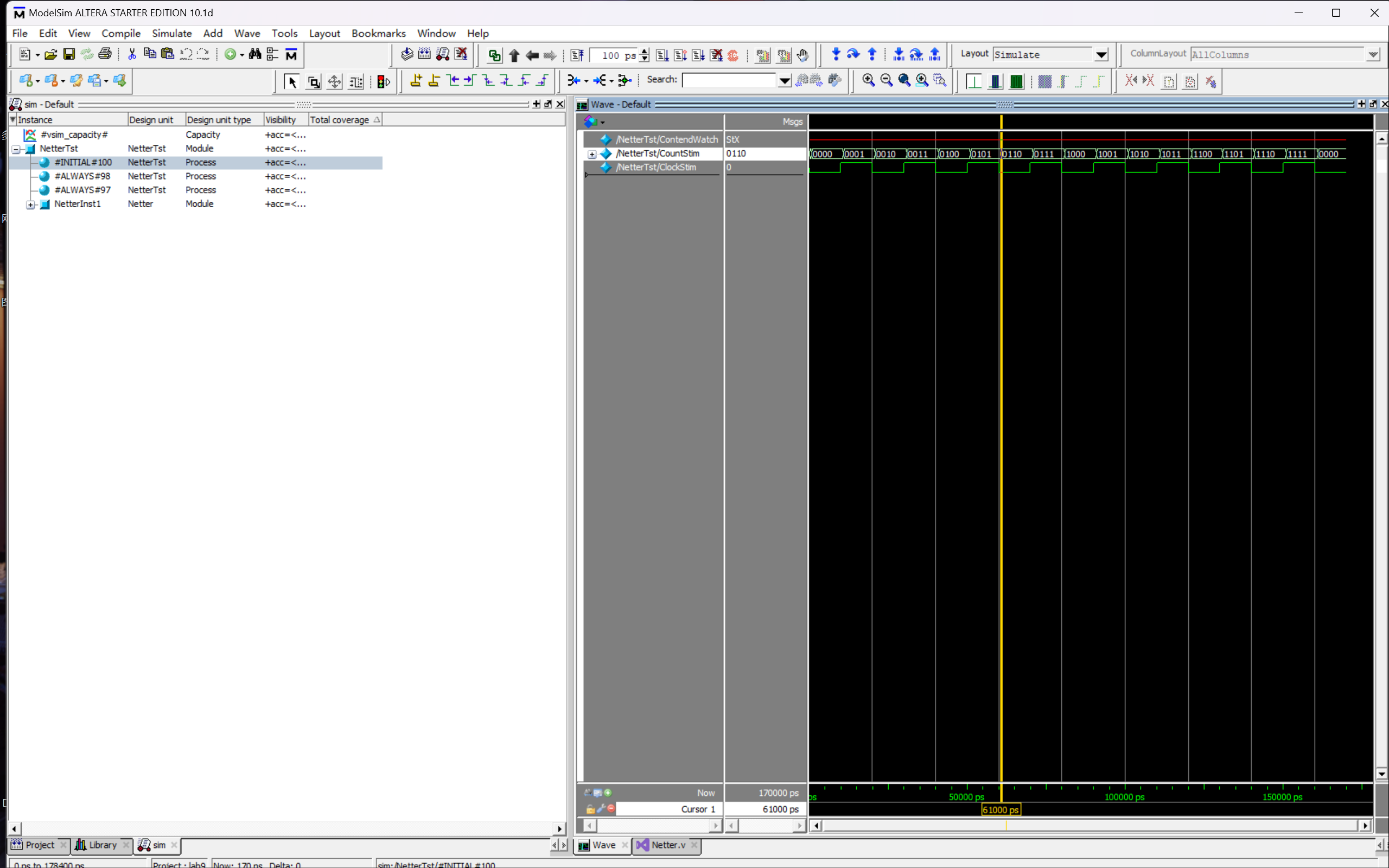
