ABCD-D

Modelling the Analog Dynamics of Digital Components using Finite State Machines

Aadithya V. Karthik

Jaijeet Roychowdhury

The University of California, Berkeley

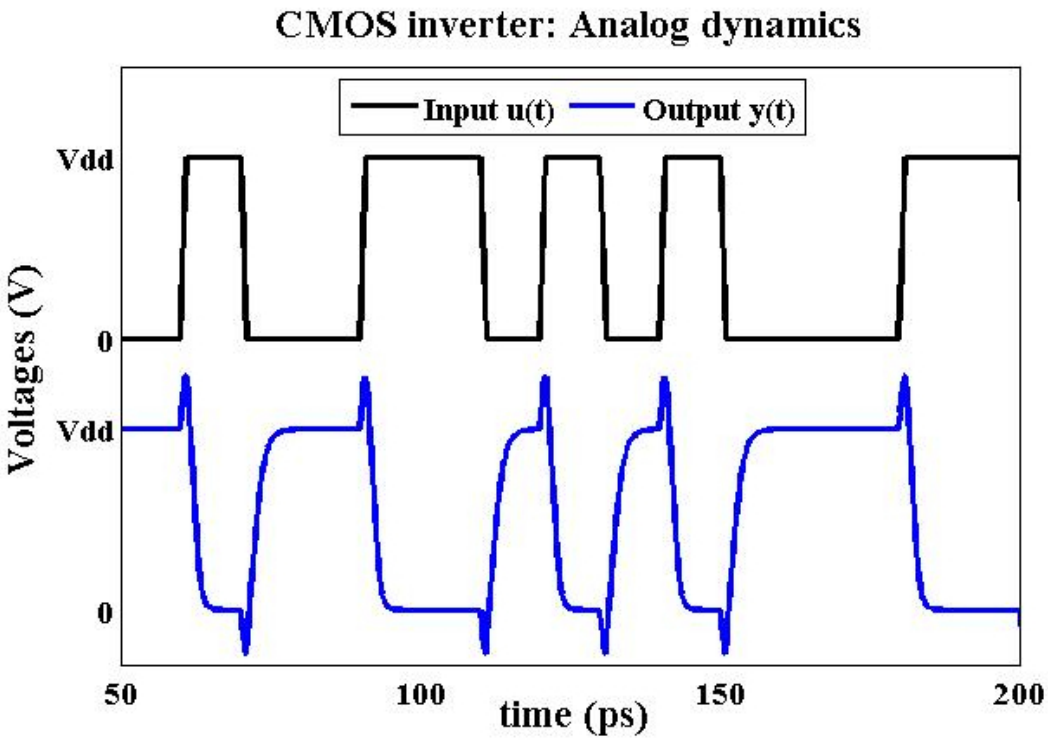
Mar 2013, TAU, Lake Tahoe

Overview of this talk

- The problem: modelling analog dynamics of digital components
  - Motivation: relevance for 32nm and below CMOS
  - Modelling goals: accuracy, simulation efficiency, formal verifiability
- ABCD-D in the context of existing techniques
  - SPICE, table-based library characterisation, ECSM/CCS, etc.
- DAE2FSM, ABCD, ABCD-L and ABCD-D
  - basket of tools to capture analog dynamics using Boolean models
- ABCD-D: the core technique, illustrated with an example
- ABCD-D: preliminary results, composability
- Summary, conclusions, and future work

