделе симуляции были добавлены файлы представленные на рис. 1-10.

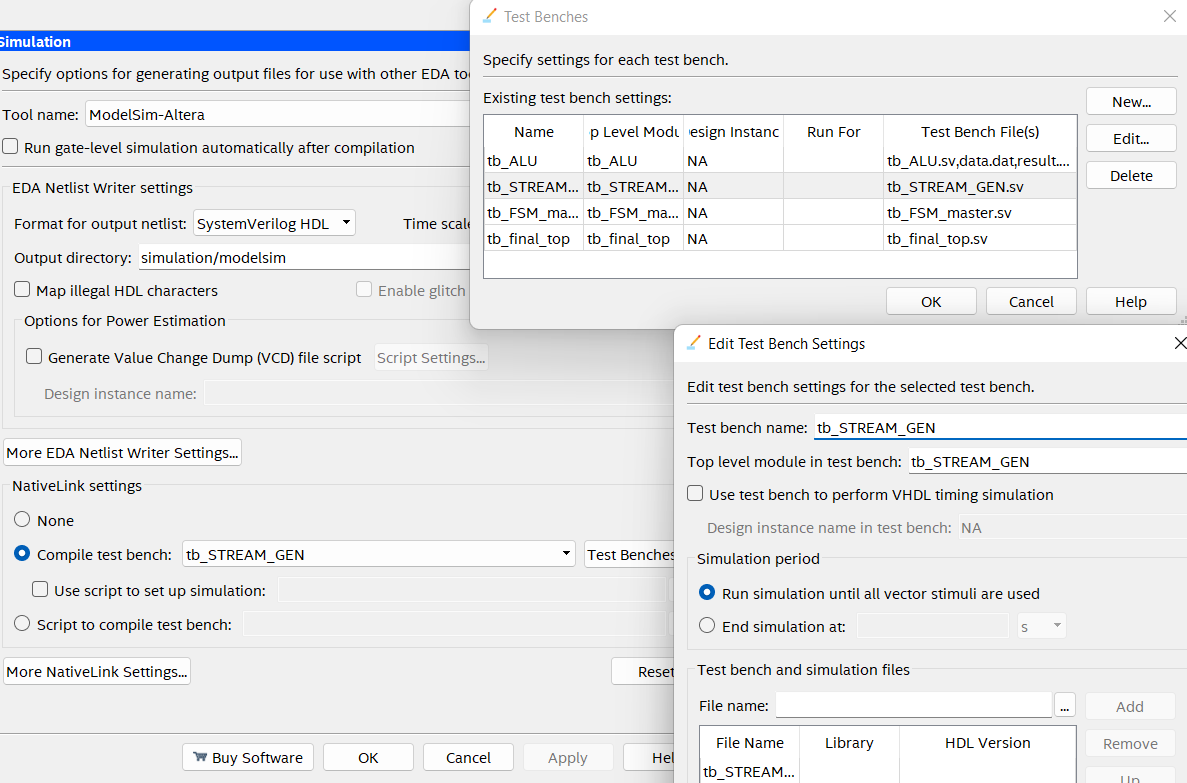


Рис. 1‑10. Настройка симуляции генератора данных.

Результаты моделирования представлены на рисунке 1-11 ниже:

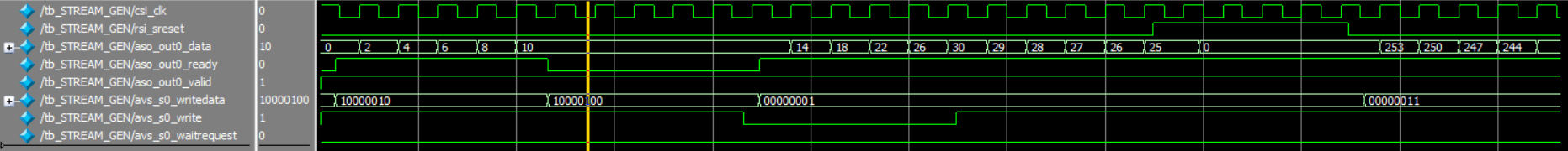


Рис. 1‑11. Результаты моделирования тестового файла tb\_ FSM\_master.sv (тип отображения - Decimal)

В результате симуляции доказанна работоспособность конечного автомата.

### Интеграция библиотечного компонента в PD

Созданное устройство было интегрировано как библиотечный компонент в Platform Designer (библиотечная папка - kurs). На рис 1-12 представлена интеграция:

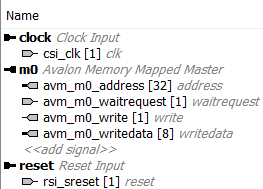


Рис. 1‑12 Настройки Signals & Interfaces в Platform Designer для my\_FSM\_master

В результате было получено новое ip-ядро в Platform Designer.

## Часть 1. Задача 4.

### Создание системы в PD

В Platform Designer было создано описание системы, включающее в себя:

* Модуль тактового сигнала
* ALU
* Четыре устройства STREAM\_GEN
* FSM\_master

Настройка компонентов приведена на рис. 1-13–1-15.

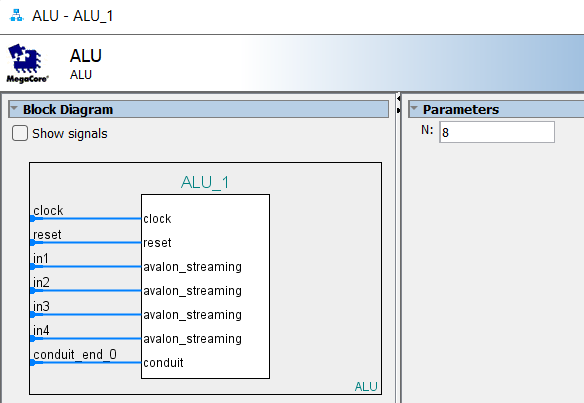


Рис. 1‑13 Настройки настройка АЛУ в Platform Designer

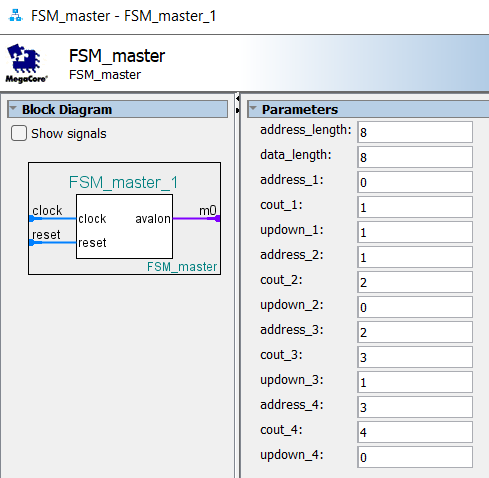


Рис. 1‑14. Настройки настройка конечного автомата в Platform Designer

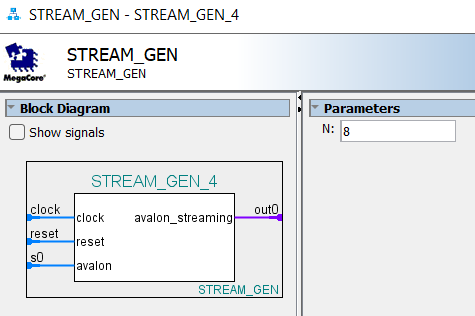


Рис. 1‑15. Настройки настройка генератора данных в Platform Designer

Рисунок построенной схемы представлен на рис. 1-16.

