实验二 算术逻辑单元（ALU）的设计与实现

一． 实验目的

1. 掌握全加器和行波进位加法器的结构；
2. 熟悉加减法运算及溢出的判断方法；
3. 掌握算术逻辑单元（ALU）的结构；
4. 熟练使用 SystemVerilog HDL 的行为建模和结构化建模方法对 ALU 进行描述实现；
5. 为“单周期 MIPS 处理器的设计与实现”奠定基础。

二． 实验环境

1. 操作系统：Windows 10 或 Ubuntu 16.04
2. 开发环境：Xilinx Vivado 2018.2
3. 硬件平台：远程 FPGA 硬件云平台

三． 实验原理

1. 全加器

如图 2-1 所示，1 位全加器具有三个输入（A, B, C_{in}）和两个输出（S, C_{out}）。输入 A 和 B 各表示 1 位二进制数，两者相加，求和的结果通过 S 输出，进位通过 C_{out} 输出。此外，进位输入 C_{in} 和进位输出 C_{out} 在采用全加器构造多位加法器的时候会被使用。一个全加器的真值表如表 2-1 所示。

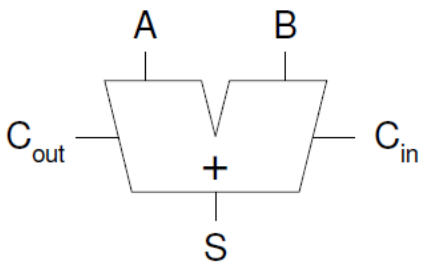

图 2-1 全加器

表 2-1 全加器的真值表

输入			输出	
C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

2. 行波进位加法器

行波进位加法器（ripple-carry adder）是最简单的一种进位传播多位加法器（CPA）。它就是将 N 个全加器进行串联，每级的 C_{out} 就是下一级的 C_{in}，则所有进位构成的通路称为进位链。一个 3 位行波进位加法器的结构如图 2-2 所示。

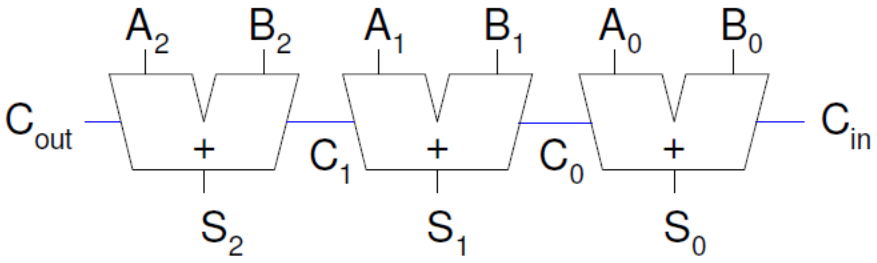

图 2-2 3 位行波进位加法器

3. 算术逻辑单元 ALU

在现代计算机中，算术逻辑单元（ALU）是专门执行算术和逻辑运算的数字电路。ALU 是计算机中处理器的最重要组成部分，甚至连功能最简单的微处理器也会包含 ALU。ALU 的主要功能是进行二进制数的算术和逻辑运算，包括加法、减法、乘法（通常不含除法）、移位运算、逻辑与、逻辑或等等，一个 N 为 ALU

如图 2-3 所示。

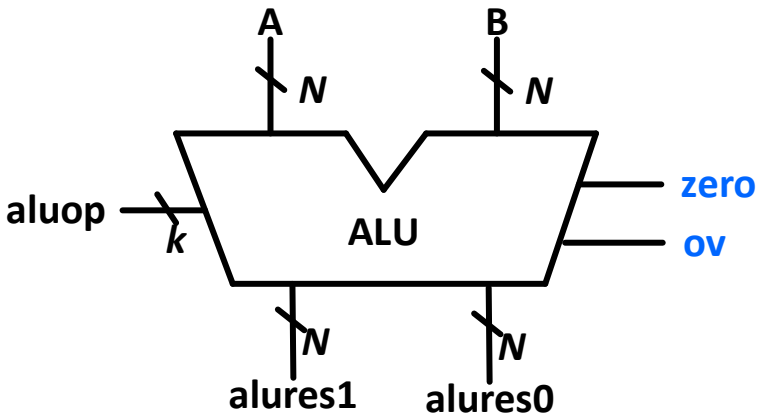

图 2-3 N 位 ALU 单元

它具有 3 个输入和 4 个输出。A 和 B 是两个 N 位的操作数，aluop 为操作类型控制信号，ALU 单元根据 aluop 信号确定对 A 和 B 进行何种操作。alures0 用于输出非乘法操作的结果或乘法运算结果的低 N 位，alures1 用于输出乘法结果的高 N 位（对于非乘法运算恒为 0）。ov 为有符号数加减法的溢出标志，zero 是表示 ALU 结果是否为 0 的零标志（对所有运算均有效）。

