

中国科学技术大学计算机学院

《计算机组成原理实验》报告



实验题目： 运算器及其应用

学生姓名： 黄瑞轩

学生学号： PB20111686

完成日期： 2022. 3. 18

计算机实验教学中心制
2020 年 09 月

实验题目

运算器及其应用

实验目的

- 熟练掌握算术逻辑单元 (ALU) 的功能
- 掌握数据通路和控制器的设计方法
- 掌握组合电路和时序电路，以及参数化和结构化的 Verilog 描述方法
- 了解查看电路性能和资源使用情况

实验环境

- Nexys4-DDR
- Vivado 2019.1

实验 1: 算术逻辑单元 (ALU)

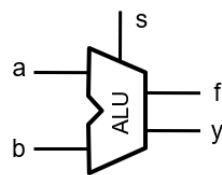
● ALU 器件的设计

➔ 要求实现的功能

32 位数据的运算，具体功能为：判断是否相等、不等、有符号数小于、有符号数大于等于、无符号数小于、无符号数大于等于，以及双目减法、加法、与运算、或运算、异或运算、逻辑和算术右移运算、左移运算。

➔ ALU 逻辑设计

器件图和端口含义：



s: 功能选择，加、减、与、或、异或、逻辑左移、逻辑右移、算术右移等运算
a, b: 两个操作数
y: 运算结果，和、差
f: 标志，相等(eq)，小于(lt, ltu)

- f[0]: 相等(eq)
- f[1]: 有符号数小于(lt)
- f[2]: 无符号数小于(ltu)
- * 表示根据运算结果设置
- x 表示与比较结果无关

| 大小关系 | f | | |
|--------------------|-----|----|----|
| | ltu | lt | eq |
| a = b | 0 | 0 | 1 |
| a ≠ b | x | x | 0 |
| a < _s b | x | 1 | 0 |
| a ≥ _s b | x | 0 | x |
| a < _u b | 1 | x | 0 |
| a ≥ _u b | 0 | x | x |

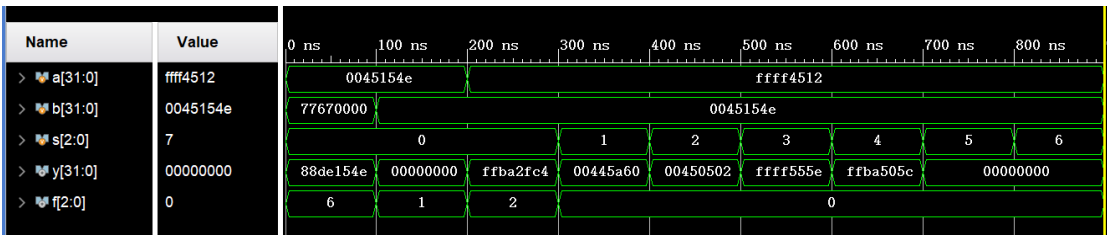
| s | y | f |
|---|---------|---|
| 0 | a - b | * |
| 1 | a + b | 0 |
| 2 | a & b | 0 |
| 3 | a b | 0 |
| 4 | a ^ b | 0 |
| 5 | a >> b | 0 |
| 6 | a << b | 0 |
| 7 | a >>> b | 0 |

➔ ALU 核心代码

见附件 alu.v。

➔ ALU 模块的仿真

为了验证 ALU 模块的正确性，编写 testbench 文件，代码见附件 testbench.v，模拟结果见下图，经检验结果正确。

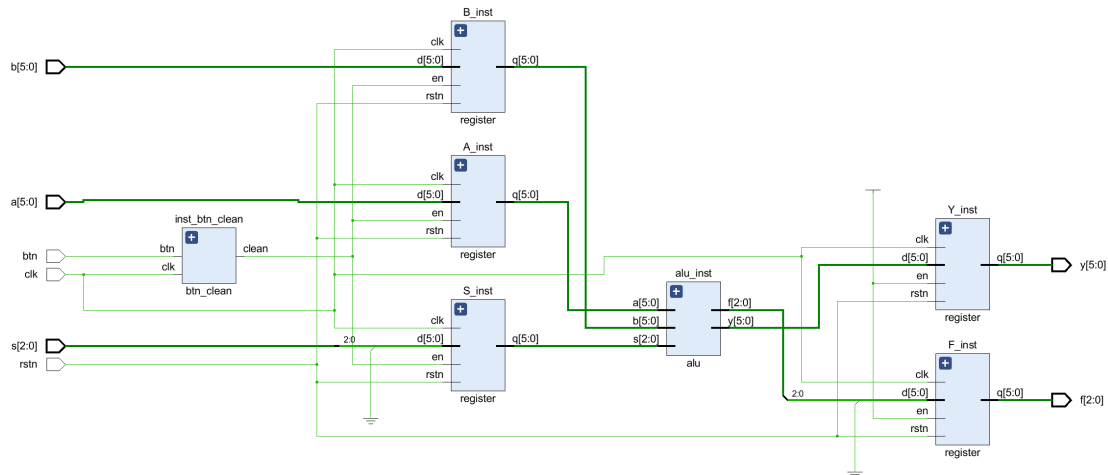


● 测试用 Top 模块编写

➔ Top 核心代码

对实验要求的 6 位 ALU 进行测试，代码见附件 top.v。对 Button 按钮去抖动、取边沿的模块见附件 btn_clean.v；寄存器代码见附件 register.v；二者不是本实验重点，不做详述。

