

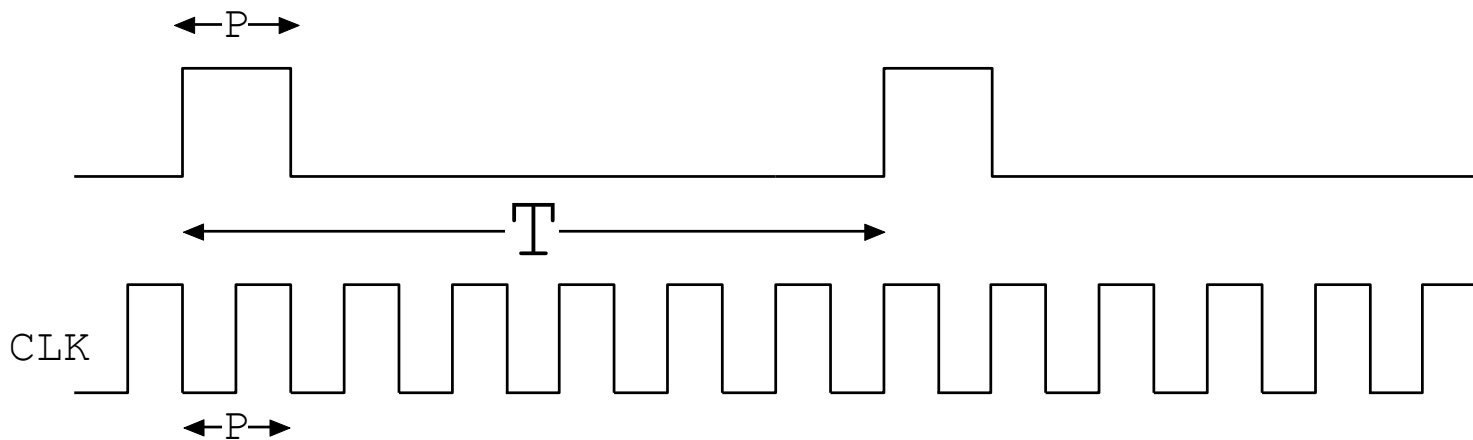
FPGA 实验

时间基准电路 和 带使能的多周期计数器

中国传媒大学 数字化工程中心 杜伟韬
duweitao@cuc.edu.cn

时间基准电路介绍

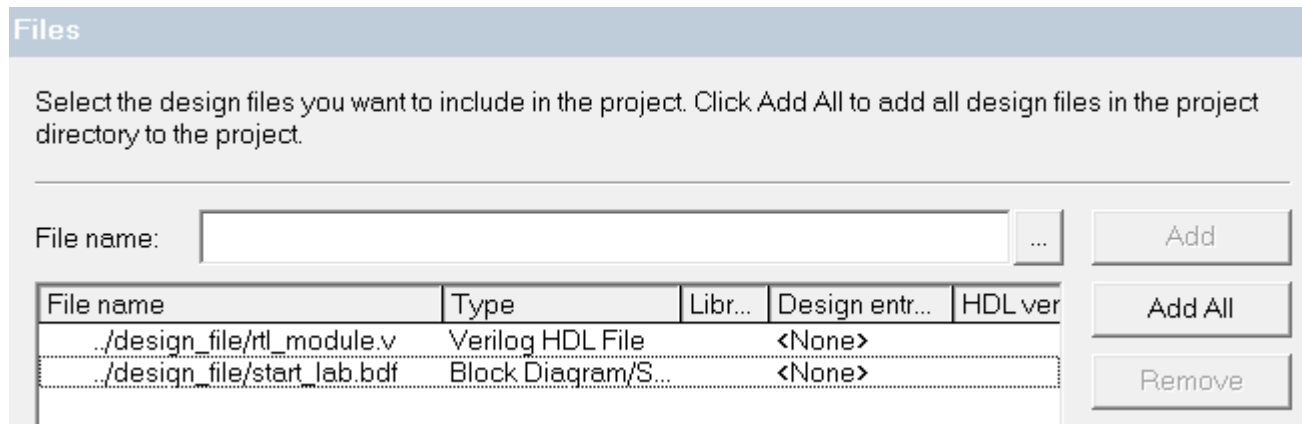
- 定时发出一个窄脉冲的电路
- 下图为典型情况时序
 - 每T时间，生成一个宽度为P时间的脉冲
 - 电路的工作时钟的周期为P
 - 使用一个计数器可以实现该时序



