Lab 2 Report

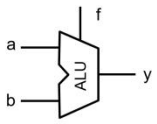
PB22020514 郭东昊

一、 实验目的与内容

实验内容

1. 设计 32 位 ALU 并优化，ALU 功能表如下

f	y
12'h001	a + b
12'h002	a - b
12'h004	$a <_s b$
12'h008	$a <_u b$
12'h010	a & b
12'h020	a | b
12'h040	~(a | b)
12'h080	a ^ b
12'h100	a << b[4:0]
12'h200	a >> b[4:0]
12'h400	a >>> b[4:0]
12'h800	b



- 比较两种不同实现方式的电路资源和性能

① 全部使用运算符

② 优化电路资源

- 利用加法实现减法
- 利用减法实现比较
- 利用右移实现左移

2. 设计 32 位单周期有符号数乘法器
3. 借助华莱士树和两位 Booth 编码，将单周期乘法器改造为两级流水线乘法器
4. 借助寄存器测试 ALU 和乘法器可接受的最大时钟频率

实验目的

1. 体验 ALU 与乘法器的设计
2. 通过对 ALU 和乘法器的优化熟悉常见的优化方法，领略它们的智慧
3. 熟练掌握查看生成电路及其性能和资源使用情况
4. 熟练掌握逻辑电路的模块化设计和 Verilog 描述方法
5. 熟练掌握利用 EDA 工具，进行逻辑电路的设计、仿真、调试、下载测试等基本方法

