



# ITMO UNIVERSITY

**Университет ИТМО.  
Отчет по лабораторной работе номер 2  
по дисциплине «Функциональная схемотехника»**

**Выполнил :** Дробыш Дмитрий Александрович  
**Группа :** Р33082

**Преподаватель :** Кустарев Павел Валерьевич

## Цели работы

1. Получить навыки описания арифметических блоков на RTL-уровне с использованием языка описания аппаратуры Verilog HDL.

## Задание

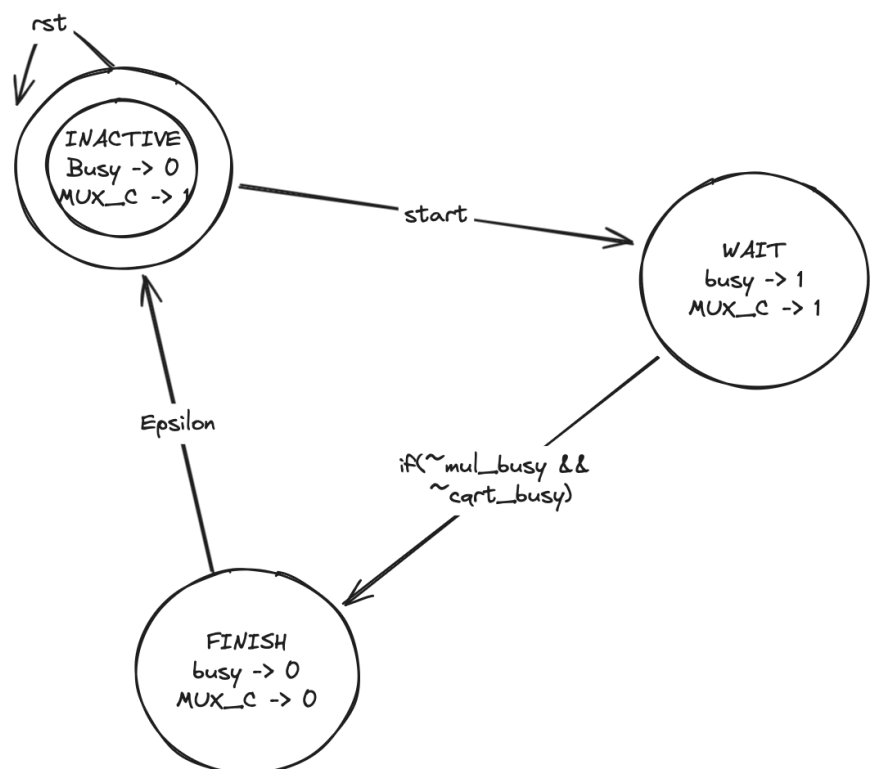
Вариант	Функция	Ограничения
5	$3 \cdot a + 2 \cdot b^{\left(\frac{1}{3}\right)}$	1 сумматор и 2 умножителя

Таблица 1. Задание на ЛР2

## Порядок выполнения работы

1. Разработайте и опишите на Verilog HDL схему, вычисляющую значение функции в соответствии с заданными ограничениями согласно варианту задания.
2. Определите область допустимых значений функции.
3. Разработайте тестовое окружение для разработанной схемы. Тестовое окружение должно проверять работу схемы не менее, чем на 10 различных тестовых векторах.
4. Проведите моделирование работы схемы и определите время вычисления результата. Схема должна тактироваться от сигнала с частотой 100 МГц.
5. Составьте отчет по результатам выполнения работы.

## Выполнение работы



STATE MACHINE внутри

Рис1. Конечный автомат модуля function

Схема основного блока Functions.

