



哈爾濱工業大學 (深圳)  
HARBIN INSTITUTE OF TECHNOLOGY

# 实验作业

开课学期: 2022 春季

课程名称: 计算机组成原理 (实验)

实验名称: Booth 乘法器设计

实验性质: 综合设计型

实验学时: 4 地点: T2506

学生班级: 12

学生学号: 200111205

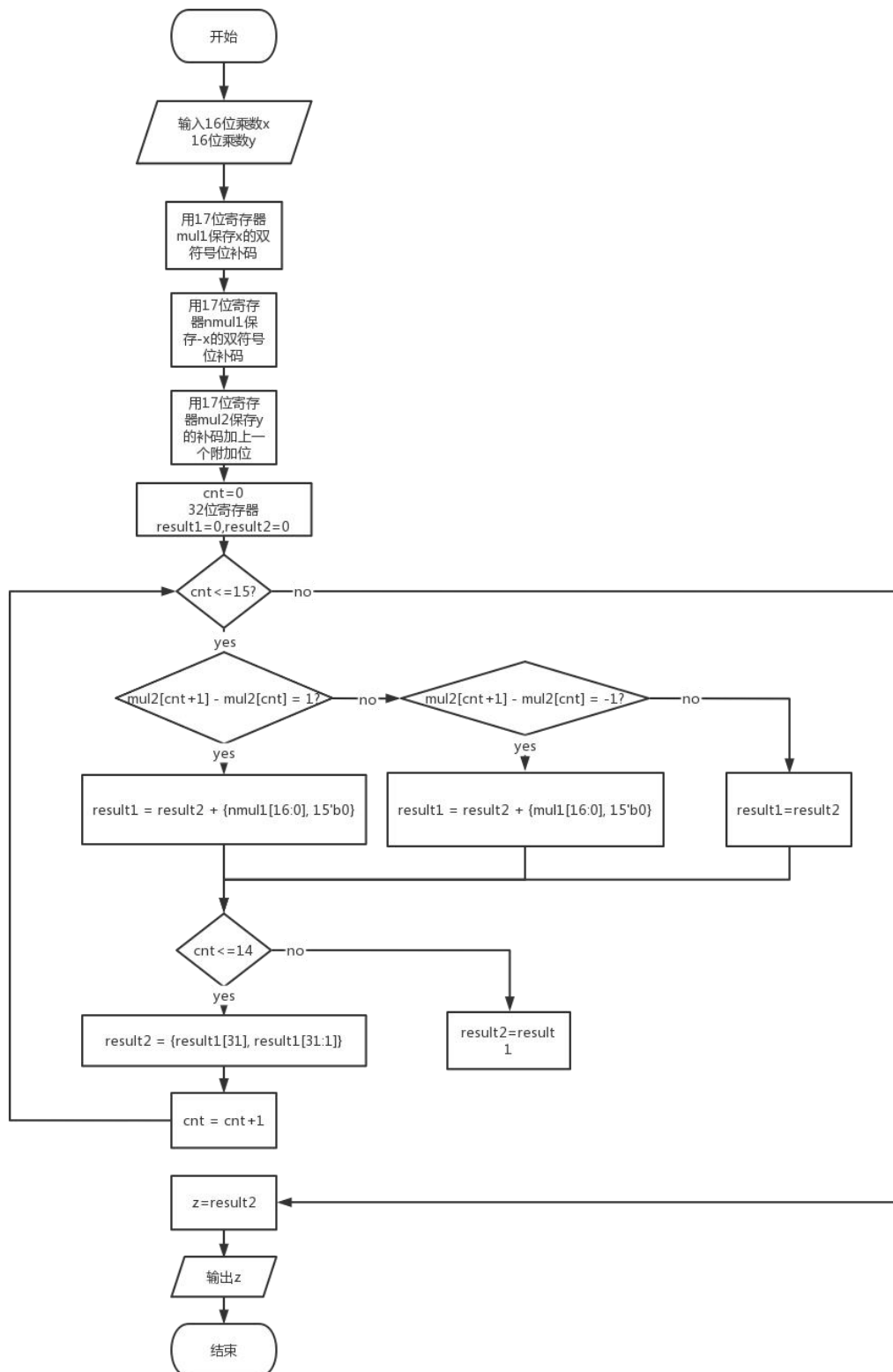
学生姓名: 李 聪

作业成绩: \_\_\_\_\_

实验与创新实践教育中心制

2022 年 4 月

## 1、Booth 乘法器算法流程图

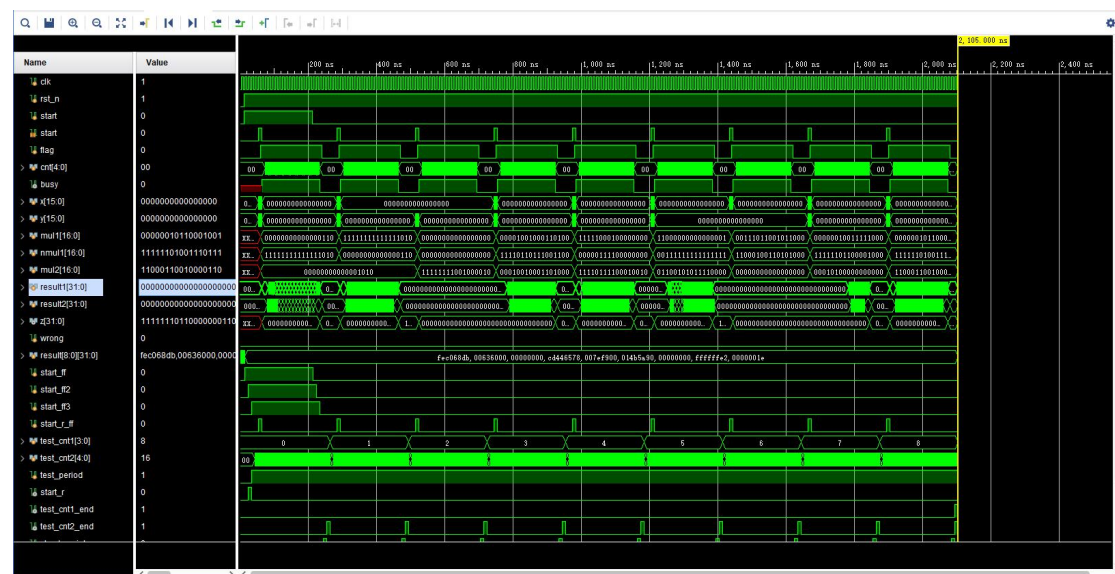


