STA in multi-GHz custom circuits made easy with NanoTime

Udi Nir

Toga Networks

June 1, 2016

SNUG Israel



TOGA NETWORKS

snug
Synopsys Users Group
Israel

Agenda

Introduction and motivation

Suggested STA flow for custom digital blocks using NanoTime

Serializer case study - building circuits and timing arcs

NanoTime customized setup steps

Flow debug and results review

Conclusions

Background - Synopsys STA tools

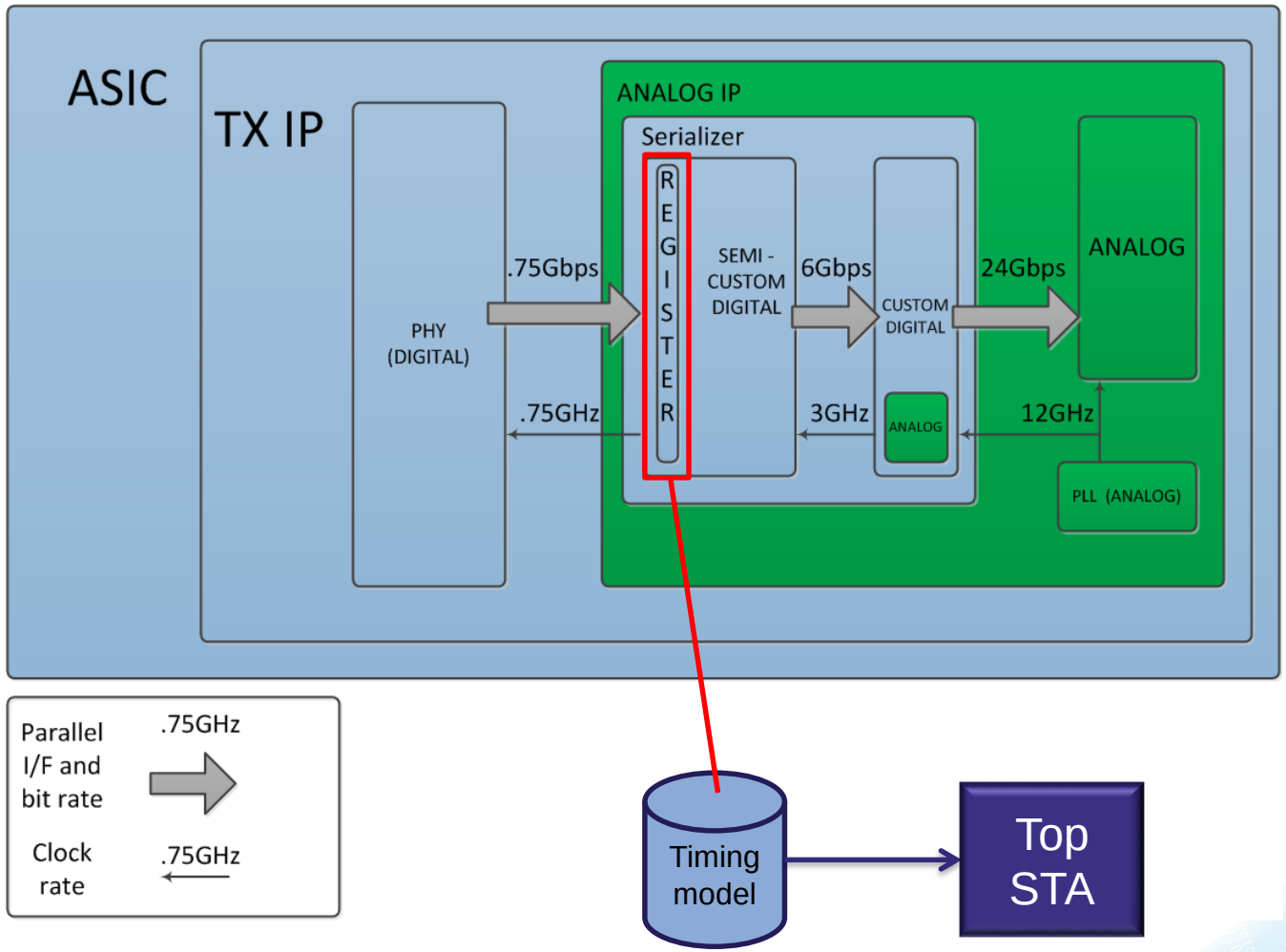
- PrimeTime:
  - Std. cells static timing analysis (STA).
  - Timing arcs between cell pins , called nodes.
  - Internal delays read from .lib database.
- NanoTime:
  - Transistor level STA.
  - Timing arcs nodes are transistor pins.
  - Internal delays based on spice level simulations

STA flow in digital top mixed signal IC

TOGA NETWORKS

snug Synopsys Users Group Israel

- IP/ ASIC Top STA verification is accomplished with PrimeTime and a timing model of the analog IP.
- For this purpose, if the IP is a simple data in **or** data out, interface registers modeling is enough
- Analog IP internal timing paths are verified in dynamic spice simulations .



