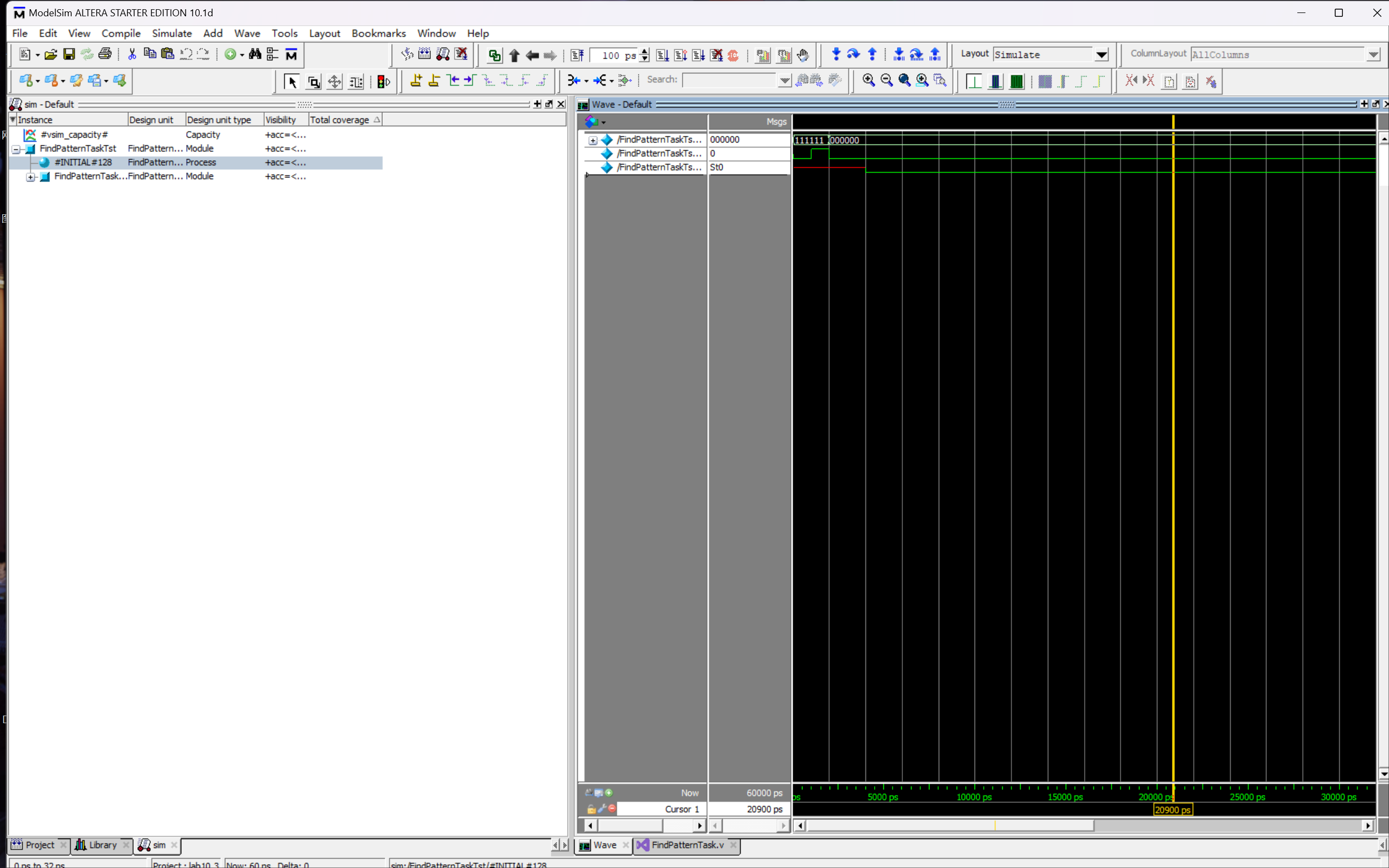
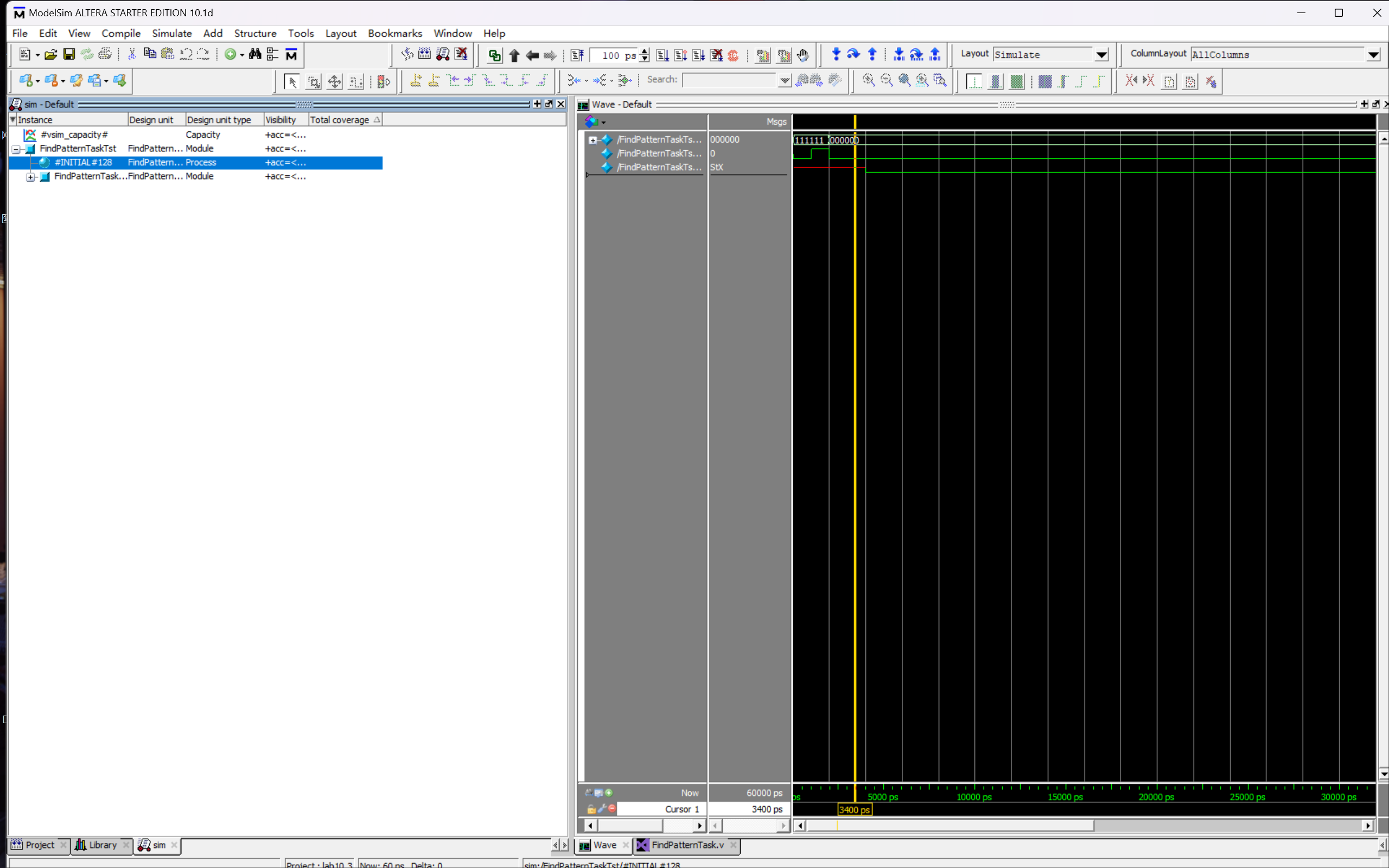
