

实验六 触发器与 PC 寄存器

1. 实验介绍

在本次实验中，我们将使用 Verilog HDL 语言实现 D 触发器、JK 触发器以及 PC 寄存器的设计和仿真。

2. 实验目标

- 深入了解 D 触发器、JK 触发器、D 触发器构成的 PC 寄存器的原理。
- 学习使用 Verilog HDL 行为描述语言设计 D 触发器、JK 触发器、D 触发器构成的 PC 寄存器。

3. 实验原理

1) 触发器

触发器是一种同步双稳态器件，用来记忆一位二进制数。所谓同步，是指触发器的记忆状态按时钟（CLK）规定的启动指示点（脉冲边沿）来改变。下面将介绍几种触发器原理：

a) SR 触发器

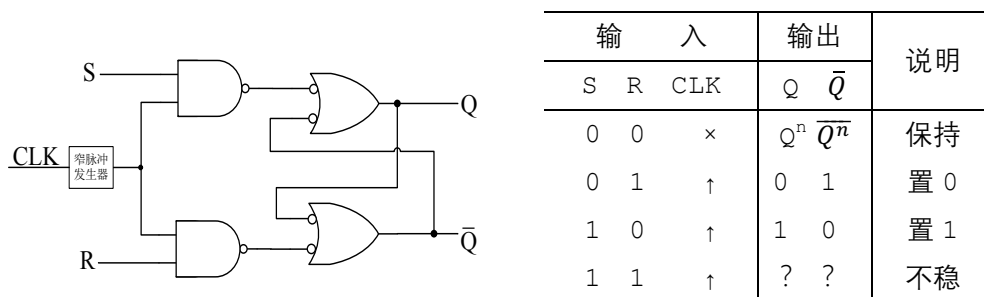


图 6.6.1 SR 触发器逻辑图及其功能表

图 6.6.1 给出了 SR 触发器的逻辑图和其功能表。其中 x 表示无关，↑ 表示时钟信号由低到高。数据输入端 S 和 R 称为同步输入，因为这个两个输入的数据只会在时钟脉冲上升沿时被传送到触发器。

- 当 S 为高 R 为低时，Q 输出在时钟脉冲上升沿时变高，触发器置 1。
- 当 S 为低 R 位高时，Q 输出在时钟脉冲上升沿时变低，触发器置 0。
- 当 S 和 R 两者都为低时，Q 输出状态不会发生变化（保持）。
- 当 S 和 R 两者都为高时，Q 输出状态是不稳定的。

b) D 触发器

D 触发器是在 SR 触发器基础上构建的。即在一个 SR 触发器上增加一个非门。图 6.6.2

给出了一个 D 触发器的逻辑图及其功能表。该触发器只有一个数据输入端 D，经反相器反相后，变成互补数据输入，送到 SR 触发器，从而避免了 SR 触发器存在的不稳态问题。当数据输入 D=1，且在时钟脉冲上升沿，触发器置位（1 状态）；当数据输入 D=0，且在时钟脉冲上升沿，触发器复位（0 状态）。

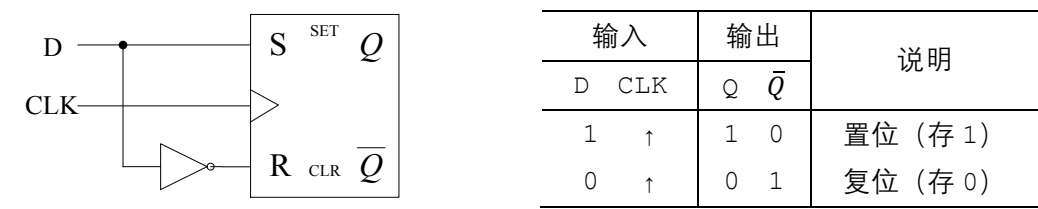


图 6.6.2 D 触发器逻辑图及其功能表

c) JK 触发器

JK 触发器是一种广泛应用的触发器类型。字母 J 和 K 只表示它们是两个数据输入端符号，没有什么特别含义。JK 触发器与 SR 触发器在置位、复位方面的功能是相同的，不同之处在于，JK 触发器改进了 SR 触发器存在的不稳定状态。当 J=1，K=1 时，对每一个连续的时钟脉冲，触发器可改变成相反状态或计数状态，这种工作方式称为交替操作。图 6.6.3 所示为 JK 触发器的内部逻辑及其功能表。

