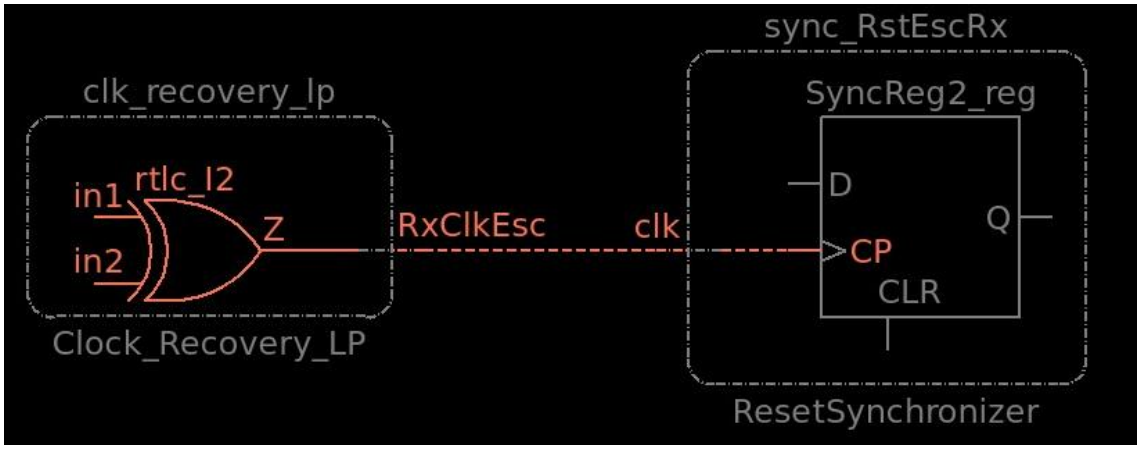
Spyglass CDC

1- Clock_reset_integrity Goal

- **Clock_check01**
  Potential glitch in clock tree due to unexpected gates in clock tree

This schematic was generated from:
Rule: **Clock_check01:** Flags unexpected gates in a clock tree
Message: Unexpected XOR gate (at slave.RxClkEscOut) in clock tree of flop (output net slave.sync_RstEscRx.SyncReg2)



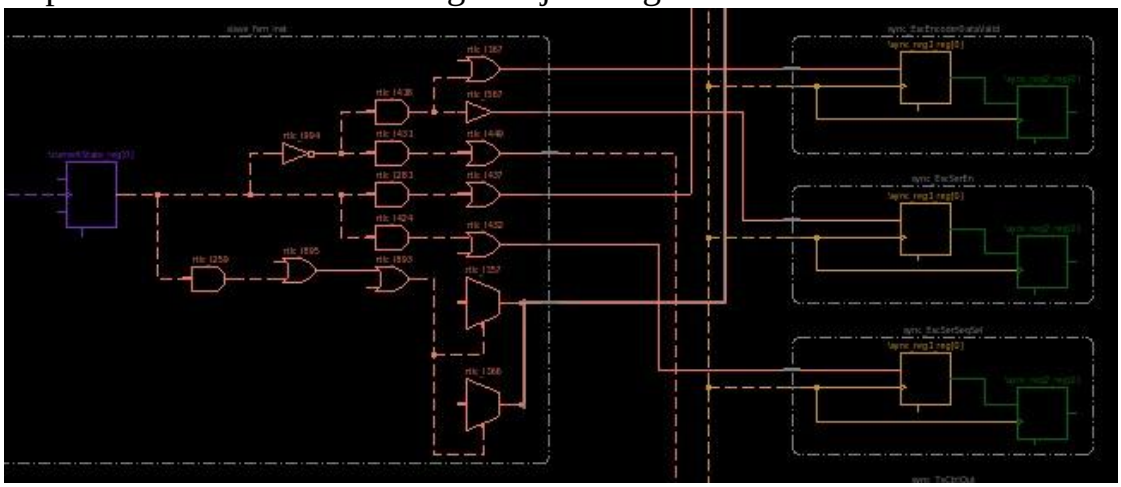
  - Solution:
    We need this gate to recover clock from the data. Waive this warning.

- **Clock_check04**
  Both positive and negative edges of clocks used in the same deign.
  - Solution:
    We need to use both edges in the low power blocks to achieve target functionality.
    Waive this warning.

