



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«МИРЭА - Российский технологический университет»**

**РТУ МИРЭА**

---

Институт Информационных Технологий  
Кафедра Вычислительной Техники (ВТ)

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1**

«Аппаратные ресурсы ПЛИС. Арифметические операции»

по дисциплине

«Схемотехника устройств компьютерных систем»

Выполнил студент группы

Стецюк В.В.

ИББО-08-22

Принял ассистент кафедры ВТ

Дуксин Н.А.

Практическая работа выполнена

«\_\_»\_\_\_\_\_2024 г.

«Зачтено»

«\_\_»\_\_\_\_\_2024 г.

## **АННОТАЦИЯ**

Данная работа включает в себя 26 рисунков и 6 листингов. Общее количество страниц в работе — 19.

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 РЕАЛИЗАЦИЯ ФУНКЦИИ С БЕЗЗНАКОВЫМИ ОПЕРАНДАМИ .....	5
1.1 Реализация функции и анализ задействованных для этого аппаратных ресурсов.....	5
1.2 Реализация функции с операндами большей разрядности и анализ задействованных для этого аппаратных ресурсов.....	7
2 РЕАЛИЗАЦИЯ ФУНКЦИИ СО ЗНАКОВЫМИ ОПЕРАНДАМИ.....	11
2.1 Реализация функции со знаковыми операндами и анализ задействованных для этого аппаратных ресурсов .....	11
2.2 Реализация функции со знаковыми операндами большей разрядности и анализ задействованных для этого аппаратных ресурсов .....	13
ЗАКЛЮЧЕНИЕ .....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	19

## ВВЕДЕНИЕ

В данной практической работе необходимо рассмотреть различные варианты реализации заданной схемы, согласно персональному варианту и проанализировать результаты синтеза и имплементации.

Персональный вариант:  $a - b \ll c * d$

# 1 РЕАЛИЗАЦИЯ ФУНКЦИИ С БЕЗЗНАКОВЫМИ ОПЕРАНДАМИ

## 1.1 Реализация функции и анализ задействованных для этого аппаратных ресурсов

Название модуля верхнего уровня – «main». Модуль имеет входные порты «a», «b», «c» и «d», а также выходной порт «res».

С помощью оператора непрерывного присваивания «assign» к выходному порту «res» подключается результат выражения, данного по персональному варианту: «a - b << c \* d».

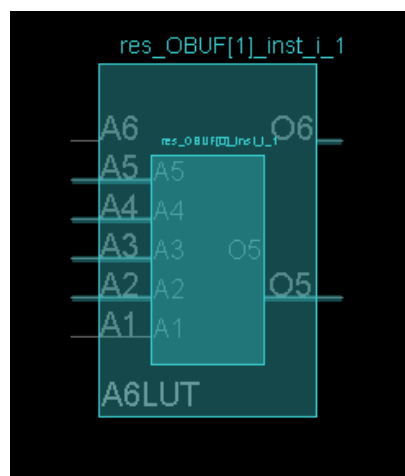
Код модуля верхнего уровня представлен в Листинге 1.1 [1].

*Листинг 1.1 – Реализация функции с беззнаковыми операндами*

```
module main(  
    input a, b, c, d,  
    output [1:0] res);  
  
    assign res = a - b << c * d;  
  
endmodule
```

Анализируя данный код, имеем входы в один бит размером, следовательно, логика вычислений не будет сложной.

Произведены синтез и имплементация [2]. Использованные аппаратные ресурсы представлены на Рисунках 1.1, 1.2 и 1.3.



