# 实验三 计数器设计

## 实验目的

（1）加深对计数器工作原理的理解，掌握计数器的Verilog实现方法；

（2）了解计数器的应用场景；

（3）进一步掌握使用Verilog HDL实现时序逻辑电路的方法；

（4）熟悉时序电路的仿真调试方法。

# 1 实验题目一：流水灯

## 1.1 实验内容

基于计数器，设计分频模块和流水灯模块，实现时间间隔为1s的硬件流水灯，其顶层模块如图1-1所示。



图1-1 流水灯顶层模块图

如图1-1所示，顶层模块内部包含分频模块和流水灯模块2个子模块。分频模块将100MHz的输入时钟降频至1Hz输出，并以此驱动流水灯模块输出时间间隔为1s的流水灯信号。

本题要求：

（1）使用拨码开关SW0作为异步复位信号rst\_n\_i，且当SW0为0时，led\_o和模块内部计数器的值均被复位；

（2）顶层模块的输入时钟需连接到EGO1的100MHz晶振时钟源（P17引脚）；

（3）使用拨码开关SW1作为同步使能信号，且当SW1为1时，流水灯正常工作；当SW1为0时，流水灯暂停工作；

（4）流水灯输出信号需连接到EGO1开发板的LED7-LED0。

引脚约束详情请同学们自行查阅[EGO1的用户手册](https://gitee.com/hitsz-datasci/course-diglogic/blob/master/stupkt/EGo1%E8%B5%84%E6%96%99.zip)。

## 1.2 实验原理

### 1.2.1 基于计数器的时钟分频

EGO1开发板上有一个频率为100MHz的晶振时钟源，输出的时钟信号与FPGA的全局时钟输入引脚P17相连。该时钟频率较高，人眼无法辨别其变化。因此，要实现时间间隔为1s的流水灯，首先需要对该时钟信号进行降频/分频处理，从而得到频率为1Hz的时钟信号。

分频电路的模块图如图1-2所示。



图1-2 分频模块图

图1-2所示的分频模块接收100MHz的时钟作为输入，对其进行分频后输出1Hz的时钟信号。当异步复位信号rst\_n有效时，输出为0。分频模块的接口信号定义如表1-1所示。

表1-1 分频模块接口信号定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位宽 | 属性 | 功能 |
| rst\_n\_i | 1 | 输入 | 异步复位，低电平有效 |
| clk\_i | 1 | 输入 | 时钟信号，频率为100MHz |
| clk\_o | 1 | 输出 | 分频后的时钟信号，频率为1Hz |

FPGA开发中的分频方式一般有两种：一是通过计数器实现分频，二是使用Vivado库中的“Clocking Wizard”IP核（Intellectual Property Core）。IP核是一种可重用模块，类似软件开发中的函数库，可以直接调用。但Vivado时钟IP核的分频倍数有限，无法将100MHz的时钟降频为1Hz。为此，本实验题目需要同学们使用计数器来实现分频。

计数器的设计应当考虑三个要素：**加1条件**、**结束条件**和**触发条件**。加1条件是计数器执行加1的条件。结束条件是计数器结束计数的条件。计数器在工作时，通常以0作为计数初值。一轮计数结束后，计数值通常也回到0。触发条件是指计数器计数到某一数值时，触发某个信号使其有效。

Verilog实现计数器的代码框架如图1-3所示。具体实现时，根据应用场景填写相应的加1条件、结束条件和触发条件即可。

wire rst = ??;

always @(posedge clk\_i or posedge rst) begin

    if (rst == 1'b1)

        cnt <= 'h0;

    else if (/\*加1条件\*/)

        cnt <= /\*结束条件\*/ ? 'h0 : cnt + 'h1;

    else

        cnt <= cnt;

end

assign out = (/\*触发条件\*/) && (cnt == ??);

图1-3 计数器代码框架图

要用Verilog实现分频模块，需要先定义一个reg变量，用于对接收到的高频时钟的上升沿个数进行计数，并在计到一定数值时对计数器清零，同时翻转某一位输出信号（该输出信号不是触发器的输出端），得到降频的时钟信号。

例如，基于上述原理，结合图1-3所示的计数器代码框架，可实现如图1-4所示的4分频电路。

reg [1:0] cnt;

always @(posedge clk\_i or posedge rst) begin

    if (rst == 1'b1) begin

        cnt <= 'h0;

    end else

        cnt <= (cnt >= 'h4) ? 'h0 : cnt + 'h1;

end

wire clk\_4div = !rst & (cnt < 'h2);