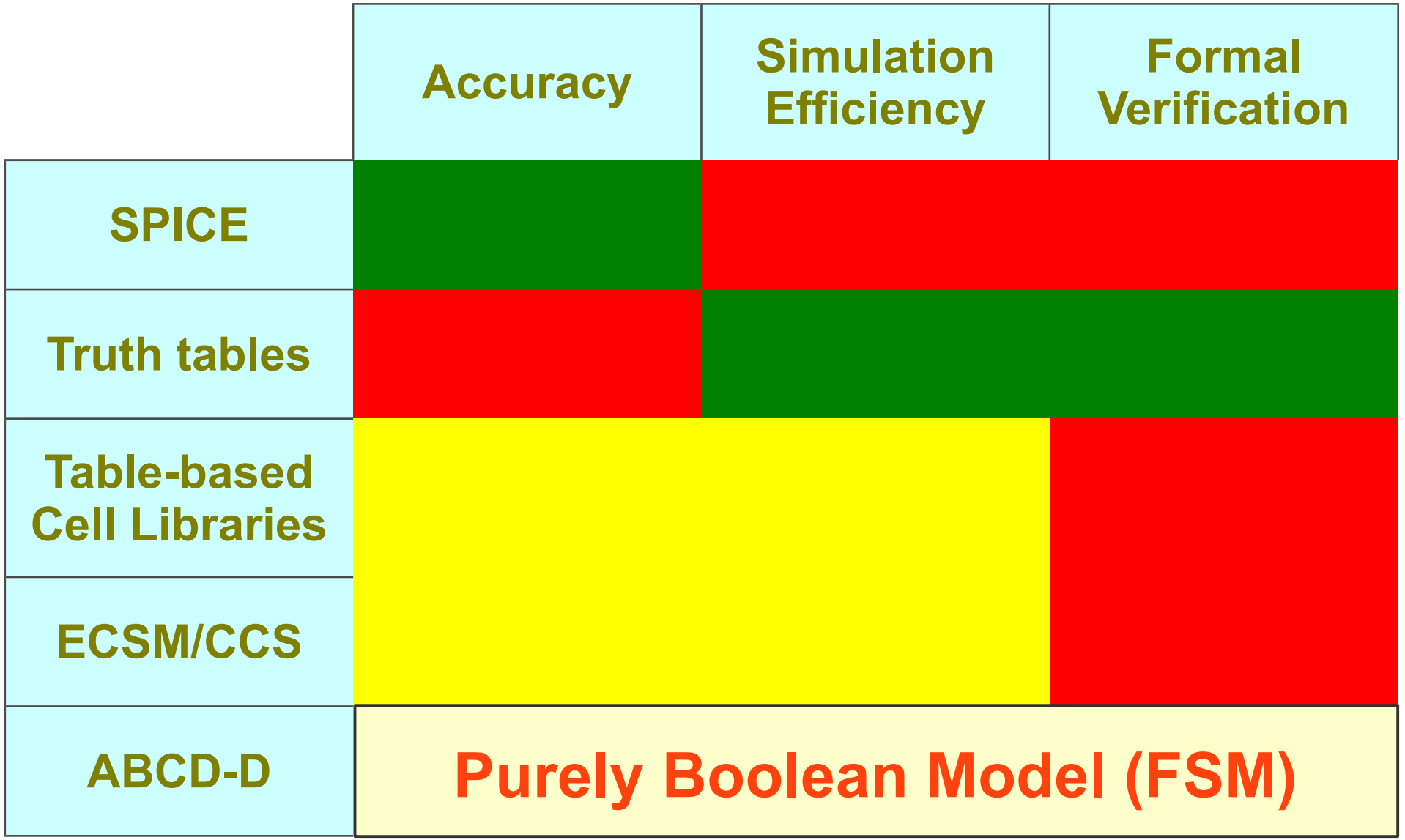
Digital Components, Analog Dynamics

- Problem known for a while
- Today's analog effects different
  - e.g., GIDL/GISL
  - short-channel effects
  - tunnelling
- New reasons for departure from purely digital behaviour
  - new models (e.g., BSIM4)



- Drivers for increased accuracy
  - aggressive performance targets (multi Gb/s throughput)
  - increasingly non-ideal devices at 22nm and below
    - many more parasitic factors, high parameter variability
- Drivers for increased simulation, formal verification efficiency
  - much bigger systems
  - much more complicated dynamics

ABCD-D vs Existing approaches



ABCD-D: Boolean but Accurate

Analog dynamics

$u(t)$

$$\frac{d}{dt}\vec{q}(\vec{x}(t)) = \vec{f}(\vec{x}(t), u(t))$$
$$y(t) = \vec{c}^T \vec{x}(t)$$

$y(t)$

Discretized into multiple levels (not just 2)