**Рисунок 1.1 — Размещение модуля на устройстве**

Resource	Utilization	Available	Utilization %
LUT	1	63400	0.00
IO	6	210	2.86

Рисунок 1.2 — Отчет по использованию аппаратных ресурсов

Ref Name	Used	Functional Category
IBUF	4	IO
OBUF	2	IO
LUT4	2	LUT

Рисунок 1.3 — Расширенный отчет по использованию аппаратных ресурсов

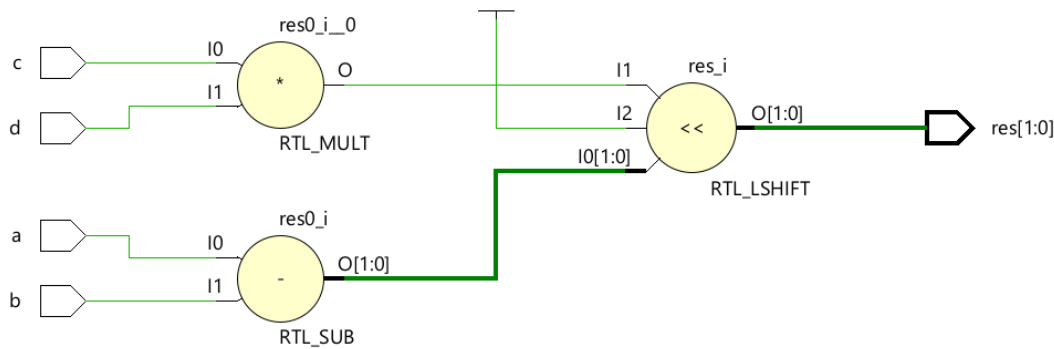


Рисунок 1.4 — RTL схема

Cell Properties					
res_OBUF[1]_inst_i_1					
I3	I2	I1	I0	O=I0 & I1 & !I2 & I3 + I2 & !I3	
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	1	
0	1	0	1	1	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	0	

Рисунок 1.5 — Таблица истинности

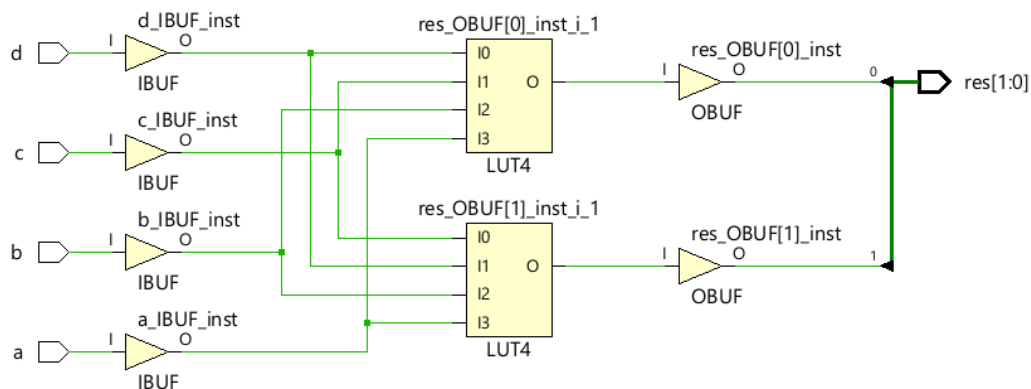


Рисунок 1.6 — Схема на этапе имплементации

## 1.2 Реализация функции с операндами большей разрядности и анализ задействованных для этого аппаратных ресурсов

После увеличения ширины входов и выходов можно увидеть, что использование умножения, как аппаратно-затратной операции, приводит к быстрому росту количества использованных LUT.

Код модуля верхнего уровня для реализации функции с операндами разрядности 3 представлен в Листинге 1.2.

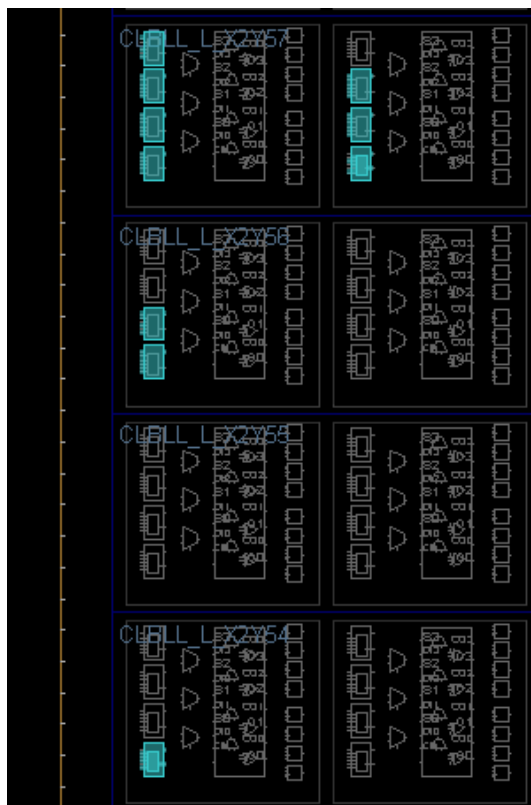
Листинг 1.2 – Реализация функции с беззнаковыми операндами разрядности 3

```
module main(
    input [2:0] a,
    input [2:0] b,
    input [2:0] c,
    input [2:0] d,
    output [3:0] res);

    assign res = a - b << c * d;

endmodule
```