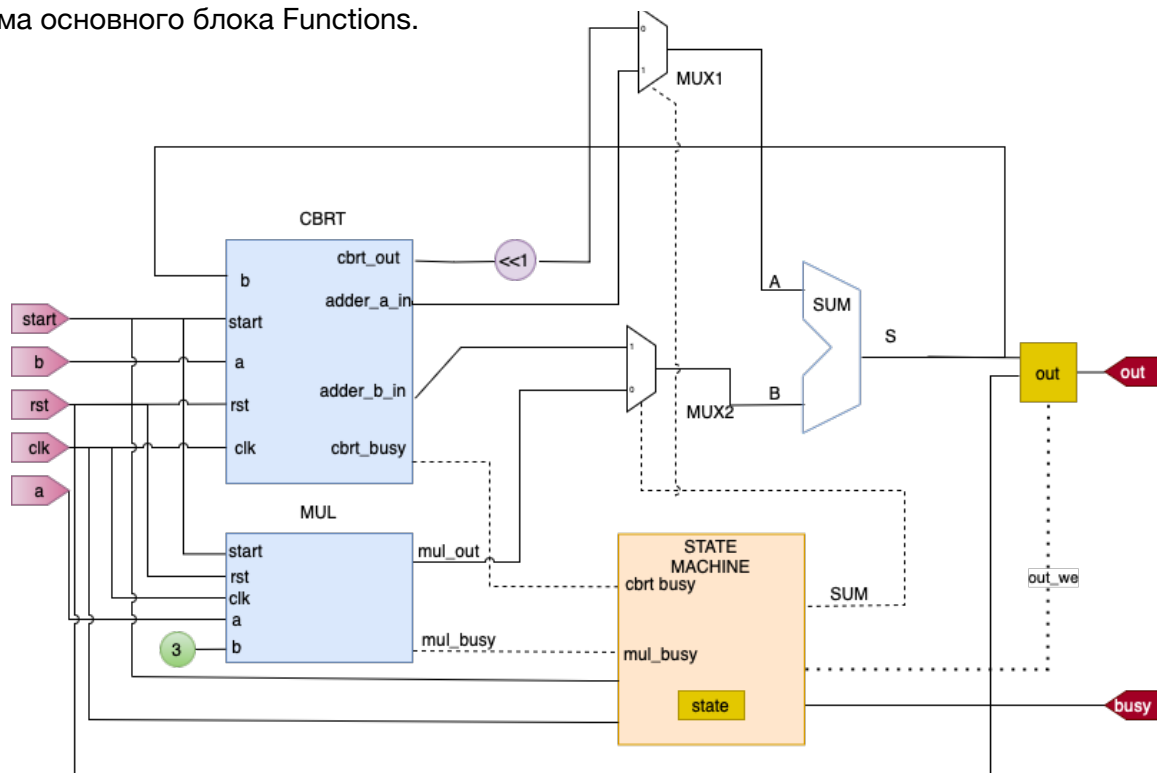


Рис2. Схема модуля function

## Описание работы модуля

На вход модулю:

clk - тактовый сигнал,  
rst - сигнал прекращения вычислений,  
start - сигнал старта вычислений,  
a - 8бит. число,  
b - 8бит. число.

На выход :

out - 10 битный результат вычисления функции  $3 \cdot a + 2 \cdot b^{(\frac{1}{3})}$   
busy - сигнал работы модуля.

Модуль function использует **один** сумматор и **один** умножитель.

Модуль вычисления кубического корня cbrt **содержит в себе умножитель**.

cbrt принимает на вход:

clk - тактовый сигнал,  
rst - сигнал прекращения вычислений,  
start - сигнал старта вычислений,  
a - 8-битный операнд  
adder\_s\_out - 16-битный выход сумматора.

На выход cbrt идут:

res - 4битный результат вычисления корня,,  
busy - сигнал работы модуля,  
adder\_a\_in и adder\_b\_in - два 16-битных аргумента внешнего сумматора.

## Область допустимых значений

Так как а и b — 8-битные числа,

$3 \cdot a$  - 10 разрядное число, не превышающее 765, а  $2 \cdot b^{\frac{1}{3}}$  - 4битное число не превышающее 12. То есть максимальный результат работы модуля это  $765 + 12 = 777$  - 10-битное число.