Motivation

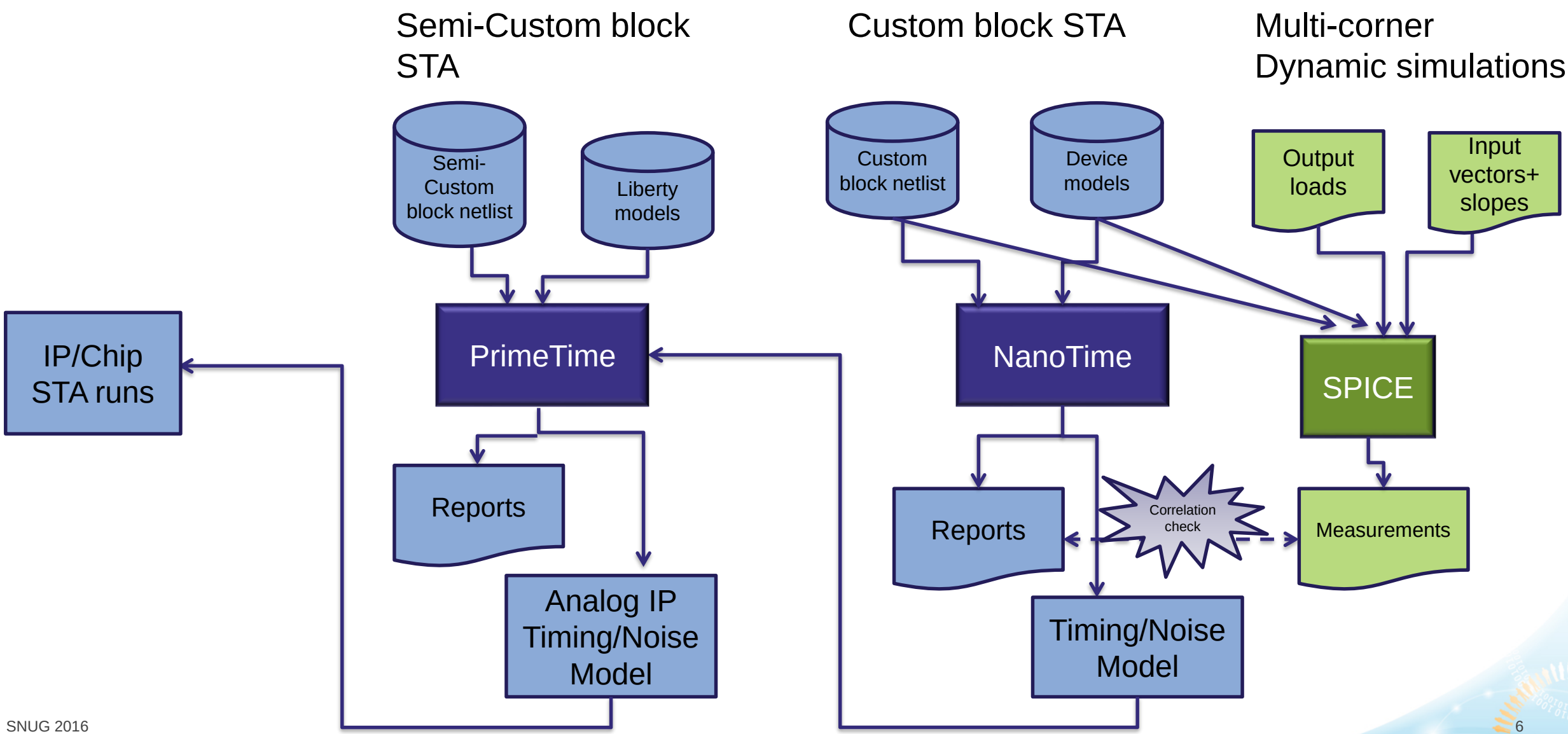
TOGA NETWORKS

snug Synopsys Users Group Israel

- The traditional flow relies heavily on circuit team expertise:
  - Critical path cross section modeling
  - Vector-dependent dynamic simulations
  - Results validation
- Furthermore, the verification quality can be insufficient
  - OCV requires hundreds of slow statistical runs
  - SI impact – coupling capacitors delay and noise – goes back to specific input vector requirements

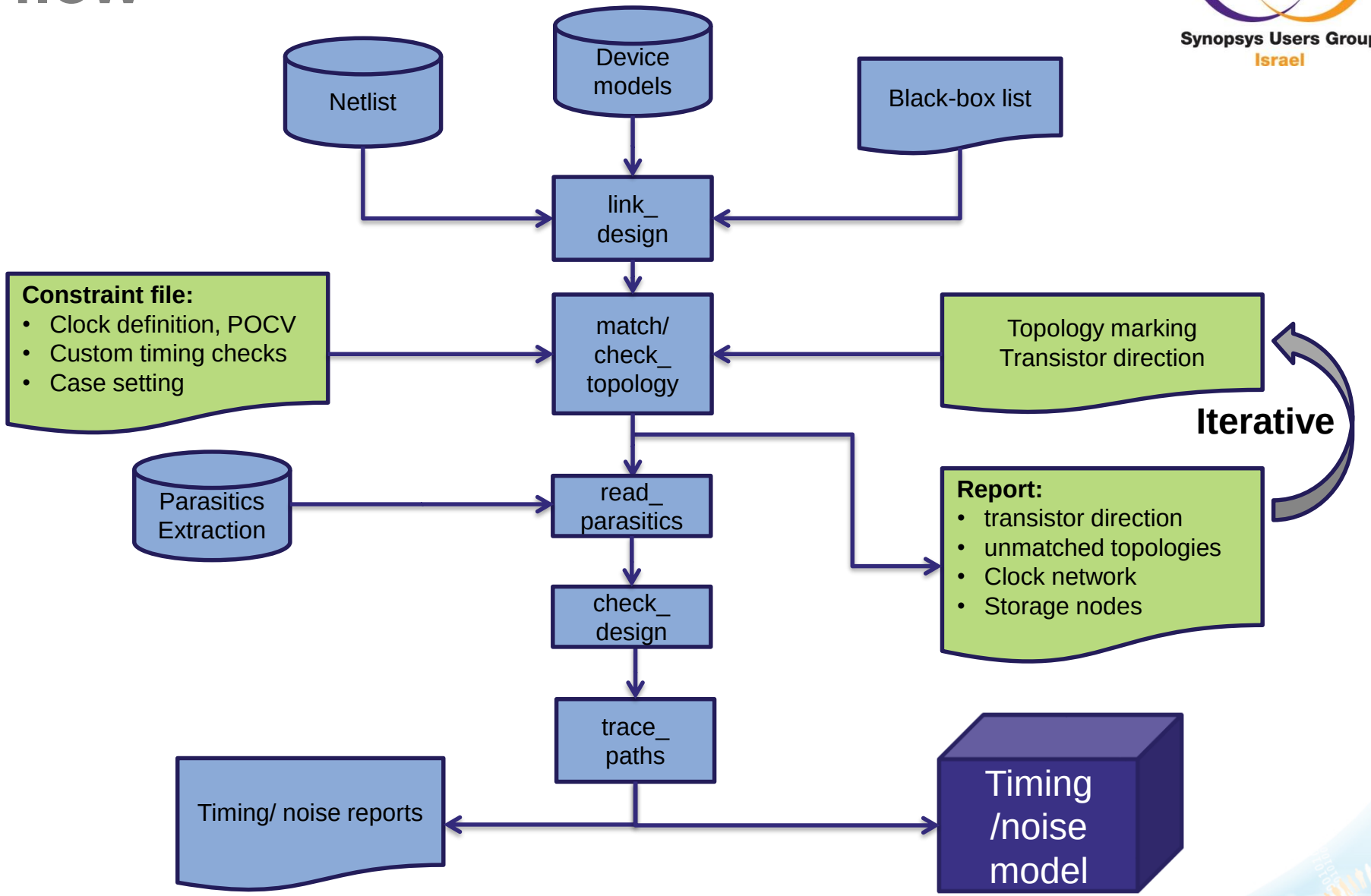
→ For accurate timing verification, a new flow is needed – which can exhaustively report all IP internal timing paths

Suggested STA flow for custom IP



NanoTime STA flow

- Topology marking and custom checks require deep understanding of the design, and consume most of the time in the flow for small size blocks



