Formal Modeling and Verification of Delay-Insensitive Circuits

Hoon Park

November 2, 2015
Portland State University
Department of Electrical and Computer Engineering (ECE)
in collaboration with the Asynchronous Research Center (ARC)

Outline

- Introduction
- Related work and challenges
- Contributions
  - ARCtimer
  - Bounded Bundled Data (BBD)
  - Enhanced Semimodularity
- Future work

Introduction (1/5)

- Modern computer systems are distributed over space
  - **Internet of things**
    the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data
    [Wikipedia]

  - **IBM's TrueNorth**
    modular chips that act like neurons and form artificial neural networks to run "deep learning algorithms", like Skype's chat translator or Facebook's facial recognition

Introduction (2/5)

- Global state is a useful model for traditional clocked hardware
  - State may change only when the clock ticks
  - All tasks must fit into the clock period
  - Global state is stable between ticks
- **Global state and global clock control fail to scale over space**
  - They assume that events are simultaneous over space
  - They ignore communication delays
    - It's over 4 hours to communicate with "New Horizons" near Pluto
    - It's over 13 minutes to communicate with "Curiosity" on Mars

    - It takes hundreds of clock domains, ten thousand clock synchronizers and many clock ticks, to communicate within a single networking chip [Jeanne Trinko, IBM, Keynote Speech ASYNC 2013]

Introduction (3/5)

- Scalable hardware for modern computer systems requires
  - self-timed components with delay-insensitive communication between them
- The ARC studies exactly this type of scalable hardware
  - Inside a component, it's as chaotic as a kindergarten playground
  - Between components, behavior is as orderly as a "crocodile"

- This system works, provided
  - Supervisors watch over the playground to avoid accidents
  - Children use the "crocodile" between playgrounds

Introduction (4/5)

- My research work provides the system with supervisors who watch over the playground to avoid accidents

Introduction (4/5)

- My research work provides the system with supervisors who watch over the playground to avoid accidents

- **Analogy:**

  playground : self-timed component
  crocodile : channel connection with handshake protocol
  system : collection of components and handshake channels
  accident : wrong protocol
  supervisors : event ordering constraints, a.k.a. **relative timing constraints**

Introduction (5/5)

- My research work provides the system with supervisors who watch over the playground to avoid accidents

- I will generate those supervisors, a.k.a. **constraints**
  - using formal modeling and verification of components
  - where the components
    - are self-timed
    - communicate with handshake protocols

Related Work

- Metric Timing (Chris Myers & Tomohiro Yoneda, U. Utah/Tokyo)
  - Model and generate **absolute delay margins**
- COHO (Mark Greenstreet & Chao Yan, U. of British Columbia)
  - Model **analog** effects
- Analyze and Artist (Ken Stevens & Yang Xu, U. of Utah)
  - Model digital effects
  - Supported by customized tools based on CCS semantics
  - Enhanced with generic model checking tools
  - Generate **relative timing (RT) constraints** on events
- Process Spaces and FIREMAPS (Radu Negulescu, Waterloo U.)
  - Model digital effects
  - Supported by customized tools based on Process Algebras
  - Generate **chain constraints** on event-paths