图1-4 4分频电路

上述4分频电路的波形如图1-5所示。

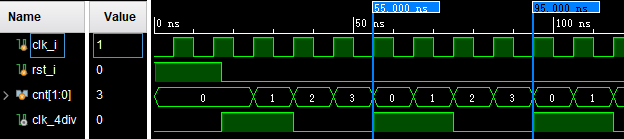


图1-5 4分频电路波形图

在图1-5中，计数器cnt不断地进行模4计数。当cnt的值小于2时输出高电平，否则输出低电平。最终得到原始时钟信号clk\_i的4分频信号clk\_4div——若clk\_i频率为100MHz，则clk\_4div的频率是25MHz。

请思考输出频率与输入频率、计数器上限值是什么关系？（仅考虑偶数倍分频与占空比是50%的分频情况。）

请同学们参照上述4分频电路的例子，基于图1-3所示的计数器代码框架，设计如图1-2所示的分频模块，将EGO1的100MHz晶振时钟源降频至1Hz。

### 1.2.2 基于计数器的流水灯

流水灯，即所有的LED灯逐个依次点亮，看起来就像是LED灯在流动。假设有LED7-LED0，则有：LED7亮1秒后熄灭->LED6亮1秒后熄灭->LED5亮1秒后熄灭->……->LED0亮1秒后熄灭->LED7亮1秒后熄灭->……。现在，让我们只观察LED7，可以发现LED7每隔7秒亮一次，而其余LED亦是如此。画出它们的波形，如图1-6所示。

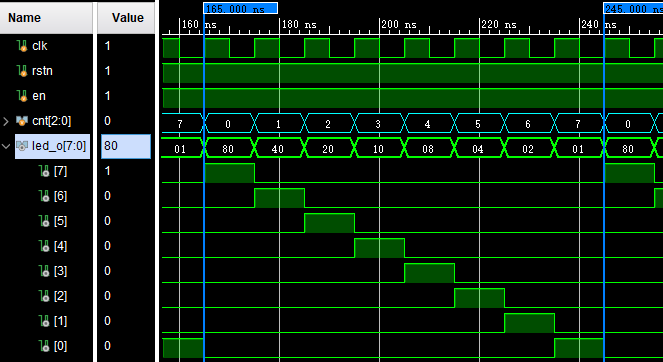


图1-6 流水灯波形图

请同学们基于图1-3所示的计数器代码框架，设计如图1-7所示的流水灯模块。



图1-7 流水灯模块图

提示：在图1-3所示的计数器代码框架中，仅有1个输出信号，因此只需考虑1个触发条件。而流水灯有8个输出信号，因此需要考虑8个触发条件。

流水灯模块的接口信号定义如表1-2所示。

表1-2 流水灯模块接口信号定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位宽 | 属性 | 功能 |
| rst\_n\_i | 1 | 输入 | 异步复位，低电平有效 |
| clk\_i | 1 | 输入 | 时钟信号，频率为1Hz |
| en\_i | 1 | 输入 | 同步使能信号，高电平有效 |
| led\_o | 8 | 输出 | 流水灯信号 |

## 1.3 工程文件要求

创建Ex\_3工程，完成flowLED\_top.v（顶层模块）、divider.v（分频模块）和flowLED.v（流水灯模块）的编写，并创建仿真文件和约束文件，最终下板验证。

## 1.4 仿真注意事项

### 1.4.1 延长仿真时间

将100MHz的频率分频到10Hz，输入时钟需要经历5000\_000次上升沿，输出才会改变一次。仿真的时候假如时钟周期是1ns，得5000us之后输出才会变化。而Vivado默认仿真时间是1000ns，显然不够。系统的默认仿真时间可以修改，点击左侧Flow Navigator下面的“PROJECT MANAGER”->“Settings”，在弹出的界面中选择左侧“Project Settings”->“Simulation”，修改xsim.simulate.runtime参数。修改全局的仿真时间显然不够灵活，这里使用Run For功能，如图1-8所示，先设置好运行时间和单位，再点击Run For，仿真器就会在当前的基础上再运行指定时间长度。

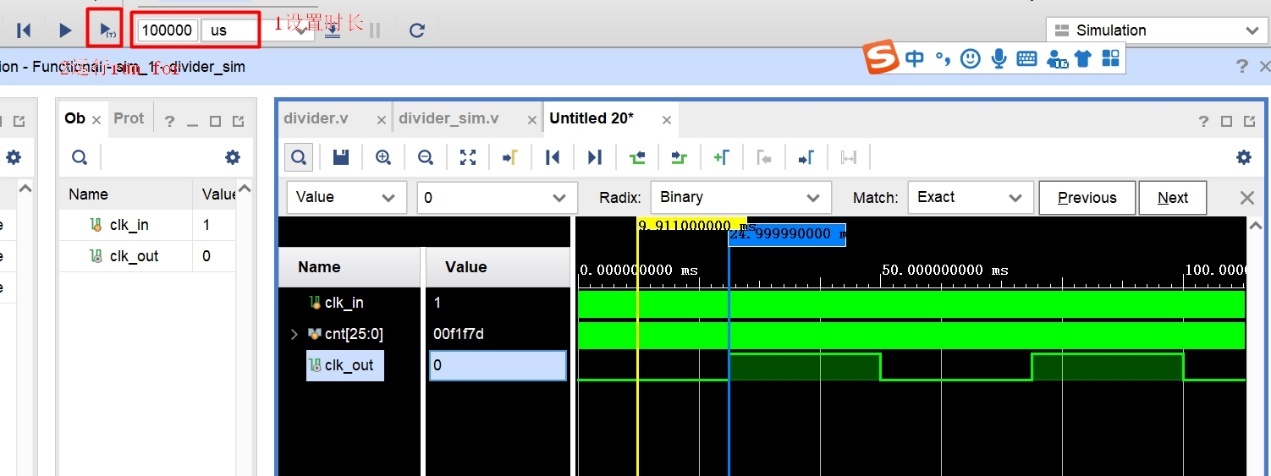


图1-8 仿真时间延长

### 1.4.2 波形仿真和测量

分频器分频精度的测量：用仿真视图上方的时钟沿工具（transition）选中输出时钟clk\_out的两个相邻的上升沿，注意得首先用鼠标选中clk\_out信号，再用工具进行标示。比较两个标尺时间之间的间隔是否为准确的1000ms（即频率是否是1Hz）？如果不是，请分析和检查所设计的分频模块是否正确。

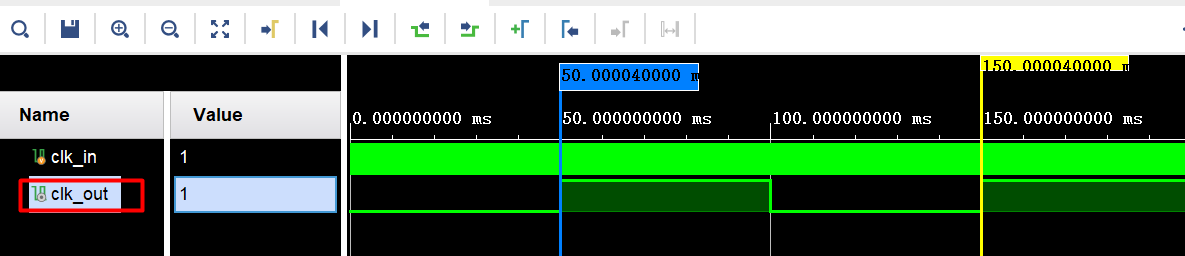


图1-9 分频器仿真波形

# 2 实验题目二：格雷码计数器

## 2.1 实验内容

基于计数器设计能循环输出4位格雷码的计数器，其电路模块如图2-1所示。



图2-1 格雷码计数器模块图

格雷码计数模块的接口信号定义如表2-1所示

表2-1 格雷码计数器模块接口信号定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位宽 | 属性 | 功能 |
| rst\_n\_i | 1 | 输入 | 异步复位，低电平有效 |
| clk\_i | 1 | 输入 | 时钟信号，频率为100MHz |
| en\_i | 1 | 输入 | 同步使能信号，高电平有效 |
| gray\_o | 4 | 输出 | 格雷码输出信号 |

本题只需仿真，不作下板要求。

本题对仿真的要求：

（1）除了表2-1所示的信号外，还要求观察内部计数器的值，如图2-2所示；



图2-2 格雷码计数器仿真信号

（2）仿真波形中的内部计数器、格雷码输出信号需设置成二进制格式显示；

（3）仿真需涵盖rst\_n\_i有效、rst\_n\_i无效、en\_i有效和en\_i无效4种情况。

## 2.2 实验原理

格雷码的特点是从一个数变为相邻的一个数时，只有一个数据位发生跳变，由于这种特点，可以避免二进制编码计数组合电路中出现的亚稳态。格雷码常用于通信，FIFO或者RAM地址寻址计数器中。十进制、二进制和格雷码的对应关系如表2-2所示。

表2-2 格雷码和十进制、二进制对应关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 十进制 | 二进制 | 格雷码 | 十进制 | 二进制 | 格雷码 |
| 0 | 0000 | 0000 | 8 | 1000 | 1100 |
| 1 | 0001 | 0001 | 9 | 1001 | 1101 |
| 2 | 0010 | 0011 | 10 | 1010 | 1111 |
| 3 | 0011 | 0010 | 11 | 1011 | 1110 |
| 4 | 0100 | 0110 | 12 | 1100 | 1010 |
| 5 | 0101 | 0111 | 13 | 1101 | 1011 |
| 6 | 0110 | 0101 | 14 | 1110 | 1001 |
| 7 | 0111 | 0100 | 15 | 1111 | 1000 |