## 2、调试报告

（仿真截图及时序分析，要求分析最少 3 次乘法运算）

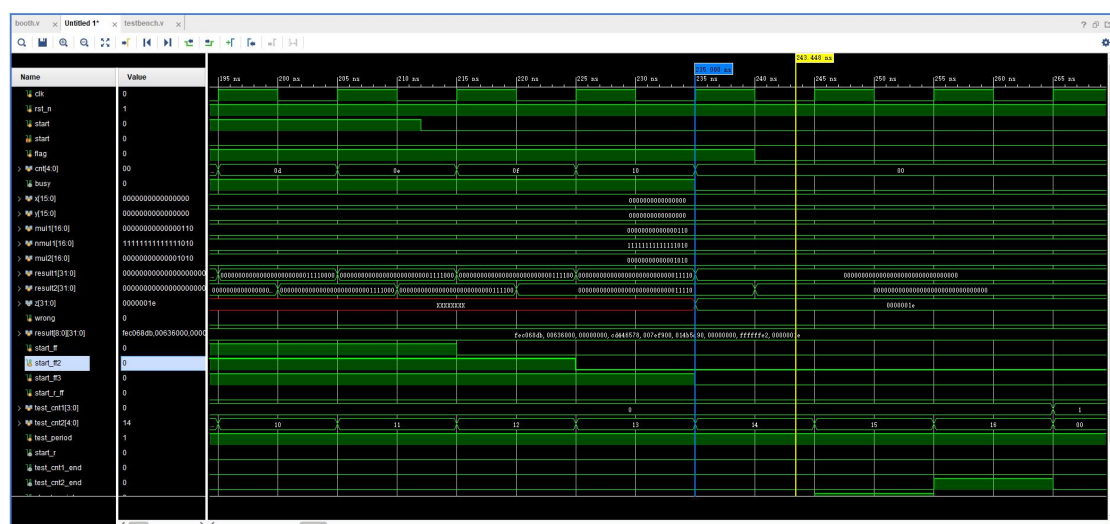
波形整体截图：

图中变量说明：**flag** 为 1 位寄存器来指示 **start** 信号是否拉高；**result1** 为 32 位寄存器，保存每次进行加法后的结果；**result2** 为 32 位寄存器，保存每次进行移位后的结果；**cnt** 为 5 位计数寄存器；**mul1** 为 17 位寄存器保存 **x** 的双符号位补码；**nmul1** 为 17 位寄存器保存 **x** 的相反数的双符号位补码；**mul2** 为 17 位寄存器保存 **y** 的补码加上附加位。



