Leveraging more from GLS: Using metric driven GLS stimuli to boost timing verification

Sowmya Ega, Richardson Jeyapaul, Kunal Jani

Agenda

I. Traditional GLS Stimuli

II. Brief introduction on Multi cycle paths, False paths and Timing critical paths.

III. Need for GLS Stimuli to complement STA

IV. Functional coverage model – GLS.

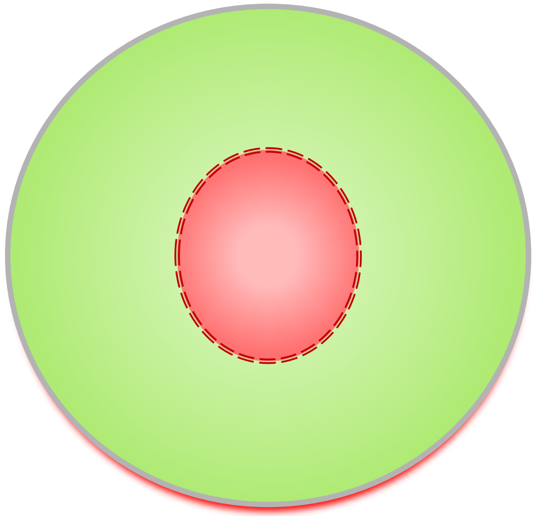
V. Limitations

VI. Conclusion

Traditional GLS Stimuli

Fewer system level tests are run on GLS, owing to following reasons

- Complex Designs
- Lengthy Regression runs
- Enormous test-list
- Schedule



Traditional GLS Stimuli (cntd) ..

Generally, GLS test-list would comprise of

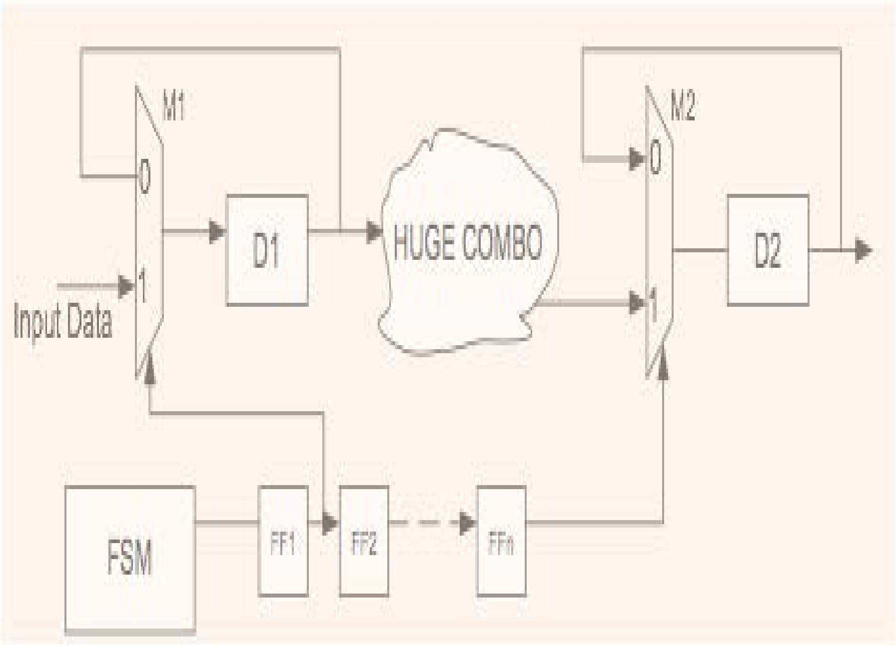
- Reset scenarios
- Critical functional paths at the system level

