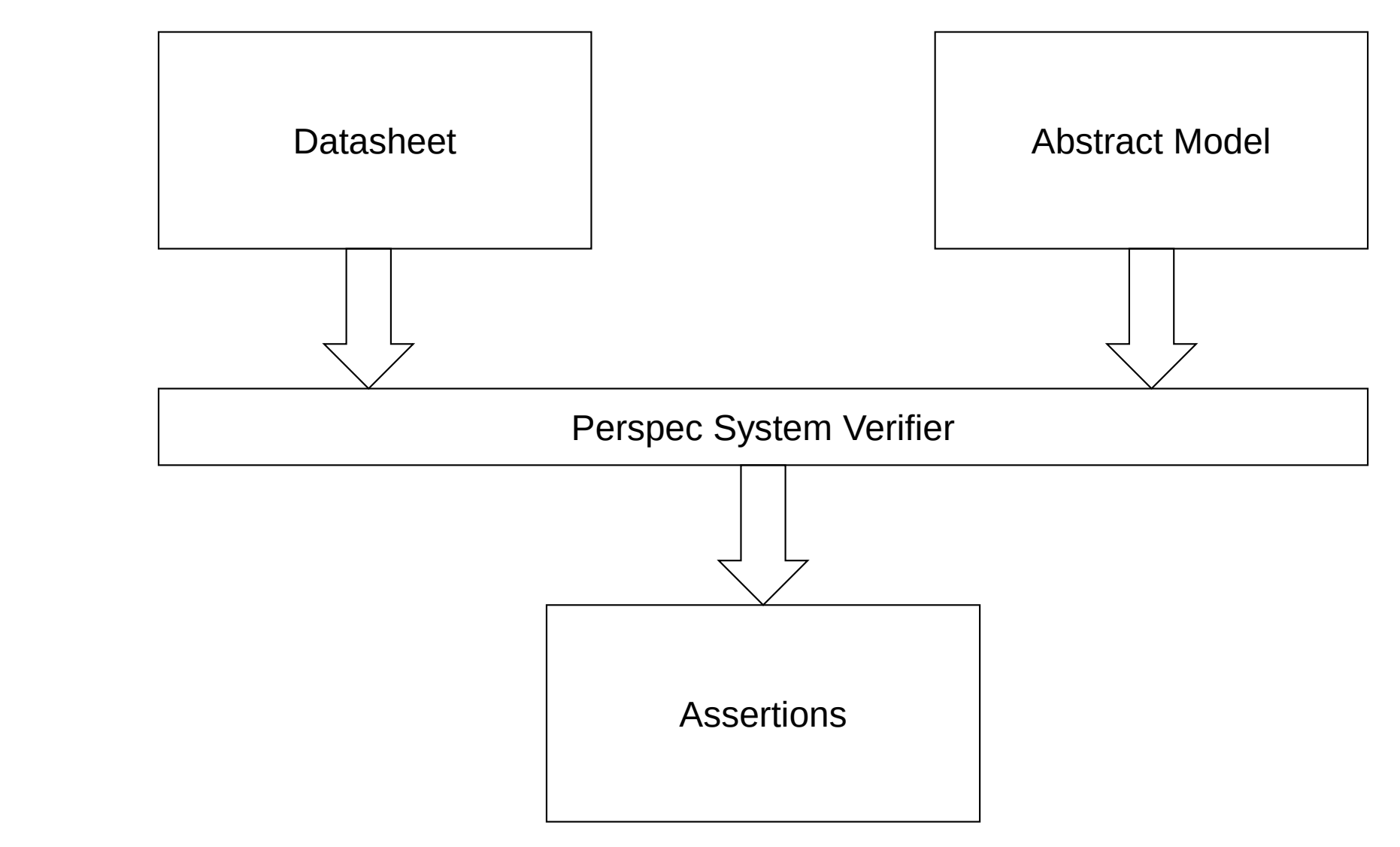
Holistic Approach to IO Timing Verification Using Portable Stimulus and Assertions

Amitesh Khandelwal and Praveen Kumar

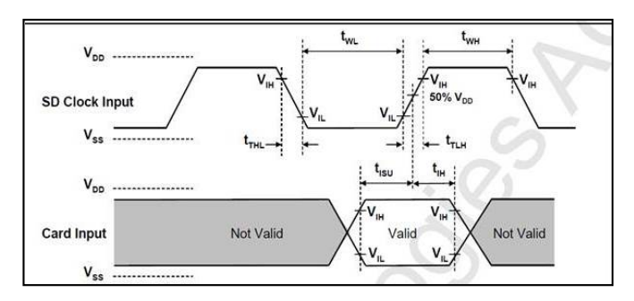
Infineon Technologies Asia Pacific Pte Ltd, 8 Kallang Sector, SINGAPORE. 349282.

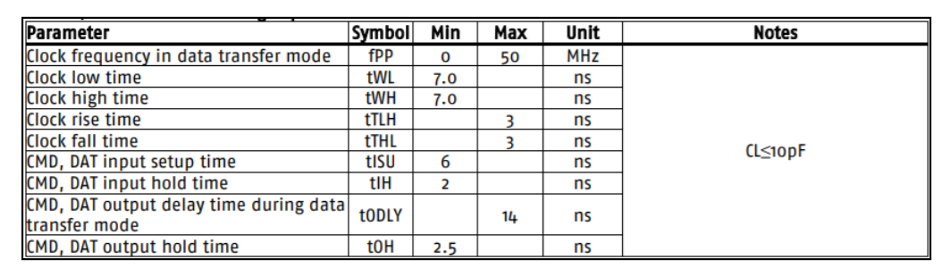
Overview



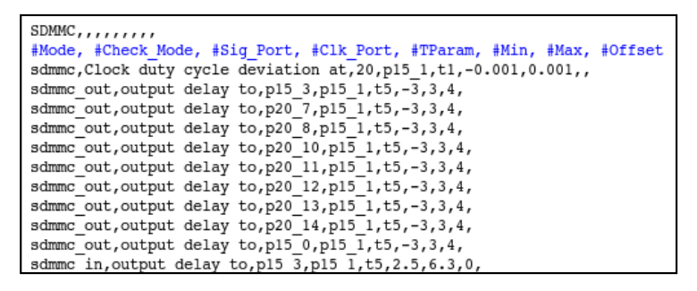
- Datasheet input must be in tabular format.
- Abstract model is coded in Cadence specific System Level Notation (SLN) Language.
- Assertions are generated in SystemVerilog Assertion (SVA) format.