Произведены синтез и имплементация. Использованные аппаратные ресурсы представлены на Рисунках 1.3, 1.4 и 1.5.



**Рисунок 1.7 — Размещение модуля на устройстве**

Resource	Utilization	Available	Utilization %
LUT	10	63400	0.02
IO	16	210	7.62

**Рисунок 1.8 — Отчет по использованию аппаратных ресурсов**

Ref Name	Used	Functional Category
IBUF	12	IO
LUT6	6	LUT
OBUF	4	IO
LUT4	4	LUT
LUT3	1	LUT
LUT2	1	LUT

**Рисунок 1.9 — Расширенный отчет по использованию аппаратных ресурсов**

Было задействовано 10 LUT, что подтверждает выдвинутые предположения. Количество использованных LUT растёт вместе с количеством входов.

Повысив количество разрядов входных переменных еще сильнее получаем код модуля представленный в Листинге 1.3.



*Листинг 1.3 – Реализация функции с беззнаковыми операндами разрядности 8*

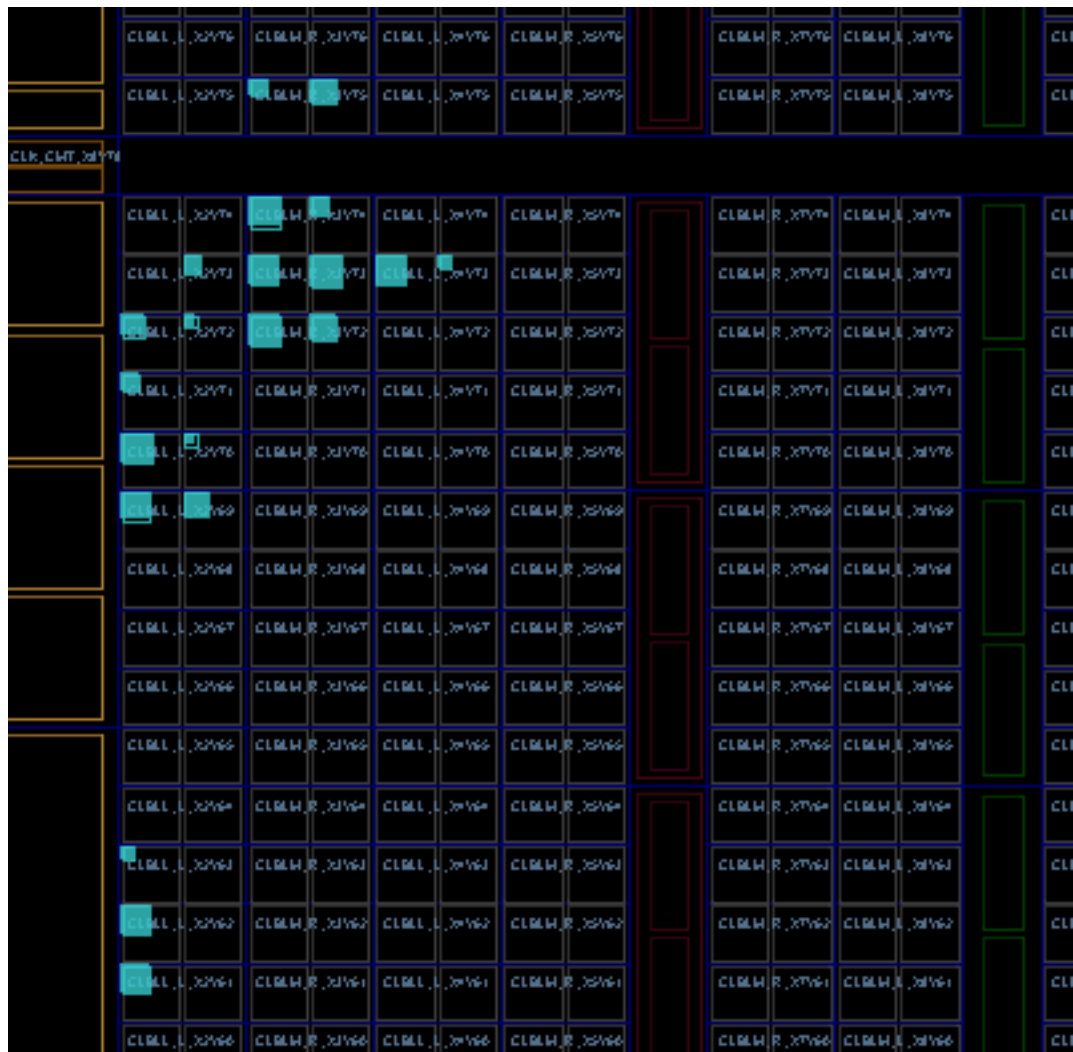
```
module main(
    input [7:0] a,
    input [7:0] b,
    input [7:0] c,
    input [7:0] d,
    output [8:0] res);

    assign res = a - b << c * d;

endmodule
```