本实验设计目标

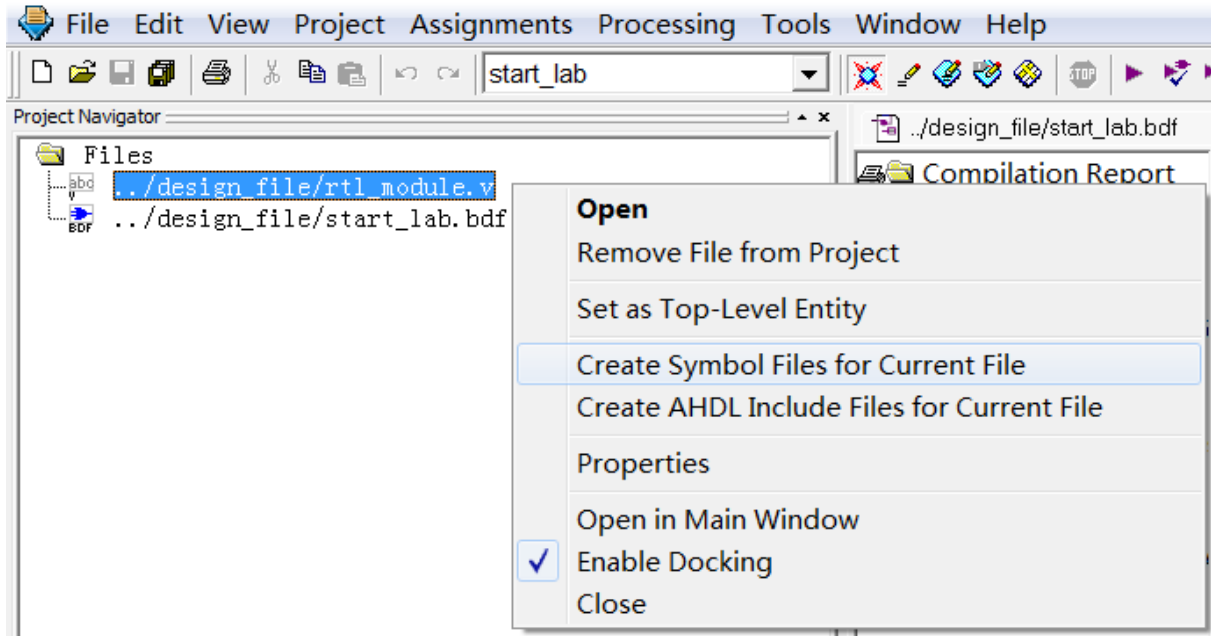
- 设计时间基准电路和带使能的多周期计数器
- 时间基准电路生成同步时间基准信号
- 多周期计数器对时间基准信号进行计数
- 本质上是一个两级计数器级联的的电路结构
 - 第一级计数器生成时间基准信号
 - 第二级计数器用时间基准信号作为计数使能