Sample

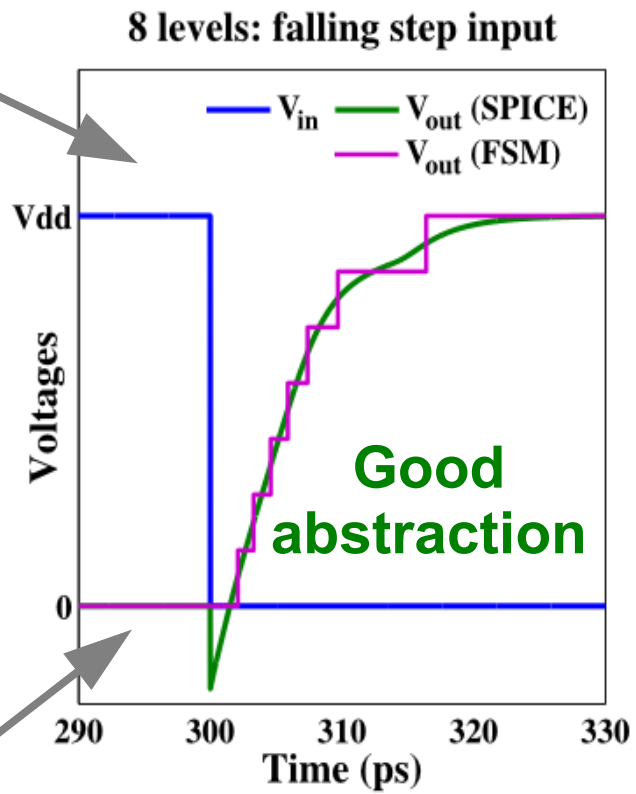
Interpolate

$$s_{i+1} = \delta(s_i, u_i)$$
$$y_i = G(s_i, u_i)$$

$u_0, u_1, u_2, \ldots$

$y_0, y_1, y_2, \ldots$

ABCD-D model



Key ideas: FSM symbols are time-sampled analog values

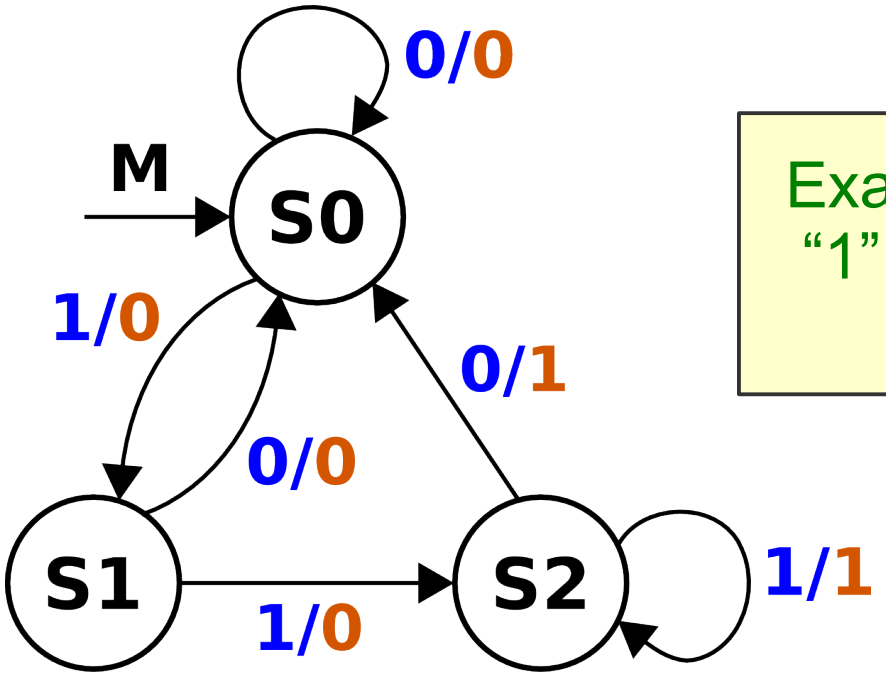
Multi-level discretization boosts accuracy

DAE2FSM, ABCD, ABCD-L, ABCD-D

- Suite of modelling techniques for Analog → Boolean
  - Motivation: fast simulation + formal analysis/verification
- DAE2FSM: first technique that was developed (Chenjie Gu)
  - works for small systems with “simple” analog dynamics
  - not very scalable (limitations of Angluin)
    - discretization has to be coarse (both time and signal)
- ABCD: **A**ccurate **B**ooleanization of **C**ontinuous **D**ynamics
  - umbrella of techniques: more scalable than DAE2FSM
  - support for systems with much richer dynamics
    - pure digital, pure analog, mixed-signal applications
  - e.g., ABCD-L for LTI systems
  - e.g., ABCD-D for analog dynamics of digital components, etc.

ABCD-D: The Core Technique

- Recap: What is an FSM?



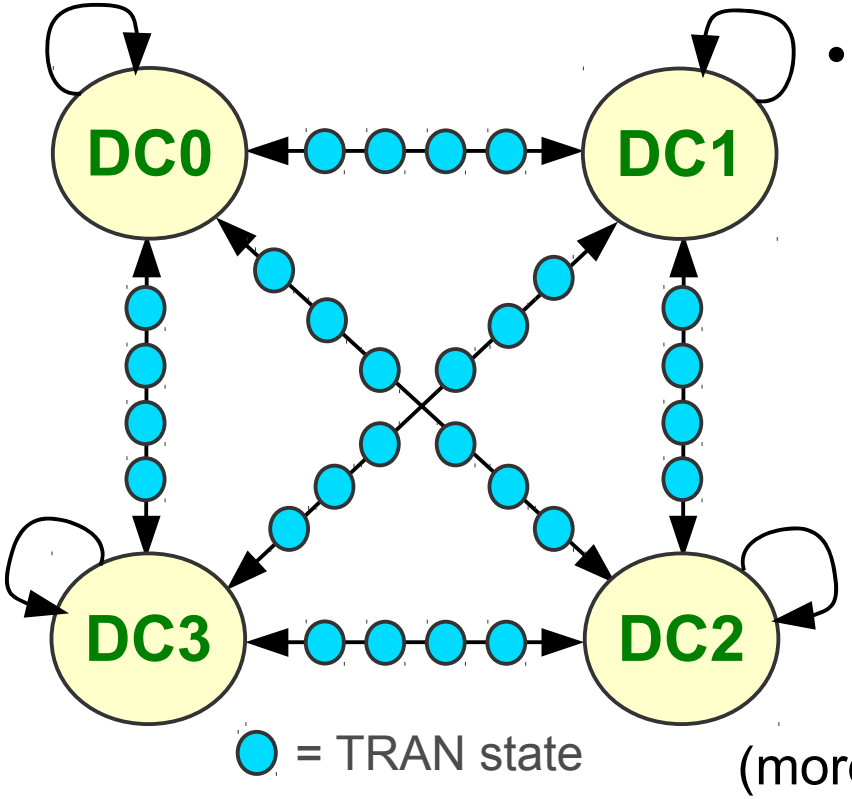
Example: This FSM outputs a "1" if and only if the previous two inputs are "1"

- System with finite set of states, finite I/O alphabet
  - Well-defined initial state
- Transition rules of the form
  - (current state, input) → (next state, output)

ABCD-D: DC, TRAN states in FSM

- Consider digital component D (say 1 i/p, 1 o/p)
  - signals discretized into M levels by ABCD-D
  - *each of the M DC inputs corresponds to a DC state in the FSM*

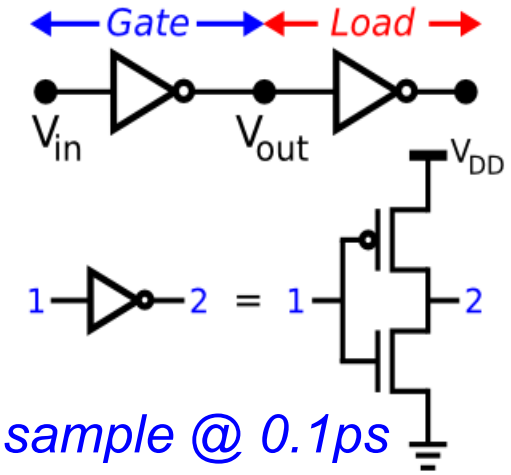
- Eg: 4 DC inputs {u0, u1, u2, u3} → 4 DC states {dc0, dc1, dc2, dc3}
  - if input settles to u2, FSM state will settle to dc2, and so on