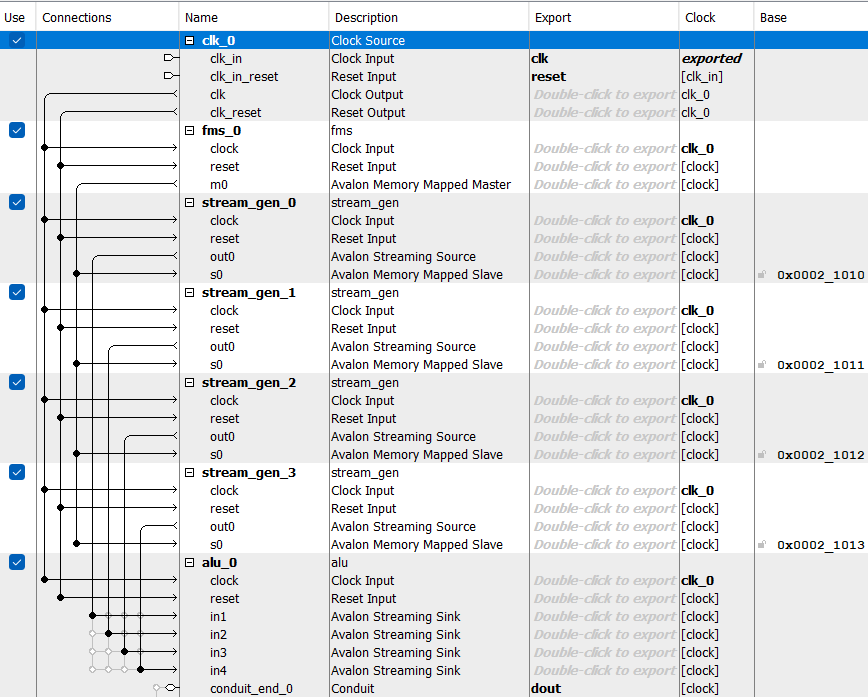


Рис. 1‑16. Схема описания системы в Platform Designer

Создадим описание созданной системы. Вкладка Generate представлена на рис. 1-17.

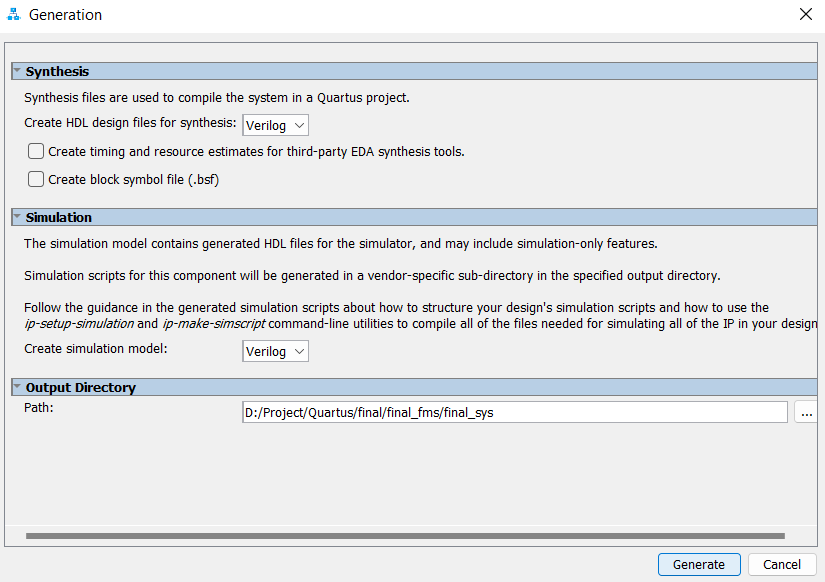
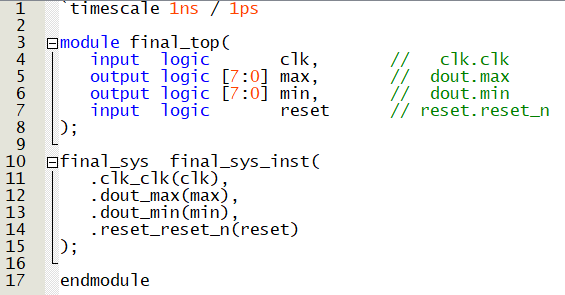


Рис. 1‑17. Генерация созданной системы

### Описание на языке SystemVerilog

Пере назовём шины для этого опишем файл верхнего уровня. Описание разрабатываемого устройства на языке SystemVerilog приведено ниже на листинге 1-7.

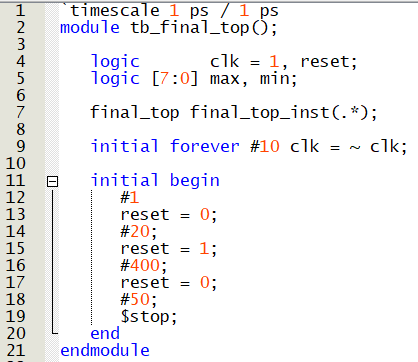
Листинг 1-7. final\_top.sv



### Моделирование

Проверим созданную системму. Для проверки правильности работы использовался тест первого класса (tb\_ final\_top.sv). В строках 4-6 обвялены входные и выходные значения; в строке 7 задана инициализация моделируемого модуля генератора данных; в строках 9-20 задано создание данных для проверки устройства. На листинге 1-8 приведено описание симуляции:

Листинг 1-8. tb\_final\_top.sv



Результаты моделирования представлены на рис. 1-18.

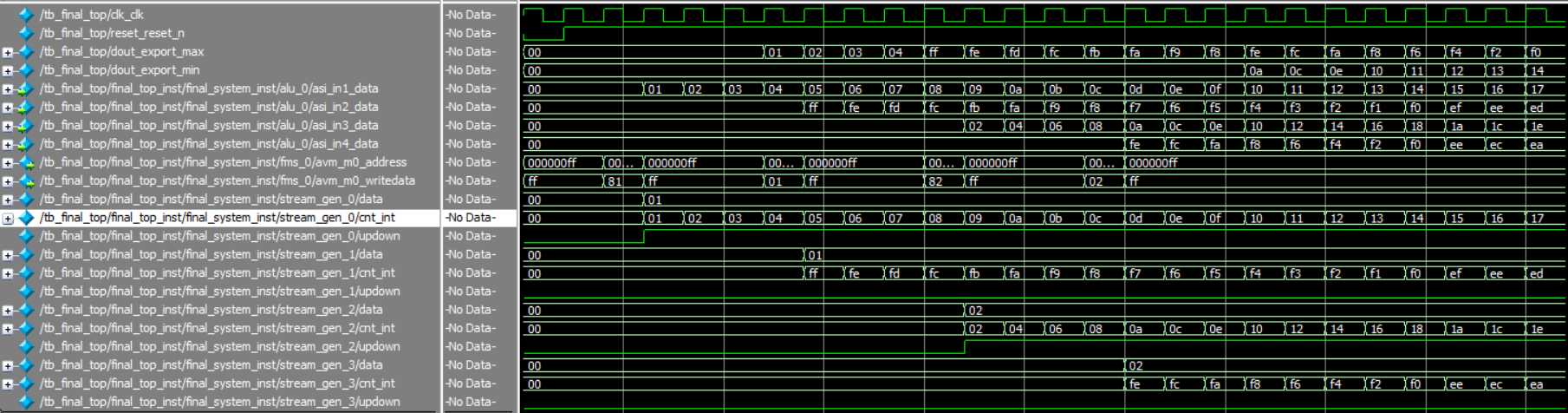


Рис. 1‑18. Результаты моделирования тестового файла tb\_ finally\_ex.sv (тип отображения - Heximal)

