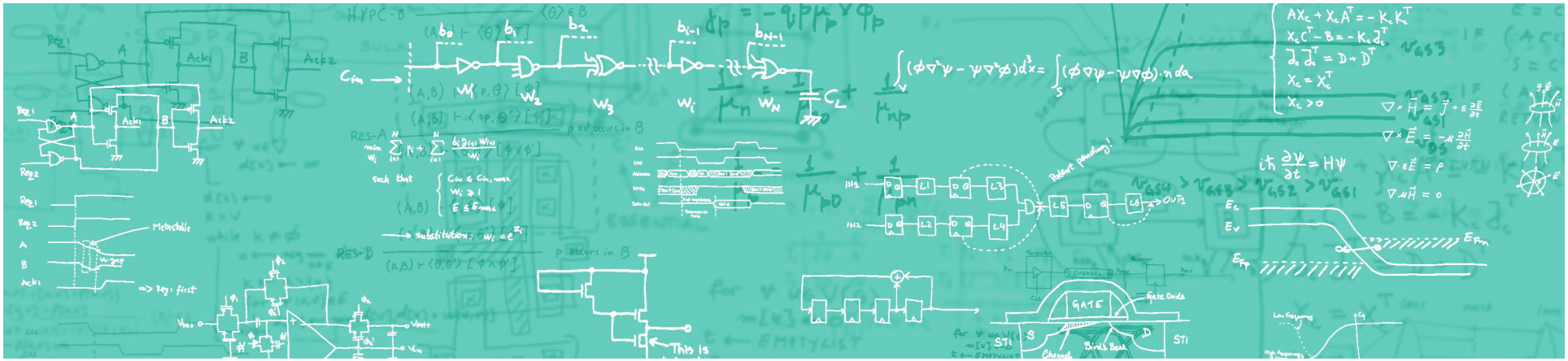


Will the 'FM' Have a Real Impact on the 'CAD'

FMCAD Panel Discussion
November 13, 2007

Andreas Kuehlmann

kuehl@cadence.com

The answer is simply…

…YES,
it had already a lot of impact

kuehl@cadence.com

Thank You

Questions/Comments?

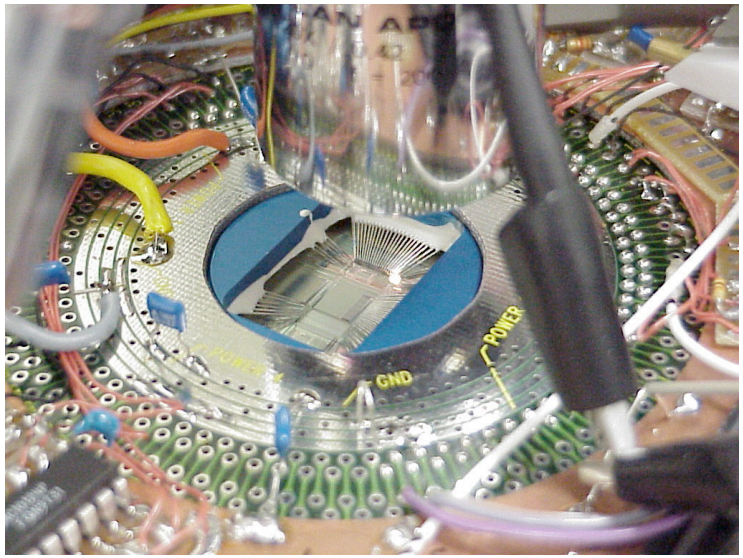
kuehl@cadence.com

There are many examples

just a few…

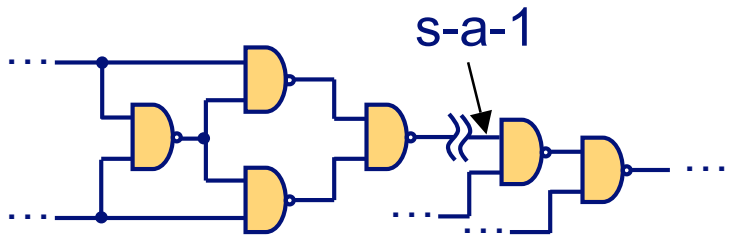
Example: ATPG

- Automatic Test Pattern Generation is one of the oldest application of FM-kind methods
  - J. Paul Roth:
    "Diagnosis of Automata Failures:
    A Calculus and a Method"
    IBM Journal, Jul. 1966, pp 278-291



  - Problem:

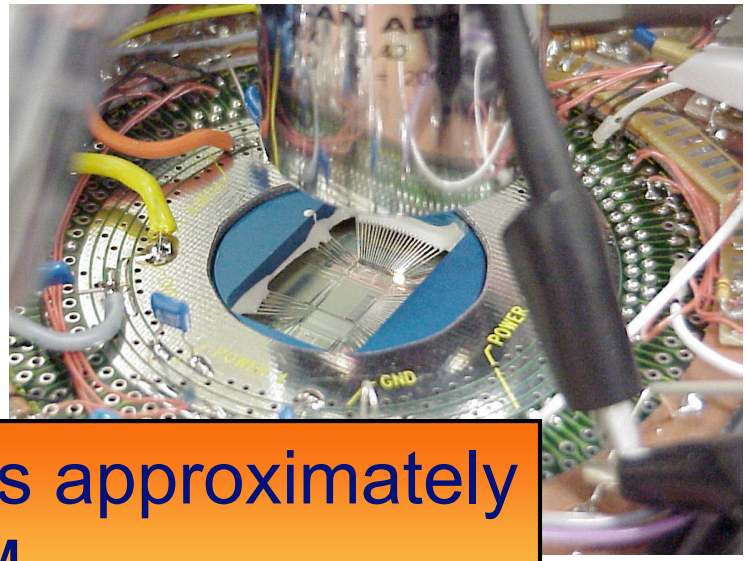
    Generate a consistent input assignment that "activates" the fault and propagates the difference to at least one output.



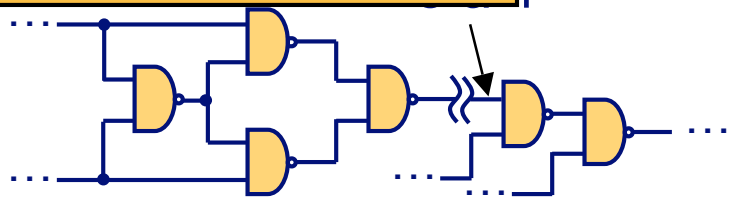
  - Heavy use of SAT-style methods to solve problem

Example: ATPG

- Automatic Test Pattern Generation is one of the oldest application of FM-kind methods
  - J. Paul Roth:
    "Diagnosis of Automata Failures:
    A Calculus and a Method"
    IBM J
  - Proble
    Generate a consistent input assignment that "activates" the fault and propagates the difference to at least one output.
  - Heavy use of SAT-style methods to solve problem

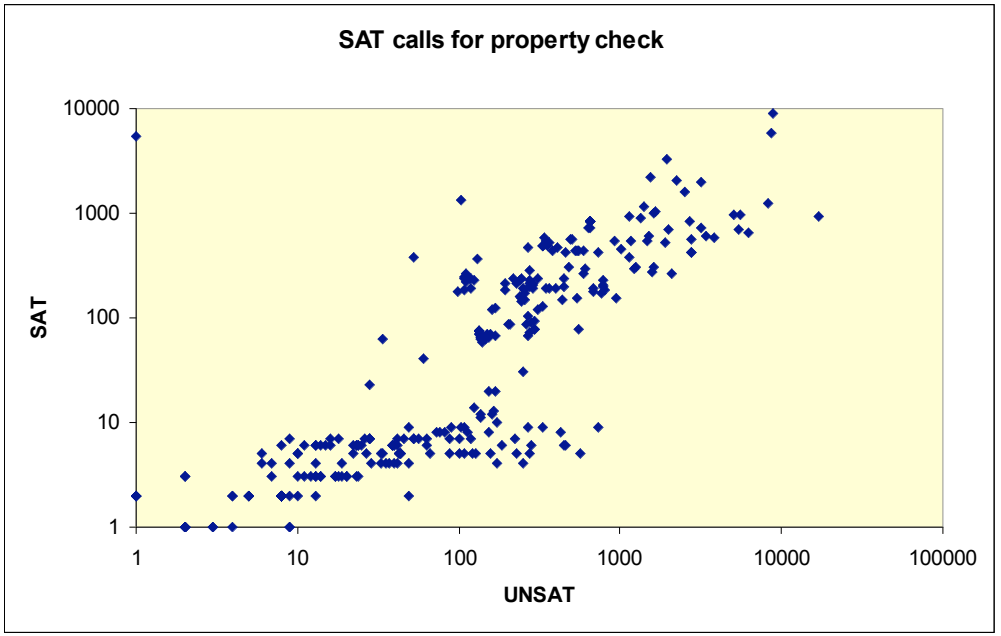


The test tool market is approximately $140M



Example: Property Checking

- After age of BDDs, SAT is being used in many core verification engines
  - Examples: BMC, CEGAR, Interpolation-based MC, …
  - Typical verification run includes large number of SAT queries