时序分析：

5\*6 波形分析



55ns 时，start 信号拉高，x、y 信号有效，x=0000000000000110，为 6 的补码，y=0000000000000101 为 5 的补码。

60ns 时，来到 flag 的上升沿，此时用一个 17 位的寄存器 mul1、nmul1、mul2 分别保存 x 的双符号位补码 0000000000000110、x 的相反数的双符号位补码 1111111111111010、y 的补码加上附加位 00000000000001010。

65ns 时，到达时钟上升沿，即 start 拉高一个周后 busy 信号拉高。进行第一次加法运算，由于 mul2[cnt+1,cnt]=10,即  $y_{i+1} = 10$ ，故在 result2 的高 17 位加上 nmul1 即  $[-x]_{补}$ ，结果保存在 result1，result1 变为 1111 1111 1111 1101 0000 0000 0000 0000。

80ns 时，到达时钟下降沿，进行第一次移位运算，将 result1 中的值进行一次算术右移，

结果保存在 result2, result2 变为 1111 1111 1111 1110 1000 0000 0000 0000

85ns, 到达时钟上升沿, 进行第二次加法运算, 由于  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 01$ , 即  $y_i y_{i+1} = 01$ , 故在 result2 的高 17 位加上 mul1 即  $[x]_{\text{补}}$ , 结果保存在 result1, result1 变为 0000 0000 0000 0001 1000 0000 0000 0000。

90ns, 到达时钟下降沿, 进行第二次移位运算, 将 result1 中的值进行一次算术右移, 结果保存在 result2, result2 变为 0000 0000 0000 0000 1100 0000 0000 0000。

95ns, 到达时钟上升沿, 进行第三次加法运算, 由于  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 10$ , 即  $y_i y_{i+1} = 10$ , 故在 result2 的高 17 位加上 mul1 即  $[-x]_{\text{补}}$ , 结果保存在 result1 中。