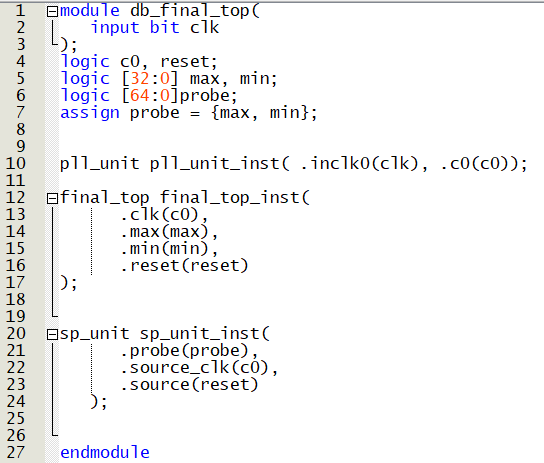
В результате симуляции доказанна работоспособность конечного автомата.

## Часть 1. Задача 5.

### Описание на языке SystemVerilog

Описание разрабатываемого устройства на языке SystemVerilog приведено ниже на листинге 1-9.

Листинг 1-9. db\_final\_top.sv



Настройка ip-ядра ALTPLL приведена на рис. 1-19 – 1-22.

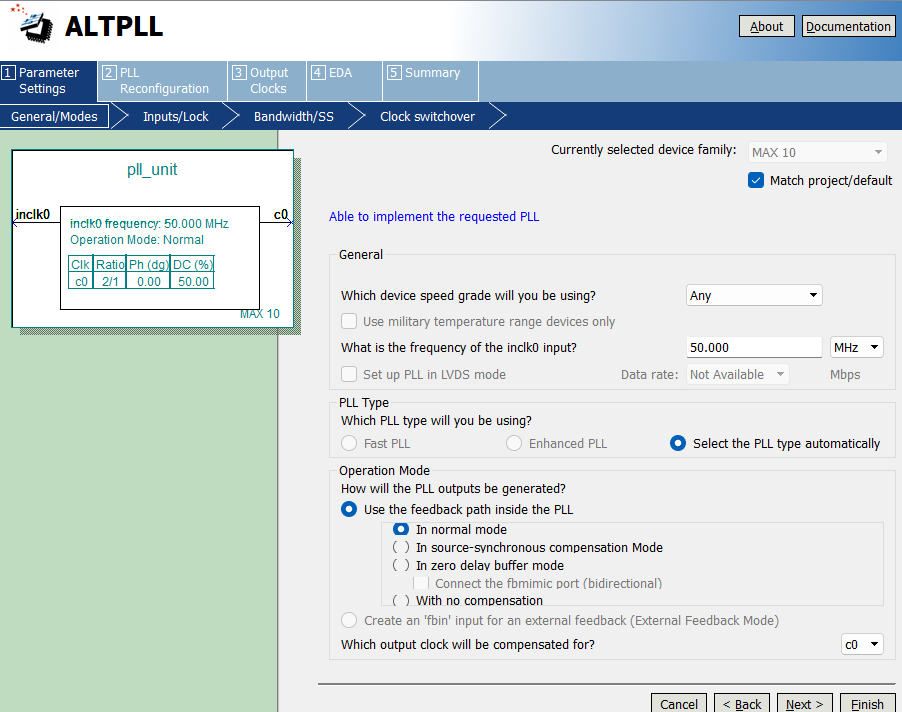


Рис. 1‑19. Настройка ALTPLL General

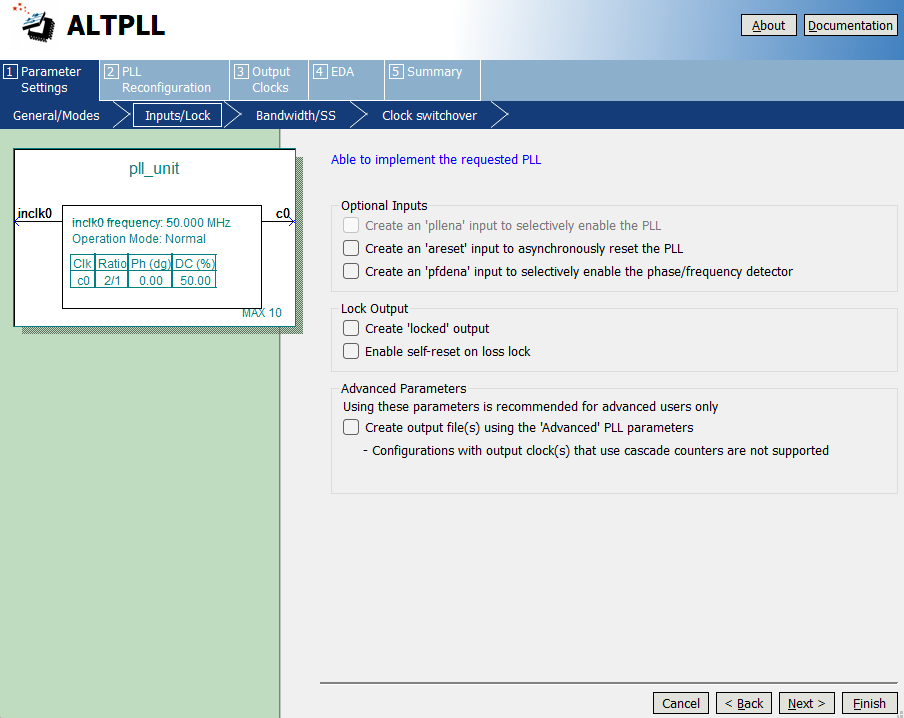


Рис. 1‑20. Настройка ALTPLL Input/Output

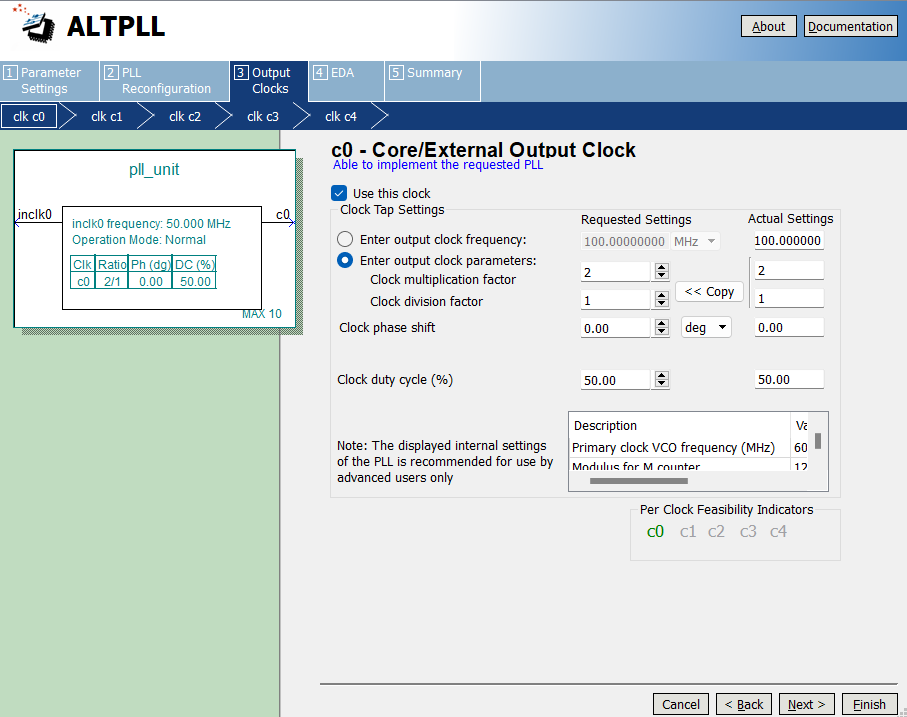


Рис. 1‑21. Настройка ALTPLL Clock

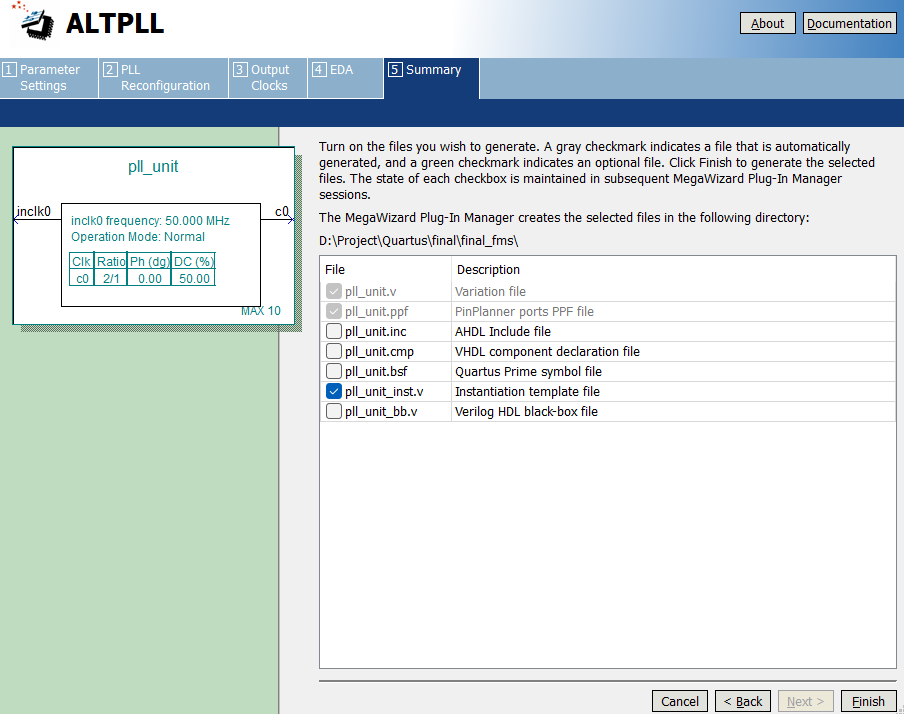


