Novel Pulse Width Insensitive Design and Verification Methods

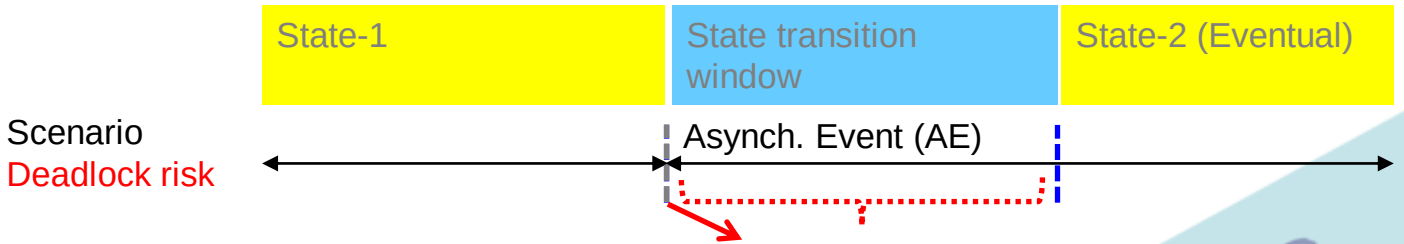
September 27, 2019

Ruchi Shankar, Shalini Eswaran, Sharavathi Bhat, Lakshmanan Balasubramanian

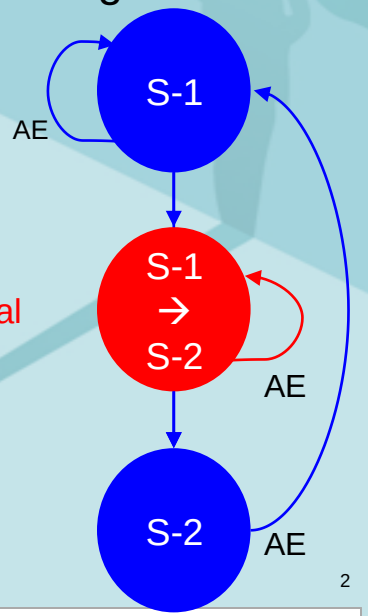
Texas Instruments (India) Pvt. Ltd., Bengaluru, India

Motivation & Problem Statement (1/2)

- Operation of electrical circuits are sensitive to input glitches i.e., pulse widths of input signals
  - Issues with system level function when built with such circuit elements
  - Critical for asynchronous data or control paths and delay chains used for sequencing operations
    - Hazard example: Reaching eventual state is critical once the transition started, else getting stuck in an irrecoverable forbidden state
  - Clock-less implementation with delay elements are sensitive to input pulse widths / glitches



Probabilistic system deadlock depending on event arrival
How to handle the effect of the event crossing this boundary?
OR
Rather how to control the accurate positioning of this boundary?
Process sensitive



Motivation & Problem Statement (2/2)

- Earlier solutions
  - Pulse width filtering ➔ Standard glitch gobbler or filtering using sequential elements, delay and other combinatorial elements
    - Cells used in the delay elements or glitch gobbler circuits are not characterised for pulse width filtering characteristic
    - Costlier SPICE based simulation is mandated at system level to verify the robustness of the design
  - Handshake based asynchronous design methods ➔ Handling external asynchronous signal sources not amenable
- Proposed: Pulse width insensitive design and/or pulse width sensitivity analysis capability
  - Identified & existing implementations are sensitive to input pulse width
  - No need of flip-flop or edge-sensitive circuit elements
  - Utilises inherent glitch filtering behaviour of combinatorial gates
- Limitations
  - No automated gate-level tools/methodology available to design, synthesize or verify such designs
- Proposed: Pulse width sensitivity analysis capability in gate level simulation