**格雷码和二进制码相互转换的规则如下：**

格雷码转二进制是从左边第二位起，将每位与左边一位二进制码的值异或，作为该位二进制码后的值（最左边一位依然不变）。伪代码描述为：

b[n] = g[n]， // g、b分别对应n位的格雷码和二进制码

b[i] = g[i] xor b[i+1] (i∈N, n-1≥i≥0)。

二进制转格雷码从最右边一位起，依次将每一位与左边一位异或（XOR），作为对应格雷码该位的值，最左边一位不变。伪代码描述为：

assign gray\_value = binary\_value ^ (binary\_value>>1);

格雷码计数的实现方式有多种，简单的方法有：根据转换规则将二进制计数输出转为格雷码格式输出，即计数器复用，只需增加一位输出即可。鼓励同学们自行尝试多种方式实现。

## 2.3 工程文件要求

工程文件名称：greyCounter； 主模块文件名：gcounter.v；

仿真文件文件名：gcounter\_sim.v； 约束文件文件名：gcounter.xdc。

## 2.4 仿真调试技巧

### 2.4.1 仿真工具快捷操作

**（1）工具栏仿真功能操作**

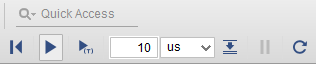


图2-3 工具栏仿真功能选项

从左到右的功能分别是：

* Restart：清空波形数据，从0时刻开始，需要再执行Run For；
* Run All：一直运行仿真到处理完所有event或手动停止仿真；
* Run For：按照设定的时间运行仿真，每点击一次都在之前的基础上再运行指定时长；
* Step：运行仿真直到下一个HDL状态；
* Break：暂停仿真运行；
* Relaunch Simulation：重新编译仿真源文件且restart仿真，当修改了源代码并且保存了文件后，只需要Relaunch即可，而不必关闭仿真再重新打开运行。

**（2）熟练使用Zoom Fit（最适波形）、波形缩小、放大功能**

一般先点击最适波形，准备大范围前后移动时，再点击波形上方工具条中的 缩小波形至合适程度，然后再拖动波形下方的滚动条至想观察的时刻附近，单击波形将游标落在这一时刻处，再点击波形上方工具条中的放大波形至你可以观察清楚信号的程度。