➔ Top RTL 分析电路



➔ Top 模块与 Nexys4 开发板的接口对应

xdc 约束文件见附件 const.xdc。

| | |
|--------------------|-------------------|
| s[2:0] → sw[15:13] | rstn → cpu_resetn |
| a[5:0] → sw[11:6] | clk → clk100mhz |
| b[5:0] → sw[5:0] | f → led[15:13] |
| en → btnc | y → led[5:0] |

➔ Top 上板下载测试

生成比特流并烧写到开发板上，用如下几组测试数据检验正确性：

```
0# s = 3'b 000; a = 6'b 101001; b = 6'b 001000; // 有符号数正负不同
1# s = 3'b 000; a = 6'b 101001; b = 6'b 101001; // 相等判断
2# s = 3'b 000; a = 6'b 101001; b = 6'b 111100; // 有无符号情况不同
3# s = 3'b 001; a = 6'b 001001; b = 6'b 001010; // 加法运算
4# s = 3'b 010; a = 6'b 101001; b = 6'b 001000; // 按位与
5# s = 3'b 011; a = 6'b 101001; b = 6'b 001110; // 按位或
6# s = 3'b 100; a = 6'b 101001; b = 6'b 001000; // 按位异或
7# s = 3'b 101; a = 6'b 101001; b = 6'b 000100; // 逻辑右移
8# s = 3'b 110; a = 6'b 101001; b = 6'b 000100; // 左移
9# s = 3'b 111; a = 6'b 101001; b = 6'b 000100; // 算数右移
```

预期结果：

```
0# f[0:2] = 3'b 010; y = 6'b 100001;
1# f[0:2] = 3'b 001; y = 6'b 000000;
2# f[0:2] = 3'b 100; y = 6'b 101101;
3# f[0:2] = 3'b 000; y = 6'b 010011;
4# f[0:2] = 3'b 000; y = 6'b 001000;
5# f[0:2] = 3'b 000; y = 6'b 101111;
6# f[0:2] = 3'b 000; y = 6'b 100001;
7# f[0:2] = 3'b 000; y = 6'b 000010;
8# f[0:2] = 3'b 000; y = 6'b 010000;
9# f[0:2] = 3'b 000; y = 6'b 111110;
```

