Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**Факультет Пиикт**

Дисциплина

‘Функциональная схемотехника’

**ЛАБОРАТОРНАЯ РАБОТА № 2**

Разработка аппаратных ускорителей математических вычислений

Вариант №-1

*Выполнил:*

Студент группы P3333

Гуменник

Петр Олегович

*Преподаватель:*

Табунщик Сергей Михайлович



Санкт-Петербург, 2024

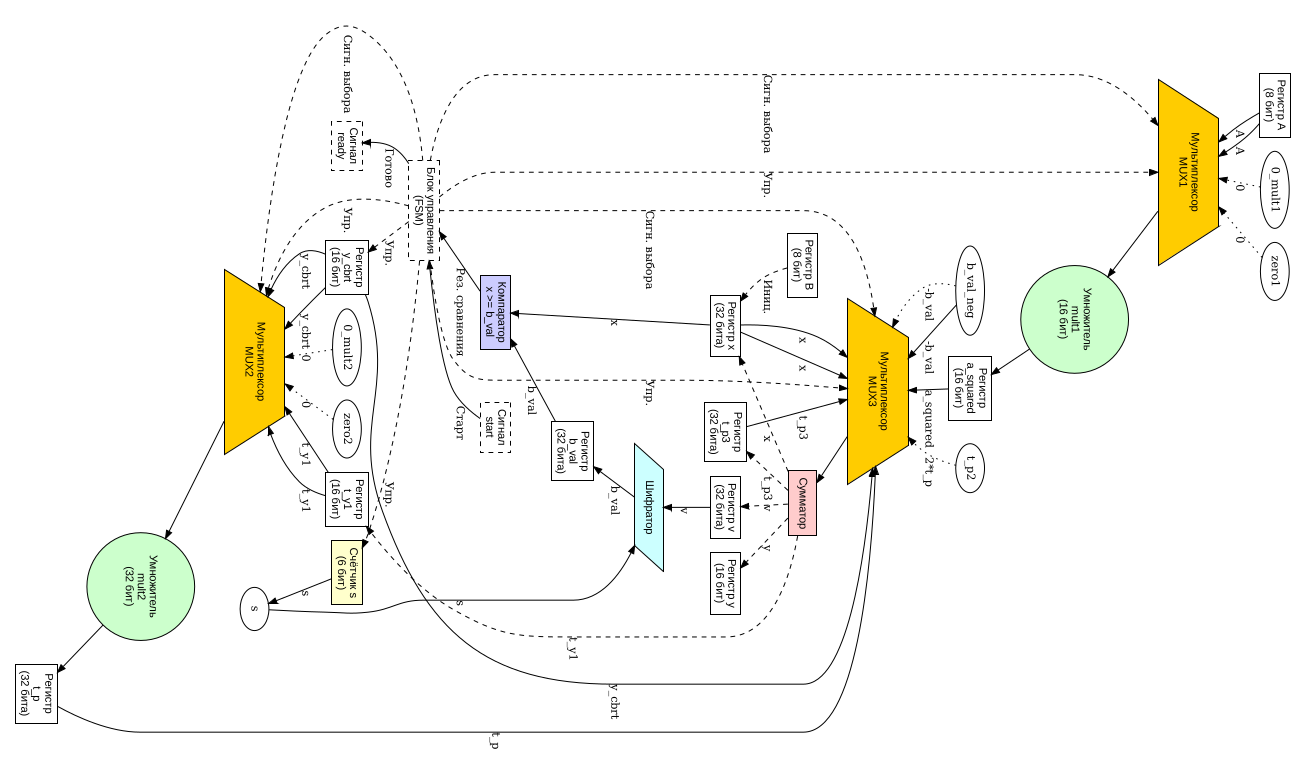
**Цели работы:**

Получить навыки описания арифметических блоков на RTL-уровне с использо-

ванием языка описания аппаратуры Verilog HDL.

**Задание по варианту:**

Реализовать функцию , с учетом ограничений: 1 сумматор и 2 умножителя.



1. Схема разработанного блока вычисления функции



2. Схема состояний FSM с сигналами переходов

**Описание работы блока**

1. Инициализация:

При подаче сигнала reset все регистры и флаги устанавливаются в начальное состояние.

Блок находится в состоянии IDLE, ожидая сигнала start.

2. Ввод данных:

При подаче сигнала start значения a и b записываются в соответствующие регистры A и B.

3. Вычисление :

FSM переходит в состояние START\_A2.

Умножитель mult1 получает значения a с регистра A.

Запускается умножение .

FSM переходит в состояние WAIT\_A2, ожидая завершения умножения.

После завершения умножения результат записывается в регистр a\_squared.

4. Инициализация для вычисления кубического корня:

FSM переходит в состояние INIT\_CUBERT.

Устанавливаются начальные значения:

x = b (расширенное до 32 бит).

y\_cbrt = 0.

s = 30.

5. Итерационный процесс для вычисления :

Цикл выполняется, пока s >= 0:

Сдвиг y\_cbrt влево на 1 бит (SHIFT\_Y):

y\_cbrt = y\_cbrt << 1.

Вычисление t\_y1 = y\_cbrt + 1 (ADD\_Y1): Используется сумматор.

Вычисление (START\_YP):

Умножитель mult2 получает операнды y\_cbrt и t\_y1.

Запускается умножение.

FSM переходит в состояние WAIT\_YP, ожидая завершения умножения.

Результат записывается в регистр t\_p.

Вычисление (CALC\_TP3):

t\_p3 = t\_p + 2 \* t\_p. Используется сумматор и сдвиги.

