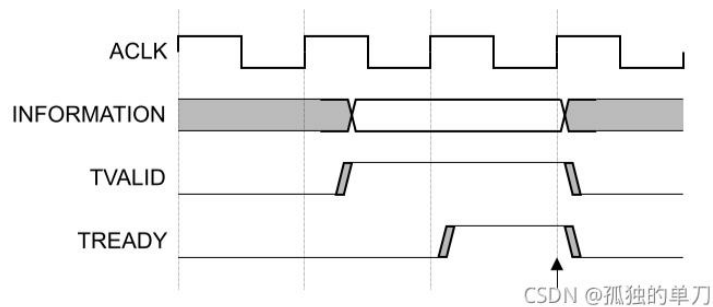


- AXI Stream 握手选择的是握手主机先于从机做好了发送数据的准备的情况，即 **valid** 等待 **ready** 拉高后在时钟上升沿再传递数据



- 激励信号全部随机产生，且均支持常量定义参数传递：

```
// 产生设置好size拍的data_in数据
always @(posedge clk or negedge rst_n)begin
    if(!rst_n)
        data_in <= 'd0 ;
    else if(ready_in && valid_in)
        data_in <= $random(seed);
    else
        data_in <= data_in ;
end
```

```
always @(*) begin
    if (ready_in && last_in != 1)
        keep_in <= {DATA_BYTE_WD{1'b1}};
    else if (ready_in )
        keep_in <= {DATA_BYTE_WD{1'b1}} << (($random) % DATA_BYTE_WD);
    else
        keep_in <= 'd0 ;
end
// last_in:最后一拍数据
assign last_in = (cnt == DATA_SEND_SIZE+1) ? 1: 0 ;
```

```
// 随机产生1拍header数据和随机的有效位信号
always @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        header_insert = 'd0 ;
        keep_insert = 'd0 ;
    end
    else if (valid_insert && ready_insert) begin
        header_insert = $random(seed) ;
        keep_insert = {DATA_BYTE_WD{1'b1}} >> (($random) % DATA_BYTE_WD);
    end
    else begin
        header_insert = header_insert ;
        keep_insert = keep_insert;
    end
end
```