Анализируя данный код, ожидается, высокое задействование входов и выходов, очередное увеличение ширины входных шин еще больше увеличивает количество задействованных LUT.

Произведены синтез и имплементация. Использованные аппаратные ресурсы представлены на Рисунках 1.7, 1.8 и 1.9.



**Рисунок 1.10 — Размещение модуля на устройстве**

Resource	Utilization	Available	Utilization %
LUT	57	63400	0.09
IO	41	210	19.52

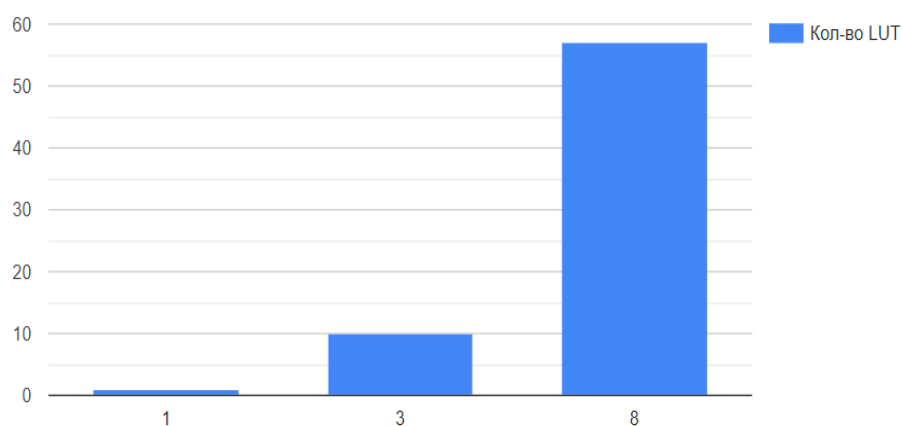
**Рисунок 1.11 — Отчет по использованию аппаратных ресурсов**

Ref Name	Used	Functional Category
IBUF	32	IO
LUT6	22	LUT
LUT2	22	LUT
LUT4	11	LUT
OBUF	9	IO
CARRY4	9	CarryLogic
LUT5	7	LUT
LUT3	5	LUT

**Рисунок 1.12 — Расширенный отчет по использованию аппаратных ресурсов**

Было задействовано 57 LUT (0,09%), 41 элемент IO, что соответствует выдвинутым ранее предположениям.

Также на Рисунке 1.13 представлена столбиковая диаграмма зависимости количества задействованных LUT от ширины входов нашего модуля для беззнаковой логики. По этой диаграмме легко сделать вывод о том, что для заданной операции количество задействованной логики для реализации стремительно растет с увеличением ширины входных шин.