2- Cdc_verify_struct Goal

- **Ar_unsync01**
  Reports unsynchronized reset signals in the design.
  - ➢ Solution:
    Adding reset synchronization to the different domains in the design.

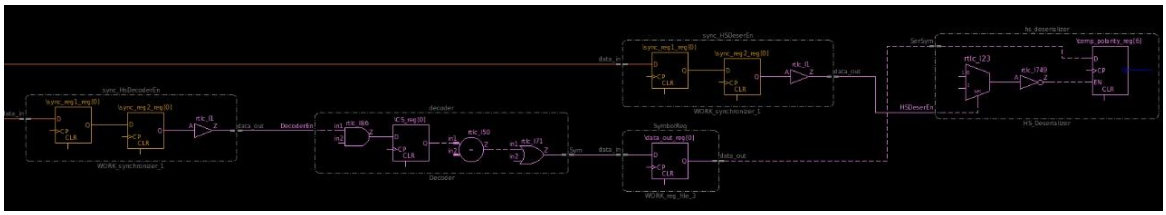
- **Ac_glitch03**
  Reports clock domain crossings subject to glitches.



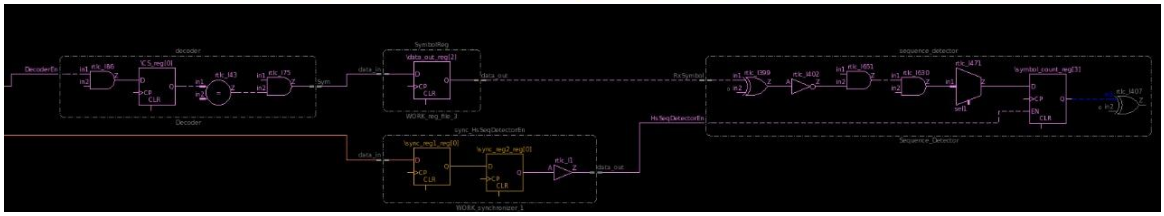
  Outputs from combinational circuit input to synchronizers which cause glitches.
  - ➢ Solution:
    Adding FFs to each output of the block before the synchronizers.

- **Ac_conv01**
  Checks for sequential convergence of properly synchronized control crossings.

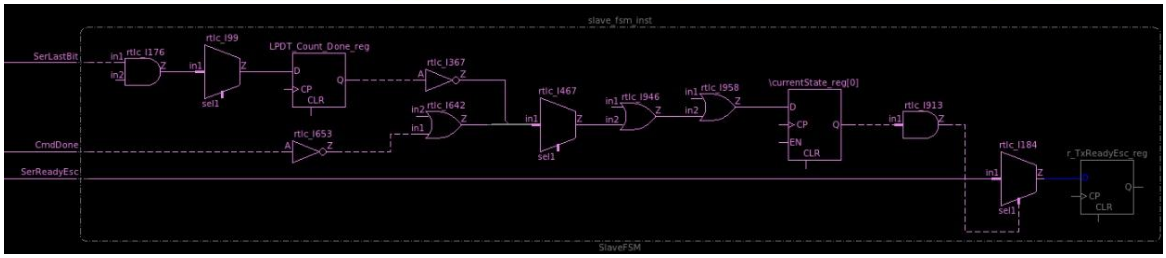
This schematic was generated from:
Rule: Ac_conv01: Checks sequential convergence of same-domain signals synchronized in the same destination domain
Message: 2 synchronizers (slave.sync_HsDecoderEn.sync_reg1[0],slave.sync_HSDeserEn.sync_reg1[0]) converge on flop 'slave.hs_deserializer.temp_polarity[6]'



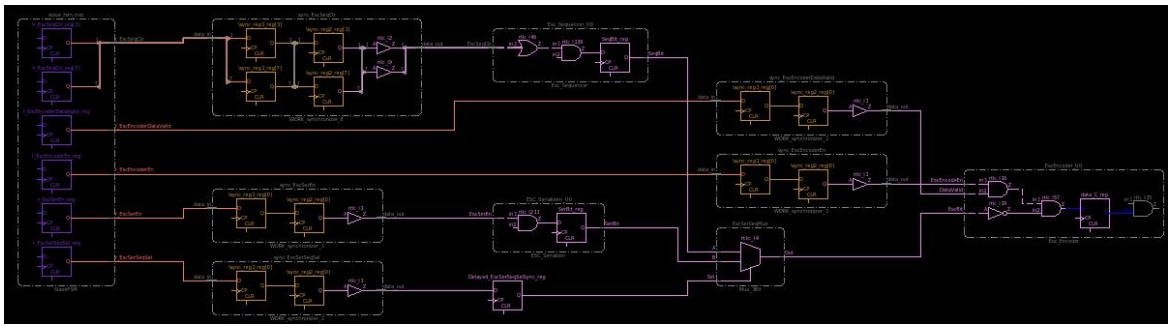
This schematic was generated from:
Rule: **Ac_conv01:** Checks sequential convergence of same-domain signals synchronized in the same destination domain
Message: 2 synchronizers (slave.sync_HsDecoderEn.sync_reg1[0],slave.sync_HsSeqDetectorEn.sync_reg1[0]) converge on flop 'slave.sequence_detector.symbol_count[3]'



This schematic was generated from:
Rule: **Ac_conv01:** Checks sequential convergence of same-domain signals synchronized in the same destination domain
Message: 3 synchronizers (slave.sync_CmdDone.sync_reg1[0],slave.sync_SerReadyEsc.sync_reg1[0] ...) converge on MUX 'slave.slave_fsm_inst.TxReadyEsc'



This schematic was generated from:
Rule: **Ac_conv01:** Checks sequential convergence of same-domain signals synchronized in the same destination domain
Message: 10 synchronizers (slave.sync_EscSerEn.sync_reg1[0],slave.sync_EscEncoderEn.sync_reg1[0] ...) converge on flop 'slave.EscEncoder_U0.data_C'



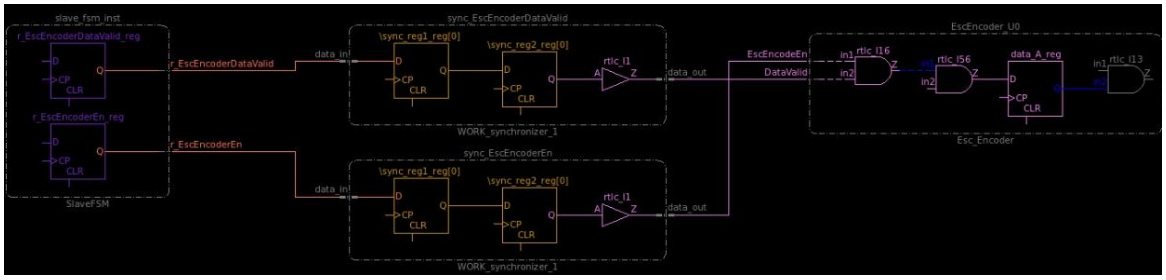
- Solution:
  Since these signals are not functionally related, convergence checks are not required

```
no_convergence_check -name {slave.HsDecoderEn_sync slave.HsSeqDetectorEn_sync} #not related
no_convergence_check -name {slave.HsDecoderEn_sync slave.HSDeserEn_sync} #not related
no_convergence_check -name {slave.SerReadyEsc_sync slave.CmdDone_sync slave.SerLastBit_sync} #not related
no_convergence_check -name {slave.EscSeqCtr_sync slave.EscEncoderDataValid_sync slave.EscEncoderEn_sync slave.EscSerEn_sync slave.EscSerSeqSelSync} #not related
```

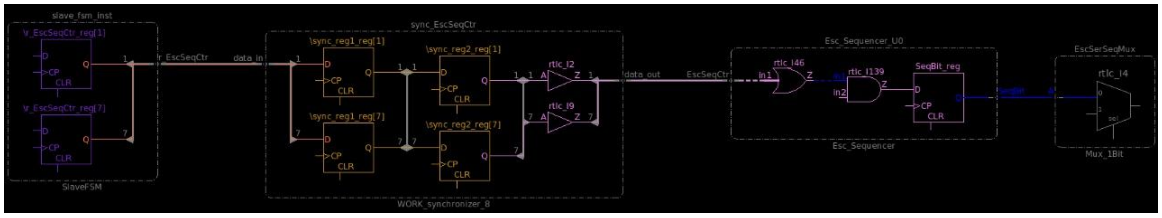
- **Ac_conv02**
  Checks for combinational convergence of properly synchronized control crossings.