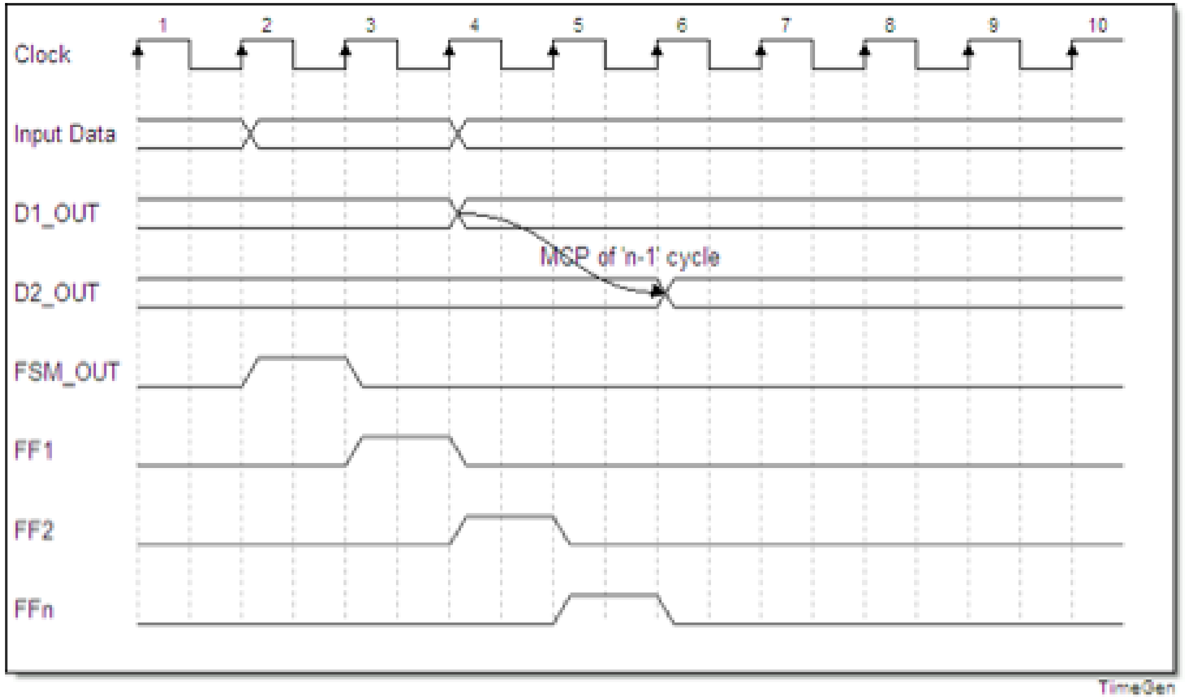
Is the GLS Stimuli Complete ???

Does the tests cover critical timing paths , Multi cycle paths , False paths ?

Multi cycle path

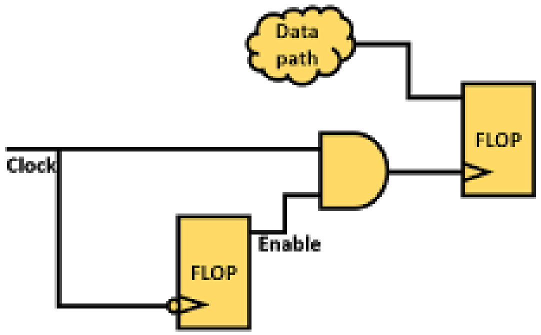
Timing path in which the data launched from one flop is allowed to take more than one clock cycle to reach the destination flop.

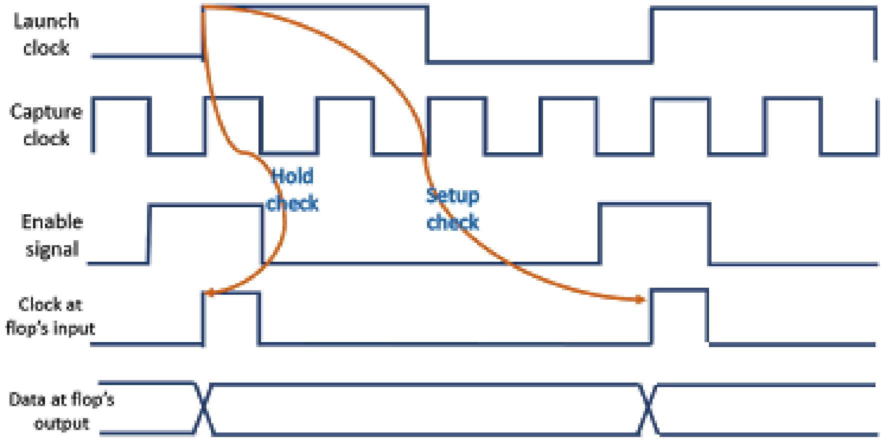




Cntd..

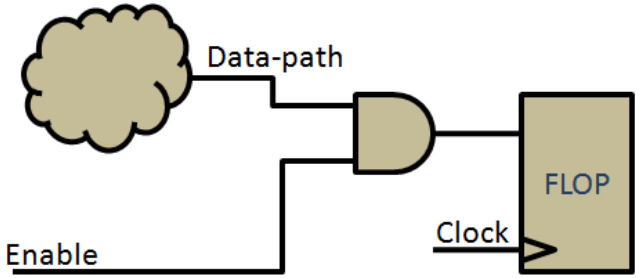
- Multi cycle paths are mostly implemented using the following approaches
  - Clock gated : Clock at the capturing flop is enabled only when the data has to be captured

