Formal Hardware Verification: Theory Meets Practice

Dr. Carl Seger
Senior Principal Engineer

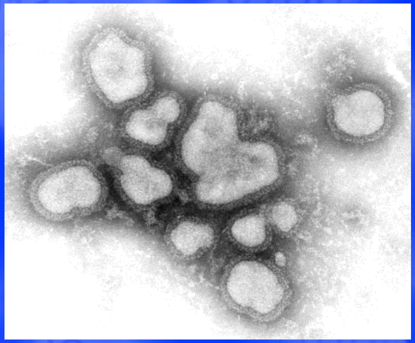
Tools, Flows and Method Group
Server Division

Intel Corp.

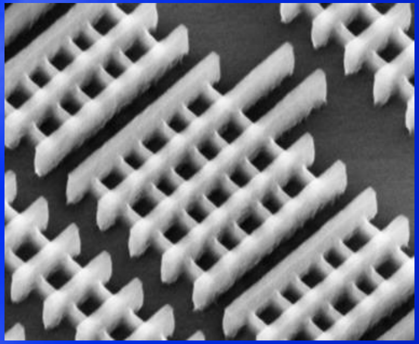
June 24, 2015

Quiz 1 – Small Numbers

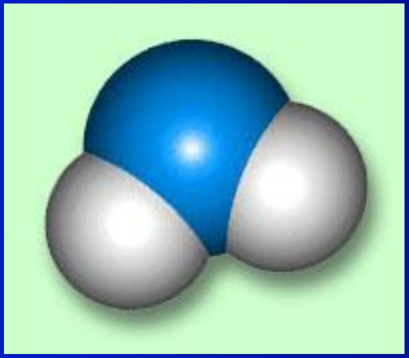
Order the following in order of size (smallest first)



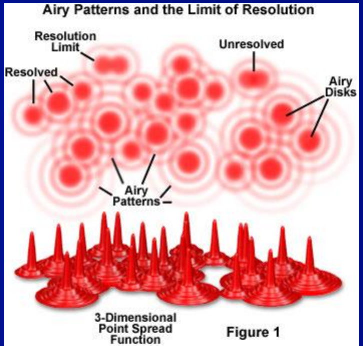
Influenza A virus



Transistor in microprocessor as of June 2015



Water molecule

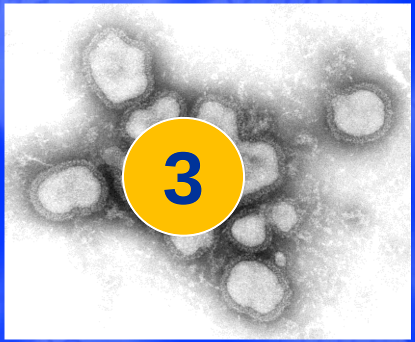

Airy Patterns and the Limit of Resolution
Resolution Limit
Unresolved
Resolved
Airy Disks
Airy Patterns
3-Dimensional Point Spread Function
Figure 1

Resolution of optical microscope

Answer Quiz 1

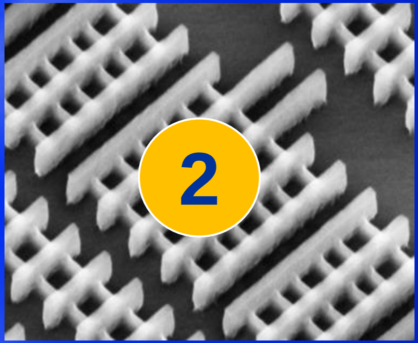
Order the following in order of size (smallest first)