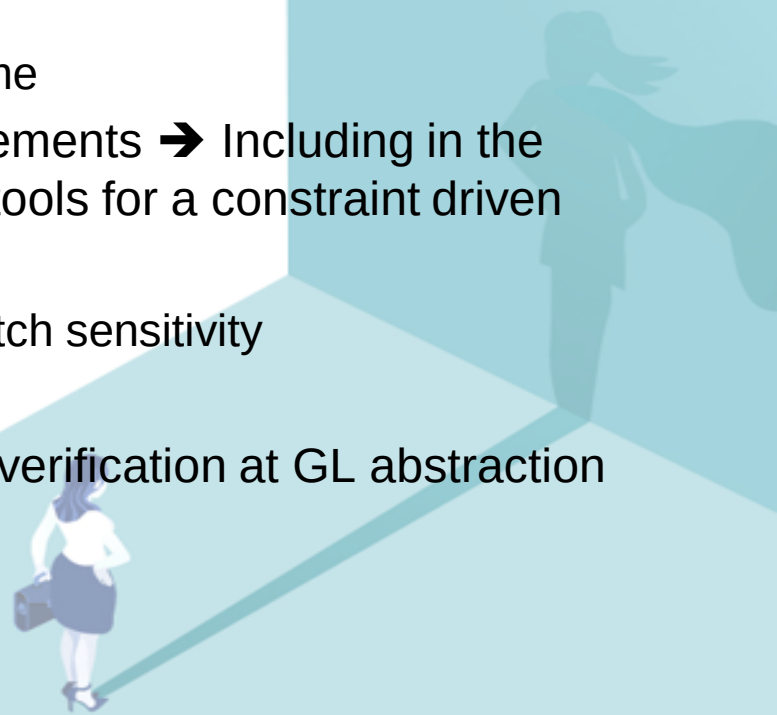
Proposed Solution – High Level Formulation

- Standard active element based delay elements exhibit glitch filtering behaviour
  - This fact is not utilised in existing design practices
- Such behaviour of any existing circuits cannot be analysed easily with existing standard verification techniques
  - May require costly transistor level (SPICE) simulations
- Pulse width insensitive design
  - Ordering of delay elements appropriately to achieve the required glitch filtering behaviour
- Simulation based & static analysis of such designs
  - Simulation methods exist for similar purpose in sequential circuits
  - Combination of such of verification components is extended for combinatorial elements

Proposed Design and Verification Methodology

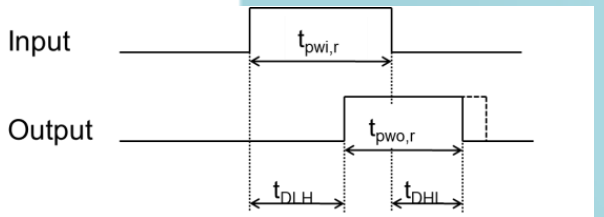
1. Novel alternate system level design technique
2. An area and power efficient circuit design technique for pulse width insensitive design
   - Only using existing delay elements
3. A) A constraints driven design automation methodology for the same
   - Pulse width sensitivity characterisation of combinatorial elements ➔ Including in the timing library ➔ Use of existing logic and physical design tools for a constraint driven automation for design synthesis

3. B) A simulation based verification method that can comprehend glitch sensitivity
   - Extension of functional model for glitch filtering behaviour
   - Use of the above components to drive a simulation based verification at GL abstraction
4. Static timing analysis

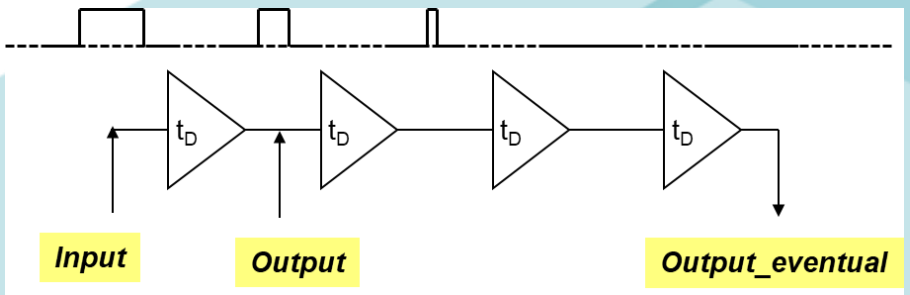


Direction of Design Method

- Delay buffers and all combinatorial logic cells are characterised by
  - Pulse width filtering ➔ Min. pulse width that will pass through (t_{mpw})
  - Pulse width modification (elongation/compression)
    - Pulse width compression is predominant
  - Delay
    - Transport delay ➔ $t_{mpw} = t_D$
      - Default behaviour of standard cell models & simulator
    - Inertial delay ➔ $t_{mpw} \neq t_D$
      - Only modelled for clock, reset and preset controls of flip-flops
      - Combinatorial cells may have negligible effect ➔ Significant for delay cells
- Effect on delay chains
  - Is it possible to build delay chains with any random combination of individual delay cells?
  - Easy to analyse in SPICE / transistor level (TL) simulations
    - Earlier implementations were part of analog modules
    - Currently implemented in semi-custom digital partition