**Рисунок 1.13 — Столбиковая диаграмма**

## 2 РЕАЛИЗАЦИЯ ФУНКЦИИ СО ЗНАКОВЫМИ ОПЕРАНДАМИ

### 2.1 Реализация функции со знаковыми операндами и анализ задействованных для этого аппаратных ресурсов

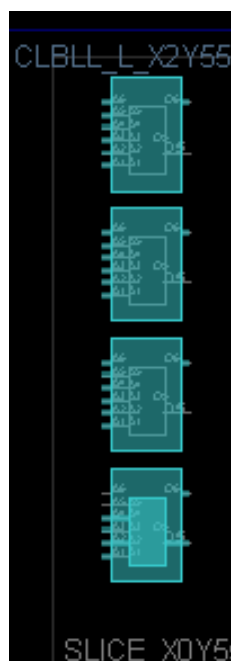
Код модуля верхнего уровня для реализации функции со знаковыми операндами разрядности 2 представлен в Листинге 2.1.

*Листинг 2.1 – Реализация функции со знаковыми операндами*

```
module main(  
    input signed [1:0] a,  
    input signed [1:0] b,  
    input signed [1:0] c,  
    input signed [1:0] d,  
    output signed [2:0] res);  
  
    assign res = a - b << c * d;  
  
endmodule
```

Анализируя данный код, не ожидается, высокого задействования аппаратных ресурсов, не смотря на наличие знаковых переменных.

Произведены синтез и имплементация. Использованные аппаратные ресурсы представлены на Рисунках 2.1, 2.2 и 2.3.



**Рисунок 2.1 — Размещение модуля на устройстве**

Resource	Utilization	Available	Utilization %
LUT	4	63400	0.01
IO	11	210	5.24

Рисунок 2.2 — Отчет по использованию аппаратных ресурсов

Ref Name	Used	Functional Category
IBUF	8	IO
OBUF	3	IO
LUT6	3	LUT
LUT4	1	LUT
LUT2	1	LUT