## Результаты тестирования

Adder:

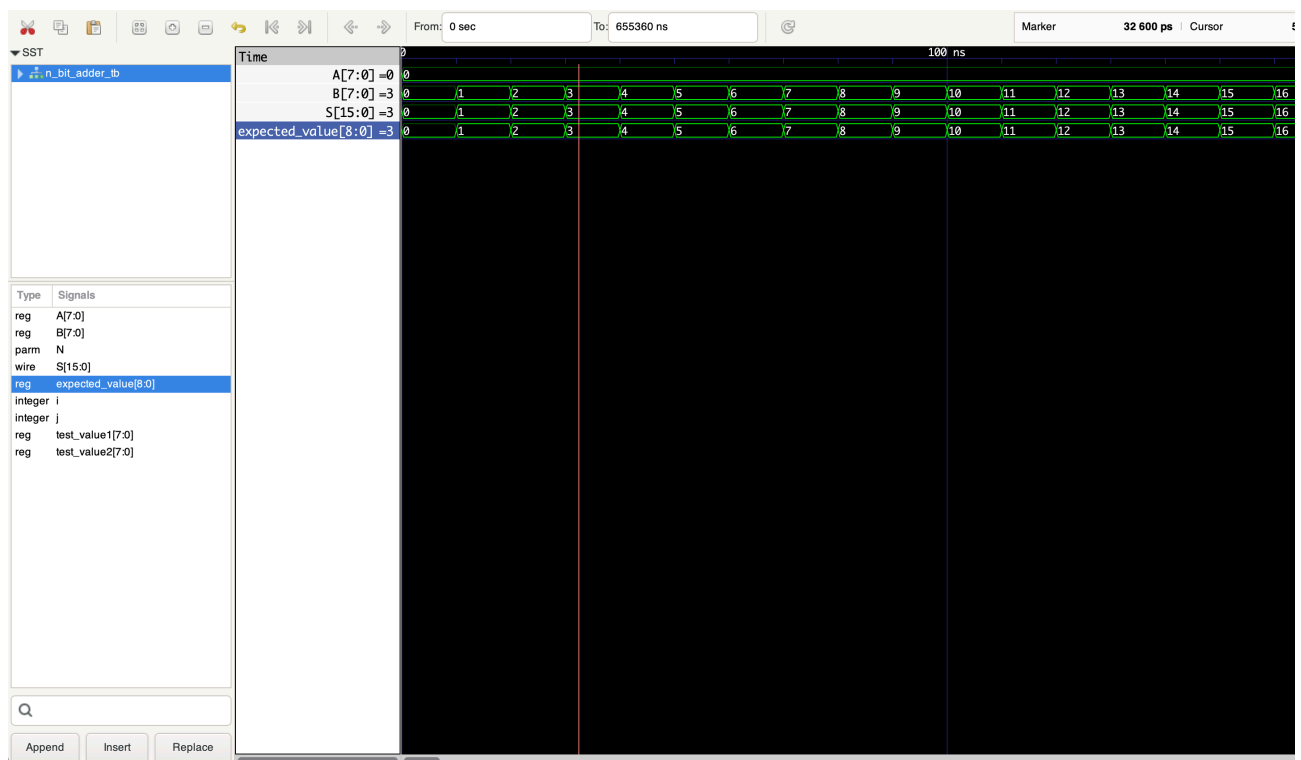


Рис3. Временная диаграмма тестирования сумматора

Test Passed!

10| a= 0, b= 1, s= 1,

Test Passed!

20| a= 0, b= 2, s= 2,

Test Passed!

30| a= 0, b= 3, s= 3,

Test Passed!

40| a= 0, b= 4, s= 4,  
 Test Passed!  
 50| a= 0, b= 5, s= 5,  
 Test Passed!  
 60| a= 0, b= 6, s= 6,  
 Test Passed!  
 70| a= 0, b= 7, s= 7,  
 Test Passed!  
 80| a= 0, b= 8, s= 8,  
 Test Passed!  
 90| a= 0, b= 9, s= 9,  
 Test Passed!  
 100| a= 0, b= 10, s= 10,  
 Test Passed!  
 110| a= 0, b= 11, s= 11,  
 Test Passed!  
 120| a= 0, b= 12, s= 12,  
 Test Passed!  
 130| a= 0, b= 13, s= 13,  
 Test Passed!  
 140| a= 0, b= 14, s= 14,  
 Test Passed!  
 150| a= 0, b= 15, s= 15,  
 Test Passed!  
 160| a= 0, b= 16, s= 16,  
 Test Passed!  
 170| a= 0, b= 17, s= 17,