**（3）多位宽信号不同进制显示**

对于多位宽信号，可以根据需要设置为二进制、十六进制、八进制、十进制等不同的进制。

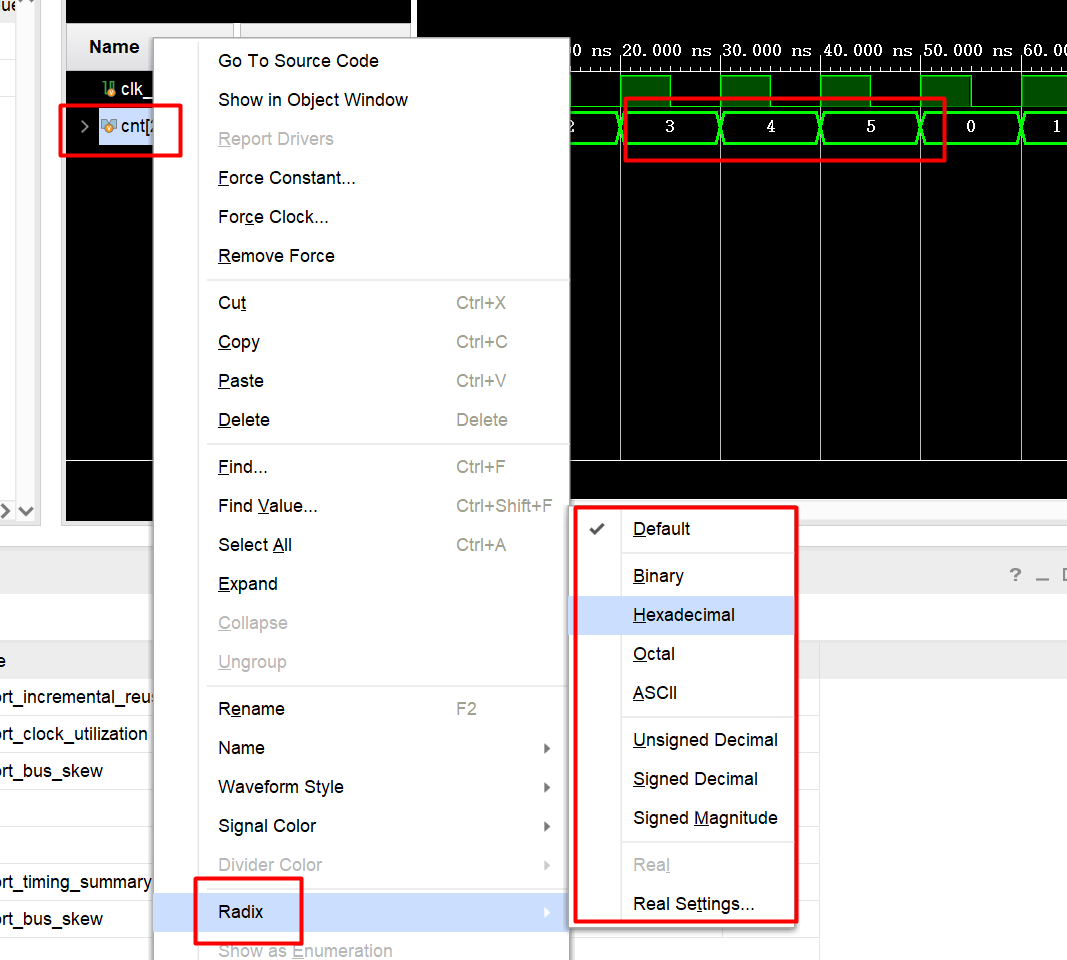


图2-4 显示进制修改

**（4）多位宽信号用值查找快速定位**

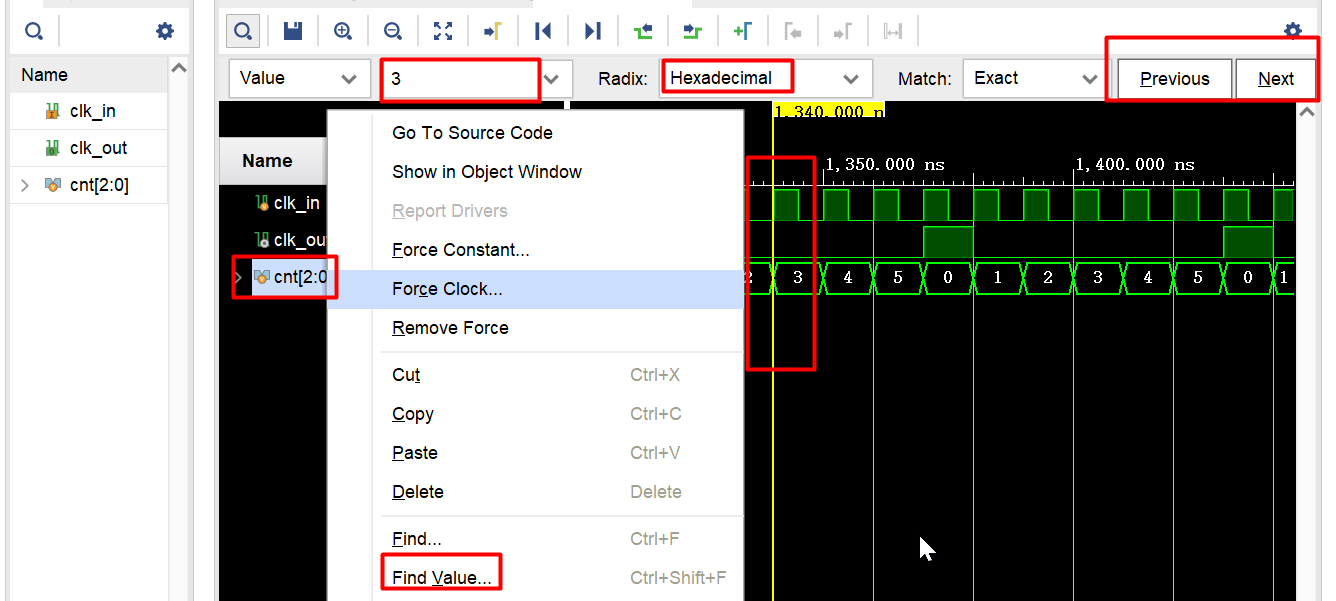
有时候需要快速找到一个多位宽信号等于某个值的时刻，建议用值查找（Find Value）的方式去找，不要用鼠标拖着信号下面的滚动条用眼睛去找，操作方式如下，需要注意Radix是否一致，输入值之后按enter键即会跳到对应值的边沿。

图2-5 find value操作

**（5）给重要的时刻做标记**

在波形分析过程中，及时给你认为重要的时刻做标记（Marker）。做标记的方法很简单，在波形上你关注的时刻处左击，此时游标（Cursor）将出现在你关注的时刻，点击波形上方工具条中的，就做好了一个标记。之后无论你是直接移动波形至Marker处还是使用波形上方工具条中的快速定位到标记处，都会大幅度提高定位效率。