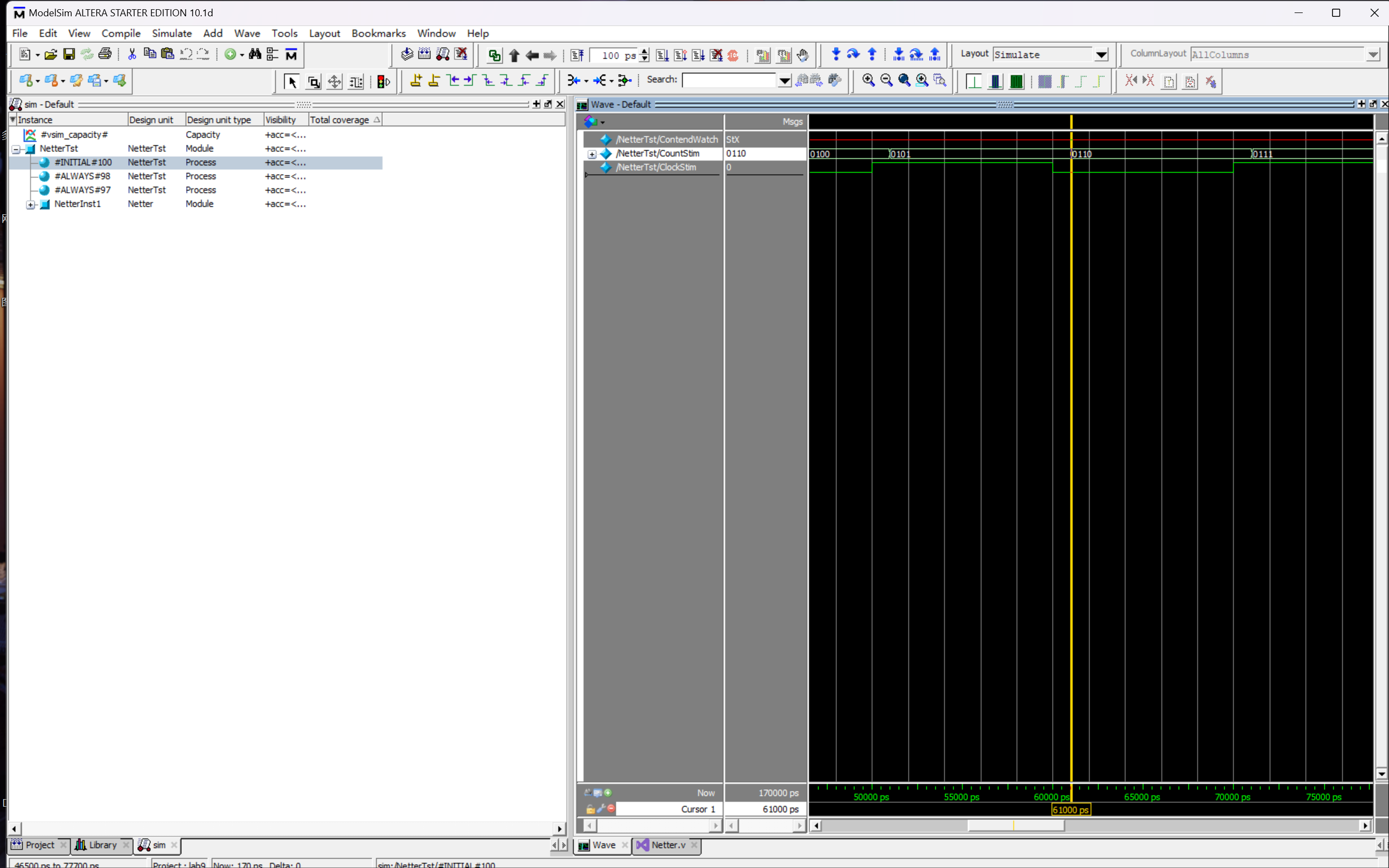
Step1-5：