My Approach

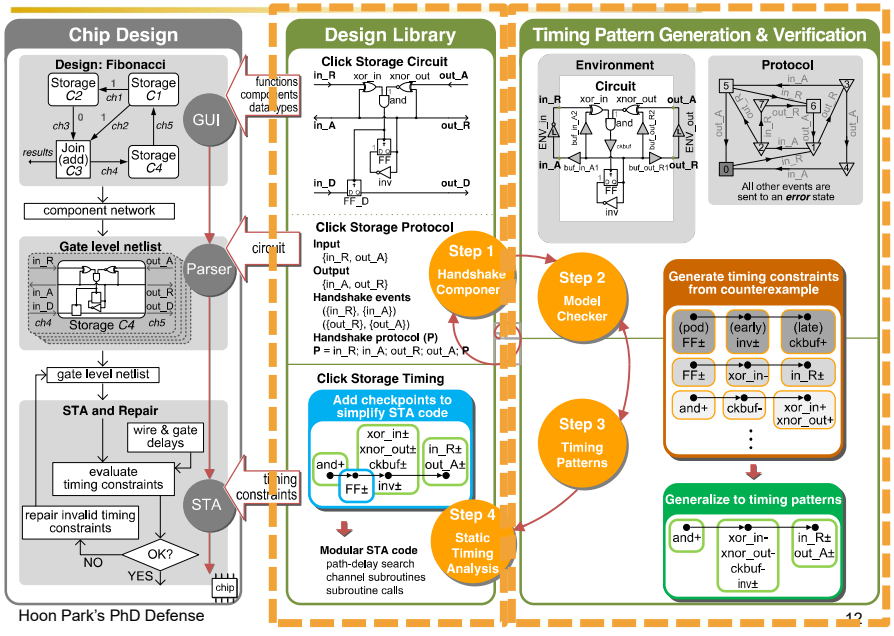
- Combines and extends that of Ken Stevens and Radu Negulescu
  - Models digital effects and relative event orderings
- Key extensions:
  - **Bundled data model** — absent from all other approaches
  - **Guarded event ordering constraint**
    - Without guards, it's a relative timing constraint
      - Useful because relative timing constraints have low complexity models
    - Guards can sequence timing constraints to form a chain constraint
      - Useful for translation into static timing analysis code
  - **Technology in-dependent constraint patterns**
  - **Modular and compositional** from constraint to static timing analysis
    - Per component constraint-generation, storage, and code-generation
- I dubbed this approach: ARCtimer
  - Timing closure foundation for ARCwelder, ARC's design compiler
  - Successfully applied to the Click circuit family and parts of GasP

Contributions Discussed In This Talk

NEW

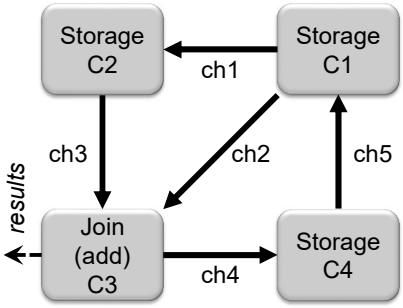
- ARCtimer overview
  - Upfront modeling on a component by component basis
  - Provides a library of verified components
    - Gate level description
    - Handshake protocol specification
    - Timing patterns and STA code
  - Provides modular and scalable system assembly and timing closure
- Data models and timing patterns for Bounded Bundled Data (BBD)
  - BBD extends Bundled Data (BD)
    - BD data must be valid during the entire handshake
    - BBD allows control signals to lead or lag behind data signals
- Relative timing aware semimodular model

ARCtimer – Overview

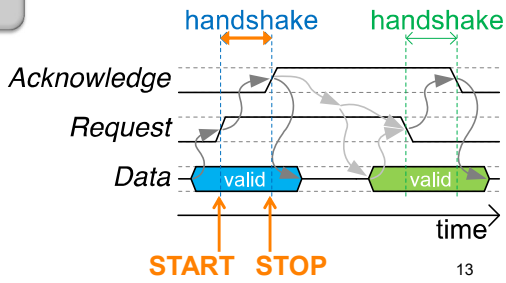


Handshake

- Handshake Channels and Bundled Data

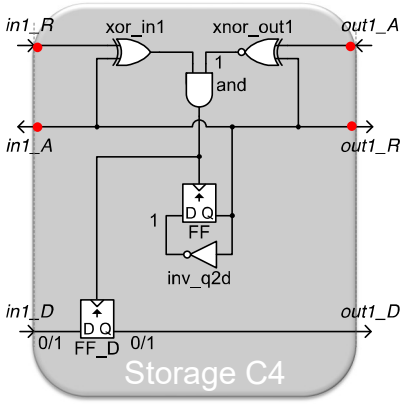


Data remains valid throughout the entire handshake.



ARCtimer – Step 1

- Circuit



- Handshake Protocol

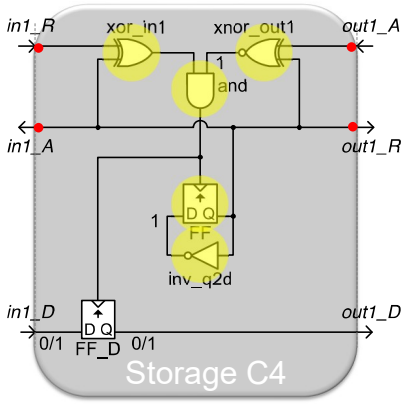
Input
{ *in1_R, out1_A* }
Output
{ *in1_A, out1_R* }

