UNIVERSIDAD JESUITA DE GUADALAJARA

ITESO

Serializer design for a SerDes chip in 130 nm CMOS Technology

EFRAIN ARRAMBIDE BARRON

Directors:

Dr. Víctor Avendaño Fernández, MSC. Cuauhtemoc Aguilera

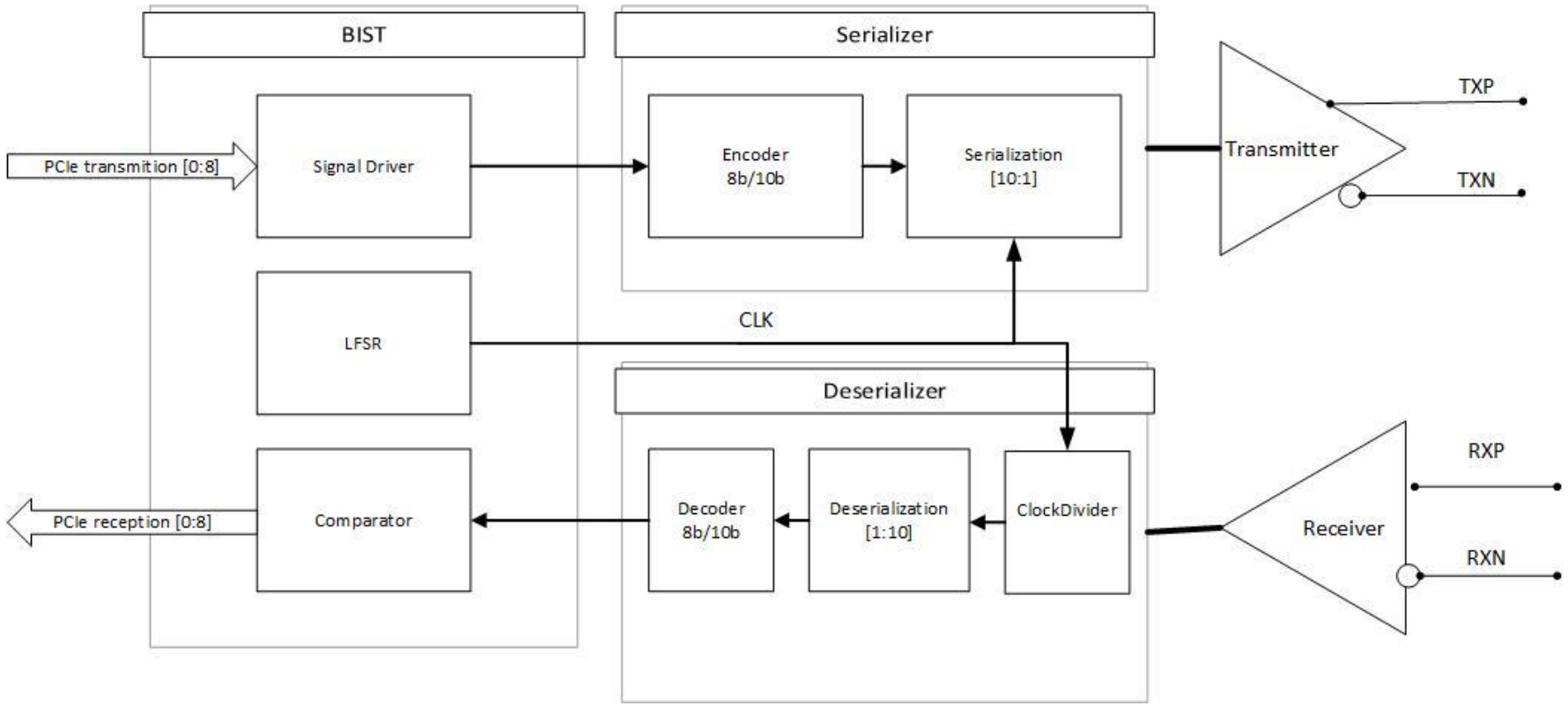
Agenda

- SerDes Specifications
- SerDes block diagram
- Serializer top diagram and pin description
- Logic synthesis
- Pyhsical synthesis
- Virtuoso

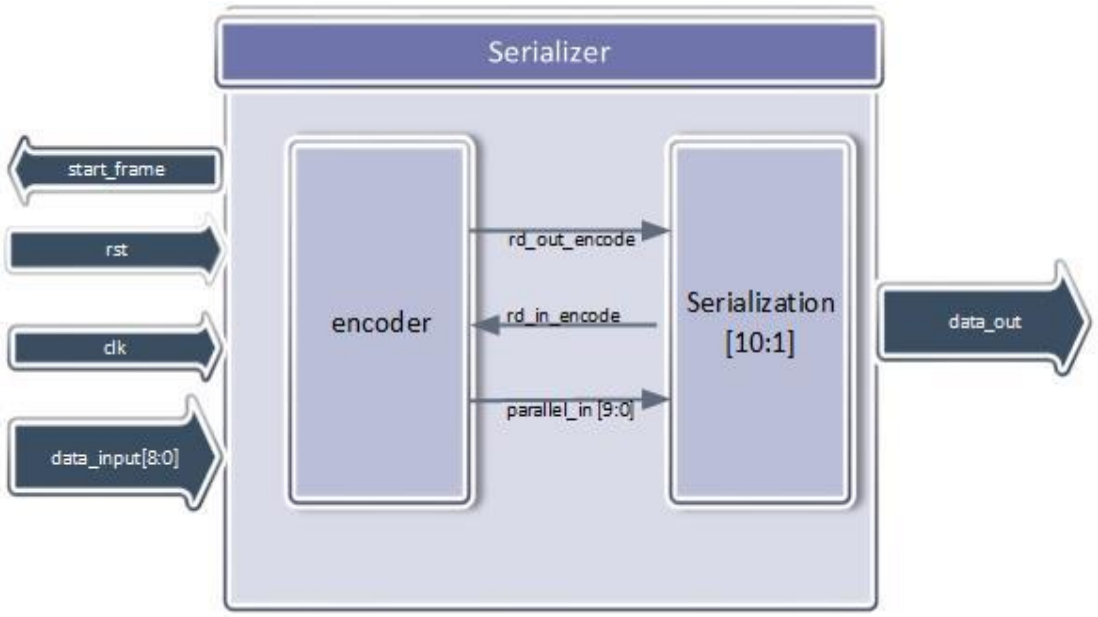
SerDes specifications

- ARM Manufacturing technology for CMOS 130nm
- DIP40 packaging provides by MOSIS.
- Communication protocol PCIe 1.0 (8b/10b decoding and frequency up to 1.25 GHz)
- Operation frequency 125 Mhz (SerDes development)
- Area Restrictions 1.5mm x1.5mm provided by MOSIS
- VDD voltage 1.2v
- Cadence RC compiler (Logical Synthesis, Power, Area, Timing and compilation errors)
- Cadence Encounter(Physical Synthesis, DRC, Geometry, Connectivity and Timing)
- Cadence Virtuoso(Schematic and Layout analysis and simulation)
- Model Sim-Altera 10.4d (RTL simulation and compilation tool)

SerDes block diagram

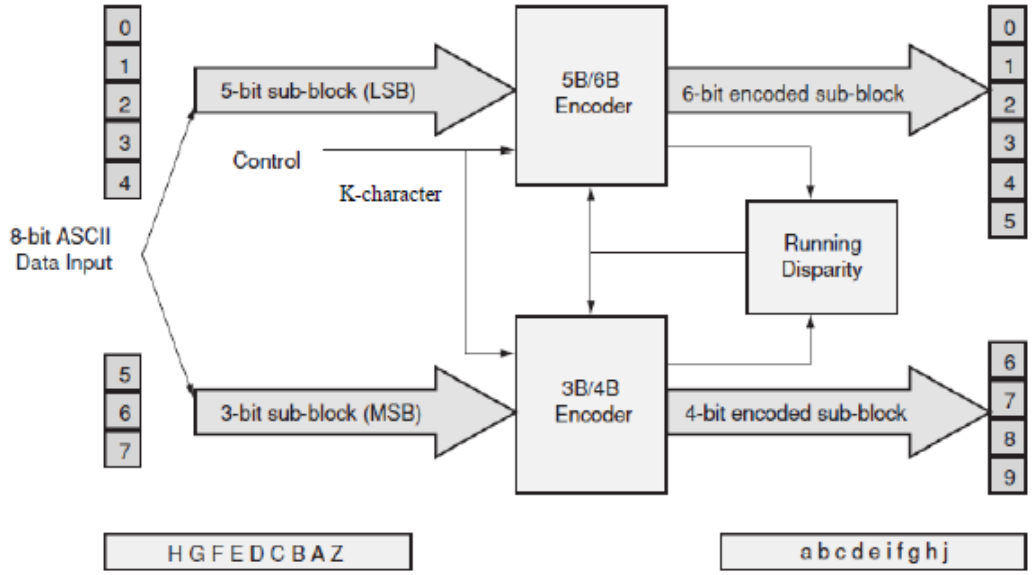


Serializer top diagram and pin description



Name	I/O Direction	Description
start_frame	output	Active high signal to synchronization with the PCIe protocol, when this in enable the Serializer is ready to receive the data from the pin data_input.
Rst	input	Active high asynchronous reset, this pin is common for all the components of the chip.
Clk	input	Transmission clock coming from the clock generator module 125 Mhz.
data_input [8:0]	Input	Data stream coming from the PCIe protocol waiting to be encoded.
data_out	output	Data bit transmitted to the analog transmitter.
Internal Signals	I/O Direction	Description
rd_out_encode	Encoder to Serialization	Run disparity out a bit, it saves during the stream transmitting to the next encoding stream.
rd_in_encode	Serialization to Encoder	Run disparity in a bit, it is returned by serialization module, the main function of it creates a balance stream data.
parallel_in[7:0]	Encoder to Serialization	This is encoded data ready to be transmitted by Serialization model.

Encoder architecture



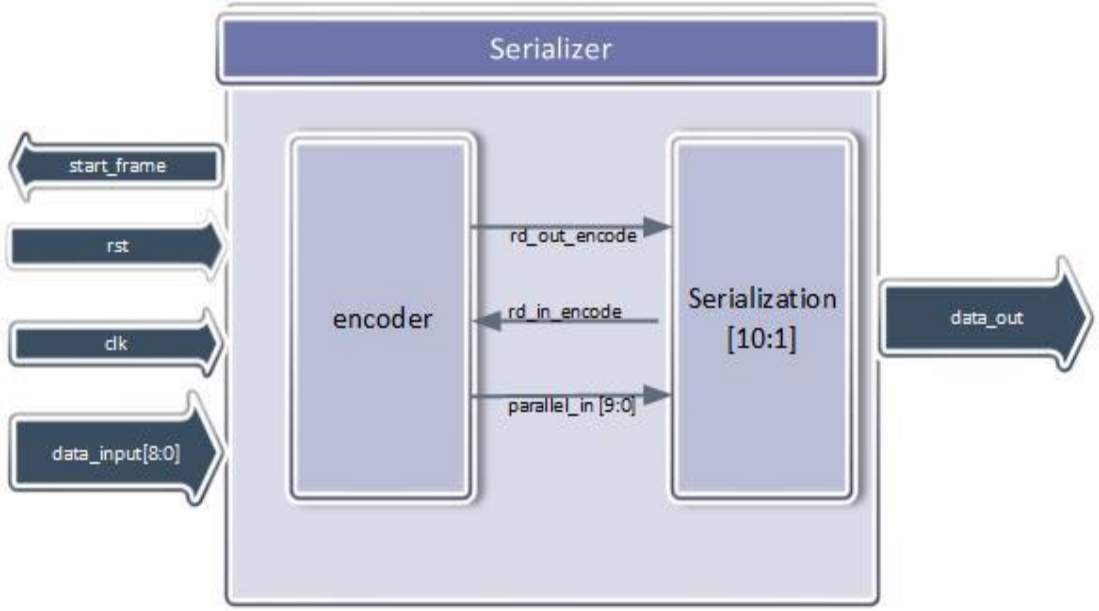
Serializer enhancement


```
module serializer(
input [9:0]par_in,
input clk,
input rst,
input disparity_d,
output ser_out,
output reg disparity_q,
output reg tx_frame_started);

reg [9:0] par_reg;
reg [3:0] counter:

always @(posedge clk, posedge rst) begin
                if (rst) begin
                                counter <= 4'b0;
                                tx_frame_started <= 1'b0;
                end
                else if (counter == 4'b0) begin
                                tx_frame_started <= 1'b1;
                                counter <= counter + 1'b1;
                end
                else if (counter < 4'b10001) begin
                                tx_frame_started <= 1'b0;
                                counter <= counter + 1'b1;
                end
                else begin
                                counter <= 4'b0;
                                tx_frame_started <= 1'b0;
                end
end

always @(posedge clk, posedge rst) begin
                if (rst) begin
                                par_reg <= 10'b0;
                end
                else if (tx_frame_started) begin
                                par_reg <= par_in;
                end
                else begin
                                par_reg <= {1'b0, par_reg[9:1]};
                end
end

assign ser_out = par_reg[0];

always @(posedge clk, posedge rst) begin
                if (rst) begin
                                disparity_q <= 1'b0;
                end
                else if (tx_frame_started)
                                disparity_q <= disparity_d;
                end
endmodule
```

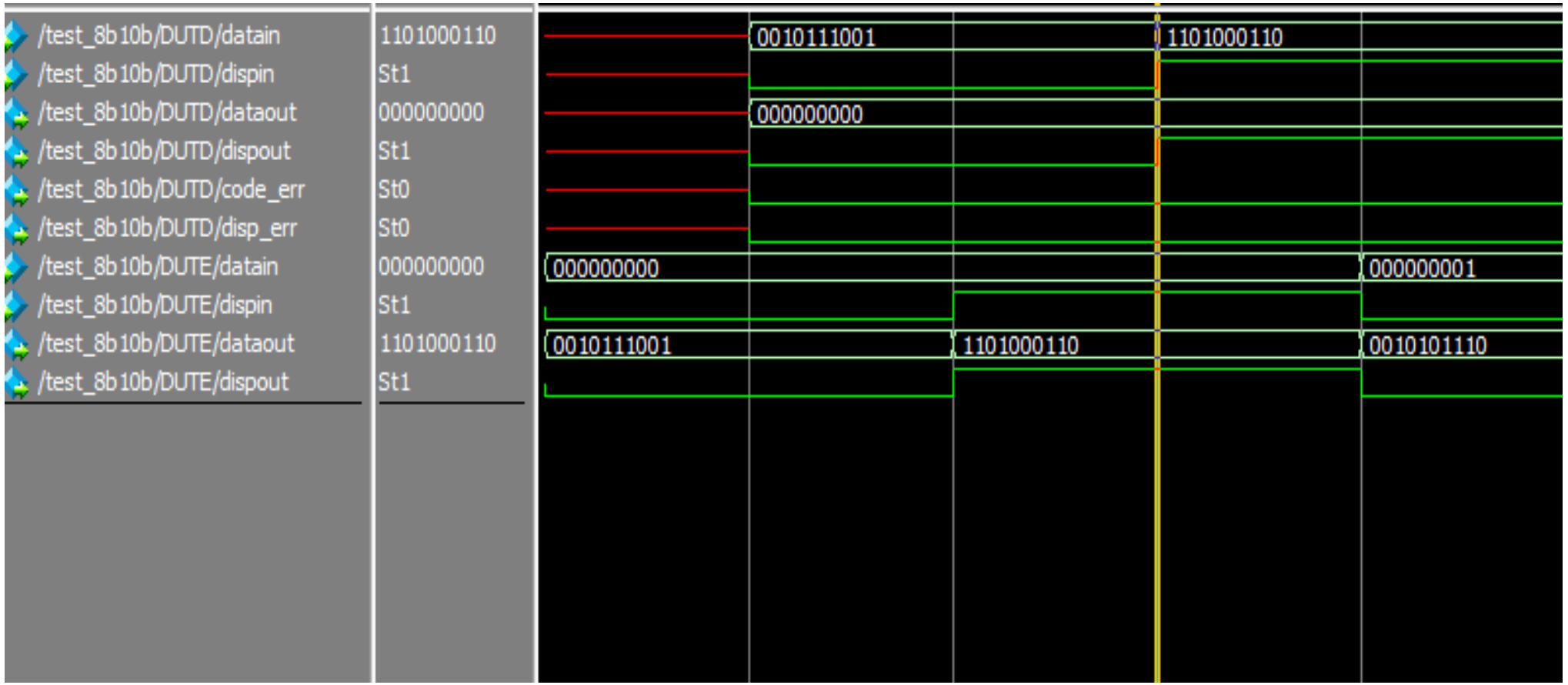
```
module Serialization(
input [9:0]par_in,
input     clk,
input     rst,
input                rd_in,
output reg  rd_out,
output ser_out,
output start_frame
);

reg [9:0] par_reg;
reg [3:0] counter;

always @(posedge clk, posedge rst) begin
                if (rst) begin
                                counter <= 4'h0;
                                rd_out <= 1'h0;
                                par_reg <= 10'h0;
                end
                else if (counter == 4'h0) begin
                                counter <= counter + 1'h1;
                                rd_out <= rd_in;
                                par_reg <= par_in;
                end
                else if (counter < 4'h9) begin
                                counter <= counter + 1'h1;
                                rd_out <= rd_out;
                                par_reg <= {1'h0, par_reg[9:1]};
                end
                else begin
                                counter <= 4'h0;
                                rd_out <= rd_out;
                                par_reg <= {1'h0, par_reg[9:1]};
                end
end

assign ser_out = par_reg[0];
assign start_frame = (counter == 4'h9)? 1'h1 : 1'h0;

Endmodule
```

Serializer enhancement

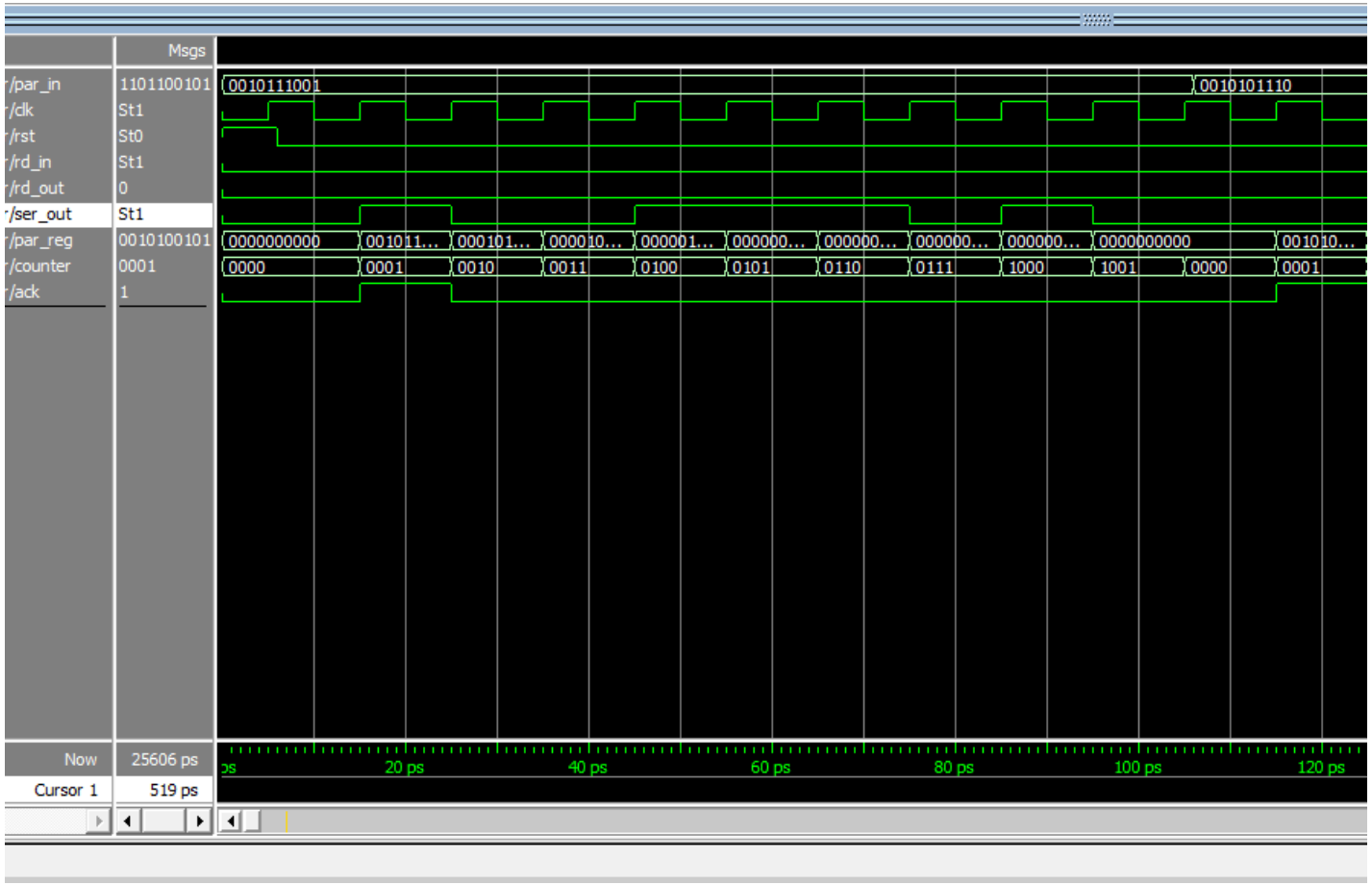
- Remove a register start_frame.
- Capute of parallel data and disparity bit one cycle after the start_frame is enable.

