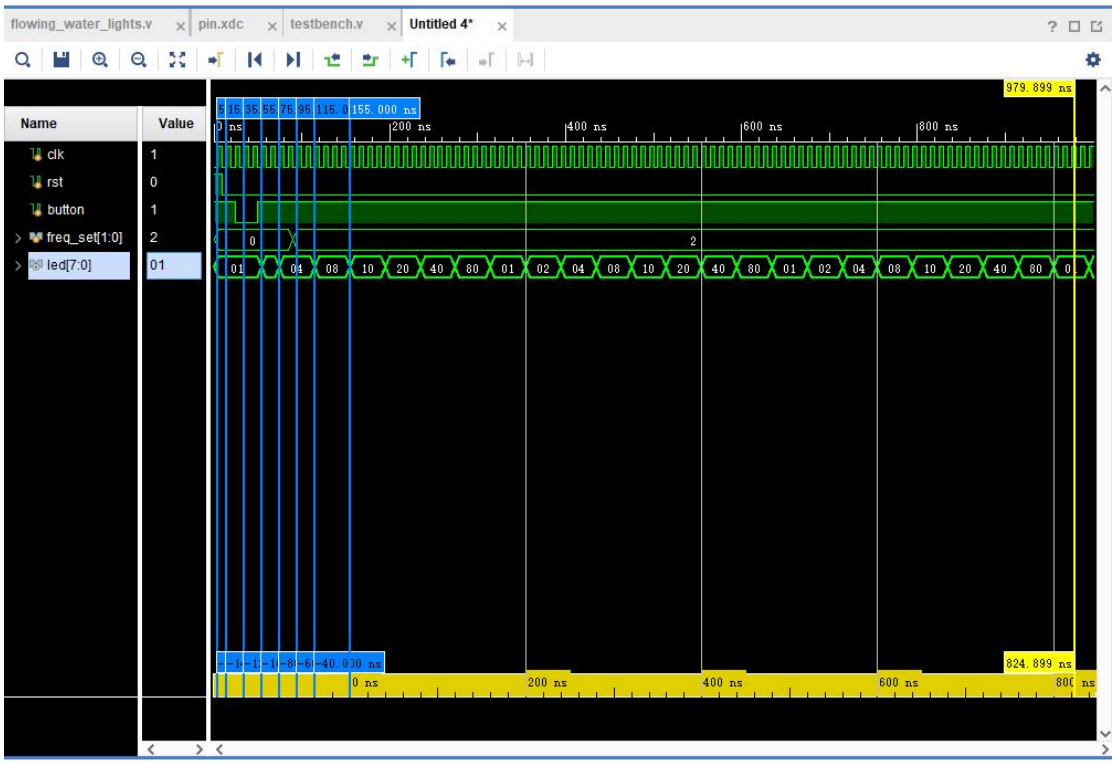
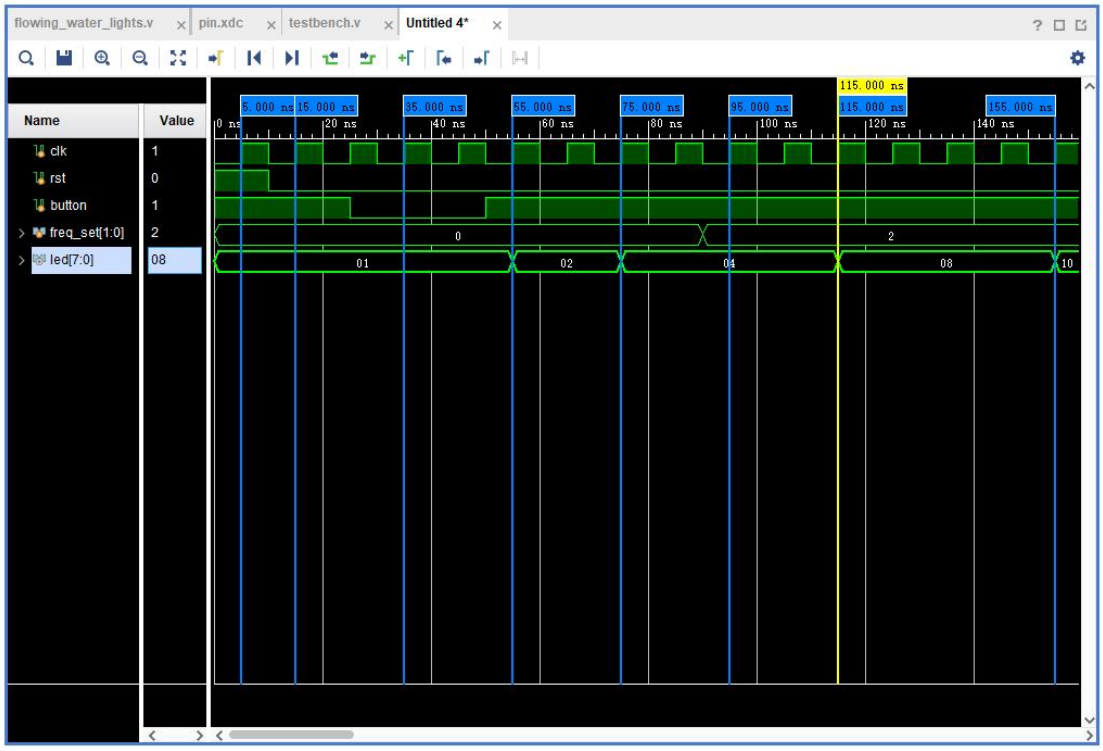
一、流水灯（flowing_water_lights）