- Transient inputs: step from u0 to u3?
  - Cannot change from dc0 to dc3 instantly
  - Introduce TRAN states between (dc0, dc3)
  - and between every pair of DC states
    - How many TRAN states?
      - based on time taken for dc0 → dc3
    - Plus, tag DC, TRAN states with outputs
    - *Result: ABCD-D FSM*

(more general model than what was presented in paper)

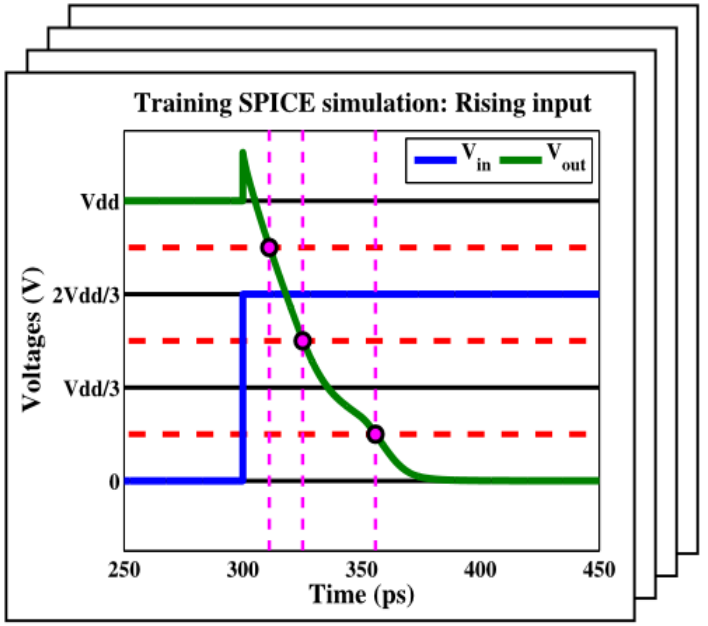
Example: CMOS inverter



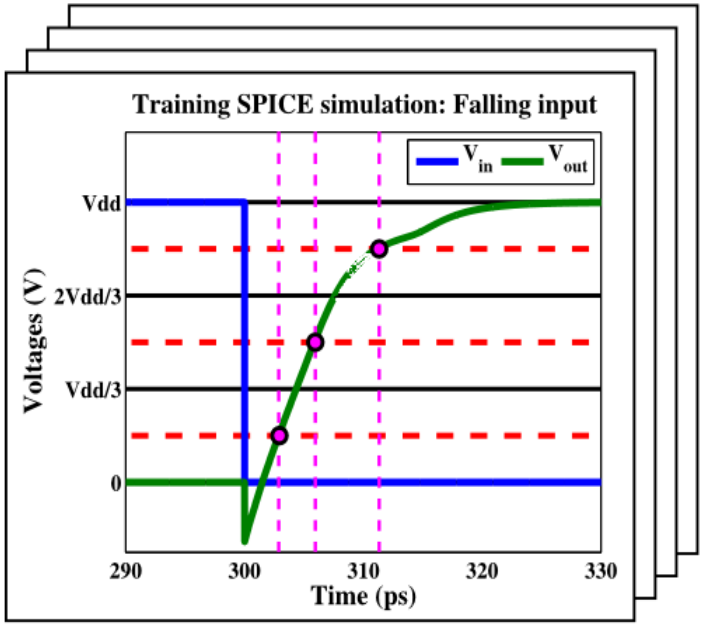
sample @ 0.1ps

4-level discretization

q0 = 0
q1 = Vdd/3
q2 = 2Vdd/3
q3 = Vdd



SPICE runs: rising input, falling output



SPICE runs: falling input, rising output

Step 1: SPICE-simulate ckt. on step inputs

Step 2: Construct tables of crossover times

Step 3: Construct DC, TRAN states based on such tables

Input	Threshold crossover times (in tenths of ps) 5Vdd/6	Vdd/2	Vdd/6
Vdd → 0	21.84	41.08	67.60

Crossover times: rising input, falling output

Input	Threshold crossover times (in tenths of ps) Vdd/6	Vdd/2	5Vdd/6
Vdd → 0	28.71	58.69	111.74