Рисунок 2.3 — Расширенный отчет по использованию аппаратных ресурсов

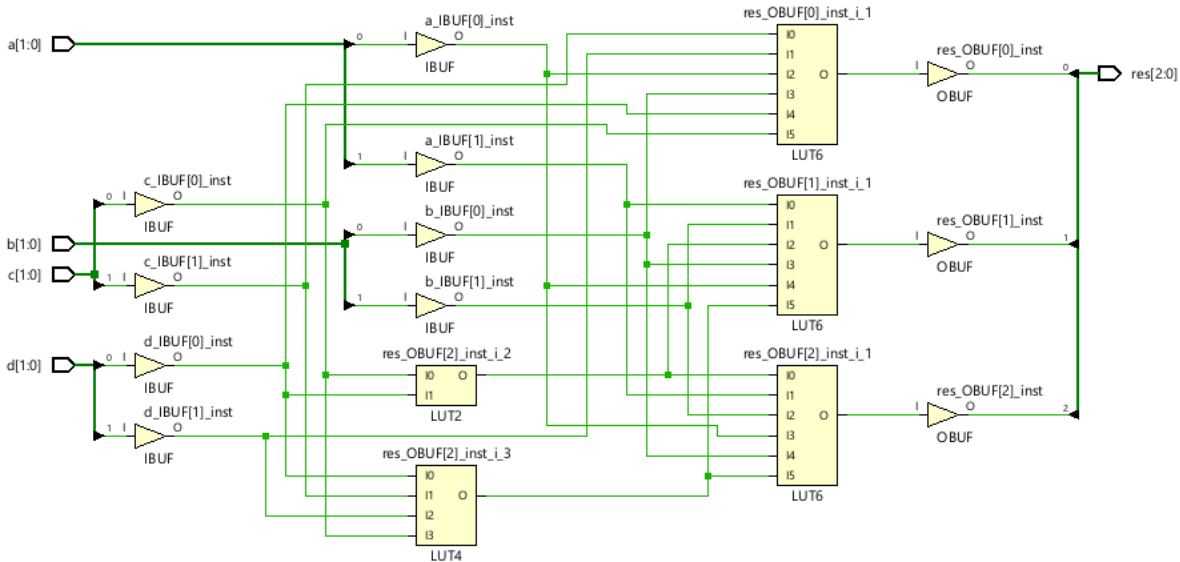


Рисунок 2.4 — Схема на этапе имплементации

Cell Properties										
res_OBUF[1]_inst_1										
I5	I4	I3	I2	I1	I0	O=I0 & I11 & I2 & I13 & I15 + I10 & I1 & I2 & I13 & I15 + I12 & I3 & I14 & I15 + I10 & I11 & I3 & I14 & I15 + I0 & I1 & I3 & I14 & I15 + I12 & I13 & I4 & I15 + I0 & I11 & I2 & I4 & I15 + I10 & I1 & I2 & I4 & I15				
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1	1	1	1
0	0	0	1	1	0	1	0	1	0	1
0	0	0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	0	1	0	1
0	0	1	0	0	1	1	1	1	1	1
0	0	1	0	1	0	1	0	1	0	1
0	0	1	0	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	0	1
0	0	1	1	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	0	1	0	1
0	1	0	0	0	1	1	1	1	1	1
0	1	0	0	1	0	1	0	1	0	1
0	1	0	0	1	1	1	1	1	1	1
0	1	0	1	0	0	0	0	0	0	0
0	1	0	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	0	1	0	1
0	1	0	1	1	1	1	1	1	1	1

Рисунок 2.5 — Таблица истинности

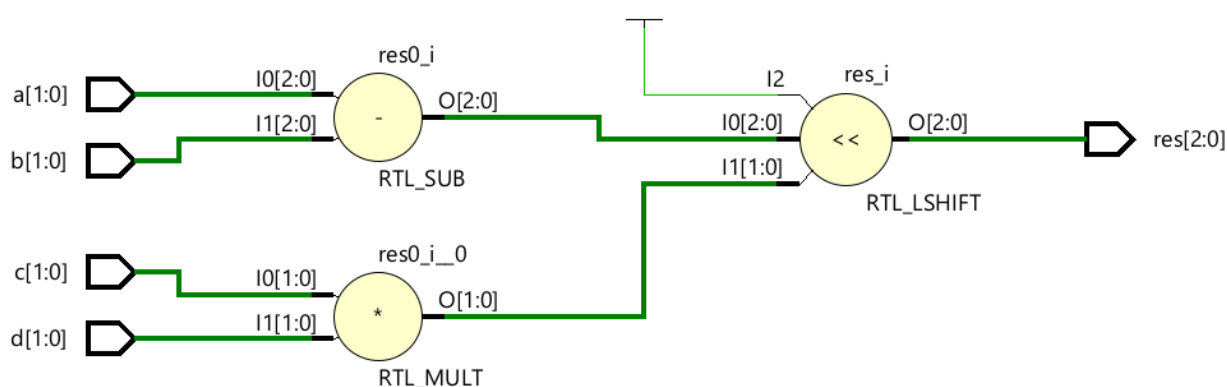


Рисунок 2.6 — RTL схема

## 2.2 Реализация функции со знаковыми операндами большей разрядности и анализ задействованных для этого аппаратных ресурсов

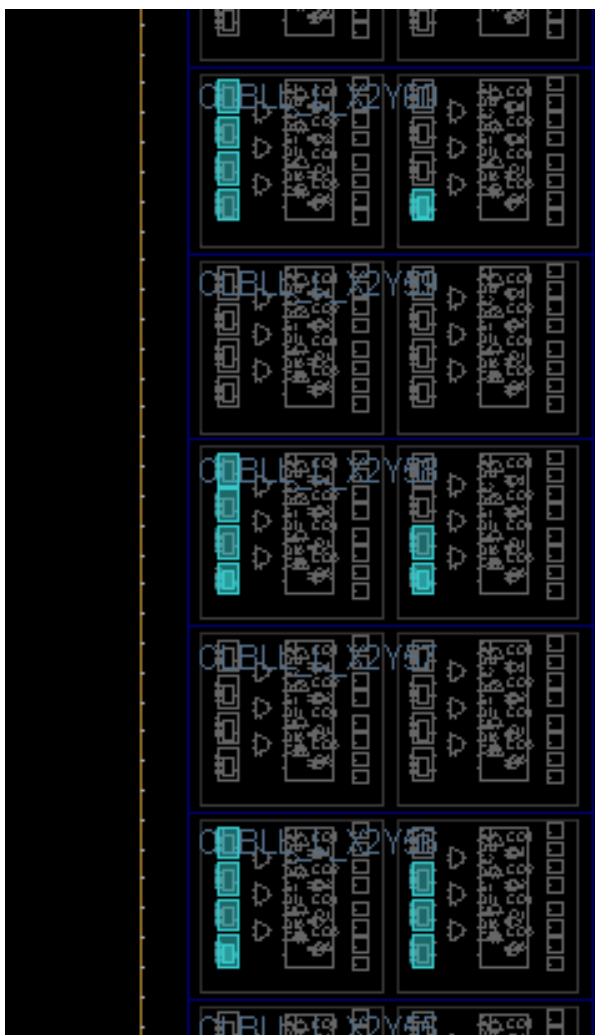
При увеличении разрядности входных переменных ожидается увеличение количества использованных LUT. В том числе ожидается использование большего количества аппаратных ресурсов, по сравнению с аналогичным без знакового типа, в связи затрат на дополнительный код.