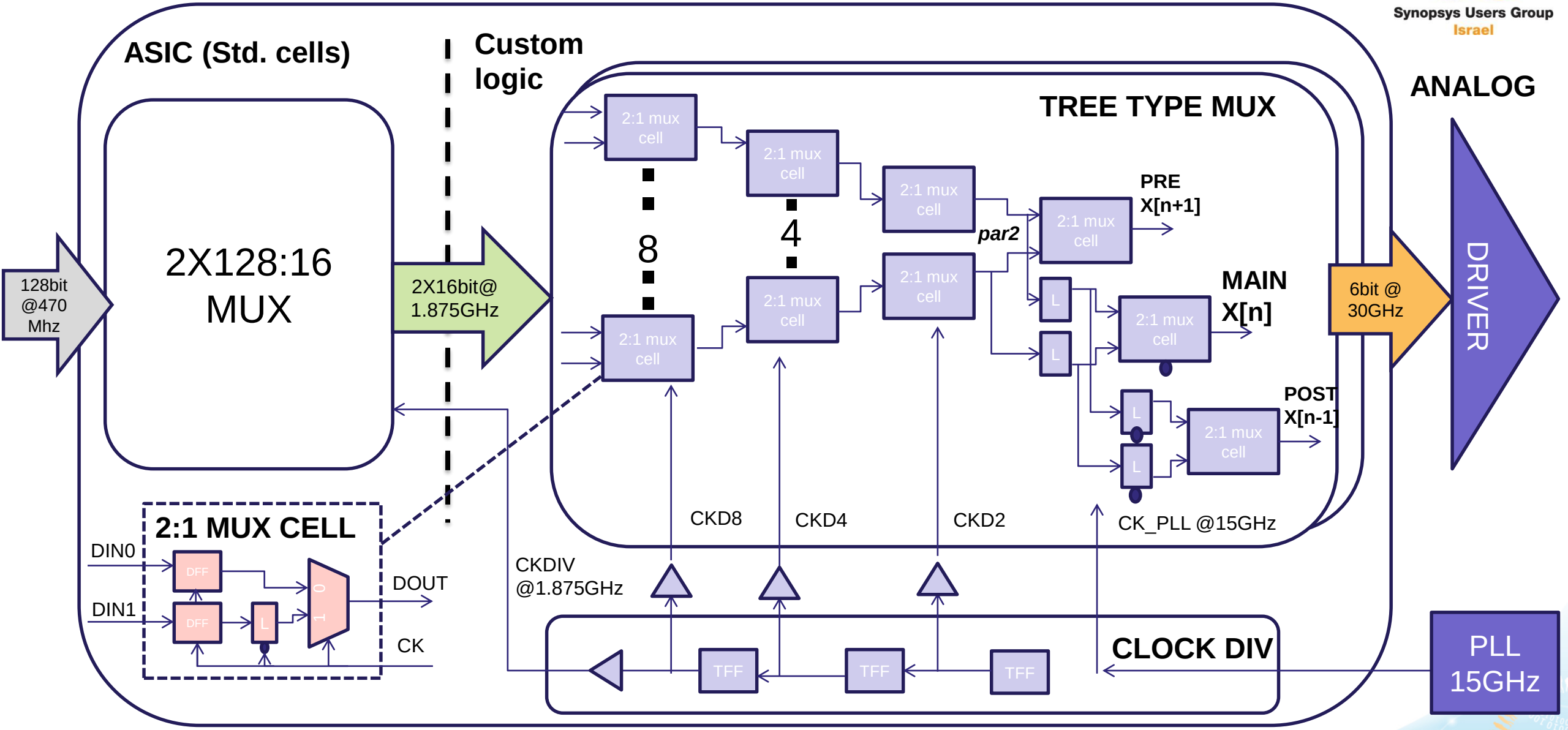
TOGA NETWORKS

snug
Synopsys Users Group
Israel

Case study – 30Gbps Serializer

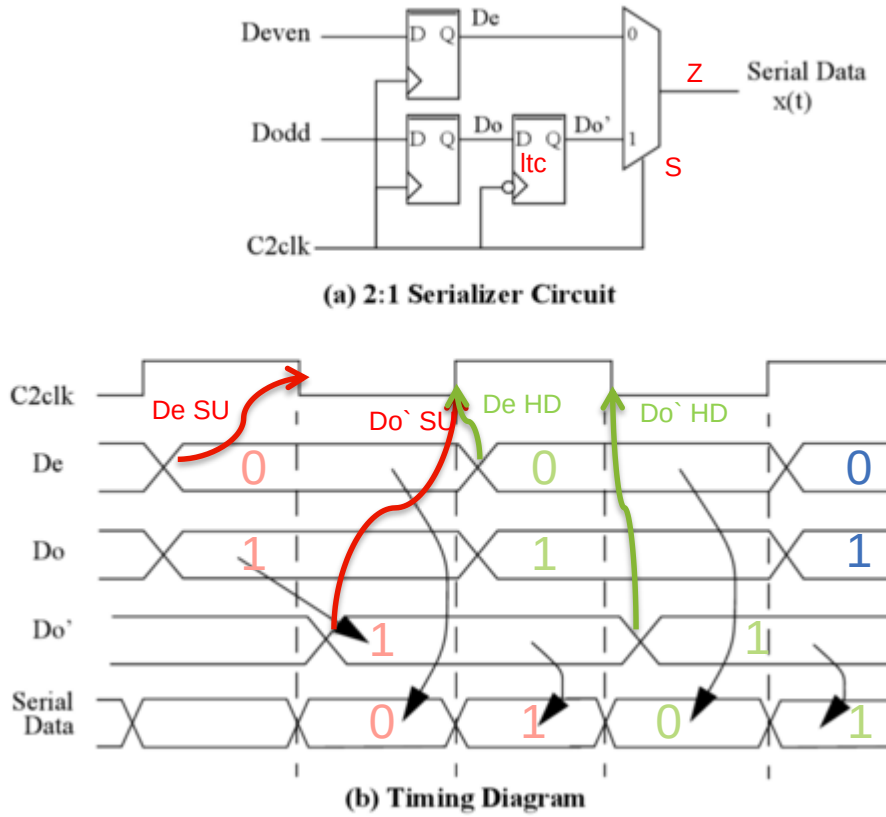


PAM4 30Gbps serializer with FFE (used in 60Gbps SERDES)



2:1 MUX cell

- The basic cell is composed of two flops to sample the data, a latch on ***Do*** and a mux. The latch is added to prevent noise on mux output and for increased hold margin .
- For low deterministic jitter on driver input (6 bits), setup and hold of ***De/Do*** should be verified against ***S*** , to make sure serializer output bits are clock driven and not data driven
- Similar checks are performed on previous stages to improve design robustness and reduce noise



High Speed Serdes Devices and Applications David Robert Stauffer et al., p132

Block statistics

Parameter	Value
Sequential cells (FF/ LTC)	200
Logic cells	1000
Timing endpoints (FF/ LTC/ MUX)	250
Input Clock rate*	15GHz/ 66p
Uncertainty	5p

* 33ps half cycle timing paths exist in the final 2to1 stage

Serializer configuration steps in NanoTime

TOGA NETWORKS

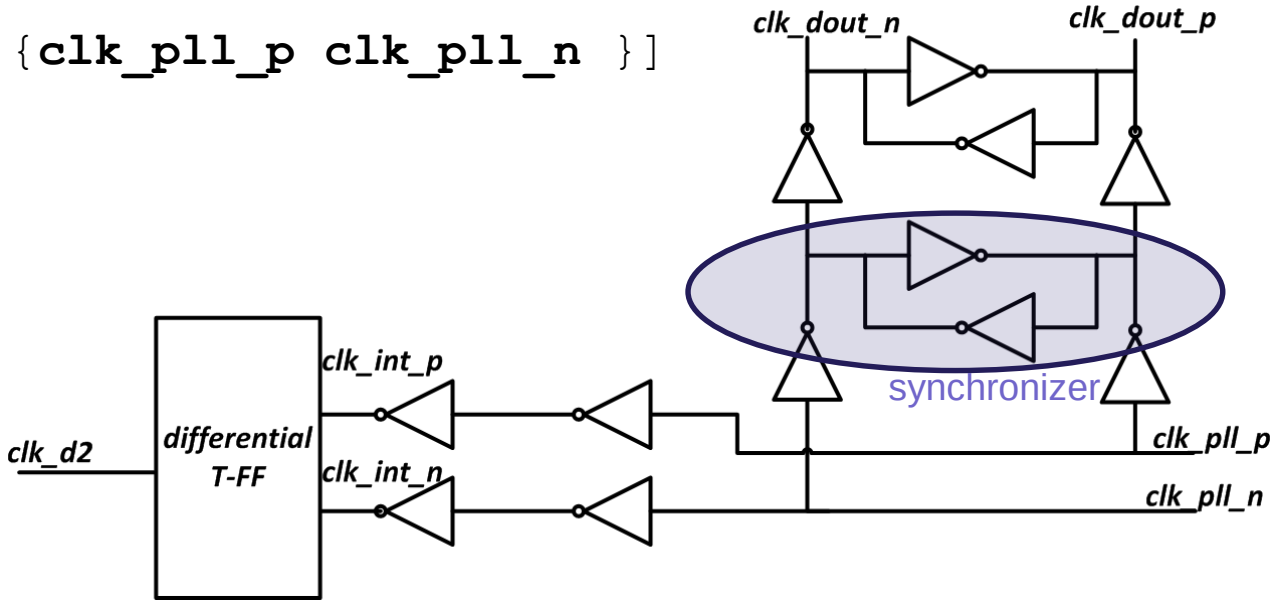
snug Synopsys Users Group Israel

- Create differential clocks
- Mark custom latches and flops
- Create generated clocks (divided clocks)
- POCV settings
- Custom timing checks

Differential clocking

- NanoTime supports differential clocking, and differential synchronizers

```
> create_clock -name clk -waveform {0 0.033} -period 0.066
[get_ports clk_pll_p]
> set_differential [get_ports {clk_pll_p clk_pll_n }]
```



- differential synchronizers are marked automatically. Following command is used for custom/ un-recognized synchronizers:

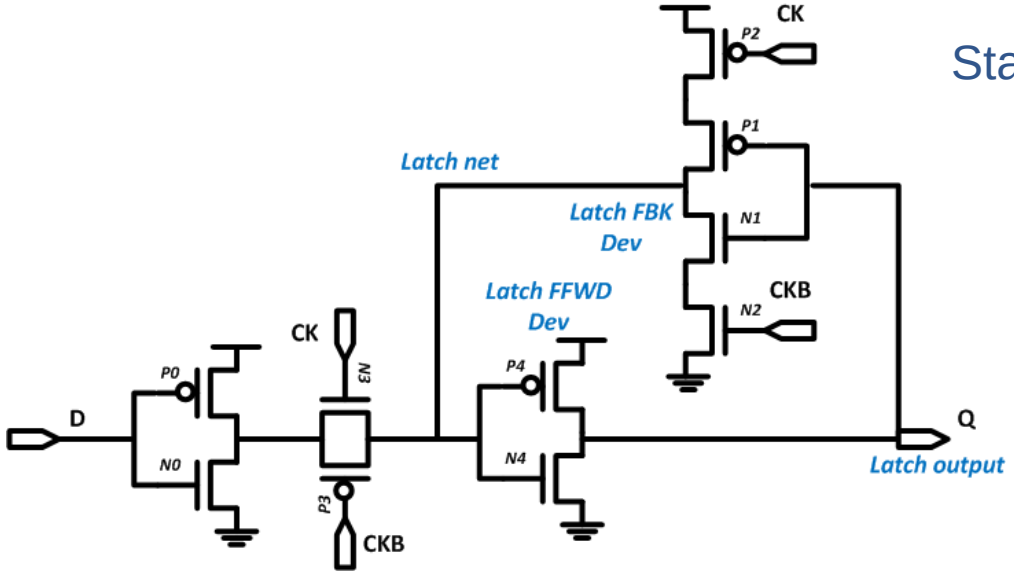
```
>  mark_differential_synchronizer -name synchronizer1 -transistors {…}
```

Custom latch/flop marking

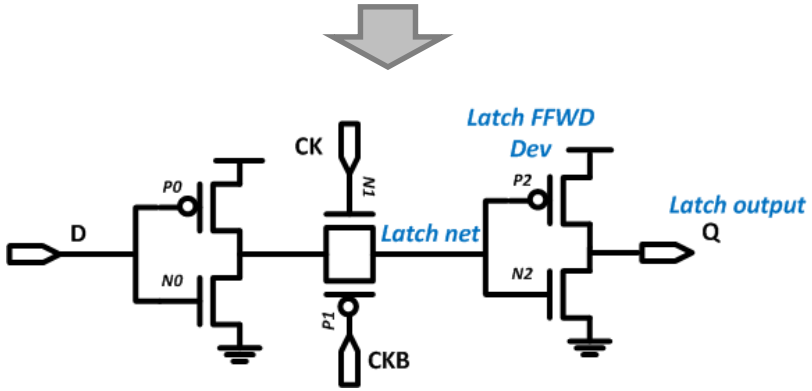
- Dynamic latches are used in high speed digital blocks to save power over CML approach.
- Such latches are not automatically recognized in match_topology stage
- Dynamic latch marking :

```
> mark_latch -latch_net qn -inputs d
    -output qn -clock {xp0/g xn0/g}
    -feedforward {n1 p1}
    -name DynLatch -setup_to input
```

- `foreach_in_collection` can be used to mark all latches iteratively



Static Latch



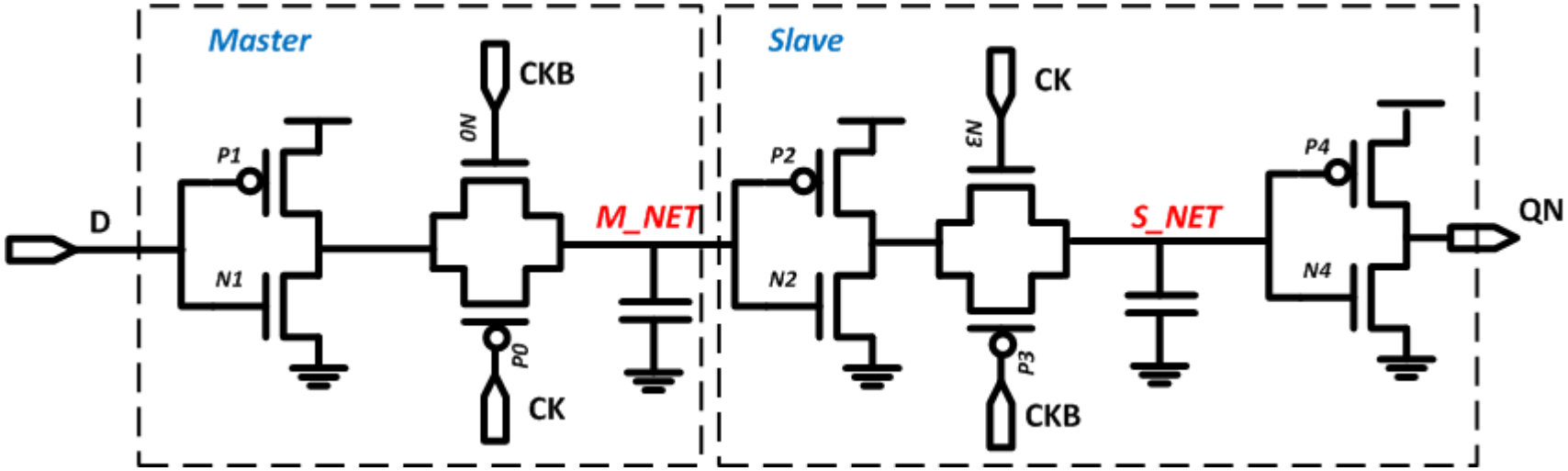
Dynamic Tgate Latch [2]

Custom latch/flop marking (cntd.)

- Incomplete marking – feedback devices are omitted - creates warning
- 2 cascaded latches can be marked as flip-flop:

> `mark_flip_flop -master_latch` **`m_net`** `-slave_latch` **`s_net`**

(Master latch FF devices omitted)

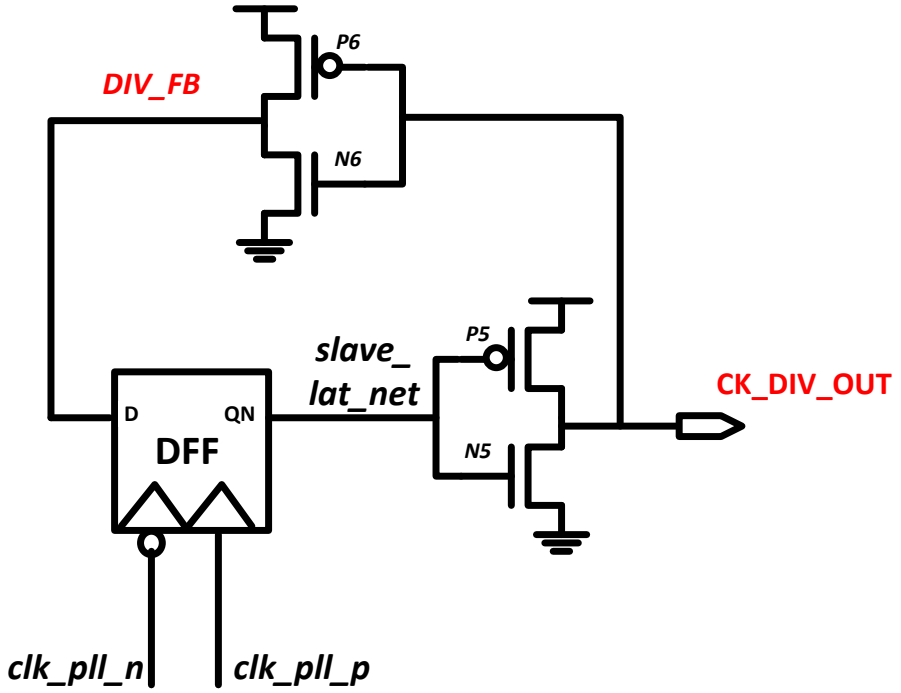


Generated clocks

```
> create_generated_clock -name clk_d2 -source [get_ports clk_pll_p]
-edges {1 3 5} [get_pins x0/xp5/g x0/xn5/g]
```

