

习题4 组合逻辑电路设计

20195633 李燕琴 计卓02

1、偶校验码生成器 (ParityCodeGenerator)

题目：设某数字系统的数据交换按半字节进行（传输数据为4位），为保证数据交换的正确性，请为该传输数据设计一个偶校验码生成器。

设计步骤：

1、功能实现

- Input: 4位（四位数字）
- Output: 1位（四位数字+偶校验码）
- 逻辑状态：
输出的前四位和输入相同；最后一位为偶校验码，且输入的4个数中，‘1’的个数为奇数个，则校验位为1，‘1’的个数为偶数个，则校验位为0。个数校验，用异或实现。

2、真值表（只展示了Even的真值）

Data[3]	Data[2]	Data[1]	Data[0]	Even
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

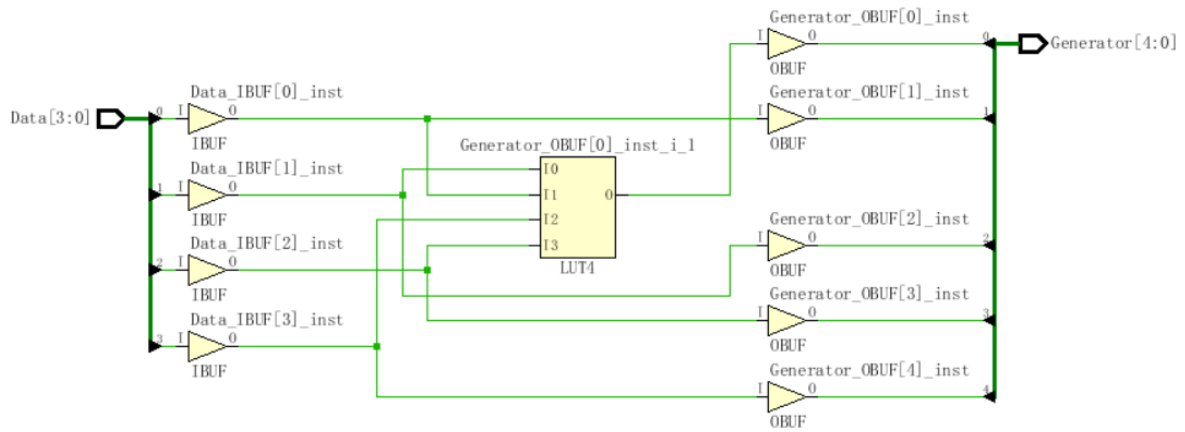
3、逻辑函数

$$Even = Data[3] \oplus Data[2] \oplus Data[1] \oplus Data[0]$$

$$Generator[0] = Even$$

$$Generator[k + 1] = Data[k], \quad k = 0, 1, 2, 3$$

4、逻辑控制电路



5、代码实现：

- 代码

//实现一：异或

even = ^Data;

//实现二：半加器

even = Data[3]+Data[2]+Data[1]+Data[0];

```
`timescale 1ns / 1ps

module ParityCodeGenerator(
    input [3:0] Data,
    output [3:0] Generator,
);
    reg even;
    reg [4:0] Generator;
    integer k;
    //    always@(*)
    //        begin
    //            even ^= Data;
    //            Generator[4] = Data[3];
    //            Generator[3] = Data[2];
    //            Generator[2] = Data[1];
    //            Generator[1] = Data[0];
    //            Generator[0] = even;
    //        end
    always@(*)
    begin
        even = ^Data;
        for(k=0; k<4; k=k+1)
            begin
                Generator[k+1] = Data[k];
            end
        Generator[0] = even;
    end
endmodule
```

- 约束引脚文件

```
set_property IOSTANDARD LVCMOS33 [get_ports {Data[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Data[2]}]
```

```

set_property IOSTANDARD LVCMOS33 [get_ports {Data[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Data[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Generator[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Generator[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Generator[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Generator[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Generator[0]}]
set_property PACKAGE_PIN V17 [get_ports {Data[0]}]
set_property PACKAGE_PIN V16 [get_ports {Data[1]}]
set_property PACKAGE_PIN W16 [get_ports {Data[2]}]
set_property PACKAGE_PIN W17 [get_ports {Data[3]}]
set_property PACKAGE_PIN U16 [get_ports {Generator[0]}]
set_property PACKAGE_PIN E19 [get_ports {Generator[1]}]
set_property PACKAGE_PIN U19 [get_ports {Generator[2]}]
set_property PACKAGE_PIN V19 [get_ports {Generator[3]}]
set_property PACKAGE_PIN W18 [get_ports {Generator[4]}]

```

2、命令启停器 (CommandStartStop)

题目：请为computer house中烧咖啡的机器老鼠设计一个命令启停器。提示：网关命令有效时，老鼠自动生成移动、咖啡注入、注入停止、开启火源、关闭火源等操作。(设计分析：老鼠移动有两个方向，一个是前行；一个是后退，所以网关命令中使用F表示前行；B表示后退。另外用I命令表示咖啡注入命令，用S表示咖啡注入停止命令。由TUON表示开启火源；而用TUOF表示关闭火源；A表示老鼠前行；C表示后退；D表示咖啡注入；E表示咖啡停止注入。)