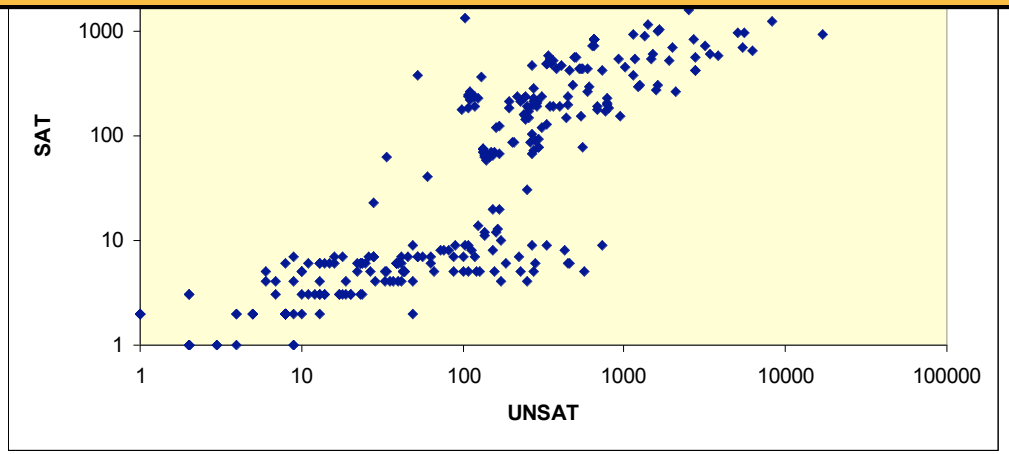


Source: Nina Amla, Cadence

Example: Property Checking

- After age of BDDs, SAT is being used in many core verification engines
  - Examples: BMC, CEGAR, Interpolation-based MC, …
  - Typical verification run includes large number of SAT queries

The FV market (EC + PC) is approximately $104M

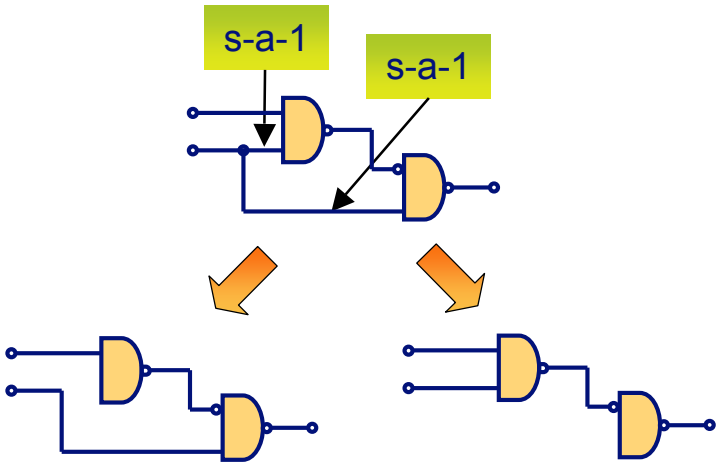


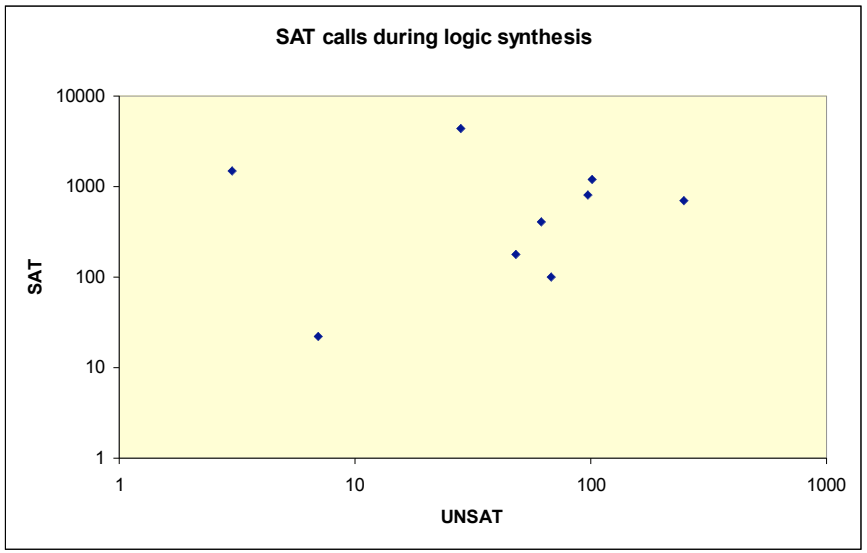
Source: Nina Amla, Cadence

kuehl@cadence.com

Example: Logic Synthesis

- Logic optimization using queries of form:
  "Is this change valid?"
  - Example: R. Dandapani, et al., "On the Design of Logic Networks with Redundancy and Testability Considerations,"
    IEEE Transactions on Computers,vol. c-23, No. 11, Nov., 1974.
  - Test whether "fault is untestable" ⇔ "connection can be removed"