实验测试结果：



0# 正确



1# 正确



2# 正确



3# 正确



4# 正确



5# 正确



6# 正确



7# 正确



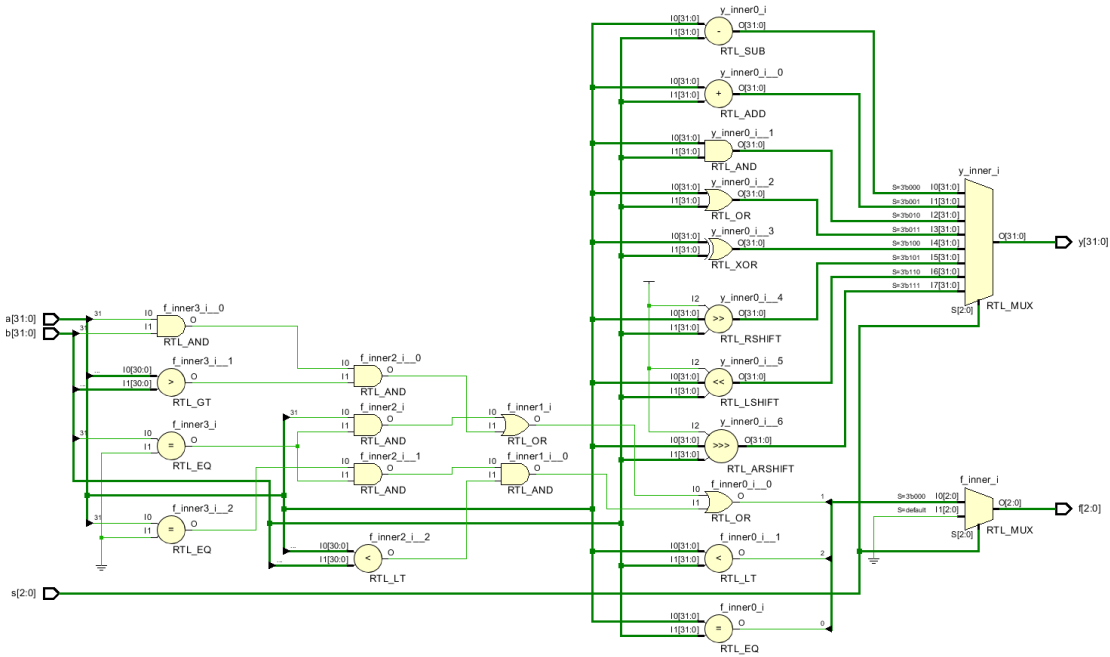
8# 正确



9# 正确

● ALU 模块电路及资源与性能

➔ ALU RTL 分析电路



➔ ALU 电路资源

ALU 单独模块电路资源

| Tcl Console | Messages | Log | Reports | Design Runs | Utilization |
|--------------------|----------|-----|---------|-------------|-------------|
| Hierarchy | | | | | |
| Hierarchy | | | | | |
| Summary | | | | | |
| Slice Logic | | | | | |
| Name | | | | | |
| Slice LUTs (63400) | | | | | |
| Bonded IOB (210) | | | | | |
| N alu | | | | | |
| 447 | | | | | |
| 102 | | | | | |

Top 模块电路资源

| Tcl Console | Messages | Log | Reports | Design Runs | Utilization |
|----------------------------|----------|-----|---------|-------------|-------------|
| Hierarchy | | | | | |
| Hierarchy | | | | | |
| Summary | | | | | |
| Slice Logic | | | | | |
| Name | | | | | |
| Slice LUTs (63400) | | | | | |
| Slice Registers (126800) | | | | | |
| F7 Muxes (31700) | | | | | |
| Bonded IOB (210) | | | | | |
| BUFGCTRL (32) | | | | | |
| N top | | | | | |
| 61 | | | | | |
| 30 | | | | | |
| 1 | | | | | |
| 27 | | | | | |
| 1 | | | | | |
| A_inst (register) | | | | | |
| 25 | | | | | |
| 6 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| B_inst (register_0) | | | | | |
| 25 | | | | | |
| 6 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| F_inst (register_1) | | | | | |
| 0 | | | | | |
| 3 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| Inst_btn_clean (btn_clean) | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| S_inst (register_2) | | | | | |
| 6 | | | | | |
| 3 | | | | | |
| 1 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| Y_inst (register_3) | | | | | |
| 1 | | | | | |
| 6 | | | | | |
| 0 | | | | | |
| 0 | | | | | |
| 0 | | | | | |

➔ ALU 电路性能

| Timing | Name | Slack | Levels | Routes | High Fanout | From | To | Total Delay | Logic Delay | Net Delay | Requirement | Source Clock | Destination Clock |
|---|------|-------|--------|--------|-------------|------|----|-------------|-------------|-----------|-------------|--------------|-------------------|
| Intra-Clock Paths - sys_clk_pin - Setup | | | | | | | | | | | | | |
| Path 1 | | | | | | | | | | | | | |
| 5.935 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| A_inst/q_reg[3]/C | | | | | | | | | | | | | |
| Y_inst/q_reg[0]/D | | | | | | | | | | | | | |
| 3.929 | | | | | | | | | | | | | |
| 1.145 | | | | | | | | | | | | | |
| 2.784 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 2 | | | | | | | | | | | | | |
| 5.935 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| A_inst/q_reg[4]/C | | | | | | | | | | | | | |
| Y_inst/q_reg[1]/D | | | | | | | | | | | | | |
| 3.929 | | | | | | | | | | | | | |
| 1.145 | | | | | | | | | | | | | |
| 2.784 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 3 | | | | | | | | | | | | | |
| 6.070 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| A_inst/q_reg[4]/C | | | | | | | | | | | | | |
| Y_inst/q_reg[3]/D | | | | | | | | | | | | | |
| 3.794 | | | | | | | | | | | | | |
| 1.145 | | | | | | | | | | | | | |
| 2.649 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 4 | | | | | | | | | | | | | |
| 6.224 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | |
| B_inst/q_reg[1]/C | | | | | | | | | | | | | |
| Y_inst/q_reg[2]/D | | | | | | | | | | | | | |
| 3.640 | | | | | | | | | | | | | |
| 1.169 | | | | | | | | | | | | | |
| 2.471 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 5 | | | | | | | | | | | | | |
| 6.234 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | |
| B_inst/q_reg[5]/C | | | | | | | | | | | | | |
| Y_inst/q_reg[4]/D | | | | | | | | | | | | | |
| 3.630 | | | | | | | | | | | | | |
| 1.169 | | | | | | | | | | | | | |
| 2.461 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 6 | | | | | | | | | | | | | |
| 6.383 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | |
| B_inst/q_reg[5]/C | | | | | | | | | | | | | |
| Y_inst/q_reg[5]/D | | | | | | | | | | | | | |
| 3.481 | | | | | | | | | | | | | |
| 1.466 | | | | | | | | | | | | | |
| 2.015 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 7 | | | | | | | | | | | | | |
| 6.907 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | |
| B_inst/q_reg[0]/C | | | | | | | | | | | | | |
| F_inst/q_reg[1]/D | | | | | | | | | | | | | |
| 2.957 | | | | | | | | | | | | | |
| 1.021 | | | | | | | | | | | | | |
| 1.936 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 8 | | | | | | | | | | | | | |
| 6.907 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | |
| B_inst/q_reg[0]/C | | | | | | | | | | | | | |
| F_inst/q_reg[2]/D | | | | | | | | | | | | | |
| 2.957 | | | | | | | | | | | | | |
| 1.021 | | | | | | | | | | | | | |
| 1.936 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 9 | | | | | | | | | | | | | |
| 7.499 | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| A_inst/q_reg[3]/C | | | | | | | | | | | | | |
| F_inst/q_reg[0]/D | | | | | | | | | | | | | |
| 2.365 | | | | | | | | | | | | | |
| 0.897 | | | | | | | | | | | | | |
| 1.468 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Path 10 | | | | | | | | | | | | | |
| 7.838 | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | |
| Inst_btn_clean/reg/C | | | | | | | | | | | | | |
| A_inst/q_reg[0]/CE | | | | | | | | | | | | | |
| 1.780 | | | | | | | | | | | | | |
| 0.773 | | | | | | | | | | | | | |
| 1.007 | | | | | | | | | | | | | |
| 10.0 | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |

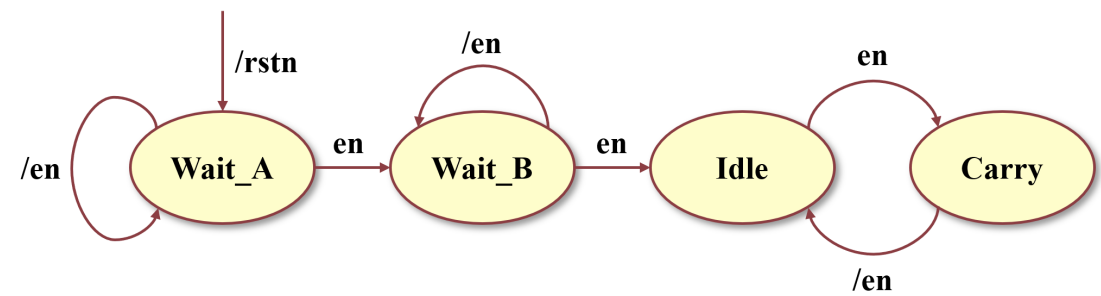
实验 2: FLS 模块

● FLS 模块设计

➔ 要求实现的功能

自主输入两个基数 a 和 b，此后每一次按下 Button（en 信号出现高电平脉冲），都计算 $a + b$ ，并将原来 a 和 b 的寄存器内容更新为 b 和 $a + b$ 。

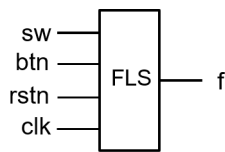
➔ FLS 时序逻辑状态图设计



状态编码：Wait_A[00]，Wait_B[01]，Idle[11]，Carry[10]。上电之后的默认状态为 Wait_A，当通过开关连续输入 a、b 后，到待机状态 Idle，一旦有使能信号出现，则计算和移动一次，返回待机状态。