Crossover times: falling input, rising output

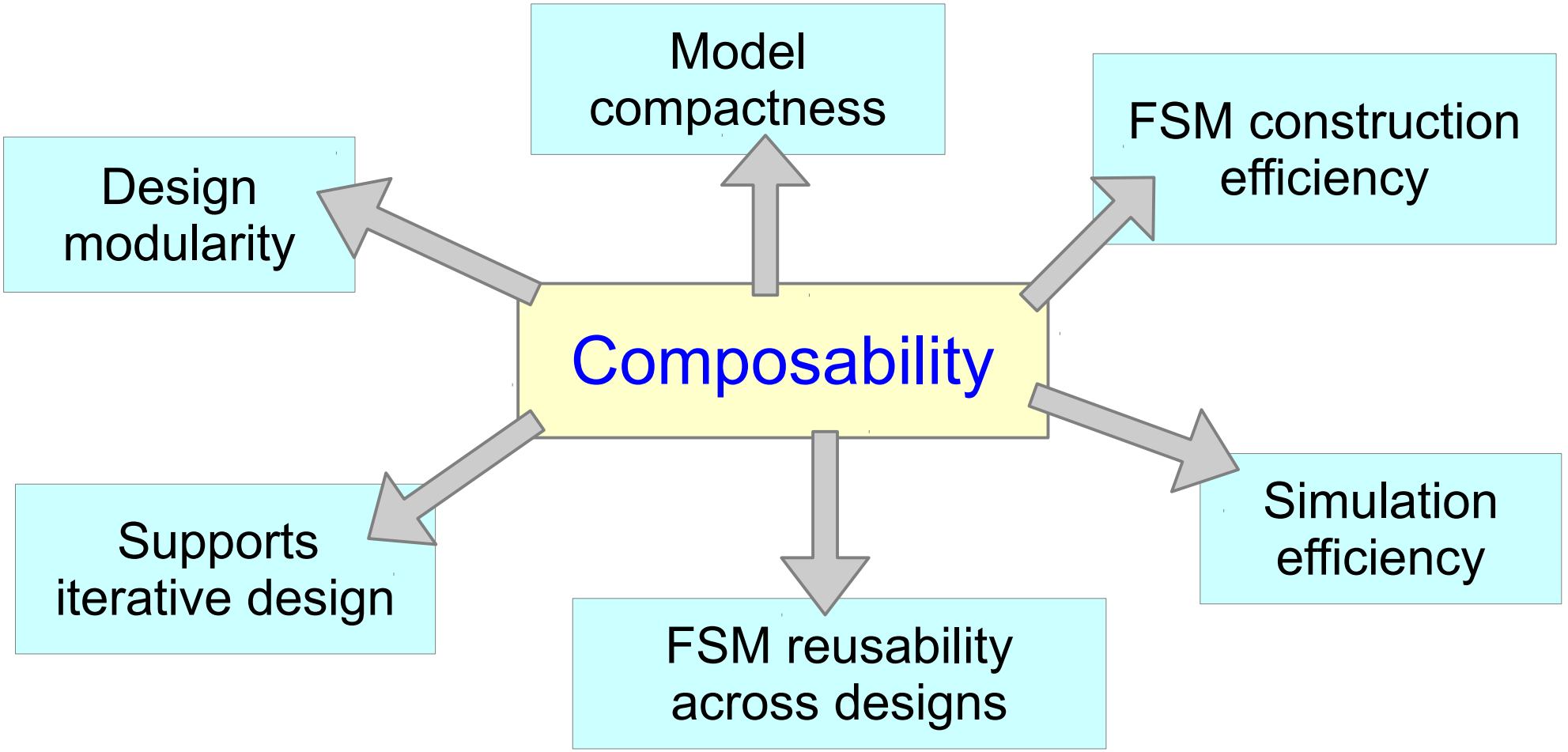
Q: What about multi-input gates, sequential logic?

A: Can be done; straightforward, but unable to discuss due to time constraints

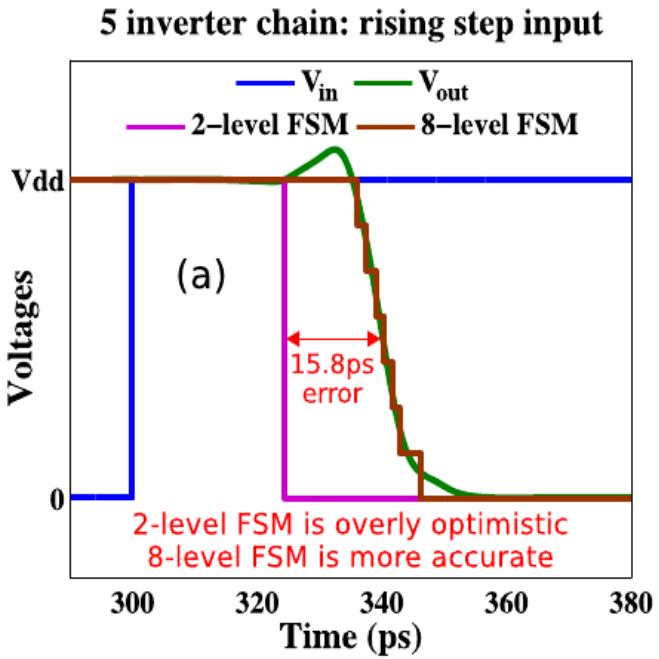
Composability of ABCD-D models

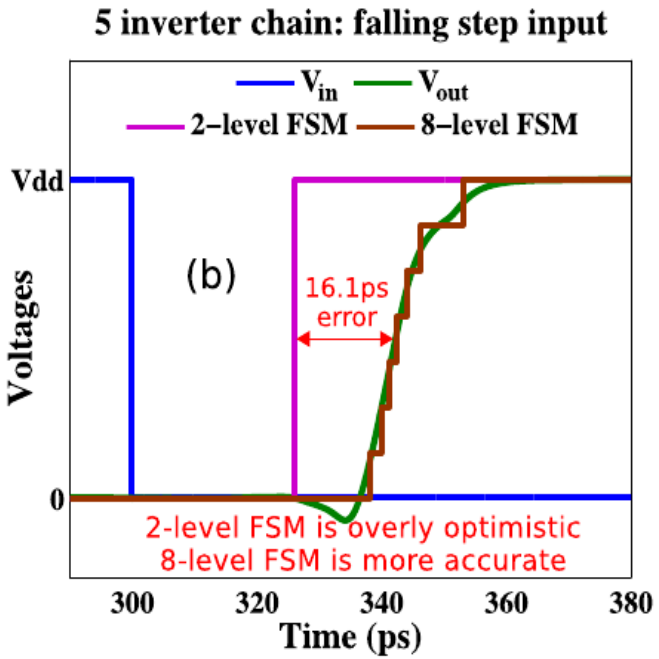
Output of one FSM can be fed as input to another