Handshake event ordering
({ *in1_R* }, { *in1_A* })
({ *out1_R* }, { *out1_A* })

Handshake protocol (P)
P = *in1_R*; *in1_A*; *out1_R*; *out1_A*; **P**

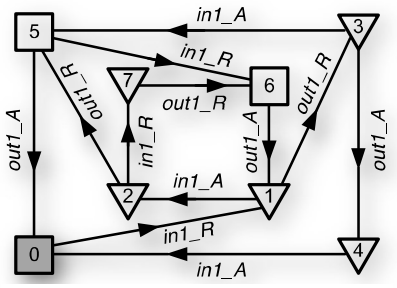
ARCtimer – Step 2

- Circuit



- Handshake Protocol

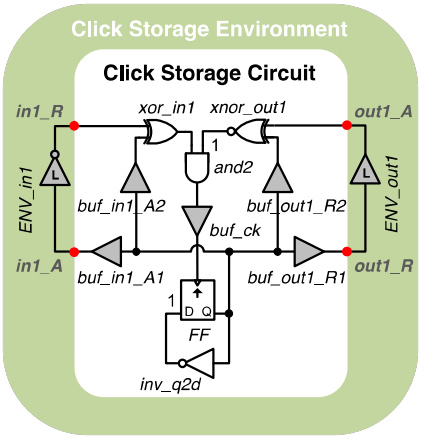
Any event not indicated in this FSM leads to an error state.



Preparation for the model checker

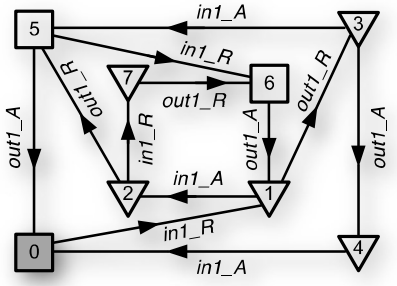
ARCtimer – Step 2

- Model of the Implementation



- Model of the Protocol

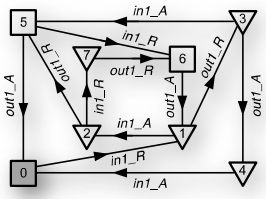
Any event not indicated in this FSM leads to an error state.



What am I checking?
PROPERTIES!

ARCtimer – Step 2

- Protocol properties
  - Never get into an error state
  - Always exit a triangle state ▽ within finite time
  - Keep existing protocol choices available



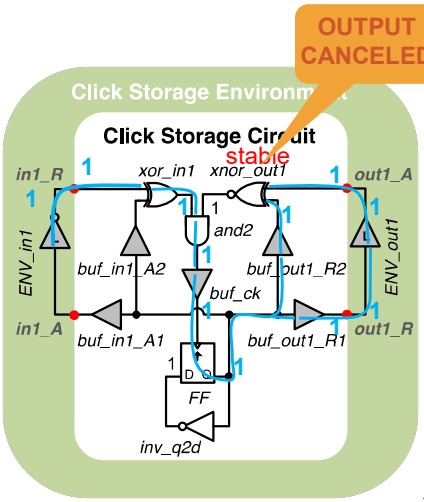
- Semimodularity property
  - Leave no event behind
  - Unstable gate can become *stable* only by changing its output.

- BBD property
  - Data setup and hold times
  - Correct propagation of valid data
  - Neither skip nor stutter data

What to do when the circuit fails a property?

ARCtimer – Step 3

- Analyze the counterexample for failing a semimodularity property



Semimodularity Failure

Trace of events:

ENV_in1+ in1_R+
xor_in1+
and2+
buf_ck+ FF.q+
buf_out1_R1+ out1_R
buf_out1_R2+
ENV_out1+ out1_A+
xnor_out1.semimodular = FALSE

FAIL

How do I fix it?