  Output signals from the same block synchronized to a block in another domain.

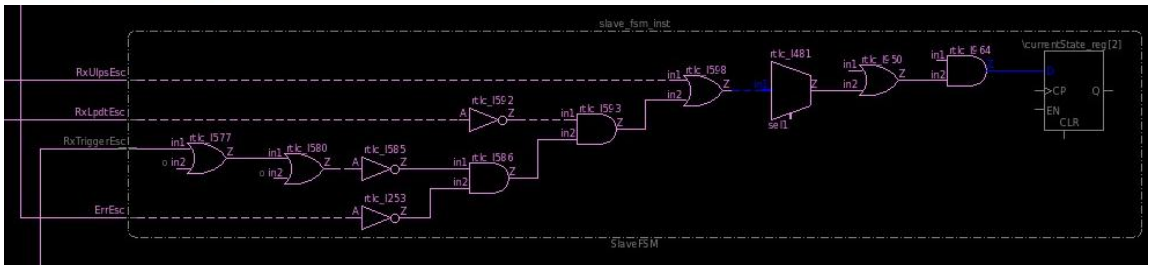
This schematic was generated from:
Rule: **Ac_conv02:** Checks combinational convergence of same-domain signals synchronized in the same destination domain
Message: 2 synchronizers (slave.sync_EscEncoderEn.sync_reg1[0],slave.sync_EscEncoderDataValid.sync_reg1[0]) converge on flop 'slave.EscEncoder_U0.data_A'. Gray encoding check: 'DISABLED'



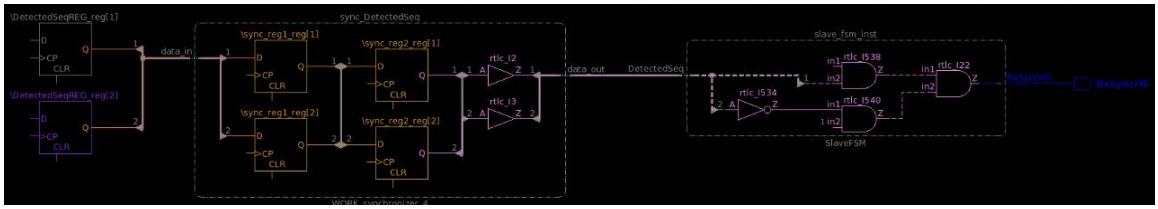
This schematic was generated from:
Rule: **Ac_conv02:** Checks combinational convergence of same-domain signals synchronized in the same destination domain
Message: 6 synchronizers (slave.sync_EscSeqCtr.sync_reg1[1:0],slave.sync_EscSeqCtr.sync_reg1[7:4]) converge on flop 'slave.SeqBit'. Gray encoding check: 'DISABLED'



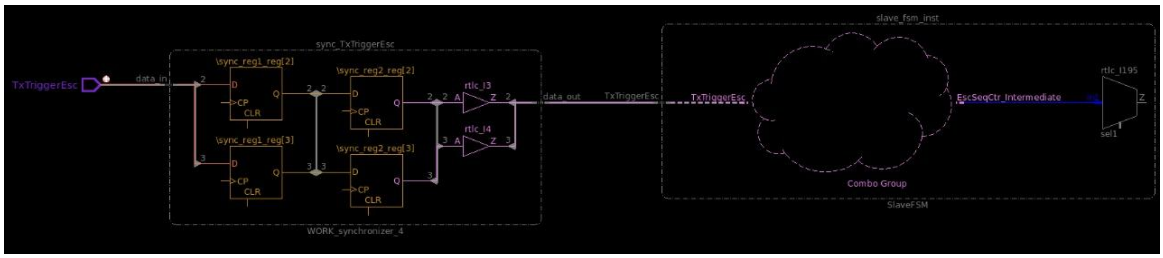
This schematic was generated from:
Rule: **Ac_conv02:** Checks combinational convergence of same-domain signals synchronized in the same destination domain
Message: 4 synchronizers (slave.sync_InRxTriggerEsc.sync_reg1[0],slave.sync_InRxUlpsEsc.sync_reg1[0] ...) converge on combinational gate 'slave.slave_fsm_inst.nextState[2]'. Gray encoding check: 'DISABLED'