Multiplier:



Рис4. Временная диаграмма тестирования умножителя

Test passed! a = 0, b = 0, Found = 0, Expected = 0  
 Test passed! a = 0, b = 1, Found = 0, Expected = 0  
 Test passed! a = 0, b = 2, Found = 0, Expected = 0  
 Test passed! a = 0, b = 3, Found = 0, Expected = 0  
 Test passed! a = 0, b = 4, Found = 0, Expected = 0  
 Test passed! a = 0, b = 5, Found = 0, Expected = 0  
 Test passed! a = 0, b = 6, Found = 0, Expected = 0  
 Test passed! a = 0, b = 7, Found = 0, Expected = 0  
 Test passed! a = 0, b = 8, Found = 0, Expected = 0  
 Test passed! a = 0, b = 9, Found = 0, Expected = 0

Test passed! a = 0, b = 10, Found = 0, Expected = 0  
 Test passed! a = 0, b = 11, Found = 0, Expected = 0  
 Test passed! a = 0, b = 12, Found = 0, Expected = 0  
 Test passed! a = 0, b = 13, Found = 0, Expected = 0  
 Test passed! a = 0, b = 14, Found = 0, Expected = 0  
 Test passed! a = 0, b = 15, Found = 0, Expected = 0  
 Test passed! a = 0, b = 16, Found = 0, Expected = 0

Cube root:

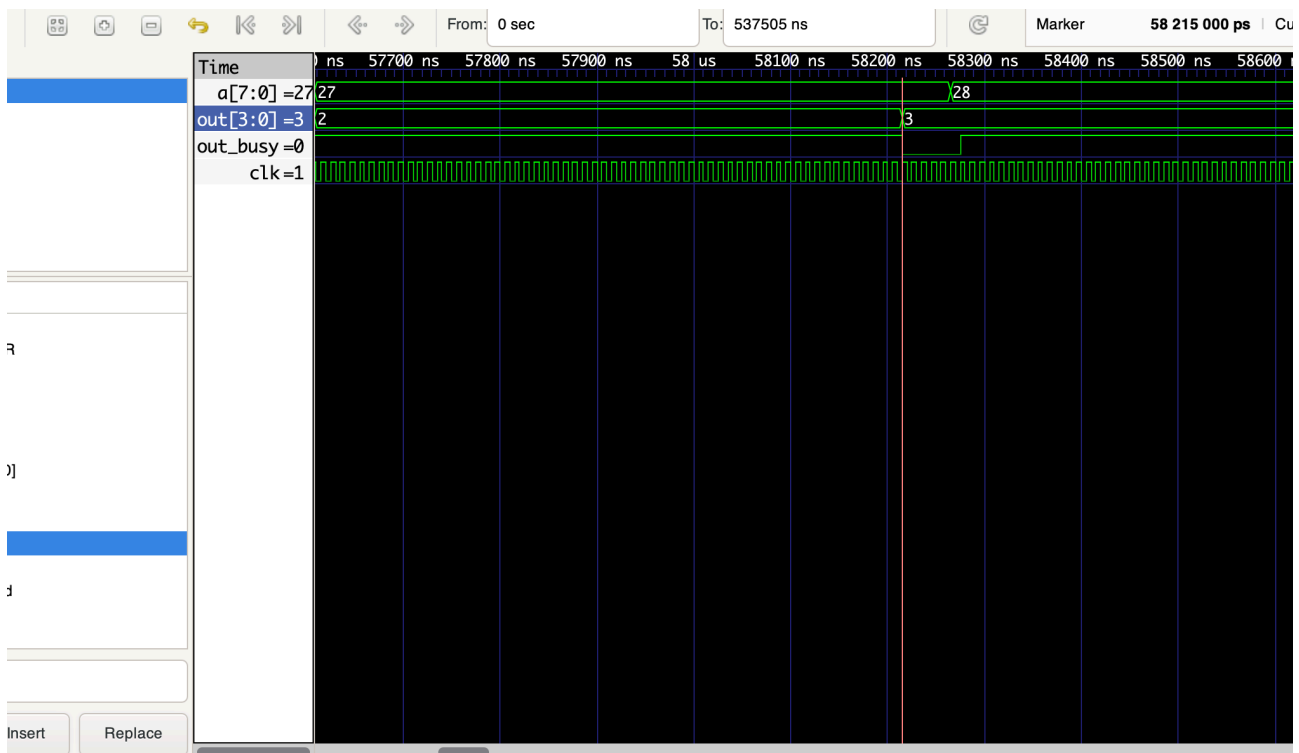
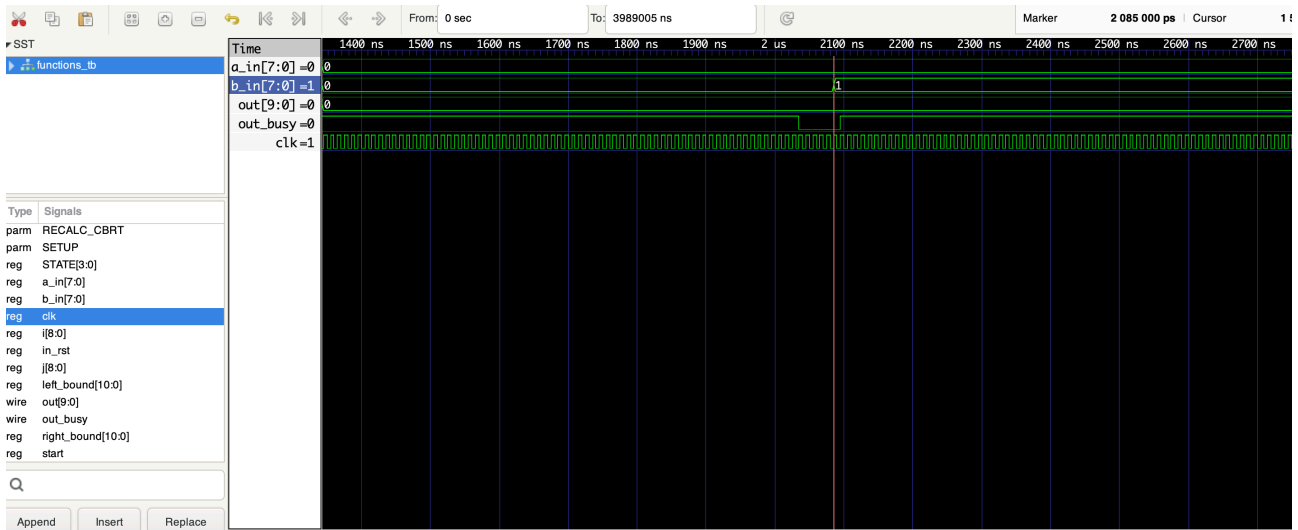