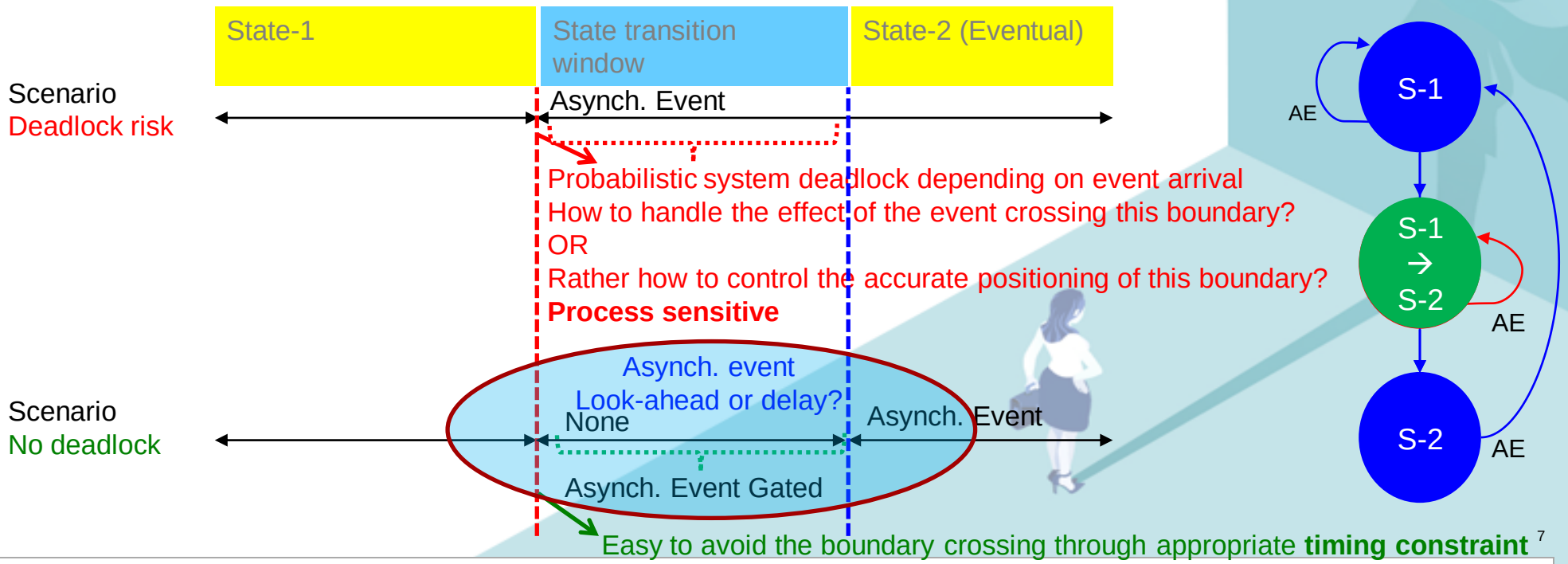


$$t_{pwo,r}=t_{pwi,r}-(t_{DLH}-t_{DHL})$$
$$t_{pwo,f}=t_{pwi,f}+(t_{DLH}-t_{DHL})$$



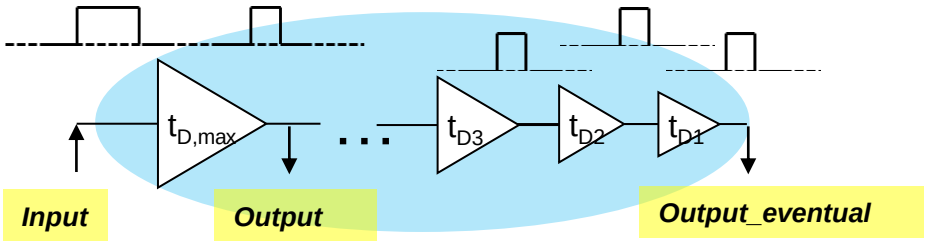
Methodology - 1: System Design

- **Asynchronous event look-ahead** for constraint driven design without architectural design cost
  - Look-ahead information used to gate the asynchronous event for a short period during transition ➔ System robustness at the cost of probabilistic loss of event in a small time window during state transition



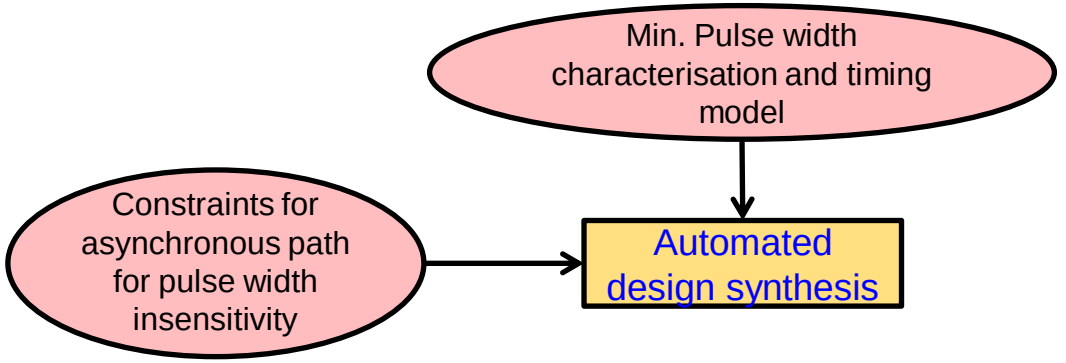
Methodology - 2: Circuit Design

- Analysis of delay buffer transistor level (TL) simulations: $t_{mpw} < t_D$ & t_{mpw} α t_D
- Synthesise the delay chain such that the first delay element provides the safest pulse width filtering ($t_{D,max}=t_n$)>t_{n-1}



- Conventional glitch gobbling function, but through natural property of the delay cell
- First cell: Delay cell with the largest delay ($t_n=t_{D,max}$) and largest pulse width propagation ($t_{mpw,max}$)
- Composition of the subsequent portion of chain is immaterial as long as all the cells $t_D < t_{D,max}$ and $t_{mpw}<t_{mpw,max}$
- It will not encounter any input with min. pulse width ($t_{mpw,i} > t_{opw,max}$)
- It should only use cells with delay < $t_{D,max}$ & $t_{mpw}<t_{mpw,max}$
- No need for a flip-flop or latch element → Area and power efficient
- Critical where any event on **Output** eventually should result in corresponding event in **Output_eventual**