设计步骤：

1、功能实现

- Input: 4位 (F、B、I、S)
- Output: 6位 (A、C、TUON、D、E、TUOF)
- 逻辑状态：

网关命令为真，执行该命令控制的操作；如F控制A，B控制C，I控制TUON和D，S控制E和TUOF（理解，咖啡注入命令下达后，老鼠会先后执行开启火源、注入咖啡两个动作，此处理解为同时执行这两个动作）；且命令的优先级别依次降低（理解，当F、B、I、S有两个以上的真值，则按照该顺序执行优先级最高的动作）。

2、真值表

F	B	I	S	A	C	TUON	D	E	TUOF
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0
0	0	1	1	0	0	1	1	0	0
0	1	0	0	0	1	0	0	0	0
0	1	0	1	0	1	0	0	0	0
0	1	1	0	0	1	0	0	0	0
0	1	1	1	0	1	0	0	0	0
1	0	0	0	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0

3、逻辑函数

$$A = F$$

$$C = \overline{F} \wedge B$$

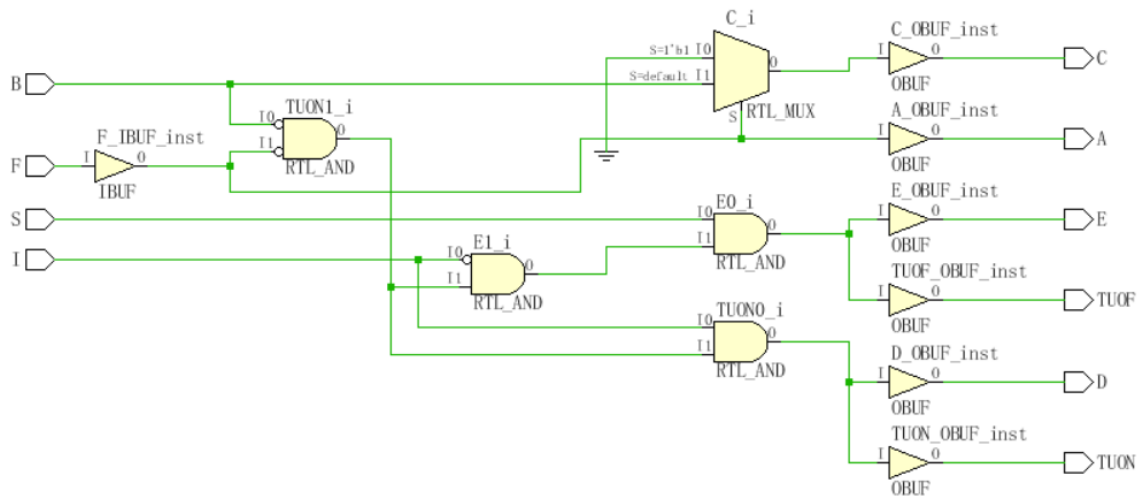
$$TUON = \overline{F} \wedge \overline{B} \wedge I$$

$$D = TUON$$

$$E = \overline{F} \wedge \overline{B} \wedge \overline{I} \wedge S$$

$$TUOF = E$$

4、逻辑控制电路



5、代码实现：

- 代码

```
`timescale 1ns / 1ps

module CommandStartStop(
    input F,          // 前行命令
    input B,          // 后退命令
    input I,          // 咖啡命令
    input S,          // 注入命令
    output reg A,      // 执行前行操作
    output reg C,      // 执行后退操作
    output reg TUON,   // 执行开火操作
    output reg D,      // 执行注入咖啡操作
    output reg E,      // 执行停止注入操作
    output reg TUOF    // 执行关火操作
);
always@(*)
begin
    A = 0;
    C = 0;
    TUON = 0;
    D = 0;
    E = 0;
    TUOF = 0;
    if(F==1) A=1;
    else if(B==1) C=1;
    else if(I==1)
        begin
            TUON = 1;
        end
end
```

```

        D = 1;
    end
    else if(S==1)
        begin
            TUOF = 1;
            E = 1;
        end
    end
endmodule

```

- 约束引脚文件

```

set_property IOSTANDARD LVCMOS33 [get_ports A]
set_property IOSTANDARD LVCMOS33 [get_ports B]
set_property IOSTANDARD LVCMOS33 [get_ports C]
set_property IOSTANDARD LVCMOS33 [get_ports D]
set_property IOSTANDARD LVCMOS33 [get_ports E]
set_property IOSTANDARD LVCMOS33 [get_ports F]
set_property IOSTANDARD LVCMOS33 [get_ports I]
set_property IOSTANDARD LVCMOS33 [get_ports S]
set_property IOSTANDARD LVCMOS33 [get_ports TUOF]
set_property IOSTANDARD LVCMOS33 [get_ports TUON]

set_property PACKAGE_PIN U16 [get_ports TUOF]
set_property PACKAGE_PIN E19 [get_ports E]
set_property PACKAGE_PIN U19 [get_ports D]
set_property PACKAGE_PIN V19 [get_ports TUON]
set_property PACKAGE_PIN W18 [get_ports C]
set_property PACKAGE_PIN U15 [get_ports A]

set_property PACKAGE_PIN V17 [get_ports S]
set_property PACKAGE_PIN V16 [get_ports I]
set_property PACKAGE_PIN W16 [get_ports B]
set_property PACKAGE_PIN W17 [get_ports F]

```