Код модуля верхнего уровня для реализации функции со знаковыми операндами разрядности 4 представлен в Листинге 2.2.

*Листинг 2.2 – Реализация функции со знаковыми операндами разрядности 4*

```
module main(  
    input signed [3:0] a,  
    input signed [3:0] b,  
    input signed [3:0] c,  
    input signed [3:0] d,  
    output signed [4:0] res);  
  
    assign res = a - b << c * d;  
  
endmodule
```

Произведены синтез и имплементация. Использованные аппаратные ресурсы представлены на Рисунках 2.4, 2.5 и 2.6.



**Рисунок 2.7 — Размещение модуля на устройстве**

Resource	Utilization	Available	Utilization %
LUT	18	63400	0.03
IO	21	210	10.00

**Рисунок 2.8 — Отчет по использованию аппаратных ресурсов**

Ref Name	Used	Functional Category
IBUF	16	IO
LUT6	11	LUT
OBUF	5	IO
LUT4	5	LUT
LUT5	3	LUT
LUT2	3	LUT

**Рисунок 2.9 — Расширенный отчет по использованию аппаратных ресурсов**

Было задействовано 18 LUT, что сходится с выдвинутыми предположениями.

Повысив количество разрядов входных переменных еще сильнее получаем код модуля представленный в Листинге 2.3.

*Листинг 2.3 – Реализация функции со знаковыми операндами разрядности 9*

```

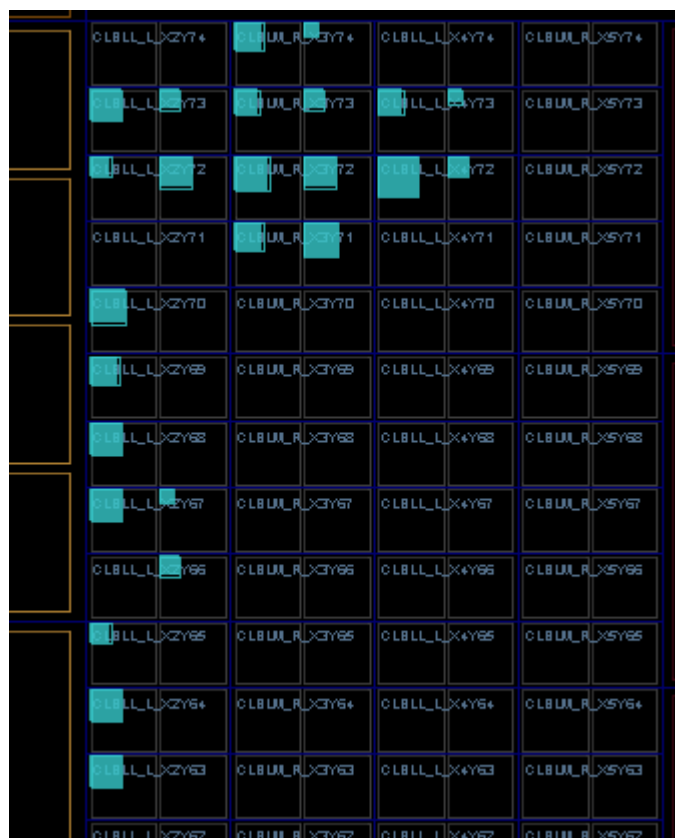
module main(
    input signed [8:0] a,
    input signed [8:0] b,
    input signed [8:0] c,
    input signed [8:0] d,
    output signed [9:0] res);

assign res = a - b << c * d;

endmodule

```

Произведены синтез и имплементация. Использованные аппаратные ресурсы представлены на Рисунках 2.7, 2.8 и 2.9.