Рис. 1‑22. Настройка ALTPLL Summary

Настройка ip-ядра Altera In-System & Probes приведена на рис. 1-23.

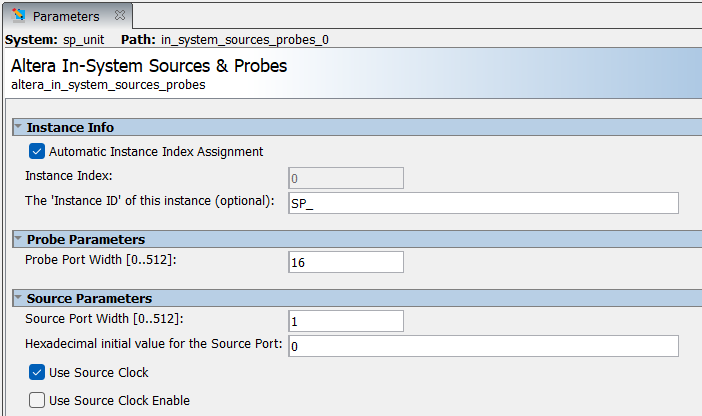


Рис. 1‑23. Настройка ALTPLL General

### Результат синтеза (RTL)

Результат синтеза описания на языке Verilog в пакете Quartus приведен ниже, на Рис. 1‑24. Изображение схемы получено с помощью приложения RTL Viewer.

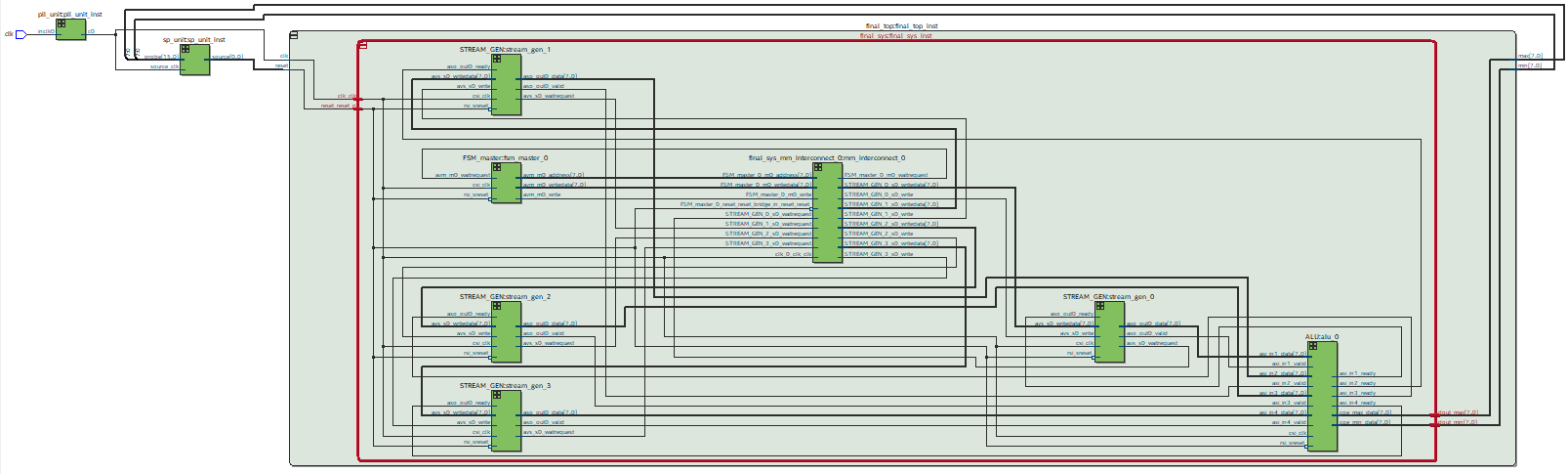


Рис. 1‑24 Синтезированная схема

### Проверка на плате с помощью Signal TapII

Настроим проверку на плате разработанной системы. Добавим сигналы и запустим захват данных на переход reset в позитивное состояние. На рис 1-25 показана настройка выводов в Signal TapII.