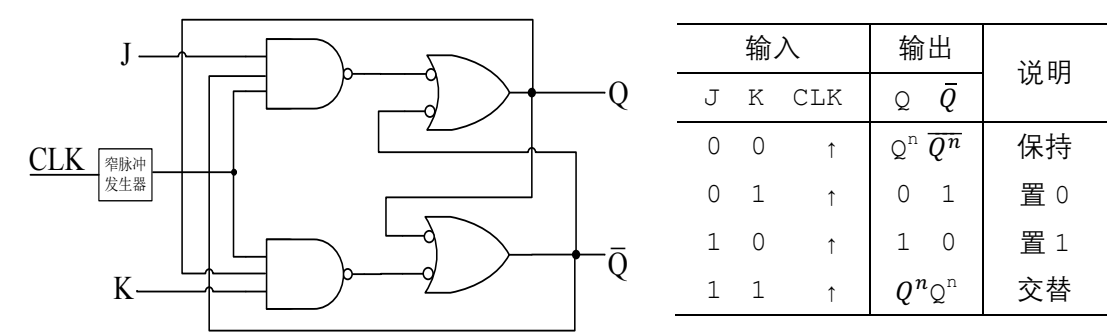


图 6.6.3 JK 触发器逻辑图及其功能表

本实验主要实现 D 触发器和 JK 触发器，同学们可以自行实现 SR 触发器，其中 D 触发器需要实现基本的 D 触发器以及同步复位 D 触发器和异步复位 D 触发器。电路图如上文所示。所谓同步复位，指的是同步复位信号就是指复位信号只在所需时钟边沿到来时才有效。所谓的异步复位，即无论时钟边沿到来与否，只要复位信号有效输出就会被复位。下面给出实验中所要建模模块的接口定义。

● 接口定义：

a) 同步复位 D 触发器

```
module Synchronous_D_FF(  
    input CLK,           //时钟信号，上升沿有效  
    input D,             //输入信号 D  
    input RST_n,         //复位信号，低电平有效
```

```

output reg Q1,    //输出信号 Q
output reg Q2     //输出信号  $\bar{Q}$ 
);

```

b) 异步复位 D 触发器

```

module Asynchronous_D_FF(
    input CLK,        //时钟信号, 上升沿有效
    input D,          //输入信号 D
    input RST_n,      //复位信号, 低电平有效
    output reg Q1,    //输出信号 Q
    output reg Q2     //输出信号  $\bar{Q}$ 
);

```

c) 异步复位 JK 触发器

```

module JK_FF(
    input CLK,        //时钟信号, 上升沿有效
    input J,          //输入信号 J
    input K,          //输入信号 K
    input RST_n,      //复位信号, 低电平有效
    output reg Q1,    //输出信号 Q
    output reg Q2     //输出信号  $\bar{Q}$ 
);

```

提示：上升沿触发使用 `always @ (posedge clk)`

● XDC 文件配置

a) D 触发器

变量	CLK	D	RST_n	Q1	Q2
N4 板上的管脚	BTNR (M17)	SW0 (J15)	BTNU (M18)	LD1 (H17)	LD2 (K15)

b) JK 触发器

变量	CLK	J	K	RST_n	Q1	Q2
N4 板上的管脚	BTNR (M17)	SW0 (J15)	SW1 (L16)	BTNU (M18)	LD1 (H17)	LD2 (K15)

2) PC 寄存器

PC 寄存器 (program counter) 是组成 CPU 的基本部件, 用来存放当前正在执行的指令, 包括指令的操作码和地址信息。我们知道, D 触发器可以用于存储比特信号。如果 D 为 1, 那么在时钟的上升沿, D 触发器的输出 Q 将变为 1。如果 D 为 0, 那么在时钟的上升沿, D 触发器的输出 Q 将变为 0。在实时数字系统中, 通常 D 触发器的时钟输入端始终有时钟信号输入。这就意味着在每个时钟的上升沿, 当前 D 值都将被锁存在 Q 中。在本实验主

利用 Verilog HDL 模块实例化基于 D 触发器模块实现一个 PC 寄存器。图 6.6.4 给出了所要建模的 PC 寄存器其功能图。

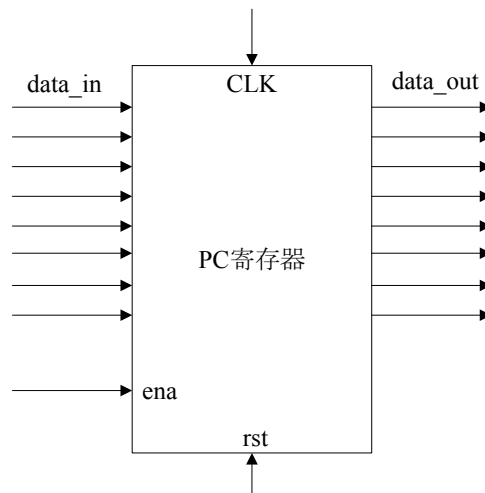


图 6.6.4 PC 寄存器功能图

● 接口定义：

```
module pcreg(
    input clk,    //1 位输入，寄存器时钟信号，上升沿时为 PC 寄存器赋值
    input rst,    //1 位输入，异步重置信号，高电平时将 PC 寄存器清零
                //注：当 ena 信号无效时，rst 也可以重置寄存器
    input ena,    //1 位输入，有效信号高电平时 PC 寄存器读入 data_in
                //的值，否则保持原有输出
    input [31:0] data_in, //32 位输入，输入数据将被存入寄存器内部
    output reg [31:0] data_out //32 位输出，工作时始终输出 PC
                //寄存器内部存储的值
);
```

4. 实验步骤

1. 新建 Vivado 工程，编写各个模块。
2. 用 ModelSim 仿真测试各模块。
3. 配置 XDC 文件，综合下板，并观察实验现象。
4. 按照要求书写实验报告。