RT: FF.q+ → xnor_out1- < out1_A+
RT: FF.q- → xnor_out1- < out1_A-

RTp: and2+ → xnor_out1- < out1_A±

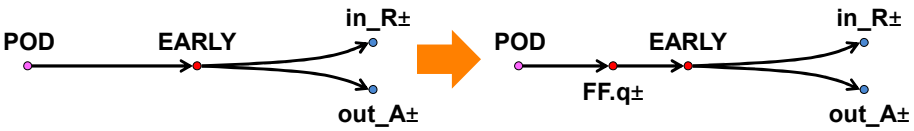
ARCtimer – Step 4

- Timing Patterns

POD	Early	Late
and2+	xor_in- xnor_out- buf_ck- inv_q2d±	in_R± out_A±

- STA Code

POD	FF	Early	Late
and2+	FF.q±	xor_in± xnor_out± buf_ck± inv_q2d±	in_R± out_A±



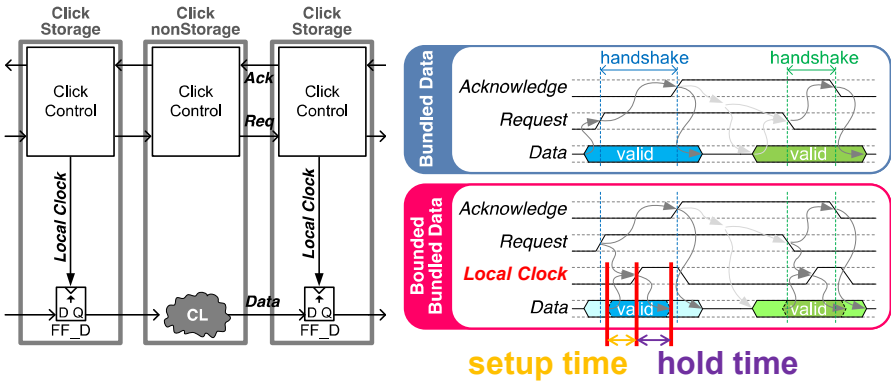
- Most STA tools cut paths at flipflops
- Some STA cannot differentiate rising and falling events
- The STA code on the right uses our new *guarded event constraint*

Contributions Discussed In This Talk

- ARCtimer overview
  - Upfront modeling on a component by component basis
  - Provides a library of verified components
    - Gate level description
    - Handshake protocol specification
    - Timing patterns and STA code
  - Provides modular and scalable system assembly and timing closure
- Data models and timing patterns for Bounded Bundled Data (BBD)
  - BBD extends Bundled Data (BD)
    - BD data must be valid during the entire handshake
    - BBD allows control signals to lead or lag behind data signals
- Relative timing aware semimodular model

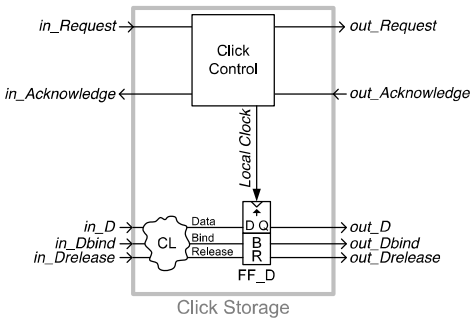
BBD

- BBD is an extension of BD
- Helpful when throughput is limited by the Combinational Logic (CL)
  - Allows time borrowing between components
  - Minimizes cost for delay matching