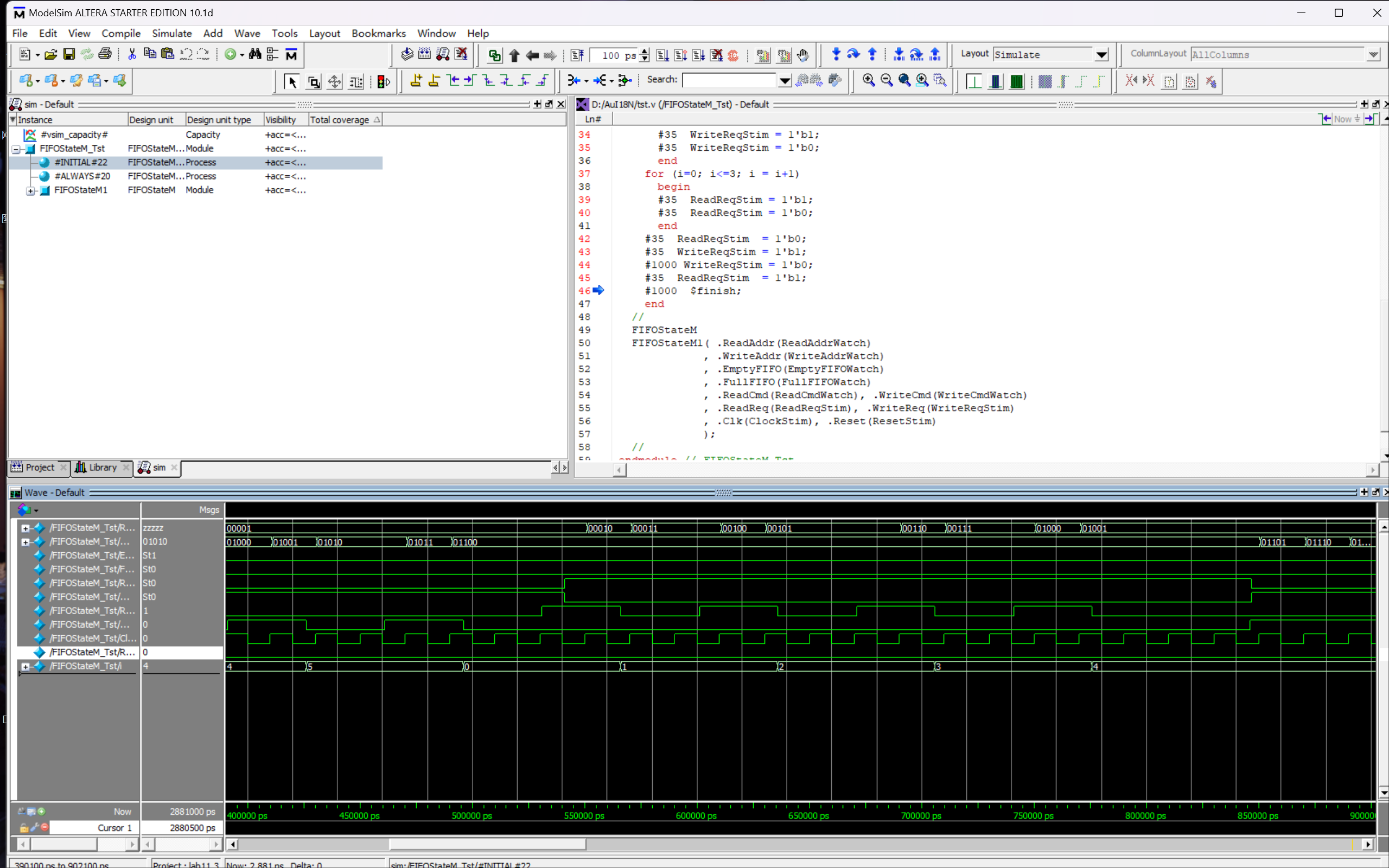
第一步：