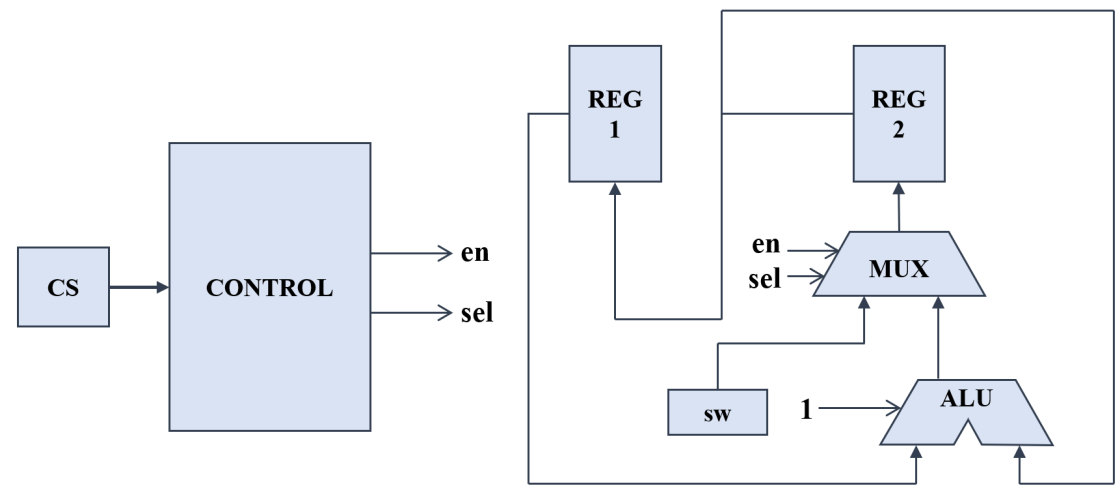
➔ FLS 封装设计

器件图及端口解释：



复位后，前 2 次按下 btn 时，f 分别输出 f_0 和 $f_1 (= sw)$ 。随后每次按下 btn 时，f 依次输出 $f_n = f_{n-2} + f_{n-1}$ ， $n > 1$ 。

➔ FLS 数据通路设计



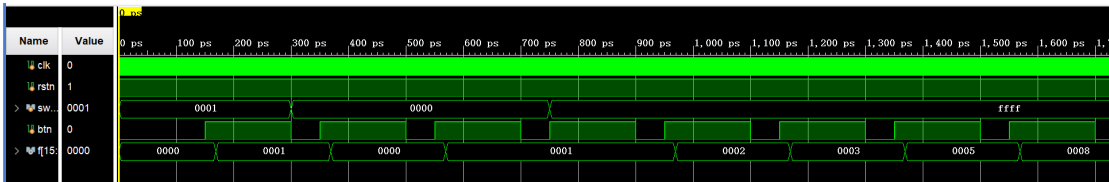
➔ FLS 核心代码

见附件 fls.v。此外，为了方便模块复用，额外编写 mux 和 control 模块，见附件 mux.v 和 control.v。

● FLS 模块仿真及下载测试

➔ FLS 模块的仿真

为了验证 FLS 模块的正确性，编写 testbench 文件，代码见附件 fls_tb.v，模拟结果见下图，经检验结果正确。



➔ FLS 模块与 Nexys4 接口对应关系

由于设计的 FLS 模块已经高度封装化，不需要再额外进行 Top 模块编写，直接生成比特流并烧写到开发板上。

FLS 模块与 Nexys4 开发板的接口对应关系如下，xdc 约束文件见附件 const.xdc。

sw[15:0] → sw[15:0]

rstn → cpu_resetrn

clk → clk100mhz

btn → btnc

f → led[15:0]

➔ FLS 模块下载测试

测试样例：a = 4, b = 7。

预期结果：（下方“点”表示熄灭，“圈”表示亮起）

hex: 0004, led:0..

hex: 0007, led:000

hex: 000b, led:0.00

hex: 0012, led:0..0.

hex: 001d, led:000.0

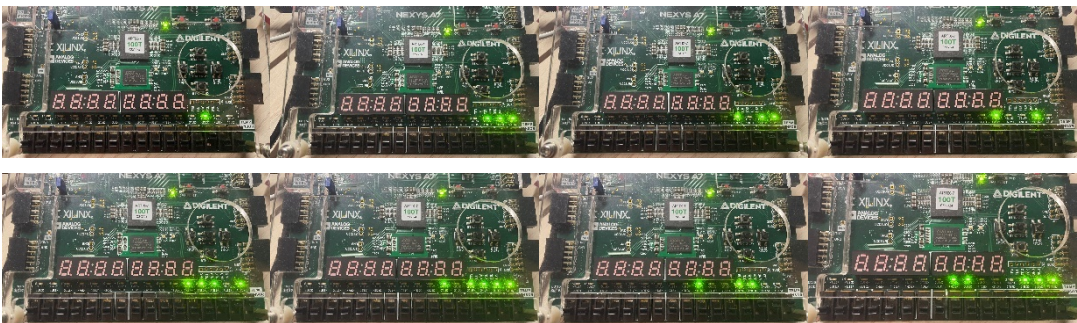
hex: 002f, led:0.0000

hex: 004c, led:0..00..