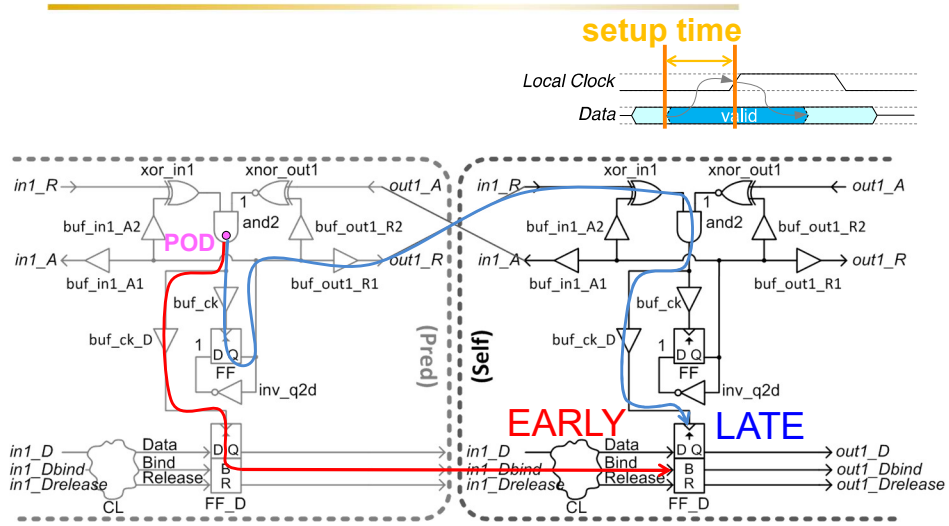
BBD – Data Model

- Data validity is modeled using ***Bind*** and ***Release***



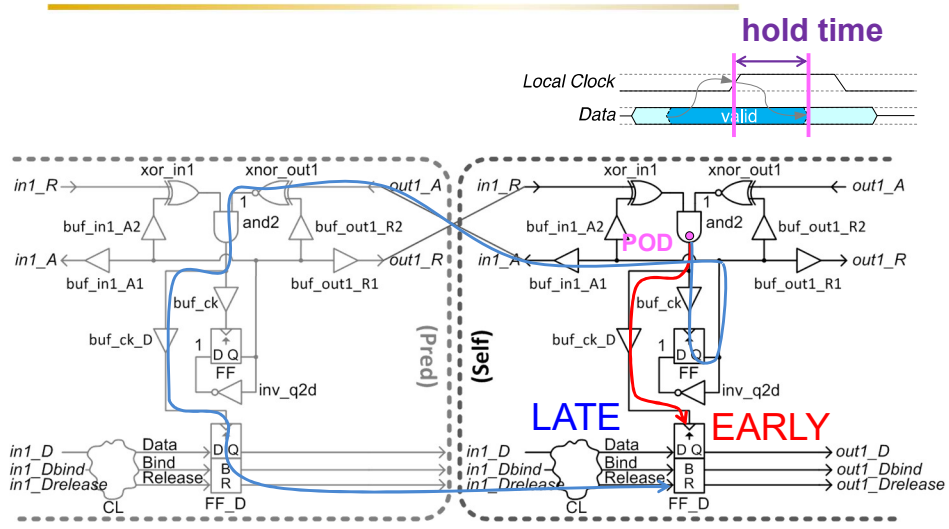
- Changing Bind declares ***new*** data valid
- Changing Release declares ***old*** data invalid

BBD – Setup Constraint



Pred.and+ → CL.bind± < buf_ck_D+

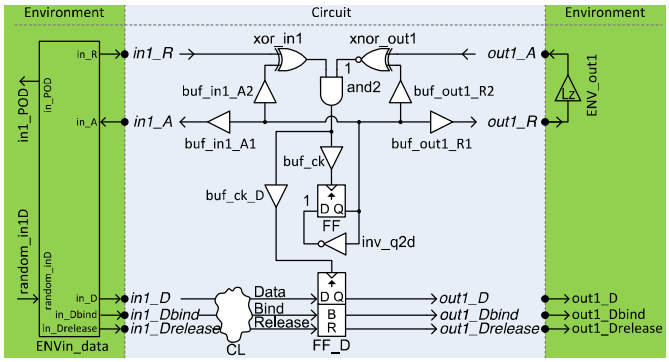
BBD – Hold Constraint



and+ → buf_ck_D+ < CL.Release±

Summary of Timing Patterns

	POD	Early	Late
Control	and2+	xor_in- xnor_out- buf_ck- inv_q2d±	in_R± out_A±
Data Setup	in_POD±	CL.bind±	and2±
Data Hold	and2+	buf_ck_D±	CL.release±



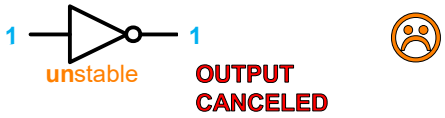
Contributions Discussed In This Talk

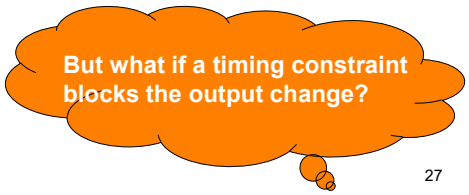
- ARCtimer overview
  - Upfront modeling on a component by component basis
  - Provides a library of verified components
    - Gate level description
    - Handshake protocol specification
    - Timing patterns and STA code
  - Provides modular and scalable system assembly and timing closure
- Data models and timing patterns for Bounded Bundled Data (BBD)
  - BBD extends Bundled Data (BD)
    - BD data must be valid during the entire handshake
    - BBD allows control signals to lead or lag behind data signals
- Relative timing aware semimodular model

Semimodularity

- Unstable gate can become stable only by changing its output.

- If an input change causes the gate to become stable,
  it is a violation of semimodularity!