1. 流水灯的波形如下：





信号说明：时钟信号 clk、复位信号 rst、流水灯启动信号 button、频率设置信号 freq_set、输出的 led 信号 led。

频率说明：为了便于仿真，将计数器的结束计数条件做了调整。以上波形图中显示的计数器分频后的时钟周期为 clk （一周期为 5ns）的 4、6、8、10 倍（即 freq_set 为 0、1、2、3 对应的时钟周期为 20ns、30ns、40ns、50ns）。仿真中仅显示 freq_set 为 0、2 的情况。

从上述波形可以看出：

(1) 第一个时钟上升沿 5ns 时，rst 为 1，进行复位操作。而此后 led 信号一直保持为 01 直至 55ns 时才发生变化，说明复位操作成功执行；

(2) 第二个时钟上升沿 15ns 时，rst 为 0，button 为 1，freq_set 为 0，流水灯启动，输出信号 led 为 01。到 35ns 的时钟上升沿，button 为 0，流水灯暂停，输出信号 led 维持 01 不变。直至 55ns 的时钟上升沿，button 为 1，流水灯再次启动，此时输出信号 led 由 01 变为 02。说明 button 可以实现流水灯 led 信号变化的启动和停止功能；

(3) 第六个时钟上升沿 55ns 至第八个时钟上升沿 75ns 期间，rst 为 0，button 为 1，freq_set 为 0，输出信号 led 维持 02 直至 75ns 时发生变化，说明 freq_set 为 0 时对应的时钟周期为 20ns（频率为 50MHz），符合预期；