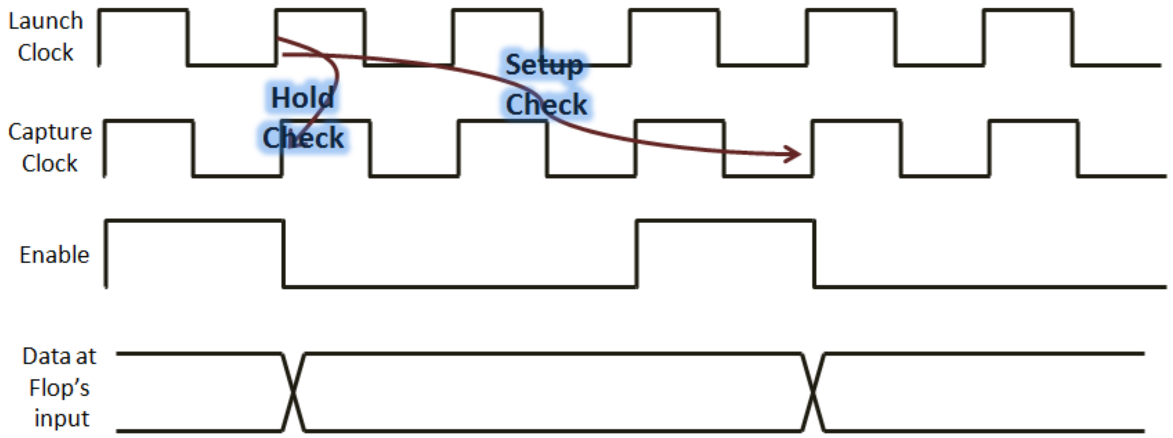




Cntd..

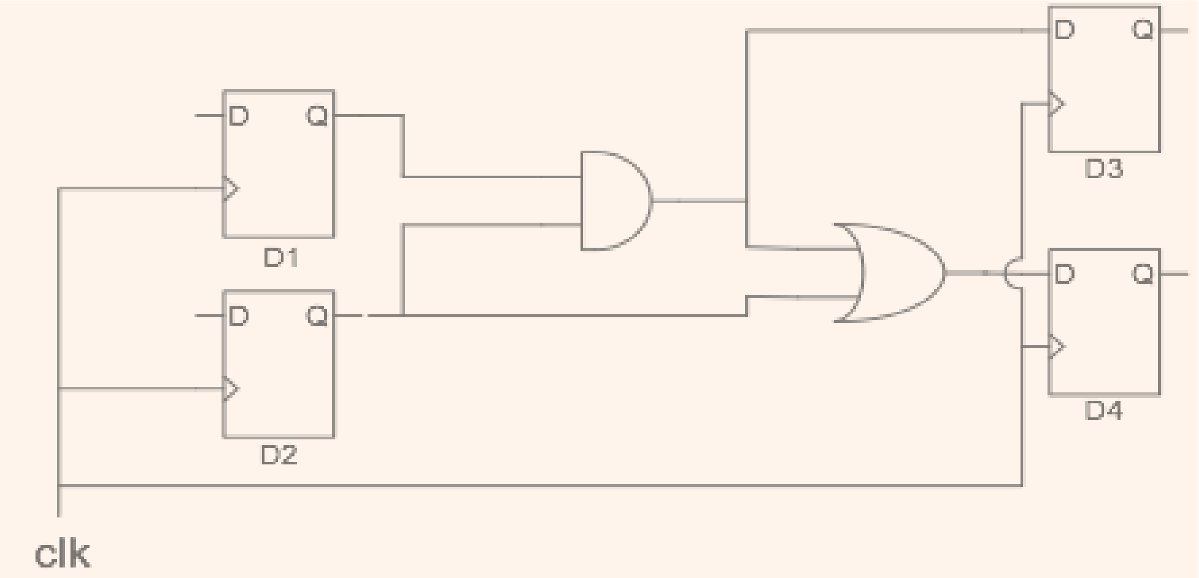
- Data gated : Data is gated at the capturing flop, and toggles only when it is to be captured.