Semimodularity

- Old definition
  - If a gate is unstable,
    it becomes stable by changing its output **OR** leaves its output unchanged.
    In either case, the input function must remain unchanged.

$$out \neq F(in_1, \cdots, in_n) \quad \rightarrow$$
$$\left[\{ out' = F(in_1, \cdots, in_n)\} \ \vee \ (out' = out)\right] \ \wedge \ \{F(in'_1, \cdots, in'_n) = F(in_1, \cdots, in_n)\}$$

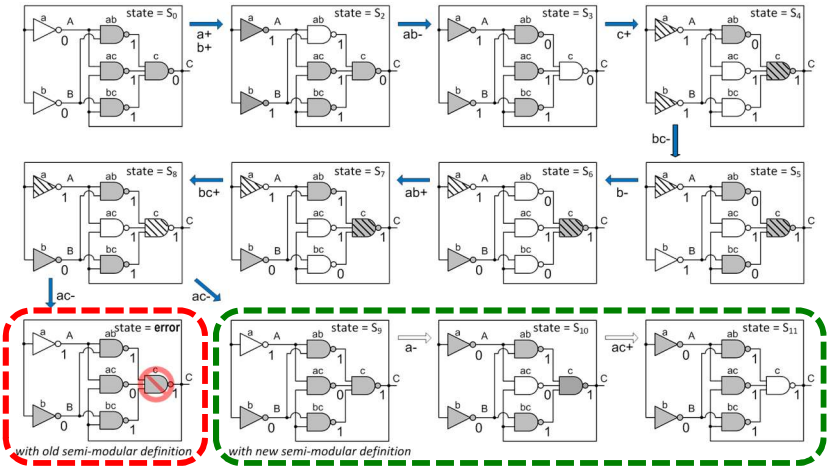
- New definition
  - If a gate is unstable **and unblocked**,
    it becomes stable by changing its output **OR** leaves its output unchanged.
    In either case, the input function must remain unchanged.

$$\{out \neq F(in_1, \cdots, in_n)\} \wedge \left[\{out \wedge \neg block(out-)\} \vee \{\neg out \wedge \neg block(out+)\}\right]$$
$$\rightarrow$$
$$\left[\{out' = F(in_1, \cdots, in_n)\} \vee (out' = out)\right] \wedge \{F(in'_1, \cdots, in'_n) = F(in_1, \cdots, in_n)\}$$

Semimodularity – C element

Constraints Set
c+ → ac- < a-
c+ → bc- < b-
c+ → ac- < c-
c+ → bc- < c-

- Avoid over-constraining the C element



Future Work

- Publish ARCtimer results for the Click circuit family
- Improve current ARCtimer automation
  - Particularly the translation of relative timing patterns to STA code
  - Increase modeling capacity
    - Add model checker for distributed computing – e.g. Lamport's TLA
    - Add theorem proving – e.g. UT Austin's ACL2
- Apply to circuit families that use *Naturalized Communication* [2]
  - Naturalized communication separates components into links and joints
  - and a standard link-joint interface shared by all self-timed circuit families.

[2] M. Roncken, S.M. Gilla, **H. Park**, N. Jamadagni, C. Cowan, and I. Sutherland. Naturalized communication and testing. In Asynchronous Circuits and Systems (ASYNC), 2015 21st IEEE International Symposium

Publications

- Journals
  - **H. Park**, A. He, M. Roncken, and X. Song.
    Semi-modular delay model revisited in context of relative timing.
    IET Electronics Letters, 51(4):332-334, 2015.
  - **H. Park**, A. He, M. Roncken, X. Song, and I. Sutherland.
    Modular timing constraints for delay-insensitive systems.
    JCST, Springer, accepted 2015.
- Conferences & Posters
  - M. Roncken, S.M. Gilla, **H. Park**, N. Jamadagni, C. Cowan, and I. Sutherland.
    Naturalized communication and testing.
    Asynchronous Circuits and Systems (ASYNC), pages 77-84, 2015.
  - **H. Park**, A. He, M. Roncken, X. Song.
    Verifying timing constraints for delay-insensitive circuits.
    Poster presentation. Asynchronous Circuits and Systems (ASYNC), May 2015.
  - M. Faust, H. Chung, **H. Park**, and J. Rodriguez.
    Introducing hardware emulation in the ECE curriculum.
    IEEE International Conference on Microelectronic Systems Education (MSE '11), pages 39-40, June 2011.