Рис5. Временная диаграмма тестирования кубического корня

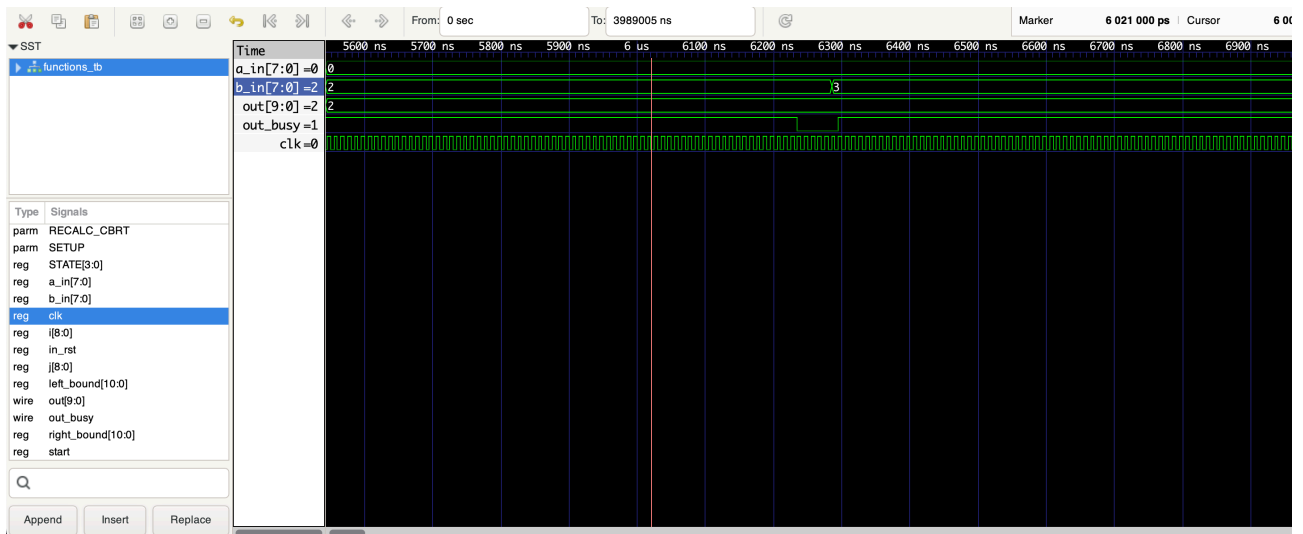
Test Passed! For : 0, Root : 0  
 Test Passed! For : 1, Root : 1  
 Test Passed! For : 2, Root : 1  
 Test Passed! For : 3, Root : 1  
 Test Passed! For : 4, Root : 1  
 Test Passed! For : 5, Root : 1  
 Test Passed! For : 6, Root : 1  
 Test Passed! For : 7, Root : 1  
 Test Passed! For : 8, Root : 2  
 Test Passed! For : 9, Root : 2  
 Test Passed! For : 10, Root : 2  
 Test Passed! For : 11, Root : 2  
 Test Passed! For : 12, Root : 2  
 Test Passed! For : 13, Root : 2  
 Test Passed! For : 14, Root : 2  
 Test Passed! For : 15, Root : 2  
 Test Passed! For : 16, Root : 2

Function:



Рисб. Временная диаграмма тестирования основного модуля

EST PASSED! a = 43, b = 190, y = 139  
TEST PASSED! a = 43, b = 191, y = 139  
TEST PASSED! a = 43, b = 192, y = 139  
TEST PASSED! a = 43, b = 193, y = 139  
TEST PASSED! a = 43, b = 194, y = 139  
TEST PASSED! a = 43, b = 195, y = 139  
TEST PASSED! a = 43, b = 196, y = 139  
TEST PASSED! a = 43, b = 197, y = 139  
TEST PASSED! a = 43, b = 198, y = 139  
TEST PASSED! a = 43, b = 199, y = 139  
TEST PASSED! a = 43, b = 200, y = 139  
TEST PASSED! a = 43, b = 201, y = 139  
TEST PASSED! a = 43, b = 202, y = 139  
TEST PASSED! a = 43, b = 203, y = 139  
TEST PASSED! a = 43, b = 204, y = 139  
TEST PASSED! a = 43, b = 205, y = 139  
TEST PASSED! a = 43, b = 206, y = 139  
TEST PASSED! a = 43, b = 207, y = 139  
TEST PASSED! a = 43, b = 208, y = 139  
TEST PASSED! a = 43, b = 209, y = 139  
TEST PASSED! a = 43, b = 210, y = 139  
TEST PASSED! a = 43, b = 211, y = 139



## Время работы

На графике видно, что при падении busy\_o, выводится верный результат. Вычисление занимает 404 такта. При частоте 100 Мгц вычисление займёт 4040 нс.

## Выводы

Были смоделированы и успешно протестированы арифметические модули вычисления кубического корня и  $3 \cdot a + 2 \cdot b^{\frac{1}{3}}$ .