新建项目 顶层是BDF文件



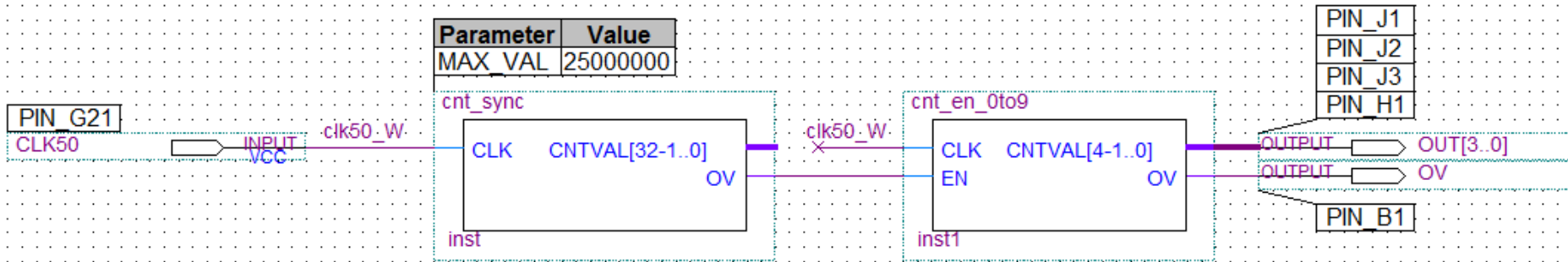
- 包含Verilog文件 rtl_module.v
- 该文件中包含要例化的子模块的原型代码

为代码文件创建符号(symbol)



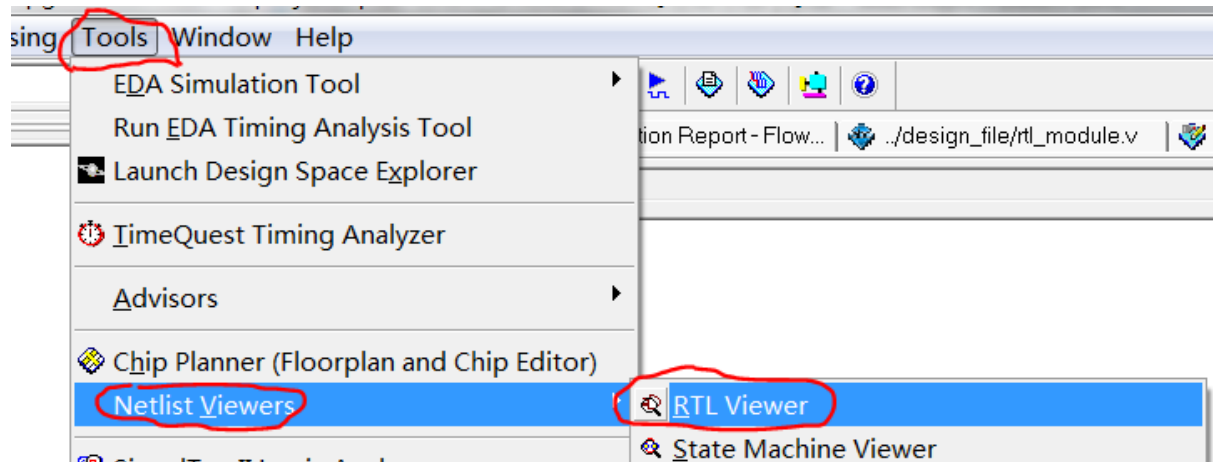
- 该过程生成代码文件对应的符号模块
- 可以在BDF文件中调用
- 代码文件中包含要使用的计数器verilog代码