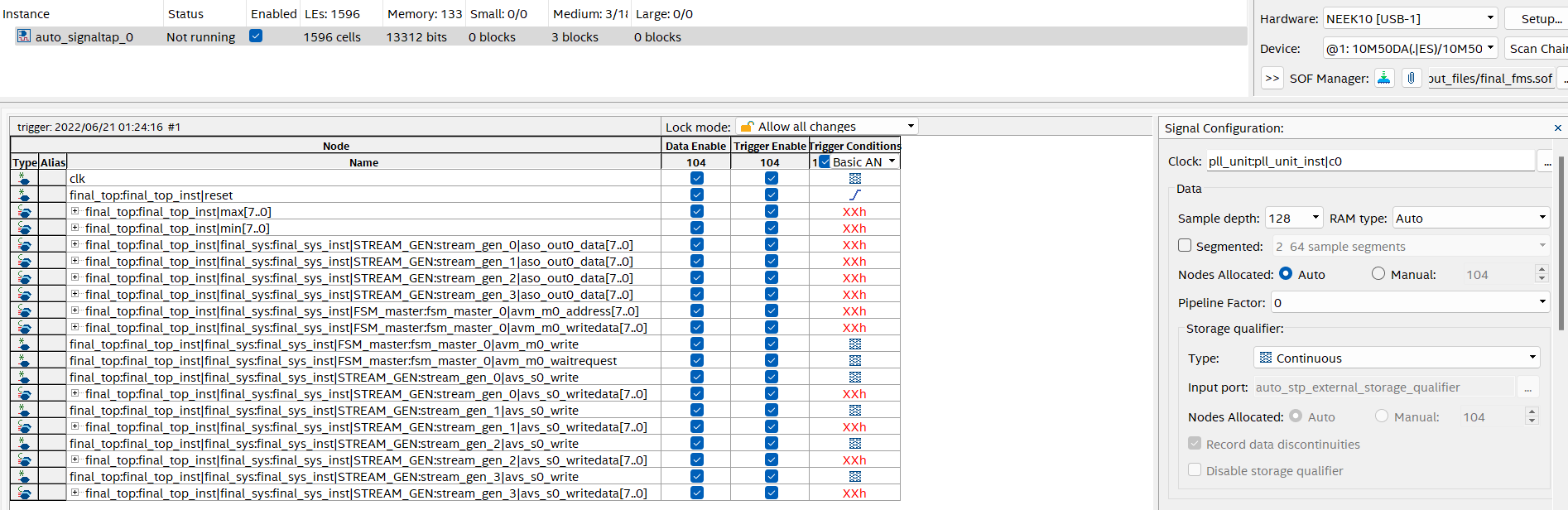


Рис. 1‑25. Настройка выводов в Signal TapII

Запустим и настроим In-System Sources & Probes как показано на рис. 1- 26. Затем установим reset в 1.

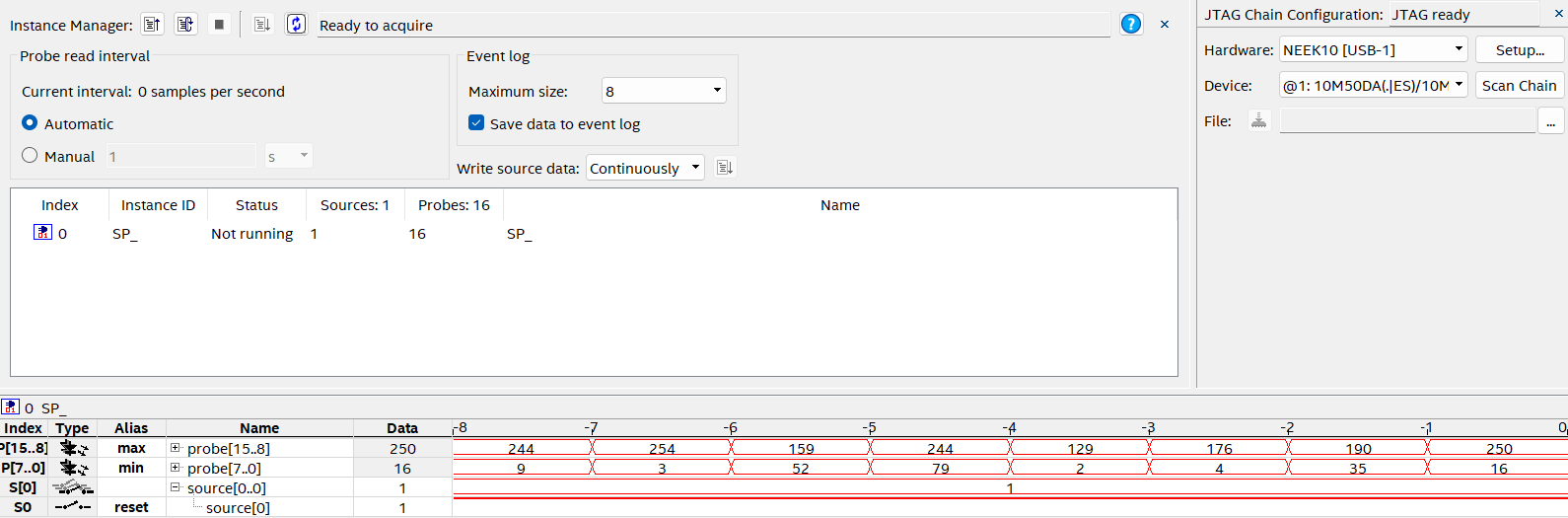


Рис. 1‑26. Настройка выводов в In-System Sources & Probes

Захваченные данные показаны на рис. 1-27.