- For correct clock propagation - define the clock on the gates of both pmos and nmos of an inverter driven by slave latch net
- prevent clock to data propagation warning:

```
> mark_clock_network -stop div_fb
```



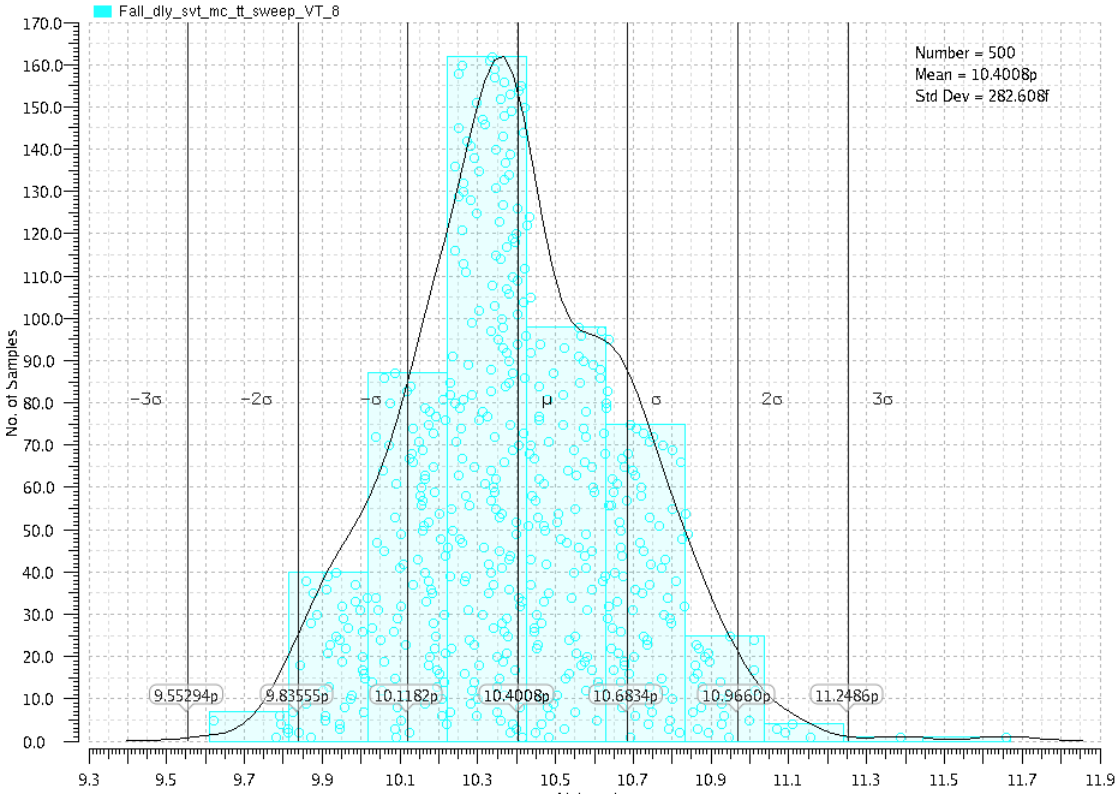
POCV settings

- Parametric OCV can be defined

per device type/ channel length and width / fin_num/ voltage level and check (min/max) .

```
> set_variation_parameters -type default
  -variation 0.03
```

- Variation value is the percentage of 1 sigma out of the device delay

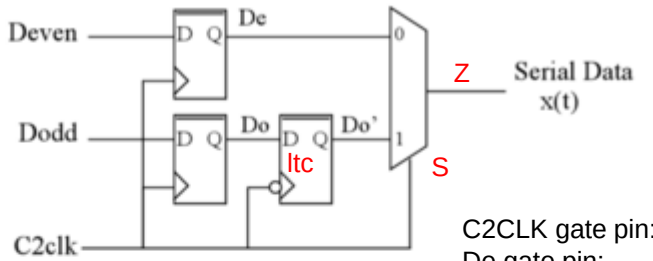


Custom timing checks

- As mentioned before , setup and hold should be checked on MUX inputs De, Do’
- The mux is not recognized automatically as legitimate endpoint
- Mux is constraint with custom timing checks – applied on the gate pins of the constraint nets. For mux phase 0:

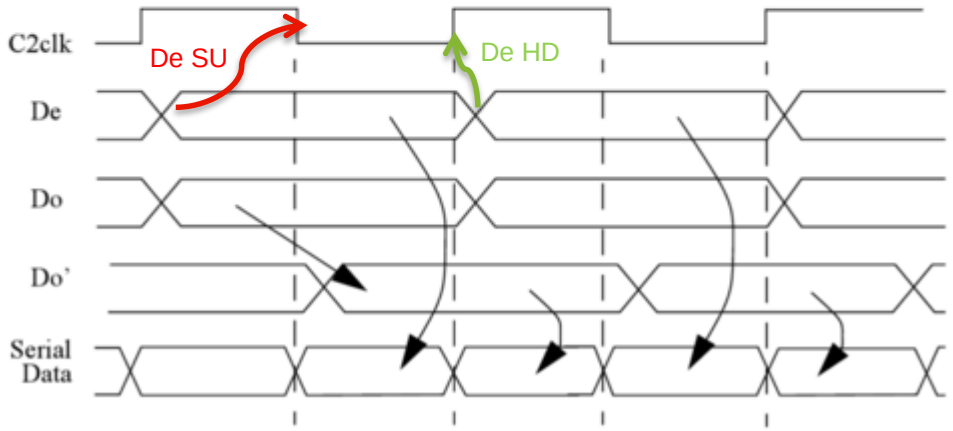
```
> create_timing_check -from
2to1/mux/m0/g  -fall_to
2to1/mux/m2/g -setup 0.010
> create_timing_check -from
2to1/mux/m1/g  -rise_to
2to1/mux/m2/g -hold 0.010
```

- Setup and hold are calculated by dynamic simulation.



(a) 2:1 Serializer Circuit

C2CLK gate pin: ***2to1/mux/m2/g***
De gate pin: ***2to1/mux/m0/g***
Do’ gate pin: ***2to1/mux/m1/g***



(b) Timing Diagram

Increasing NanoTime accuracy

- Increased accuracy is required for correct timing of differential circuits:

`> set_accuracy_effort -level` **`high/ultra`** **(**NanoTime tcl script supplied by Synopsys.)