Static False Path

- Timing paths where the change in source registers are not expected to get captured at the destination register within a particular time interval
- These are the paths which exist in the design but those are logically/functionally incorrect

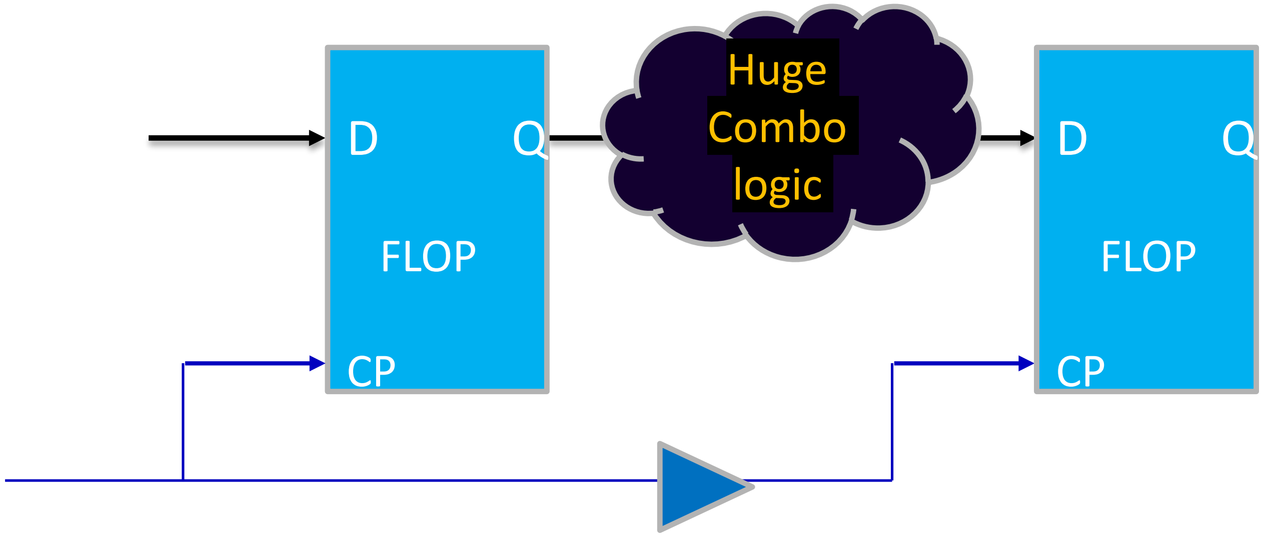


D1->D4 can be treated as a false path.
path

Critical Timing path

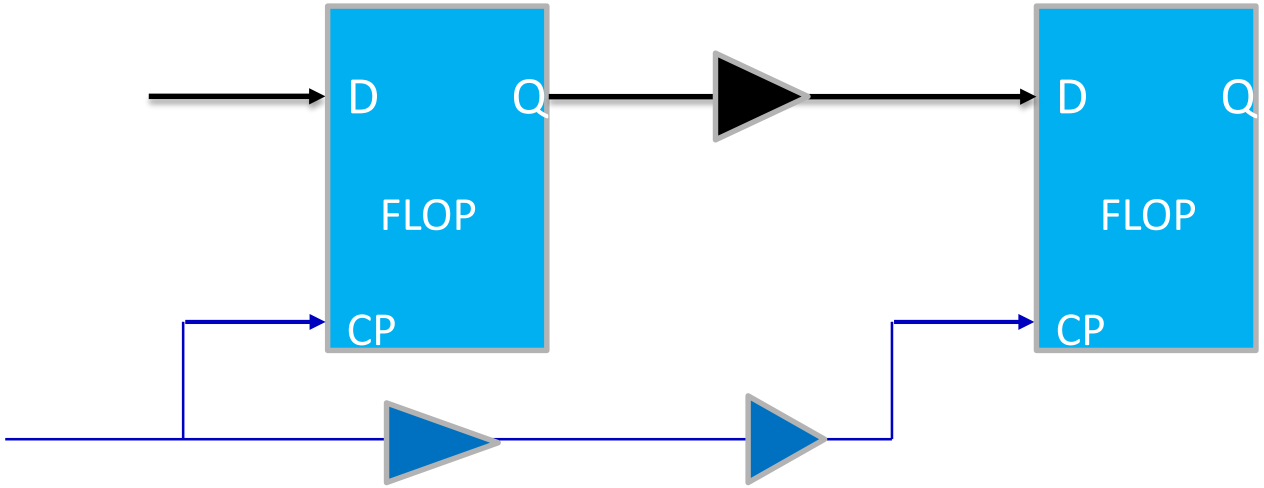
Timing paths whose timing is met marginally by STA tool, and are reported as such by STA tool. These timing paths can be critical on either setup, hold or both.

Setup critical timing path : Critical paths where the delay in the data path is greater than the delay in clock path.



Cntd..

Hold critical timing path: Critical paths where the delay in the clock path is greater than the delay in data path.



GLS stimuli to complement STA

When Static Timing Analysis is run:

- There is no explicit timing verification done on Multi cycle paths, false paths
  - If a Path is wrongly defined as an multi cycle path/false path & whose timing is closed accordingly, is not an multi cycle path/false path – results in a silicon bug.

