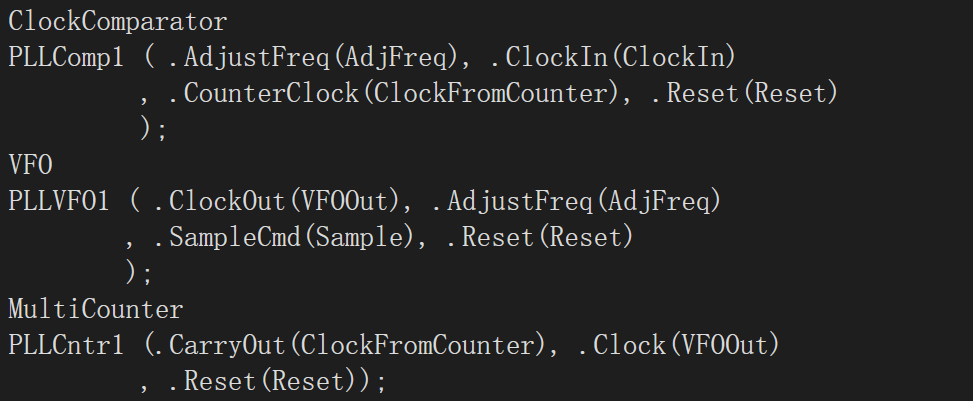
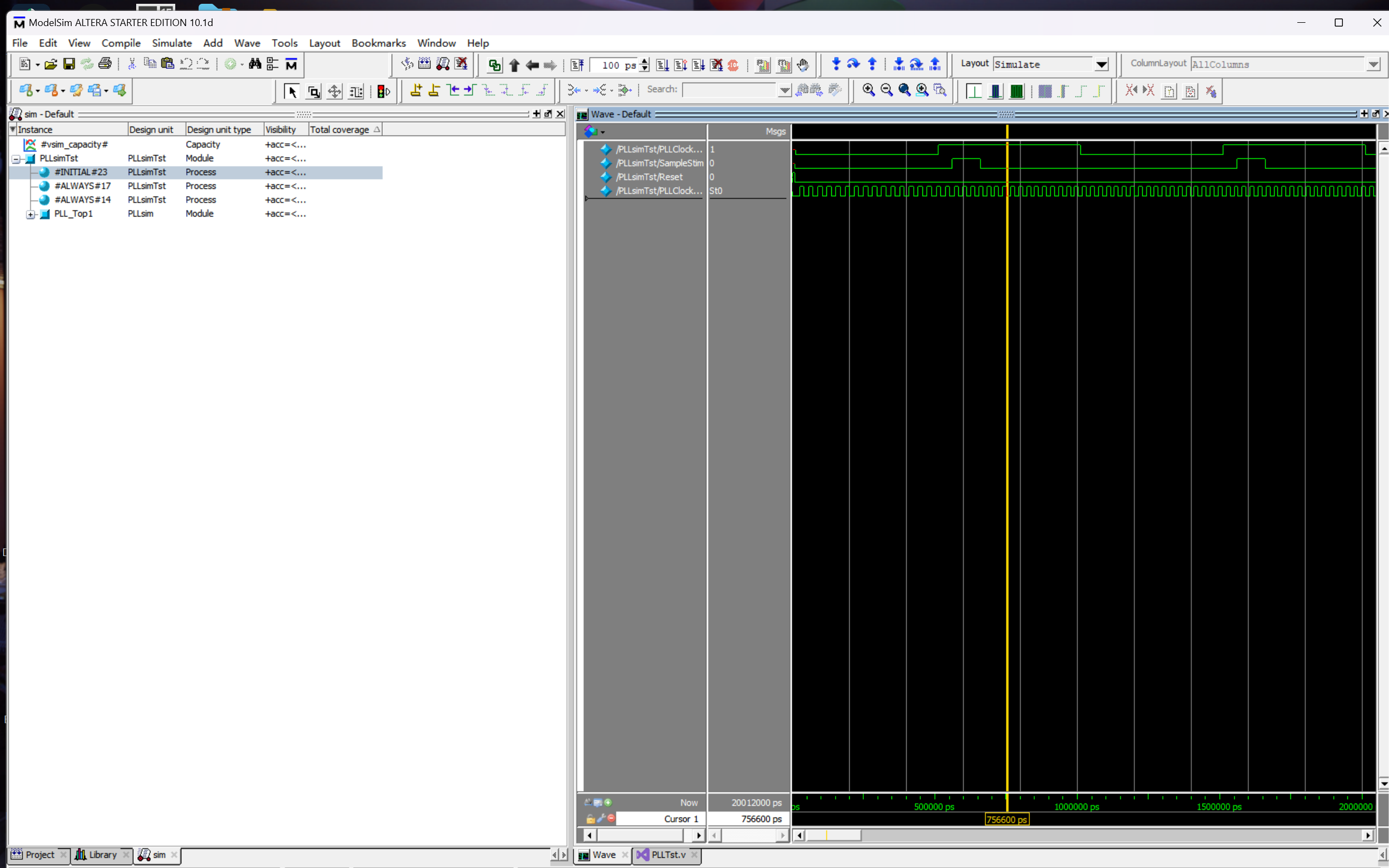
第一至五步：完成PLLsim模块里的端口连线