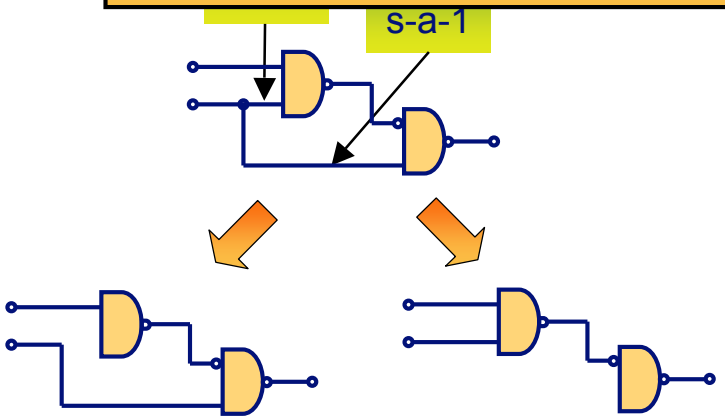


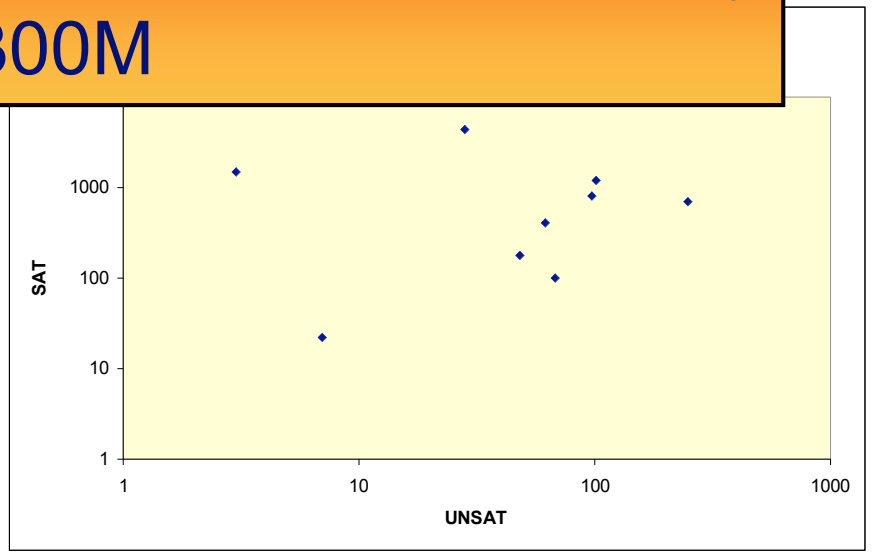


Source: Christoph Albrecht, Cadence

Example: Logic Synthesis

- Logic optimization using queries of form:
  "Is this change valid?"
  - Example: R. Dandapani, et al., "On the Design of Logic Networks with Redundancy and Testability Considerations,"
    IEEE Transactions on Computers,vol. c-23, No. 11, Nov., 1974.
  - **The logic synthesis market is approximately $300M**





Source: Christoph Albrecht, Cadence

Example: Multi-Domain Clock Scheduling

- Optimize clock distribution using multiple clocking domains
  - K. Ravindan, et al. "Multi-Domain Clock Skew Scheduling", ICCAD 2003
  - Model clock domain assignment as conditional graph edges
  - Clock-scheduling for fixed graph done by Bellman-Ford algorithm
  - "SMT"-style search performed to find optimal clock domain assignment and clock schedule
    - Including learning of "negative cycles"

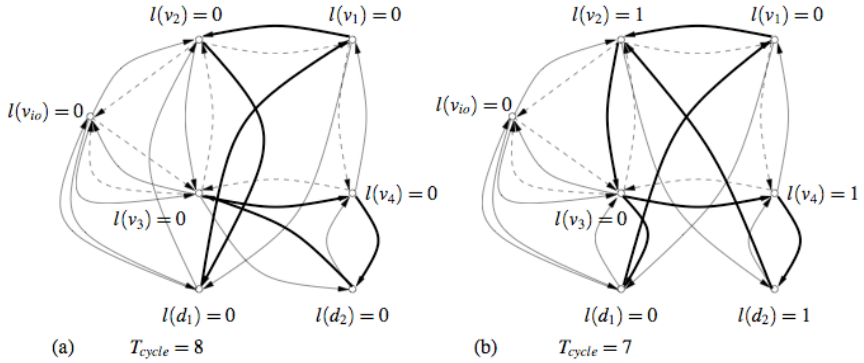


Figure 2: Two register-domain assignments for the circuit from Figure 1 optimized for two clocking domains: (a) 1. configuration: $\{x(v_1,d_1) = x(v_2,d_1) = x(v_{io},d_1) = 1, x(v_3,d_2) = x(v_4,d_2) = 1\}$; Critical cycles: (d_1,v_1,v_2), (d_2,v_3,v_4); $T_{cycle} = 8$, (b) 2. configuration: $\{x(v_1,d_1) = x(v_3,d_1) = x(v_{io},d_1) = 1, x(v_2,d_2) = x(v_4,d_2) = 1\}$; Critical cycles: (d_1,v_1,v_2,v_3), (d_2,v_2,v_3,v_4); $T_{cycle} = 7$.