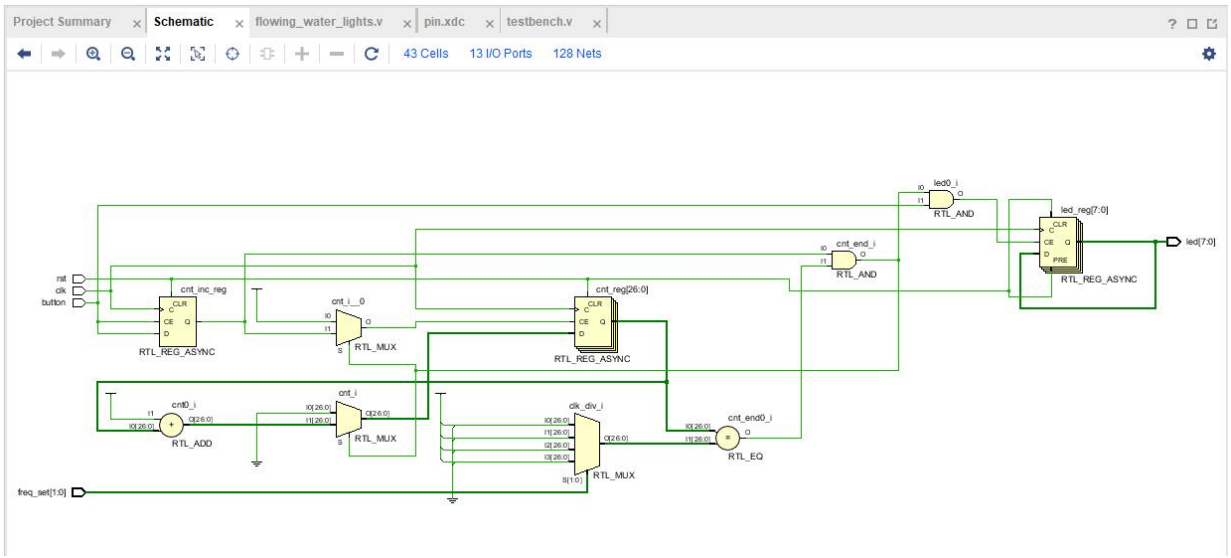
(4) 第十个时钟上升沿 95ns 时，rst 为 0，button 为 1，freq_set 变为 2，流水灯间隔切换，输出信号 led 为 04，但存续时间明显长于 freq_set 为 0 时对应的时钟周期（55ns-75ns 期间），符合预期；

(5) 第十二个时钟上升沿 115ns 时至第十六个时钟上升沿 155ns 期间，

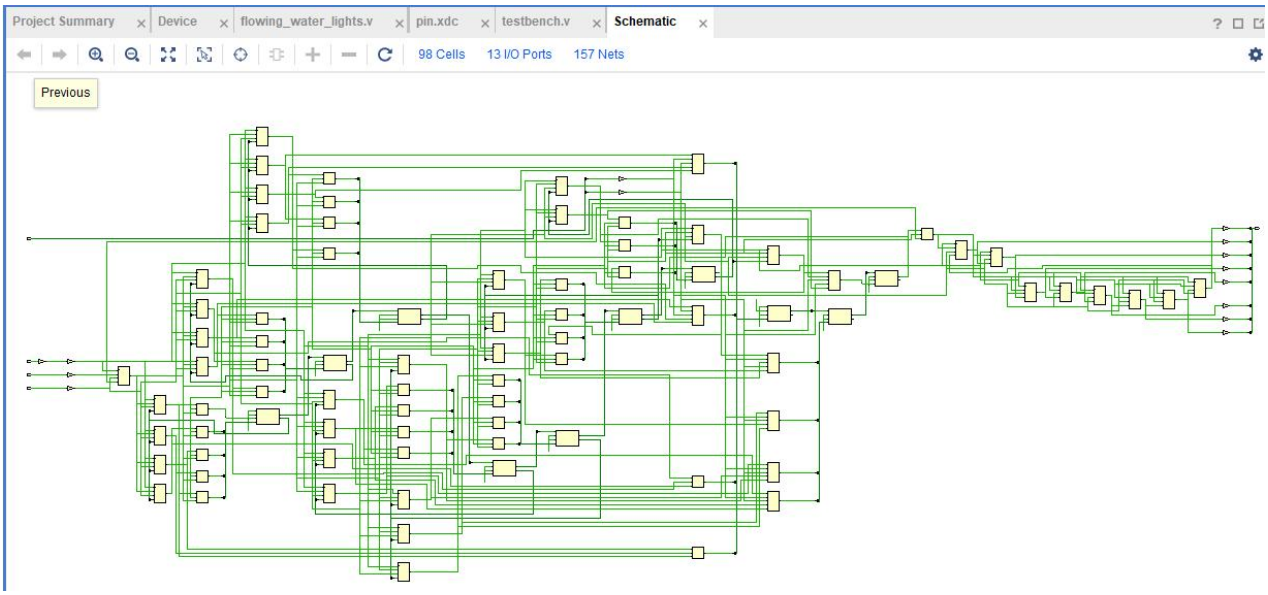
rst 为 0，button 为 1，freq_set 为 2，输出信号 led 在 115ns 时变为 08 并维持直至 155ns 时发生变化，说明 freq_set 为 2 时对应的时钟周期为 40ns（频率为 25MHz），符合预期。也说明流水灯间隔切换功能有效；