- The timing critical paths are verified only on Static stimuli.
  - No check to understand the behavior of timing critical paths with real functional stimuli (Dynamic)

GLS stimuli to complement STA (cntd..)

- Hence, to boost timing verification, a need arises to check if the GLS test-list covers Timing critical paths, multi cycle paths and false paths.

- But, how can we check if the GLS test-list covers the timing paths of interest??
    - Develop a Functional Coverage model on Timing paths of interest

How Functional coverage model on Timing paths of interest helps ??

- Functional cover property monitors if a particular timing path gets triggered in a simulation or not.
- Integrate these Functional cover properties in GLS testbench environment.
- Run a regression using GLS testlist on full design.
- Coverage achieved on the Functional Cover properties defines if GLS Stimuli exercises the timing paths of interest , and thus complements the STA by boosting the verification on timing paths of interest.

Automating the generation of Functional cover properties :

Zero-in on the timing paths of interest with the help of Design & Implementation team.

Get the launch-flop (Begin Point) and capture-flop (End Point) of the timing paths of interest.

Based on the timing path (Timing critical / Multi cycle / False path), the construct of the Functional cover property differs.

Develop a script which takes the list of Begin-Points & End-Points of timing paths and automate the generation of Functional cover properties

Functional cover property for a Timing critical path :

Sample Construct :

```
property TimingPath_Sample;
    bit ep_val;
    time lead_time;
    time trail_time;
     bit ep_val_capture;
    @(posedge BeginPoint.CP)   disable iff (!start_chk || (BeginPoint.Q === 1'bx) || (BeginPoint.CP === 1'bx))
     ( $changed(BeginPoint.Q) && ($past(BeginPoint.Q) !== 1'bx)) ##0  (1,ep_val = EndPoint.D,lead_time =
    $time )  |=>  @(posedge EndPoint.CP)  (($changed(EndPoint.D) && EP.D !== ep_val) &&
    ($past(EndPoint.D) !== 1'bx )) ##0 (1,trail_time = $time, report_check(lead_time,
    trail_time,1),ep_val_capture = EndPoint.D, report_capture(ep_val,ep_val_capture,1));
endproperty
TimingPath_sample_cover : cover property(TimingPath_sample) $display(" Timingpath_sample  : BP is %d,
EP is %d, check_x is %d test is %s " ,BP,EP,check_x(BP,EP),getenv("TESTRUN"));
```

BeginPoint & EndPoint operate on different c

Change in Data at BeginPoint

Capture Data at EndPoint

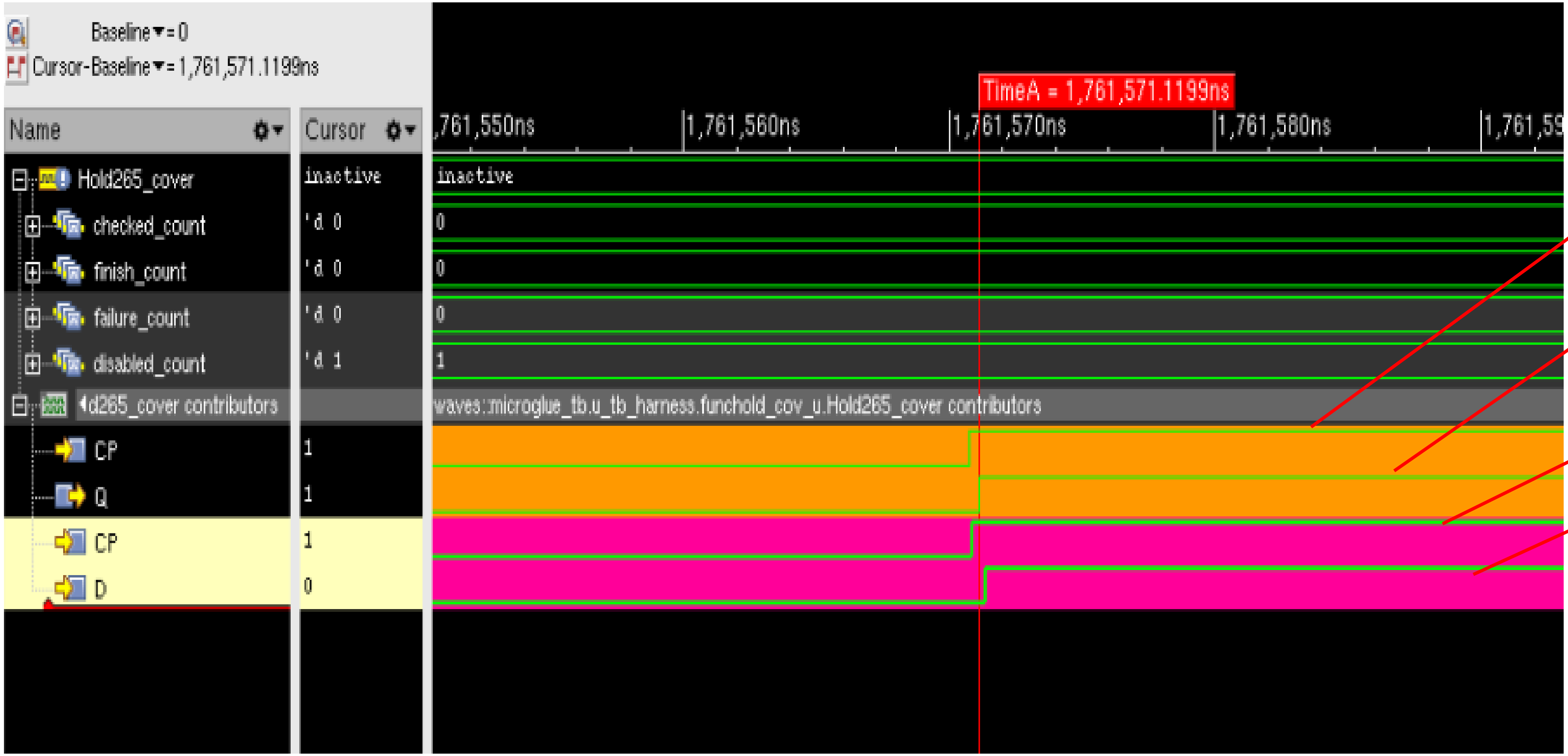
Wait for change in Data at EndPoint at the nearest edge of EndPoint clock after BeginPoint clock edge

