

实验内容

1、实验目的

- 熟悉MINISYS实验板的功能和使用方法；
- 掌握Vivado的开发环境及开发流程；
- 掌握Verilog语言描述组合逻辑电路，掌握约束文件编写，理解仿真波形。

2、实验内容

2.1 译码器

按照实验指导书[Vivado使用教程](#)完成3-8译码器项目创建和文件的添加，运行综合、实现、生成比特流文件，下载到Minisys开发板，自行验证结果。

所给的3-8译码器参考实现接口信号定义如下，完整代码在指导书中已给出。功能同74HC138芯片，en[2:0]是使能信号，为3'b100时使能有效，若使能无效，data_out输出高电平。

```
module decoder_38 (
    input wire [2:0] en,           // 3位译码器使能，接拨码开关SW23-S21
    input wire [2:0] data_in,      // 3位译码器数据输入，接拨码开关SW2-SW0
    output reg [7:0] data_out     // 8位译码后的输出，接GLD7-0
);
endmodule
```

当使能en=3'b100有效时的真值表如下。

data_in[2:0]		data_out[7:0]
0 0 0	1 1 1 1 1 1 1 0	
0 0 1	1 1 1 1 1 1 0 1	
0 1 0	1 1 1 1 1 0 1 1	
0 1 1	1 1 1 1 0 1 1 1	
1 0 0	1 1 1 0 1 1 1 1	
1 0 1	1 1 0 1 1 1 1 1	
1 1 0	1 0 1 1 1 1 1 1	
1 1 1	0 1 1 1 1 1 1 1	

2.2 多路复用器

使用Verilog实现2输入4位多路复用器，拨码开关作为输入，输出驱动LED显示，运行仿真、上板验证。详细要求如下：

- 拨码开关SW23作为使能信号en输入；
- 拨码开关SW22作为功能选择信号mux_sel输入；
- 拨码开关SW3-SW0作为input_a[3:0]输入，SW7-SW4作为input_b[3:0]输入；
- 输出信号output_c连接到开发板的GLD3-0
- 使能en=0时，输出全1。en=1时，若mux_sel为1，计算a-b作为输出；mux_sel为0，计算a+b作为输出。直接用+、-运算符实现，无需考虑正负，无需考虑溢出。

待实现的多路复用器接口定义：

```
module mux (
    input wire en,           // 1位使能
    input wire mux_sel,      // 1位选择信号
    input wire [3:0] input_a, // 4位输入数据a
    input wire [3:0] input_b, // 4位输入数据b
    output reg [3:0] output_c // 4位输出数据，驱动LED显示
);
endmodule
```

多路复用器真值表：

en	mux_sel	input_a	input_b	output_c
1	1	xxxx	xxxx	input_a - input_b
1	0	xxxx	xxxx	input_a + input_b
0	1	xxxx	xxxx	1111
0	0	xxxx	xxxx	1111

3、实验步骤

3.1 译码器

按照[Vivado使用教程](#)的步骤实现3-8译码器，上板自行验证功能，不做检查。上板验证时请将开发板按如下方向放置：正面向上，电源接口置于顶部，拨码开关置于底部，基本的验证操作如下：

- en=000，输出8'ff，即左侧SW23-SW21都往下拨，右侧8个绿色的LED灯GLD7-0全亮。
- en=100，data_in输入0，输出8'hfe，即SW23往上拨，SW22-21仍然往下拨，数据输入的SW2-0往下拨，输出GLD0不亮，其他7个绿色的LED灯亮。

- en=100，data_in输入1，输出是8'hfd，使能信号不变，输入SW0往上拨，SW2-1不变，输出GLD1不亮，其他7个绿色的LED灯亮。

3.2 多路复用器

- 创建工程，工程名为mux；
- 自行编写**并添加设计文件mux.v；
- 添加仿真文件testbench.v，代码如下，完成仿真；

```
`timescale 1ns/1ps
```

```
module mux_sim();
```

```
    wire en;
    wire mux_sel;
    wire [3:0] input_a;
    wire [3:0] input_b;
    wire [3:0] output_c;
```

```
    reg [9:0] switch;           // 中间变量
    assign {en, mux_sel, input_a, input_b} = switch; // 位拼接符{}组合信号
```

```
    initial
    begin
        #0 switch = 10'b0_1_0010_0001; // 2组用例测试: en=0
        #5 switch = 10'b0_0_0010_0001;
```

```
        #5 switch = 10'b1_0_0001_0000; // 4组用例测试: en=1, mux_sel=0, a+b
```

```
        #5 switch = 10'b1_0_0001_0001;
        #5 switch = 10'b1_0_1000_0100;
        #5 switch = 10'b1_0_0100_0001;
        #10 $finish;
```

```
    end
```

```
    mux u_mux (
        .en(en),
        .mux_sel(mux_sel),
        .input_a(input_a),
        .input_b(input_b),
        .output_c(output_c)
    );
endmodule
```