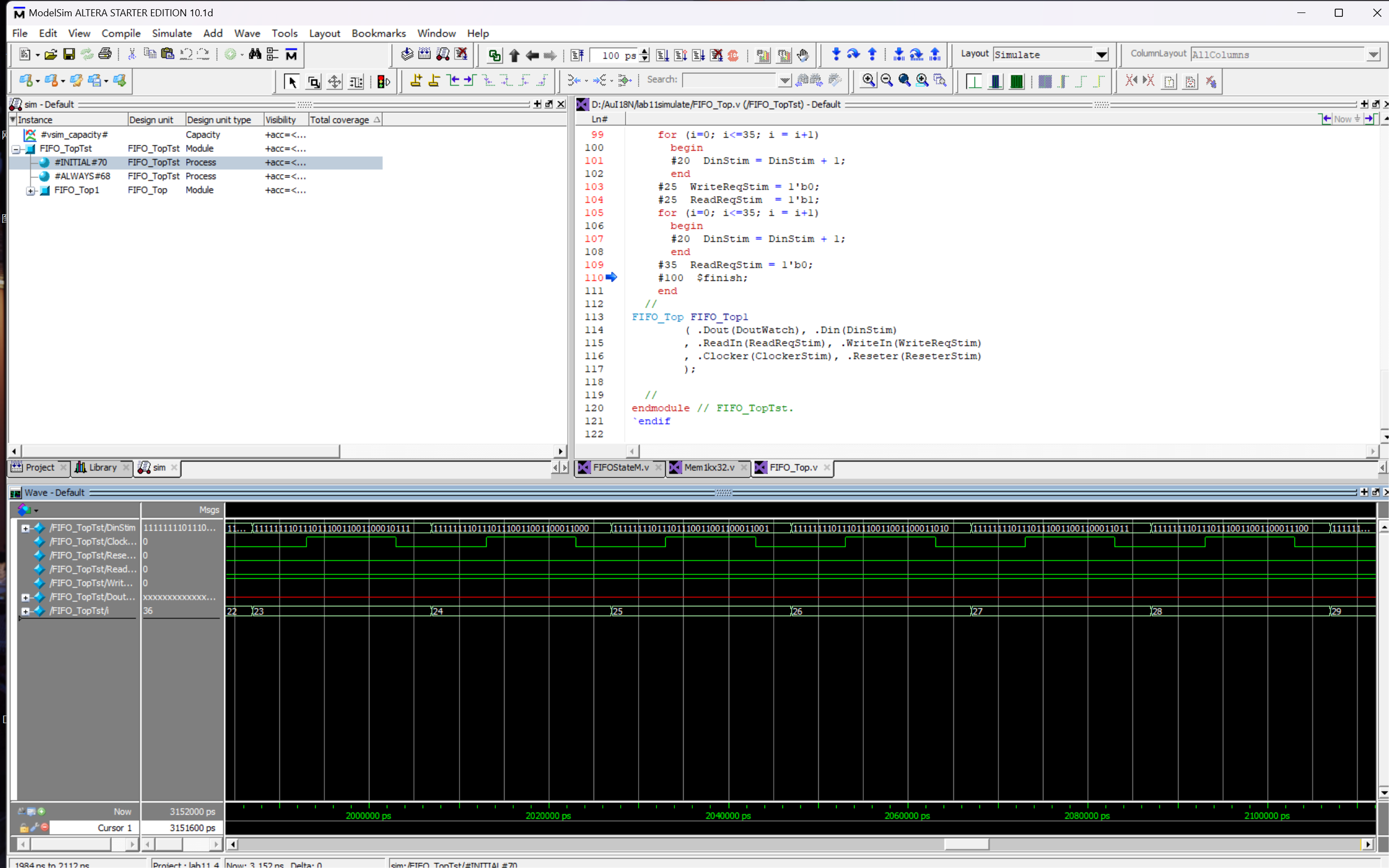