Extraction of IO timing parameters



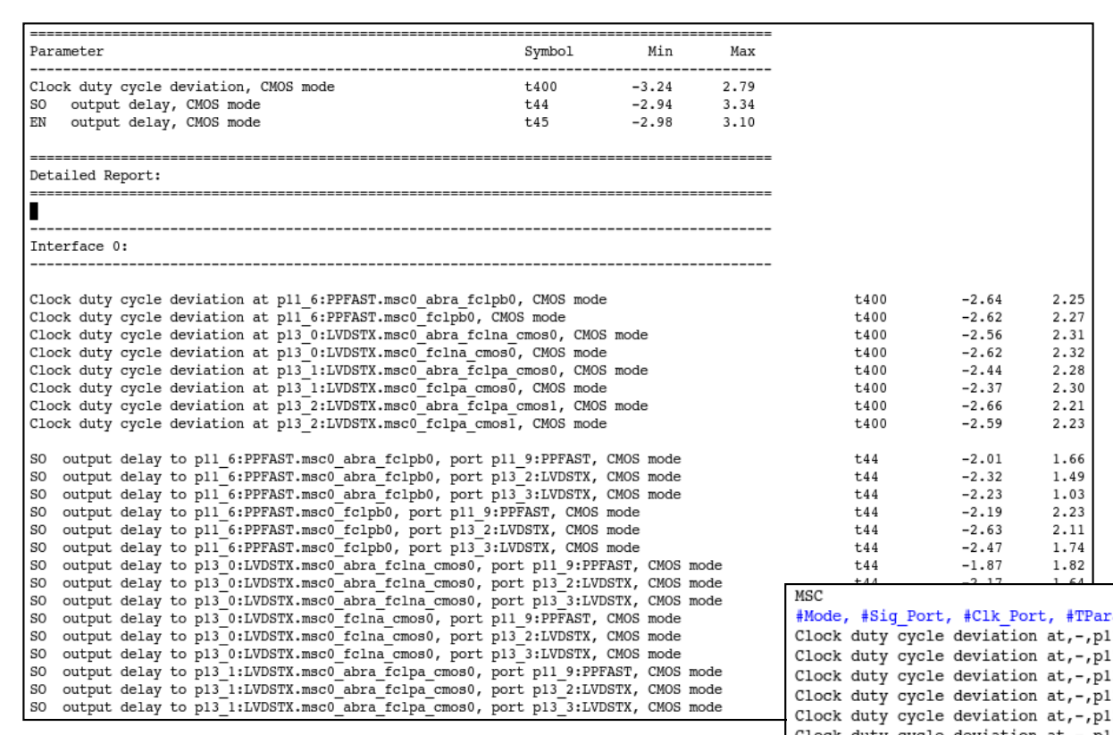


Parameter	Symbol	Min	Max	Unit	Notes
Clock frequency in data transfer mode	fPP	0	50	MHz	CL≤10pF
Clock low time	tWL	7.0		ns	
Clock high time	tWH	7.0		ns	
Clock rise time	tTLH		3	ns	
Clock fall time	tTHL		3	ns	
CMD, DAT input setup time	tISU	6		ns	
CMD, DAT input hold time	tIH	2		ns	
CMD, DAT output delay time during data transfer mode	tODLY		14	ns	
CMD, DAT output hold time	tOH	2.5		ns	

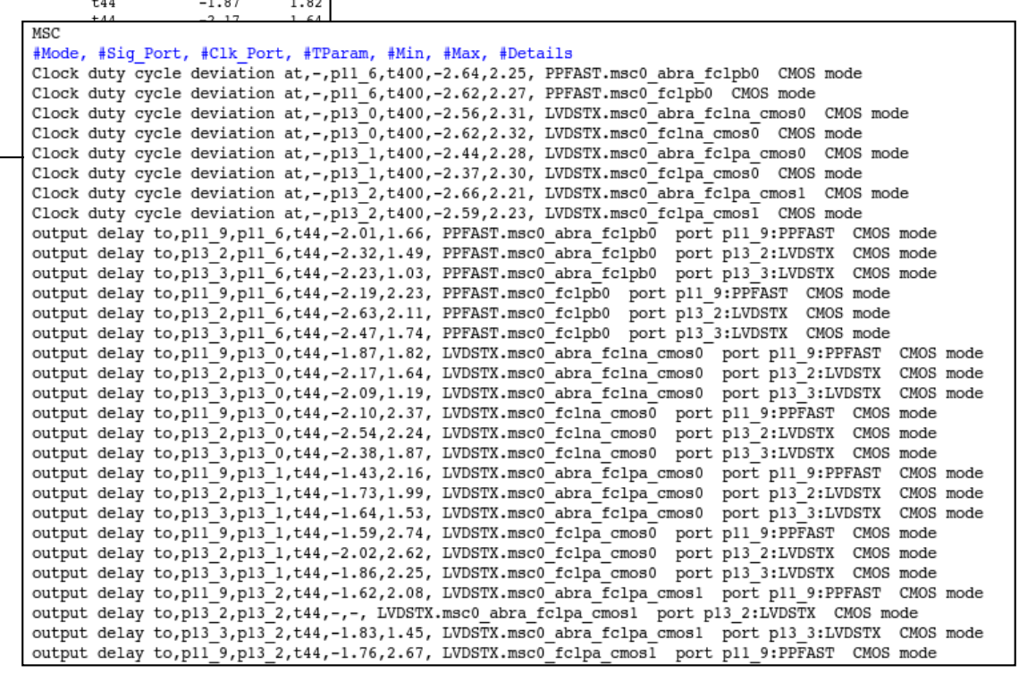


```
SDMMC,,,,,,,,,
#Mode, #Check_Mode, #Sig_Port, #Clk_Port, #TParam, #Min, #Max, #Offset
sdmmc,Clock duty cycle deviation at,20,p15_1,t1,-0.001,0.001,,
sdmmc_out,output delay to,p15_3,p15_1,t5,-3,3,4,
sdmmc_out,output delay to,p20_7,p15_1,t5,-3,3,4,
sdmmc_out,output delay to,p20_8,p15_1,t5,-3,3,4,
sdmmc_out,output delay to,p20_10,p15_1,t5,-3,3,4,
sdmmc_out,output delay to,p20_11,p15_1,t5,-3,3,4,
sdmmc_out,output delay to,p20_12,p15_1,t5,-3,3,4,
sdmmc_out,output delay to,p20_13,p15_1,t5,-3,3,4,
sdmmc_out,output delay to,p20_14,p15_1,t5,-3,3,4,
sdmmc_out,output delay to,p15_0,p15_1,t5,-3,3,4,
sdmmc_in,output delay to,p15_3,p15_1,t5,2.5,6.3,0,
```

If IO Timing Parameters are specified in the datasheet, extract them to a csv (comma separated vector) format file.



```
========================================================================================
Parameter                                                  Symbol        Min       Max
----------------------------------------------------------------------------------------
Clock duty cycle deviation, CMOS mode                      t400         -3.24      2.79
SO   output delay, CMOS mode                               t44          -2.94      3.34
EN   output delay, CMOS mode                               t45          -2.98      3.10

========================================================================================
Detailed Report:
========================================================================================

----------------------------------------------------------------------------------------
Interface 0:
----------------------------------------------------------------------------------------

Clock duty cycle deviation at p11_6:PPFAST.msc0_abra_fclpb0, CMOS mode            t400     -2.64    2.25
Clock duty cycle deviation at p11_6:PPFAST.msc0_fclpb0, CMOS mode                 t400     -2.62    2.27
Clock duty cycle deviation at p13_0:LVDSTX.msc0_abra_fclna_cmos0, CMOS mode       t400     -2.56    2.31
Clock duty cycle deviation at p13_0:LVDSTX.msc0_fclna_cmos0, CMOS mode            t400     -2.62    2.32
Clock duty cycle deviation at p13_1:LVDSTX.msc0_abra_fclpa_cmos0, CMOS mode       t400     -2.44    2.28
Clock duty cycle deviation at p13_1:LVDSTX.msc0_fclpa_cmos0, CMOS mode            t400     -2.37    2.30
Clock duty cycle deviation at p13_2:LVDSTX.msc0_abra_fclpa_cmos1, CMOS mode       t400     -2.66    2.21
Clock duty cycle deviation at p13_2:LVDSTX.msc0_fclpa_cmos1, CMOS mode            t400     -2.59    2.23

SO  output delay to p11_6:PPFAST.msc0_abra_fclpb0, port p11_9:PPFAST, CMOS mode          t44     -2.01    1.66
SO  output delay to p11_6:PPFAST.msc0_abra_fclpb0, port p13_2:LVDSTX, CMOS mode          t44     -2.32    1.49
SO  output delay to p11_6:PPFAST.msc0_abra_fclpb0, port p13_3:LVDSTX, CMOS mode          t44     -2.23    1.03
SO  output delay to p11_6:PPFAST.msc0_fclpb0, port p11_9:PPFAST, CMOS mode               t44     -2.19    2.23
SO  output delay to p11_6:PPFAST.msc0_fclpb0, port p13_2:LVDSTX, CMOS mode               t44     -2.63    2.11
SO  output delay to p11_6:PPFAST.msc0_fclpb0, port p13_3:LVDSTX, CMOS mode               t44     -2.47    1.74
SO  output delay to p13_0:LVDSTX.msc0_abra_fclna_cmos0, port p11_9:PPFAST, CMOS mode     t44     -1.87    1.82
SO  output delay to p13_0:LVDSTX.msc0_abra_fclna_cmos0, port p13_2:LVDSTX, CMOS mode     t44     -2.17    1.64
SO  output delay to p13_0:LVDSTX.msc0_abra_fclna_cmos0, port p13_3:LVDSTX, CMOS mode
SO  output delay to p13_0:LVDSTX.msc0_fclna_cmos0, port p11_9:PPFAST, CMOS mode
SO  output delay to p13_0:LVDSTX.msc0_fclna_cmos0, port p13_2:LVDSTX, CMOS mode
SO  output delay to p13_0:LVDSTX.msc0_fclna_cmos0, port p13_3:LVDSTX, CMOS mode
SO  output delay to p13_1:LVDSTX.msc0_abra_fclpa_cmos0, port p11_9:PPFAST, CMOS mode
SO  output delay to p13_1:LVDSTX.msc0_abra_fclpa_cmos0, port p13_2:LVDSTX, CMOS mode
SO  output delay to p13_1:LVDSTX.msc0_abra_fclpa_cmos0, port p13_3:LVDSTX, CMOS mode
```

