

用 Verilog 实现基于 FPGA 的通用分频器

华北电力大学(北京)信息工程系 / 唐晓燕, 梁光胜, 王玮

在复杂数字逻辑电路设计中,经常会用到多个不同的时钟信号。介绍一种通用的分频器,可实现 2~256 之间的任意奇数、偶数、半整数分频。首先简要介绍了 FPGA 器件的特点和应用范围。接着介绍了通用分频器的基本原理和分类,并以分频比为奇数 7 和半整数 6.5 的分频器设计为例,介绍了在 Quartus II 开发软件下,利用 Verilog 硬件描述语言来设计数字逻辑电路的过程和方法。

在数字逻辑电路设计中,分频器是一种基本电路。我们常会遇到偶数分频、奇数分频、半整数分频等,在同一个设计中有时要求多种形式的分频。通常由计数器或计数器的级联构成各种形式的偶数分频和奇数分频,实现较为简单。但对半整数分频实现较为困难。但在某些场合下,时钟源与所需的频率不成整数倍关系,此时可采用小数分频器进行分频。例如:时钟源信号为 130MHz,而电路中需要产生一个 20MHz 的时钟信号,其分频比为 6.5,因此根据不同设计的需要,本文利用 Verilog 硬件描述语言,通过 MAX+plus II 开发平台,使用 Altera 公司的 FLEX 系列 EPF10K10LC84-3 型 FPGA,设计了一种能够满足上述各种要求的较为通用的分频器。

基于查找表(LUT)的FPGA的结构特点

查找表(Look-Up-Table)简称为LUT,LUT本质上就是一个RAM。目前FPGA中多使用4输入的LUT,所以每一个LUT可以看成是一个有4位地址线的16x1的RAM。当用户通过原理图或HDL语言描述了一个逻辑电路以后,PLD/FPGA开发软件会自动计算逻辑电路的所有可能的结果,并把结果事先写入RAM,这样,每输入一个信号进行逻辑运算就等于输入一个地址进行查表,找出地址对应的内容,然后输出即可。由于LUT主要适合SRAM工艺生产,所以目前大部分FPGA都是

基于SRAM工艺的,而SRAM工艺的芯片在掉电后信息就会丢失,一定需要外加一片专用配置芯片,在上电的时候,由这个专用配置芯片把数据加载到FPGA中,然后FPGA就可以正常工作,由于配置时间很短,不会影响系统正常工作。也有少数FPGA采用反熔丝或Flash工艺,对这种FPGA,就不需要外加专用的配置芯片。

FPGA(Field programmable Gates Array,现场可编程门阵列)都是可编程逻辑器件,它们是在PAL、GAL等逻辑器件基础上发展起来的。同以往的PAL、GAL相比,FPGA/CPLD的规模比较大,适合于时序、组合等逻辑电路的应用。它可以替代几十甚至上百块通用IC芯片。这种芯片具有可编程和实现方案容易改动等特点。由于芯片内部硬件连接关系的描述可以存放在磁盘、ROM、PROM、或EPROM中,因而在可编程门阵列芯片及外围电路保持不动的情况下,换一块EPROM芯片,就能实现一种新的功能。它具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无需测试、质量稳定以及实时在检验等优点,因此,可广泛应用于产品的原理设计和产品生产之中。几乎所有应用门阵列、PLD和中小规模通用数字集成电路的场合均可应用FPGA和CPLD器件。在现代电子系统中,数字系统所占的比例越来越大。系统发展的趋势是数字化和集成化,而FPGA作为可编程ASIC(专用集成电路)器件,它将在数字逻辑系统中发挥越来越重要的作用。

通用分频器基本原理

整数分频包括偶数分频和奇数分频,对于偶数N分频,通常是由模N/2计数器实现一个占空比为1:1的N分频器,分频输出信号模N/2自动取反。对于奇数N分频,上述方法就不适用了,而是由模N计数器实现非等占空比的奇数N分频

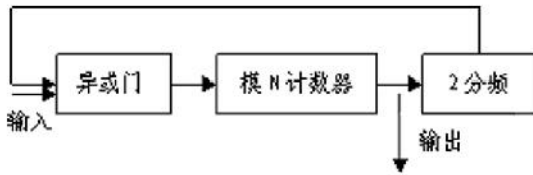


图1：半整数分频器电路组成

器，分频输出信号取得是模N计数中的某一位（不同N值范围会选不同位）。这种方法同样适用于偶数N分频，但占空比不总是1:1，只有2的n次方的偶数（如4、8、16等）分频占空比才是1:1。这种方法对于奇数、偶数具有通用性。

半整数分频器也是在这种方法基础上实现的。除了一个模N计数器，还需要一个异或模块和一个2分频模块。半整数分频器原理如图1所示：

半整数分频器设计思想：通过异或门和2分频模块组成一个改变输入频率的脉冲添加电路，也就是说N-0.5个输入信号周期内产生了N个计数脉冲，即输入信号其中的一个含一个脉冲的周期变为含两个脉冲的周期。而这一改变正是输入频率与2分频输出异或的结果。由2分频输出决定一个周期产生两个脉冲有两种方式：当一个输入信号来一个脉冲（前半周期）时，2分频输出变为‘1’，clk_in取反，后半周期就会产生一个脉冲；2分频输出由‘1’变为‘0’时，clk_in刚把一个周期（前半周期）内低电平变为高电平产生一个脉冲，而后半周期的脉冲与‘0’异或不变。从而实现N-0.5分频。