Among others, this script enables:

- Nonlinear waveform analysis
- Miller effect capacitance
- Reduce SI differential pessimism
- Dynamic Clock Simulation(DCS) – entire differential simulation unit is contained within DCS region

- Enable differential PBSA analysis to remove differential common point pessimism

```
> set pbsa_allow_reconvergent_common_net true
> trace_paths –pocv –pbsa
```

Flow debug and results review

TOGA NETWORKS

snug
Synopsys Users Group
Israel

- Topology recognition
- Clock network
- Parasitics annotation
- Debugging incorrect path timing
- Correlating path timing with spice simulation

Review topology reports

- Check sequential, clock gate and MUX/ tgate topology counts:

```
> report_topology
Report : topology
Design : P2S
…
****************************************
```

Structure Type	Auto Marked	Manually Marked	User Library Marked	Global Library Marked	Total
clock_gate	1	0	0	0	1
feedback	2	0	0	0	2
latch	0	96	0	0	96
mux	32	0	0	0	32
flipflop	0	32	0	0	32

- Verify no transistor direction is left unresolved:

```
> report_transistor_direction
```

Review clock reports

TOGA NETWORKS

snug Synopsys Users Group Israel

> report_clock: Correct period, waveform and attributes

```
Clock              Period           Waveform                Attrs          Sources
-------------------------------------------------------------------------------
clk                 0.066           {0.000 0.033}             pD          {clk_pll_p clk_pll_n}
clk_d2              0.132           {0.000 0.066}             pG          {x0/p5/g x0/xn5/g}
clk_d4              0.264           {0.000 0.132}             pG          {x1/p5/g x1/xn5/g}
```

Propagated Differential

Propagated Generated

> report_clock_arrivals: No missing clock nets, clock domains, clock arrival times

```
Net       Min Rise          Min Fall          Max Rise          Max Fall          Period Attrs Clock
-------------------------------------------------------------------------------------------
clk_dout_n  0.006           0.038             0.009             0.042             0.066   D    clk
clk_dout_p  0.038           0.006             0.042             0.009             0.066   D    clk
clk_int_n   0.006           0.038             0.009             0.042             0.066   D    clk
clk_int_p   0.038           0.006             0.042             0.009             0.066   D    clk
clk_d2      0.070           0.003             0.072             0.011             0.132        clk_d2
clk_d4      0.137           0.005             0.140             0.010             0.264        clk_d4
```

> report_clock_network: No invalid sequential points

Parasitics annotation

- Read parasitics after check_topology step:

```
> read_parasitics -format DSPF top.dspf
```

- For partial extraction of lower hierarchy:

```
> read_parasitics -increment -complete_with zero -path
Xhier1 -format DSPF hier.dspf
```

Errors will be issued on sub-hierarchy input and output nets in the annotation report. Nevertheless, annotated parasitics are accounted for correctly.

- Support layout only devices with <*layout_only*> prefix:

```
> set parasitics_xref_layout_instance_prefix layout_only
```

report_path - debug example

TOGA NETWORKS

snug Synopsys Users Group Israel

```
> report_path -type min -path_type full_clock_expanded -transition
```

Generated Clock has huge transition difference between data and clock path.

```
Startpoint:          xck_d2_div_out/xmm1_1/g
Endpoint:            xmux_4to2<7>/xhmuxb2/xm2q/d
Path Type:           min
Constraint:          user specified hold check
PBSA Common Net:     clk_p
PBSA Complement:     clk_n

                                                 Coeff     Xtalk
    Path      Incr     Adjust       Var          adj      Delta      Trans       Cap  NT     Point
--------  --------  --------  --------  --------  --------  --------  --------  ----  -- --------------------------------
  0.0000              0.0000                                          0.0400    0.0120  C     f clk_p (inout)
  0.0000    0.0000                0.0000    0.0000    0.0000    0.0400    0.0120  C     f xck_p_int0<0>/xmm0_1/g (INVD2)
  0.0077    0.0077                0.0000    0.0000    0.0000    0.0148    0.0038  C     r xck_p_int2<0>/xmm1_1/g (INVD8)
  0.0117    0.0040                0.0000    0.0000    0.0000    0.0053    0.0077  C     f xdiv_14to7_dff/xdsltc/xm1/g (p2s_cpl_ltcn_uu_4)
  0.0155    0.0038                0.0000    0.0000    0.0000    0.0074    0.0038  GC    r xck_d2_div_out/xmm1_1/g (INVD6)
  0.0188    0.0033                0.0004    0.0000    0.0000    0.0039    0.0071  C     f xinv_ck_d2_b/xmm0_1/g (INVD12)
  0.0234    0.0046                0.0005    0.0000    0.0000    0.0066    0.0339  C     r xmux_4to2<7>/xltcn1_1/xm3/g (p2s_cpl_ltcn_uu_1)
  0.0266    0.0032                0.0004    0.0000    0.0000    0.0044    0.0007  L     f xmux_4to2<7>/xbit1_buff/xmm0/g (INVD1)
  0.0296    0.0030                0.0004    0.0000    0.0000    0.0060    0.0017        r xmux_4to2<7>/xhmuxb2/xm2q/g (p2s_cpl_ltcn_uu_2)
  0.0334    0.0038                0.0005    0.0000    0.0000    0.0058    0.0012  U     f xmux_4to2<7>/xhmuxb2/xm2q/d (p2s_cpl_ltcn_uu_2)
  0.0319              -0.0014                                                             POCV adjustment
  0.0319                                                                                  data arrival time
            0.0334    0.0141    0.0010              0.0000                              Total

  0.0000              0.0000                                          0.0400    0.0120  C     r clk_n (inout)
  0.0000    0.0000                0.0000    0.0000    0.0000    0.0400    0.0120  C     r xck_p_int0<1>/xmm1_1/g (INVD2)
  0.0085    0.0085                0.0000    0.0000    0.0000    0.0148    0.0038  C     f xck_p_int2<1>/xmm0_1/g (INVD8)
  0.0118    0.0033                0.0000    0.0000    0.0000    0.0055    0.0076  C     r xdiv_14to7_dff/xdsltc/xm3/g (p2s_cpl_ltcn_uu_4)
  0.0160    0.0041                0.0000    0.0000    0.0000    0.1871    0.0038  GC    r xck_d2_div_out/xmm1_1/g (INVD6)
  0.0285    0.0126                0.0008    0.0000    0.0000    0.0374    0.0071  C     f xinv_ck_d2_b/xmm0_1/g (INVD12)
  0.0381    0.0095                0.0006    0.0000    0.0000    0.0165    0.0339  C     r xinv_ck_d2/xmm1_1/g (INVD12)
  0.0447    0.0067                0.0004    0.0000    0.0000    0.0106    0.0287  C     f xmux_4to2<7>/xhmuxb2/xm3/g (p2s_cpl_ltcn_uu_2)
            0.0447    0.0160    0.0010              0.0000                              Total
  0.0497    0.0050                                                                      hold time
  0.0497    0.0000                                                                      clock uncertainty
  0.0497                                                                                data required time
-------------------------------------------------------------------------------------------------------------------------
  0.0319                                                                                data arrival time
 -0.0497                                                                                data required time
-------------------------------------------------------------------------------------------------------------------------
 -0.0178                                                                                slack (VIOLATED)
```

report_path - debug example – cntd.

Debug steps

`> write_spice:` we need to compare the section that creates the generated clock – use reference_timing_path and prefix_path_id attributes:

```
> set data_path [get_timing_path …]
> set clock_path [get_attribute $data_path reference_timing_path]
> set data_prefix_path [get_attribute $data_path prefix_path_id]
> set clock_prefix_path [get_attribute $clock_path prefix_path_id]
> write_spice $data_prefix_path -output datapath
> write_spice $clock_prefix_path -output clkpath
```