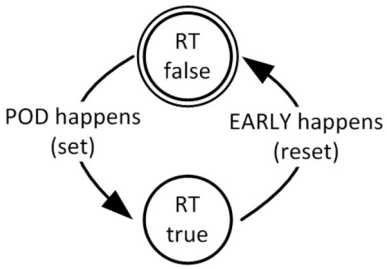
Acknowledgements

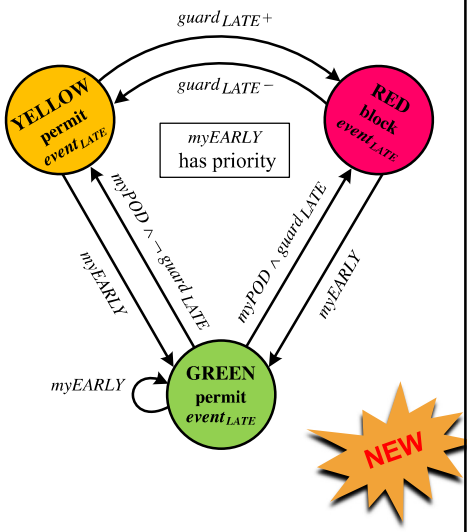
- Advisors & Collaborators
  - Dr. Xiaoyu Song
  - Marly Roncken
  - Anping He (Collaborator)
- Committees
  - Dr. Douglas Hall
  - Dr. Fu Li
  - Dr. Christof Teuscher
  - Dr. Jingke Li
- ARCwelders
  - Willem Mallon
  - Ivan Sutherland
  - Swetha Mettala-Gilla
  - Chris Cowan
  - Navaneeth Jamadagni

Backup Slides

Relative Timing Stoplight Model

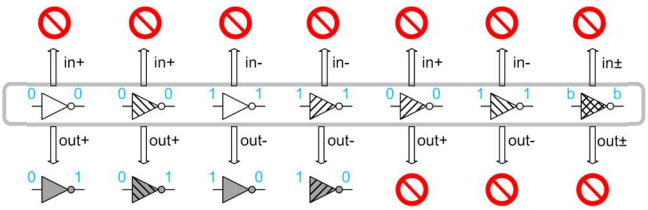
- Old
- New



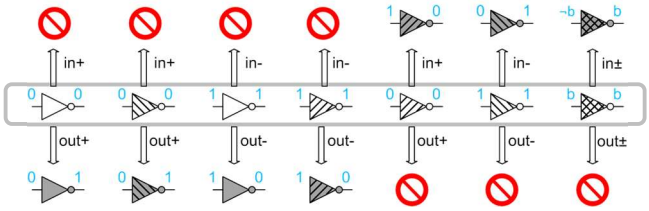


Semimodularity

- Old definition



- New definition



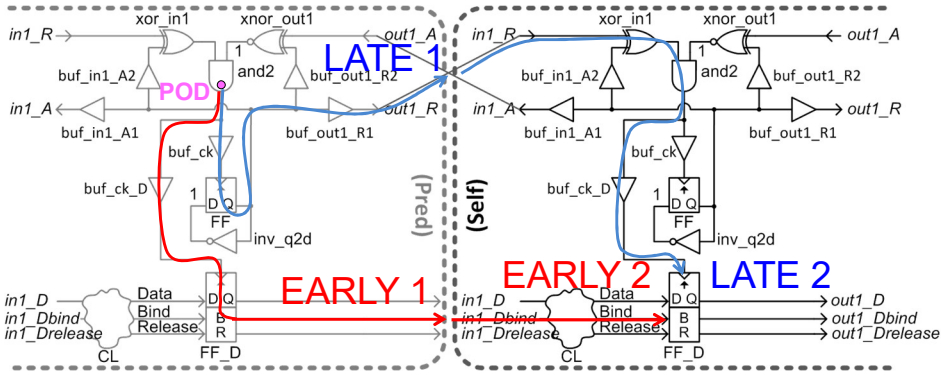
BBD – Setup Constraint for STA

- Timing Patterns

	POD	Early	Late
Data Setup	in_POD±	CL.bind±	and2±

- STA Code

	POD	Early	Late
Data Setup	in_POD±	FF_D.Data±	FF_D.ck+



Pred.and+ → CL.bind± < buf_ck_D+