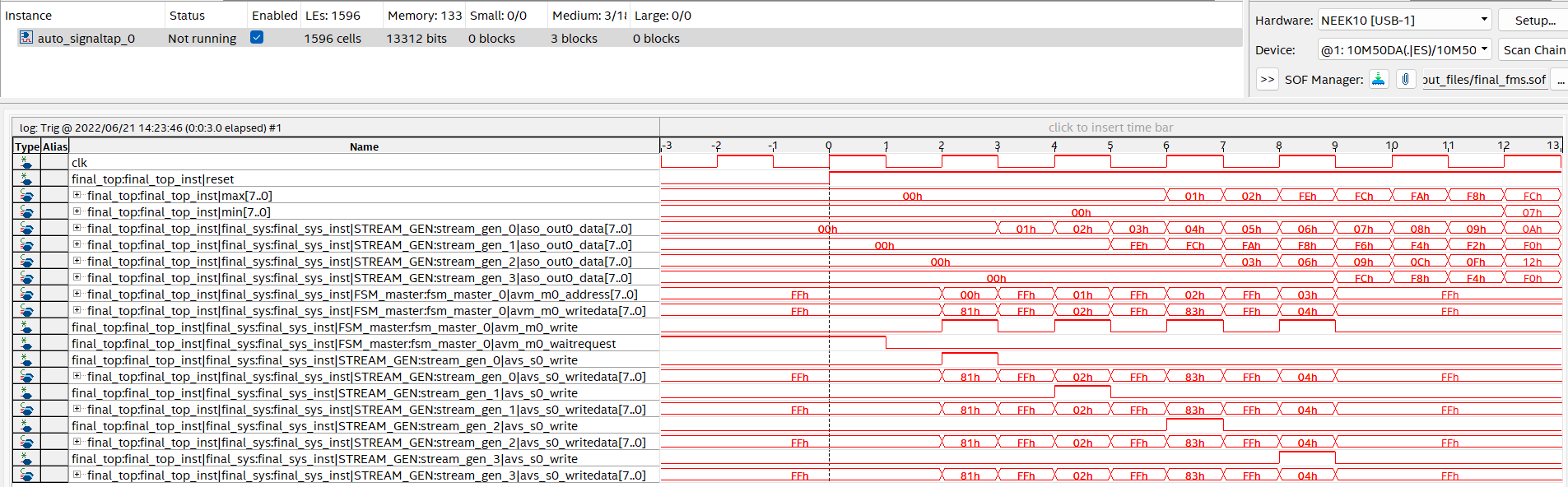


Рис. 1‑27. Временная диаграмма, которая получена с помощью In-System Sources & Probes и Signal TapII

## Часть 2. Задача 2.

### Создание системы в PD

В Platform Designer было создано описание системы, включающее в себя:

* Модуль тактового сигнала
* ALU
* Четыре устройства STREAM\_GEN
* FSM\_master

Настройка компонентов приведена на рис. 1-28–1-29.

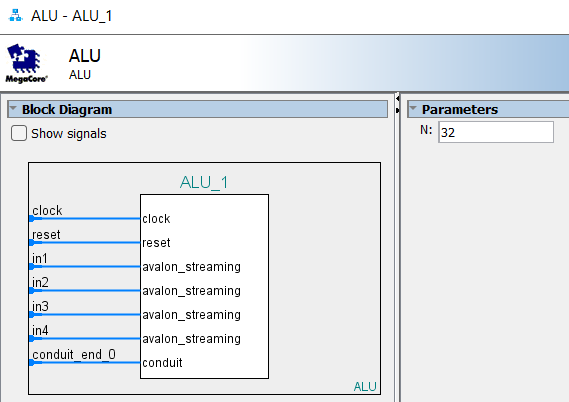


Рис. 1‑28 Настройки настройка АЛУ в Platform Designer

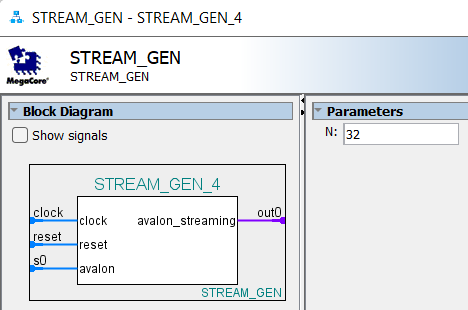


Рис. 1‑29. Настройки настройка генератора данных в Platform Designer

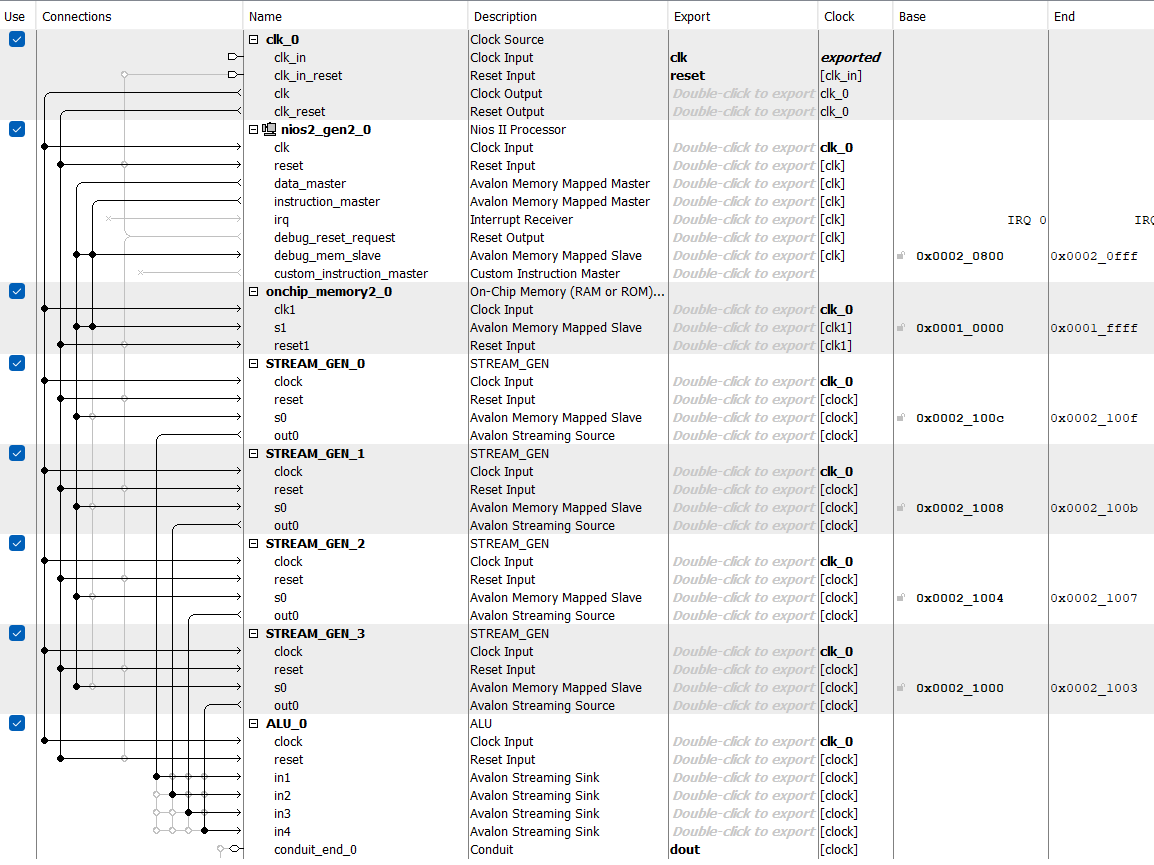
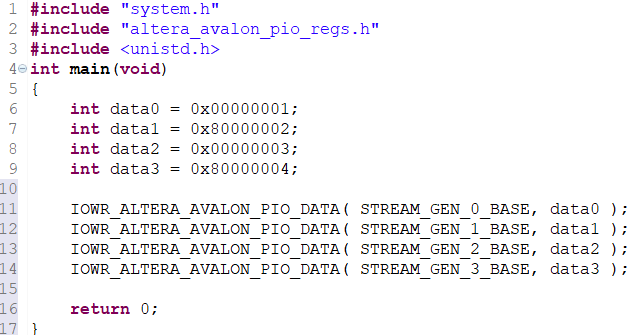
Рисунок построенной схемы представлен на рис. 1-30. 