Example for a Critical Timing path:

```
`define Begin_Point
microglue_tb.u_muska.u_muskaDigital_top.u_muskaDigital.u_digital_top_lv.u_dig_pwr_gate.u_dig_core.u_subsys.u_apb32_periph_top.pmg_clk_regs.hfoscdivsyn_gen_reg
`define End_Point
microglue_tb.u_muska.u_muskaDigital_top.u_muskaDigital.u_digital_top_lv.u_dig_pwr_gate.u_dig_core.u_subsys.u_apb32_periph_top.pmg_clk_regs.hfoscdivsyn_gen_d_reg


property Hold_265;
 bit ep_val;  time lead_time;  time trail_time;  bit ep_val_capture;
  @(posedge Begin_Point.CP)   disable iff (!start_chk ||  (Begin_Point.Q === 1'bx) || (Begin_Point.CP === 1'bx) ) ( $changed(Begin_Point.Q ) && ($past(Begin_Point.Q) !== 1'bx)) ##0 (1,ep_val = End_Point.D,lead_time = $time)  |=>  @(posedge End_Point.CP) ##0   (($changed(End_Point.D)  &&End_Point.D !== signed’(ep_val)) && ($past(End_Point.D) !== 1'bx )) ##0 (1,trail_time = $time, report_check(lead_time, trail_time,265),ep_val_capture = End_Point.D, report_capture(ep_val,ep_val_capture,265));
endproperty
Hold_265_cover : cover property(Hold_265) $display(" 265  : BP is %d, EP is %d, check_x is %d test is %s " , Begin_Point.Q,End_Point.D,samp_fun::check_x(Begin_Point.Q,End_Point.D),getenv("TESTRUN"))
```

Waveform Snippets :