hex: 007b, led:0000.00

hex: 00c7, led:00...000

测试结果：



与预期结果完全一致。

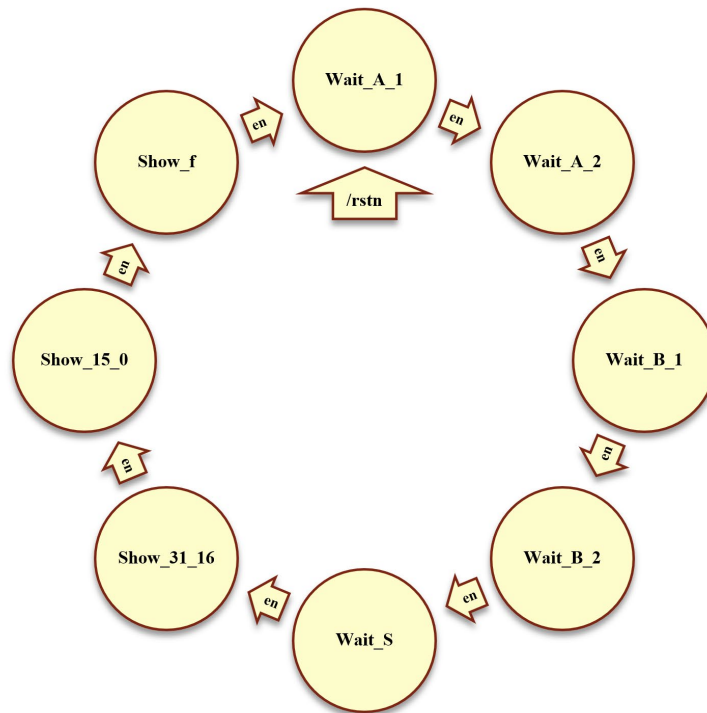
实验选项: 32 位 ALU 设计及下载测试

● 32 位 ALU 设计

➔ 相对于 6 位 ALU 下载测试的改进设计

由于 Nexys4 开发板一次最多输入 16 位，所以需要设计时序状态，分阶段输入。

➔ 32 位 ALU 状态图设计



➔ 32 位 ALU 逻辑设计

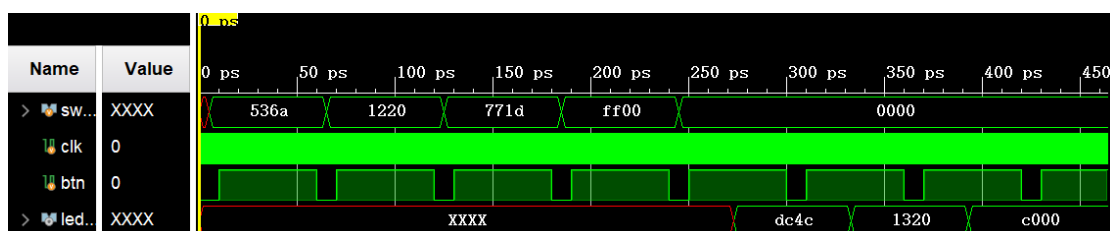
上电之后默认状态为 Wait_A_1，此时拨动开关 sw，按下 btn 后读取 sw 值为操作数 a 的高 16 位，进入下一个状态；此时拨动开关 sw，按下 btn 后读取 sw 值为操作数 a 的低 16 位，进入下一个状态；此时拨动开关 sw，按下 btn 后读取 sw 值为操作数 b 的高 16 位，进入下一个状态；此时拨动开关 sw，按下 btn 后读取 sw 值为操作数 b 的低 16 位，进入下一个状态；此时拨动开关 sw，按下 btn 后读取 sw[15:13] 为操作数 s，进入下一个状态；此时 led 将展示运算结果的高 16 位，再按下 btn，进入下一个状态；此时 led 将展示运算结果的低 16 位，再按下 btn，进入下一个状态；此时 led[15:13] 将展示 f[2:0]，一个计算循环到此结束。

➔ 32 位 ALU 核心代码

见附件 alu_32.v。

➔ 32 位 ALU 的仿真

仿真文件见附件 alu_32_tb.v，仿真结果见下图。即进行运算：0x536a1220 - 0x771dff00，结果为 0xdc4c1320，经验证正确。f[2:0] 给出仅 ltu、lt 两者为高电平，与事实一致。



● 32 位 ALU 下载测试

➔ 32 位 ALU 与 Nexys4 端口对应

sw[15:0] → sw[15:0]

clk → clk100mhz

btn → btnc

led[15:0] → led[15:0]

➔ 32 位 ALU 下载测试

测试用例：

0# s = 3'b 000; a = 32'h ff8710cd; b = 32'h 896a557f;

1# s = 3'b 000; a = 32'h 4543ffff; b = 32'h abcd1234;

2# s = 3'b 001; a = 32'h ff8710cd; b = 32'h 896a557f;

3# s = 3'b 111; a = 32'h ff8710cd; b = 32'h 0000000f;

预期结果：

0# y = 32'h 761cbb4e; f = 3'b 010;

1# y = 32'h 9976edcb; f = 3'b 100;

2# y = 32'h 88f1664c; f = 3'b 000;