~100nm



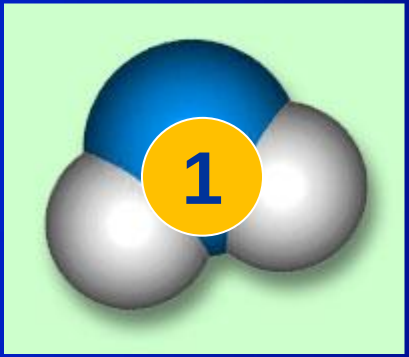
Influenza A virus

~14nm



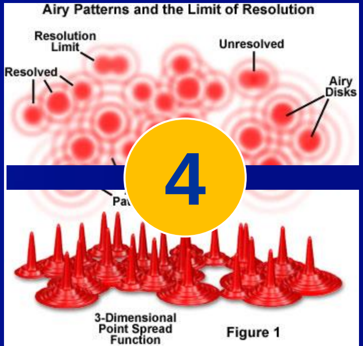
Transistor in microprocessor as of May 2014

~0.3nm



Water molecule

~300nm


Airy Patterns and the Limit of Resolution
Resolution Limit
Unresolved
Resolved
Airy Disks
Pa
4
3-Dimensional Point Spread Function
Figure 1

Resolution of optical microscope

Quiz 2 – Large Numbers

Order the following in order of size (largest first)

Number of light bulbs in the world

Number of atoms in the Empire State Building

Number of transistors in a 2014 cell phone

Number of patterns needed to simulate all possible inputs to one AVX instruction (two 256-bit inputs)

Answer Quiz 2

Order the following in order of size (largest first)

$\sim 10^{10}$

Number of light bulbs in the world

$\sim 10^{31}$

Number of atoms in the Empire State Building

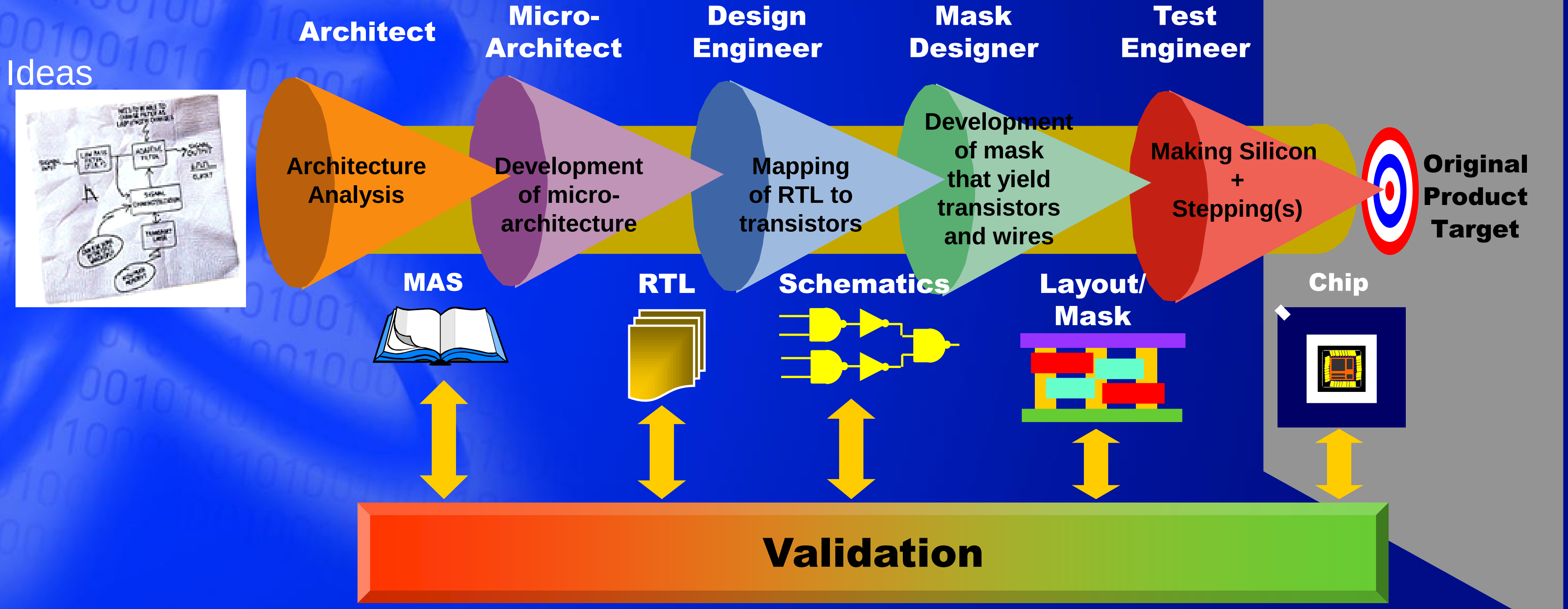
$\sim 10^{11}$

Number of transistors in a 2014 cell phone

$\sim 10^{154}$

Number of patterns needed to simulate all possible inputs to one AVX instruction (two 256-bit inputs)

The Design Process at 10,000 ft



Ideas

Architect

Architecture Analysis

Micro-Architect

Development of micro-architecture

Design Engineer

Mapping of RTL to transistors

Mask Designer

Development of mask that yield transistors and wires

Test Engineer

Making Silicon + Stepping(s)

Original Product Target

MAS

RTL

Schematics

Layout/ Mask

Chip

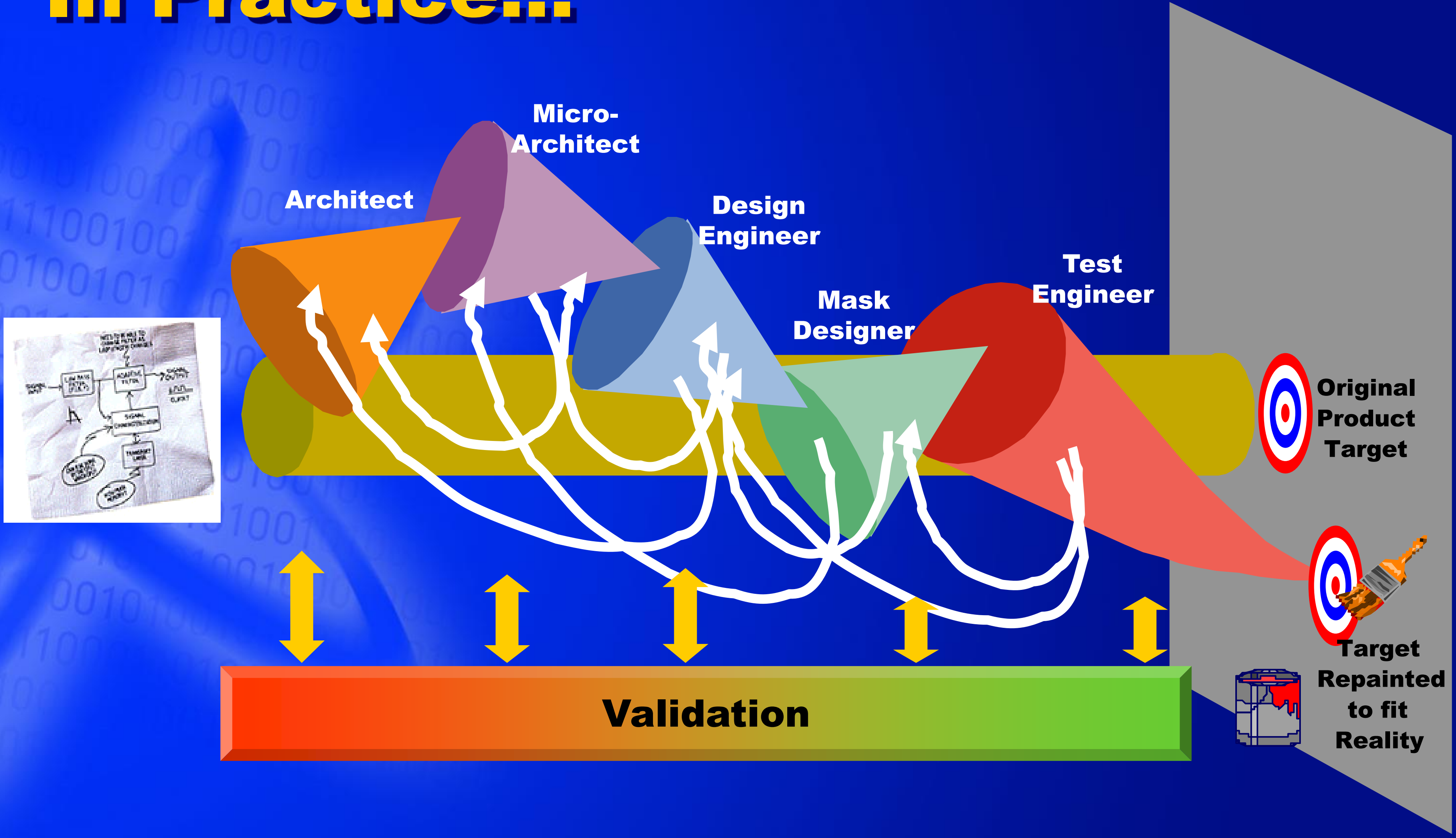
Validation

MAS: Micro-Architecture Specification

RTL: Register-Transfer Language

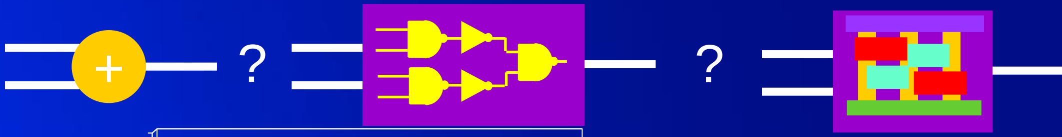
This is the theory…

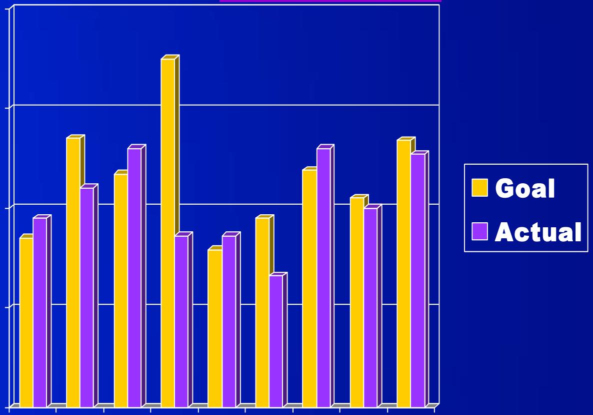
In Practice...

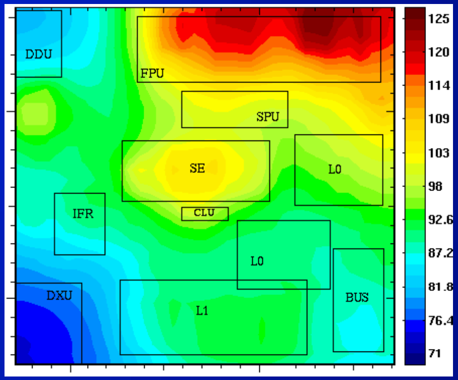

Micro-Architect
Architect
Design Engineer
Mask Designer
Test Engineer
Original Product Target
Target Repainted to fit Reality
Validation

What Needs to be Validated?

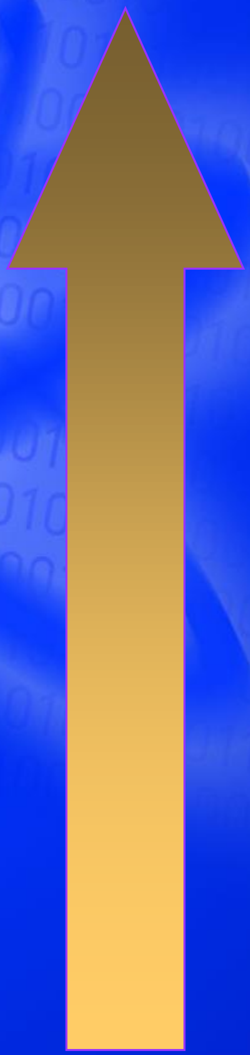
- Functionality
- Performance
- Power & Thermal
- Physical form
- Documentation
- Reliability
- Testing procedure
- …







Functional Validation Approaches

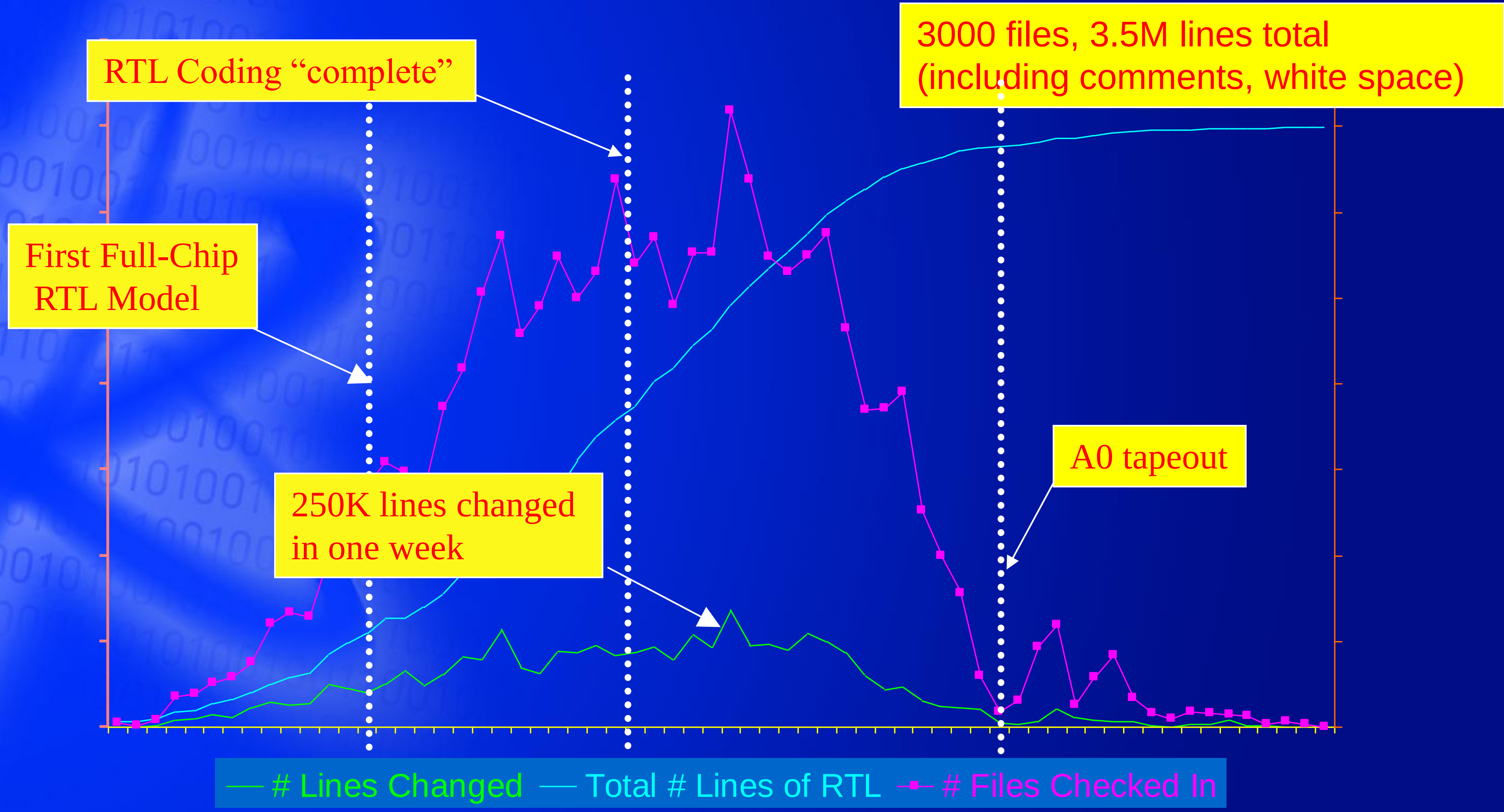


100 % Covered

Low % Covered

	Pro	Con
Formal Verification	• 100% coverage • Proves absence of bugs	• Requires special skills • Constrained by complexity
Directed Random Tests	• Targets areas most likely to be of concern • Greatly reduces cycle requirements • Develops strong uArch knowledge	• Requires strong uArch knowledge
Generic Random Tests	• After generator created, easy to write • Requires little uArch knowledge • Can create things no one would ever think of	• Requires almost ∞ cycles / time • Difficult / impossible to avoid broken features
Directed Tests	• Easy to write • Easy to understand • Easy to reuse	• Requires almost ∞ number of tests • Difficult to hit uArch conditions