This schematic was generated from:
Rule: Ac_conv02: Checks combinational convergence of same-domain signals synchronized in the same destination domain
Message: 3 synchronizers (slave.sync_DetectedSeq.sync_reg1[2:0]) converge on combinational gate 'slave.RxSyncHS'. Gray encoding check: 'DISABLED'



This schematic was generated from:
Rule: Ac_conv02: Checks combinational convergence of same-domain signals synchronized in the same destination domain
Message: 4 synchronizers (slave.sync_TxTriggerEsc.sync_reg1[3:0]) converge on combinational gate 'slave.slave_fsm_inst.EscSeqCtr_Intermediate[0]'. Gray encoding check: 'DISABLED'

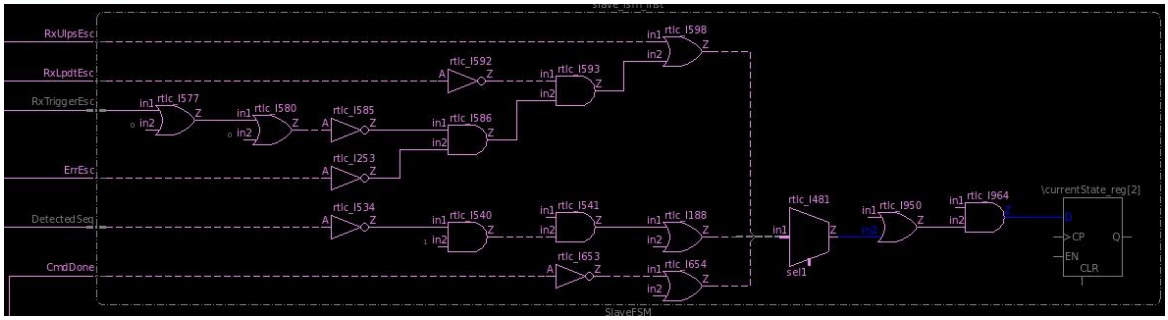


- Solution:
  For each warning, we make sure in the design that these signals are grey encoded.

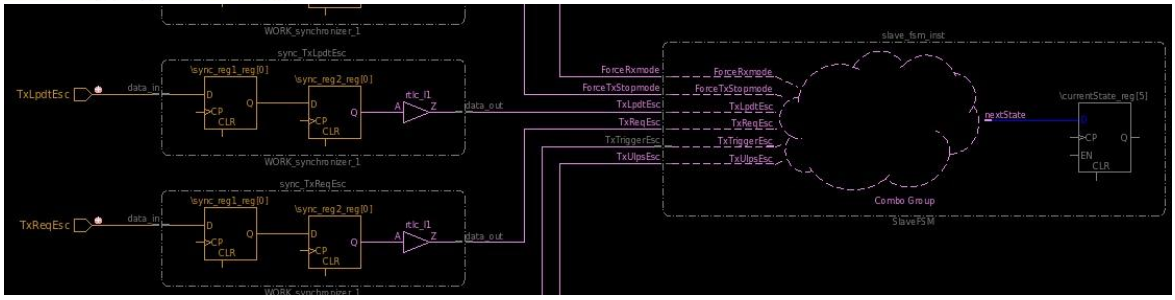
```
cdc_attribute -exclusive {slave.TxTriggerEsc[3:0]} #gray encoded
cdc_attribute -exclusive {slave.ErrEsc slave.RxLpdtEsc slave.RxUlpsEsc slave.RxTriggerEsc[0] } #gray encoded
cdc_attribute -exclusive {slave.EscEncoderDataValid slave.EscEncoderEn } #gray encoded
cdc_attribute -exclusive {slave.EscSeqCtr[7:0]} #gray encoded
cdc_attribute -exclusive {slave.sync_DetectedSeq.sync_reg1[0:2]} #gray encoded
```

- **Ac_conv03**
  Checks different-domain signals synchronized in the same destination domain and are converging.