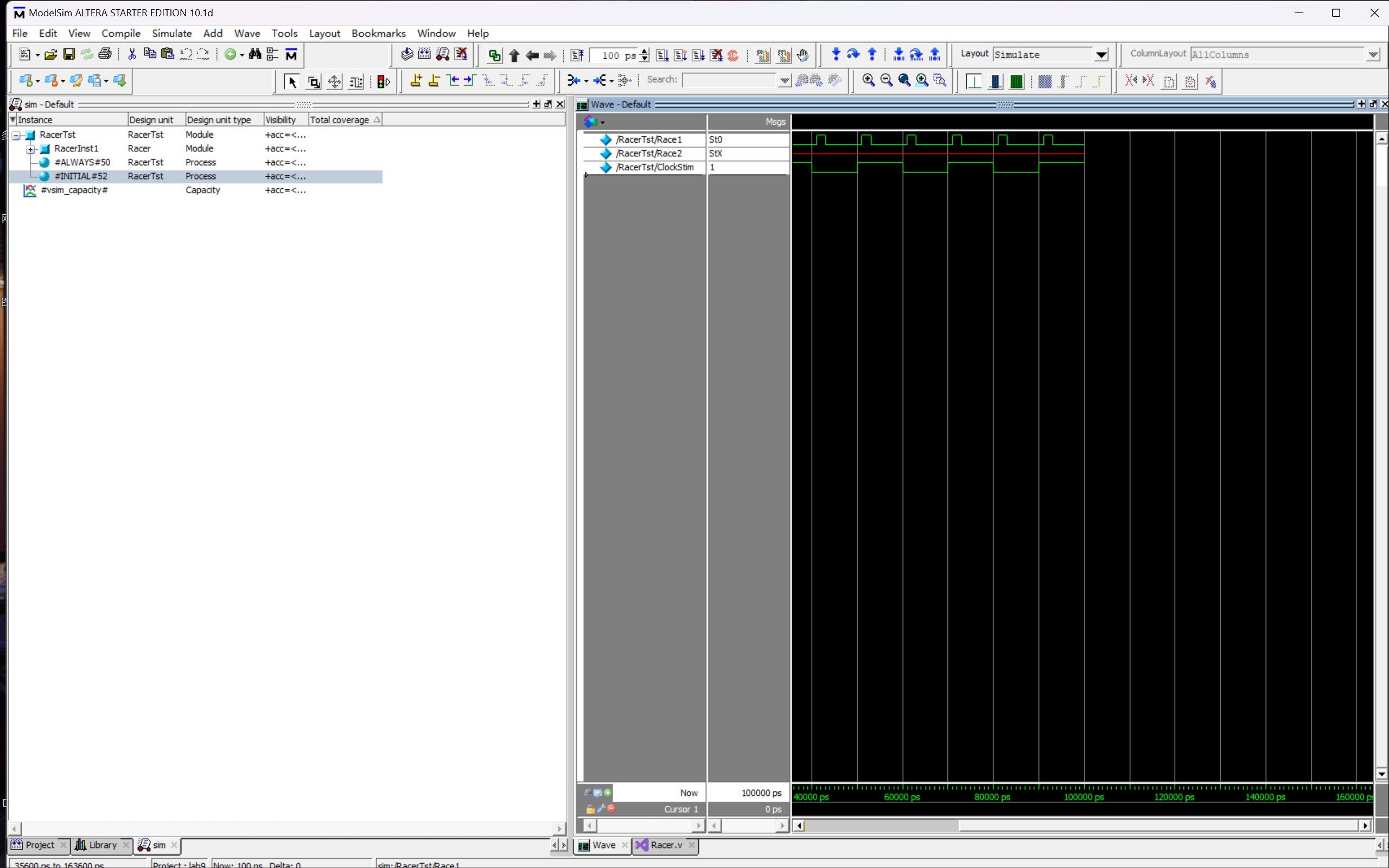
always@(ClockStim) #10 ClockStim <= ~ClockStim;%20ns为一个时钟周期