注意

后续的实验中，若指定了工程名、文件名请按照指定的名字命名，没有指定名字的可自行命名。

实验中凡是需要用到但指导书未提供的代码，都需要自己写。

4、课上检查

- 3-8译码器自行按步骤说明验证无需检查。
- 多路复用器开发板验证通过：
 - 检查时提前将bit文件烧录到开发板，并自行做一遍基础的功能测试，再举手示意等老师检查。(此为通用要求，每个实验都应在检查前准备好)
 - 具体验证步骤参考如下：
 - 测试使能无效：所有拨码开关都往下拨，输入低电平，此时en=0，使能无效，右侧的GLD3-0四个LED亮。改变输入input_a和input_b，输出不变。
 - 测试a+b：拨码开关SW23往上拨en=1，SW22往下拨mux_sel=0，SW7-SW0自行调整构造输入数据input_a和input_b，观察GLD3-0的情况是否等于input_a + input_b。
 - 测试a-b：拨码开关SW23往上拨en=1，SW22往下拨mux_sel=1，SW7-SW0自行调整构造输入数据input_a和input_b，观察GLD3-0的情况是否等于input_a - input_b。

- 检查截止时间：**第8周周五(2025.10.24)**，截止时间之后检查扣除一定的分数。

注意

当次实验课没做完，请于下次课找本班老师检查。

备份好比特流文件，下次可在Vivado首页点击“Open Hardware Manager”连接开发板选择比特流文件烧录，无需重新生成。

Vivado 2023.2

File Flow Tools Window Help Q- Quick Access

AMD Vivado ML Edition

Quick Start

Create Project >

Open Project >

Open Example Project >

Tasks

Manage IP >

Open Hardware Manager >

Vivado Store >

5、课后提交

实验1应提交下面这些文件，打包成zip格式提交：

```
└── mux.v
└── pin.xdc
└── 学号_姓名.pdf
```

实验报告内容请参照以下要求，工具不限，最后导出pdf提交。

实验1：Vivado使用与组合电路
姓名： 学号： 班级：

一、多路复用器仿真波形分析
1.1 仿真波形截图

- 能正确、清晰体现所要求的功能
- 实验室电脑win+shift+s截图后默认是复制，可以直接粘贴到文档或者聊天窗口

1.2 波形分析
结合模块功能，对关键信号和信号的关键变化做说明分析

二、多路复用器电路图
2.1 RTL Analysis (RTL分析图) 截图
注意不是Device图

2.2 Synthesis schematic (综合后电路图) 截图

三、请回答：
(1) 比特流文件 (.bit文件) 默认在Vivado工程的哪个子目录下?
(2) 以指导书所给的3-8译码器工程为例，若芯片型号选成xa7a12tcpg238-2I是否可以?
如果不可以，会在什么阶段出现什么样的错误?
请修改测试，给出“出错的截图”，并说明怎么修改已有的Vivado工程芯片型号。

• RTL Analysis和Synthesis schematic获取请查看：[综合实现、生成比特流和上板一节](#)。

• 仿真分析参考：[3-8译码器仿真分析示例](#)

• 其他提交要求请再次仔细查看：[课程概况：实验报告提交](#)，此部分通用要求后续不再重复说明。

• 尤其注意：截止时间后会继续开放作为补交，迟交1周（7*24小时）以内的扣除报告分10%，迟交1周以上的扣除报告分的20%。

当次实验课没做完，请于下次课找本班老师检查。

备份好比特流文件，下次可在Vivado首页点击“Open Hardware Manager”连接开发板选择比特流文件烧录，无需重新生成。

注意

当次实验课没做完，请于下次课找本班老师检查。

备份好比特流文件，下次可在Vivado首页点击“Open Hardware Manager”连接开发板选择比特流文件烧录，无需重新