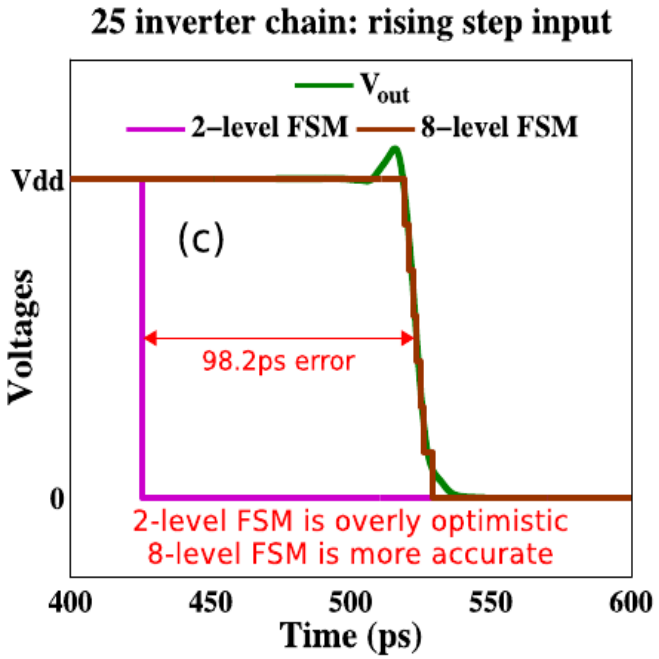
Predict o/p of large circuits by composing FSMs together

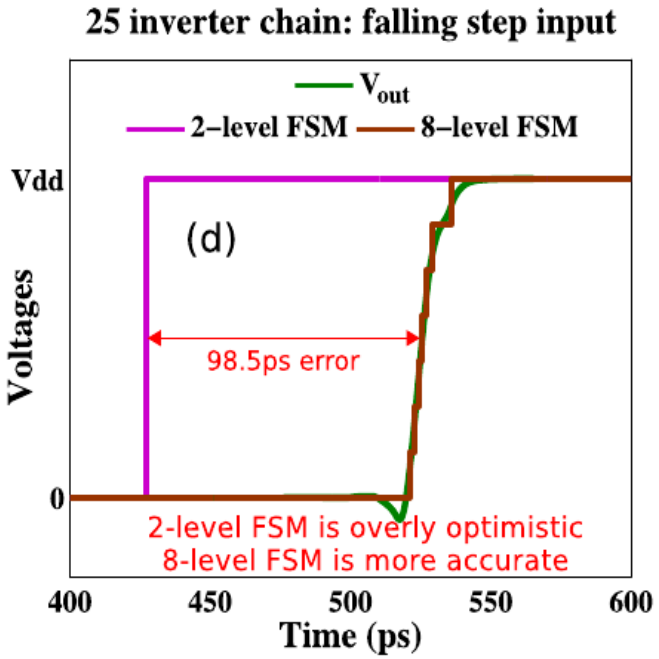


Composability: Chain of inverters





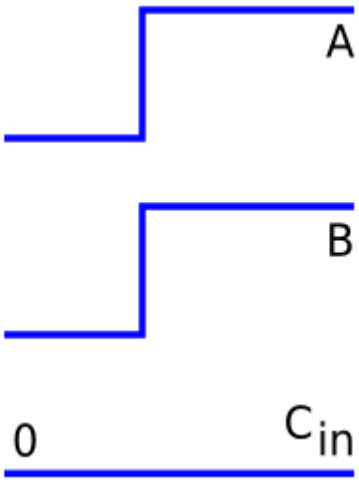


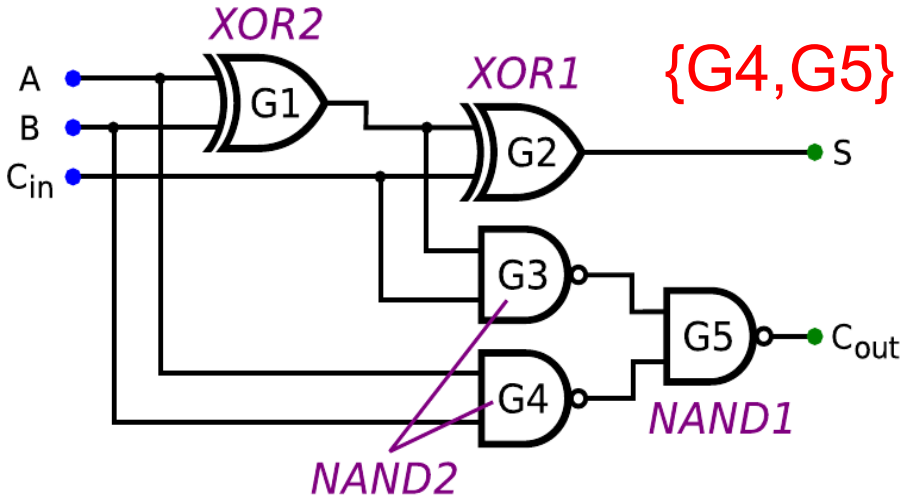


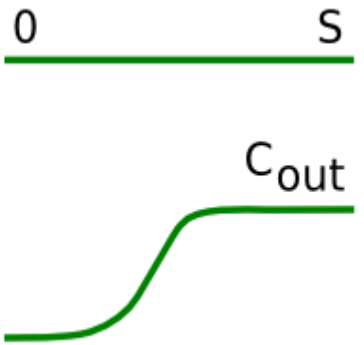
Finer quantization, better accuracy!