**（6）将内部信号添加到波形窗口**

仿真窗口默认值显示仿真顶层模块中module的输入输出信号，但debug有时需要查看内部信号的变化。按如下操作添加内部信号的波形窗口。

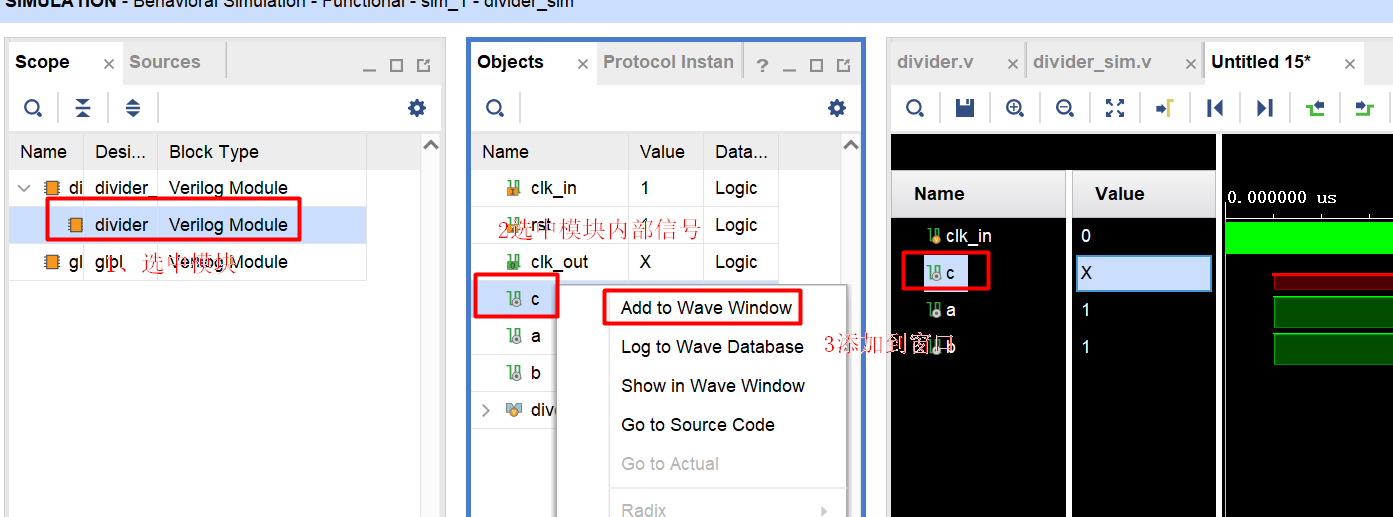


图2-6 添加内部信号到波形窗口

会发现新加入的信号没有波形数据，需要重新运行仿真才会有。如果需要仿真一次把所有信号都记录下来需要修改Vivado的全局配置。具体操作如下图所示，打开全局settings,在右部选择“Simulation”标签，然后在下面找到“xsim.simulate.log\_all\_signals”选项，将其勾选上，点击OK保存配置。