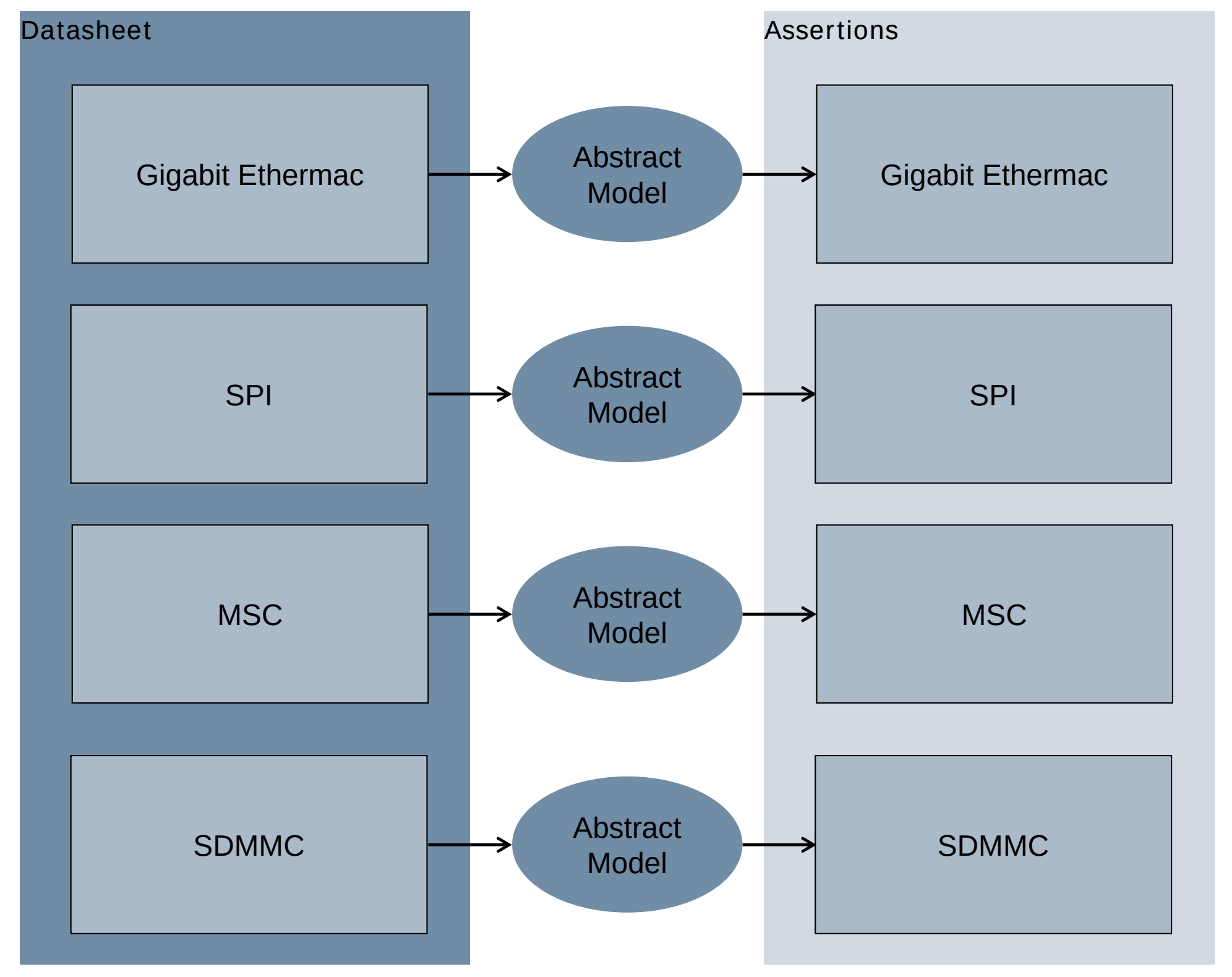


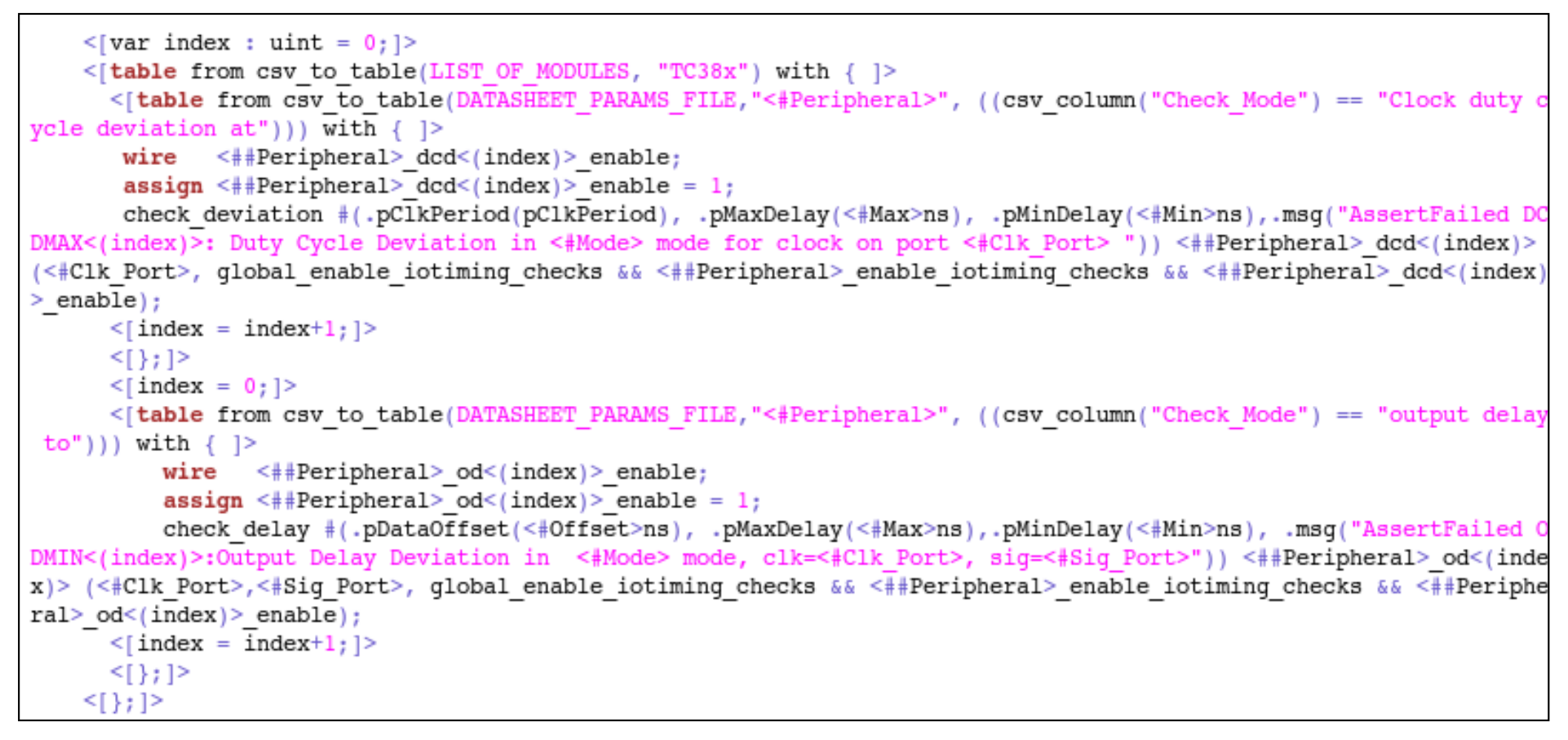
```
MSC
#Mode, #Sig_Port, #Clk_Port, #TParam, #Min, #Max, #Details
Clock duty cycle deviation at,-,p11_6,t400,-2.64,2.25, PPFAST.msc0_abra_fclpb0  CMOS mode
Clock duty cycle deviation at,-,p11_6,t400,-2.62,2.27, PPFAST.msc0_fclpb0  CMOS mode
Clock duty cycle deviation at,-,p13_0,t400,-2.56,2.31, LVDSTX.msc0_abra_fclna_cmos0  CMOS mode
Clock duty cycle deviation at,-,p13_0,t400,-2.62,2.32, LVDSTX.msc0_fclna_cmos0  CMOS mode
Clock duty cycle deviation at,-,p13_1,t400,-2.44,2.28, LVDSTX.msc0_abra_fclpa_cmos0  CMOS mode
Clock duty cycle deviation at,-,p13_1,t400,-2.37,2.30, LVDSTX.msc0_fclpa_cmos0  CMOS mode
Clock duty cycle deviation at,-,p13_2,t400,-2.66,2.21, LVDSTX.msc0_abra_fclpa_cmos1  CMOS mode
Clock duty cycle deviation at,-,p13_2,t400,-2.59,2.23, LVDSTX.msc0_fclpa_cmos1  CMOS mode
output delay to,p11_9,p11_6,t44,-2.01,1.66, PPFAST.msc0_abra_fclpb0  port p11_9:PPFAST  CMOS mode
output delay to,p13_2,p11_6,t44,-2.32,1.49, PPFAST.msc0_abra_fclpb0  port p13_2:LVDSTX  CMOS mode
output delay to,p13_3,p11_6,t44,-2.23,1.03, PPFAST.msc0_abra_fclpb0  port p13_3:LVDSTX  CMOS mode
output delay to,p11_9,p11_6,t44,-2.19,2.23, PPFAST.msc0_fclpb0  port p11_9:PPFAST  CMOS mode
output delay to,p13_2,p11_6,t44,-2.63,2.11, PPFAST.msc0_fclpb0  port p13_2:LVDSTX  CMOS mode
output delay to,p13_3,p11_6,t44,-2.47,1.74, PPFAST.msc0_fclpb0  port p13_3:LVDSTX  CMOS mode
output delay to,p11_9,p13_0,t44,-1.87,1.82, LVDSTX.msc0_abra_fclna_cmos0  port p11_9:PPFAST  CMOS mode
output delay to,p13_2,p13_0,t44,-2.17,1.64, LVDSTX.msc0_abra_fclna_cmos0  port p13_2:LVDSTX  CMOS mode
output delay to,p13_3,p13_0,t44,-2.09,1.19, LVDSTX.msc0_abra_fclna_cmos0  port p13_3:LVDSTX  CMOS mode
output delay to,p11_9,p13_0,t44,-2.10,2.37, LVDSTX.msc0_fclna_cmos0  port p11_9:PPFAST  CMOS mode
output delay to,p13_2,p13_0,t44,-2.54,2.24, LVDSTX.msc0_fclna_cmos0  port p13_2:LVDSTX  CMOS mode
output delay to,p13_3,p13_0,t44,-2.38,1.87, LVDSTX.msc0_fclna_cmos0  port p13_3:LVDSTX  CMOS mode
output delay to,p11_9,p13_1,t44,-1.43,2.16, LVDSTX.msc0_abra_fclpa_cmos0  port p11_9:PPFAST  CMOS mode
output delay to,p13_2,p13_1,t44,-1.73,1.99, LVDSTX.msc0_abra_fclpa_cmos0  port p13_2:LVDSTX  CMOS mode
output delay to,p13_3,p13_1,t44,-1.64,1.53, LVDSTX.msc0_abra_fclpa_cmos0  port p13_3:LVDSTX  CMOS mode
output delay to,p11_9,p13_1,t44,-1.59,2.74, LVDSTX.msc0_fclpa_cmos0  port p11_9:PPFAST  CMOS mode
output delay to,p13_2,p13_1,t44,-2.02,2.62, LVDSTX.msc0_fclpa_cmos0  port p13_2:LVDSTX  CMOS mode
output delay to,p13_3,p13_1,t44,-1.86,2.25, LVDSTX.msc0_fclpa_cmos0  port p13_3:LVDSTX  CMOS mode
output delay to,p11_9,p13_2,t44,-1.62,2.08, LVDSTX.msc0_abra_fclpa_cmos1  port p11_9:PPFAST  CMOS mode
output delay to,p13_2,p13_2,t44,-,-, LVDSTX.msc0_abra_fclpa_cmos1  port p13_2:LVDSTX  CMOS mode
output delay to,p13_3,p13_2,t44,-1.83,1.45, LVDSTX.msc0_abra_fclpa_cmos1  port p13_3:LVDSTX  CMOS mode
output delay to,p11_9,p13_2,t44,-1.76,2.67, LVDSTX.msc0_fclpa_cmos1  port p11_9:PPFAST  CMOS mode
```

If parameters are not available in the datasheet, extract them from STA report into a csv file.

Abstract Model

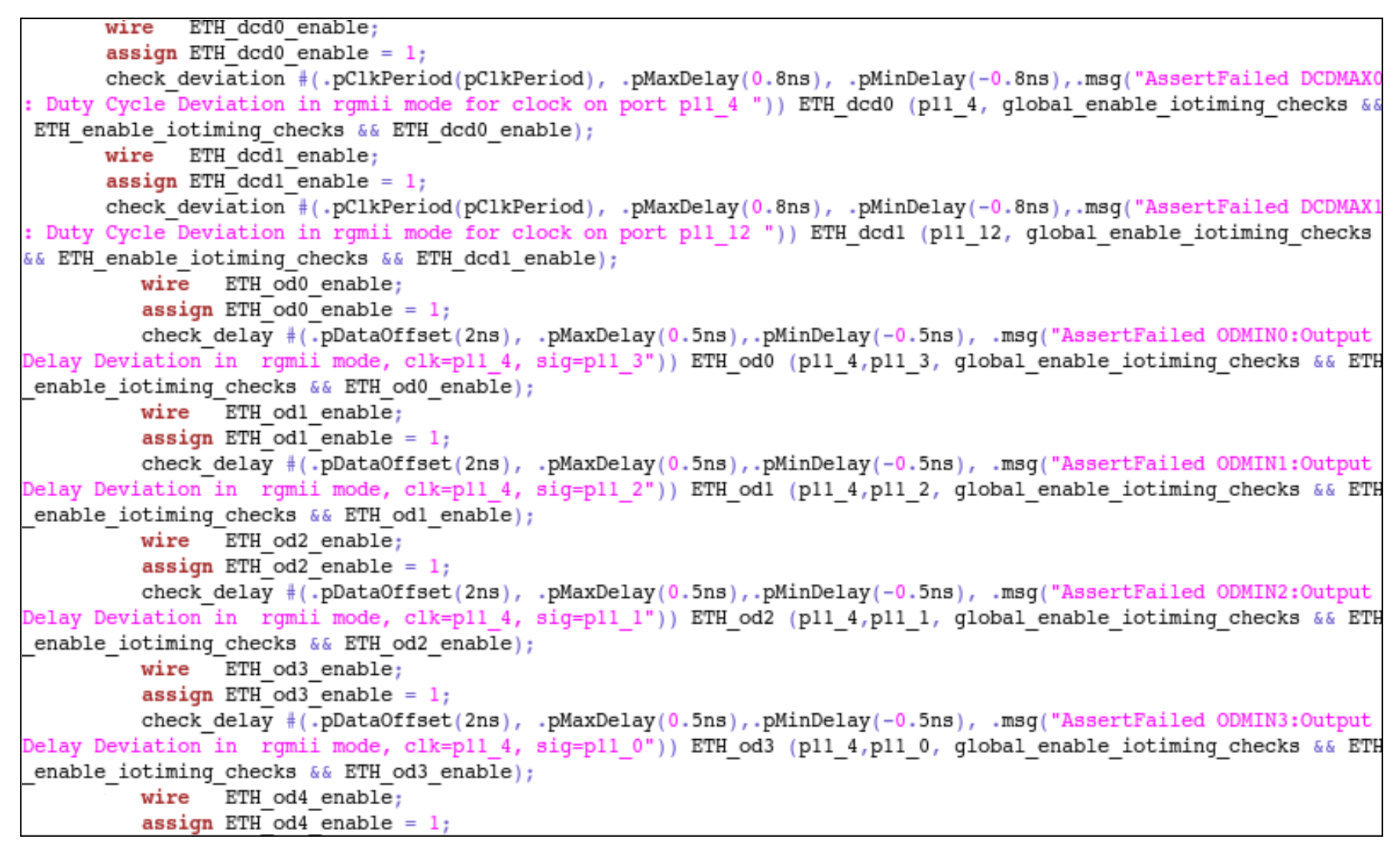
The abstract model to generate the assertions is written in SLN. It can be reused for different modules i.e. Gigabit Ethermac, SPI, MSC and SDMMC etc





```
    <[var index : uint = 0;]>
    <[table from csv_to_table(LIST_OF_MODULES, "TC38x") with { ]>
      <[table from csv_to_table(DATASHEET_PARAMS_FILE,"<#Peripheral>", ((csv_column("Check_Mode") == "Clock duty c
ycle deviation at"))) with { ]>
       wire   <##Peripheral>_dcd<(index)>_enable;
       assign <##Peripheral>_dcd<(index)>_enable = 1;
       check_deviation #(.pClkPeriod(pClkPeriod), .pMaxDelay(<#Max>ns), .pMinDelay(<#Min>ns),.msg("AssertFailed DC
DMAX<(index)>: Duty Cycle Deviation in <#Mode> mode for clock on port <#Clk_Port> ")) <##Peripheral>_dcd<(index)>
(<#Clk_Port>, global_enable_iotiming_checks && <##Peripheral>_enable_iotiming_checks && <##Peripheral>_dcd<(index)
>_enable);
      <[index = index+1;]>
      <[};]>
      <[index = 0;]>
      <[table from csv_to_table(DATASHEET_PARAMS_FILE,"<#Peripheral>", ((csv_column("Check_Mode") == "output delay
 to"))) with { ]>
         wire   <##Peripheral>_od<(index)>_enable;
         assign <##Peripheral>_od<(index)>_enable = 1;
         check_delay #(.pDataOffset(<#Offset>ns), .pMaxDelay(<#Max>ns),.pMinDelay(<#Min>ns), .msg("AssertFailed O
DMIN<(index)>:Output Delay Deviation in  <#Mode> mode, clk=<#Clk_Port>, sig=<#Sig_Port>")) <##Peripheral>_od<(inde
x)> (<#Clk_Port>,<#Sig_Port>, global_enable_iotiming_checks && <##Peripheral>_enable_iotiming_checks && <##Periphe
ral>_od<(index)>_enable);
      <[index = index+1;]>
      <[};]>
    <[};]>
```

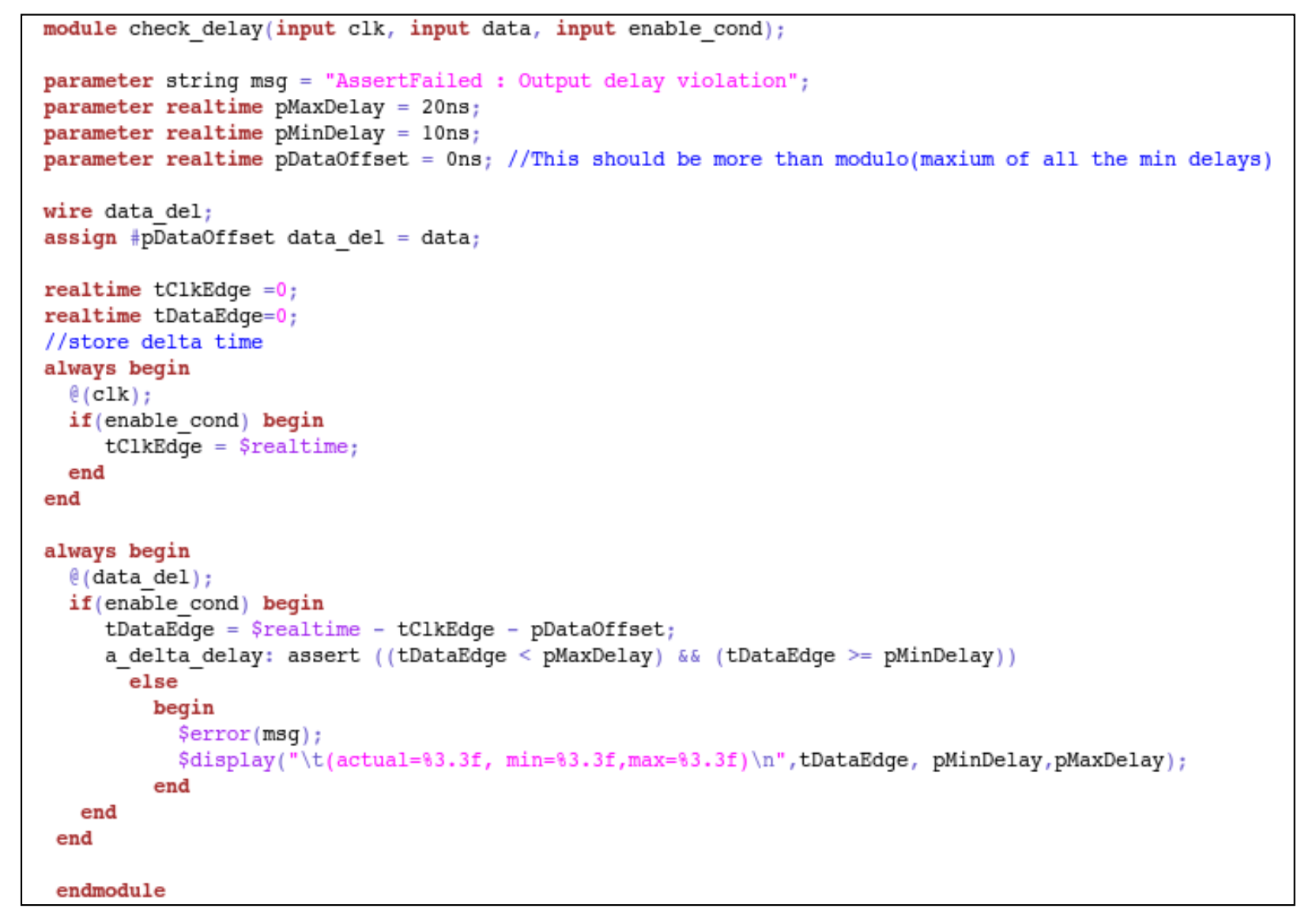
Perspec generated assertions



```
      wire   ETH_dcd0_enable;
      assign ETH_dcd0_enable = 1;
      check_deviation #(.pClkPeriod(pClkPeriod), .pMaxDelay(0.8ns), .pMinDelay(-0.8ns),.msg("AssertFailed DCDMAX0
: Duty Cycle Deviation in rgmii mode for clock on port p11_4 ")) ETH_dcd0 (p11_4, global_enable_iotiming_checks &&
 ETH_enable_iotiming_checks && ETH_dcd0_enable);
      wire   ETH_dcd1_enable;
      assign ETH_dcd1_enable = 1;
      check_deviation #(.pClkPeriod(pClkPeriod), .pMaxDelay(0.8ns), .pMinDelay(-0.8ns),.msg("AssertFailed DCDMAX1
: Duty Cycle Deviation in rgmii mode for clock on port p11_12 ")) ETH_dcd1 (p11_12, global_enable_iotiming_checks
&& ETH_enable_iotiming_checks && ETH_dcd1_enable);
         wire   ETH_od0_enable;
         assign ETH_od0_enable = 1;
         check_delay #(.pDataOffset(2ns), .pMaxDelay(0.5ns),.pMinDelay(-0.5ns), .msg("AssertFailed ODMIN0:Output
Delay Deviation in  rgmii mode, clk=p11_4, sig=p11_3")) ETH_od0 (p11_4,p11_3, global_enable_iotiming_checks && ETH
_enable_iotiming_checks && ETH_od0_enable);
         wire   ETH_od1_enable;
         assign ETH_od1_enable = 1;
         check_delay #(.pDataOffset(2ns), .pMaxDelay(0.5ns),.pMinDelay(-0.5ns), .msg("AssertFailed ODMIN1:Output
Delay Deviation in  rgmii mode, clk=p11_4, sig=p11_2")) ETH_od1 (p11_4,p11_2, global_enable_iotiming_checks && ETH
_enable_iotiming_checks && ETH_od1_enable);
         wire   ETH_od2_enable;
         assign ETH_od2_enable = 1;
         check_delay #(.pDataOffset(2ns), .pMaxDelay(0.5ns),.pMinDelay(-0.5ns), .msg("AssertFailed ODMIN2:Output
Delay Deviation in  rgmii mode, clk=p11_4, sig=p11_1")) ETH_od2 (p11_4,p11_1, global_enable_iotiming_checks && ETH
_enable_iotiming_checks && ETH_od2_enable);
         wire   ETH_od3_enable;
         assign ETH_od3_enable = 1;
         check_delay #(.pDataOffset(2ns), .pMaxDelay(0.5ns),.pMinDelay(-0.5ns), .msg("AssertFailed ODMIN3:Output
Delay Deviation in  rgmii mode, clk=p11_4, sig=p11_0")) ETH_od3 (p11_4,p11_0, global_enable_iotiming_checks && ETH
_enable_iotiming_checks && ETH_od3_enable);
         wire   ETH_od4_enable;
         assign ETH_od4_enable = 1;
```

