

加减法器设计

实验目的

- (1) 掌握一位全加器的逻辑功能，学会多位全加器的设计原理。
- (2) 学会利用加法器完成减法器的设计与实现。
- (3) 掌握多个 7 段数码管分时复用显示原理及应用。
- (4) 掌握位拼接运算符的使用。

实验内容

- 1) 设计一个 1 位全加器，然后由 4 个 1 位全加器组成 4 位加法器。
- 2) 在加法器中增加进位标志和溢出标志，设计实现一个带进位标志和溢出标志的 4 位加法器，并编写顶层模块组合加法器模块和 7 段数码管模块，将加法器操作数和结果显示到七段数码管和 led 灯上。
- 3) 用全加器来构建全减器，画出电路图，设计并实现带进位位和溢出位的 4 位加/减法器，
- 4) 将 4 位加减法器改成 32 位加减法器，并进行仿真验证和下板验证。
- 5) 将所设计的加减法器与系统自带的“+/-”进行比较。可以从 RTL 电路分析、仿真波形、开发板资源使用情况等方面进行比较。

注意：加减法器设计不能使用系统自带的“+/-”号。

提示：二进制数按补码进行运算，数值和显示应能明确解释说明是有符号数或无符号数。

实验原理

数码管显示原理参见文档“七段数码管显示原理 basys3”。

实现 32 位加减法器时，当开发板 I/O 资源不足时，可以在通过在设计文件中内置操作数或者在模块实例化时利用位拼接运算符对端口的某些位直接传入高/低电平等方法来解决。如下图 addsubx4 模块在实例化 U1 时，分别将输入端口 a、b 的高 24 位直接传入低电平，赋值为 0。

```

addsubx4 U1(
  {24'b0,a},
  {24'b0,a},
  cin,
  sub,
  cout,
  overflow,
  sum
);

```



```

module addsubx4(
  input [31:0] a,
  input [31:0] b,
  input cin,
  input sub,
  output cout,
  output overflow,
  output [31:0] sum
);

```

实验步骤

- 1) 学习指导书“七段数码管显示原理 basys3”，掌握 FPGA 开发板上的 7 段数码管的显示原理及数码管分时复用显示原理。
- 2) 在 Vivado 中新建项目，设计一个 1 位全加器，然后由 4 个 1 位全加器组成 4 位加法器，并通过写仿真文件、看 RTL 电路图验证其正确性。
- 3) 在步骤 2) 实现的加法器中增加进位标志和溢出标志，设计实现一个带进位标志和溢出标志的 4 位加法器。
- 4) 调用七段数码管显示实验中实现的 16 进制显示模块，然后编写顶层模块组合加法器模块和 7 段数码管显示模块，将加法器操作数和结果显示到七段数码管和 led 灯上。将 SW3~SW0 作为第一操作数（可显示在左起第一个 7 段数码管上），SW7~SW4 作为第二操作数（可显示在左起第二个 7 段数码管上），将结果显示在最右侧的 7 段数码管上，进位和溢出标志通过 LED 灯显示。
- 5) 在 Vivado 中新建项目，用全加器来构建全减器，画出电路图，设计并实现带进位位和溢出位的 4 位加/减法器，组合加/减法器模块和 7 段数码管显示模块，将 SW3~SW0 作为第一操作数（可显示在左起第一个 7 段数码管上），SW7~SW4 作为第二操作数（可显示在左起第二个 7 段数码管上），SW15=0 (1) 时做加（减）法，将结果显示在最右侧的 7 段数码管上，进位和溢出标志可通过 LED 灯显示。
- 6) 在 Vivado 中新建项目，将 4 位加减法器改成 32 位加减法器，并进行仿真验证和下板验证，开发板资源不够时，可以在设计文件中内

置操作数或者在模块实例化时利用位拼接运算符对端口的某些位直接传入高/低电平等方法来解决。

- 7) 在 Vivado 中新建项目，利用系统自带的“+/-”实现加/减法器，通过仿真、下板验证其正确性，并与步骤 6) 中所设计的加减法器进行比较。可以从 RTL 电路分析、仿真波形、开发板资源使用情况等方面进行比较。