Begin Point clock

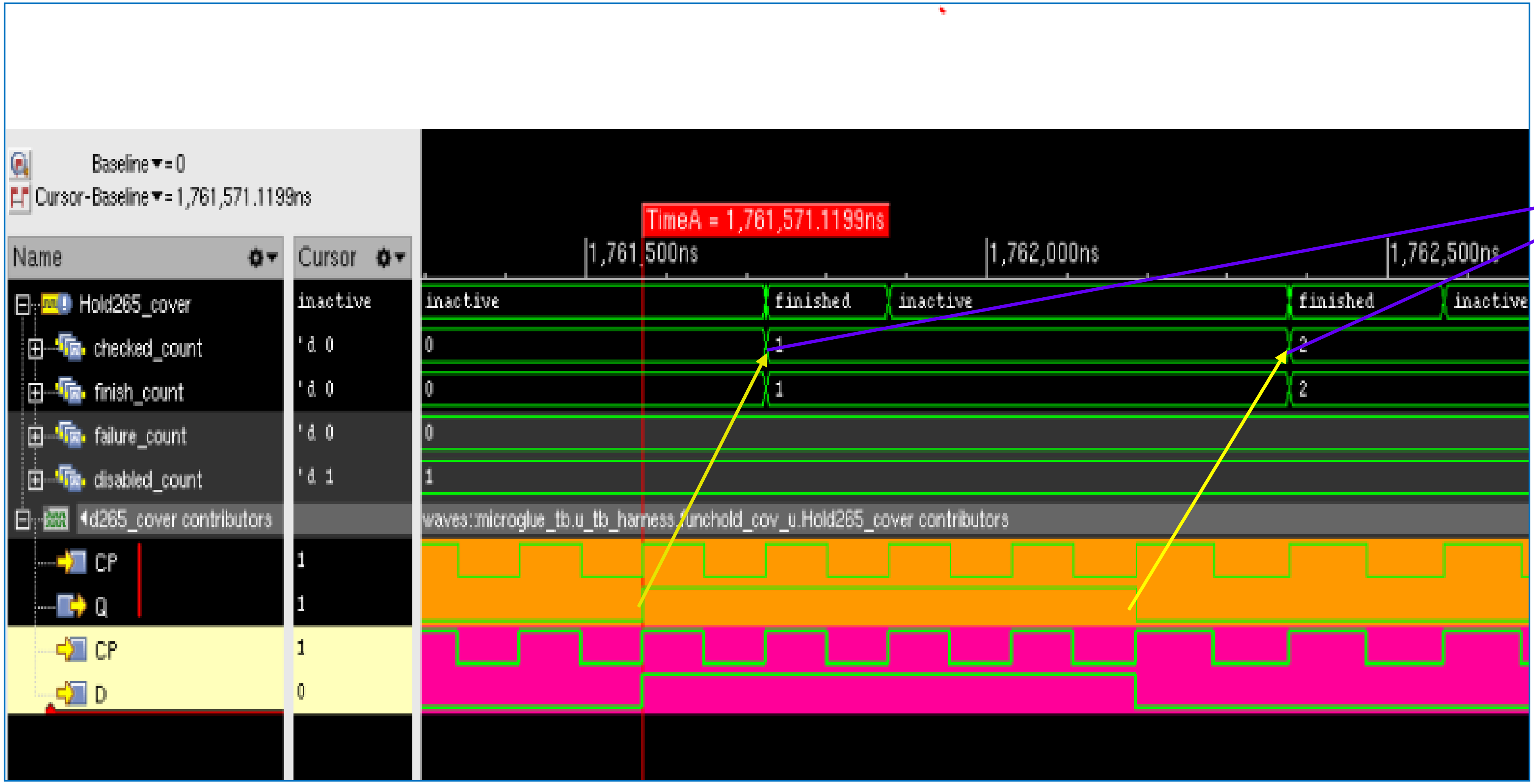
Begin Point Q transition

Endpoint Clock

Endpoint D transition

the transitions

Cntd..



Cover property capturing the transitions

Functional cover property for a clock gated Multi cycle path :

Sample construct for a clock gated MCP :

```
property Hold_Sample_MCP;
  bit ep_val;
  time lead_time;
  time trail_time;
  bit ep_val_capture;
  @(posedge BeginPoint.CP)   disable iff (!start_chk || (BeginPoint.Q === 1'bx) || (BeginPoint.CP === 1'bx)) (
  $changed(BeginPoint.Q) && ($past(BeginPoint.Q) !== 1'bx)) ##0 (1,ep_val = EndPoint.D,lead_time = $time )
  |-> @(posedge BeginPoint.CP) ($stableBeginPoint.D)  |=> @ (posedge EndPoint.CP )
  ($changed(EndPoint.D) && EP.D !== ep_val) && ($past(EndPoint.D) !== 1'bx )) ##0 (1,trail_time = $time,
  report_check(lead_time, trail_time,1),ep_val_capture = EndPoint.D,
  report_capture(ep_val,ep_val_capture,1));

endproperty

  Hold_sample_mcp_cover : cove play(" Hold_sample  : BP is %d, EP is %d,
  check_x is %d test is %s " ,BP, N"));
```

