



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.

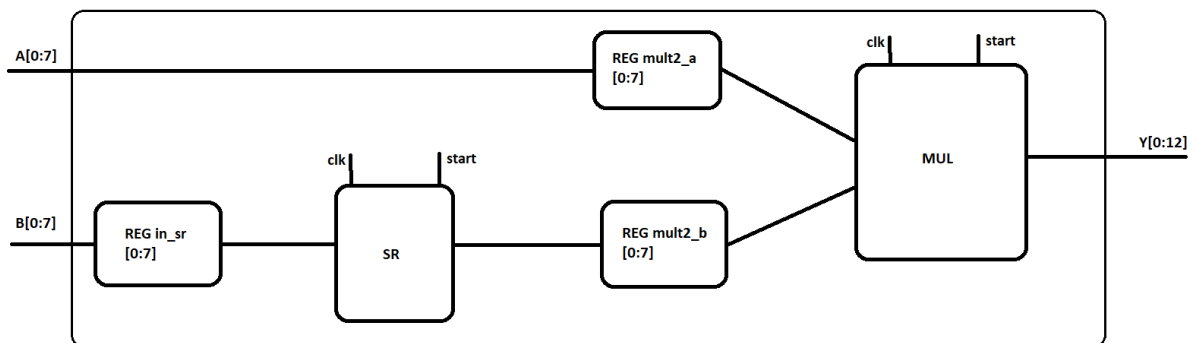
Задание в соответствии с вариантом

Вариант: 7

Функция: $y = a * \sqrt{b}$

Ограничения: 2 сумматора и 2 умножителя

Схема разработанного блока вычисления функции в терминах БОЭ



Описание работы разработанного блока

На вход разработанного блока подаются следующие сигналы:

- `start` – сообщаящий 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

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 для тестирования разработанного блока доступен в файле function_test.v и имеет следующий вид:

```
initial begin
    in_rst = 'd1;
    start = 'd0;
    count = 'd0;
    #1
    in_rst = 'd0;
    forever
        #1
        if (~out_busy && ~start && count == 0) begin//8(0x8)
            in_a = 'd1;
            in_b = 'd64;
            start = 1;
            #1
            start = 0;
            count = count + 1;
        end else if (~out_busy && ~start && count == 1) begin//10(0xa)
            in_a = 'd2;
            in_b = 'd25;
            start = 1;
            #2
            start = 0;
            count = count + 1;
        end else if (~out_busy && ~start && count == 2) begin//6(0x6)
            in_a = 'd3;
            in_b = 'd4;
            start = 1;
            #2
            start = 0;
            count = count + 1;
        end else if (~out_busy && ~start && count == 3) begin//0(0x0)
            in_a = 'd3;
            in_b = 'd0;
            start = 1;
            #2
            start = 0;
            count = count + 1;
```

Всего разработано 11 тестовых векторов, принимающих различные значения как граничного вида (минимальные и максимальные значения аргументов, проверка округления с шагом 1), так и просто случайные наборы.

Ниже представлена таблица входных и выходных параметров тестов:

№ (count)	A ₁₀	B ₁₀	Y ₁₀	Y ₂ (12 бит)
0	1	64	8	0000 0000 1000
1	2	25	10	0000 0000 1010
2	3	4	6	0000 0000 0110
3	3	0	0	0000 0000 0000
4	5	1	5	0000 0000 0101
5	6	121	66	0000 0100 0010
6	100	65	800	0011 0010 0000
7	100	63	700	0010 1011 1100
8	10	9	30	0000 0001 1110
9	1	10	3	0000 0000 0011
10	0	0	0	0000 0000 0000
11	255	255	3825	1110 1111 0001

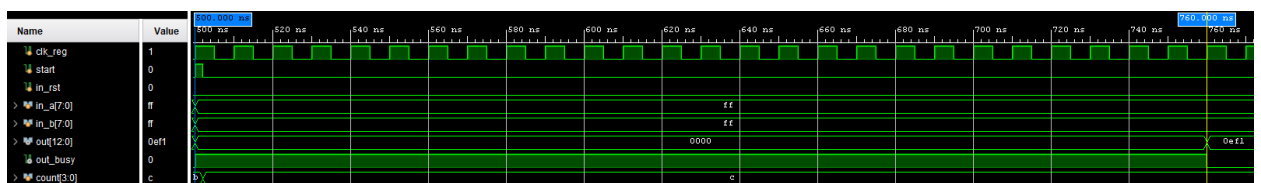
Временная диаграмма тестирования выглядит следующим образом:



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

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.