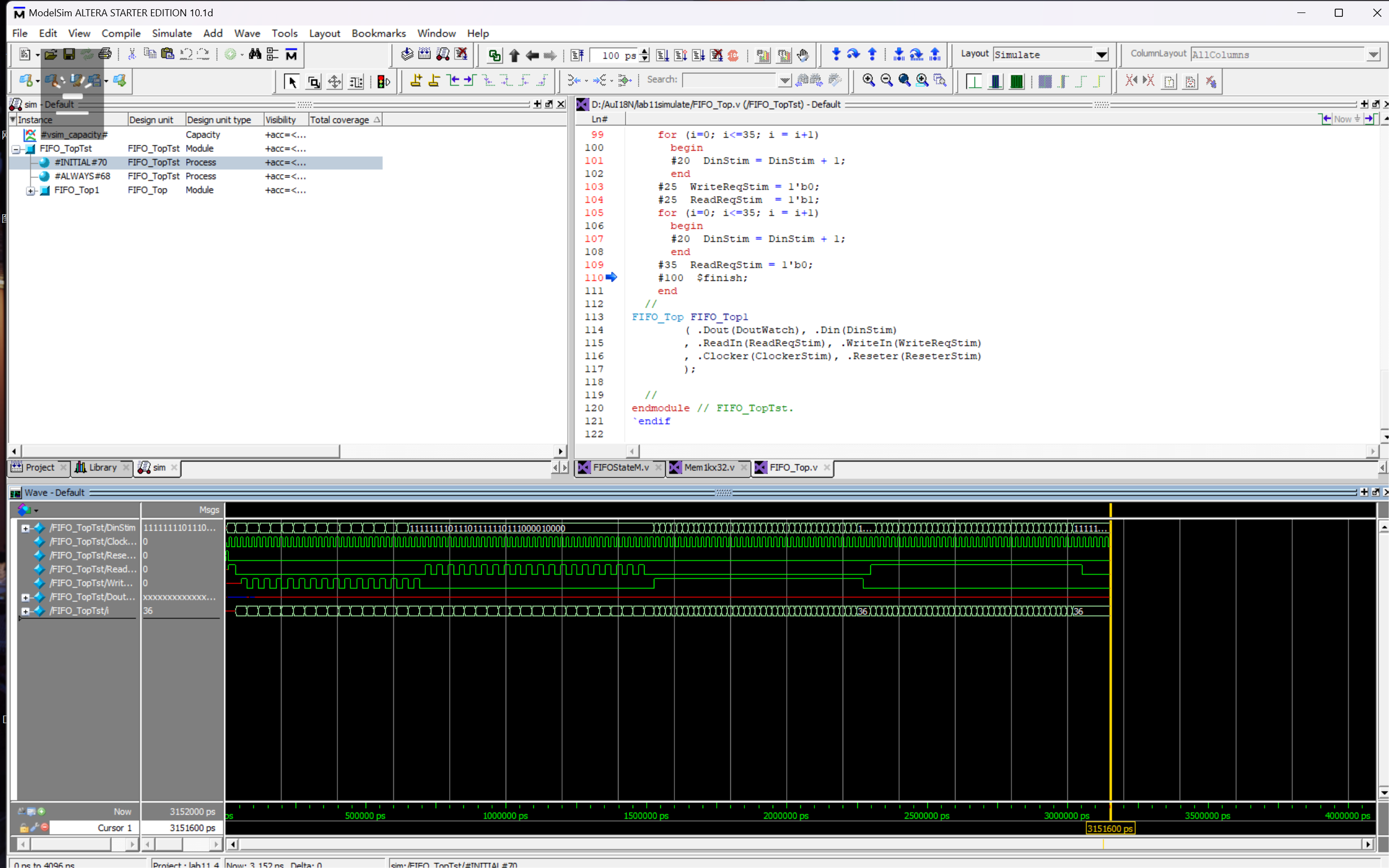
第二步：  