• 不同位宽，帧长验证功正常，如：64 位宽 11 拍（图 1） 32 位宽 7 拍（图 2） 128 位宽 8 拍

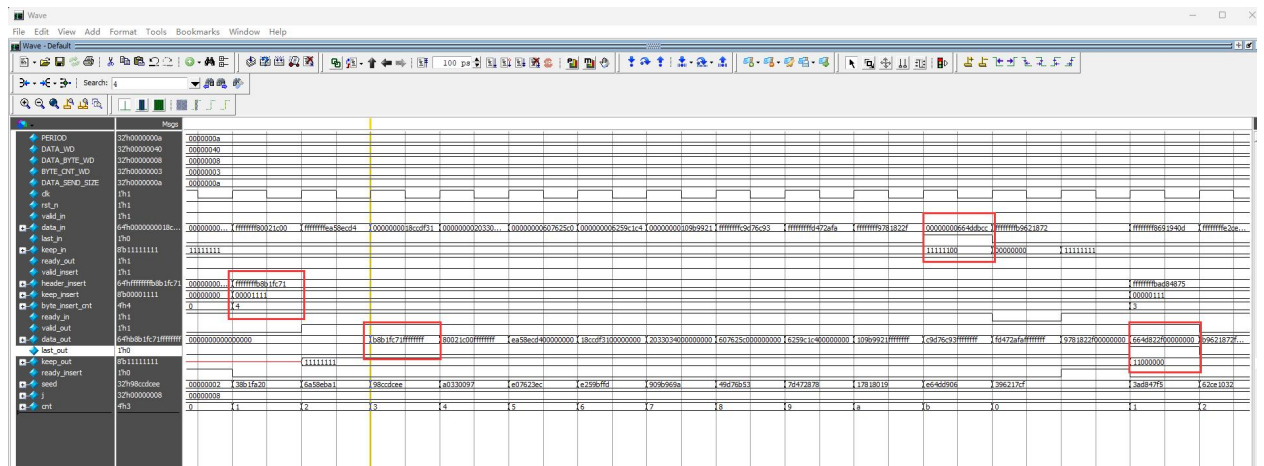


图 1 64 位宽 10 拍 head 后 4 字节有效 data 前 6 字节有效

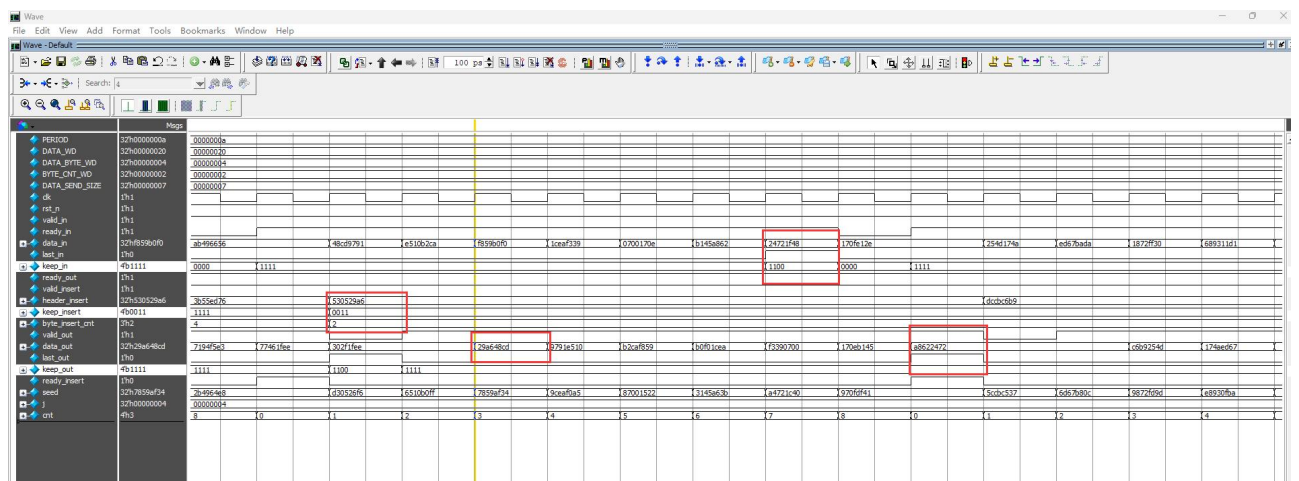


图 2 32 位宽 7 拍 head 后 2 字节有效 data 前 2 字节有效

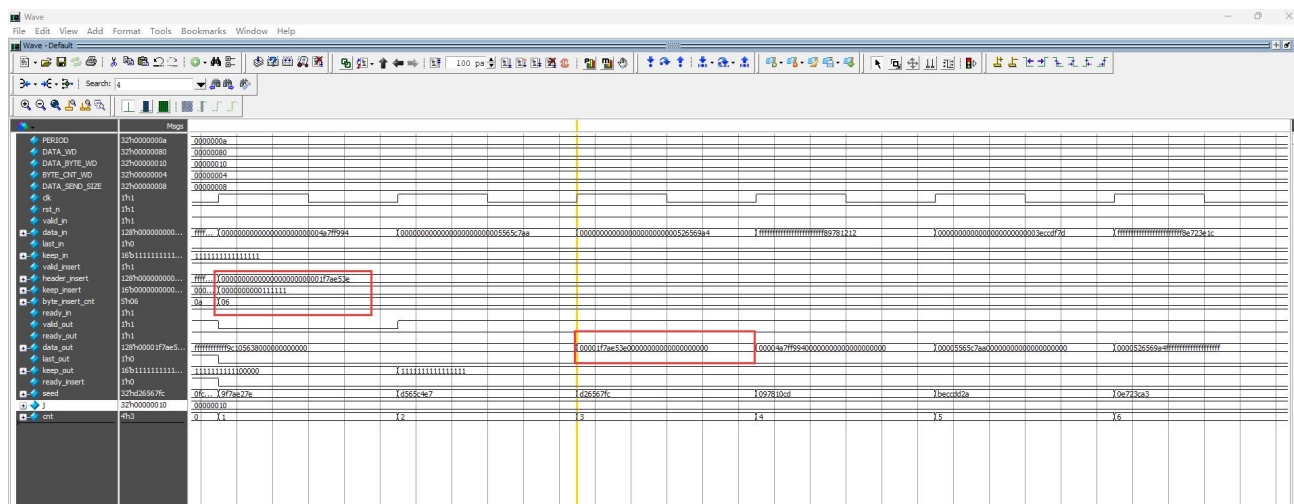
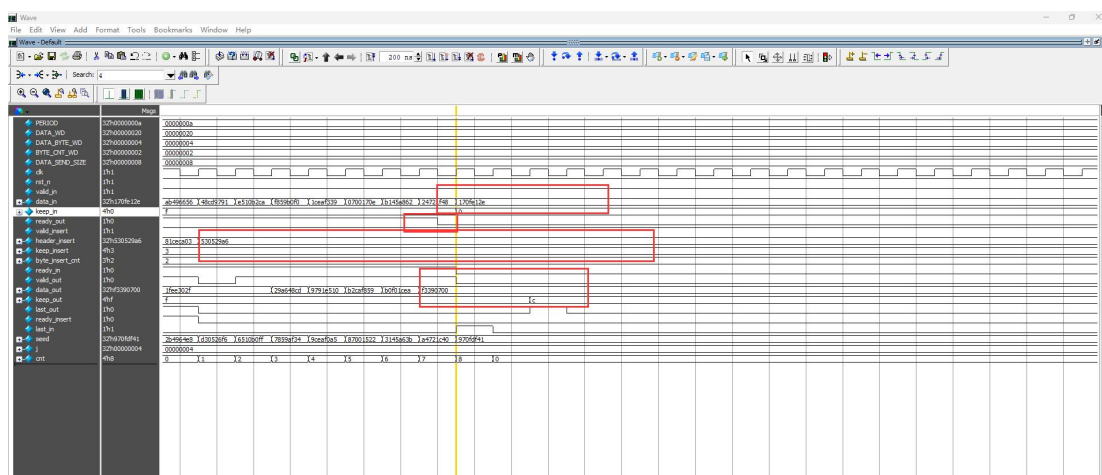


图3 128位宽 8拍 head后6字节有效 data前12字节有效

- 验证逐级反压，设置 ready\_out 信号在仿真中拉低一段时间后再拉高：



可见 data\_out 不再输出，且上一级不再有 data 和 head 传输

一段时间后拉高 ready\_out:

