

实验一 基础练习

1 实验目的

本实验的目的是熟悉实验环境，学习如何安装 Vivado、如何使用 Vivado 2018 创建工程、代码编辑、RTL 分析、仿真等一系列设计流程。

2 实验内容

(1) Vivado 设计流程学习

学习视频和文档资料，了解 Vivado 设计流程和功能：

- EGo 五分钟快速上手-HDL 流程视频

(<https://www.bilibili.com/video/av24701535/>)

- Ego 五分钟搭建你的数字积木视频

(<https://www.bilibili.com/video/av10888635?from=search&seid=1807344940224421836>)

- Vivado 安装的两个视频

- Vivado 设计流程中的基本概念.pdf，什么是约束文件.ppt

(2) Vivado 代码编辑和 RTL 分析

创建一个工程，自己指定工程位置和工程名称，新建空白源程序文件，依次完成下面代码编辑和 RTL 分析：

分别编写教材图 2.37、2.38 例子，观察 vivado 工具的 RTL 分析结果，截图放入实验报告；

```
module example1 (x1, x2, s, f);  
    input x1, x2, s;  
    output f;  
  
    not (k, s);  
    and (g, k, x1);  
    and (h, s, x2);  
    or (f, g, h);  
  
endmodule
```

```
module example2 (x1, x2, x3, x4, f, g, h);  
    input x1, x2, x3, x4;  
    output f, g, h;  
  
    and (z1, x1, x3);  
    and (z2, x2, x4);  
    or (g, z1, z2);  
    or (z3, x1, ~x3);  
    or (z4, ~x2, x4);  
    and (h, z3, z4);  
    or (f, g, h);  
  
endmodule
```

分别编写教材图 2.40、2.41 例子，观察 vivado 工具的 RTL 分析结果，截图放入实验报告；

```

module example3 (x1, x2, s, f);
    input x1, x2, s;
    output f;

    assign f = (~s & x1) | (s & x2);

endmodule

module example4 (x1, x2, x3, x4, f, g, h);
    input x1, x2, x3, x4;
    output f, g, h;

    assign g = (x1 & x3) | (x2 & x4);
    assign h = (x1 | ~x3) & (~x2 | x4);
    assign f = g | h;

endmodule

```

分别编写教材图 3.18、3.20 例子，对比 vivado 工具的 RTL 分析结果，截图放入实验报告，给出你对结果的理解；

```

module fulladd (Cin, x, y, s, Cout);
    input Cin, x, y;
    output s, Cout;

    xor (s, x, y, Cin);
    and (z1, x, y);
    and (z2, x, Cin);
    and (z3, y, Cin);
    or (Cout, z1, z2, z3);
endmodule

module fulladd (Cin, x, y, s, Cout);
    input Cin, x, y;
    output s, Cout;

    assign s = x ^ y ^ Cin;
    assign Cout = (x & y) | (x & Cin) | (y & Cin);
endmodule

```

编写教材图 3.22 例子，观察 vivado 工具的 RTL 分析结果，截图放入实验报告，给出你对结果的理解；

```

module adder4 (carryin, x3, x2, x1, x0, y3, y2, y1, y0, s3, s2, s1, s0, carryout);
    input carryin, x3, x2, x1, x0, y3, y2, y1, y0;
    output s3, s2, s1, s0, carryout;

    fulladd stage0 (carryin, x0, y0, s0, c1);
    fulladd stage1 (c1, x1, y1, s1, c2);
    fulladd stage2 (c2, x2, y2, s2, c3);
    fulladd stage3 (c3, x3, y3, s3, carryout);

endmodule

module fulladd (Cin, x, y, s, Cout);
    input Cin, x, y;
    output s, Cout;

    assign s = x ^ y ^ Cin;
    assign Cout = (x & y) | (x & Cin) | (y & Cin);

endmodule

```

(3) 基本 Verilog 练习

在线完成课程中心（202.204.62.165）平台下“第三周作业”下面的题目。

(4) Vivado 设计流程练习 - 流水灯设计

- ✓ 按照“01-流水灯设计.pdf”，完成流水灯实验,请务必按照 pdf 文件的要求命名.v 文件。
- ✓ 其中 counter.v 的代码如下：

```
module counter(  
    input clk,  
    input rst,  
    output clk_bps  
);  
    reg [13:0]cnt_first,cnt_second;  
    always @( posedge clk or posedge rst )  
        if( rst )  
            cnt_first <= 14'd0;  
        else if( cnt_first == 14'd10000 )  
            cnt_first <= 14'd0;  
        else  
            cnt_first <= cnt_first + 1'b1;  
    always @( posedge clk or posedge rst )  
        if( rst )  
            cnt_second <= 14'd0;  
        else if( cnt_second == 14'd10000 )  
            cnt_second <= 14'd0;  
        else if( cnt_first == 14'd10000 )  
            cnt_second <= cnt_second + 1'b1;  
    assign clk_bps = cnt_second == 14'd10000 ? 1'b1 : 1'b0;  
endmodule
```

- ✓ 其它.v 文件在资料中已提供，请自行添加到工程中

3 实验要求

1) 观看提供的所有视频资料；学习 Vivado 设计流程中的基本概念.pdf、约束文件.ppt。在实验报告中回答以下问题：

- a) 描述 Vivado 的设计流程
 - b) 什么是网表
 - c) 什么是约束文件？通过 IO planning 完成的是什么方面的约束？
 - d) Vivado 设计流程中，Synthesis 的作用是什么？
 - e) Vivado 设计流程中，Implementation 的作用是什么？
- 2) 在实验报告中提交实验内容（2）的 RTL 详细设计图及相关文字说明；
- 3) 在实验报告中提交实验内容（4）的仿真结果截图及板上运行结果照片；
- 4) 实验报告提交到 202.204.62.165 平台；
- 5) 实验内容（4）的流水灯板上运行结果找助教或老师演示验收，可以第

二次上机时验收（未验收者实验 1 无成绩）