故根据上述分析，流水灯可以正确实现复位、启动、暂停、间隔切换的功能。

2. RTL Analysis



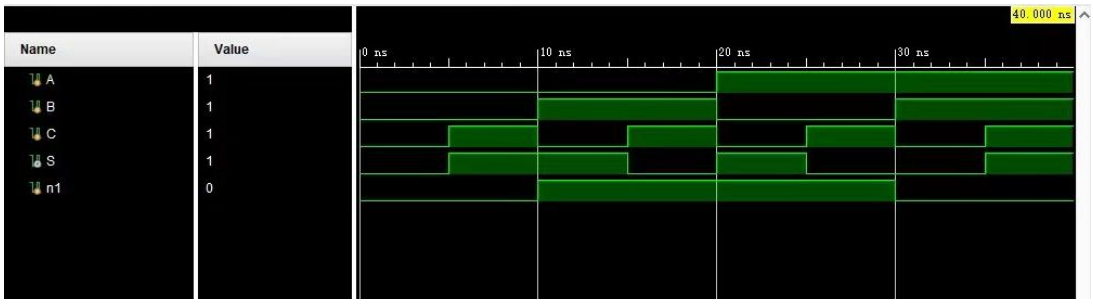
3. Synthesis schematic



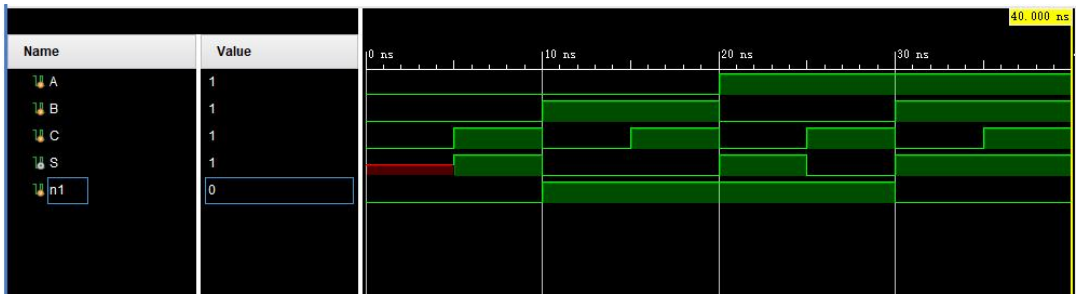
二、课后作业

1. 仿真波形

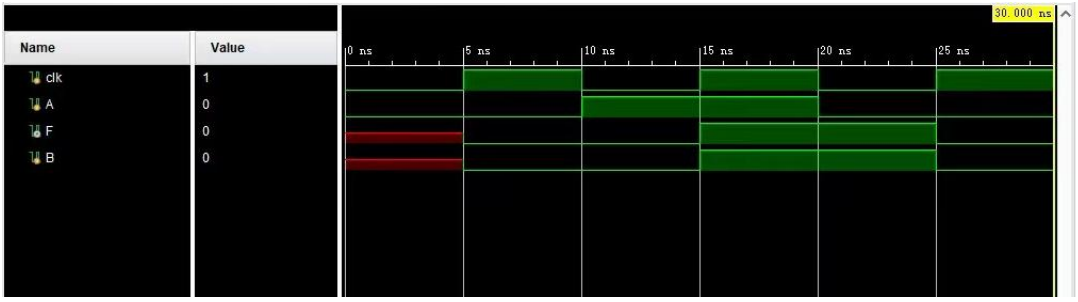
blockingEx1:



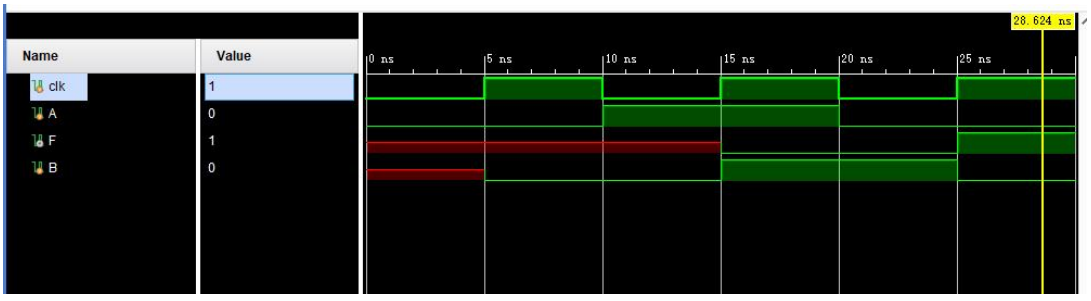
blockingEx2:



blockingEx3:

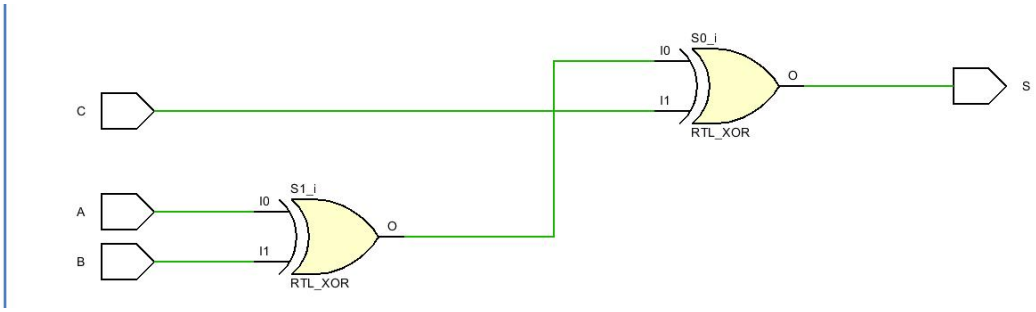


blockingEx4:

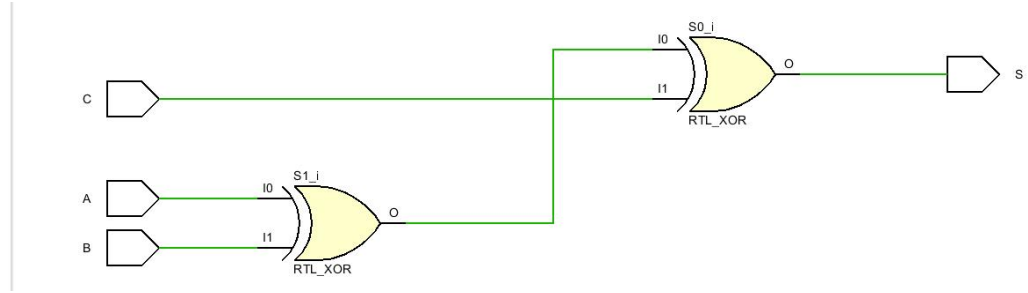


2. RTL Analysis

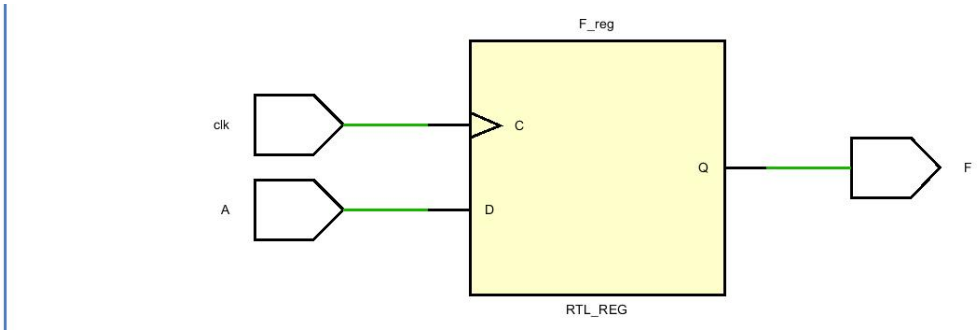
blockingEx1:



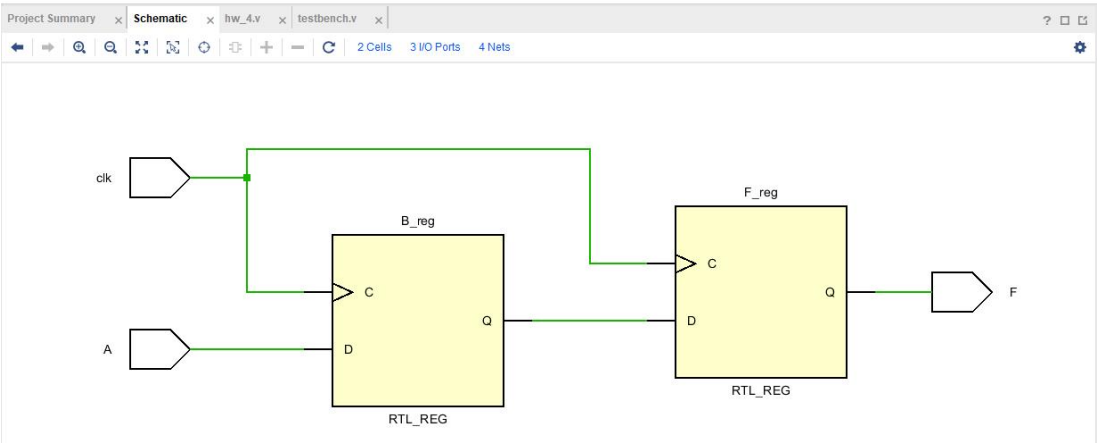
blockingEx2:



blockingEx3:

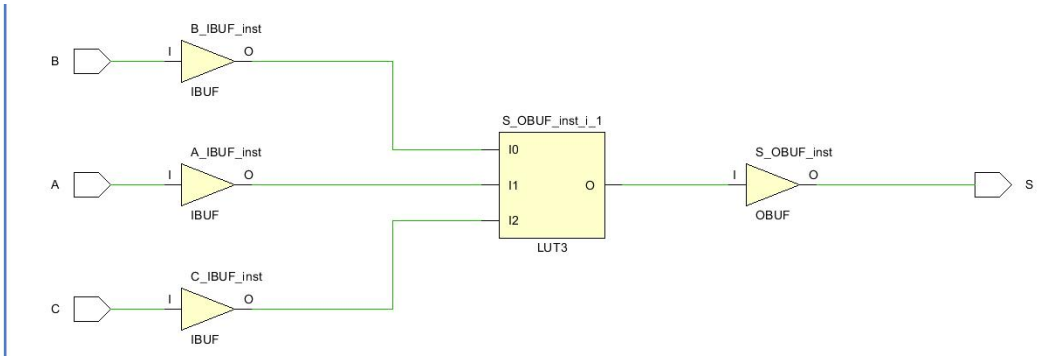


blockingEx4:

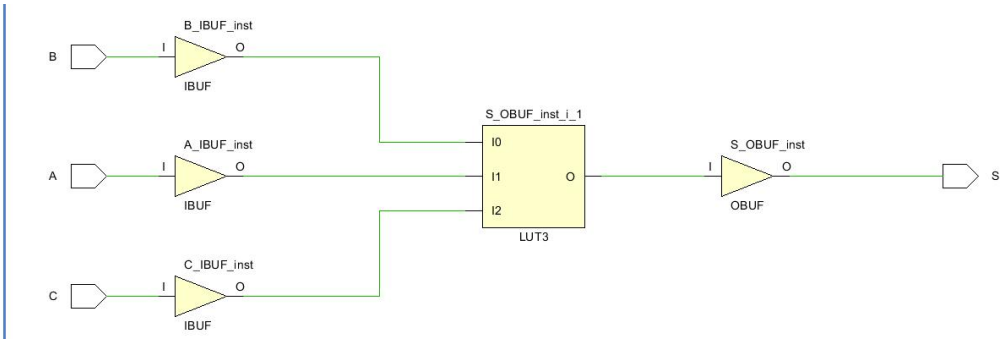


3. Synthesis schematic

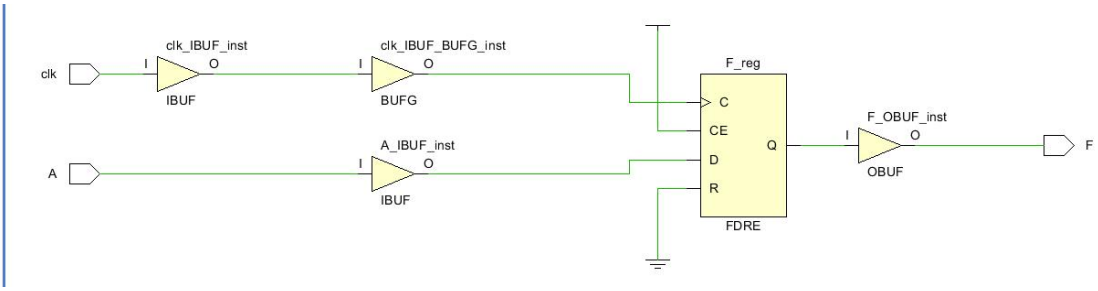
blockingEx1:



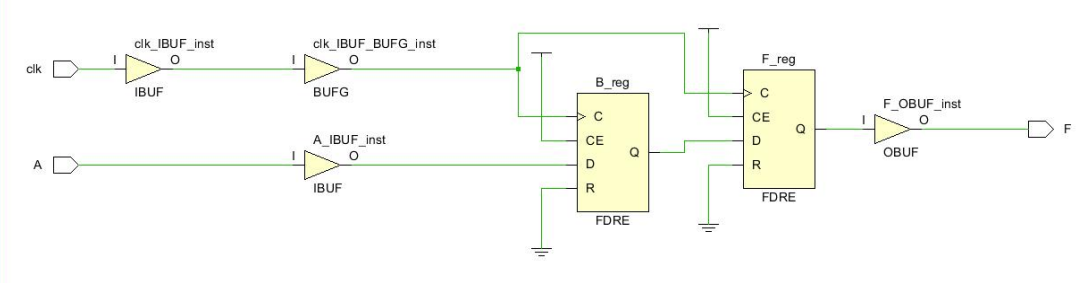
blockingEx2:



blockingEx3:



blockingEx4:



4. 整体分析

（1）对比 1

在 blockingEx1 的波形中始终有 n1=A^B 和 S=C^n1。表明全过程中总是先执行 n1=A^B，之后才会用更新后的 n1 的值去执行 S=C^n1。

而在 blockingEx2 的波形中，每当 A、B、C 发生变化时，n1 虽然也会更新为 A^B，但 S 的值总是由此时的 C 与更新前的 n1 异或得到的，即更新后的 n1 值并不能被用于执行 S=C^n1。

在 10ns 和 30ns 时，blockingEx1 和 blockingEx2 的 S 值不同：10ns 时，A、B、C 的值分别为 0、1、0，在 blockingEx1 的波形中 S 维持为 1，而在 blockingEx2 的波形中 S 由 1 变为 0；30ns 时，A、B、C 的值分别为 1、1、0，在 blockingEx1 的波形中 S 维持为 0，而在 blockingEx2 的波形中 S 由 0 变为 1。由此可以判断出阻塞赋值的两句是依次执行的（即 n1 的更新对后续有影响），但非阻塞赋值的两句是同时执行的。这导致执行 S=C^n1 时，n1 仍然为更新前的值，故最终结果不同。

（2）对比 2

在 blockingEx3 的波形中，每个时钟上升沿时，A、B、F 的值总是一样的。说明 B 先被赋予 A 的值，然后 F 再被赋予 B 的值，与代码中的“B = A;F = B;”相符合。但在 blockingEx4 的波形中，每个时钟上升沿时，F 的值总会变为 B 在时钟上升沿到来前的值，而 B 的值总会变为 A 在时钟上升沿到来前的值。说明在 blockingEx4 中，B 被赋予 A 的值和 F 再赋予 B 的值这两步是同时执行的。

由此同样可看出阻塞赋值的两句是依次执行的，但非阻塞赋值的两句

是同时执行的。