在BDF文件中添加模块

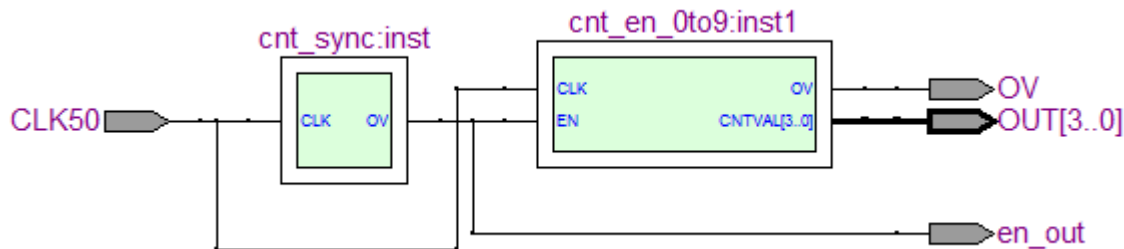


- 时间基准生成模块和带使能的计数器模块
- 该模块来自工程项目的verilog文件中
- 编译下载电路，观察现象

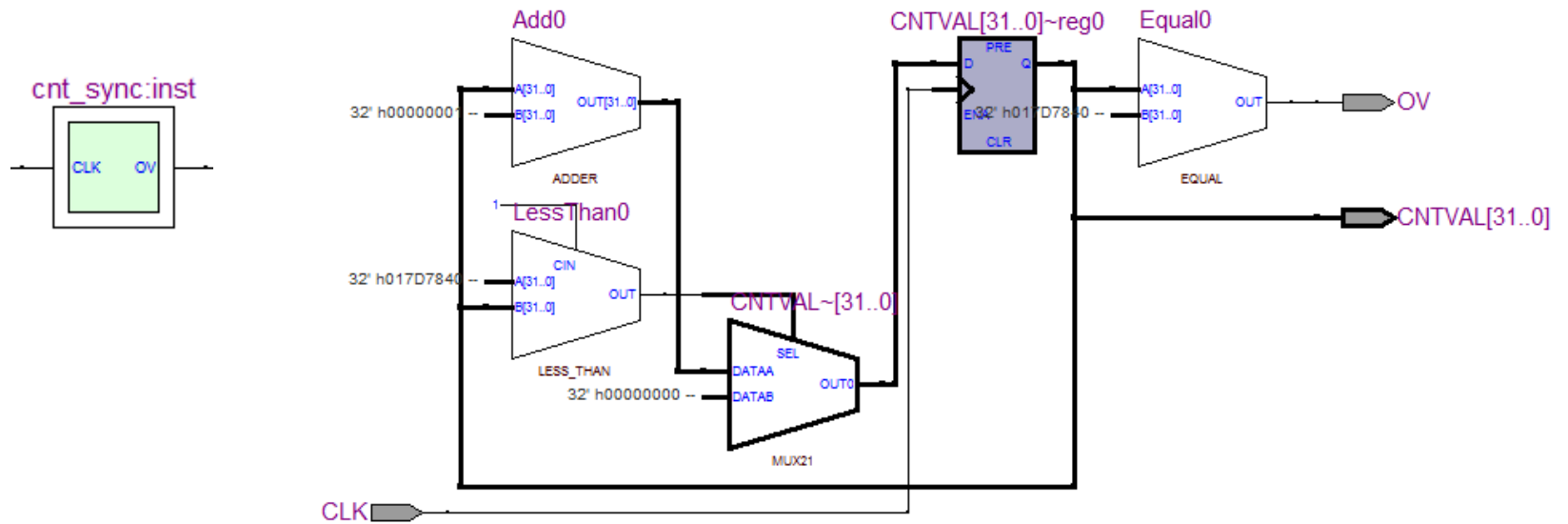
Analysis and Synthesis 观察RTL视图



- 先进行部分编译 
- 然后观察RTL视图
- 双击RTL模块可以观察其内部的子模块结构

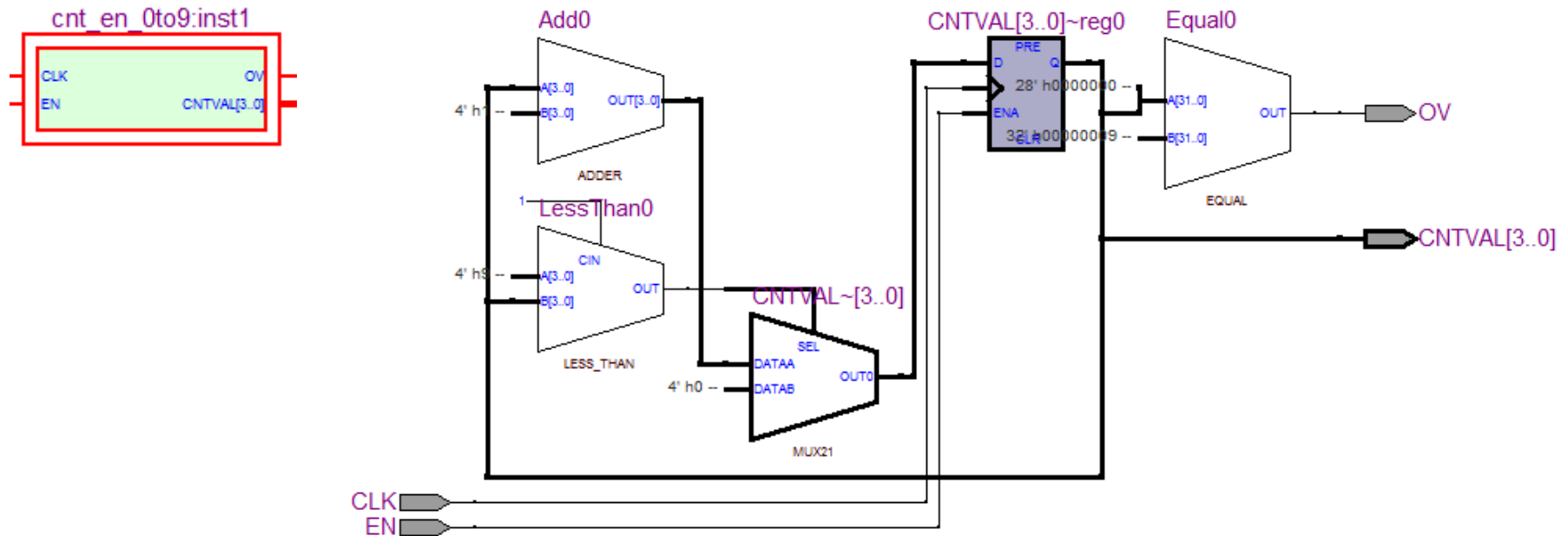


观察时基计数器模块内部电路结构



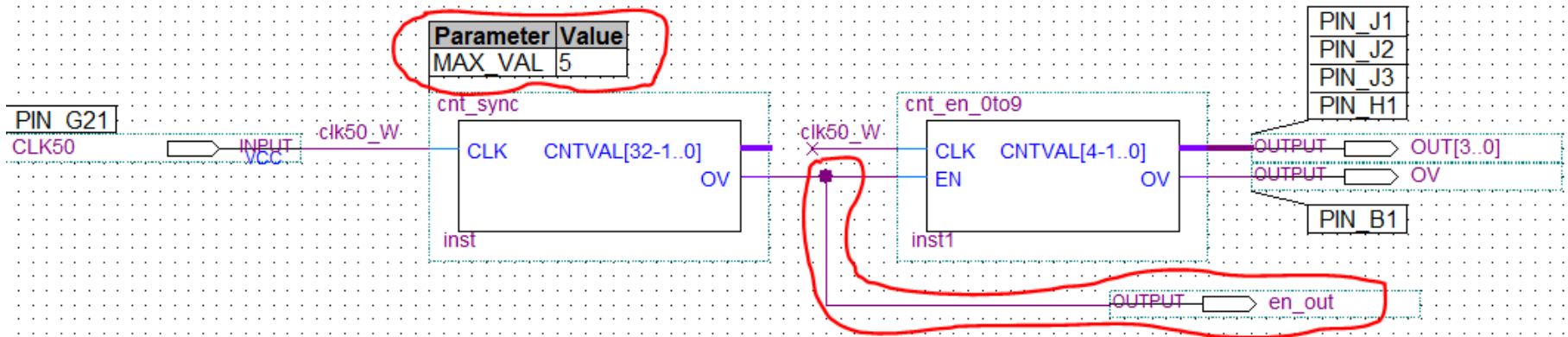
- 确认电路结构的正确性

观察使能计数器模块内部电路结构



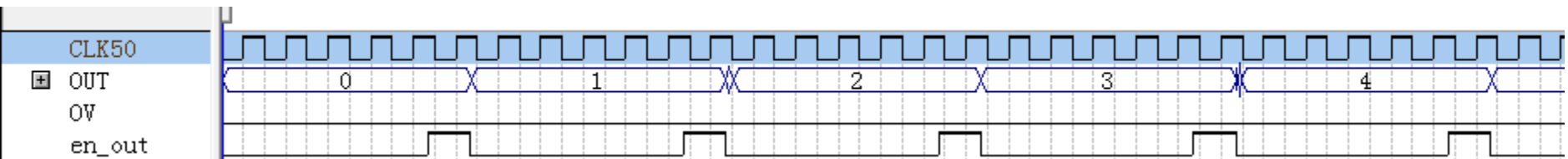
- 确认电路结构正确性
- 请思考，编译器是如何处理使能信号的
- 请和不带使能的计数器RTL对比一下区别

对电路进行波形仿真