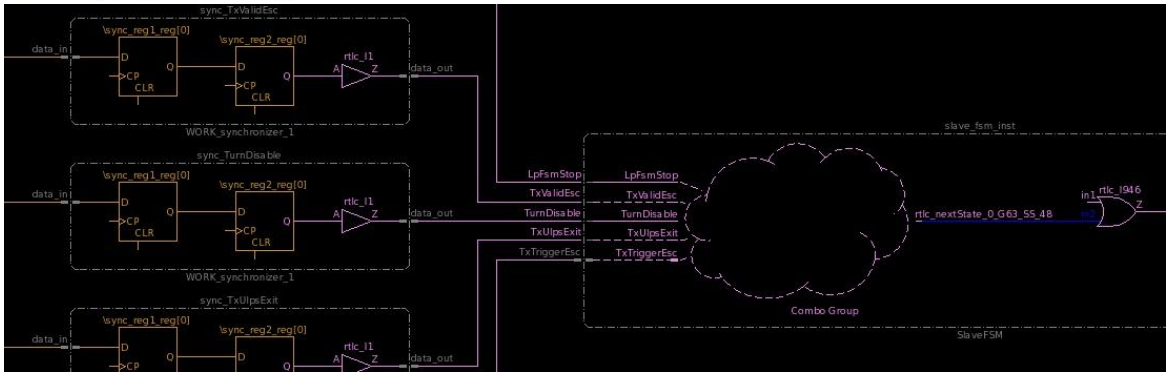
This schematic was generated from:
Rule: Ac_conv03: Checks different-domain signals synchronized in the same destination domain and are converging
Message: 8 synchronizers (slave.sync_DetectedSeq.sync_reg1[2:0],slave.sync_InRxTriggerEsc.sync_reg1[0] ...) converge on combinational gate 'slave.slave_fsm_inst.nextState[2]'



This schematic was generated from:
Rule: Ac_conv03: Checks different-domain signals synchronized in the same destination domain and are converging
Message: 9 synchronizers (slave.sync_TxReqEsc.sync_reg1[0],slave.sync_TxUlpsEsc.sync_reg1[0] ...) converge on combinational gate slave.slave_fsm_inst.nextState[5]'



This schematic was generated from:
Rule: Ac_conv03: Checks different-domain signals synchronized in the same destination domain and are converging
Message: 7 synchronizers (slave.sync_LpFsmStop.sync_reg1[0],slave.sync_TxValidEsc.sync_reg1[0] ...) converge on combinational gate 'slave.slave_fsm_inst.nextState[0]'



- Solution:
  For each warning, the signals involved are unrelated and do not change simultaneously; therefore, a convergence check is not required

```
no_convergence_check -name {slave.InErrEsc_sync slave.InRxLpdtEsc_sync slave.InRxUlpsEsc_sync slave.InRxTriggerEsc_sync[0] slave.CmdDone_sync slave.DetectedSeq_sync[2:0]} #not related
no_convergence_check -name {slave.ForceRxmodeSync slave.ForceTxStopmodeSync slave.TxLpdtEsc_sync slave.TxReqEsc_sync slave.TxTriggerEsc_sync[3] slave.TxUlpsEsc_sync} #not related
no_convergence_check -name {slave.LpFsmStop_sync slave.slave_fsm_inst.TxValidEsc slave.TurnDisableSync slave.TxUlpsExit_sync slave.TxTriggerEsc_sync[2]} #not related
```

- **Clock_sync05**
  Reports primary inputs that are multi-sampled.
  In the design, inputs A, B, and C are sampled by both the low-power clock and the high-speed clock
  - Solution:
    Sampling of inputs A, B, and C by both the low-power and high-speed clock domains is necessary to enable functionality across different operating modes.
    So, waive the warning.

- **Clock_sync06**
  Reports primary outputs driven by multiple clock domain flip-flops or latches.
  In the design, the outputs related to the contention detection block are driven by both the Rx and Tx domains.
  - Solution:
    The contention detection block monitors inputs from different clock domains to detect any contention errors between them.
    So, waive the warning.

3- Cdc_verify Goal

- **Ac_ccd01a**

  Checks data-loss for multi-flop or sync cell or qualifier synchronized clock domain crossings

  - Solution:

    For all these signals, we ensure in the design that each signal is held high or low long enough to be reliably captured by the slow clock domain without data loss.

    So, waive this rule.