由图像可以看出VFO准确无误的产生了时钟

由宏的定义可知，PLLClockStim的频率是500.0ns，则其周期是1us。

注意事项：

①代码中使用的宏都要在PLLsim.v文件中进行定义，否则会报错undefined

②PLLsim.v必须放在和设计文件同样深度但不是同一个文件夹中，否则会报错找不到该文件

第六到七步：串行帧编码器。

串行帧编码器是一种数字电路，用于将并行数据转换为串行数据并进行编码，以便在传输过程中进行有效的数据传输。其工作原理如下：

并行输入：串行帧编码器接收并行输入数据，通常由多个数据位组成，例如8位或16位等。

帧开始标记：编码器开始接收并行数据时，它需要知道何时开始处理新的数据帧。因此，在传输开始时，通常会有一个帧开始标记，告诉编码器新的数据帧已经开始。

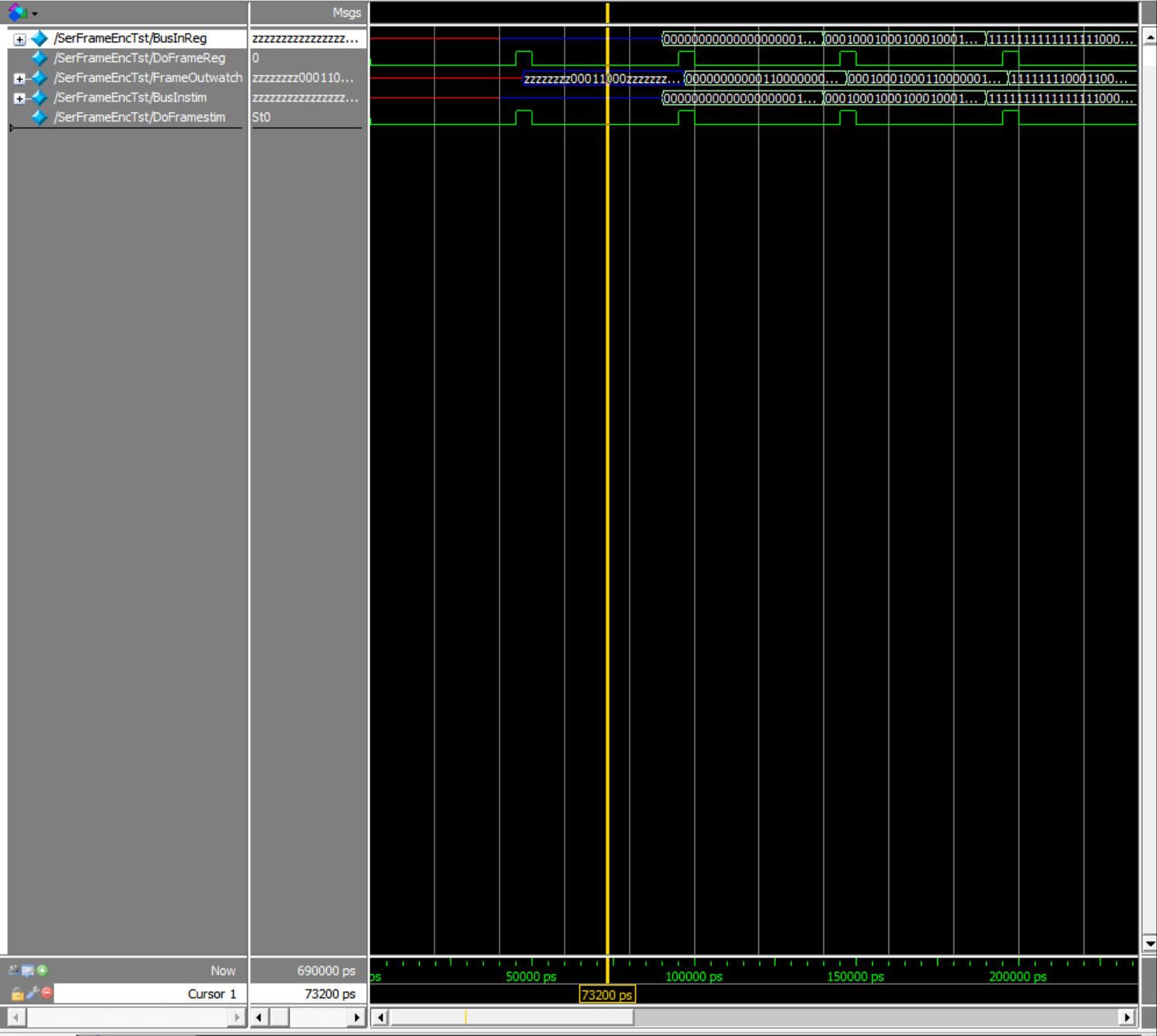
编码：编码器将并行输入数据转换为相应的串行数据，并对其进行编码。编码过程可以根据应用的要求进行，常见的编码方式包括差分曼彻斯特编码、8B/10B编码等。

传输：编码后的串行数据通过传输线路发送到接收端。

帧结束标记：在数据帧的末尾，通常会有一个帧结束标记，告诉接收端当前数据帧已经传输完成。

接收端解码：接收端接收到串行数据后，会进行解码操作，将串行数据转换回并行数据，以恢复原始的并行数据。