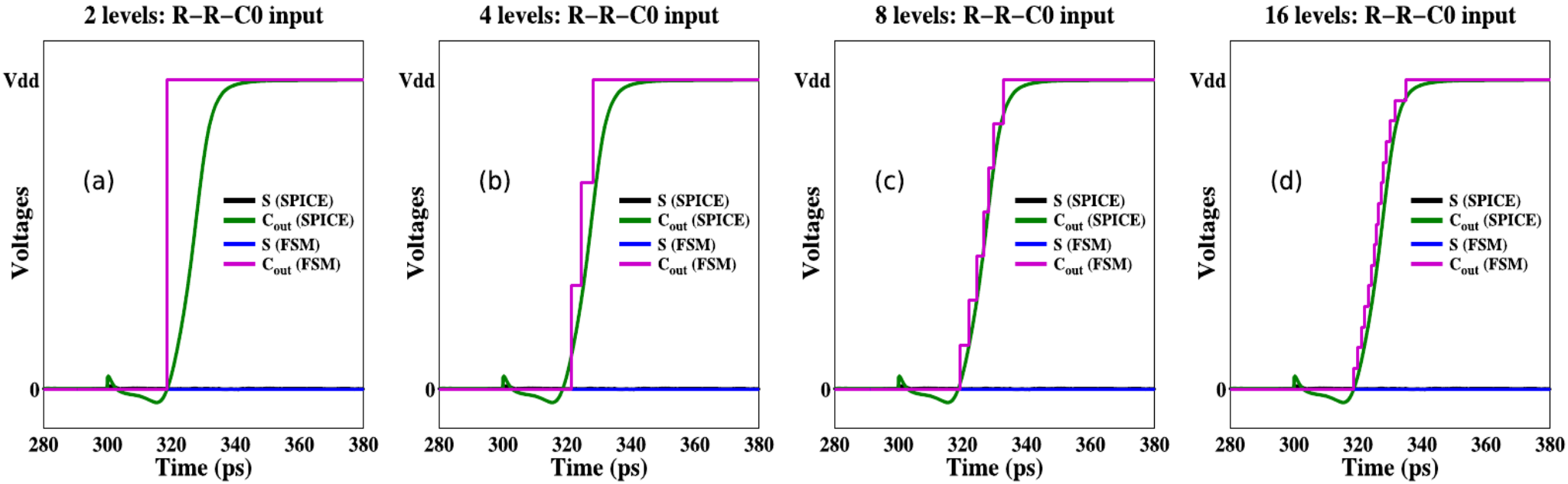
FSM ~1% of SPICE

Multi-input Composability: Full Adder (1/2)