二、 逻辑设计

1. 运算器 ALU

示意图

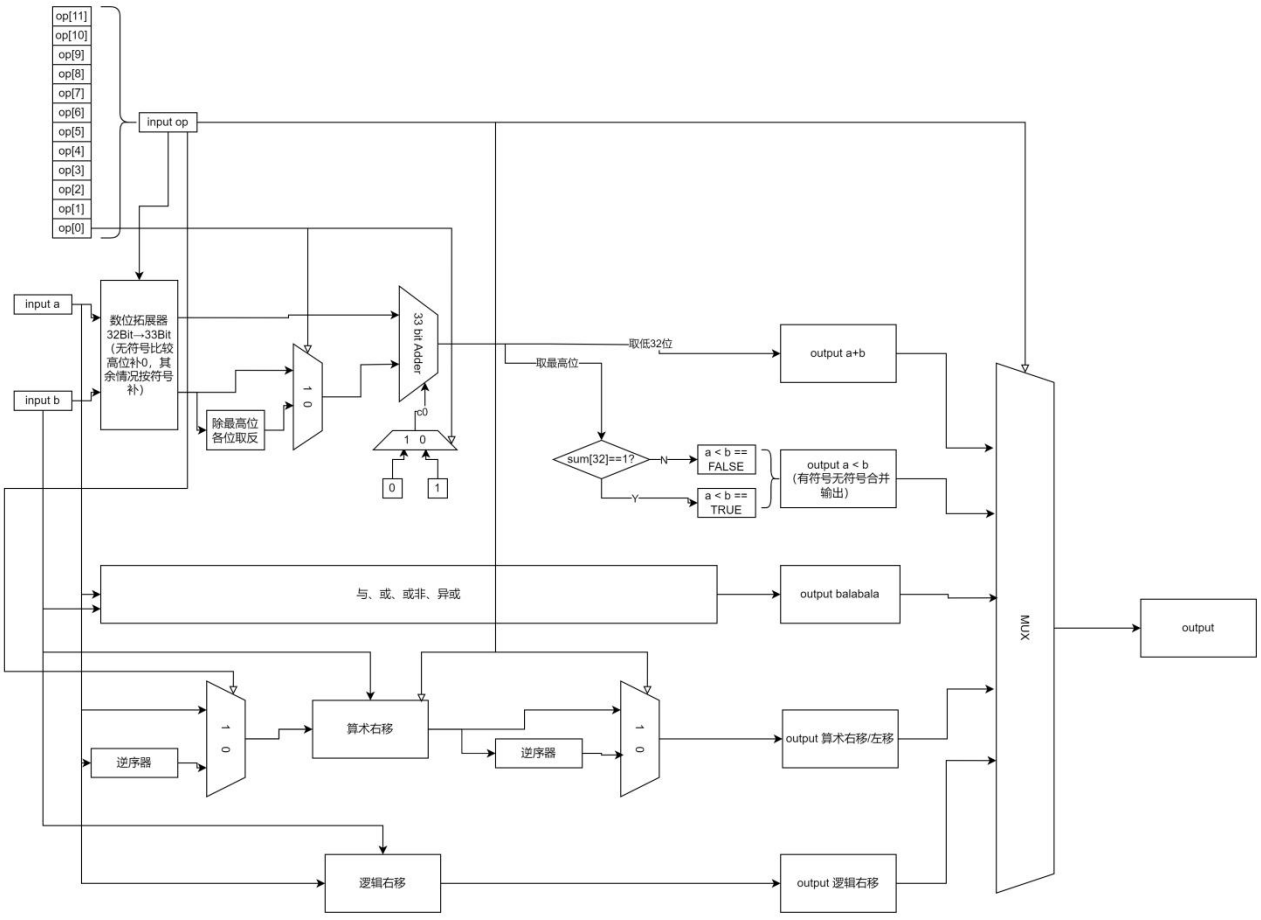


图 1 ALU 框图

上述设计实现了：

(1) 利用加法实现减法

(2) 利用减法实现比较

(3) 利用右移实现左移

提高了代码复用率，节省电路资源

2. 单周期有符号数乘法器

示意图

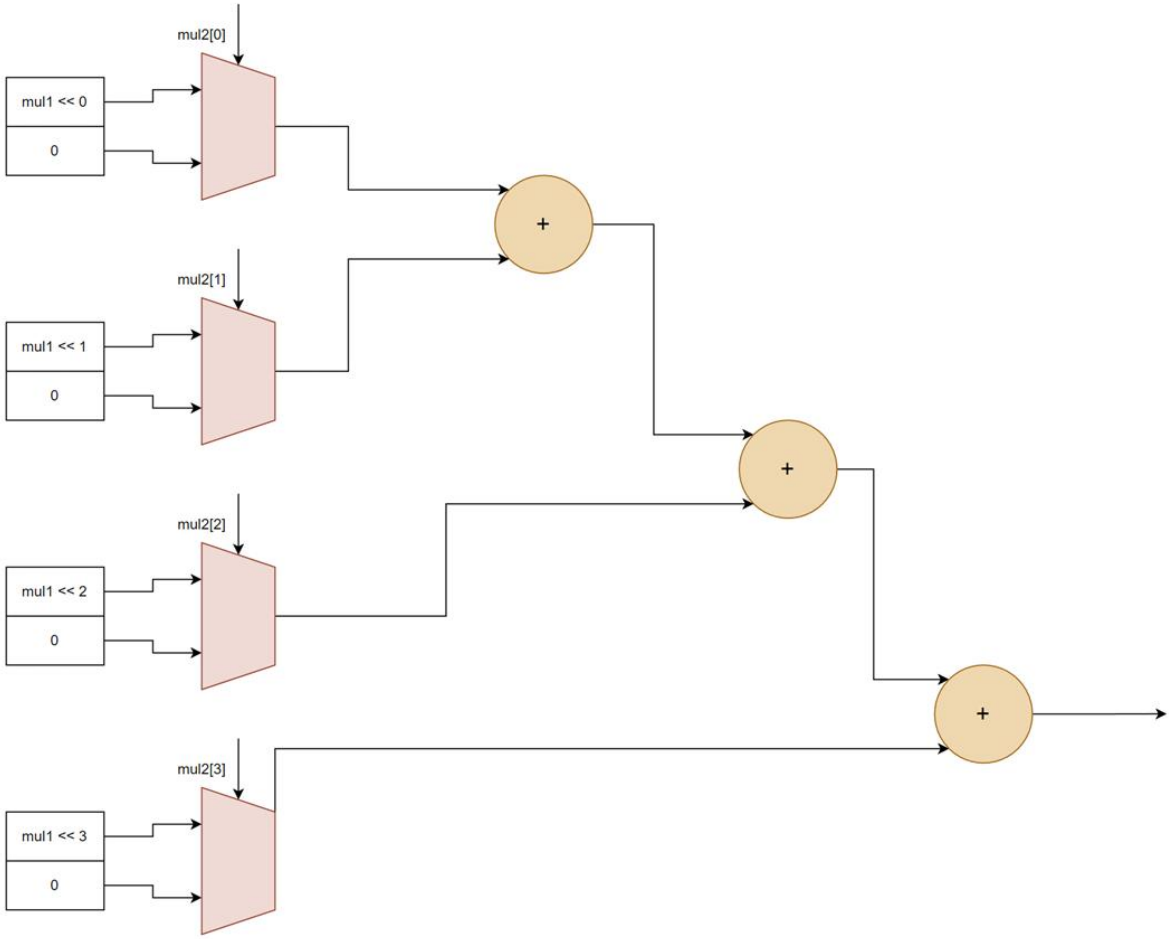


图 2 单周期乘法器

注：补码直接运算，部分积均为补码，最后一个部分积（最高位对应的部分积）需要被减去，在图中没有体现出来

三、 仿真结果与分析

1. 运算器 ALU

运行截图

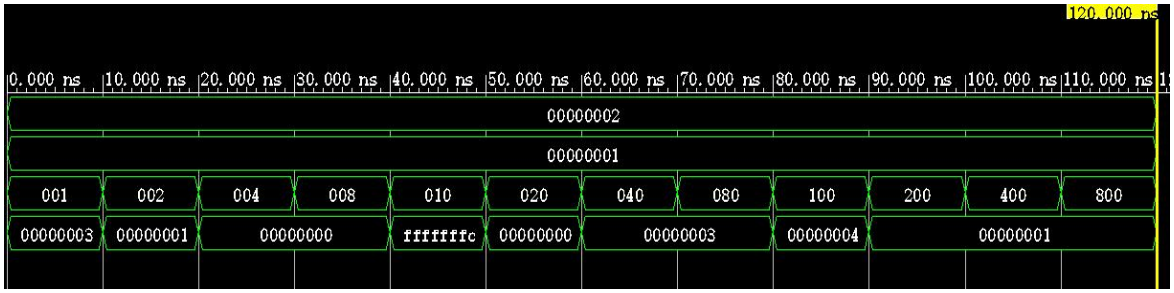


图 3 ALU 仿真运行截图

输入数据 a = 2, b = 1

op 遍历所有操作，输出结果正常

2. 单周期有符号数乘法器

运行截图

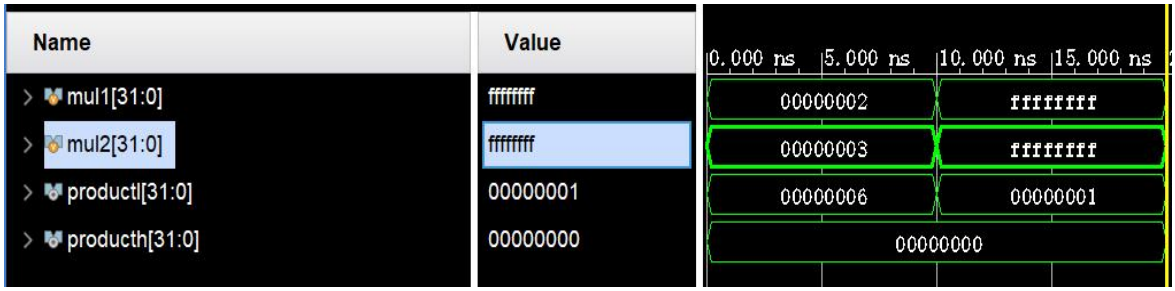


