

1. 偶校验码生成器

偶校验码即为：数据中 1 的个数为偶数则校验码为 1，否则为 0

当输入 bin 时， \wedge bin 按位异或，偶数个 1 时输出为 0，否则为 1，再取反，则偶数个 1 时输出为 0，否则为 1。

代码

```
module oddcheck(bin, even_numbers);  
    input[3:0] bin;  
    output even_numbers;  
    assign even_numbers = ~(^bin);  
endmodule
```

电路图：



2. 命令启停器

方法一：用与门实现

ABCD 表示输出，将输出表示为输入的逻辑函数

$$\begin{aligned} A &= \overline{F} \overline{B} \overline{I} \overline{S} \\ C &= \overline{F} B \overline{I} \overline{S} T V O \overline{F} \\ D &= I \overline{S} \\ E &= \overline{I} S T V O N \end{aligned}$$

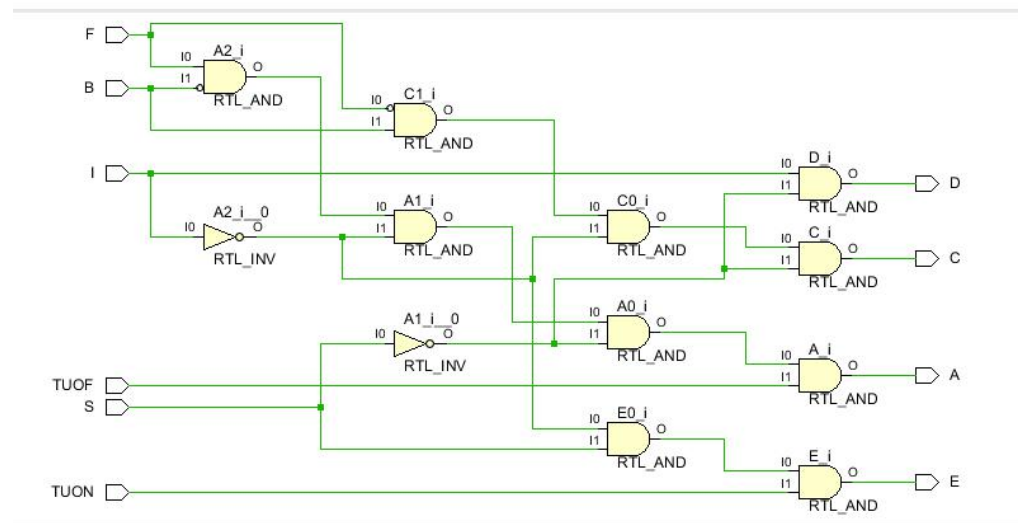
代码

```

module commander (F, B, I, S, TUON, TUOF, A, C, D, E);
    input F, B, I, S, TUON, TUOF;
    output A, C, D, E;
    and(A, F, ~B, ~I, ~S, TUOF);
    and(C, ~F, B, ~I, ~S);
    and(D, I, ~S);
    and(E, ~I, S, TUON);
endmodule

```

电路:



方法二:

用八选一多路选择器实现，用一个三位二进制数 **sel** 代表选择输入，输出命令分别为 **F,B,I,S,TUON,TUOF**,用 **out** 表示输出

代码:

```

12 |
13 | module commander(sel, F, B, I, S, TUON, TUOF, out);
14 |     input [2:0] sel;
15 |     input F, B, I, S, TUON, TUOF;
16 |     output reg out;
17 |     always@(*)
18 |     begin
19 |         case(sel)
20 |             3'b000: out = F;
21 |             3'b001: out = B;
22 |             3'b010: out = I;
23 |             3'b011: out = S;
24 |             3'b100: out = TUON;
25 |             3'b101: out = TUOF;
26 |             default: out = 1'b0;
27 |         endcase;
28 |     end;
29 | endmodule

```

电路：