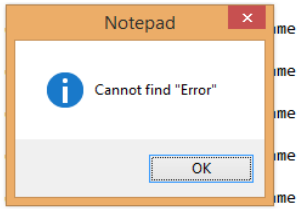


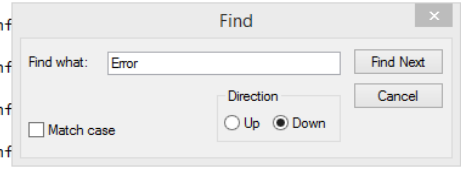
Encoder verification



Serializer verification

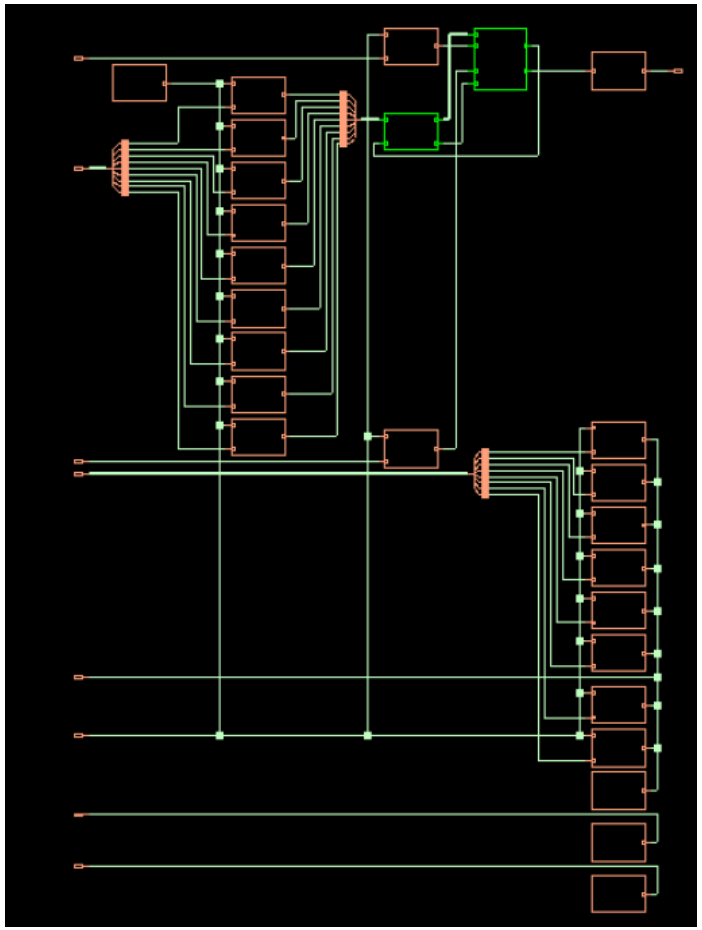


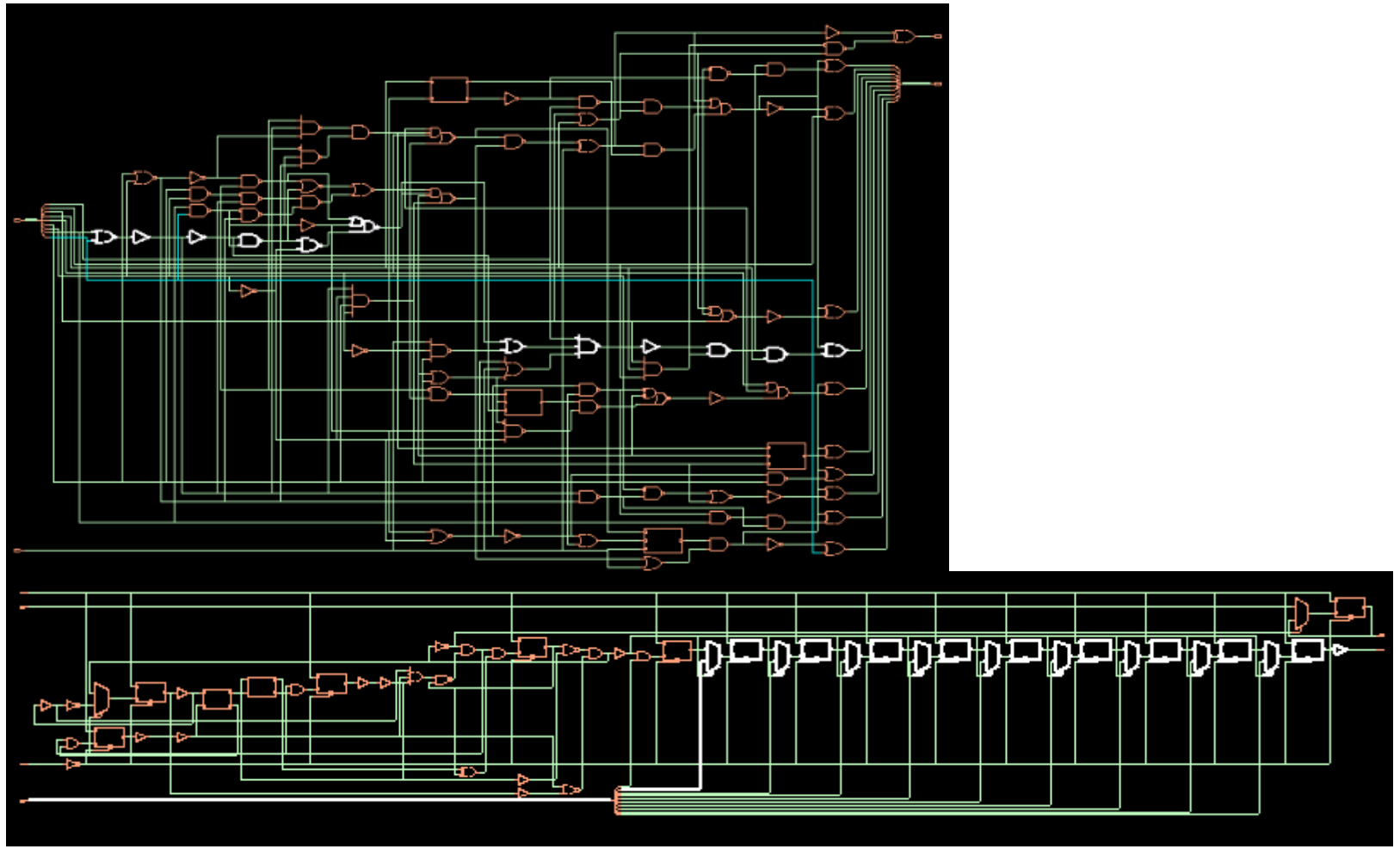





```
-Info- Generate data = 1111111110, Serial data bit = 0, Run                                    me 0
-Info- Generate data = 1111111110, Serial data bit = 1, Run                                    me 0
-Info- Generate data = 1111111110, Serial data bit = 1, Run                                    me 0
-Inf                                                    = 1, Run                               me 0
-Inf                                                    = 1, Run                               me 0
-Inf                                                    = 1, Run disparity value 1, start_frame 0
-Inf                                                    = 1, Run disparity value 1, start_frame 0
-Info- Generate data = 1111111110, Serial data bit = 1, Run disparity value 1, start_frame 0
-Info- Generate data = 1111111110, Serial data bit = 1, Run disparity value 1, start_frame 1
-Info- Generate data = 1111111110, Serial data bit = 1, Run disparity value 1, start_frame 0
-Info- Match Generate data = 1022,  Expected data = 1022, Serial data bit = 1, Run disparity 1
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 1
-Info- Generate data = 1111111111, Serial data bit = 1, Run disparity value 0, start_frame 0
-Info- Match Generate data = 1023,  Expected data = 1023, Serial data bit = 1, Run disparity 0
```