Methodology – 3: Design Automation

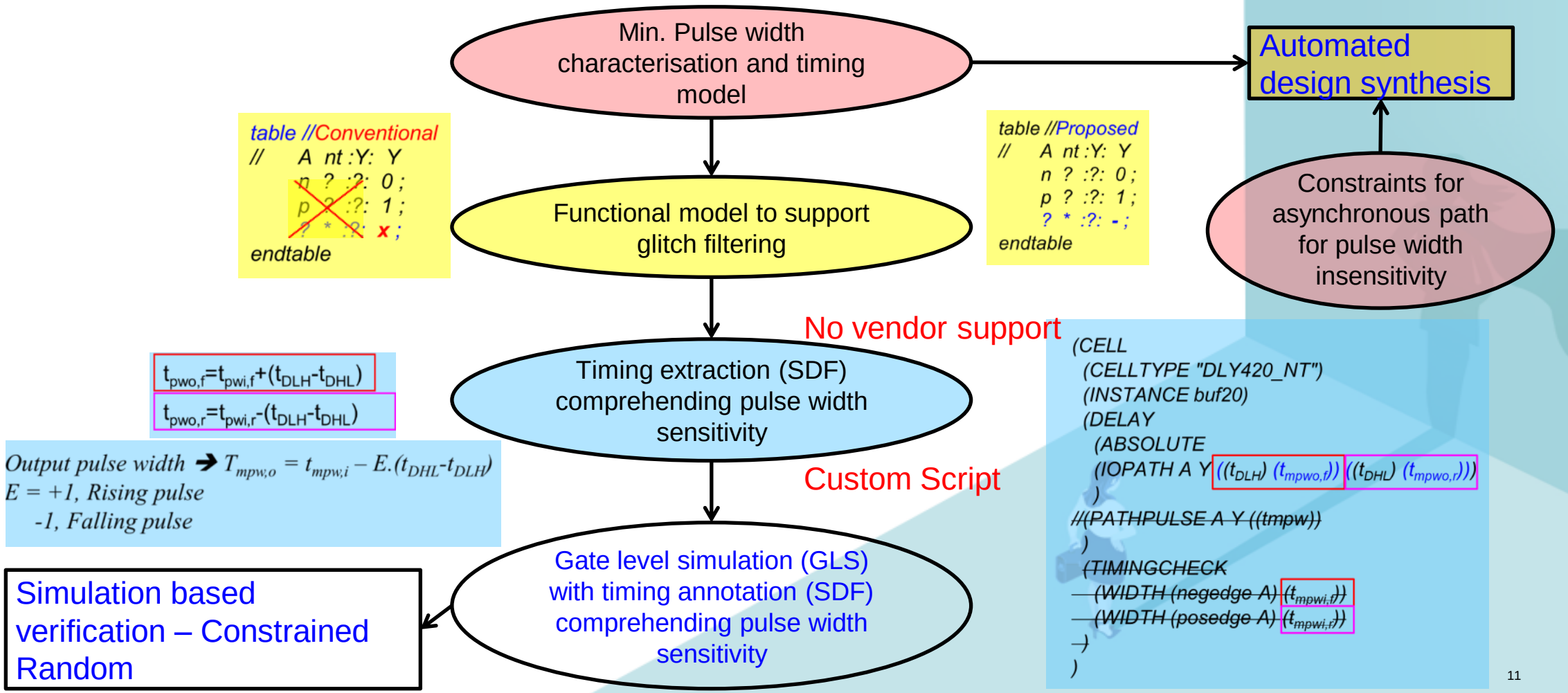


1. Apply input stimulus
2. Simulate the cell with SPICE netlist
3. Sweep the input pulse width
4. Check for correctness of propagated output value
    1. If pass Repeat 3 & 4
    2. If fail store the pulse width as t_{mpw}

Methodology – 3: Design Automation
Design Synthesis (Cadence Genus®)

- Define timing arc
- Define reference clock
- List of delay cells considered with t_{mpw} & t_D information
- Define the design constraint
  - Optional: Max. & min. number of stages (N_{max}, N_{min})
  - Recommended: Input transition control with strong, smallest delay buffer ($t_{slew,in}$)
  - Total delay ($t_{D,target}$)
  - Arrive at the delay line structure such that
    - $t_{D,target} = \sum_{j=1}^{N_{min} \le N \le N_{max}} t_{D,i}$
      –➔ Obtain N & Delay cell ordering
    - For j=1, N
      –Calculate $t_{mpw,o}$
      –$t_{mpw,o} = t_{mpw,i} – E.(t_{DHL}-t_{DLH})$
      –E = +1, Rising pulse
      -1, Falling pulse
      –such that, $t_{mpw,o,j-1} \geq t_{mpw,i,j}$

Methodology – 3: Design Automation & Verification



Methodology – 4: Static Timing Analysis

- Define timing arc
- Define reference clock
- List of delay cells considered with t_{mpw} & t_D information
- Find N=Number of cells in the timing arc
  - For j=1, N

    Query $t_{mpw,i}$

    Calculate $t_{mpw,o}$

    (Automatically done by STA tools)

    $t_{mpw,o} = t_{mpw,i} - E.(t_{DHL}-t_{DLH})$

    E = +1, Rising pulse

    -1, Falling pulse

  - Check for violation of design constraint

    (Covered by min. pulse width check in STA & synthesis tools)

    $t_{mpw,o,j-1} \geq t_{mpw,i,j}$

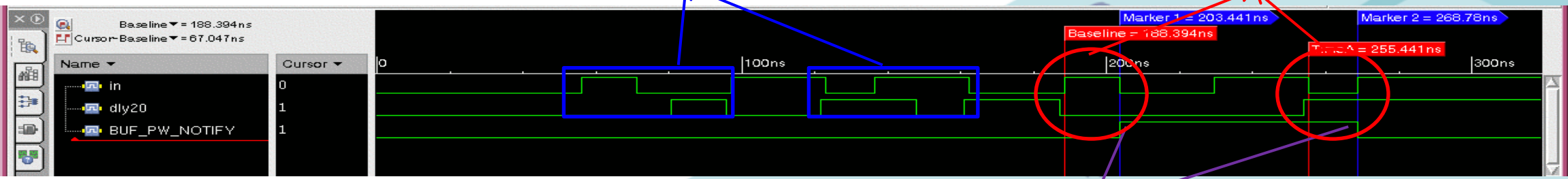
- Check $t_{D,target} = \Delta_{error} + \sum_{j=1}^{N_{min} \leq N \leq N_{max}} t_{D,i}$

