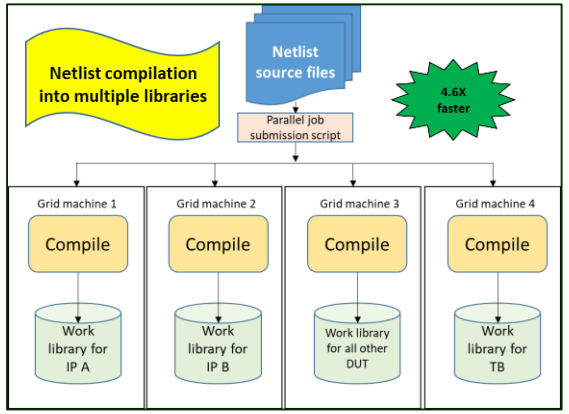
Tired of slow Gate-Level Design Verification? Use these efficient modelling styles and methodology

Mentor® A Siemens Business

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

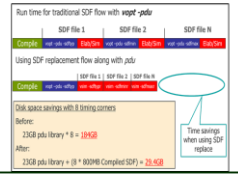
Improving throughput of gate-level simulations

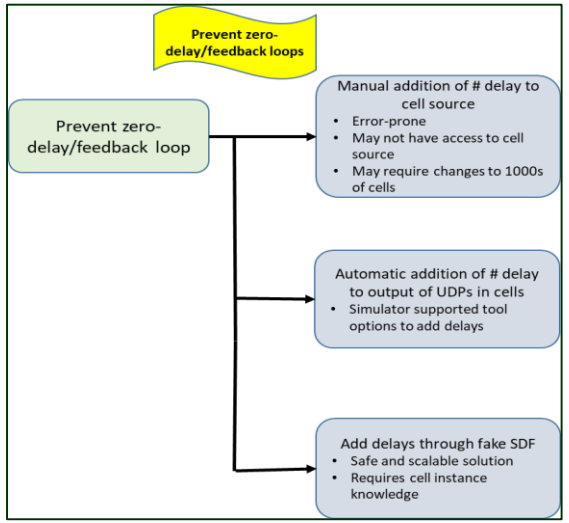


Guidelines to improve GLS performance throughput

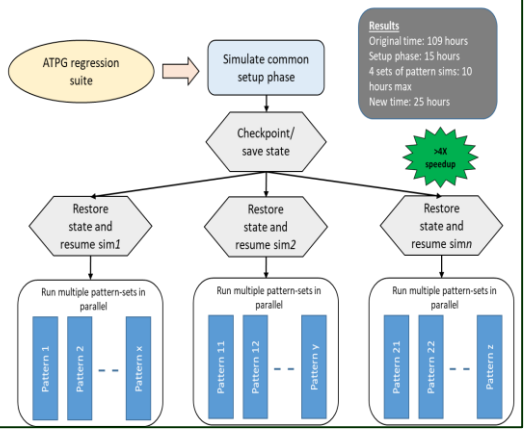
- Do not simulate timing checks till complete, clean SDF is available
- Do not run simulation at finer time resolution if it can be run at coarser resolution
- Do not log cell internals for debugging.
- Replace verified GLS blocks with RTL stubs
- Switch or replace timing corners during simulation

5% to multiple X faster





Test-pattern based partitioning scheme for ATPG and BIST simulations



Performance impact of badly modeled cells

Do not read delayed output port internally in cell

```
`celldefine
module buf_cell (in1, clk, rst, out1);
 output out1;
 input  in1, clk, rst;
 wire tmp1, tmp2, tmp3;

  buf inst1 (tmp1 , in1);
  buf inst2 (tmp2 , tmp1);
  buf inst3 (tmp3 , out1);
  buf inst4 (out1 , tmp2);
  buf inst5 (tmp4 , out1);

 specify
  (posedge clk => (out1 +:1'b1)) =
(98.047401, 108.824997);
  (posedge rst => (out1 +:1'b0)) = (3,
4);
 endspecify
endmodule
```

```
`celldefine
module buf_cell (in1, clk, rst, out1);
 output out1;
 input  in1, clk, rst;
 wire tmp1, tmp2, tmp3;

  wire out1_tmp;
  buf inst1 (tmp1 , in1);
  buf inst2 (tmp2 , tmp1);
  buf inst3 (tmp3 , out1_tmp);
  buf inst4 (out1_tmp , tmp2);
  buf inst5 (tmp4 , out1_tmp);
  buf inst6 (out1 , out1_tmp);

 specify
  (posedge clk => (out1 +:1'b1)) =
(98.047401, 108.824997);
  (posedge rst => (out1 +:1'b0)) = (3,
4);
 endspecify
endmodule
```

Do not annotate delays inside cells as distributed delays

```
`celldefine
module cell1 (a, b, sel, z);
 output z;
 input  a, b, sel;
    buf inst1 ( abuf0, a ) ;
    buf inst2 ( bbuf0, b ) ;
    buf inst3 ( selbuf0, sel ) ;
    prim1 #1 inst0 ( z , abuf0, bbuf0
, selbuf0 ) ;
 endmodule// cell1
`endcelldefine
```

```
`celldefine
module cell1 (a, b, sel, z);
 output z;
 input  a, b, sel;
    buf inst1 ( abuf0, a ) ;
    buf inst2 ( bbuf0, b ) ;
    buf inst3 ( selbuf0, sel ) ;
    prim1 inst0 ( z , abuf0, bbuf0,
selbuf0 ) ;
 specify
    (a=>z) = (1,1);
    (b=>z) = (1,1);
    (sel=>z) = (1,1);
 endspecify
 endmodule// cell1
`endcelldefine
```

Do not use non-delayed reference nets in behavioral model

```
`celldefine
module LAT ( Q, D, G );
 input D, G;
 output Q;
 reg NOTIFIER;
 wire n0;
 wire dD, dG;
    not I0 (iclk, G);
    buf I1 (flgclk, G);
    udp_tlat I2 (n0, D, iclk, xRN, xSN, NOTIFIER);
    buf I3 (Q, n0);
 specify
    $setuphold(negedge G, posedge D, 0, 0, NOTIFIER, , ,dG,dD);
    $setuphold(negedge G, negedge D, 0, 0, NOTIFIER, , ,dG,dD);
    $width(posedge G,0,0,NOTIFIER);
    (D *> Q) = (0,0);
    (posedge G *> (Q : 1'bx)) = (1,1);
 endspecify
endmodule //LAT
`endcelldefine
```

```
`celldefine
module LAT ( Q, D, G );
 input D, G;
 output Q;
 reg NOTIFIER;
 wire n0;
 wire dD, dG;
   `ifdef NTC
      buf inst1 (Gbuf, dG);
      buf inst2 (Dbuf, dD);
   `else
      buf inst1 (Gbuf, G);
      buf inst2 (Dbuf, D);
   `endif
    not I0 (iclk, Gbuf);
    buf I1 (flgclk, Gbuf);
    udp_tlat I2 (n0, Dbuf, iclk, xRN, xSN, NOTIFIER);
    buf I3 (Q, n0);
 specify
    $setuphold(negedge G, posedge D, 0, 0, NOTIFIER, , ,dG,dD);
    $setuphold(negedge G, negedge D, 0, 0, NOTIFIER, , ,dG,dD);
    $width(posedge G,0,0,NOTIFIER);
    (D *> Q) = (0,0);
    (posedge G *> (Q : 1'bx)) = (1,1);
 endspecify
endmodule //LAT
`endcelldefine
```


Questions?

Rohit Jain
(rohit_jain@mentor.com)

Shobana Sudhakar
(shobana_sudhakar@mentor.com)