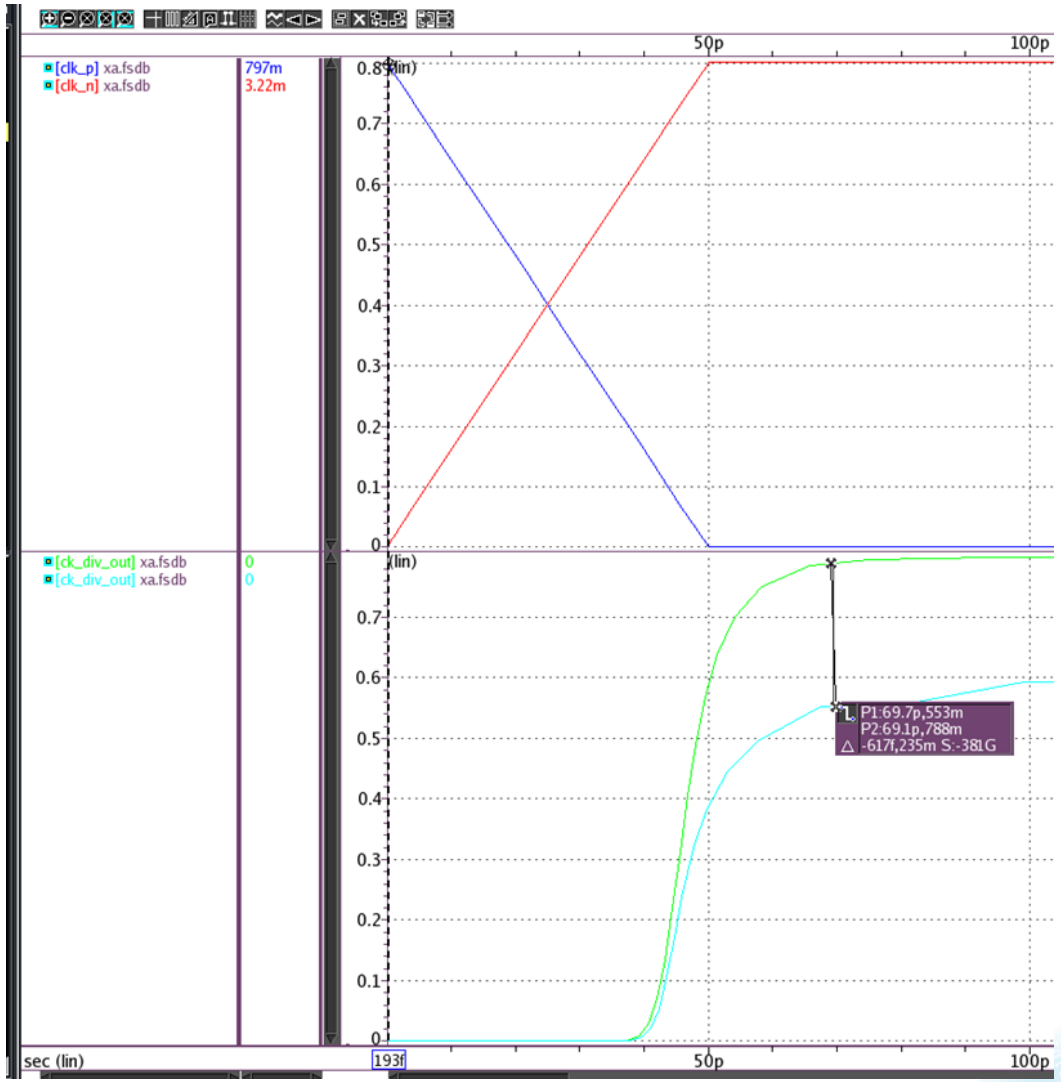
- Execute spice decks with hspice/xa

report_path - debug example – cntd.

Debug steps

- Inspect waveforms with WaveView – in clkpath GC clock is not reaching the supply rail - see diagram on the right
- **Conclusion**: The clock path had the Clock P type device of the slave latch in off state (The tool assumed this setting will cause WC timing)
- But this latch is driven by differential clock, which are on/off simultaneously
- **Solution**: The following flag needs to be set to true (default value):

```
> set topo_tgate_mark_all_pairs true
```



Correlating path timing with spice simulation

TOGA NETWORKS

snug Synopsys Users Group Israel

NT path report, compared measurements are marked

The selected path is the custom check created on final mux inputs.

```
Startpoint:          clk_p (inout port)
Endpoint:            xp2s_mux_out<0>/x0/xhmux_p0/xm0/d
Path Type:           max
Constraint:          user specified setup check
PBSA Common Net:     ck_p_buf<6>

                                                 Xtalk
    Path      Incr     Adjust       Var        Delta      Trans       Cap    NT    Point                                  Net
--------  --------  --------  --------  --------  --------  --------  ----  -- ------------------------------ ------------------------------
  0.0000              0.0000                                0.0400    0.0201  C&   f clk_p (inout)                clk_p
  0.0009    0.0009                0.0000    0.0000    0.0400    0.0201  C&   f xck_tree/xck_inv2<0>/xmm0_1/g (invd12)
                                                                                  clk_p
  0.0115    0.0106                0.0000    0.0000    0.0173    0.0324  C&   r xck_tree/xckinv3<6>/xmm1_1/g (invd6)
                                                                                  xck_tree/ck_p_i2<0>
  0.0220    0.0106                0.0000    0.0000    0.0141    0.0170  C&   f xp2s_mux_out<0>/x0/xltcn0/xm1/g (p2s_cpl_ltcn_uu_4)
                                                                                  ck_p_buf<6>
  0.0267    0.0046                0.0002    0.0002    0.0123    0.0045  L&   r xp2s_mux_out<0>/x0/xl3inv/xmm1_1/g (invd2bwp16p90cpdulvt)
                                                                                  xp2s_mux_out<0>/x0/dyn_bit0_n
  0.0319    0.0053                0.0002    0.0004    0.0084    0.0025  &    f xp2s_mux_out<0>/x0/xhmux_p0/xm0/g (p2s_cpl_ltcn_uu_2)
                                                                                  xp2s_mux_out<0>/x0/bit0_p
  0.0360    0.0041                0.0002    0.0000    0.0061    0.0015  U&   r xp2s_mux_out<0>/x0/xhmux_p0/xm0/d (p2s_cpl_ltcn_uu_2)
                                                                                  xp2s_mux_out<0>/x0/xhmux_p0/i_b
  0.0364                0.0004                                                    POCV adjustment
  0.0364                                                                          data arrival time
            0.0360      0.0004    0.0004    0.0007                              Total

  0.0355                0.0355                                0.0400    0.0201  C&   r clk_p (inout)                clk_p
  0.0359    0.0004                0.0000    0.0000    0.0400    0.0201  C&   r xck_tree/xck_inv2<0>/xmm1_1/g (invd12)
                                                                                  clk_p
  0.0436    0.0077                0.0000    0.0000    0.0173    0.0324  C&   f xck_tree/xckinv3<6>/xmm0_1/g (invd6)
                                                                                  xck_tree/ck_p_i2<0>
  0.0531    0.0095                0.0000    0.0000    0.0136    0.0170  C&   r xp2s_mux_out<0>/x0/xhmux_p0/xm3/g (p2s_cpl_ltcn_uu_2)
                                                                                  ck_p_buf<6>
  0.0569                0.0003                                                    PBSA adjustment
            0.0176      0.0358    0.0000    0.0000                              Total
  0.0469   -0.0100                                                               setup time
  0.0419   -0.0050                                                               clock uncertainty
  0.0419                                                                          data required time
-------------------------------------------------------------------------------------------------------------------------------------------
  0.0419                                                                          data required time
 -0.0364                                                                          data arrival time
-------------------------------------------------------------------------------------------------------------------------------------------
  0.0055                                                                          slack (MET)
```

