P1 使用Verilog HDL设计功能部件

V1.0@2015.10.26

# 实验目的

1. 能熟练使用ISE工具，掌握建立工程、编写代码和仿真的方法。
2. 掌握Verilog HDL语言的语法要点，能编写出可综合的代码和Testbench测试文件。
3. 会设计较复杂的部件级模块。

# 开发目标

1. 完成4/8/32位加法器、GPR以及ALU等模块，并编写Testbench测试其功能的正确性。

# ADD全加器模块定义（模块分别命名为：add1，add4，add8， add32）

1. 基本描述
   1. 1-bit全加器

1-bit全加器，将三个1-bit数进行相加，输入为A，B以及Cin，其中A,B为二进制补码操作数，Cin为进位位，参见p0的样例程序。全加器以级联的方式构建，可形成4/8/32/64位加法器。

* 1. 4位全加器

4位全加器是简单4个1-bit加法器的级联，将一个加法器的carry-out作为另一个carry-in，如下图所示。



Figure 1 4-bit全加器

1. 8位加法器模块端口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| A[7:0] | I | 第一个加数 |
| B[7:0] | I | 第二个加数 |
| C0 | I | 初始进位 |
| SUM[7:0] | O | 两数相加之和 |
| Overflow | O | 溢出标志位  Overflow=C8 |

1. 按上面类似的方法可以实现32位的加法器，端口可以类推，模块可以分别命名为：add1，add4，add8，add16，add32。

# ALU模块定义（模块名：alu）

1. 基本描述

ALU的主要功能是对输入到ALU的两个二进制数进行加法、减法、按位或、与等操作。

1. 模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| ALU\_DA[31:0] | I | 参与ALU计算的第一个值 |
| ALU\_DB[31:0] | I | 参与ALU计算的第二个值 |
| ALUOp[2:0] | I | ALU功能的选择信号  000：ALU进行加法运算  001：ALU做减法运算  010：ALU进行与运算  100：ALU进行或运算  101：ALU进行异或运算 |
| ALU\_DC[31：0] | O | ALU的计算结果 |
| ALU\_Zero | O | 1：ALU\_DC为0  0：ALU\_DC不为0 |

1. 功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 加运算 | ALU\_DC= ALU\_DA+ ALU\_DB |
| 2 | 减运算 | ALU\_DC= ALU\_DA- ALU\_DB |
| 3 | 与运算 | ALU\_DC= ALU\_DA& ALU\_DB |
| 4 | 或运算 | ALU\_DC= ALU\_DA|ALU\_DB |
| 5 | 异或运算 | ALU\_DC= ALU\_DA^ALU\_DB |

# GPR模块定义（模块名：gpr）

1. 基本描述

GPR以32个32位具有写使能的寄存器为基础，其主要功能是对寄存器堆进行存取操作。

1. 模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Clk | I | MIPS-C处理器时钟 |
| Reset | I | 复位信号 |
| RS1[4:0] | I | 读寄存器文件时的第1个寄存器下标 |
| RS2[4:0] | I | 读寄存器文件时的第2个寄存器下标 |
| RD[4:0] | I | 写寄存器文件时的寄存器下标 |
| RegWrite | I | 寄存器文件写使能 |
| WData[31:0] | I | 寄存器文件写入数据 |
| RData1[31:0] | O | 读寄存器文件时的第1个寄存器的输出 |
| RData2[31:0] | O | 读寄存器文件时的第2个寄存器的输出 |

1. 功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读寄存器 | RData1输出RS1[4:0]所寻址的寄存器。  RData2输出RS2[4:0]所寻址的寄存器。 |
| 2 | 写寄存器 | 当时钟上升沿到来时，并且RegWrite有效时，WData被写入RD[4:0]所寻址的寄存器 |
| 3 | 复位 | reset=1时，使32个gpr单元清零 |

# 问答

1. 为什么减法操作可以通过加法操作来完成？请在实验报告中进行回答。

# Project的提交

1. 首先按照 “二、开发目标”的内容，编程实现部件级模块并编写Testbench进行仿真验证，并且将\*.v文件上传至“MOOC平台/课下Projects/P1/P1测试点”进行测试，测试后将\*.v文件提交至课程系统平台（http://10.254.25.5/course）。文件的命名格式：p1\_模块名\_学号.v。时间截止要求：实验课前一天。
2. 实验报告的撰写请按照“报告书写规则”的要求进行撰写，撰写后与[15]中的\*.v文件一起打包上传，实验报告的命名格式：p1\_ 学号\_姓名.doc。打包文件的命名：**p1\_ 学号\_姓名.zip**。时间截止要求：实验课前一天。

# 成绩及实验测试要求

1. 实验成绩包括但不限于以下内容：
2. 在课程系统平台提交的Project情况。
3. 在实验课上，首先会通过MOOC平台完成一个30分钟的客观题目的测试，题型为选择题。主要考察对Verilog HDL语法的掌握情况，该部分成绩计入本次实验成绩。
4. 客观题目的测试后，进行主观题目的测试，现场编程实现一个部件级模块并进行编写Testbench进行仿真，然后通过MOOC平台的“课上project”功能，自动测试。教师检查结果时，要能回答教师的提问。该部分的得分情况计入本次实验成绩。
5. 实验报告的情况。

# 开发环境资料

本次实验使用ISE集成开发环境，ISE是集成综合环境的缩写，它是Xillinx FPGA/CPLD 的综合性集成设计平台，该平台集成了设计、输入、仿真、逻辑综合、布局布线与实现、芯片下载与配置、等几乎所有流程所需要的工具。

1. Verilog HDL的学习

关于Verilog HDL语法的相关PPT和视频已经在mooc.buaa.edu.cn 开放，可以通过课程名称《计算机EDA技术》MOOC平台进行，访问地址：http://www.mooc.buaa.edu.cn/courses/BUAA/M\_F06D4310/2015\_T1/about 。

1. 软件下载和安装

实验使用的开发环境软件的版本是ISE14.7，下载地址：<http://www.xilinx.com/support/download/index.htm>，安装请看视频《ISE 工具的安装》。

1. 工具使用视频

工具集（Logisim, Mars, ISE）相关教学辅助视频已在 mooc.buaa.edu.cn 开放，课程名称《数字系统设计工具集》，访问地址： <http://mooc.buaa.edu.cn/courses/BUAA/M_G06B2830/2014_T1/about>。

关于ISE 工具的使用视频的说明：详见P0。