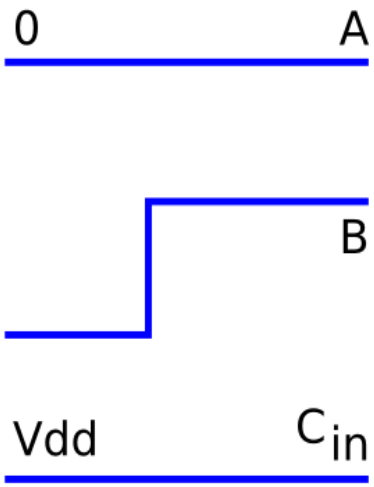


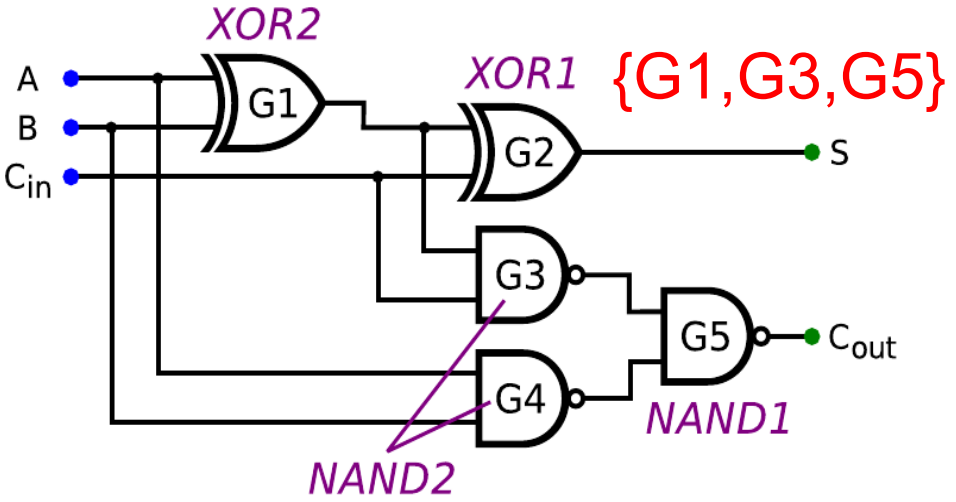


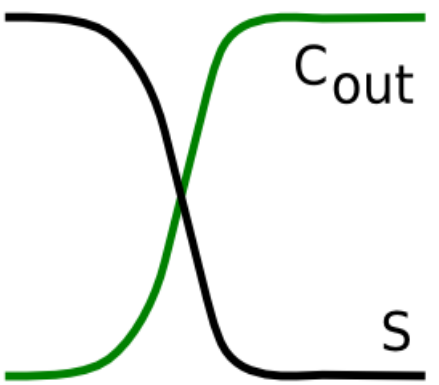


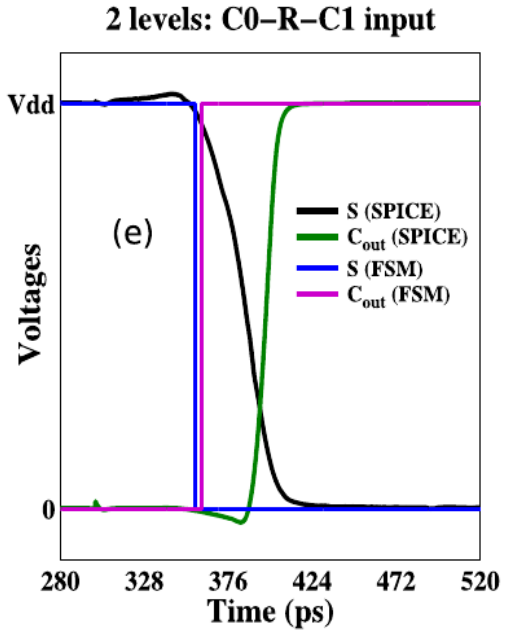


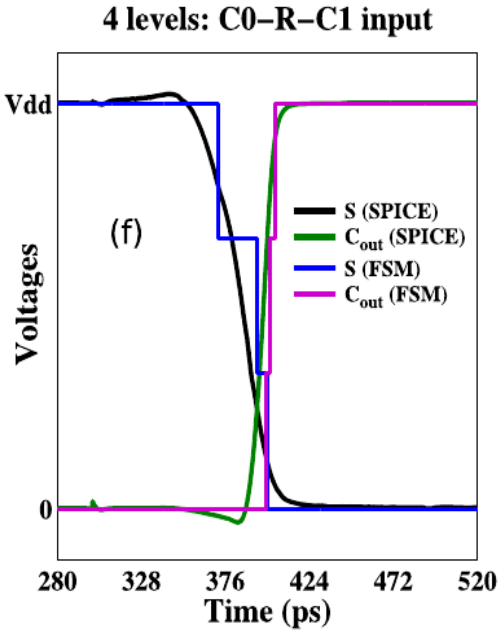
Multi-input Composability: Full Adder (2/2)

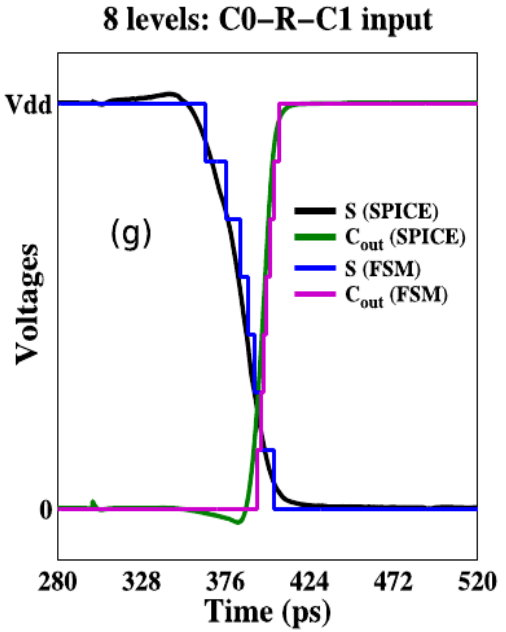


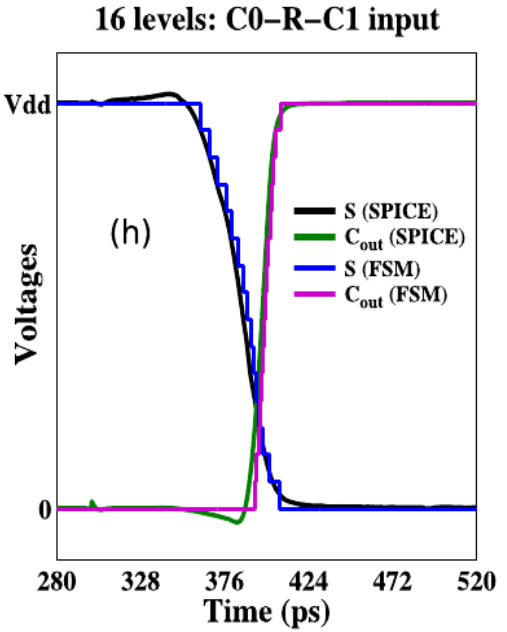












Summary and conclusions

- ABCD-D: technique to model analog dynamics in digital components, using purely Boolean models (FSMs)
- Key idea: Multi-level discretization of ckt. signals
  - enables near-SPICE accuracy
- FSM construction involving DC, TRAN states
- Key property of ABCD-D models: Composability
- ABCD-D enables fast simulation, formal verification
  - even in the presence of analog effects

Future work

- Larger examples (e.g, 64-bit adder)
- Logic synthesis and formal verification (w/ ABC)
- ABCD-D + ABCD-L, for interconnect analysis

Questions?