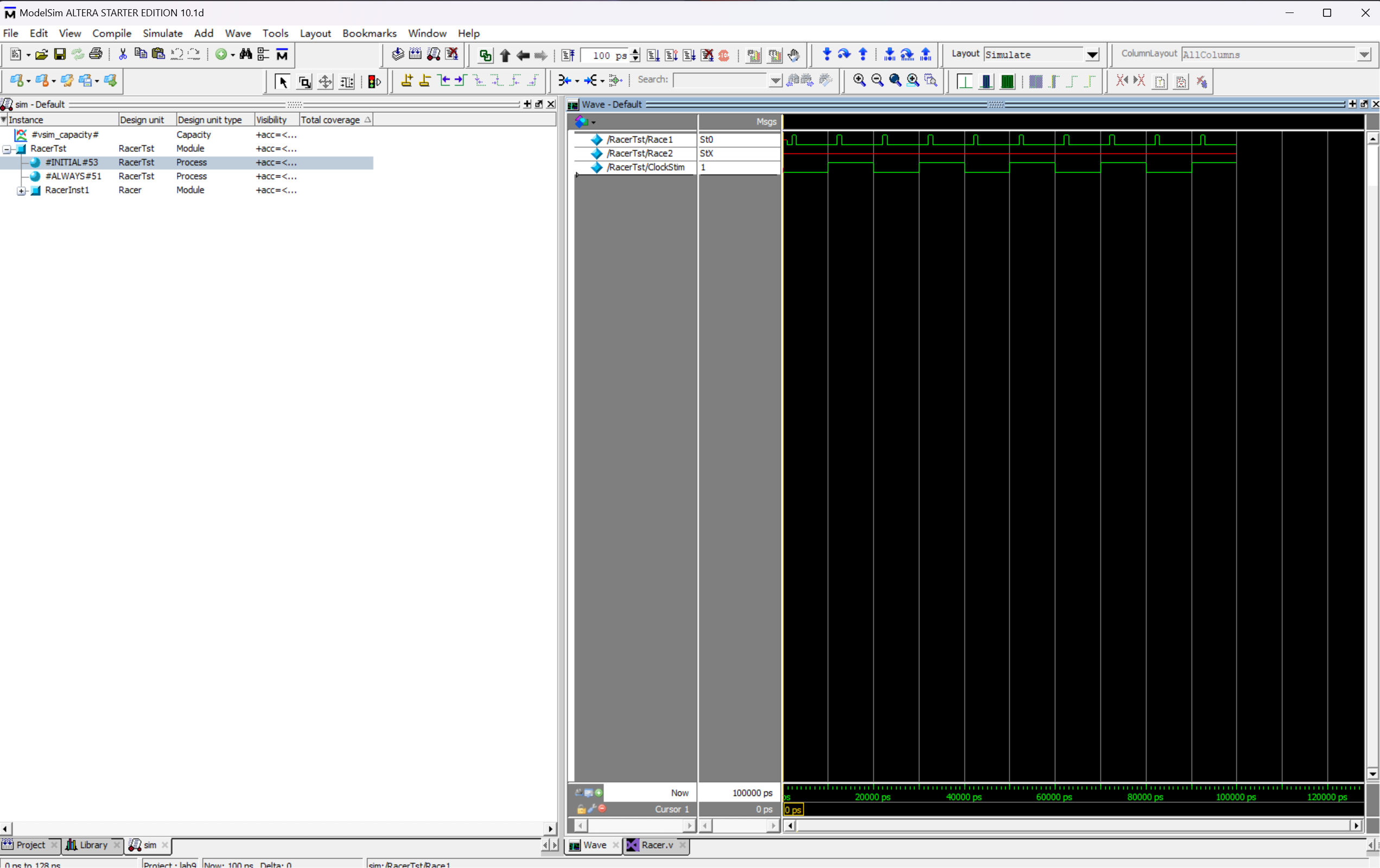
门器件的输出具有延迟，这些信号仍然只产生了组合逻辑。

注意关系表达式对于不确定性处理：关系表达式会把x理解成非真也非假，既不是0也不是1，x或z会导致结果不能为真，此时的if，else语句就不起作用了。

Step6：

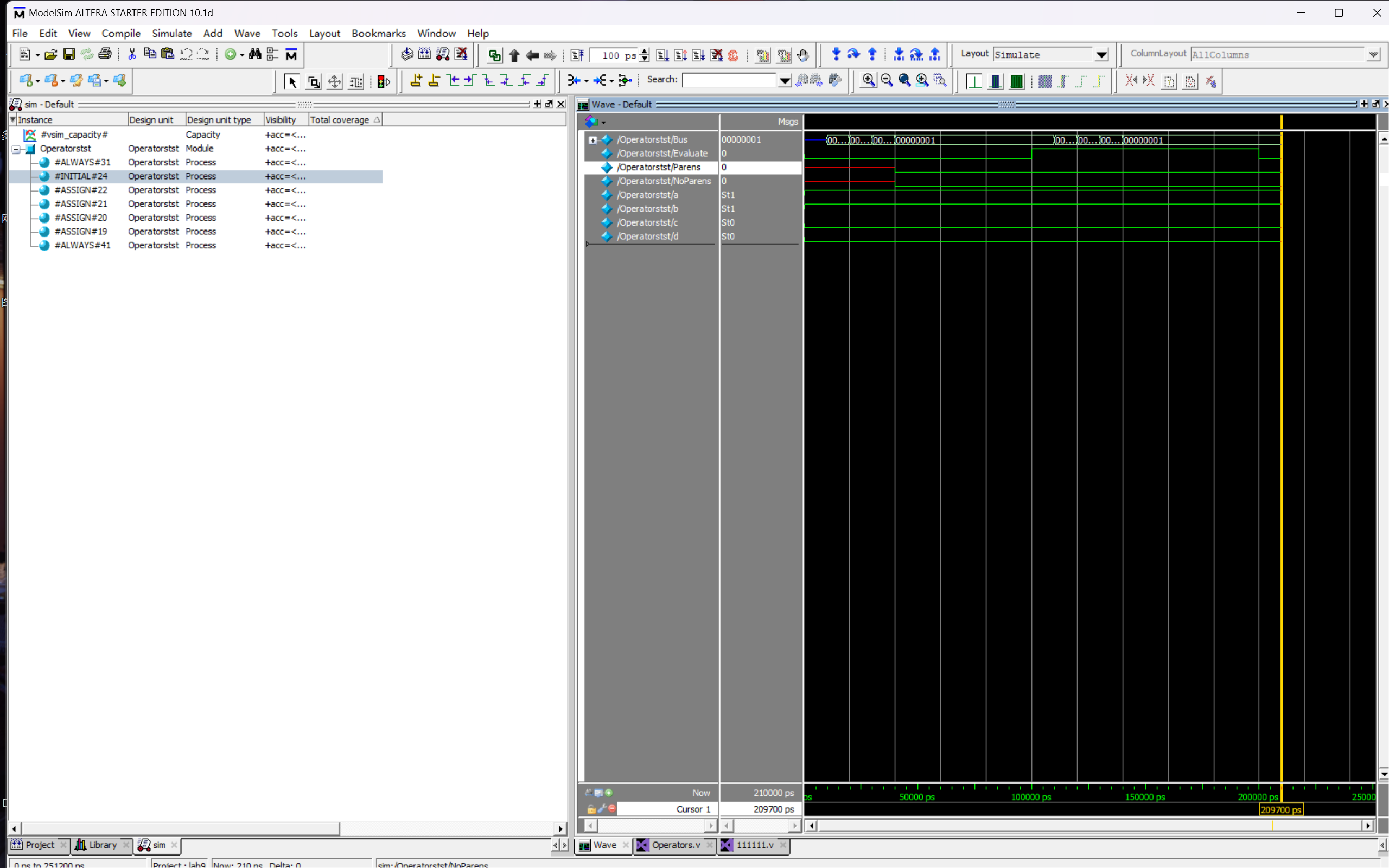


原顺序：由于产生DoPulse边沿后立即产生RaceReg脉冲，可能导致RaceReg第一次被赋值为0之前RaceReg就已经发生了翻转，从而旧值被0代替。



改变两个always块顺序后：首先发生RaceReg的翻转，然后在DoPulse边沿上产生脉冲，这样能保证在RaceReg在第一次被赋值为0之前不会出现翻转，从而不会导致旧值被0代替。

Step7：



两个结果都是0

VCS和Questasim如何处理竞争：

VCS采用了“随机模拟”（Random Simulation）的方法来处理竞争。在随机模拟中，仿真工具对于同时发生的事件的执行顺序是随机的，这可以在一定程度上模拟出硬件设计中的异步和并发行为。当存在竞争时，VCS可能会在不同的仿真运行中产生不同的结果。这是因为在不同的仿真运行中，随机模拟的执行顺序可能不同。

为了提高仿真效率，VCS通常使用了一些优化和近似技术，这些技术可能会导致竞争的结果在某些情况下与实际硬件行为有所不同。因此，在使用VCS进行仿真时，特别是对于涉及到竞争的设计，需要进行仔细的验证和调试。

QuestaSim采用了“确定性模拟”（Deterministic Simulation）的方法来处理竞争。在确定性模拟中，仿真工具对于同时发生的事件的执行顺序是确定的，这意味着在每次仿真运行中，事件的执行顺序都是相同的。这种方法可以更好地精确地模拟硬件设计中的行为，尤其对于处理竞争情况时会更加可靠。

由于QuestaSim采用了确定性模拟，因此在处理竞争时，其仿真结果通常在不同的仿真运行中是一致的。这为设计验证和调试带来了更多的可靠性。

尽管两个并行的语句对同一个信号赋值相同的逻辑电平，竞争仍然可能导致不确定的结果。因此，在设计硬件时，应该特别注意并发事件之间的竞争问题，并采取适当的同步措施来避免潜在的问题。

在逻辑运算中，"等于"操作符通常用于比较两个值是否相等，而"全等"操作符用于比较两个值是否相等且类型相同。在处理 "x"（未知值）时，"等于"和"全等"操作在结果上是不同的。

等于操作符 "==": 当使用 "==" 进行比较时，任何未知值 "x" 都会被视为不等于任何其他值，包括另一个 "x"。所以，如果有一个或多个操作数是 "x"，那么 "==" 操作的结果将始终是 "0" (不相等)。

例如：

(1'b1 == 1'b0) 的结果是 0

(1'b0 == 1'bx) 的结果是 0

(1'bx == 1'bx) 的结果是 0

全等操作符 "===": 当使用 "===" 进行比较时，只有两个操作数都是 "x" 时，结果才是 "1" (相等)。否则，任何一个操作数是 "x"，结果都是 "0" (不相等)。

例如：

(1'b1 === 1'b0) 的结果是 0

(1'b0 === 1'bx) 的结果是 0

(1'bx === 1'bx) 的结果是 1