What can we conclude so far?

- Q: Has the “FV” a real impact on the “CAD”?
- A: Absolutely yes and it will continue to have in many existing and new application areas.
  However, one should not limit “formal methods” to “formal property proof” only.

- Q: And what about “formal property proofs”?
- H: They will continue to be important in CAD but remain one of the many ingredients in an overall verification flow.
  - Challenges
    - Algorithmic complexity
    - Existence of specification
    - Correctness of specification!!!

kuehl@cadence.com

What are the FV opportunities?

- Circuit level:
  - Higher level specifications and synthesis to allow more abstract verification approaches (e.g. SMT based methods)
  - Combining statistical simulation (testing) with FV
  - Analog Mixed Signal (AMS) formal verification
  - Proof of not-purely-functional properties such as power, reliability, etc,

- Chip and System level:
  - Power will drive distributed architectures - further separating
    - Computation
    - Storage
    - Communication
  - “End of scaling” might drive increased unification of HW platforms
    - Big shift to software verification

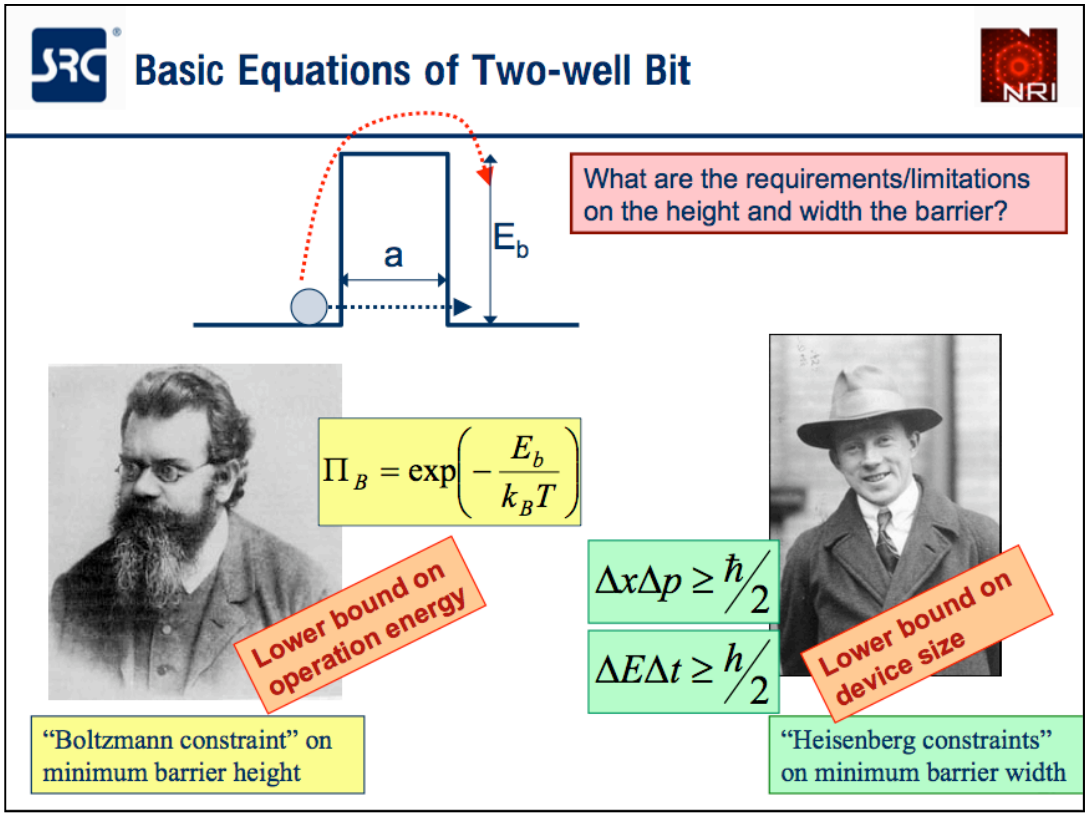
Example: Analog-mixed Signal FV

- Since the days of SPICE, circuit simulation is the vastly dominating vehicle to do analog design and verification
  - No, or very little “separation of concerns” has happened in AMS
    - In Digital: Use of synchronous implementation style combined with static timing analysis and formal equivalence checking allowed the use of cycle simulation on RTL for functional verification
    - W/o it, we would not be able to verify today’s chips
  - Can we have a more structures AMS verification flow?

In its inner guts, SPICE is also discretizing time, voltages, currents, etc.

Chip-Level

- Scaling is coming to an end and there is not much in “nanotechnologies” for computation
- Are standardized, distributed platforms the future?
  - Clear separation between:
    - Computation
    - Memory
    - Communication
- Opportunity for structured verification!

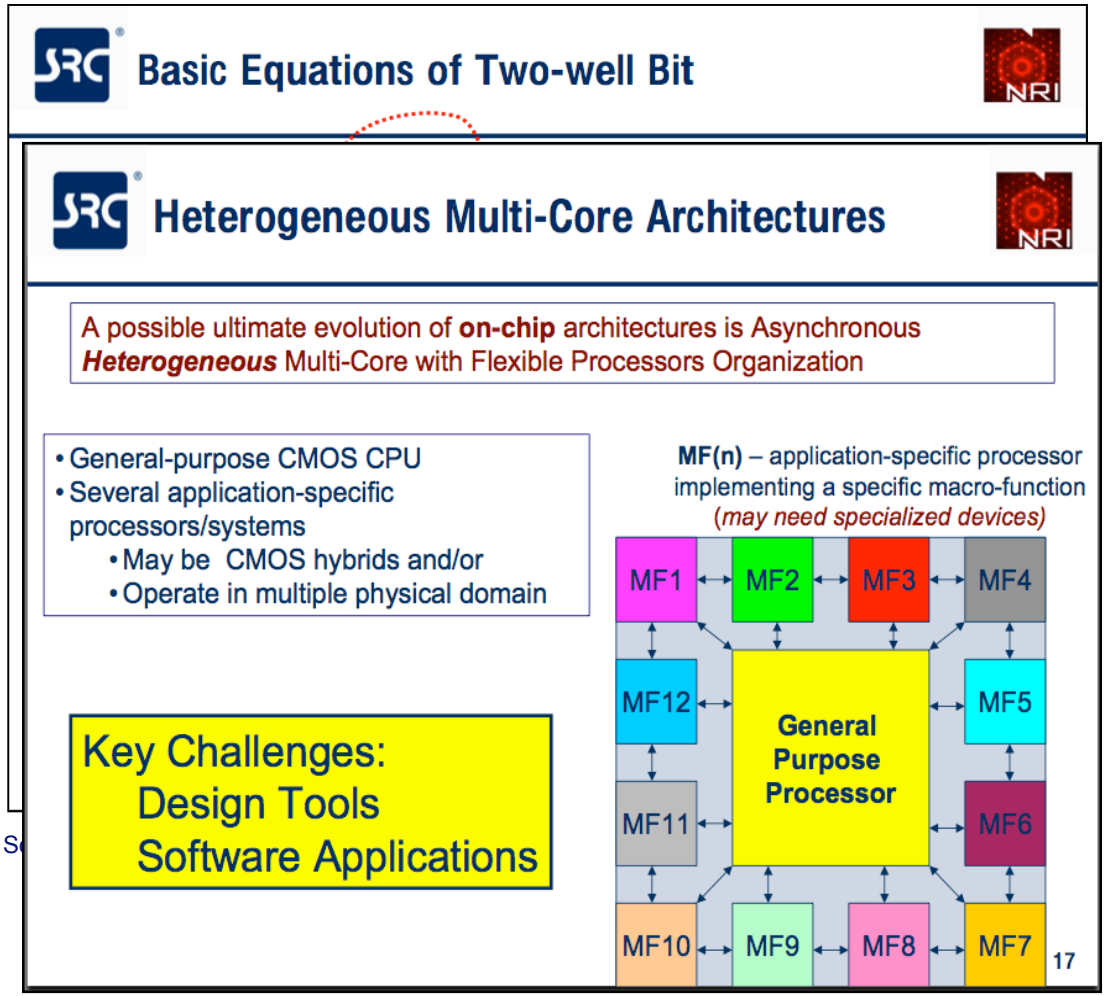


Source: Jeff Welser, IBM: Keynote – ICCAD, November 2007

kuehl@cadence.com

Chip-Level

- Scaling is coming to an end and there is not much in “nanotechnologies” for computation
- Are standardized, distributed platforms the future?
  - Clear separation between:
    - Computation
    - Memory
    - Communication
- Opportunity for structured verification!