Begin Point and End point clock edges

Change in Data at BeginPoint

Check that the Data transitioned is stable

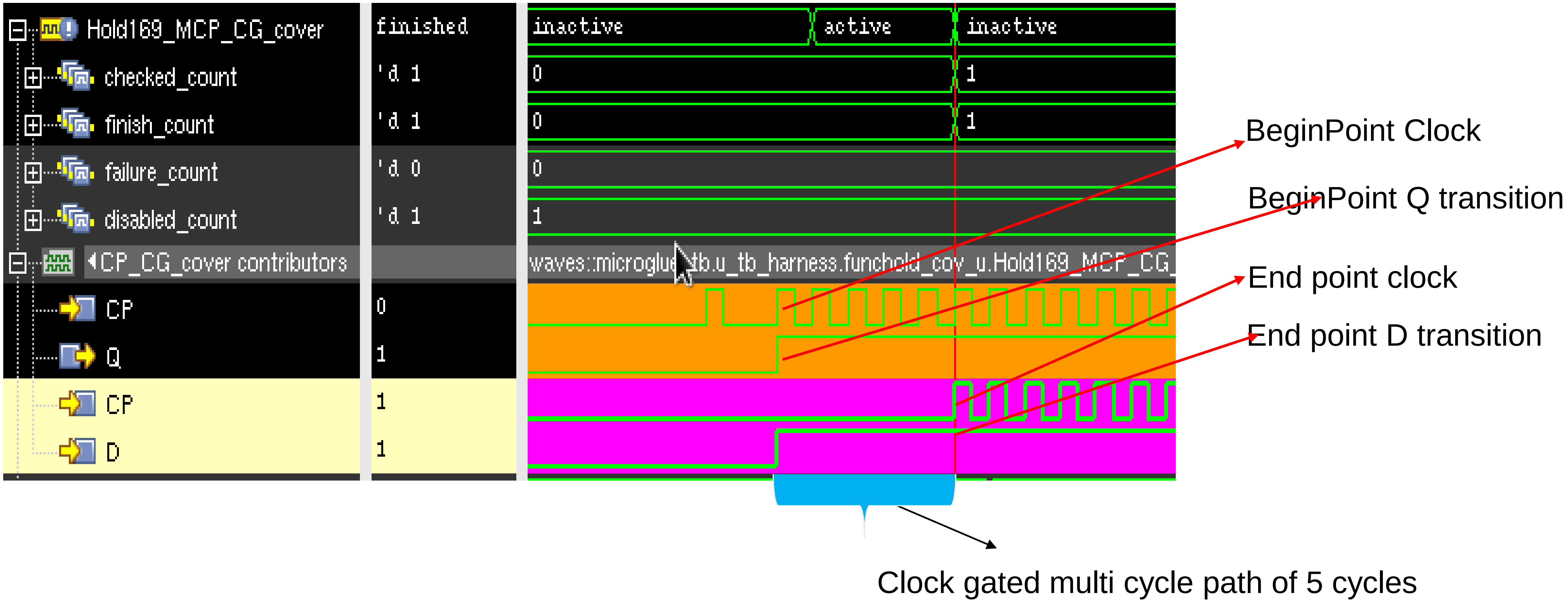
Wait for change in Data at EndPoint at the nearest edge of EndPoint clock after BeginPoint clock edge

Example for a clock gated MCP:

```
`define Launch_Path
proj_tb.u_top.u_projDigital_top.u_projDigital.u_digital_top.aon..edgeDet_clock_dead_edgeDet_syncQ1_reg
`define Capture_Path
proj_tb.u_top.u_projDigital_top.u_projDigital.u_digital_top.aon..edgeDet_clock_dead_edgeDet_syncQ2_reg


property Hold169_MCP_CG;
        bit ep_val; time lead_time; time trail_time;   bit ep_val_capture;
        @(posedge Launch_Path.CP)   disable iff (!start_chk || (Launch_Path.Q === 1'bx) || (Launch_Path.CP === 1'bx) ) (
$changed(Launch_Path.Q ) && ($past(Launch_Path.Q) !== 1'bx)) ##0 (1,ep_val =Capture_Path.D,lead_time = $time)  |-> @(posedge
Launch_Path.CP) $stable(Launch_Path.D)  ##1  @(posedge Capture_Path.CP) ##0  ($changed(Capture_Path.D) &&
($past(Capture_Path.D) !== 1'bx ));
endproperty
Hold169_MCP_CG_cover : cover property(Hold169_MCP_CG) $display(" Hold169_MCP_CG  : BP is %d, EP is %d, check_x is %d
test is %s", Launch_Path.Q,Capture_Path.D,samp_fun::check_x(Launch_Path.Q,Capture_Path.D),getenv("TESTRUN"));
```

Waveform snippet :



Backup

GLS stimuli to complement STA

- But, how do we check if the GLS test-list covers the timing paths of interest??
  - Functional Coverage on Timing paths of interest

GLS stimuli to complement STA

Emphasis on Timing critical paths, Multi cycle paths & False paths :

- Timing critical paths are the paths whose timing is marginally met in STA
  - To understand the behavior of timing critical paths with real functional stimuli (Dynamic)
- Multi cycle paths and False paths are user(Designer) defined
  - If a Path is wrongly defined as an multi cycle path/false path & whose timing is closed accordingly, is not an multi cycle path/false path – results in a silicon bug.

.