100ns, 到达时钟下降沿, 进行第三次移位运算, 将 result1 中的值进行一次算术右移, 结果保存在 result2。

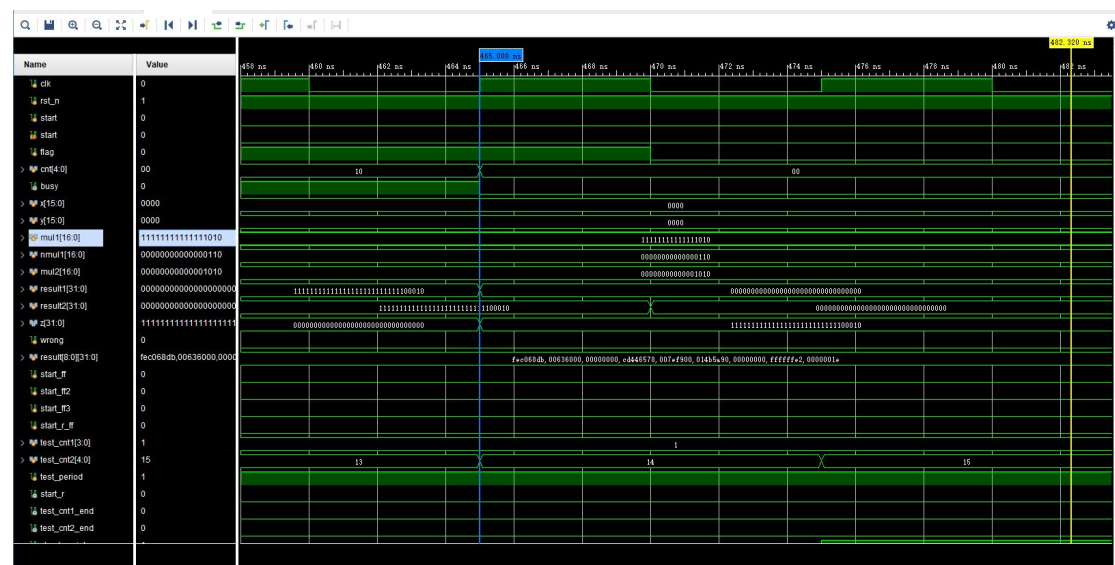
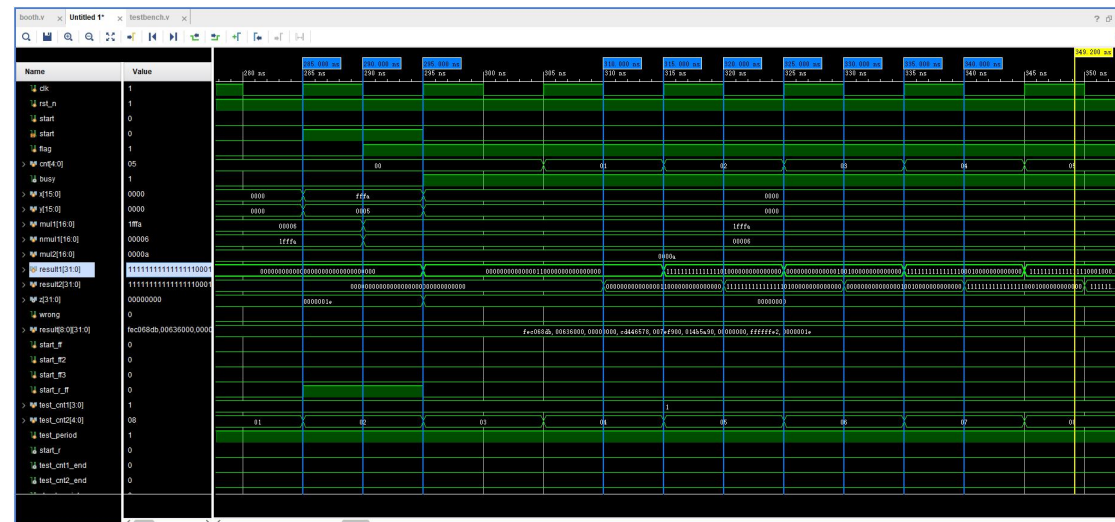
105ns, 到达时钟上升沿, 进行第四次加法运算, 由于  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 01$ , 即  $y_i y_{i+1} = 01$ , 故在 result2 的高 17 位加上 mul1 即  $[x]_{\text{补}}$ , 结果保存在 result1 中。

110ns, 到达时钟下降沿, 进行第四次移位运算, 将 result1 中的值进行一次算术右移, 结果保存在 result2。

之后由于所有的  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 00$ , 即  $y_i y_{i+1} = 00$ , 只将结果进行算术右移, 再经过 12 次算术右移得到最终结果

235ns 时, z 的值变为 0x0000001e, 即 30, 乘法器工作完成时拉低 busy, 符合预期结果。

## -6\*5 波形分析:



285ns 时, start 信号拉高, x,y 信号有效, x=1111111111111010 为-6 的补码, y=00000000000000110 为 5 的补码。

290ns 时, 来到 flag 的上升沿, 此时用一个 17 位的寄存器 mul1、nmul1、mul2 分别保存 x 的双符号位补码 1111111111111010、x 的相反数的双符号位补码 00000000000000110、y 的补码加上附加位 000000000000001010。

295ns 时, 到达时钟上升沿, 即 start 拉高一个周后 busy 信号拉高。进行第一次加法运算,

由于  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 10$ , 即  $y_i y_{i+1} = 10$ , 故在  $\text{result2}$  的高 17 位加上  $\text{nmul1}$  即  $[-x]_{\text{补}}$ , 结果保存在  $\text{result1}$ 。

310ns 时, 到达时钟下降沿, 进行第一次移位运算, 将  $\text{result1}$  中的值进行一次算术右移, 结果保存在  $\text{result2}$ 。

315ns, 到达时钟上升沿, 进行第二次加法运算, 由于  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 01$ , 即  $y_i y_{i+1} = 01$ , 故在  $\text{result2}$  的高 17 位加上  $\text{mul1}$  即  $[x]_{\text{补}}$ , 结果保存在  $\text{result1}$ 。

320ns, 到达时钟下降沿, 进行第二次移位运算, 将  $\text{result1}$  中的值进行一次算术右移, 结果保存在  $\text{result2}$ 。

