

Changes to the OLT synchronizer to resolve problems with missing gearbox and the fact that bit errors could impair the append_inbuffer action.

Rename "BlockFromGearbox" function to "BlockFromPMA" at every occurrence.

Redefine BlockFromGearbox as follows:

BlockFromPMA

Function that accepts the received data from the PMA. Conceptually, this function serializes the ~~16-16-bit~~ rx_data_group<~~0..15~~15:0> to a bit stream at 10.3125_Gb/s, and then deserializes the resulting bit stream into a 66 bit wide rx_coded<~~065..:650~~> block of data. It does not return until 66 bits have been transferred. Note that the internal design by which this function is accomplished is an implementation choice; however, the design must operate such that a new ~~66-66-bit~~ block is made available at the regular interval of 6.4_ns, and the transfers are made synchronous to the XGMII clock.

Redefine SLIP_One_Bit as follows:

SLIP_One_Bit

Causes the next candidate block sync position to be tested. The next candidate must be exactly one bit later than the previous candidate - no burst alignments may be skipped. Following the conceptual model mentioned in "BlockFromPMA," this function transfers one more bit from the ~~16-bit~~ serializer to the ~~66-66-bit~~ deserializer.

Modify the Append_inbuffer as follows:

```
Append_inbuffer()
{
    BlockFromPMA()
    if(sh_cnt<27)
    {
        inbuffer[input_buffer_location]=rx_coded<1>
        input_buffer_location++
    }

    for(i=2, i<66, i++)
    {
        inbuffer[input_buffer_location]=rx_coded<i>
        input_buffer_location++
    }
}
```

Augment the description of BlockToDescrambler to read:

BlockToDescrambler

Function that sends the next rx_coded_corrected<65:0> block to the descrambler. It does not return until the transfer is completed, and each transfer shall take 6.4 ns and be synchronized to the XGMII clock.