



Лабораторная работа №2

по дисциплине: Функциональная схемотехника

Вариант: 7

Выполнил: Неграш Андрей, Р33301

Преподаватель: Салонина Екатерина Александровна

Санкт-Петербург, 2023

Цель работы

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

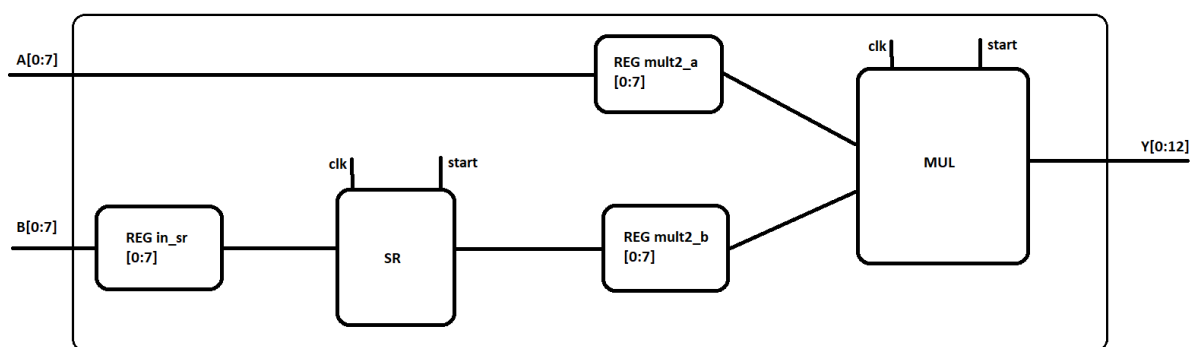
Задание в соответствии с вариантом

Вариант: 7

Функция: $y = a * \sqrt{b}$

Ограничения: 2 сумматора и 2 умножителя

Схема разработанного блока вычисления функции в терминах БОЭ



В модуле вычисления квадратного корня (SR) используется 1 сумматор, а в модуле (MUL) - 1 умножитель. Таким образом данная схема успешно вписывается в заданные ограничения.

Описание работы разработанного блока

На вход разработанного блока подаются следующие сигналы:

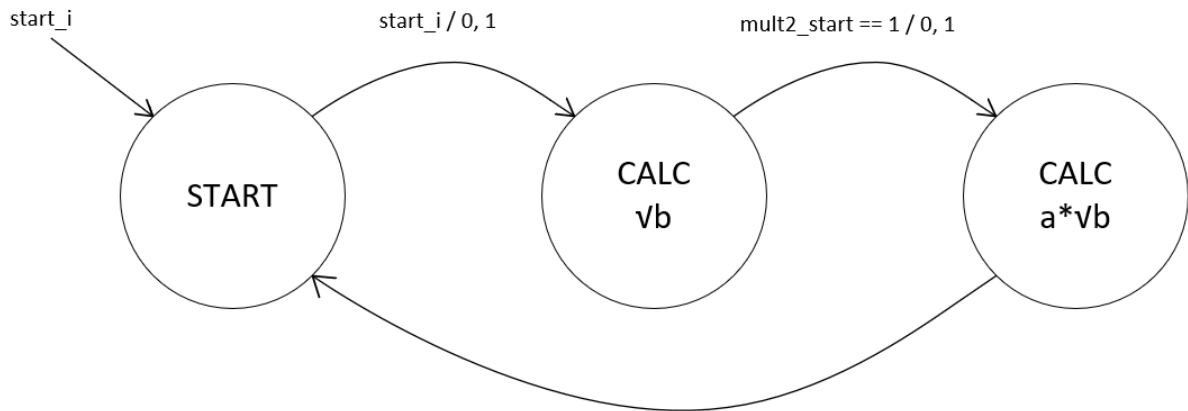
- start – сообщающий о начале вычислений
- in_rst – сигнал сброса
- clk_reg – посылающий синхроимпульсы
- in_a – аргумент A
- in_b – аргумент B
- count – сигнал для обозначения итерации тестирования (вспомогательный)

Алгоритм вычисления можно представить следующим образом:

- Разработанный модуль получает на вход 1 по сигналу start
- Сразу же после start у модуля выставляется значение 1 в выход busy_o, который означает, что началась работа модуля
- На вход извлекающего квадратный корень модуля подаётся значение in_b и ожидается выполнение извлечения квадратного корня
- результат извлечения квадратного корня подаётся в качестве аргумента на умножитель вместе со считываемым значением in_a

- результат работы умножителя подаются на выход

Конечный автомат Мура:



ОДЗ для разработанного блока

Согласно заданию, оба операнда функции являются беззнаковыми целыми числами разрядностью 8 бит. Следовательно максимальное число, которое можно получить на выходе функции $255 \cdot 15 = 3825$. В двоичном виде это число выглядит так: 111011110001. Для записи данного числа потребуется 12 бит, соответственно и выходной сигнал будет иметь 12-битную разрядность.

Результат тестирования разработанного блока

Код тестирования доступен в function_test.v и имеет следующий вид:

```
initial begin
    in_rst = 'd1;
    start = 'd0;
    count = 'd0;
    #1
    in_rst = 'd0;
    count_failed = 0;

    for (i = 0; i <= 255; i = i+1) begin
        for (j = 0; j <= 255; j = j+1) begin
            in_a = i;
            in_b = j;
            start = 1;
            @(negedge out_busy);
            start = 0;
            expected = i * $floor($sqrt(j));
            if (expected == out) begin
                $display("Correct! a= %d, b=%d, y=%d", in_a, in_b, out);
            end else begin
                $display("Wrong((( a= %d, b=%d, y=%d", in_a, in_b, out);
                count_failed = count_failed + 1;
            end
        end
    end

    if (count_failed == 0) begin
        $display("All tests passed successfully!");
    end else begin
        $display("Testing is not completed. Find %d failed test(s).", count_failed);
    end
end
```

Все тесты проходят успешно. Скриншот вывода в консоль прилагаю:

```
Correct! a= 255, b=250, y=3825
Correct! a= 255, b=251, y=3825
Correct! a= 255, b=252, y=3825
Correct! a= 255, b=253, y=3825
Correct! a= 255, b=254, y=3825
Correct! a= 255, b=255, y=3825
All tests passed successfully!
run: Time (s): cpu = 00:00:10 ; elapsed = 00:00:06 . Memory (MB): peak = 980.191 ; gain = 21.551
```

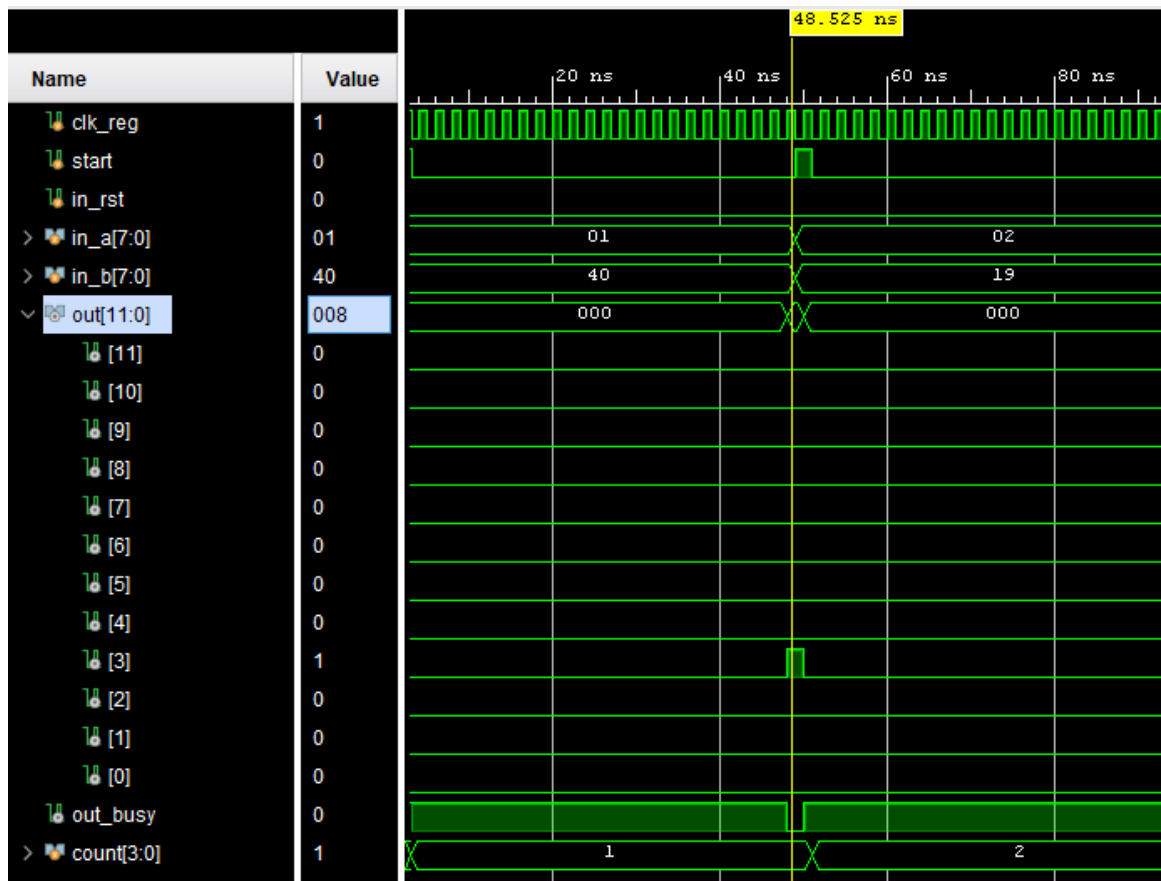
Временная диаграмма тестирования выглядит следующим образом:



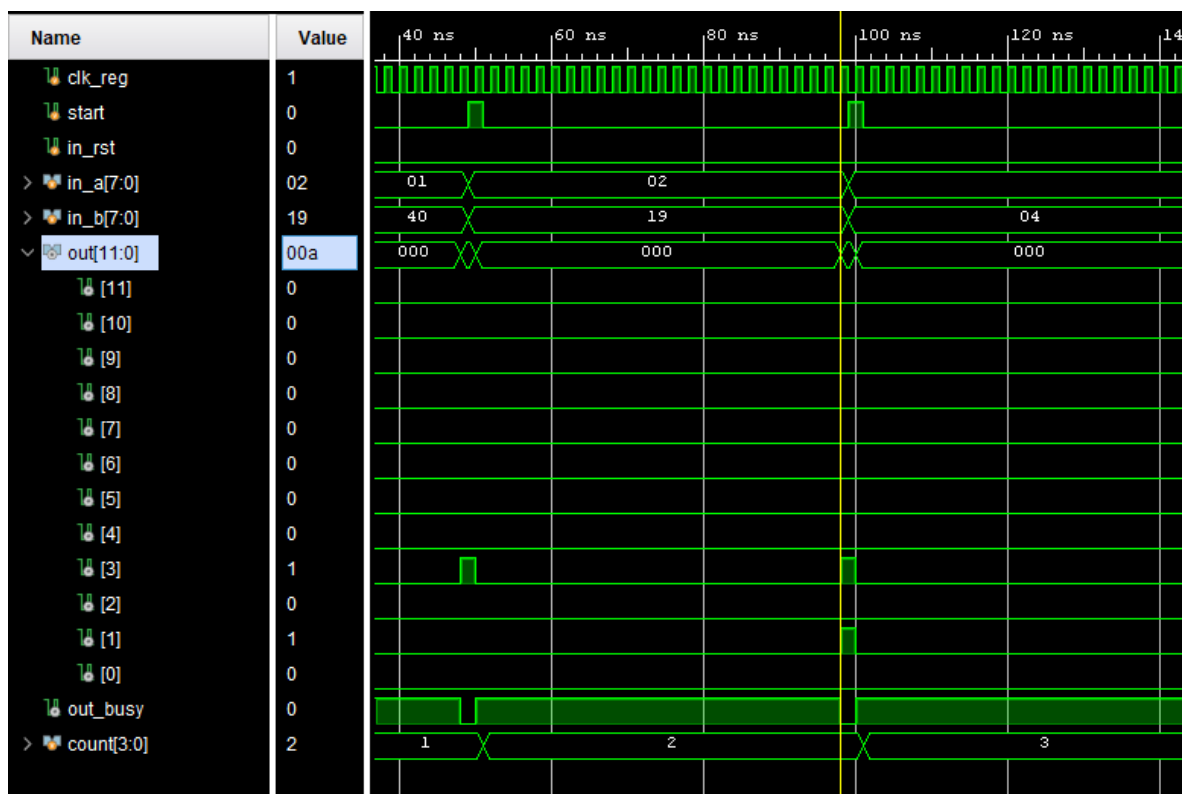
После вывода результата значение Y обнуляется со следующим синхроимпульсом. Все тесты работают верно, выдавая правильный результат.

Ниже представлены ещё несколько скриншотов тестирования, на которых видно, что на выходе out появляется правильный ответ. Все входные и выходные данные представлены в 16-ричной системе счисления.

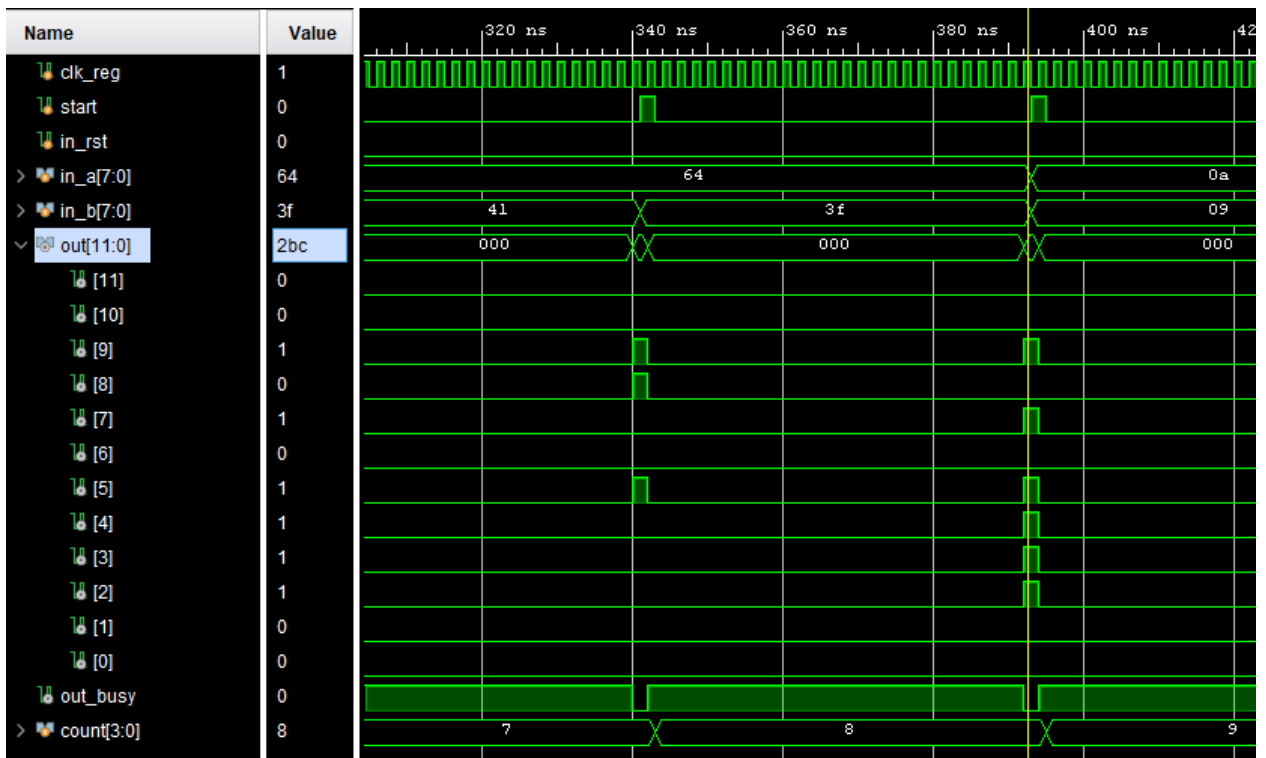
Тест 1 (a=1, b=40, y=8)



Тест 2 (a=2, b=19, y=A)

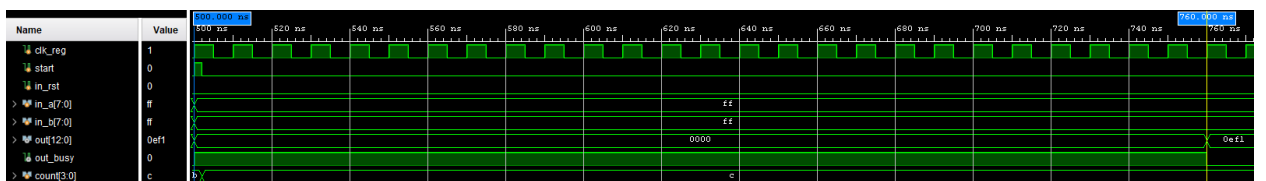


Тест 3 (a=64, b=3F, y=2BC)



Время вычисления результата при частоте в 100МГц

Тактовая частота в 100МГц – это 1 синхроимпульс за 10нс. Увеличим задержку с 1нс до 5нс в блоке работы часов. Временная диаграмма тестирования для самых больших входных чисел из ОДЗ:



По скриншоту временной диаграммы видно, что на вычисление результата функции при тактовой частоте 100МГц потребовалось 260нс.

Вывод

Во время выполнения данной лабораторной работы я разработал блок вычисления функции по заданному преподавателем варианту и реализовал его на языке описания аппаратуры Verilog, а также протестировал его работоспособность при помощи 11 различных тестовых векторов.