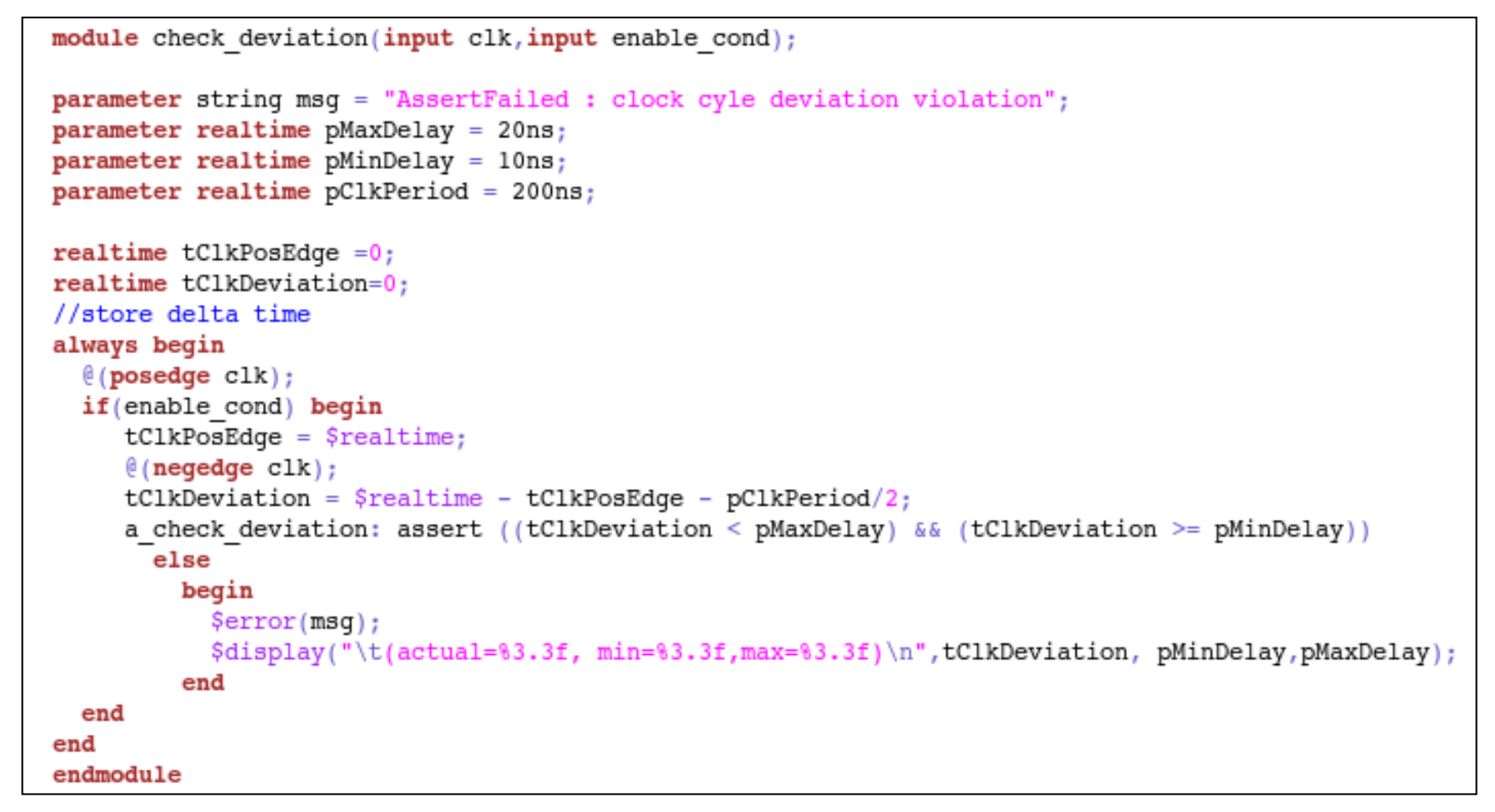
The code snippet above shows how Perspec reads the timing parameters from the csv files to generate all the required assertions for Ethermac module. Similar assertions are generated for all the modules.