图 4 单周期有符号数乘法器运行截图

四、 电路设计与分析

1. 运算器 ALU

RTL 电路图

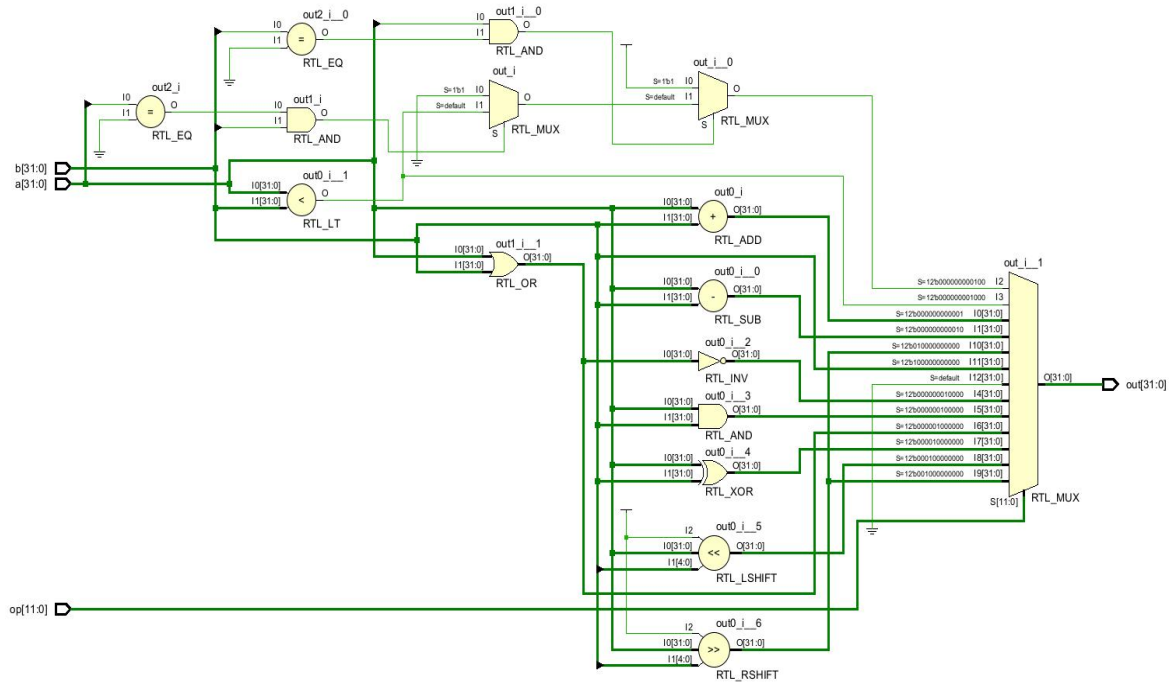


图 5 ALU 电路图

资源使用情况

LUT	FF
367	0

图 6 ALU 资源使用情况

WNS
3.926

图 7 ALU_top 的 WNS 非负

2. 单周期乘法器

RTL 电路图

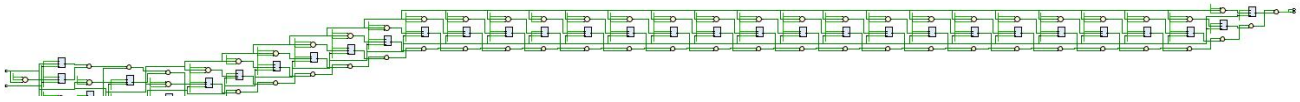


图 8 乘法器电路图

资源使用情况

LUT	FF
1361	0

图 9 乘法器资源使用情况

WNS	TNS
6.606	0.000

图 10 乘法器 WNS 非负

五、 测试结果与分析

1. 运算器 ALU

12 种运算太多了，验收已通过，这里仅展示 1 + 1 = 2

图 11 src1 = 1

图 12 src2 = 2

图 13 op = 1

图 14 out = 2

2. 单周期乘法器

(1) -1 * (-1) = 1

图 15 a = -1

图 16 b = -1

图 17 out = 1

(2) -1 * 0 = 0

图 18 a = 0

图 19 b = -1

图 20 out = 0

(3) 3 * 3 = 9

图 21 a = 3

图 22 b = 3

图 23 out = 9

六、 总结

通过本次实验，我学习了有符号数乘法的基本原理，设计了 32 位 ALU 并设计了优化方案，设计了 32 位单周期有符号数乘法器，实现了利用 ulu + top.sv 的上板操作，学习了性能测试的一些技巧。

由于各种事务缠身，未能将 ALU 的优化方案最终实现，也只是粗浅理解了 Booth 编码的原理和华莱士树的巧妙构思，未能设计 Booth 编码器以及为 32 位有符号数乘法器设计华莱士树，我感到十分遗憾，希望今后有空时能将它们一一实现。

此外，对 top 文件，ulu 文件的理解还是不够深入，只停留在会应用的层次；对仿真文件的书写还是不够得心应手。我应该继续学习 Verilog 语法，更深入掌握硬件设计的技巧。