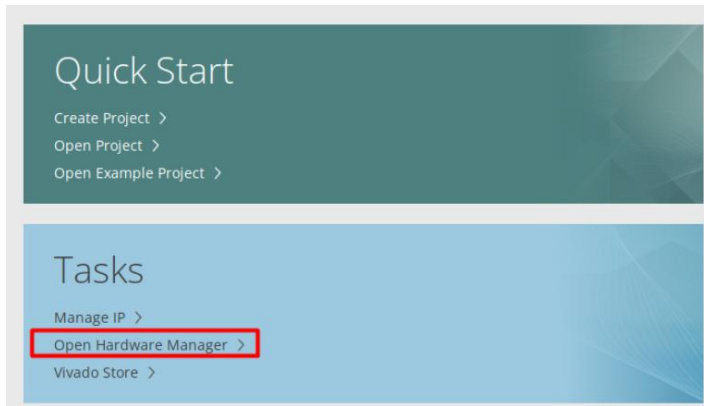


- ❑ 课后需要一定的时间才能完成实验，3个实验室在课后正常开放
- ❑ 本次实验课没做完，请于下次课找本班老师检查
- ❑ 备份好比特流文件，下次可直接在Vivado首页点击 “Open Hardware Manager” 连接开发板烧录，无需再重新生成



# 实验2 寄存器



HITSZ 实验与创新实践教育中心  
Education Center of Experiments and Innovations, HITSZ

# 实验目的

---

- ❑ 掌握Verilog语言描述时序逻辑电路，理解仿真波形
- ❑ 掌握仿真文件的编写，理解时序电路的仿真

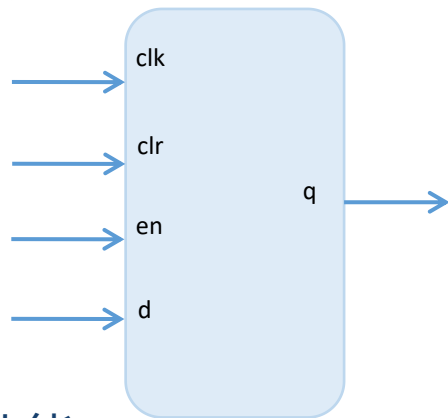


# 实验内容（1）：D触发器



## □ 异步复位、同步使能的D触发器

- clk: 时钟信号，接Y18管脚的板载100MHz时钟源
- clr: 异步复位高电平有效，接按键开关S1
- en: 高电平使能，接拨码开关SW23
- d: 数据输入，接拨码开关SW0
- q: 输出，接最右侧的GLD0



## □ 仿真文件已提供，自行编写RTL和约束代码，自行上板验证功能

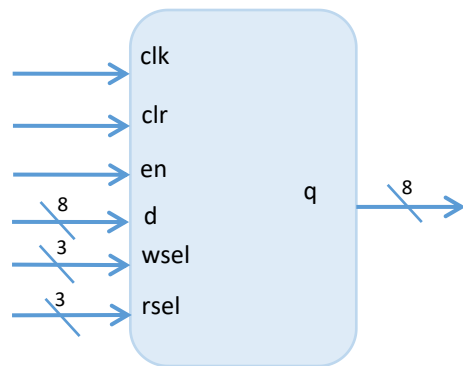
## □ 按键开关，只要按一下，不是一直按着！！！！

# 实验内容（2）：寄存器文件



## □ 8个8位寄存器组成的寄存器文件

- 将D触发器扩展为8位，数量扩展到8个
- wsel[2:0]：写选择端，即写入哪个寄存器，接拨码开关SW22-SW20
- rsel[2:0]：读选择端，即读取哪个寄存器，接拨码开关SW10-SW8
- 读端口是异步逻辑，地址改变输出数据跟着变，无需与时钟边沿同步
- 所有实验中未注明的细节可自行决定



# 寄存器文件的RTL代码

---

- ❑ 多个多位宽寄存器定义

  - ✓ 向量类型的数组: `reg [7:0] regfile [7:0]`

- ❑ 一个文件只定义一个module

- ❑ module名应与文件名一致

- ❑ 时钟使用上升沿posedge触发

- ❑ 实验规范要求详见指导书 “Verilog代码规范” 一节



# 仿真文件编写：D触发器为例

❑ `timescale 1ns/1ps : 仿真时间单位和精度

❑ module testbench () : 模块定义

❑ 内部信号定义和激励信号构造

❑ 时钟信号的生成

❑ 模块例化

```
`timescale 1ns/1ps

module testbench ();

    reg clk;
    reg clr;
    reg en;
    reg d;
    wire q;

    initial begin
        clr = 1'b1; // 初始复位，所有输入初始化
        en = 1'b0;
        clk = 0;
        d = 0;

        #10; // 写入
        clr = 1'b0;
        en = 1'b1;
        d = 1'b1;
        #10 d = 1'b0;
        #10 d = 1'b1;

        #10; // 读取
        en = 1'b0;
        d = 1'b0;

        #10 clr = 1'b1; // 异步清零

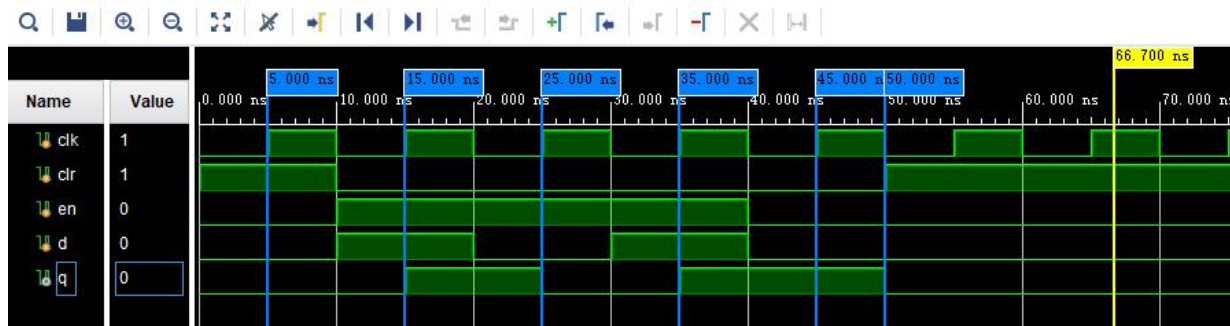
        #50 $finish;
    end

    always #5 clk = ~clk; // 生成时钟

    dff u_dff( // 模块例化
        .clk(clk),
        .clr(clr),
        .en(en),
        .d(d),
        .q(q)
    );

endmodule
```

