



# verilog 奇数，偶数分频逻辑

WILLIAM 2022-02-16 FPGA 2 COMMENTS

在fpga 设计中，我们有时需要对输入的时钟进行分频。例如：1/2，1/3，1/4，1/5 等等。在偶数分频时，分频逻辑比较容易实现，在奇数分频时，fpga 设计相对麻烦一些。

偶数分频逻辑设计：

even\_div.v

```
1.  module even_div #
2.  (
3.      parameter [31:0] N = 2
4.  )
5.  (
6.      input  clk,
7.      output clk_out,
8.      input  reset
9.  );
10.
11.
12.
13.  reg [31:0] cnt;
14.  always @(posedge clk or posedge reset)
15.  begin
16.      if(reset)
17.          cnt <= 0;
18.      else if(cnt == N - 1)
19.          cnt <= 0;
20.      else cnt <= cnt + 1;
21.  end
22.
23.
24.  reg clk_reg;
```

其他操作

管理站点

注销

Search ...



本分类下文章

Quartus 工具 In-System Memory Content editor 使用

图像采集与传输系统（摄像头控制模块）

模拟电路源极跟随器（源随器）

Intel MAX10 FPGA 设计完整性指南

C语言中的内存管理如何工作？

米勒 CMOS运算跨导放大器的设计

地址解析协议（ARP）

```

25.
26.  always @(posedge clk or posedge reset)
27.  begin
28.      if(reset)
29.          clk_reg <= 0;
30.      else if(cnt == N/2 - 1)
31.          clk_reg <= ~clk_reg;
32.      else if(cnt == N - 1)
33.          clk_reg <= ~clk_reg;
34.  end
35.
36.  assign clk_out = clk_reg;
37.
38.  endmodule

```

偶数分频相对比较简单，程序中使用cnt 作为计数器，当计数器计数到分频值的一半时（N/2），clk 反转；当计数器计数到分频值（N）时，clk 再次反转。

odd\_div.v

```

1.  module  odd_div #
2.  (
3.      parameter [31:0] N = 3
4.  )
5.  (
6.      input  clk,
7.      output clk_out,
8.      input  reset
9.  );
10.
11.
12.
13.  reg [31:0] cnt_p , cnt_n;
14.  reg        clk_p , clk_n;
15.
16.
17.
18.  localparam BAUD_HALF = (N-1)/2 - 1;
19.  localparam BAUD_TICK = N - 1;
20.
21.
22.

```

RISC-V指令集讲解

(2) I-Type整数寄存器-立即数指令

RISC-V 总线和流水线 (4) RISC-V

CPU流水线介绍

Vivado下利用ROM显示数码管实验课

无线信号多径衰落

RISC-V-privileged pdf英文版

网络协议基础什么是网络协议？

Python Lambda函数

数字集成电路设计中的标准单元库介绍

以太网UDP 实验 (eth\_udp\_tx.v)

低于1V的基准电压和其它应用

数字集成电路设计概述

图像采集与传输系统的设计

I2c 协议实现 EEPROM 仿真

ZYNQ 系统中，自建axi-lite 寄存器模式下的gpio模块

指针

RISC-V LSU, SRAM, GPIO模块

```
23. always @(posedge clk or posedge reset)
24. begin
25.     if(reset)
26.         cnt_p <= 4'd0;
27.     else if(cnt_p == BAUD_TICK)
28.         cnt_p <= 0;
29.     else
30.         cnt_p <= cnt_p + 1'b1;
31. end
32.
33.
34. always @(negedge clk or posedge reset)
35. begin
36.     if(reset)
37.         cnt_n <= 4'd0;
38.     else if(cnt_n == BAUD_TICK)
39.         cnt_n <= 0;
40.     else
41.         cnt_n <= cnt_n + 1'b1;
42. end
43.
44. //clk_p
45. always @(posedge clk or posedge reset)
46. begin
47.     if(reset)
48.         clk_p <= 1;
49.     else if(cnt_p == BAUD_HALF)
50.         clk_p <= ~clk_p;
51.     else if(cnt_p == BAUD_TICK)
52.         clk_p <= ~clk_p;
53. end
54.
55. //clk_n
56. always @(negedge clk or posedge reset)
57. begin
58.     if(reset)
59.         clk_n <= 1;
60.     else if(cnt_n == BAUD_HALF)
61.         clk_n <= ~clk_n;
62.     else if(cnt_n == BAUD_TICK)
63.         clk_n <= ~clk_n;
64. end
65.
66.
67. assign clk_out = clk_n | clk_p;
68.
```

