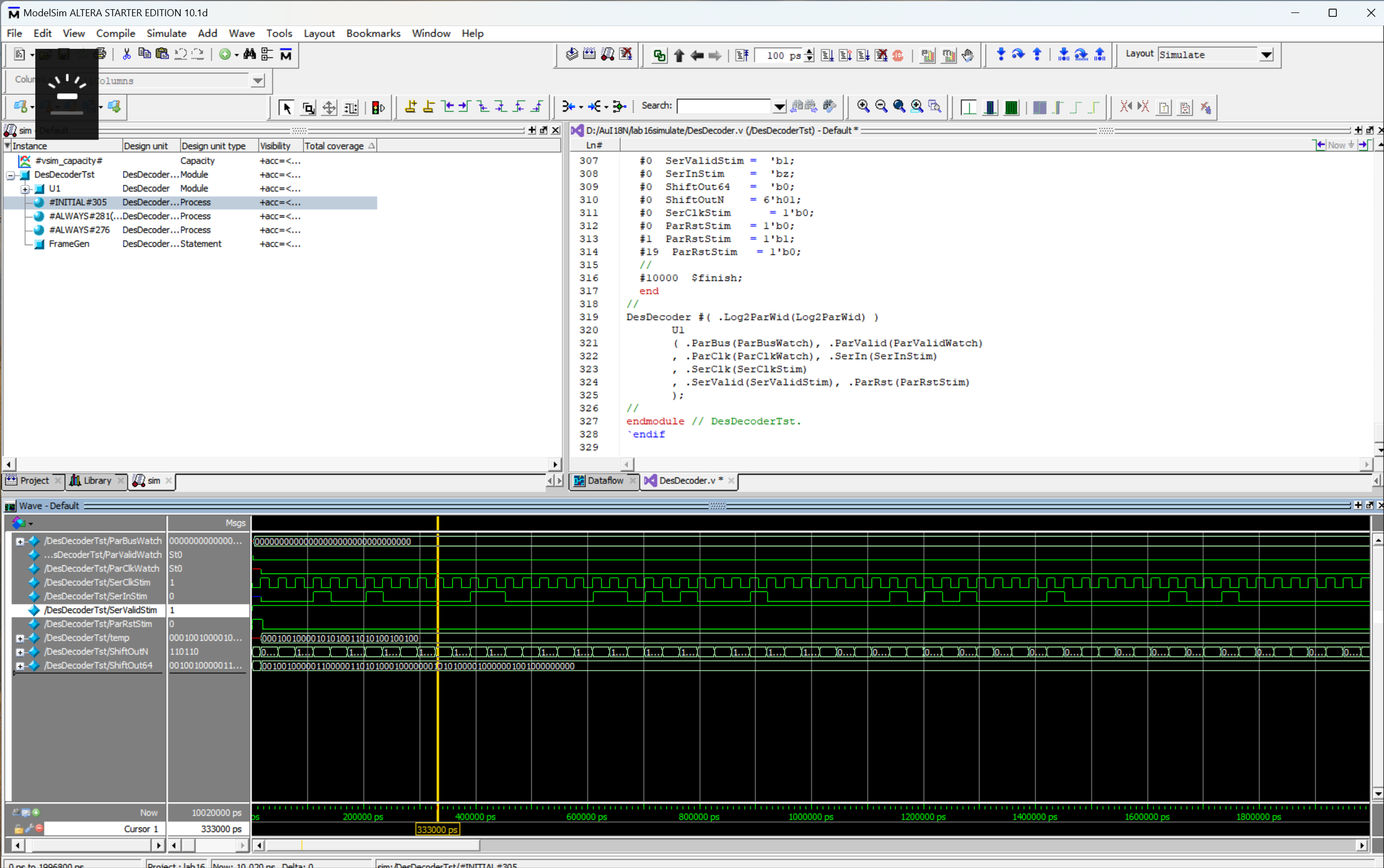
第一步：



模块参数:

Log2ParWid 是一个表示并行数据总线宽度的参数的基于2的对数。

ParHi 是从 Log2ParWid 计算得到的，表示并行数据总线的最高索引。

模块端口:

ParBus、ParValid 和 ParClk 是并行总线的输出信号。

SerIn、SerClk、SerValid 和 ParRst 是串行输入信号。

内部信号:

FrameSR: 一个用于存储输入串行数据的移位寄存器。

其他信号如 DecodeReg、ParBusReg、ParValidReg 等，用于存储中间结果和控制信号。

任务:

Unload32: 一个任务，将 DecodeReg 寄存器的内容复制到并行输出总线 ParBus。它还控制 ParValid 信号。

连续赋值 (assign 语句):

对 ParBus、ParClk 和 ParValid 进行赋值，值来自对应的寄存器。

时钟生成 (always 块):

ClkGen: 基于复位条件生成串行时钟 SerClk 的门控版本。

行为块 (always 块):

Decode4: 使用一个串行时钟周期的延迟处理串行数据。

Shift1: 使用一个串行时钟周期的延迟对 FrameSR 移位寄存器进行移位。

Unload32: 基于 UnLoad 信号控制 Unload32 任务的行为。

DesDecoderTst 测试台

模块参数:

Log2ParWid 和 ParHi 的值与 DesDecoder 模块中的相同。

内部信号:

SerClkStim、SerInStim、SerValidStim、ParRstStim：用于为 DesDecoder 模块生成刺激信号。

temp、ShiftOutN、ShiftOut64：用于生成刺激并进行移位操作的临时信号。

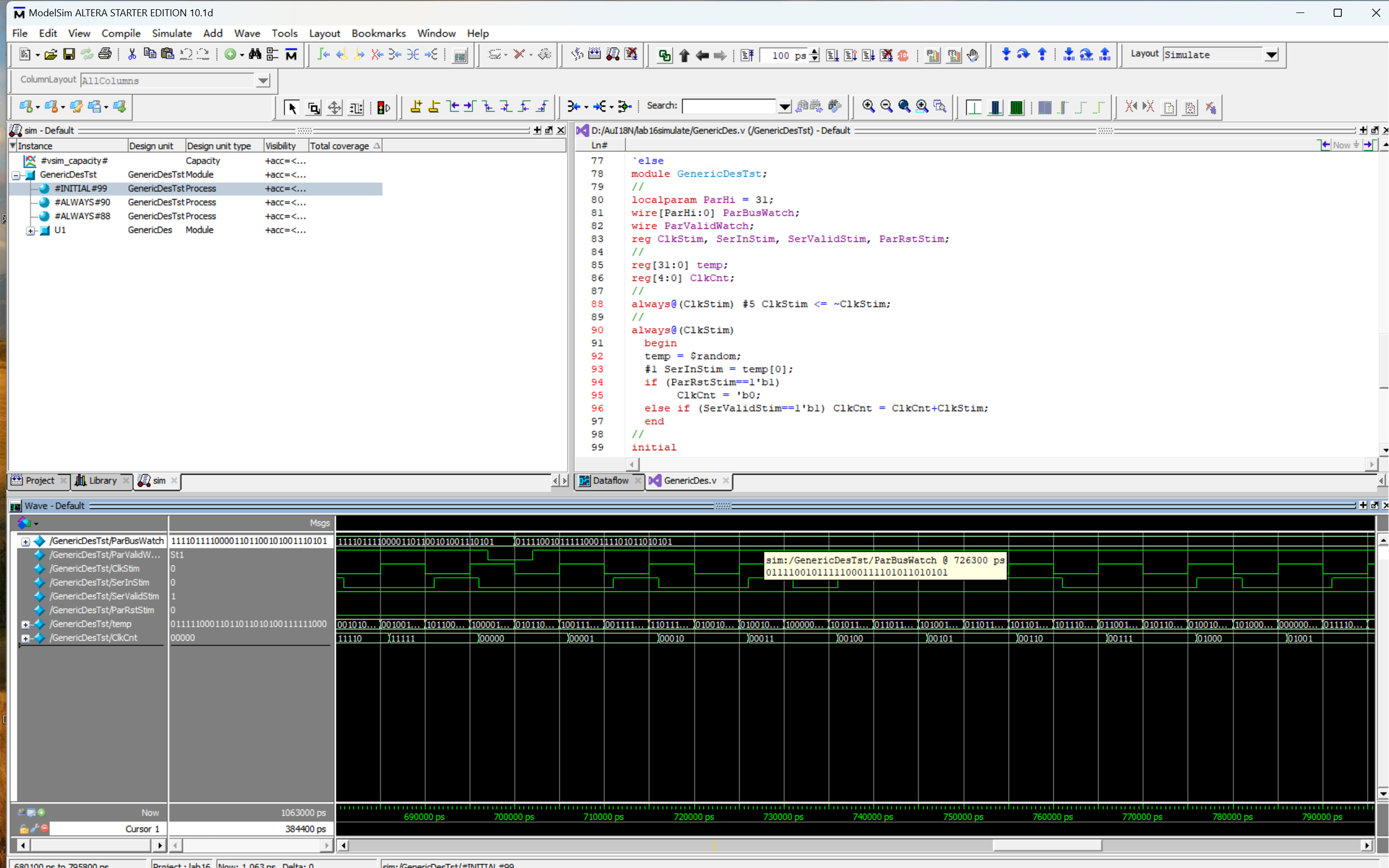
时钟生成 (always 块):

生成模拟串行时钟信号 (SerClkStim)，频率约为 32 MHz。

帧格式化 (always 块 - FrameGen):

该块为串行输入生成刺激，并测试 SerValid 信号的行为。它从 ShiftOut64 移位并将数据送入串行输入线。

第二步：



模块参数：

ParHi：并行数据总线的最高索引。默认为31。

模块端口：

ParBus：并行数据总线输出。

ParValid：并行数据有效信号输出。

SerIn：串行输入数据。

SerValid：串行数据有效信号。

ParClk：并行时钟信号。

ParRst：并行复位信号。

内部信号：

ParSR：输入移位寄存器。

ParBusReg：驱动并行数据总线的寄存器。

ParValidReg：并行数据有效信号的寄存器。

Unload32 任务：

将并行移位寄存器的内容复制到总线上。

在指定的时序内进行操作，首先将 ParValidReg 置为低，然后将 ParBusReg 赋值为 ParSR，清除 ParSR，最后将 ParValidReg 置为高。

Shift1 函数：

实现一个移位函数，将输入的旧移位寄存器左移一位，并将新的比特位放入最低位。

运行时行为 (assign 语句)：

用于在一个模拟时间单位后将寄存器的值赋给输出。

always 块：

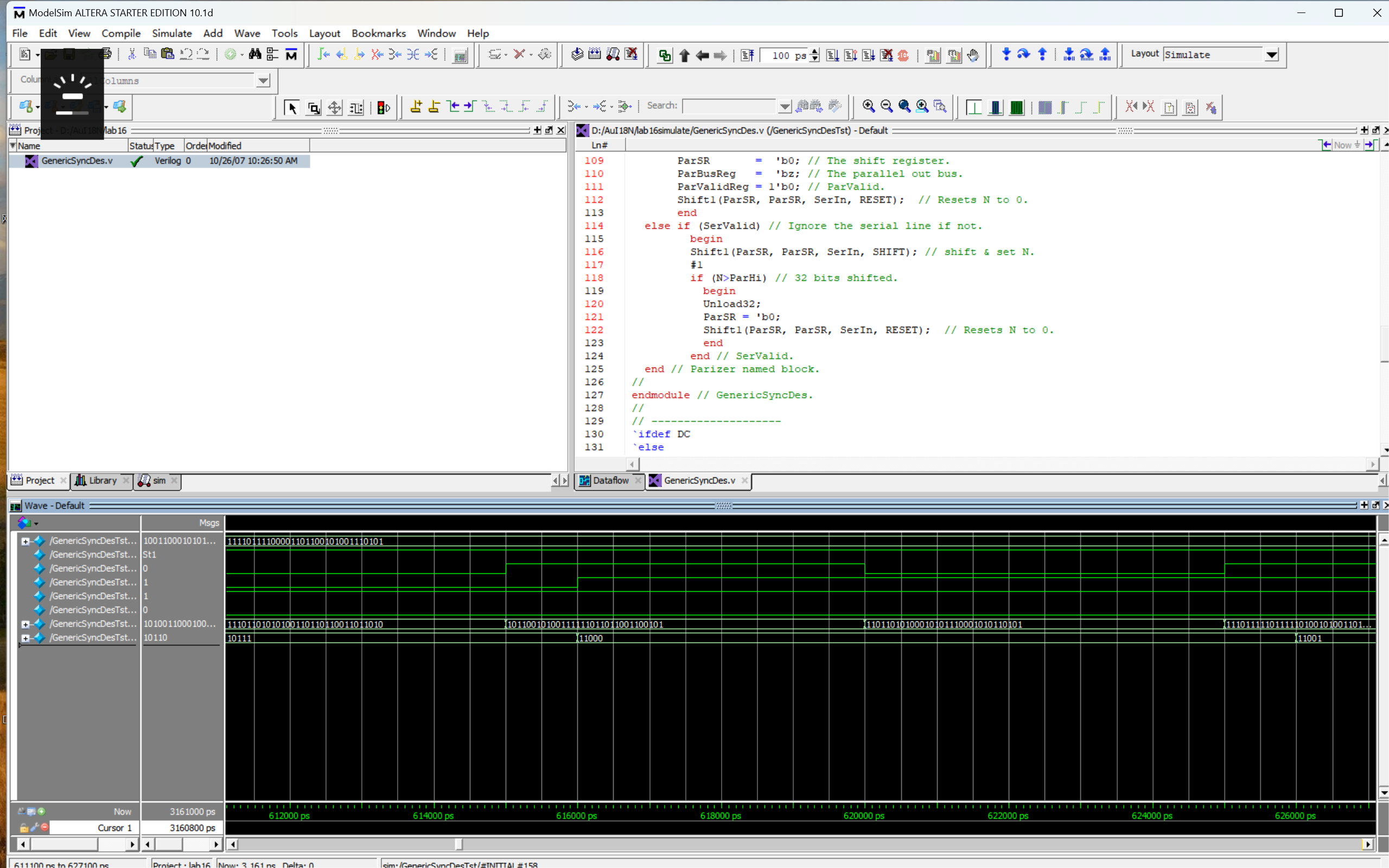
在上升沿的时钟和复位条件下执行移位操作。

如果复位信号为高，则进行初始化操作，将计数器和寄存器重置为初始状态。

如果串行数据有效 (SerValid)，则将输入数据放入移位寄存器中，并更新计数器 (N)。