# 时钟信号的仿真



```
reg clk;  
  
initial begin  
    clk = 0;  
end  
  
always #5 clk = ~clk;
```

- 定义时钟信号clk
- 首先在initial块中将clk初始化为0
- 另用一个always块周期性的让clk信号翻转



# 可综合&不可综合

---

- ❑ verilog大部分语法用于仿真，不可综合（不能生成电路）
  - ❑ 比如延迟5个时间单位：#5，无法综合



```
<module_name> <instance_name> (  
    //被例化的端口与例化的端口  
    .<port_name1> (<signal_name1>),  
    .<port_name2> (<signal_name2>)  
);
```

- ❑ 调用模块的过程称为实例化 (instantiation)
- ❑ 功能文件、仿真文件都可以例化，语法格式相同
- ❑ 每个例化模块称为模块的实例 (instance)
- ❑ 端口连接规则：命名例化方法
  - ❑ 明确定义将信号连接到端口

# D触发器例化

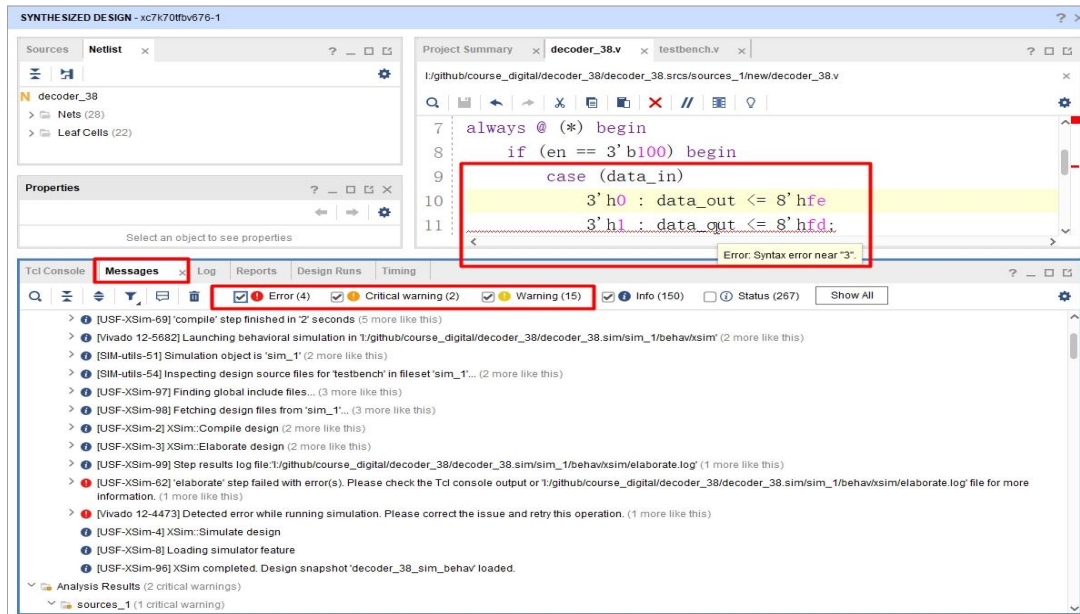


```
dff u_dff(  
    .clk(clk),  
    .clr (clr),  
    .en (en ),  
    .d  (d ),  
    .q  (q )  
);
```

- ❑ dff为模块名， u\_dff为实例名
- ❑ 明确定义将信号连接到的端口，通常两者命名相同或相近

# 错误信息查看

- ❑ Errors, 无法走后续的流程
- ❑ Critical Warnings严重警告



# 实验步骤：D触发器

---

- ❑ 创建工程，工程名为dff;
- ❑ 添加设计文件dff.v;
- ❑ 添加仿真文件运行仿真;
- ❑ 添加约束文件，并综合实现，生成比特流;
- ❑ 将生成的比特流下载到开发板验证。

# D触发器上板自行验证

---

- ❑ 写入：SW23往上拨使能打开，上下拨动SW0改变数据输入，输出GLD0需同步变化
- ❑ 读取：SW23往下拨使能关闭，上下拨动SW0改变数据输入，输出GLD0无变化
- ❑ 清零：在GLD0亮的情况下，按下按键开关S1，GLD0灭

# 实验步骤：寄存器文件

---

- ❑ 创建工程，工程名为reg8file;
- ❑ 编写并添加设计文件reg8file.v;
- ❑ 根据仿真分析要求，编写并添加仿真文件testbench.v;
- ❑ 编写并添加约束文件，并综合实现，生成比特流;
- ❑ 将生成的比特流下载到开发板验证;

# 课上检查&课后提交

---

## ❑ 寄存器文件上板检查

- ❑ 检查截止时间：第9周周五（2025.10.31）

## ❑ 课后提交

- ❑ 请参考在线指导书的要求

- ❑ 提交DDL：第10周周1





# 开始实验



HITSZ 实验与创新实践教育中心  
Education Center of Experiments and Innovations, HITSZ