**Рисунок 2.10 — Размещение модуля на устройстве**

Resource	Utilization	Available	Utilization %
LUT	70	63400	0.11
IO	46	210	21.90

**Рисунок 2.11 — Отчет по использованию аппаратных ресурсов**

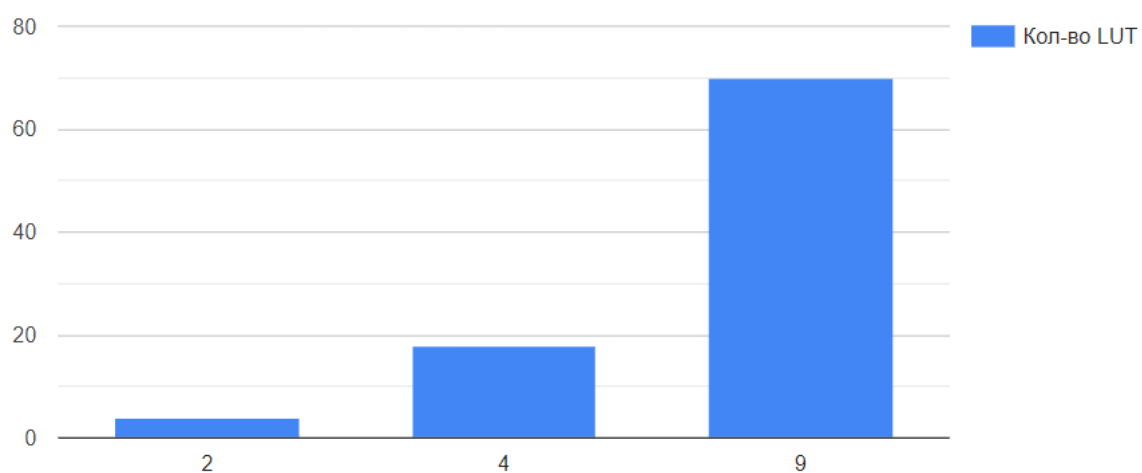
Ref Name	Used	Functional Category
IBUF	36	IO
LUT6	35	LUT
LUT2	26	LUT
LUT4	11	LUT
CARRY4	11	CarryLogic
OBUF	10	IO
LUT3	6	LUT
LUT5	5	LUT

**Рисунок 2.12 — Расширенный отчет по использованию аппаратных ресурсов**

На Рисунке 2.13 представлена столбиковая диаграмма для анализа используемых аппаратных ресурсов. Сравнив ее с первой диаграммой, можно сделать логичный вывод - знаковая логика задействует больше аппаратных



ресурсов, но в зависимости от сложности функции и её реализации, дополнительные аппаратные ресурсы могут не потребоваться.



**Рисунок 2.13 — Столбиковая диаграмма**

## **ЗАКЛЮЧЕНИЕ**

В ходе практической работы рассмотрены различные варианты реализации заданной схемы, проанализированы результаты синтеза и имплементации: при увеличении числа разрядов, количество затрачиваемых аппаратных ресурсов заметно увеличивается, знаковые операнды также влияют на количество затрачиваемых ресурсов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по ПР № 1 — URL: [https://online-edu.mirea.ru/pluginfile.php?file=%2F1225651%2Fmod\\_assign%2Fintroattachment%2F0%2F%D0%9F%D1%80%D0%B5%D0%B7%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F\\_%D0%9F%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%BA%D0%B0\\_1.pdf&forcedownload=1](https://online-edu.mirea.ru/pluginfile.php?file=%2F1225651%2Fmod_assign%2Fintroattachment%2F0%2F%D0%9F%D1%80%D0%B5%D0%B7%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F_%D0%9F%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_1.pdf&forcedownload=1) (Дата обращения: 27.02.2024).
2. Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. — М.: Горячая линия — Телеком, 2021. — 538 с.: ил.
3. Смирнов С.С. Информатика [Электронный ресурс]: Методические указания по выполнению практических и лабораторных работ / С.С. Смирнов — М., МИРЭА — Российский технологический университет, 2018.