可以看到波形并没有改变

第三步:



第四步：



如何判断FIFO工作状态：

设定仿真时钟：为了模拟实际情况，设置仿真时钟信号。确保时钟频率和时钟周期与实际应用中的时钟一致。

提供输入数据：为FIFO提供不同的输入数据序列，包括写入数据和读取数据。可以使用测试向量或随机数生成器来产生输入数据。

控制FIFO操作：在测试平台中，使用控制信号来控制FIFO的写入和读取操作。例如，写入数据时将写使能信号置为高，读取数据时将读取使能信号置为高。

仿真运行：运行仿真，并观察FIFO的输出。检查输出数据是否按照FIFO的设计规格正确输出。

结果验证：对比FIFO的输出数据与预期的数据是否一致。如果FIFO能够按照预期工作并满足设计规格，说明设计是正确的。

通过观察可以看出FIFO能够正确地写入和读取数据，并且正确地更新空和满标志，FIFO的设计是有效的。

跨时钟域的FIFO（First-In-First-Out）设计和格雷码之间的关系：

跨时钟域的FIFO（First-In-First-Out）设计和格雷码之间在设计中没有直接的关系。它们是两个独立的概念，用于不同的设计需求。

跨时钟域的FIFO设计：跨时钟域的FIFO设计用于在不同时钟域之间进行数据传输和缓冲。当两个模块运行在不同的时钟频率下时，数据传输之间可能会出现时序问题，例如数据突发或者数据丢失。为了解决这些时序问题，可以使用FIFO来缓冲数据，使得数据在不同时钟域之间的传输更稳定和可靠。

格雷码：格雷码是一种二进制编码方式，在相邻的数值之间只有一个位的变化。格雷码常用于减少数字转换时的误差，特别是在数字信号传输中，避免由于二进制数位转换引起的抖动或震荡。格雷码还广泛应用于数字编码器和旋转编码器等应用中。

虽然跨时钟域的FIFO设计和格雷码没有直接的关系，但在某些应用中，你可能会同时使用它们。例如，在数据传输中，你可能使用格雷码来编码数据以减少误差，并且使用跨时钟域的FIFO来处理时序问题，确保数据在不同时钟域之间的稳定传输。两者可以在不同的层面上为设计提供互补的功能，但它们并不是一个因果关系。