当计数器达到 ParHi（指定的并行总线宽度）时，调用 Unload32 任务进行数据传输，然后重置计数器和移位寄存器。

第三步：



模块参数：

ParHi：并行数据总线的最高索引。默认为31。

模块端口：

ParBus：并行数据总线输出。

ParValid：并行数据有效信号输出。

SerIn：串行输入数据。

SerValid：串行数据有效信号。

ParClk：并行时钟信号。

ParRst：并行复位信号。

内部信号：

ParSR：输入移位寄存器。

ParBusReg：驱动并行数据总线的寄存器。

ParValidReg：并行数据有效信号的寄存器。

N：用于保存位移计数的寄存器。

Unload32 任务：

将并行移位寄存器的内容复制到总线上，与之前的解释相同。

Shift1 任务：

实现一个移位任务，将输入的旧移位寄存器左移一位，并将新的比特位放入最低位。

如果满足特定的同步模式（StopPattern 或 StartPattern），会进行特殊处理。

在这里，任务会根据同步模式进行移位、重置位移计数 N，以及保存/恢复上下文状态。

运行时行为 (assign 语句)：

同之前的解释，用于在一个模拟时间单位后将寄存器的值赋给输出。

always 块：

在上升沿的时钟和复位条件下执行操作。

如果复位信号为高，则进行初始化操作，将计数器和寄存器重置为初始状态，并执行 Shift1 任务以重置位移计数 N。

如果串行数据有效 (SerValid)，则执行 Shift1 任务进行数据移位，设置位移计数 N，并在满足条件时调用 Unload32 任务。