(1) exu\_lsu模块

SPI 通讯协议 (5)  
SPI FLASH (verilog)  
工程解析  
(spi\_cmd.v)

Python语言编程应用教材

C语言的文件管理

跨导放大器的串联-串联反馈

下载后安装Oracle VM VirtualBox

什么是开源共享精神?

AXI总线基础

近期文章

Sytem Verilog 教学(视频)

System Verilog 教材

系统验证基础-OVM与UVM

第六节 三极管的特性曲线及参数

第五节 三极管电流分配关系及电流放大系数

第四节 半导体三极管的原理

无线多径衰落系数

69. endmodule

奇数分频相对比较麻烦， 步骤如下：

- 1) 需要两个计数器：clk 上升沿计数器； clk 下降沿计数器。这两个计数器实现N计数。
- 2) 分别在clk 的上升沿和下降沿 实现 clk\_p, clk\_n。当cnt\_x == N/2 – 0.5 时， clk\_x 反转； 当cnt\_x == N 时， clk\_x 再次反转； 这样生成的时钟clk\_x 是一个占空比不等的时钟（高电平 – 低电平 = 1）即要么高电平多0.5， 或者低电平多0.5 个时钟周期。
- 3) clk\_out = clk\_p | clk\_n; 平衡占空比不等的现象， 通过使用 或操作， 实现%50 、 %50 占空比的时钟。

仿真程序：

```
1. `timescale 1ns / 1ps
2.
3. module tb
4. (
5. );
6.
7. parameter PERIOD = 20 ; //50Mhz
8.
9. reg clk;
10. reg reset = 1;
11. initial
12. begin
13.     clk = 0;
14.     #(PERIOD/2);
15.
16.     forever
17.         #(PERIOD/2) clk = ~clk;
18. end
19.
20. initial
21. begin
22.     reset = 1;
23.     @(posedge clk);
24.     reset = 0;
25.
```

的统计特性

无线信号多径衰落

无线信号建模-多径效应

第三节 半导体二极管

分类

选择分类

本周热门文章

Verilog语言编程与FPGA应用教材

RISC-V 教学教案

芯片天地-教材与教案 (文章)

芯片天地-网络视频学院 (视频)

RISC-V CSR寄存器  
(1) CSR简介和CSR指令

Python 查找list平均值

FPGA与Verilog 教学视频

Verilog 文件操作-\$fdisplay, \$fwrite, \$fstrobe, \$fmonitor

互联网网线及接口的基本知识

```

26.         #800000;
27.         $stop;
28.     end
29.
30.     wire clk_out1;
31.     odd_div #
32.     (
33.         .N (7)
34.     )
35.     odd_div_inst
36.     (
37.         .clk      (clk),
38.         .clk_out   (clk_out1),
39.         .reset     (reset)
40.     );
41.
42.
43.
44.     wire clk_out2;
45.     even_div #
46.     (
47.         .N (4)
48.     )
49.     even_div_inst
50.     (
51.         .clk      (clk),
52.         .clk_out   (clk_out2),
53.         .reset     (reset)
54.     );
55.
56.
57. endmodule

```

Verilog 中not ~ !  
的区别

数字集成电路

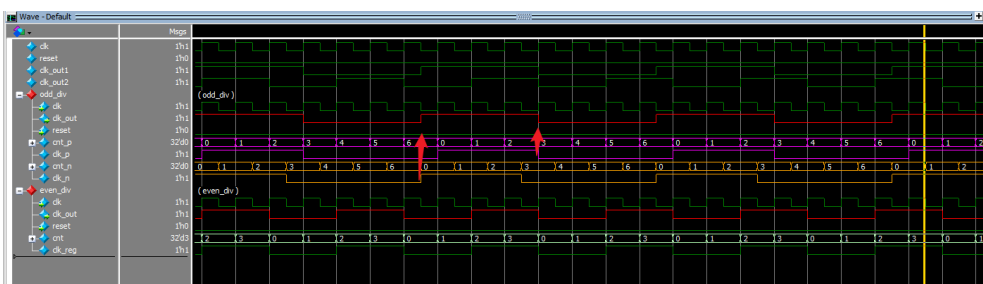
RISC-V指令集讲解  
(2) I-Type整数寄存器-立即数指令

Verilog 文件操作-\$fopen, \$fclose

电流镜 ( Current Mirror)

