FPGA奇数分频

**数字ICer**

微信号 Studying\_Times

功能介绍 研究生在读，分享数字IC、FPGA、Linux、Python、UVM、SV、信号处理等学习笔记，快乐学习，一起进步。

**设计中尽量还是要避免使用自己计数分频得到的时钟，去使用厂家自带的分频IP（如Vivado中的clock wizard）。**

**偶数分频比较简单，这里略过。**

**对于不要求占空比为50%的奇数分频，也比较简单，直接模N计数，期间进行两次翻转就可以了。**

**这里重点介绍要求占空比为50%的奇数分频。**

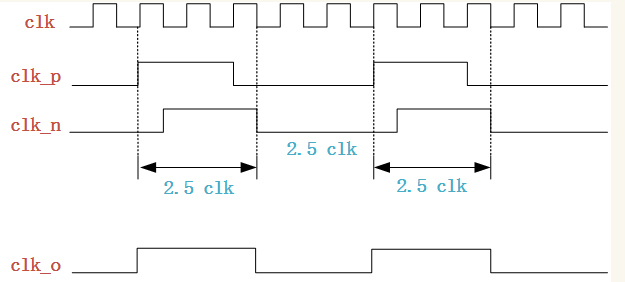
**步骤：**

　　1. 在时钟上升沿，进行模N计数，选定到某个值（比如选择1）时翻转，经过(N-1)/2 个时钟再进行翻转，产生一个临时时钟clk\_p；

　　2. 在时钟下降沿，进行模N计数，选定到某个值（和上升沿选定的值相同）时翻转，经过(N-1)/2 个时钟再进行翻转，产生一个临时时钟clk\_n；

　　3.  输出时钟clk\_o = clk\_p | clk\_n;

**图解：**（假如需要5分频）



**Verilog代码：**

module CLK\_DIV5(

input clk\_i,

input rst\_n,

output clk\_o

);

reg [2:0] cnt1,cnt2;

reg clk\_p,clk\_n;

always @(posedge clk\_i,negedge rst\_n)

if(!rst\_n) begin

cnt1 <= 3'b0;

clk\_p <= 1'b0;

end

else begin

if(cnt1 == 3'b100) begin

cnt1 <= 3'b0;

clk\_p <= clk\_p;

end

else begin

cnt1 <= cnt1 + 1'b1;

if(cnt1 == 3'b1 || cnt1 == 3'b11)

clk\_p <= ~clk\_p;

end

end

always @(negedge clk\_i,negedge rst\_n)

if(!rst\_n) begin

cnt2 <= 3'b0;

clk\_n <= 1'b0;

end

else begin

if(cnt2 == 3'b100) begin

cnt2 <= 3'b0;

clk\_n <= clk\_n;

end

else begin

cnt2 <= cnt2 + 1'b1;

if(cnt2 == 3'b1 || cnt2 == 3'b11)

clk\_n <= ~clk\_n;

end

end

assign clk\_o = clk\_p | clk\_n;

endmodule