

- [CDC \(Clock Domain Clock跨时钟域传输\) 学习记录](#)
 - [学习博客](#)
 - [任务](#)
 - [数据跨时钟域传输三大方法：](#)
 - [快时钟域到慢时钟域](#)
 - [慢时钟域到快时钟域](#)
 - [异步fifo](#)
 - [结果展示](#)
 - [1.读写数据](#)
 - [2.fifo空指示](#)
 - [3.fifo满指示](#)

CDC (Clock Domain Clock跨时钟域传输) 学习记录

学习博客

[跨时钟域传输总结 \(包含verilog代码|Testbench|仿真结果\) _verilog跨时钟域处理-CSDN博客](#)

[异步FIFO设计原理与设计方法以及重要问题汇总 \(包含verilog代码|Testbench|仿真结果\) _请简述异步fifo的基本方案 并给出框图、关键代码-CSDN博客](#)

任务

设计32位宽，1024深度的异步fifo

数据跨时钟域传输三大方法：

- [mux同步器法](#)
- [握手协议](#)
- [异步fifo](#)

[快时钟域到慢时钟域](#)

核心思想：延长信号长度

[慢时钟域到快时钟域](#)

FPGA设计的“打拍（寄存）”和“亚稳态”到底是什么？_fpga打拍的作用-CSDN博客

常用：双极触发器缓存，两拍

tips：第一级寄存器产生亚稳态并经过自身后可以稳定输出的概率为 70%~80%左右，第二级寄存器可以稳定输出的概率为 99%左右，后面再多加寄存器的级数改善效果就不明显了，所以 数据进来后一般选择打两拍即可

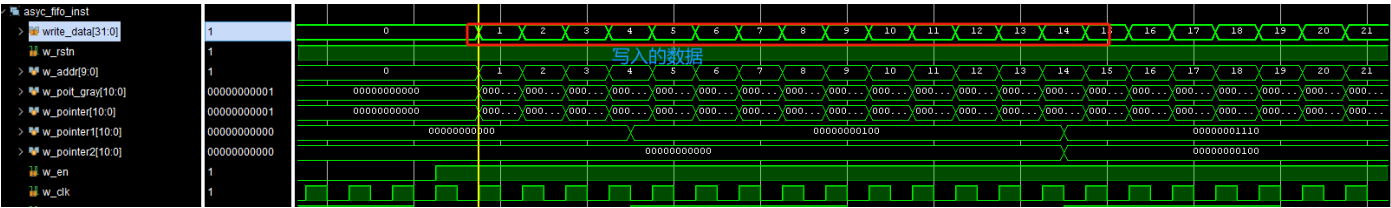
异步fifo

tips：跨时钟域同步读写信号：读信号在写信号的时钟域与其同步产生full信号，防止fifo被写爆，同理，写信号在读信号的时钟域与其同步生成empty信号

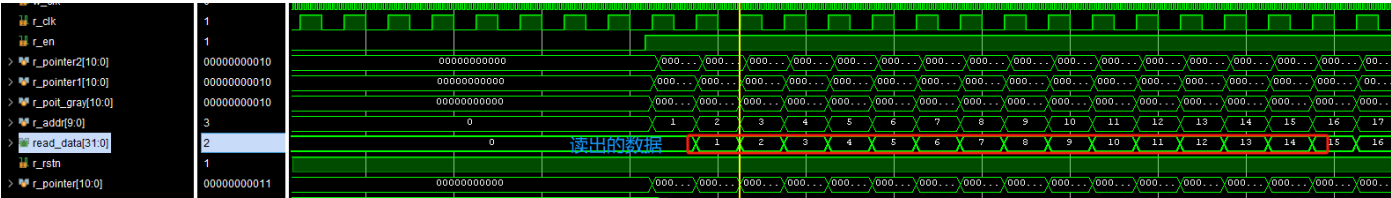
结果展示

1.读写数据

写入的数据（按1234567写入，从0写到1023）

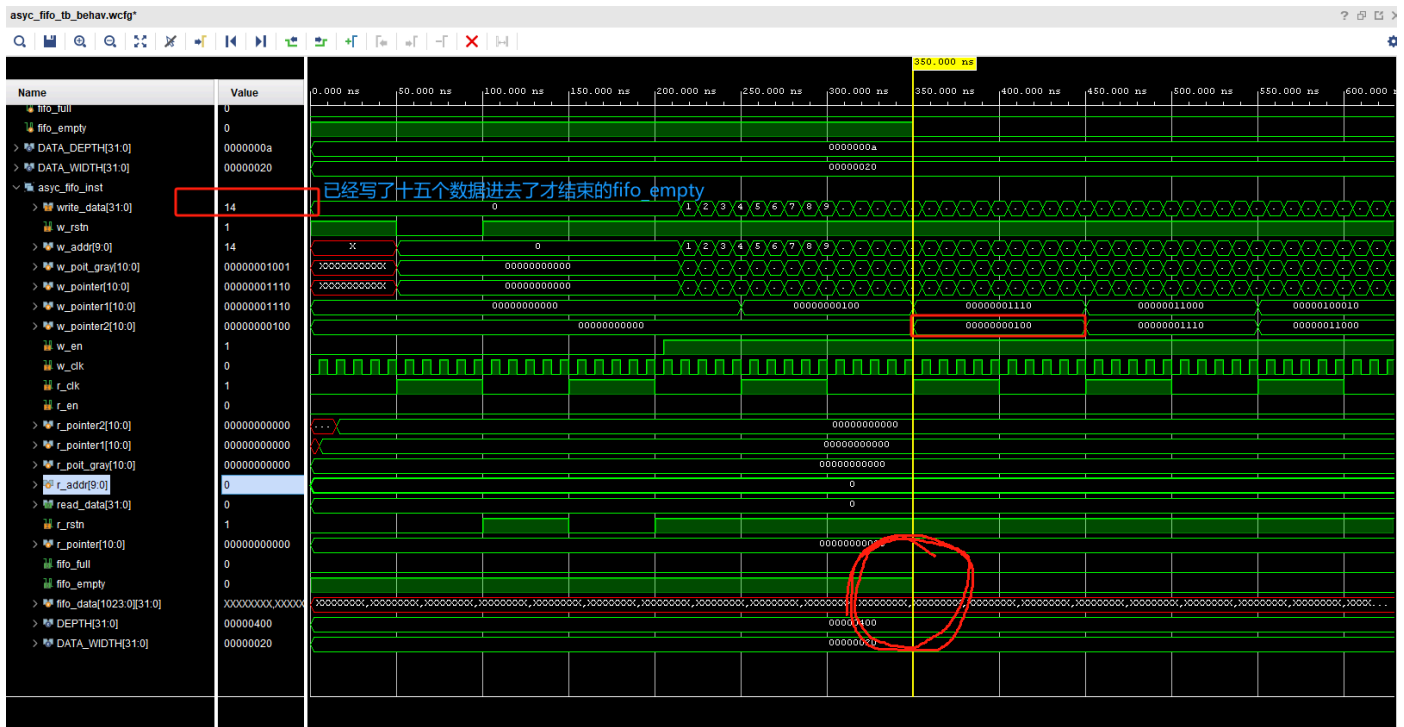


读出的数据



2.fifo空指示

当写指针延迟两拍之后和读指针不相等时，empty信号才拉低，事实是这个时候已经写入了15个数据了，这就是所谓的“虚空”



3.fifo满指示

判断逻辑如下，即标志位相同，其他位不同的时候，整整差了一圈，这个时候fifo写满

```
(r_pointer2 ^ w_pointer) == $pow(2, DEPTH_WIDTH)
```