- 把时间基准模块的最大计数值参数，改为5。
- 把带使能的计数器的输入使能信号引出作为输出，不必分配管脚。
- 思考为什么要这样做？提示，波形仿真的时间尺度不宜太长。编译器有可能会优化掉中间结果。

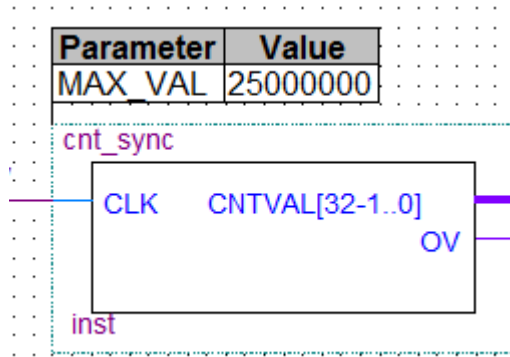
波形仿真结果



- 可以见到，带使能的计数器是一种多周期的时序行为，即：有使能信号的周期，该计数器的计数值加1，没有使能的周期，计数值保持不动。这种“使能驱动”的电路设计原则非常重要。在复杂电路中会非常频繁的使用。

使用SignalTap观测电路

- 把时间基准计数器的最大值改为25000000



Instance Manager: Ready to acquire

Instance	Status	LEs: 812	Memory: 4864	M512/LUTRAM: 0	M4K/M9K: 2
auto_signaltap...	Not running	812 cells	4864 bits	0 blocks	2 blocks

JTAG Chain Configuration: JTAG ready

Hardware: USB-Blaster [USB-0] Setup...

Device: @1: EP3C16 (0x020F20DD) Scan Chain

SOF Manager: []

trigger: 2013/10/09 15:23:29 #1 Allow all changes

Type	Alias	Name	Data En...	Trigger Ena...	Trigger Condi...
		en_out	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Basic
		OUT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXb
		OV	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		...ync:inst CNTVAL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXXXX...