四． 实验内容

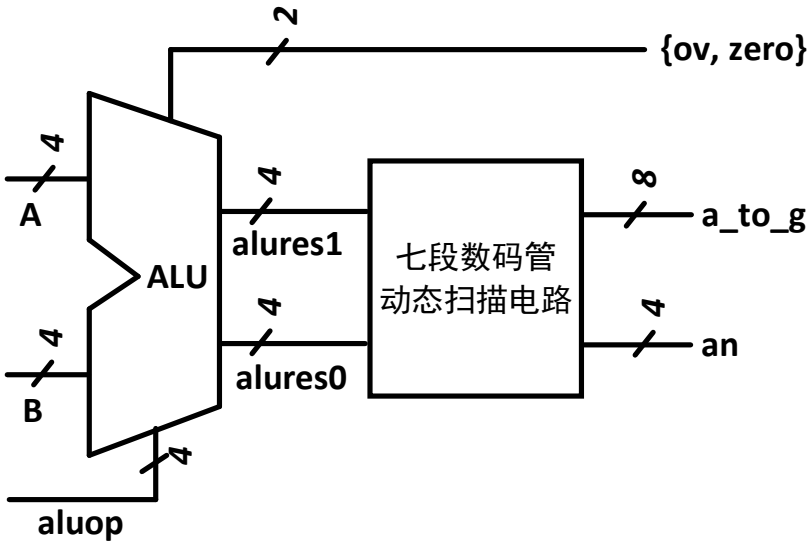

图 2-4 实验的顶层模块

基于 SystemVerilog HDL 设计并实现一个 4 位 ALU 单元。整个工程的顶层模块如图 2-4 所示，输入/输出端口如表 2-2 所示。注意，顶层模块由两个子模块组成，其中，一个是 ALU 单元，另一个是 7 段数码管动态显示扫描单元。**同学们只需要实现 ALU 单元即可**，动态显示扫描单元在工程中直接提供。

表 2-2 输入/输出端口

端口名	方向	宽度（位）	说明
A	输入	4	操作数 A。
B	输入	4	操作数 B。
aluop	输入	4	操作类型。
ov	输出	1	有符号数加/减法溢出标志。
zero	输出	1	零标志（对所有操作均有效）。
a_to_g	输出	8	连接七段数码管的数据输入端 CA~CG 和 dp 段，用于显示 ALU 单元的运算结果。
an	输出	4	连接 4 个七段数码管的使能端，其中 an[0]对应的数码管显示非乘法结果或乘法结果的低 4 位，an[1]对应的数码管显示乘法结果的高 4 位。剩余两个数码管不点亮。

ALU 单元所支持的运算功能如表 2-3 所示。

表 2-3 ALU 单元所支持的运算功能

aluop	助记符	功能	说明[&]
0000	AND	按位与	alures0 = A & B，alures 1 = 0
0001	OR	按位或	alures0 = A | B，alures1 = 0
0010	XOR	按位异或	alures0 = A ⊕ B，alures1 = 0
0011	NAND	按位与非	alures0 = ~(A & B)，alures1 = 0
0100	NOT	逻辑非	alures0 = ~A，alures1 = 0
0101	SLL	逻辑左移	alures0 = A << B（B 取低 3 位），alures1 = 0
0110	SRL	逻辑右移[&]	alures0 = A >> B（B 取低 3 位），alures1 = 0
0111	SRA	算术右移	alures0 = A>>>B（B 取低 3 位），alures1 = 0
1000	MULU	无符号数乘法	alures0 = $(A * B)_{[3:0]}$，alures1 = $(A * B)_{[7:4]}$
1001	MUL	有符号数乘法	alures0 = $(A * B)_{[3:0]}$，alures1 = $(A * B)_{[7:4]}$
1010	ADD	有符号数加法	alures0 = A + B，alures1 = 0，需要设置 ov
1011	ADDU	无符号数加法	alures0 = A + B，alures1 = 0
1100	SUB	有符号数减法	alures0 = A - B，alures1 = 0，需要设置 ov
1101	SUBU	无符号数减法	alures0 = A - B，alures1 = 0
1110	SLT	有符号数比较	alures0 = (A < B)? 1 : 0，alures1 = 0
1111	SLTU	无符号数比较	alures0 = (A < B)? 1 : 0，alures1 = 0