Source: Jeff Welser, IBM: Keynote- ICCAD, November 2007

kuehl@cadence.com

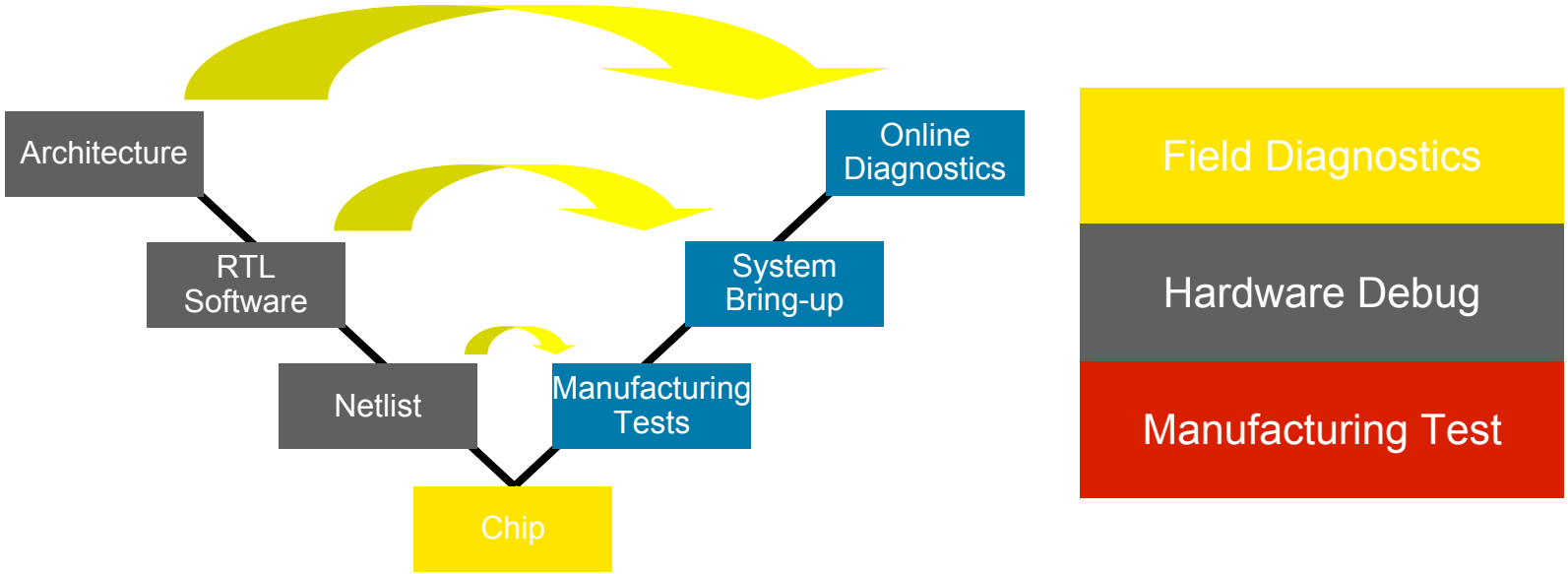
Opportunity or Nightmare?

- Asynchronous commutation adds additional level of non-determinism
- Dynamic power management (HW and SW controlled) will add another level of non-determinism
- Limited reliability of system components (not devices but chips, boards, boxes, communication infrastructure will add yet another level of non-determinism
- Time scales of computation will differ by several orders of magnitude requires rigorous abstraction
- ….

kuehl@cadence.com

Example: Post-silicon Debug + System Bringup

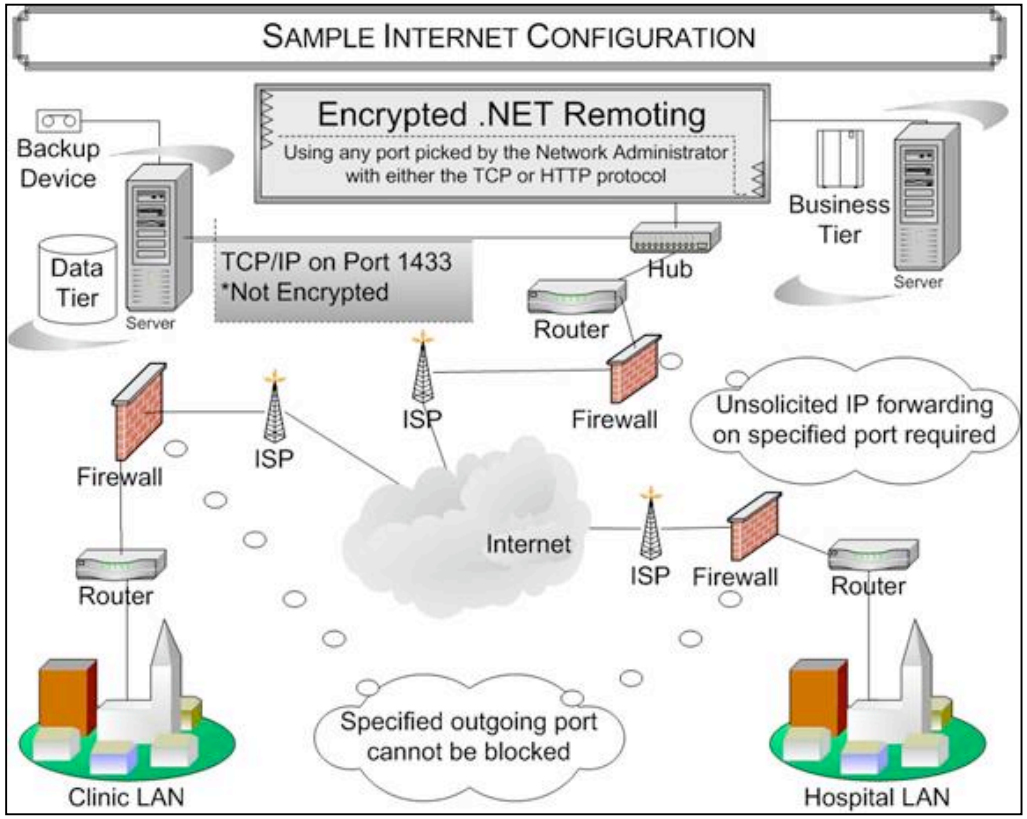
- Opportunity:
  - Cost of post-silicon debug, system bring-up, and in-field diagnosis has dramatically increased in past years
  - Limited observability and controllability on chip + limited reproducibility of asynchronous environment events make debug extremely challenging