要实现奇数、偶数、半整数通用分频器只需再加一个控制选择信号sel。当sel=‘1’时，clk_in与2分频输出异或，实现半整数分频；当sel=‘0’时，只选通clk_in，实现整数分频。通用分频器原理如图2所示：



图2：通用分频器电路组成

Verilog语言的实现

本设计采用层次化的设计方法，首先设计通用分频器中各组成电路元件，然后通过元件例化的方法，调用各元件，实现通用分频器。

1、选择异或门模块 half_select :

```

module half_select(sel,a,b,c);
    output c;
    input sel,a,b;
    xor u1(w,a,b);
    assign c= sel? w:a;
    (当sel='1'时,clk_in与2分频输出异或,实现半整数分频;当sel='0'时,只选通clk_in,实现整数分频。)
endmodule

```

2、模N计数器 counter_n :

实现参数化设计N可取2~256，也可增加count位数使N可取更大的值。以N=7为例通过设置sel分别实现奇数7分频和半整数6.5分频。

```

module counter_n(reset,en,clk_in,clk_out,count);
    parameter N=7;
    input reset,en,clk_in;
    output clk_out;
    output [7:0] count;
    reg clk_out;
    reg [7:0] count;
    always @(posedge clk_in)
    begin
        if (reset)
            begin
                count[7:0]=0;
            end
        else if (en)
            begin
                if (count==(N-1))
                    count=0;
                else
                    count=count + 1;
            end
        end
    always
        begin
            if (N<=2)
                clk_out=count[0];

```

```

else if (N<=4)
clk_out=count[1];
else if (N<=8)
clk_out=count[2];
else if (N<=16)
clk_out=count[3];
else if (N<=32)
clk_out=count[4];
else if (N<=64)
clk_out=count[5];
else if (N<=128)
clk_out=count[6];
else if (N<=256)

clk_out=count[7];
end
endmodule
    
```

3、2分频模块 fenpin_2 :

```

module fenpin_2(clk_in,clk_out);
input clk_in;
output clk_out;
reg clk_out;
always @(posedge clk_in)
begin
    clk_out=~clk_out;
end
endmodule
    
```

4、最后通过元件例化的方法，调用各元件，实现通用分频器最高层设计：

```

module fenpin(reset,en,sel,clk_in,clk_out,count);
input reset,en,sel,clk_in;
output clk_out;
output[3:0] count;
half_selectu1(sel,clk_in,b,c);(调用half_select 元件例化)
counter_nu2(reset,en,c,clk_out,count);(调用counter_n,
元件例化)
fenpin_2 u3(clk_out,b);(调用fenpin_2 元件例化)
endmodule
    
```

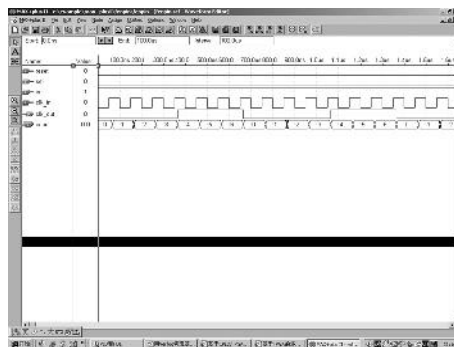


图3 : 7分频仿真结果

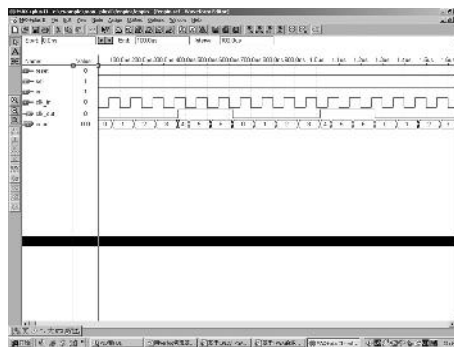


图4 : 6.5分频仿真结果

仿真结果及硬件电路测试

当 sel= ' 0 ' , N=7 时，实现奇数 7 分频。如图 3 所示
 当 sel= ' 1 ' , N=7 时，实现半整数 6.5 分频。如图 4 所示
 由此可见，只要改变异或门选通控制 sel 和 N 的值就可实现 2 ~ 256 之间的任意奇数、偶数、半整数分频。本设计在 Altera 公司的 FLEX 系列 EPF10K10LC84-3 型 FPGA 构成的测试平台上测试通过，性能良好。

表1 半整数 6.5 分频器适配分析结果:

选用器件	I/O 延迟时间	使用引脚数	工作频率
EPF10K10LC84-3	15.7ns	9/84 (10.7%)	39.52MHz

结语

本文旨在介绍一种进行 FPGA 开发时，所需多种分频的实现方法，如果设计所需分频形式较多，可以直接利用本设计，通过对程序的稍微改动以满足自己设计的要求。如果设计中需要分频形式较少，可以利用本设计部分程序，以节省资源。