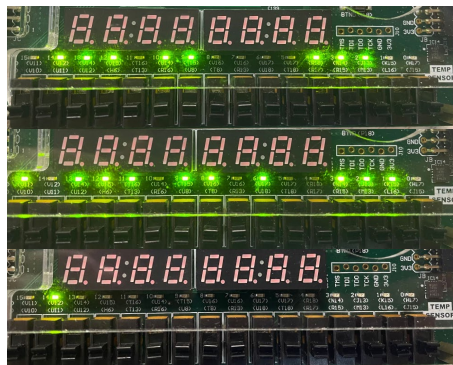
3# y = 32'h ffffffff0e; f = 3'b 000;

测试结果：

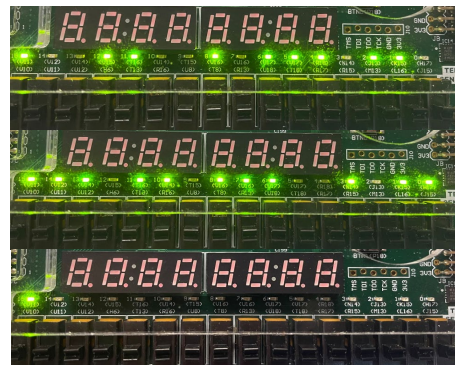
结果高位

结果低位

f[2:0]



0# 正确

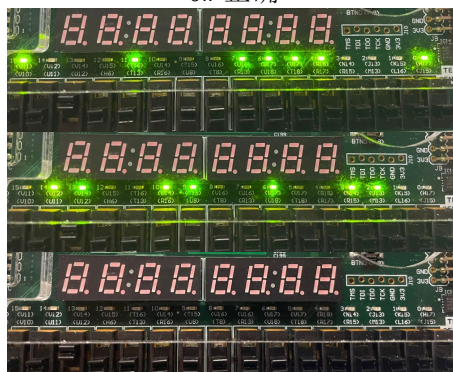


1# 正确

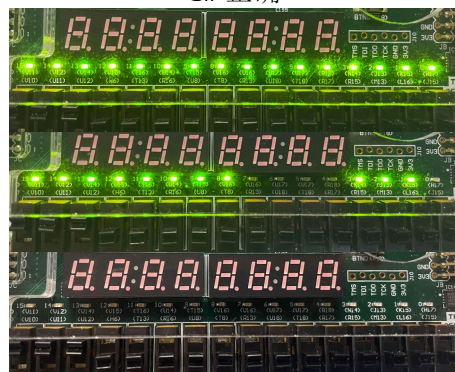
结果高位

结果低位

f[2:0]



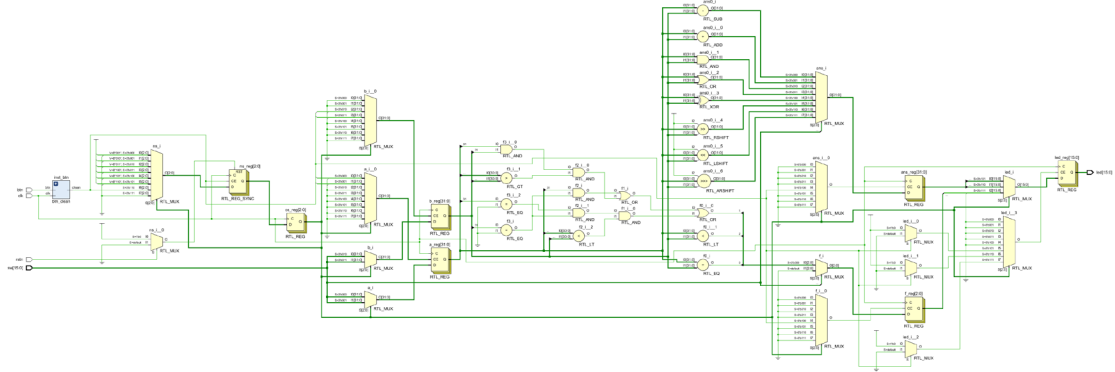
2# 正确



3# 正确

● 32 位 ALU 电路性能

➔ 32 位 ALU RTL 分析电路



➔ 32 位 ALU 电路资源

Tcl ConsoleMessagesLogReportsDesign RunsUtilization x

Q

≡

⬇

Q

≡

⬇

%

Hierarchy

Hierarchy

Summary

▼ Slice Logic

▼ Slice LUTs (1%)

LUT as Logic (1%)

| Name | Slice LUTs (63400) | Slice Registers (126800) | F7 Muxes (31700) | F8 Muxes (15850) | Bonded IOB (210) | BUFCTRL (32) |
|------------------------|--------------------|--------------------------|------------------|------------------|------------------|--------------|
| ▼ alu_32 | 461 | 127 | 38 | 19 | 34 | 1 |
| ❏ inst_btn (btn_clean) | 10 | 6 | 0 | 0 | 0 | 0 |