$：所有运算均需要设置 zero 位。

&：算术右移高位补符号位的前提是对有符号数进行右移，需要特别注意这一点。

完成上述 ALU 单元的设计，必需满足如下几点要求：

1. ALU 单元的输入 A 和 B 均是补码形式。
2. 实现加法和减法时，不能使用“+”和“-”两种运算符，且只能通过一个行波进位加法器和其它必要的逻辑电路实现。
3. 可以使用“*”运算符实现乘法，但该运算符在只适用无符号数的乘法，有符号数的乘法需要同学们考虑如何处理。
4. 实现算术右移时，可以使用运算符“>>>”。

五． 实验步骤

1. 解压缩 ALU_4bits_stu.rar，打开教师提供的工程文件 ALU_4bits.xpr，如图 2-5 所示。目前工程中已经提供了 7 段数码管动态扫描电路的网表文件（xseg7_d.edif）和端口声明文件（xseg7_d_stub.sv），以及顶层文件（ALU_4bits.sv）。前两个有关 7 段数码管的文件无需做任何修改。同学们只需设计实现图 2-4 中的 ALU 单元。

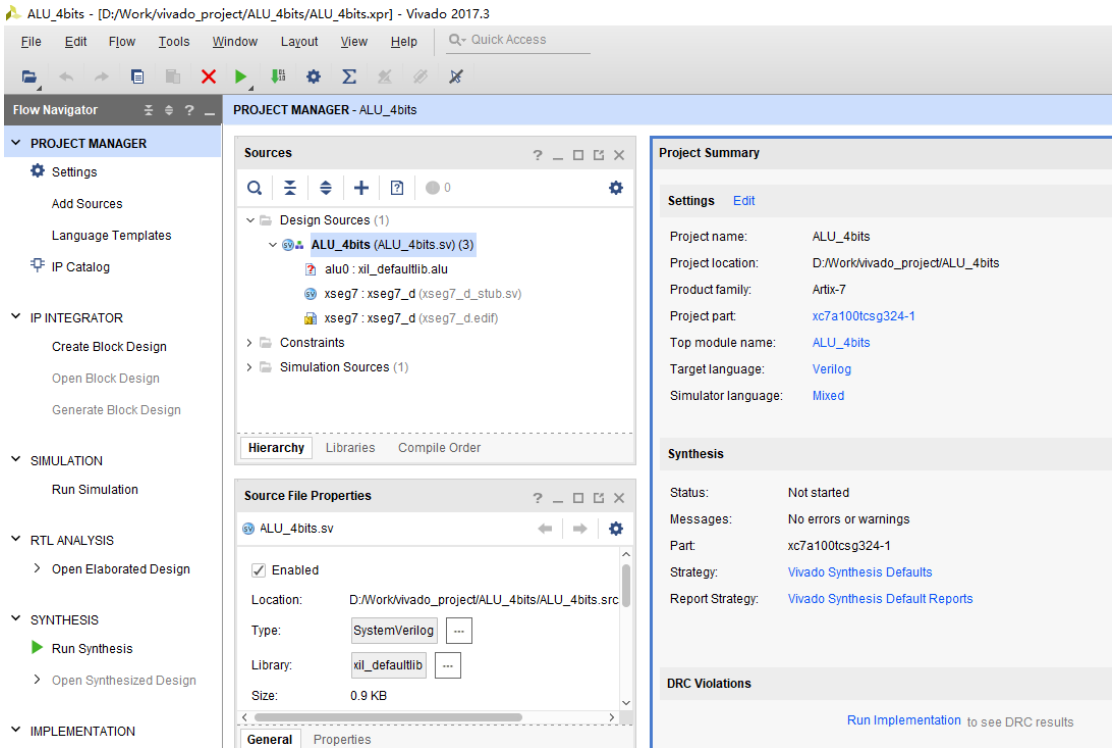

图 2-5 工程 ALU_4bits

2. 在工程中，添加文件 **fulladder.sv**，实现全加器。

3. 在工程中，添加文件 **rca.sv**，调用全加器，实现 4 位行波进位加法器。

4. 在工程中，添加文件 **alu.sv**，通过调用行波进位加法器和添加必要的逻辑电路实现表 2-3 所示的所有运算操作。根据顶层文件对 ALU 单元的调用形式，**alu.sv 文件中必须按如下形式定义模块**。

```
module alu(
    input  [3 : 0]       A,
    input  [3 : 0]       B,
    input   [3 : 0]      aluop,
    output logic [7 : 0] alures,
    output logic         ZF,
    output logic         OF
    );
```