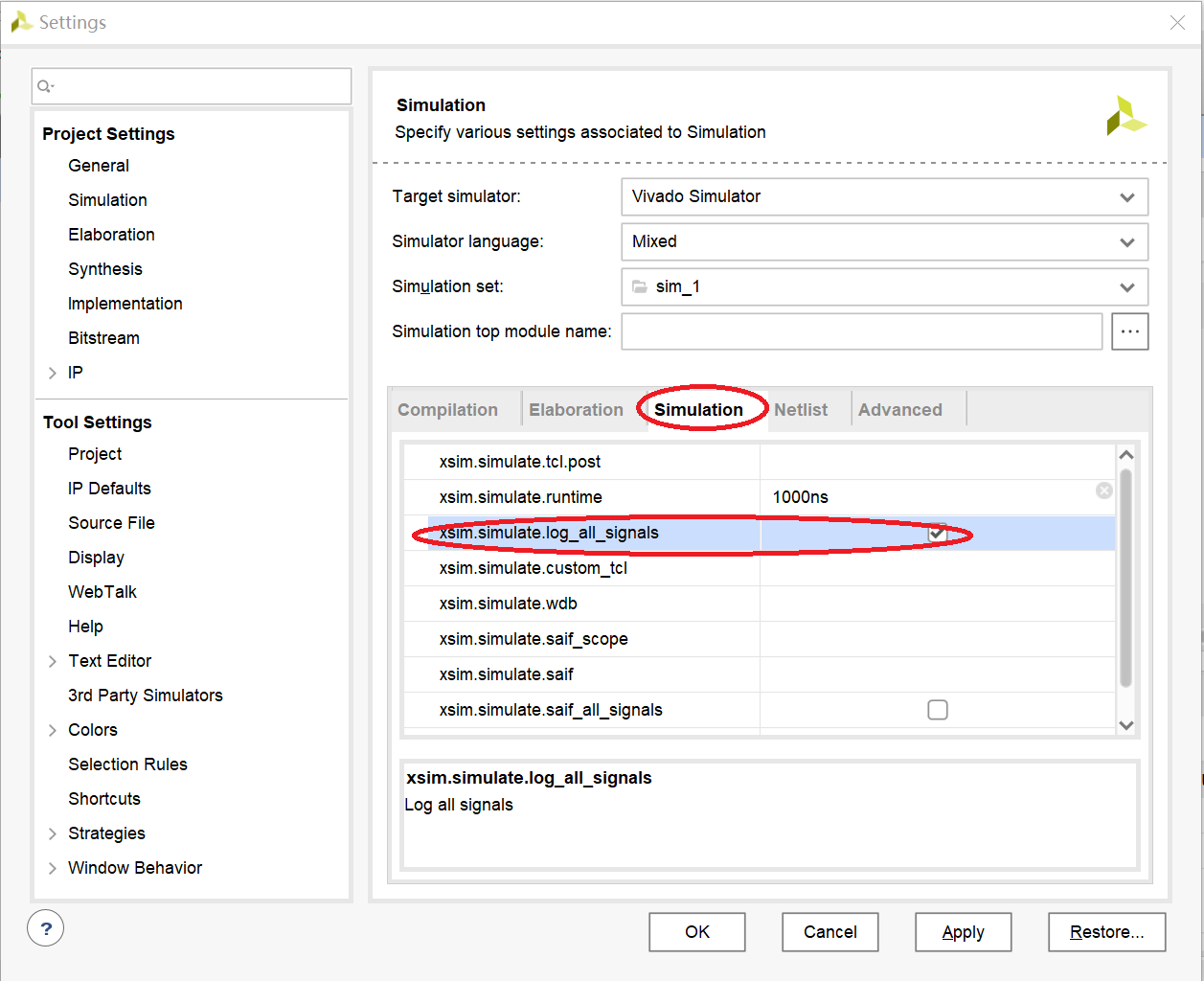


图2-7 添加新信号自动显示波形的设置

该设置使得调试更方便，但也使得单次仿真运行时间比不配置该选项要长，记录的数据也更多，日志文件更大，因为一次仿真需要记录所有信号的仿真数据。建议配置上，避免来回运行仿真浪费时间。

上面介绍了常用的仿真技巧，更多的波形分析操作需要在实践中学习。

### 2.4.2 波形异常类错误调试

波形异常类错误是指那些不需要分析电路设计的功能直接观察波形图就能判断的错，比如波形中信号出现“X”了。通常分为下面5类，指导书只介绍前面2类。

（1）信号为“X”；

（2）信号为“Z”；

（3）波形停止；

（4）越沿采样：上升沿采样到被采样数据在上升沿后的值；

（5）其他，波形怪异：仿真波形图怪异，与设计的电路功能无关的错误。

**（1）信号为“X”**

“X”表示不定值，往往是由于以下两个原因之一导致的。

（a）RTL里声明为reg型的变量，从未被赋值或被错误赋值。比如下面计数器的输出，clk\_out是reg型变量，但未被赋值。如果赋的值仍然是X，clk\_out仍然是X。

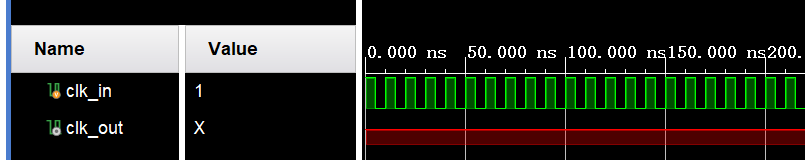
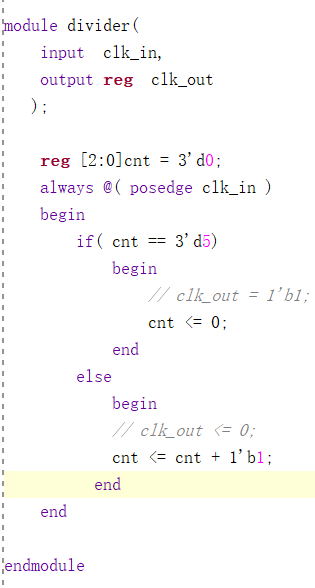


图2-8 reg信号未赋值代码与仿真波形

（b）RTL里写成了多驱动的代码，可能会导致“X”。多驱动即2个及2个以上电路单元驱动同一信号。Vivado对于多驱动的代码，只是可能导致“X”，有些多驱动代码可能会导致综合时失败，并且会在综合的信息输出报多驱动的Critial warning，但仿真过程不会有任何提示。下图是多驱动示例：

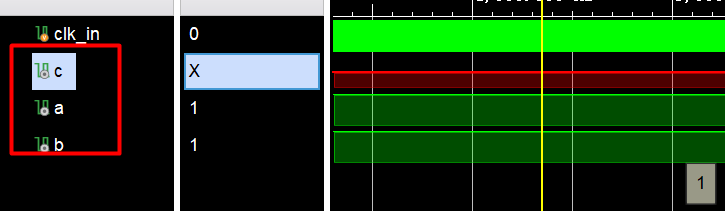
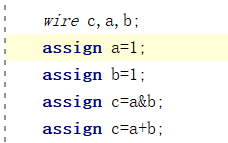