325ns, 到达时钟上升沿, 进行第三次加法运算, 由于  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 10$ , 即  $y_i y_{i+1} = 10$ , 故在  $\text{result2}$  的高 17 位加上  $\text{mul1}$  即  $[-x]_{\text{补}}$ , 结果保存在  $\text{result1}$  中。

330ns, 到达时钟下降沿, 进行第三次移位运算, 将  $\text{result1}$  中的值进行一次算术右移, 结果保存在  $\text{result2}$ 。

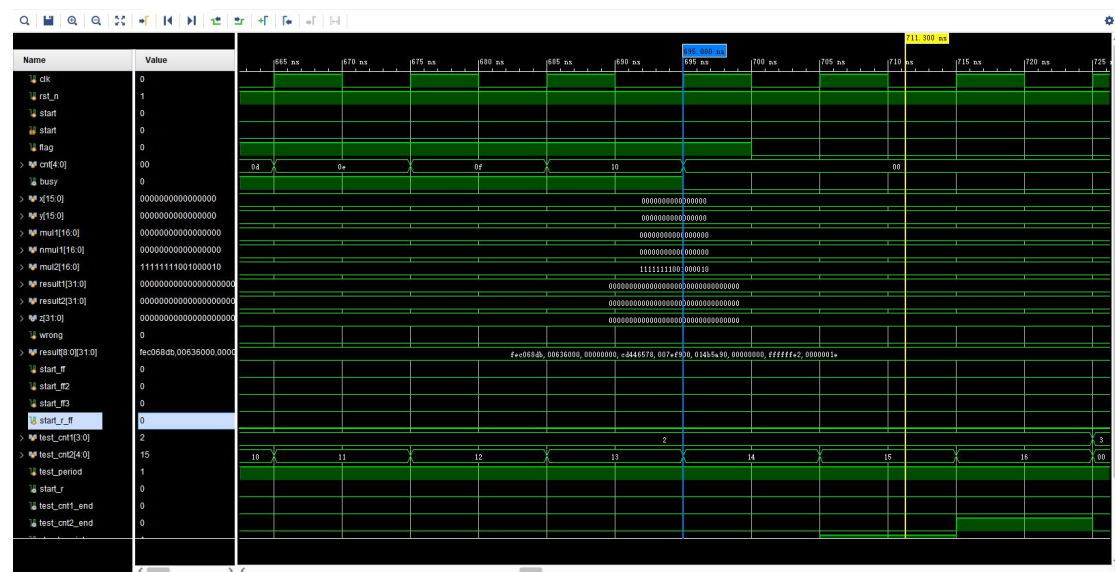
335ns, 到达时钟上升沿, 进行第四次加法运算, 由于  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 01$ , 即  $y_i y_{i+1} = 01$ , 故在  $\text{result2}$  的高 17 位加上  $\text{mul1}$  即  $[x]_{\text{补}}$ , 结果保存在  $\text{result1}$  中。

340ns, 到达时钟下降沿, 进行第四次移位运算, 将  $\text{result1}$  中的值进行一次算术右移, 结果保存在  $\text{result2}$ 。

之后由于所有的  $\text{mul2}[\text{cnt}+1, \text{cnt}] = 00$ , 即  $y_i y_{i+1} = 00$ , 只将结果进行算术右移, 再经过 12 次算术右移得到最终结果

465ns 时,  $z$  的值变为  $0\text{xfffffe}2$ , 即 -30 的补码形式, 乘法器工作完成时拉低  $\text{busy}$ , 符合预期结果。

0\* (-223)



515ns 时，start 信号拉高，x,y 信号有效，x=0000000000000000 为-6 的补码，y=1111111100100001 为-233 的补码。

520ns 时，来到 flag 的上升沿，此时用一个 17 位的寄存器 mul1、nmul1、mul2 分别保存 x 的双符号位补码 00000000000000000、x 的相反数的双符号位补码



0000000000000000、y 的补码加上附加位 1111111001000010.

525ns 时，到达时钟上升沿，即 **start** 拉高一个周后 **busy** 信号拉高。进行第一次加法运算。由于被乘数为 0，故每次的加法以及以后的结果都是 0，因此该计算过程中 **result1,result2** 始终保持为 0

695ns 时，乘法器工作完成时拉低 **busy**，z 的值为 0，符合预期结果。