Вычисление v = t\_p3 + 1 (ADD\_ONE): Используется сумматор.

Сдвиг v влево на s бит (SHIFT\_V):

b\_val = v << s.

Сравнение x и b\_val (COMPARE):

Если x >= b\_val, переходим к UPDATE\_XY, иначе к NEXT\_ITER.

Обновление x и y\_cbrt (UPDATE\_XY):

x = x - b\_val.

y\_cbrt = y\_cbrt + 1.

Переход к следующей итерации (NEXT\_ITER):

s = s - 3.

Если s >= 0, возвращаемся к SHIFT\_Y, иначе переходим к ADD\_FINAL.

6. Завершение вычислений:

Вычисление y = a\_squared + y\_cbrt (ADD\_FINAL):

Используется сумматор.

Результат записывается в регистр y.

Установка флага ready: Указывает на завершение вычислений.

FSM переходит в состояние FINISH, ожидая сброса сигнала start.

7. Готовность к следующему вычислению:

После того, как сигнал start становится низким, FSM возвращается в состояние IDLE, готовый принять новые данные.сание работы разработанного блока

**Область допустимых значений**

Входные данные:

: 8-битное беззнаковое число.

: 8-битное беззнаковое число.

Вычисляемые величины:

:

Минимальное значение: .

Максимальное значение: .

:

Значения кубического корня для принимают значения от 0 до 6.

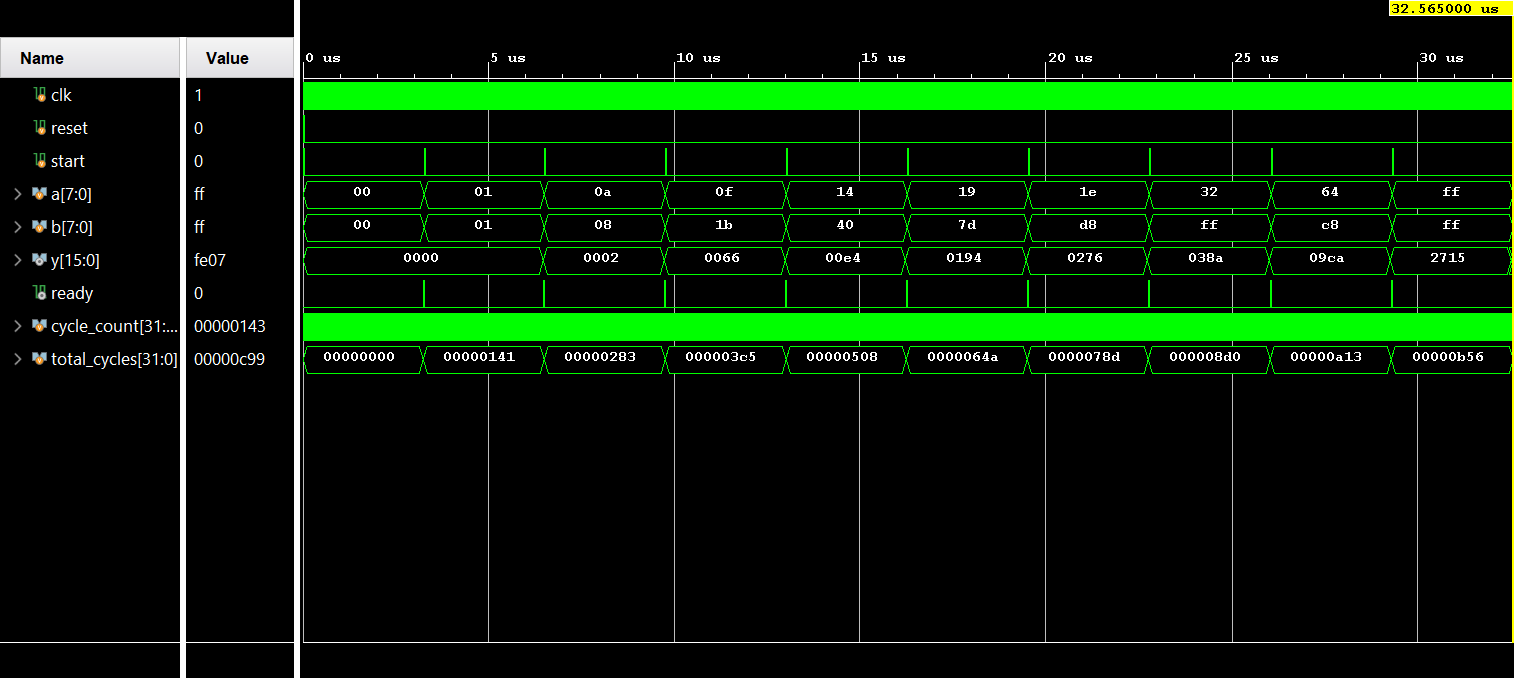
:

Минимальное значение: .

Максимальное значение: .

Вывод: Результат лежит в диапазоне от 0 до 65,031. Для хранения результата достаточно 16-битного беззнакового регистра.

**Время вычисления результата при частоте тактового сигнала в 100 МГц** – примерно 3210-3230 наносекунд.



**Вывод:**

В этом проекте был разработан модуль Verilog для вычисления функции , где a и b - 8-битные беззнаковые входы, а представляет собой целую часть кубического корня из b. Реализация придерживается строгих ограничений, используя только один сумматор и два умножителя. Для соблюдения этих ограничений был разработан собственный 16-битный модуль последовательного умножения и интегрирован в основной вычислительный модуль. Также был создан тест бенч для проверки функциональности и измерено время вычислений при тактовой частоте 100 МГц.