Logic synthesis, slack 94 ps 80 % in the Encoder





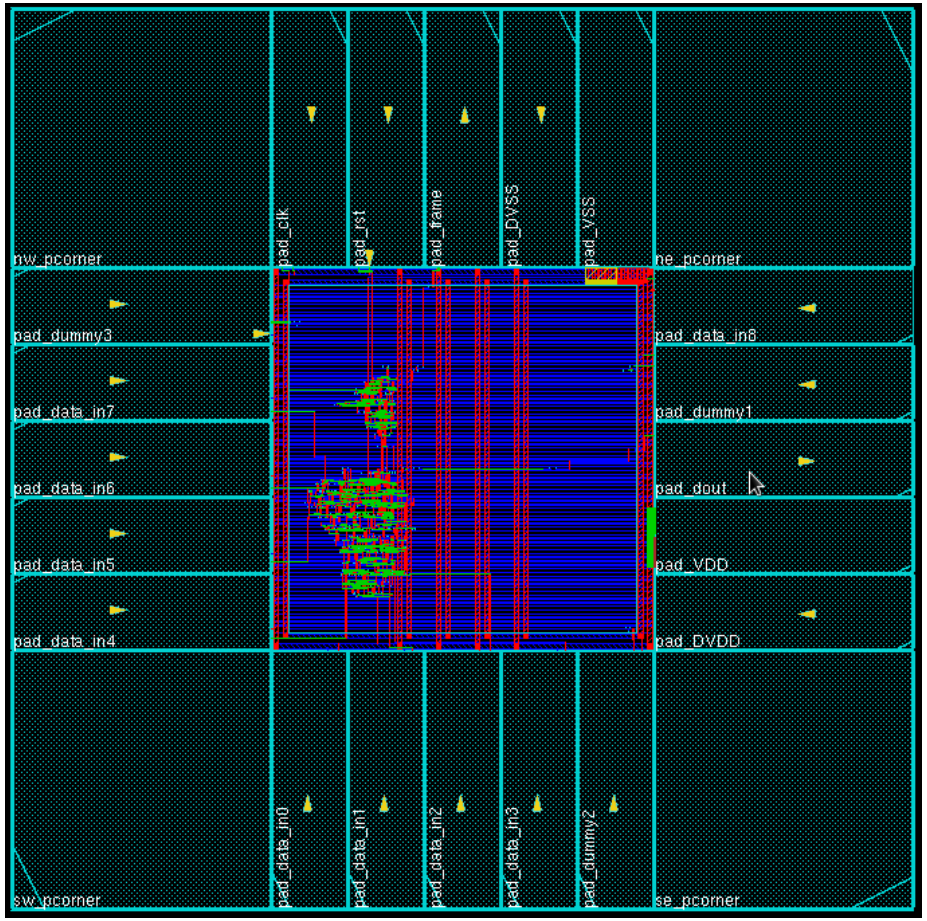
Physical synthesis logs

```
 *** Starting Verify Geometry (MEM: 837.4) ***

  VERIFY GEOMETRY ..... Starting Verification
  VERIFY GEOMETRY ..... Initializing
  VERIFY GEOMETRY ..... Deleting Existing Violations
  VERIFY GEOMETRY ..... Creating Sub-Areas
                  ..... bin size: 2560
VG: elapsed time: 0.00
Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 0
  Overlap     : 0
End Summary

  Verification Complete : 0 Viols.  0 Wrngs.

**********End: VERIFY GEOMETRY**********
```