串行帧编码器的工作原理确保了数据在传输过程中的有效性和可靠性。它在许多通信系统和数据传输中都得到广泛应用，例如以太网、串行通信、串行总线等。编码过程可以使数据传输更可靠，减少误码率，并在传输中提供时钟同步。



代码解析：

模块定义：SerFrameEnc 是一个模块的定义，它有两个参数 InHi 和 OutHi，用于指定输入和输出数据的位宽。还有两个本地参数 DatHi 和 PadHi，用于指定数据和填充字节的位宽。Pad3、Pad2、Pad1 和 Pad0 是定义的填充字节的值。

模块端口：FrameOut 是输出端口，用于输出编码后的串行数据。BusIn 是输入端口，用于接收并行输入的数据。DoFrame 是控制信号，当它的上升沿到来时，开始进行编码。

数据存储：模块内部定义了一个 FrameReg 寄存器，用于存储编码后的数据。

编码过程：在 always@(posedge DoFrame) 块中，当 DoFrame 上升沿到来时，开始进行编码。编码的过程是将并行输入 BusIn 的数据切分成不同的字节，并在字节之间插入填充字节，最终得到串行输出 FrameOut。

输出串行数据：通过 assign #2 FrameOut = FrameReg; 将编码后的串行数据赋值给输出端口 FrameOut。

总体来说，这个串行帧编码器的模块接收并行输入数据，对输入数据进行编码，并将编码后的串行数据输出。编码后的数据帧由填充字节分隔，并且在整个编码过程中只需要一个时钟边沿。