Clock: CLK50

Data

Sample depth: 128 RAM type: Auto

Segmented: 128 1 sample segments

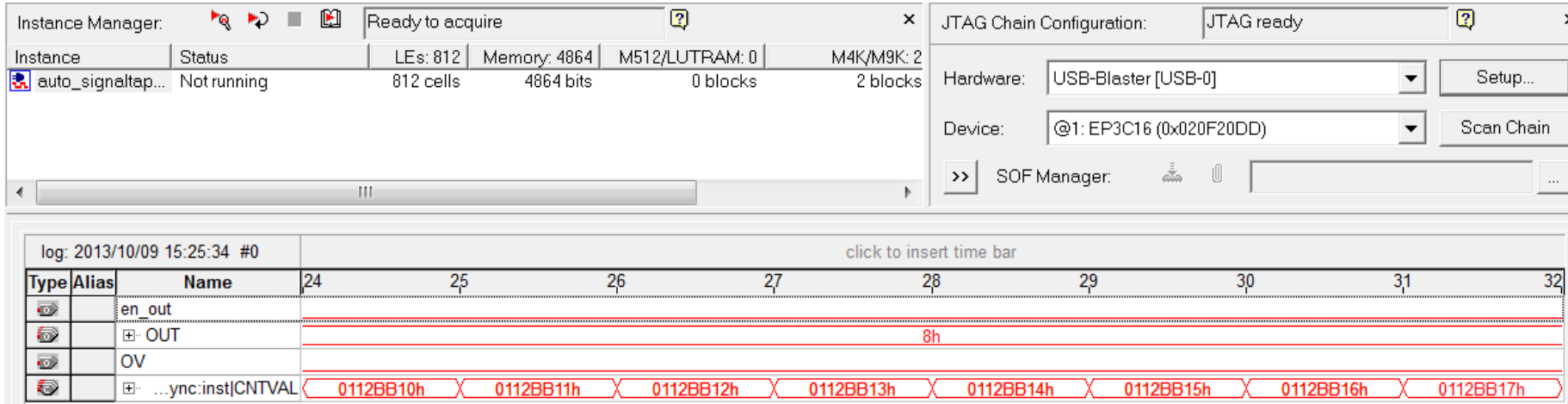
Trigger

Trigger flow control: Sequential

Trigger position: Pre trigger position

Trigger conditions: 1

观测结果类似如下截图



- 问题，同一时刻，只能观测到一个计数值
- 不利于调试
- 更高要求，如果希望更多关注每次en_out信号为1的时刻附近的其他信号情况？如何办？

高级触发方式-分段触发

- 配置如下图，使用en_out为触发信号，
- 选中segmented模式，8段16样点

The screenshot displays the Xilinx ISE configuration environment. At the top, the Instance Manager shows a project named 'auto_signaltap...' with a status of 'Not running'. The JTAG Chain Configuration window is also visible, showing the hardware as 'USB-Blaster [USB-0]' and the device as '@1: EP3C16 (0x020F20DD)'. The main configuration window is titled 'trigger: 2013/10/09 15:37:27 #0' and features a table of trigger nodes and a configuration panel on the right.

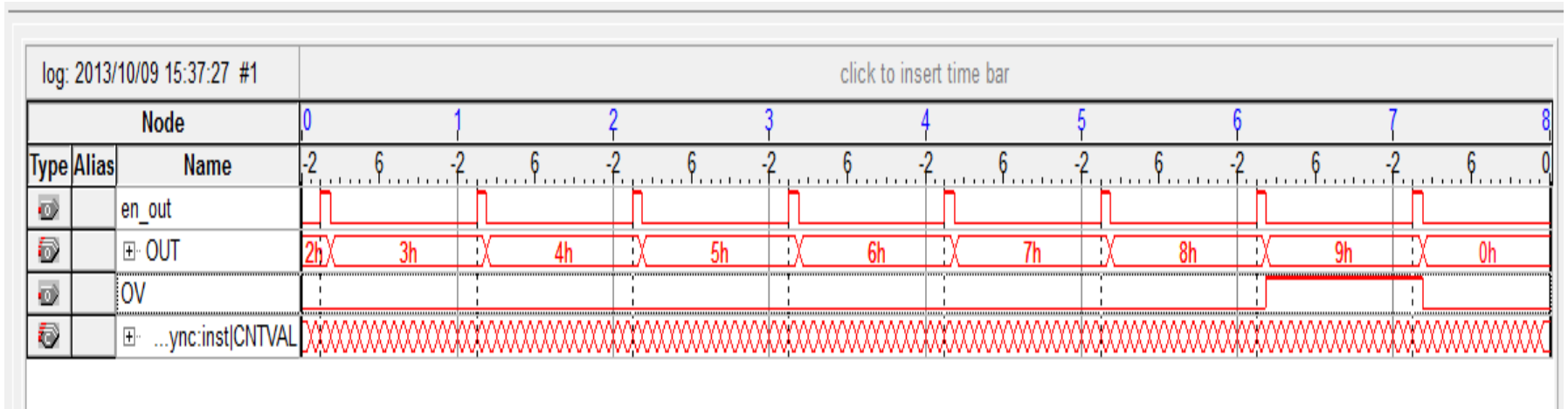
Type	Alias	Node Name	Data En...	Trigger Ena...	Trigger Condi...
		en_out	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Basic
		OUT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXb
		OV	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		...ync:inst CNTVAL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXXXX...