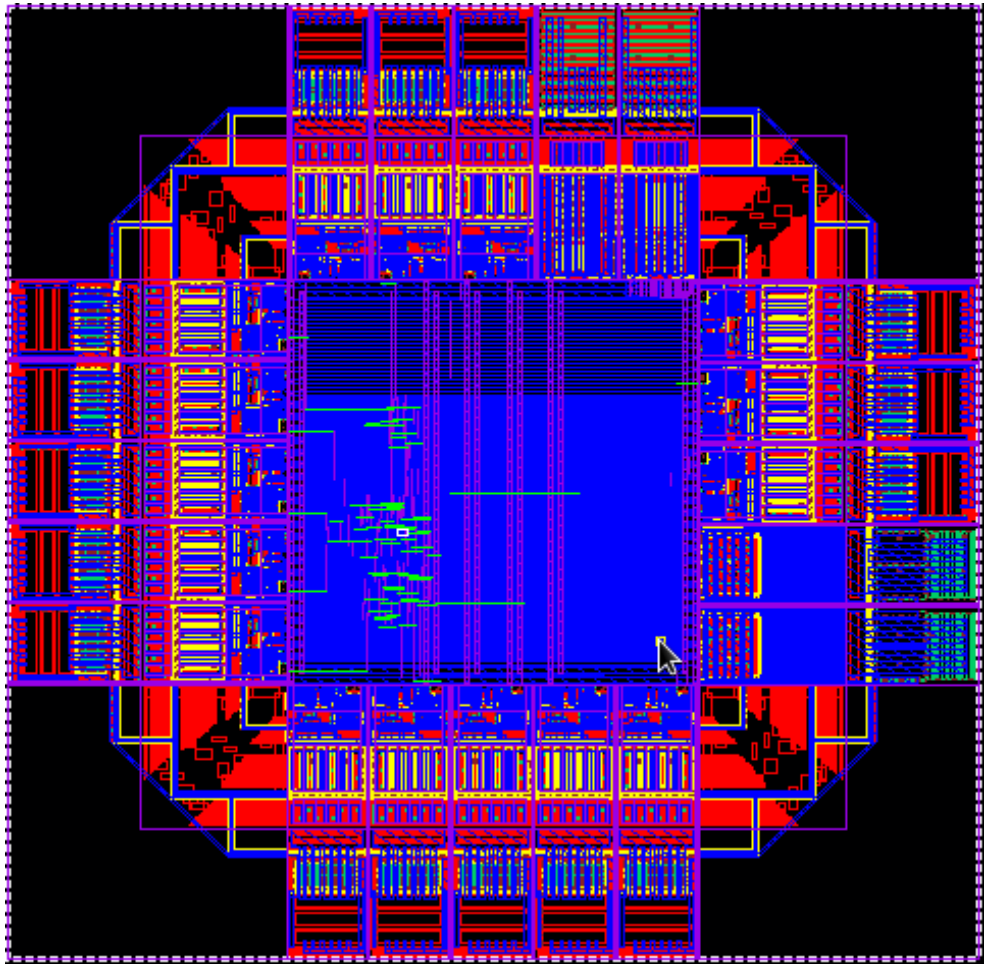
```
******** Start: VERIFY CONNECTIVITY ********
Start Time: Sun Jun 19 15:38:29 2016

Design Name: Chip_Serial_Transmisor
Database Units: 1000
Design Boundary: (0.0000, 0.0000) (1159.0000, 1159.0000)
Error Limit = 1000; Warning Limit = 50
Check all nets

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Sun Jun 19 15:38:29 2016
Time Elapsed: 0:00:00.0

******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
```

```
------------------------------------------------------------
    optDesign Final Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg |reg2cgate| default |
+--------------------+---------+---------+---------+---------+
|           WNS (ns):|  0.155  |  0.236  |   N/A   |  0.155  |
|           TNS (ns):|  0.000  |  0.000  |   N/A   |  0.000  |
|    Violating Paths:|    0    |    0    |   N/A   |    0    |
|          All Paths:|   32    |   15    |   N/A   |   28    |
+--------------------+---------+---------+---------+---------+

+--------------------+---------+---------+---------+---------+
|     Hold mode      |   all   | reg2reg |reg2cgate| default |
+--------------------+---------+---------+---------+---------+
|           WNS (ns):|  0.486  |  1.116  |   N/A   |  0.486  |
|           TNS (ns):|  0.000  |  0.000  |   N/A   |  0.000  |
|    Violating Paths:|    0    |    0    |   N/A   |    0    |
|          All Paths:|   32    |   15    |   N/A   |   28    |
+--------------------+---------+---------+---------+---------+
```