5. 在顶层文件 ALU_4bits.sv 通过结构化建模，完成如图 2-4 所示的系统集成。
6. 添加测试文件 **ALU_4bits_tb.sv**，对所实现的 ALU 单元进行行为仿真（**注意：仅对 alu.sv 进行仿真即可，不要仿真整个系统**）。仿真过程中必须采用**自动化测试**的方法（注：读取文件时，如果不使用绝对路径，请将存放测试向量的文本文件拷贝到“工程路径/ALU_4bits.sim/sim_1/ behave/xsim”目录下，否则读取文件将失败）。
7. 如果第 6 步行为仿真通过，则对工程进行综合、实现、生成 bin 文件。**注意，约束文件已在工程中提供，不需要再添加。**
8. 登录远程 FPGA 硬件云平台，直接导入教师提供的验证平台文件（**4 位算术逻辑单元(ALU).epl**），无需进行绘制，验证平台如图 2-6 所示。其中操作数 A、B 以及操作类型控制信号 aluop 通过多位输入控件进行控制；使用 4 位七段数码管显示 ALU 的运算结果（最低位数码管用于显示非乘法结果或乘法结果

的低 4 位，次低位数码管显示乘法结果的高 4 位，剩下两位数码管在本实验中不被点亮)；使用 2 个 LED 灯显示溢出标志 ov 和零标志 zero。烧写所生成的 bin 文件，然后运行实验，通过调整操作数 A、B 和 aluop 的值验证所设计 ALU 的正确性。

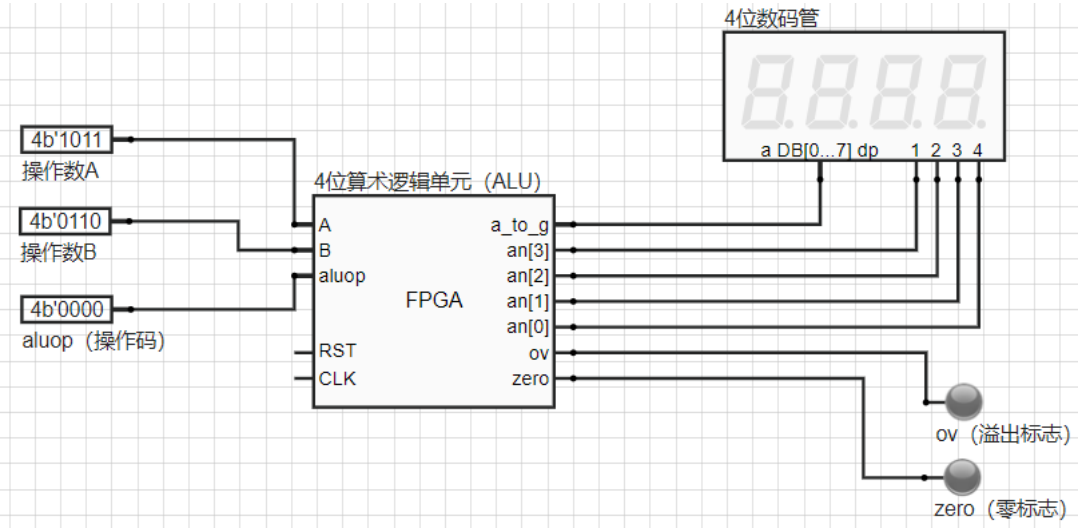

图 2-6 验证平台

六． 实验方式

每位同学独立上机编程实验，实验指导教师现场指导。

七． 参考内容

1. 教材内容和课件

八． 实验报告

1. 画出实现加/减法运算的逻辑电路原理图，并说明为什么加/减法可以只使用一个加法器进行实现?
2. 给出有符号数加/减法溢出的判断规则?
3. 给出 ALU 单元的 SystemVerilog HDL 代码。
4. 给出具有自动化测试功能的仿真程序和对应的波形图截图，并说明为什么选取这些测试向量?