The configuration panel on the right shows the following settings:

- Clock: CLK50
- Data: Sample depth: 128, RAM type: Auto
- Segmented: 8 16 sample segments
- Trigger: Trigger flow control: Sequential, Trigger position: Pre trigger position, Trigger conditions: 1
- Trigger in

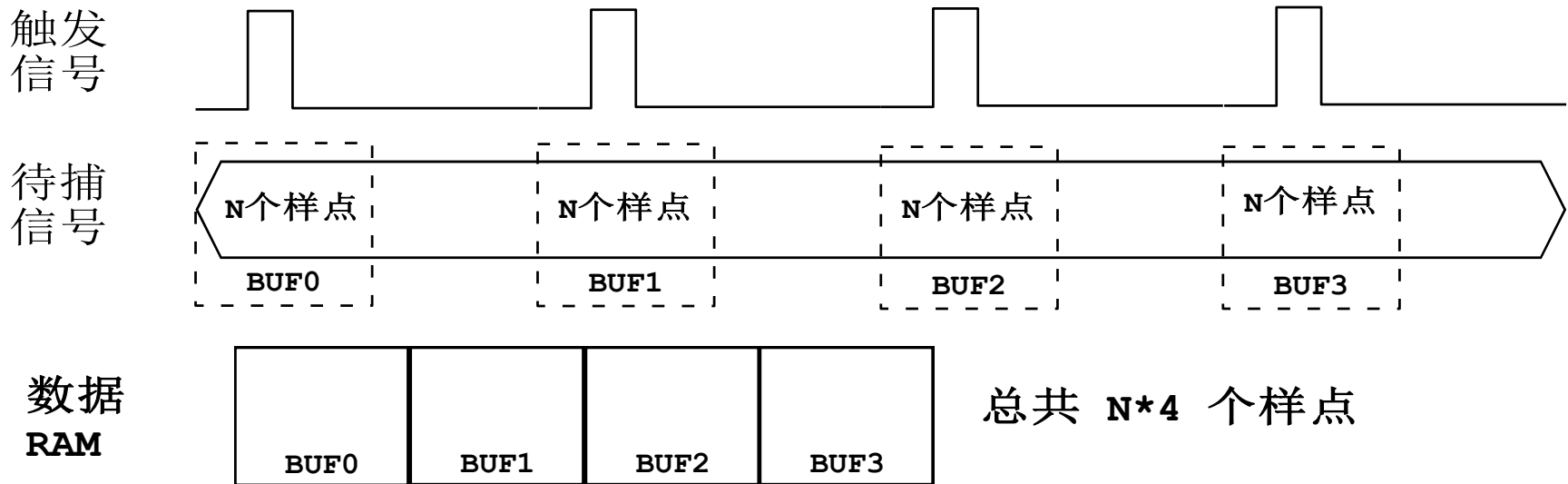
捕获结果

- 每次抓取触发事件附近的波形样点
- 抓取多次



分段触发的捕获原理

- 一次填充一段数据RAM
- 多次填充
- 全部RAM被填充之后，由JTAG口送PC机



学生实验

- 仿照本实验，请完成以下任务
- 1、参照代码，把后级计数器的计数范围改为0-15。
- 2、把计数器的0-15计数值经过译码，在DE0 的 HEX LED上显示成0-9-A-F的十六进制数
- 3、修改时间基准发生器，设计一个使用2个HEX LED，精度为0.1秒，范围为0-9.9秒的计时秒表。
- 4、自行设计上面计时器的附加控制功能（清零、暂停）。