Physical synthesis final chip



Serializer in Virtuoso



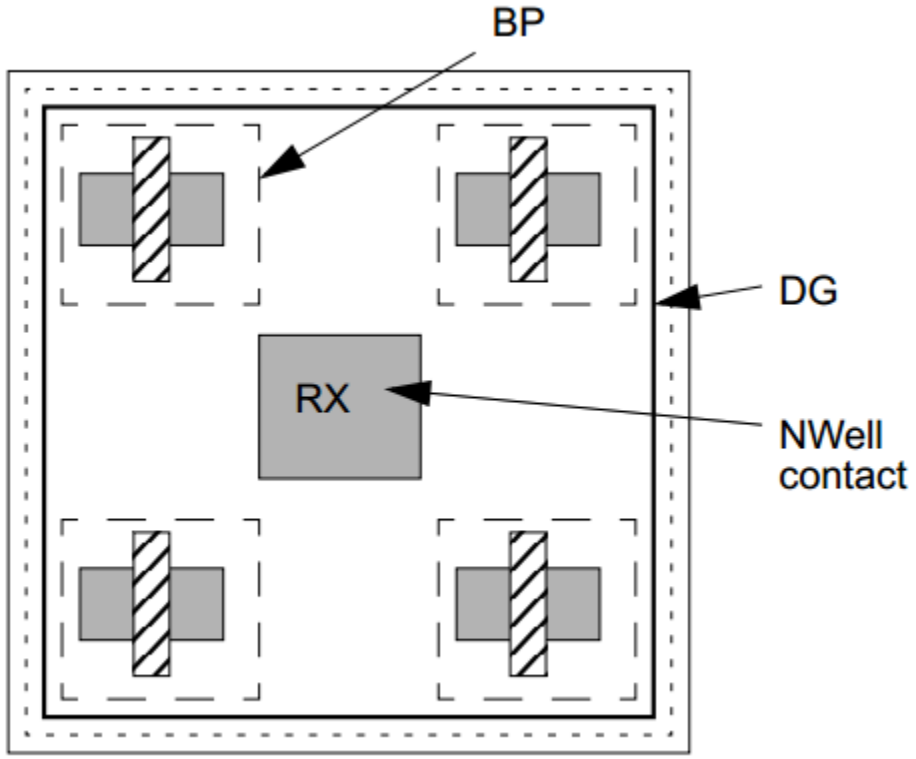
DRC in Virtuso

- GRLOGIC violations.
  - fixing using the GRLOGIC, DRC was complying because the standard cells don't have a connection to the substrate. A wrap of GRLOGIC was created around the chip to tell DRC test these zone of layout should be traded as a standard cell and avoid this violation rule
- N-well spacing issues
  - NW connexion it was a violation the distance so we procedure to join all the spaces between the NW material

DRC Virtuoso

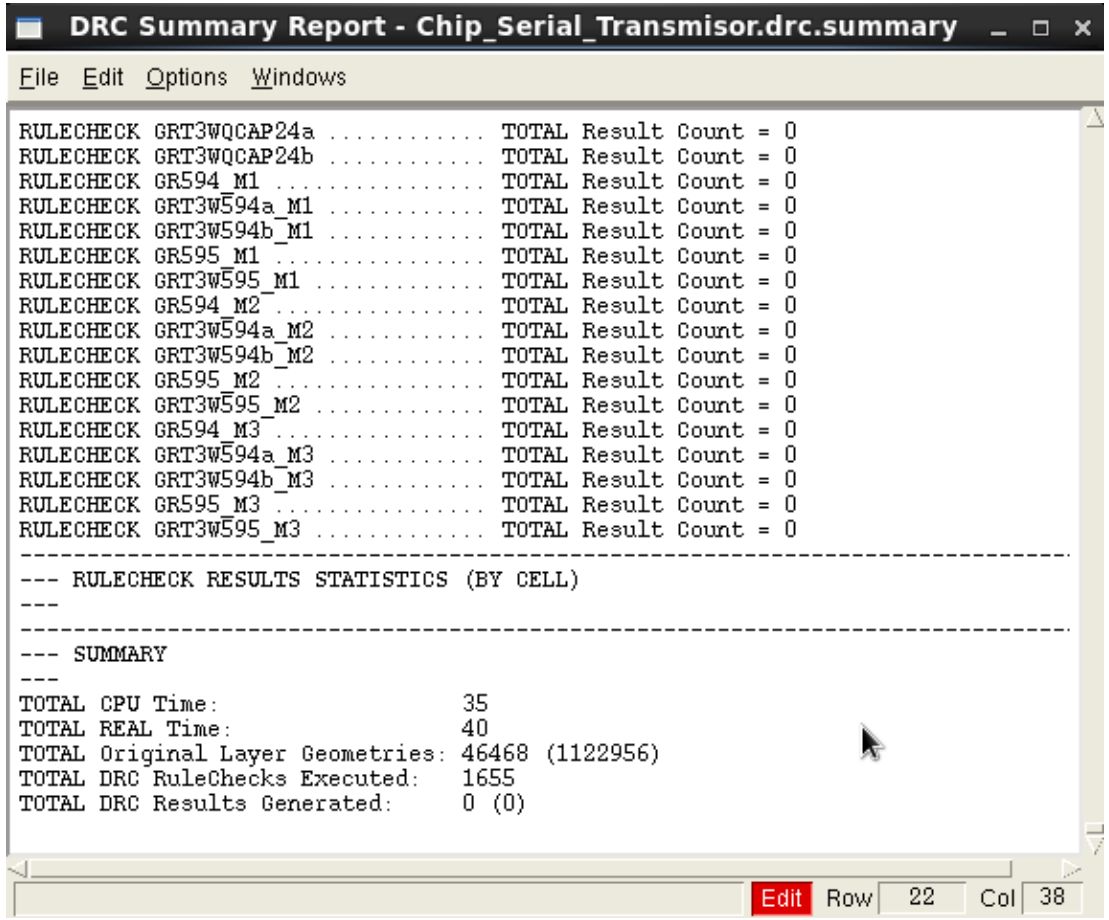
- GRLUP10 violation

This explanation suggests that the NWell contact needs to have a minimum of area related to the BP contact area, this is to avoid short circuits between the substratum and ground



DRC Virtuoso

- Final result



Conclusion

- Serializer start_frame was modifed, it helps to reduce the amount of register used in this module.
- The capture of parallel data was modify giving a better performance in the operation speed.
- Serializer module passed all the verifications test in logic and physical implementations and layout verifications (LVS is pending).

Future work

- LVS mismatching correction
- SerDes integration
- Increase the speed in digital modules (costume cells)

Thanks for your attention

- Q&A