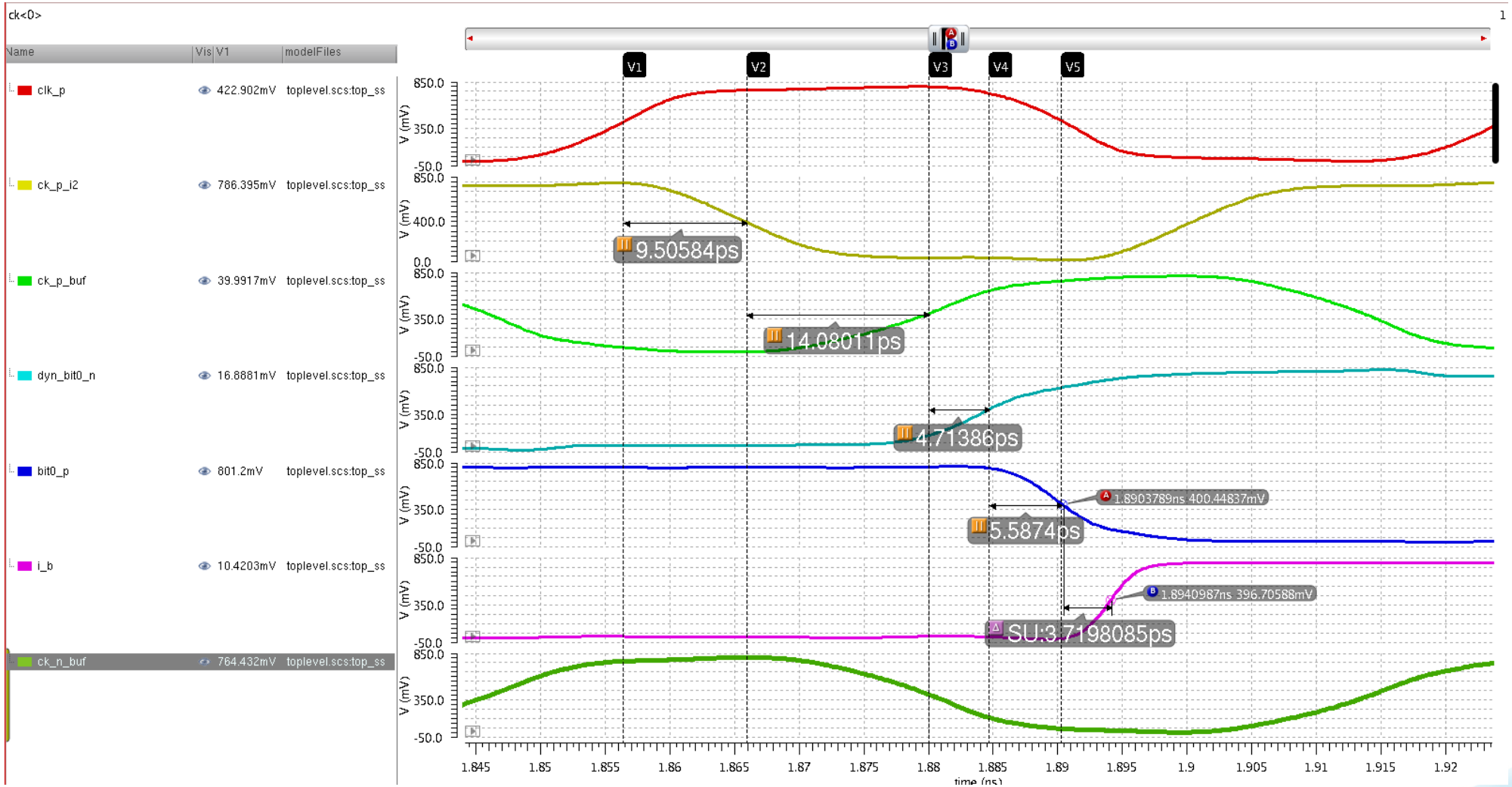
&=Annotated delay

Correlating path timing with spice simulation

TOGA NETWORKS

snug Synopsys Users Group Israel

Path delay in block level dynamic simulation



Correlating path timing with spice simulation

TOGA NETWORKS

snug Synopsys Users Group Israel

Comparison summary

- Entire path falls inside the advertised accuracy of +/- 5%.
- Intermediate nodes have higher differences – result of linear ramp used as the input to the stage by the tool, and prev. stage output by spice

Stage	report_path	write_spice	Other vendor
clk_p→ ck_p_i2	.9+10.6= **11.5p**	11.2p **(-2.5%)**	9.5p **(-17.4%)**
ck_p_i2→ ck_buf	10.6p	10.1p **(-4.7%)**	14.1p **(+33%)**
ck_buf→ dyn_bit0_n	4.6p	5.0p **(+8.7%)**	4.7p **(+2.2%)**
dyn_bit0_n→ bit0_p	5.3p	4.5p **(-15%)**	5.6p **(+5.6%)**
bit0_p→ i_b	4.1p	4.2p **(+2.5%)**	3.7p **(-10%)**
Whole path: clk_p-->-i_b	36p	35p **(-2.8%)**	37.6p **(+4.4 %)**

Correlating path timing with spice simulation (cntd.)

TOGA NETWORKS

snug Synopsys Users Group Israel

Full path comparison

- Entire path falls inside the advertised accuracy of +/- 5%.

Path ID	report_path	write_spice
3138	36.1p	35.9p(-0.5%)
3114	36.4p	36.4p(-0%)
4432	28.4p	27.8p(-2.2%)
4411	28.4p	27.8p(-2.2%)

Run time comparison

TOGA NETWORKS

snug
Synopsys Users Group
Israel

Nanotime run:

```
Information: Defining new variable 'dut'. (CMD-041)
Number of simulations in this session: 14876
Percent of restored simulation units used: 0
Percent of current design aided by restored simulation units: 0
Maximum memory usage for this session: 219.70 MB
Maximum extra memory used by a child process: 0.00 MB
CPU usage for this session: 61 seconds
Diagnostics summary: 9 errors, 6322 warnings, 9408 informationals
```

Single dynamic simulation with moderate accuracy , including parasitics:

```
Aggregate audit (1:34:33 PM, Sun May 1, 2016):
Time used: CPU = 7.47 ks (2h 4m 29s), elapsed = 2.45 ks (40m 45.6s), util. = 305%.
Time spent in licensing: elapsed = 3.3 s.
Peak memory used = 3.31 Gbytes.
```

Conclusions

TOGA NETWORKS

snug Synopsys Users Group Israel

- The proposed NanoTime flow was executed on 30Gbps serializer
- All custom latches, differential clocking scheme and clocked mux topologies were recognized and timed correctly
- Parasitic annotation was applied only partially, since full layout was not available at the time this experiment took place.
- Timing report of a path with parasitic annotation was verified against spice simulations - both NanoTime generated spice deck and original schematic simulation. Overall path delay was found to be within (+/-5%).
- Some of the capabilities of the tools couldn’t be evaluated at this stage , and will be deferred to the point when full extraction is available: SI delay and Noise, extensive timing path correlation, Extracted Timing Model (ETM) generation.

Conclusions (cntd.)

- The effort required for the setup of NanoTime should not be under estimated. However, the env. can be reused later for similar design without extra work.
- It can be made easier if more warnings are issued, for example:
  - Generate Clock failure – there is no warning if the clock is not created correctly, and report_clock_arrivals has missing values for some nodes
  - No warning when net not completing full swing – as shown in the debug example explained before.
- **It is evident that the original goal of increased timing coverage of custom IP can be achieved using NanoTime, with spice level measurements accuracy , and at a run time which is a fraction of a single dynamic simulation.**

TOGA NETWORKS

snug
Synopsys Users Group
Israel

Thank You



References

- **High Speed Serdes Devices and Applications** David Robert Stauffer, Jeanne Trinko-Mechler, Michael A. Sorna, Kent Dramstad, Clarence Rosser Ogilvie, Amanullah Mohammad, James Donald Rockrohr
- **Flip-Flops for Accurate Multiphase Clocking:Transmission Gate Versus Current Mode Logic** Ramen Dutta et al. - IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, 2013
- **Transistor-Level Timing and Noise Analysis of Peripheral Logic of High Speed Memory Design** Johnie Au, Sunil Koduru, Jun Li, Sahil Bargal, Cypress/Synopsys - 2013
- **Static Timing Of Full Swing Differential Circuits Differential Clock and Topology Handling in NanoTime** Norb Heindl , 2014
- **Analysis and Characterization of Memories and Custom Digital Blocks Using NanoTime** Chad Lawrence , 2013
- **Accurate Transistor-Level STA Methodology for 20nm Custom SRAM Macro Using NanoTime** Shahnaz Nagle, Mandeep Singh, Charles Jiang , Samsung/ Synopsys
- **Transistor-level Static Timing Analysis and Characterization of Embedded SRAM Using NanoTime for Memories** Clint Parker , Charles Jiang , HP/ Synopsys

Acknowledgements

- Toga Network Custom Layout – Gil Mittelman
- Synopsys Israel - Uri Golan, Isaac Zafrany, Dan Gilboa
- Synopsys NanoTime Global Support - Norb Heindl