kuehl@cadence.com

System-Level

- Systems are growing rapidly in complexity and heterogeneity
- We complain that there is no full spec for chips! Does anyone think there is one for this?
- Verification becomes much more than just ensuring that some spec is implemented!



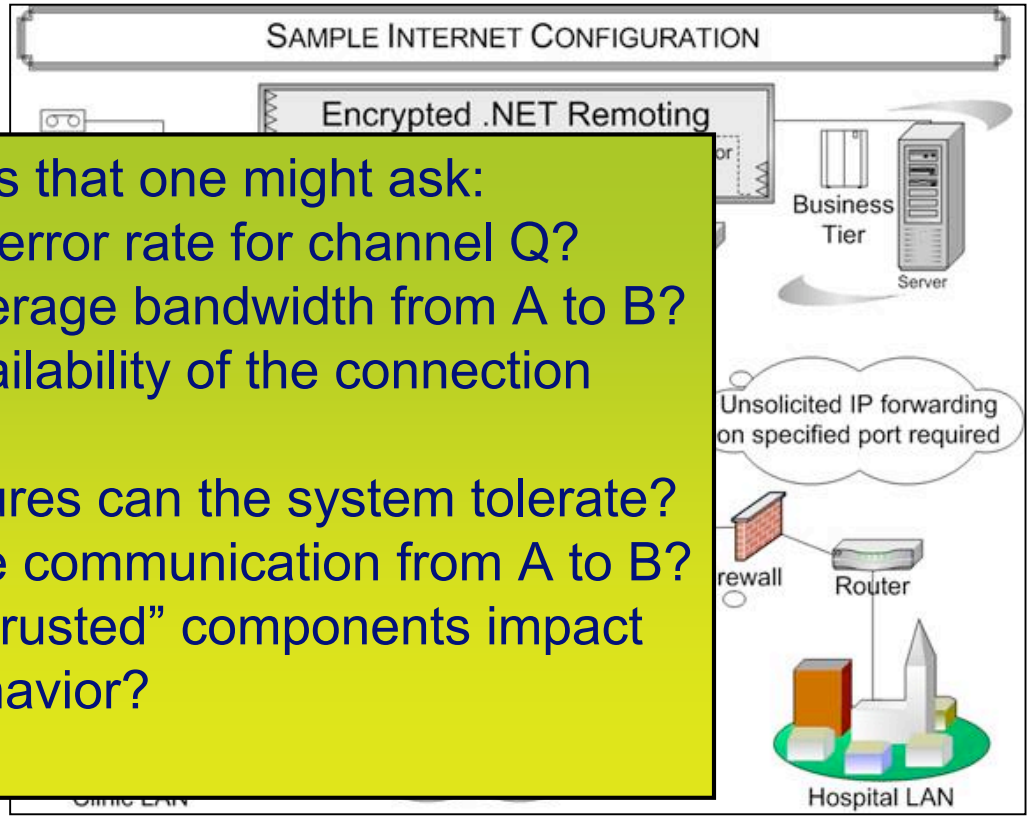
Source: http://hab.hrsa.gov/tools/v4_usersguide.htm

System-Level

- Systems are growing rapidly in complexity and heteroge
- We comp no full sp Does any is one for
- Verificati much mo ensuring that some spec is implemented!

Sample questions that one might ask:

- What is the bit error rate for channel Q?
- What is the average bandwidth from A to B?
- What is the availability of the connection from A to B?
- How many failures can the system tolerate?
- How safe is the communication from A to B?
- How could “untrusted” components impact the system behavior?
- …….



Source: http://hab.hrsa.gov/tools/v4_usersguide.htm

Do we need to change our thinking?

- FV has always asked:
  - "Does the implementation comply with the spec?"
- What if it is intractable to put an entire spec together?
  - Complexity of system
  - Ambiguity of standards
  - Intractability of checking for compliance when many suppliers provide parts
  - …
- What about two-part spec:
  - An incomplete spec for behavior we would like to see
  - An safety spec stating what components will not do for sure
    - "Burn up the box"
  - Can we check minimal behavior in worst case scenarios?

Thank You - Again

Questions/Comments?

kuehl@cadence.com