Results

- GLS illustration: Simple buffer chain with pulse width filtering modelled
- Applied on a ultra-low power & low cost mixed-signal SoC
- Identified system dead-lock scenarios with conventional design method were verified to indeed cause irrecoverable state using proposed verification method
- Redesign using proposed design method
- The scenarios that were earlier failing were verified to be behaving robustly in the newer design

Output propagation:
Output pulse width > $t_{mpw,o}$

Output filtered:
Output pulse width < $t_{mpw,o}$



Timing check on input pulse width

Conclusions

Conclusions

- Pulse width insensitive design and verification method conceptualised & applied
- An area and power efficient design method for pulse width insensitive design
- Introduces a constraint driven, automated design synthesis methodology that can comprehend the pulse width sensitivity of the design components
- Introduces a simulation based verification method to analyse such behaviours and designs
  - Avoids costly SPICE based simulations to comprehend such scenarios
- Handles reliable design of asynchronous sections in an otherwise synchronous design
  - Known asynchronous design methods are difficult or costlier to apply
- Enabled cost and power competitive low power design implementation for mixed-signal SoC

Future scope to bridge limitations

- Fully automated back-end flow that comprehends pulse width sensitivity for design & SDF generation ➔ EDA Vendor engagement
- Exploration of formal design and verification methods

References

- "System Drivers: Mixed-signal Evolution," ITRS, 2011.
- Lakshmanan Balasubramanian, Puneet Sabbarwal, Rajesh Kumar Mittal, Prakash Narayanan, Ranjit Kumar Dash, Anand Devendra Kudari, Srikanth Manian, Sudhir Polarouthu, Harikrishna Parthasarathy, Ravi C Vijayaraghavan, Sachin Turkewadikar, “Circuit and DFT techniques for robust and low cost qualification of a mixed signal SoC with integrated power management system,” Design Automation & Test in Europe Conference & Exhibition (DATE) 2011, pp. 1-4, 2011..
- Rajesh Mittal, Lakshmanan Balasubramanian, Adesh Sontakke, Harikrishna Parthasarthy, Prakash Narayanan, Puneet Sabbarwal, Rubin A. Parekhji, “DFT for Extremely Low Cost Test of Mixed Signal SOCs with Integrated RF and Power Management,” 2011 IEEE International Test Conference, pp 1-10, 2011.
- Rubin A Parekhji, Michael Nicolaidis, “SOC Design Readiness for Automotive Applications,“ Design Automation & Test in Europe Conference & Exhibition (DATE) 2013, pp. 1-4, 2013.
- Lakshmanan Balasubramanian, Ruchi Shankar, Atul Ramakant Lele, Vijay Kumar Sankaran, Sharavathi Bhat, Vinaykumar S Hegde, “Surprise or Shock? Transistor level functional analysis of digital circuits and systems are still needed!,” CDNLive India, 2014.
- Lakshmanan Balasubramanian, Ruchi Shankar, Sharavathi Bhat, Vinaykumar S Hegde, “Novel Pulse Width Insensitive Design and Verification Methods,” Design Automation Conference (DAC), Designer Special, 2018. DOI=http://www2.dac.com/55th/proceedings/posters/51_3.pptx.
- Sudhakar Surendran, Venkatraman Ramakrishnan, “Timing Modeling Methodology for Formal Verification Tools,” Design Automation Conference (DAC), Designer Special, 2018.
- Sudhakar Surendran, Bernd Schneider, Kiran Rajmohan, Martin Schneider, Sebastian Fritz, “Formal & Simulation Method to Detect Unstable States in Asynchronous State Machines,” Design Automation Conference (DAC), Designer Special, 2018.

Acknowledgements

- Sameer Dabadghav(Ex. TI) of Analog Devices Inc. & Somasekar J, Neeraj Saxena, Padmini Sampath(Ex. TI), Ajith Subramonia, & Vivek Singhal of Texas Instruments Inc. for their motivation and support in pursuing the unknown
- Niraj Mahapatro of Sankalp Semiconductor and Vinaykumar S Hegde(Ex. TI) of Robby Technologies, Inc. for the library characterisation support
- Nikhil Chandrakant Sangani of Texas Instruments Inc. for help in identifying PATHPULSE SDF construct
- Ramakrishna Reddy(Ex. TI) of AMD Inc. for help towards several simulation and flow issues
- Somshubra Paul, Anand Kumar G of Texas Instruments Inc. & Ayon Dey(Ex. TI) of AMD Inc. for several discussions on analog and system level requirements, scenarios, care-abouts and limitations

Q&A

Thanks your for your interest, time and attention!