Basic Assertions



```
module check_delay(input clk, input data, input enable_cond);

parameter string msg = "AssertFailed : Output delay violation";
parameter realtime pMaxDelay = 20ns;
parameter realtime pMinDelay = 10ns;
parameter realtime pDataOffset = 0ns; //This should be more than modulo(maxium of all the min delays)

wire data_del;
assign #pDataOffset data_del = data;

realtime tClkEdge =0;
realtime tDataEdge=0;
//store delta time
always begin
  @(clk);
  if(enable_cond) begin
     tClkEdge = $realtime;
  end
end

always begin
  @(data_del);
  if(enable_cond) begin
     tDataEdge = $realtime - tClkEdge - pDataOffset;
     a_delta_delay: assert ((tDataEdge < pMaxDelay) && (tDataEdge >= pMinDelay))
       else
         begin
           $error(msg);
           $display("\t(actual=%3.3f, min=%3.3f,max=%3.3f)\n",tDataEdge, pMinDelay,pMaxDelay);
         end
   end
 end

 endmodule
```

Basic assertion to check output delay parameter

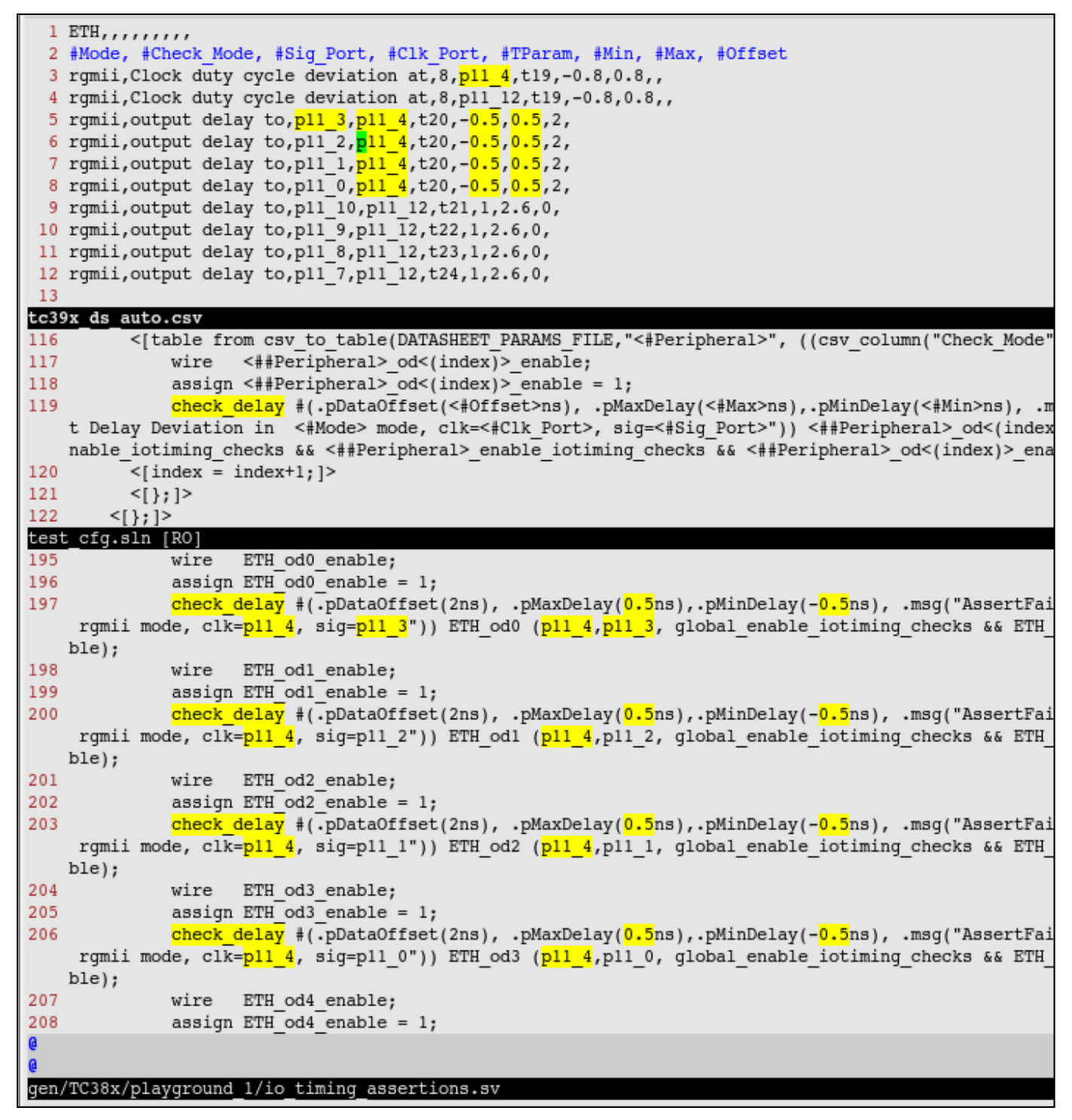


```
module check_deviation(input clk,input enable_cond);

parameter string msg = "AssertFailed : clock cyle deviation violation";
parameter realtime pMaxDelay = 20ns;
parameter realtime pMinDelay = 10ns;
parameter realtime pClkPeriod = 200ns;

realtime tClkPosEdge =0;
realtime tClkDeviation=0;
//store delta time
always begin
  @(posedge clk);
  if(enable_cond) begin
     tClkPosEdge = $realtime;
     @(negedge clk);
     tClkDeviation = $realtime - tClkPosEdge - pClkPeriod/2;
     a_check_deviation: assert ((tClkDeviation < pMaxDelay) && (tClkDeviation >= pMinDelay))
       else
         begin
           $error(msg);
           $display("\t(actual=%3.3f, min=%3.3f,max=%3.3f)\n",tClkDeviation, pMinDelay,pMaxDelay);
         end
  end
end
endmodule
```

Basic assertion to check clock cycle deviation parameter

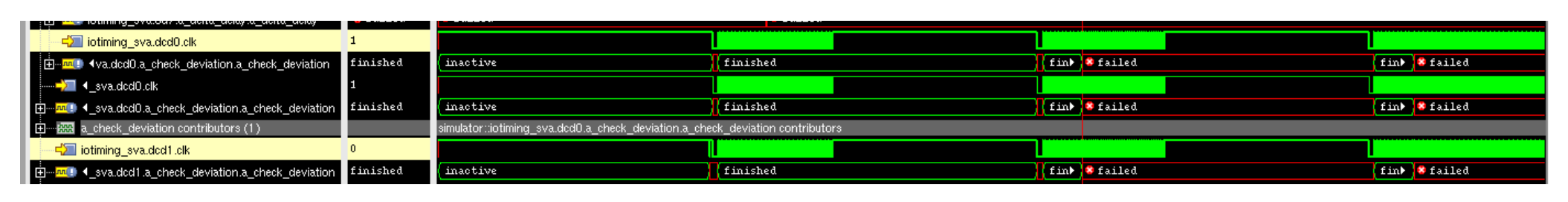
Complete Flow

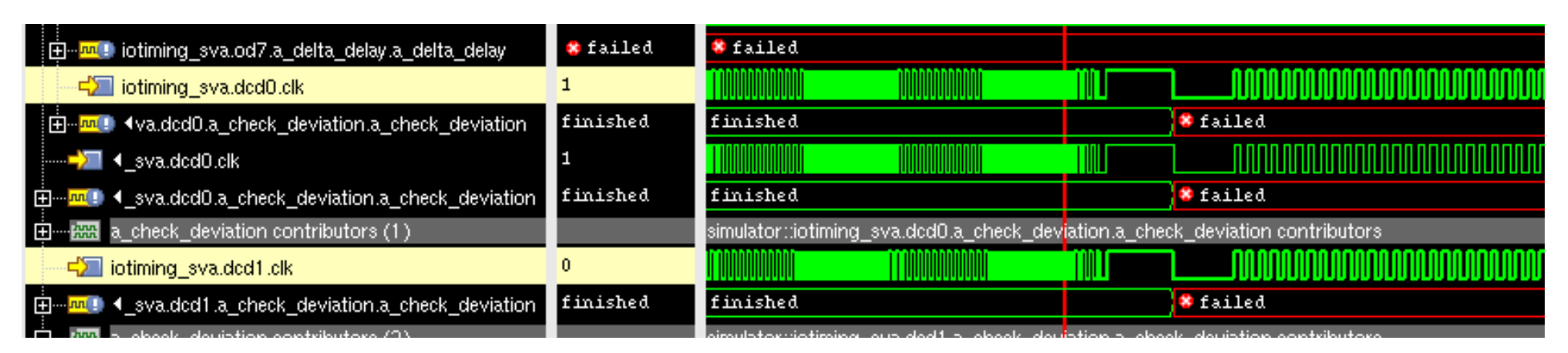
Look at all the steps together in the snippet below which highlights how information from datasheet csv is extracted and instantiated in the generated assertions