RISC-V 指令集手册  
中文2.1版 (User-Level ISA)

仿真波形:



从仿真波形上可以看出, 通过或操作实现 %50 , %50 的占空比时钟, 从而实现计数分频。 偶数分频相对简单一些, 只要通过计数分频即可。

Posted in [FPGA](#), [FPGA 教材教案](#), [Verilog](#), [教材与教案](#), [文章](#)

Tagged [偶数分频逻辑](#), [奇数分频逻辑](#)

## 发表回复

以 Byron 的身份登录。 [编辑您的个人资料](#)。 [注销?](#) 必填项已用\*标注

评论 \*

☒如果有人回复我的评论，请通过电子邮件通知我。

发表评论

## 相关链接

[系统验证基础-OVM与UVM](#)

[第五节 三极管电流分配关系及电流放大系数](#)

[第四节 半导体三极管的原理](#)

[无线信号多径衰落](#)

[无线信号建模-多径效应](#)

[第三节 半导体二极管](#)

[第二节 PN结原理及特性](#)

[第一节 半导体材料及特性](#)

[第五章，第六节习题答案](#)

## 第二章 半导体材料与晶体管

### 第五节 上拉、下拉电阻对电平的影响

### 第六节 分立器件TTL门电路

今日热门文章	近期评论	本月热门文章	本站热门文章
Verilog语言编程与FPGA应用教材		Verilog语言编程与FPGA应用教材	Verilog语言编程与FPGA应用教材
RISC-V 教学教案		FPGA与Verilog 教学视频	FPGA与Verilog 教学视频
FPGA与Verilog 教学视频		RISC-V 教学教案	RISC-V 教学教案
芯片天地-教材与教案（文章）		Python 查找list平均值	芯片天地-教材与教案（文章）
RISC-V CSR寄存器（1）CSR简介和CSR指令		芯片天地-教材与教案（文章）	RISC-V 教学教案-视频
芯片天地-网络视频学院（视频）		芯片天地-网络视频学院（视频）	芯片天地-网络视频学院（视频）
Verilog 乘法器的设计及仿真注意事项		Verilog 文件操作-\$fdisplay, \$fwrite, \$fstrobe, \$fmonitor	电流镜 ( Current Mirror)
Python 查找list平均值		RISC-V CSR寄存器（1）CSR简介和CSR指令	Python 查找list平均值
System Verilog 教材		Verilog 文件操作-\$fgetc, \$fgets, \$fscanf, \$fread	Verilog 文件操作-\$fgetc, \$fgets, \$fscanf, \$fread
Verilog 文件操作-\$fdisplay, \$fwrite, \$fstrobe, \$fmonitor		Verilog 文件操作-\$fopen, \$fclose	Verilog 文件操作-\$fdisplay, \$fwrite, \$fstrobe, \$fmonitor
SOC 教学教案		电流镜 ( Current Mirror)	RISC-V CSR寄存器（1）CSR简介和CSR指令
Verilog 中not ~ ! 的区别		第三节 卡诺图原理及构成	RISC-V 指令集手册中文2.1版 (User-Level ISA)
Vivado DDR3 控制器生成步骤（3）		Verilog 中not ~ ! 的区别	Verilog 文件操作-\$fopen, \$fclose
Vivado软件的使用		RISC-V 指令集手册中文2.1版 (User-Level ISA)	Python time.sleep() 为代码添加延迟
数字电路逻辑综合 (Synopsys Design Compiler)		IO接口基础知识	SOC 教学教案



Copyright © 2023 芯片天地

Verilog always过程

CMOS模拟集成电路设计教材

Verilog 文件操

数字集成电路设计中的标准单元库介绍

作-\$fseek, \$ftell, \$feof

RISC-V指令集讲解

I2C接口基础知识一

(2) I-Type整数寄存器-立即数指令