➔ 32 位 ALU 电路性能

| | | | | | | | | | | | | | |
|---|--|----------|--------|---------|-------------|-------------|---------------|---------------|-------------|-----------|-------------|--------------|--|
| Tcl Console | | Messages | Log | Reports | Design Runs | Timing | | x Utilization | | ? | | □ | |
| Intra-Clock Paths - sys_clk_pin - Setup | | | | | | | | | | | | | |
| Timer Settings | | | | | | | | | | | | | |
| Design Timing Summary | | | | | | | | | | | | | |
| Clock Summary (1) | | | | | | | | | | | | | |
| Check Timing (33) | | | | | | | | | | | | | |
| Intra-Clock Paths | | | | | | | | | | | | | |
| sys_clk_pin | | | | | | | | | | | | | |
| Setup 4.990 ns (10) | | | | | | | | | | | | | |
| Hold 0.132 ns (10) | | | | | | | | | | | | | |
| Pulse Width 4.500 ns (30) | | | | | | | | | | | | | |
| Inter-Clock Paths | | | | | | | | | | | | | |
| Other Path Groups | | | | | | | | | | | | | |
| Intra-Clock Paths - sys_clk_pin - Setup | | | | | | | | | | | | | |
| Name | | Slack | Levels | Routes | High Fanout | From | To | Total Delay | Logic Delay | Net Delay | Requirement | Source Clock | |
| Path 1 | | 4.990 | 5 | 6 | 23 | a_reg[28]/C | ans_reg[22]/D | 4.874 | 1.269 | 3.605 | 10.0 | sys_clk_pin | |
| Path 2 | | 4.992 | 5 | 6 | 21 | a_reg[27]/C | ans_reg[21]/D | 4.872 | 1.269 | 3.603 | 10.0 | sys_clk_pin | |
| Path 3 | | 5.037 | 5 | 6 | 106 | b_reg[1]/C | ans_reg[28]/D | 4.827 | 1.269 | 3.558 | 10.0 | sys_clk_pin | |
| Path 4 | | 5.054 | 6 | 7 | 77 | a_reg[31]/C | ans_reg[12]/D | 4.810 | 1.642 | 3.168 | 10.0 | sys_clk_pin | |
| Path 5 | | 5.059 | 5 | 6 | 21 | a_reg[27]/C | ans_reg[20]/D | 4.805 | 1.269 | 3.536 | 10.0 | sys_clk_pin | |
| Path 6 | | 5.061 | 5 | 6 | 24 | a_reg[29]/C | ans_reg[24]/D | 4.803 | 1.269 | 3.534 | 10.0 | sys_clk_pin | |
| Path 7 | | 5.061 | 5 | 6 | 24 | a_reg[30]/C | ans_reg[25]/D | 4.803 | 1.269 | 3.534 | 10.0 | sys_clk_pin | |
| Path 8 | | 5.068 | 5 | 6 | 23 | a_reg[28]/C | ans_reg[23]/D | 4.796 | 1.269 | 3.527 | 10.0 | sys_clk_pin | |
| Path 9 | | 5.078 | 5 | 6 | 21 | a_reg[25]/C | ans_reg[15]/D | 4.786 | 1.269 | 3.517 | 10.0 | sys_clk_pin | |
| Path 10 | | 5.182 | 5 | 6 | 111 | b_reg[2]/C | ans_reg[26]/D | 4.682 | 1.269 | 3.413 | 10.0 | sys_clk_pin | |

总结篇

● 收获

经过这次实验，我重温了 Verilog 各种语法和状态机的设计实现，为今后的实验做好了准备。这次实验是我第一次拿到 FPGA 电路板 Nexys4 并在其上进行烧写测试，感觉很新奇，也很独特，这令枯燥的实验增添了几分色彩。此外，在进行这次实验的过程中，我多次尝试编写 testbench 来调试各个小模块，这在以前的数电实验中是欠缺的，这次实验补足了我这方面的能力，我也因此解决了很多编写上的问题，可以算是达到了实验的目的。

● 建议

- (1) 这次实验中，我一开始发现算术右移和逻辑右移的结果是完全一模一样的，经过搜索才知道右移时 Verilog 默认左操作数按无符号数解释，并且可以用 \$signed() 函数强制按有符号数解释；但在编写有符号数小于的功能时，单用 \$signed() 函数又不起作用。希望老师在讲 Verilog 基础语法之外，也可以适当补充一些 Verilog 的特性和内置函数。
- (2) 本次实验的要求之一是查看电路资源和电路性能，但是老师并没有给出分析这两项报告的标准，使得这两项任务变得较为鸡肋。建议删去，或者补充。