Рис. 1‑30. Схема описания системы в Platform Designer

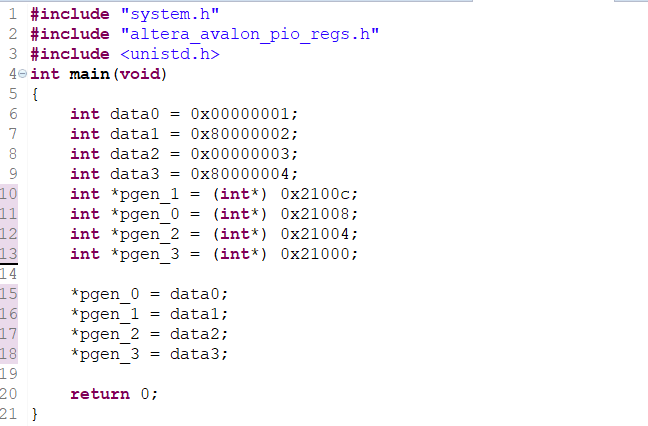
### Программирование Nios 2

Программа, созданная для софт-процессора Nios на языке С представлена в листинге 1-10 представлен вариант с библиотечными функциями, а в листинге 1-11 представлен вариант с адресами.

Листинг 1-10. final\_sourse.c библиотечные функции



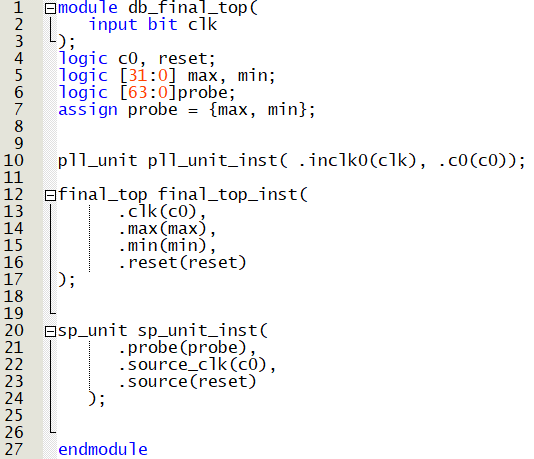
Листинг 1-11. final\_sourse.c задание адресами



### Отладка в Signal Tabe Logic Analyzer

Описание разрабатываемого устройства на языке SystemVerilog приведено ниже на листинге 1-12.

Листинг 1-12. final\_top.sv



### Результат синтеза (RTL)

Результат синтеза описания на языке Verilog в пакете Quartus приведен ниже, на Рис. 1‑30. Изображение схемы получено с помощью приложения RTL Viewer.

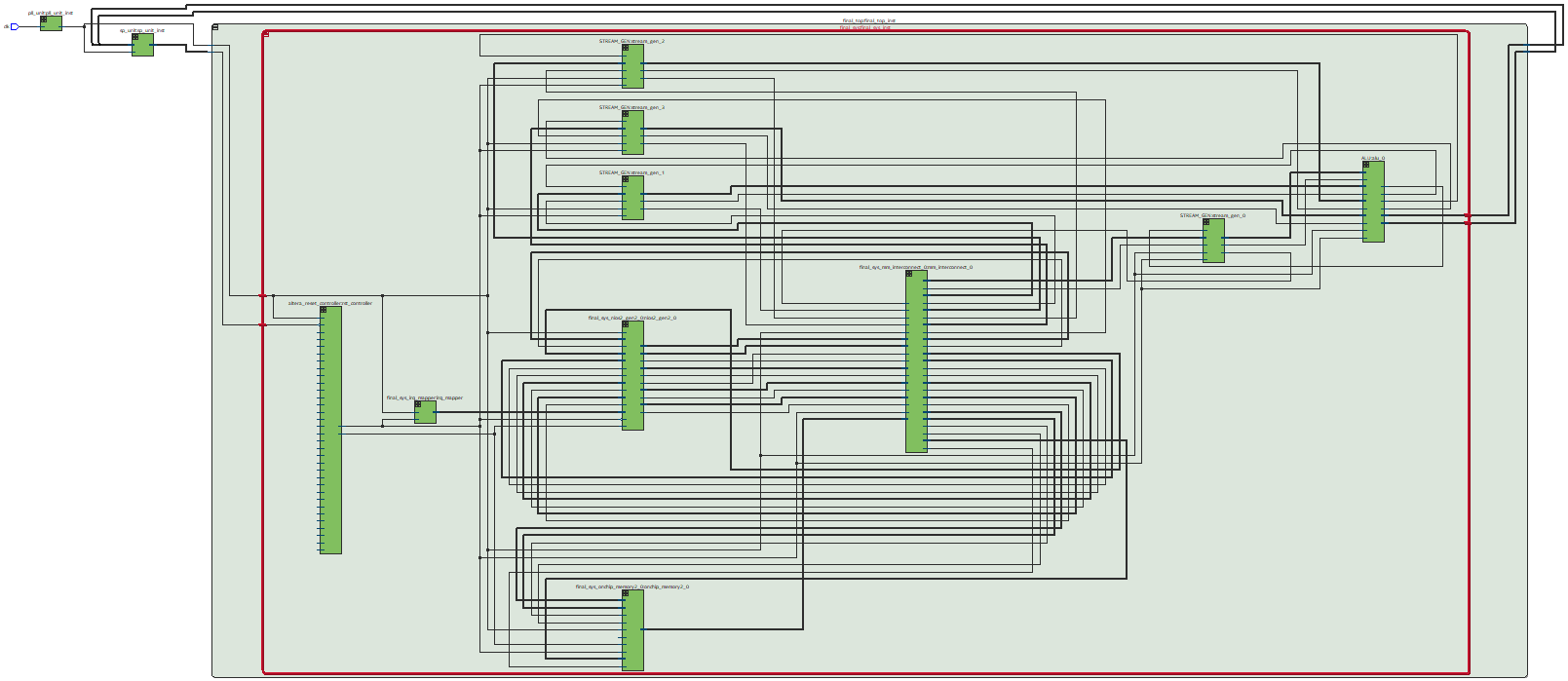


Рис. 1‑30 Синтезированная схема

### Проверка на плате с помощью Signal TapII

Настроим проверку на плате разработанной системы. Добавим сигналы и запустим захват данных на переход reset в позитивное состояние. На рис 1-31 показана настройка выводов в Signal TapII.