图2-9多驱动代码和仿真波形

针对信号为“X”情况，有以下几点建议：

（a）一旦发现仿真错误来自某个信号为“X”，则向前追踪产生该信号的因子信号，看是哪个为“X”，一直追踪下去直到追踪到某个信号未赋值，随后修正；

（b）如果因子信号都没有为“X”的，则很可能是多驱动导致的，则进行综合然后排查Error和Critical warning；

（c）寄存器型信号如果没有复位值，在复位阶段其值可能也为“X”，但复位阶段时间比较短，一般不影响；

（d）“X”和1进行或运算结果为1，“X”和0进行与运算结果为0。

**（2）信号为“Z”**

“Z”表示高阻，比如电路断路了就是显示为高阻，往往是由于以下两个原因之一导致的：

（a）RTL里声明为wire型的变量从未被赋值；

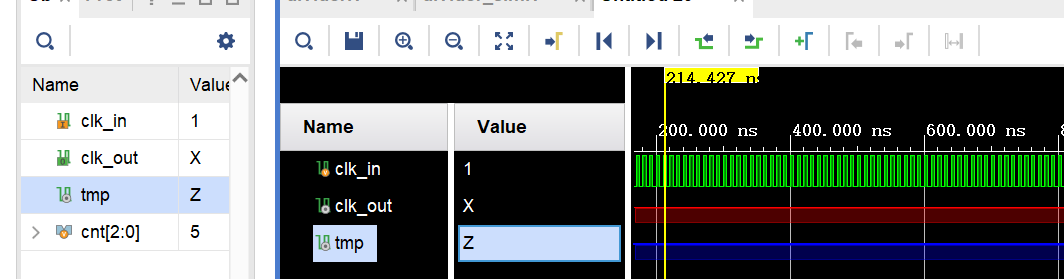
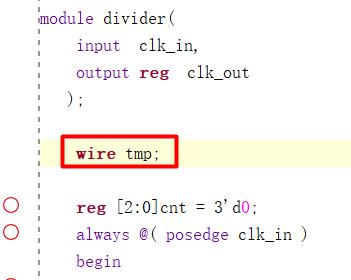


图2-10 wire型变量未赋值代码与仿真波形

（b）模块调用的信号未连接导致的信号悬空。下图给的例子虽然是仿真sim文件示例化divider模块没有连接，设计中未连接效果是一样的。从仿真波形来看，顶层的仿真模块里clk\_in信号是正常的，但是实例化模块因未连接是Z状态。

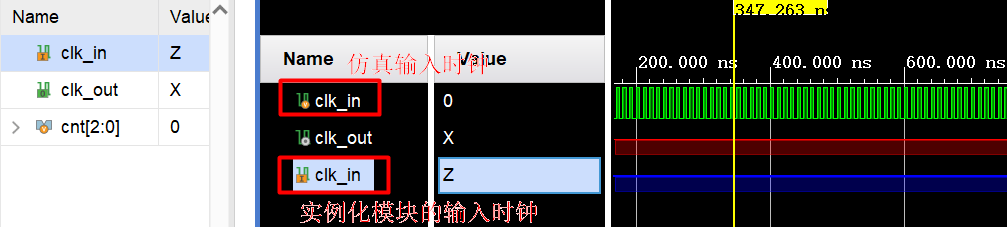
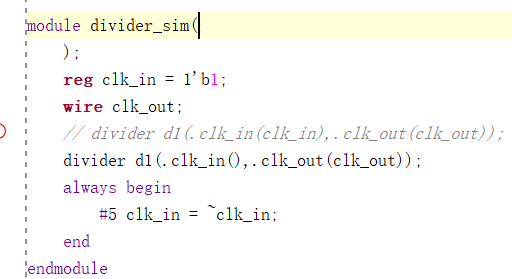


图2-11 实例化input信号未连接代码与仿真波形

模块调用时信号未连接，未连接包括两种：显式的未连接和隐式的未连接（端口直接不写）。“显式的未连接”一般是人为故意设置的，只针对output类接口,母模块里可以不使用该信号。“隐式的未连接”则是疏忽，属于代码不规范，往往也是导致信号值为“Z”的主要原因。所有的input类接口被调用时不允许悬空。

针对信号为“X”情况，有以下几点建议：

（a）RTL编写时注意代码规范，特别是模块调用时，按接口顺序一一对应；

（b）所有input类接口被调用时不允许悬空；

（c）一旦发现一个信号为“Z”，向前追踪产生该信号的因子信号，看是哪个为“Z”，一直追踪下去直到追踪到该模块里的input接口，随后进行修正；

（d）有可能“Z”只出现在向量信号里的某几位上，也是一样的追踪，有可能调用时某个接口存在宽度不匹配也会带来该接口上某些位为“Z”。

# 3 实验检查与提交要求

题目一：现场检查实验现象（1分），分频精度需看仿真波形；

题目二：检查提交.v文件（含仿真文件）、仿真结果截图及文字说明（1分）。

指导书提到的仿真调试技巧多加练习。