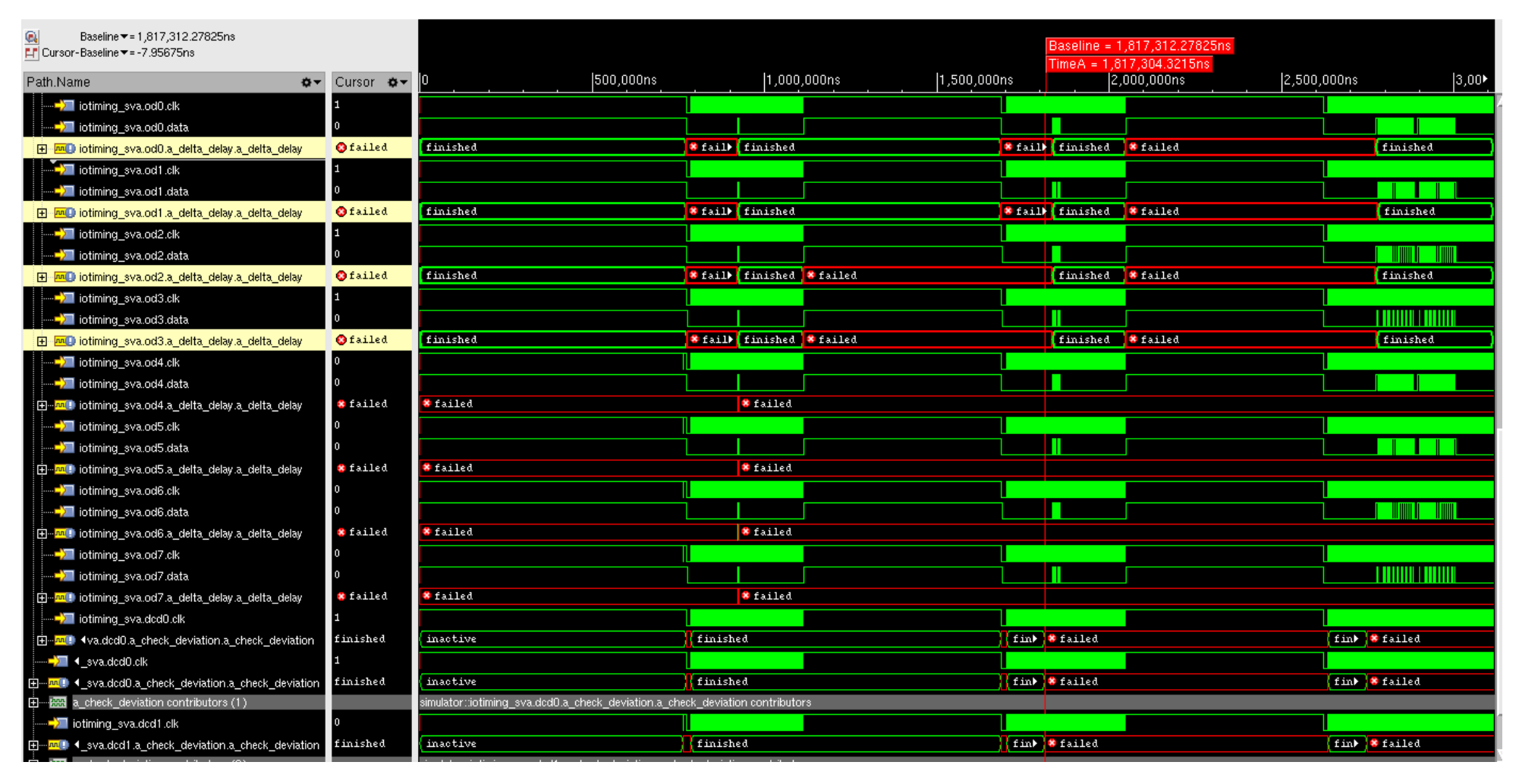
```
ETH,,,,,,,,,
#Mode, #Check_Mode, #Sig_Port, #Clk_Port, #TParam, #Min, #Max, #Offset
rgmii,Clock duty cycle deviation at,8,p11_4,t19,-0.8,0.8,,
rgmii,Clock duty cycle deviation at,8,p11_12,t19,-0.8,0.8,,
rgmii,output delay to,p11_3,p11_4,t20,-0.5,0.5,2,
rgmii,output delay to,p11_2,p11_4,t20,-0.5,0.5,2,
rgmii,output delay to,p11_1,p11_4,t20,-0.5,0.5,2,
rgmii,output delay to,p11_0,p11_4,t20,-0.5,0.5,2,
rgmii,output delay to,p11_10,p11_12,t21,1,2.6,0,
rgmii,output delay to,p11_9,p11_12,t22,1,2.6,0,
rgmii,output delay to,p11_8,p11_12,t23,1,2.6,0,
rgmii,output delay to,p11_7,p11_12,t24,1,2.6,0,

tc39x_ds_auto.csv
      <[table from csv_to_table(DATASHEET_PARAMS_FILE,"<#Peripheral>", ((csv_column("Check_Mode"
          wire   <##Peripheral>_od<(index)>_enable;
          assign <##Peripheral>_od<(index)>_enable = 1;
          check_delay #(.pDataOffset(<#Offset>ns), .pMaxDelay(<#Max>ns),.pMinDelay(<#Min>ns), .m
t Delay Deviation in  <#Mode> mode, clk=<#Clk_Port>, sig=<#Sig_Port>")) <##Peripheral>_od<(index
nable_iotiming_checks && <##Peripheral>_enable_iotiming_checks && <##Peripheral>_od<(index)>_ena
      <[index = index+1;]>
      <[};]>
    <[};]>
test_cfg.sln [RO]
          wire   ETH_od0_enable;
          assign ETH_od0_enable = 1;
          check_delay #(.pDataOffset(2ns), .pMaxDelay(0.5ns),.pMinDelay(-0.5ns), .msg("AssertFai
 rgmii mode, clk=p11_4, sig=p11_3")) ETH_od0 (p11_4,p11_3, global_enable_iotiming_checks && ETH_
ble);
          wire   ETH_od1_enable;
          assign ETH_od1_enable = 1;
          check_delay #(.pDataOffset(2ns), .pMaxDelay(0.5ns),.pMinDelay(-0.5ns), .msg("AssertFai
 rgmii mode, clk=p11_4, sig=p11_2")) ETH_od1 (p11_4,p11_2, global_enable_iotiming_checks && ETH_
ble);
          wire   ETH_od2_enable;
          assign ETH_od2_enable = 1;
          check_delay #(.pDataOffset(2ns), .pMaxDelay(0.5ns),.pMinDelay(-0.5ns), .msg("AssertFai
 rgmii mode, clk=p11_4, sig=p11_1")) ETH_od2 (p11_4,p11_1, global_enable_iotiming_checks && ETH_
ble);
          wire   ETH_od3_enable;
          assign ETH_od3_enable = 1;
          check_delay #(.pDataOffset(2ns), .pMaxDelay(0.5ns),.pMinDelay(-0.5ns), .msg("AssertFai
 rgmii mode, clk=p11_4, sig=p11_0")) ETH_od3 (p11_4,p11_0, global_enable_iotiming_checks && ETH_
ble);
          wire   ETH_od4_enable;
          assign ETH_od4_enable = 1;
@
@
gen/TC38x/playground_1/io_timing_assertions.sv
```

Simulation results for Gigabit Ethermac





Clock duty cycle check fails because testcase changed the clock frequency midway through the simulation.



Some output delay checks pass while some fail. This would require offline debug using logs and waveforms.

Summary

Use of SVA to verify IO timings is very useful. However manually creating all the required assertions could be very challenging. Deploying the power of portable stimulus using Cadence's Perspec tool helps to overcome these challenges with minimal effort in modelling and scripting.

A successful regression with these assertions running in full timing gate level simulation gives a very high confidence for the IO timing sign off.

Contact information

Infineon Technologies Asia Pacific Pte. Ltd.

8, Kallang Sector, Singapore. 349282

T: +65 6876 2096
E: amitesh.khandelwal@infineon.com

www.PosterPresentations.com