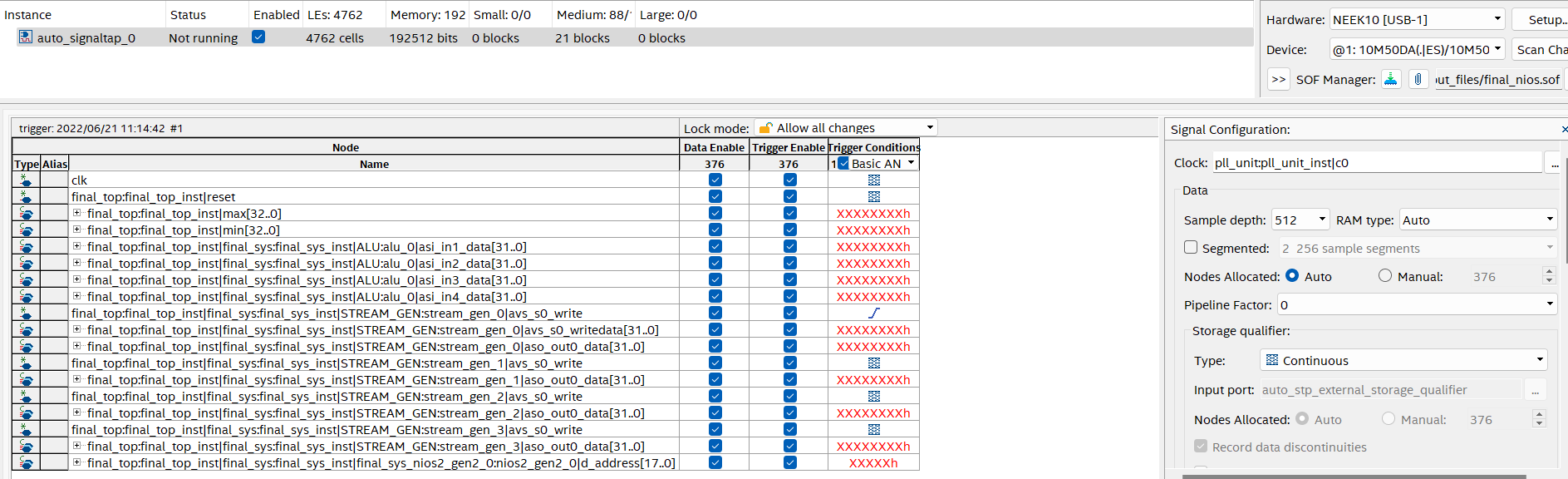


Рис. 1‑31. Настройка выводов в Signal TapII

Запустим и настроим In-System Sources & Probes как показано на рис. 1- 32. Затем установим reset в 1.

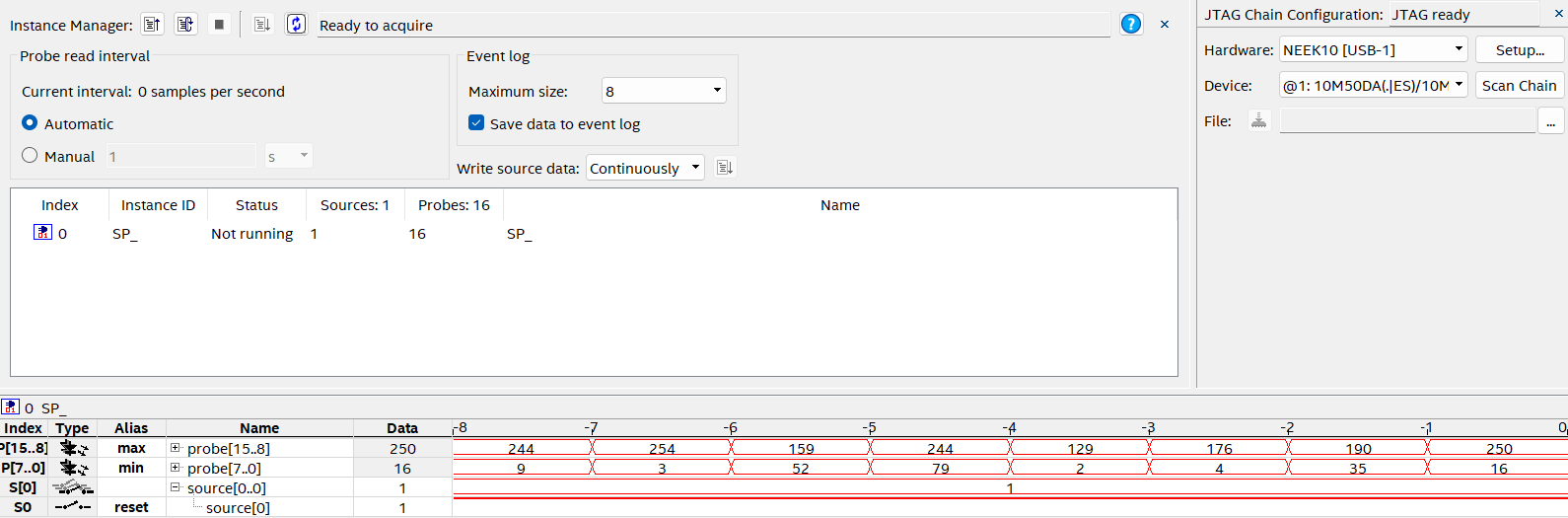


Рис. 1‑32. Настройка выводов в In-System Sources & Probes

Захваченные данные показаны на рис. 1-33.

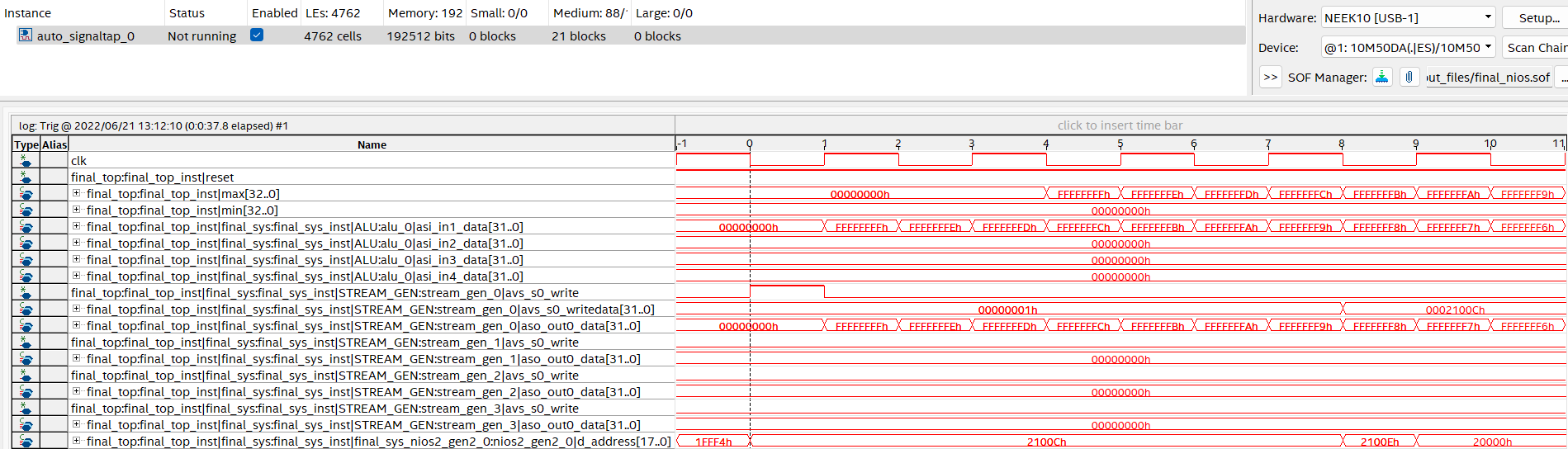


Рис. 1-33. Временная диаграмма, которая получена с помощью In-System Sources & Probes и Signal TapII

## Выводы

В результате был получен опыт разработки устройства для Platform Designer. Так же был приобретены навыки программирования для soft-процессора Nios 2. Разработанная система исправно функционирует и полностью соответствует заданию.