RTL Changes Constantly



RTL Coding “complete”

3000 files, 3.5M lines total (including comments, white space)

First Full-Chip RTL Model

250K lines changed in one week

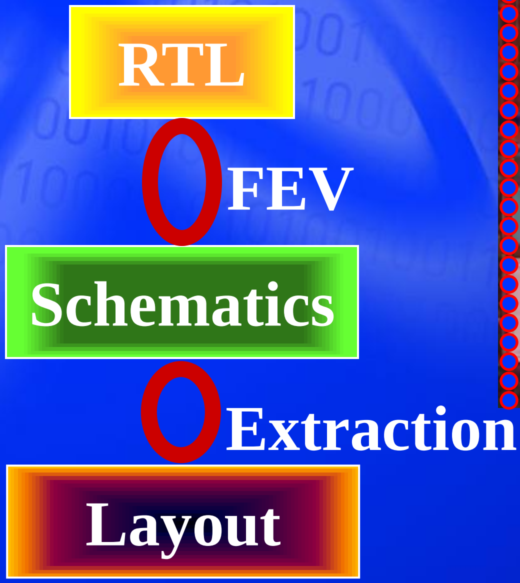
A0 tapeout

Validation focus

Functionality **Timing**

Formal Equivalence Verification

- Use of symbolic/algebraic methods to completely verify that a circuit implements a specification



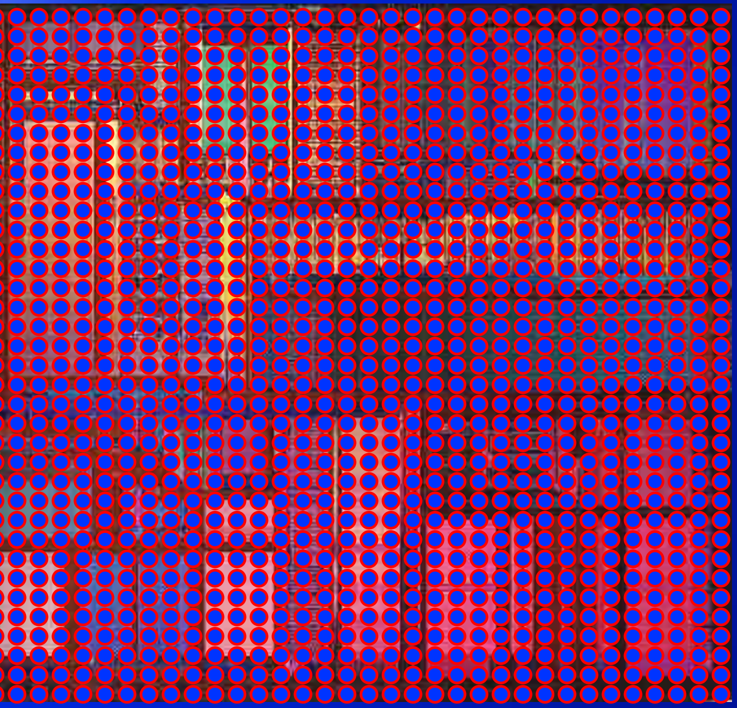
RTL

FEV

Schematics

Extraction

Layout



Today: 100% of a design is run through FEV before tape-out

Extremely successful application of math, logic and computer science in practical engineering!

Usability high enough that every design engineer is able to run the verification.

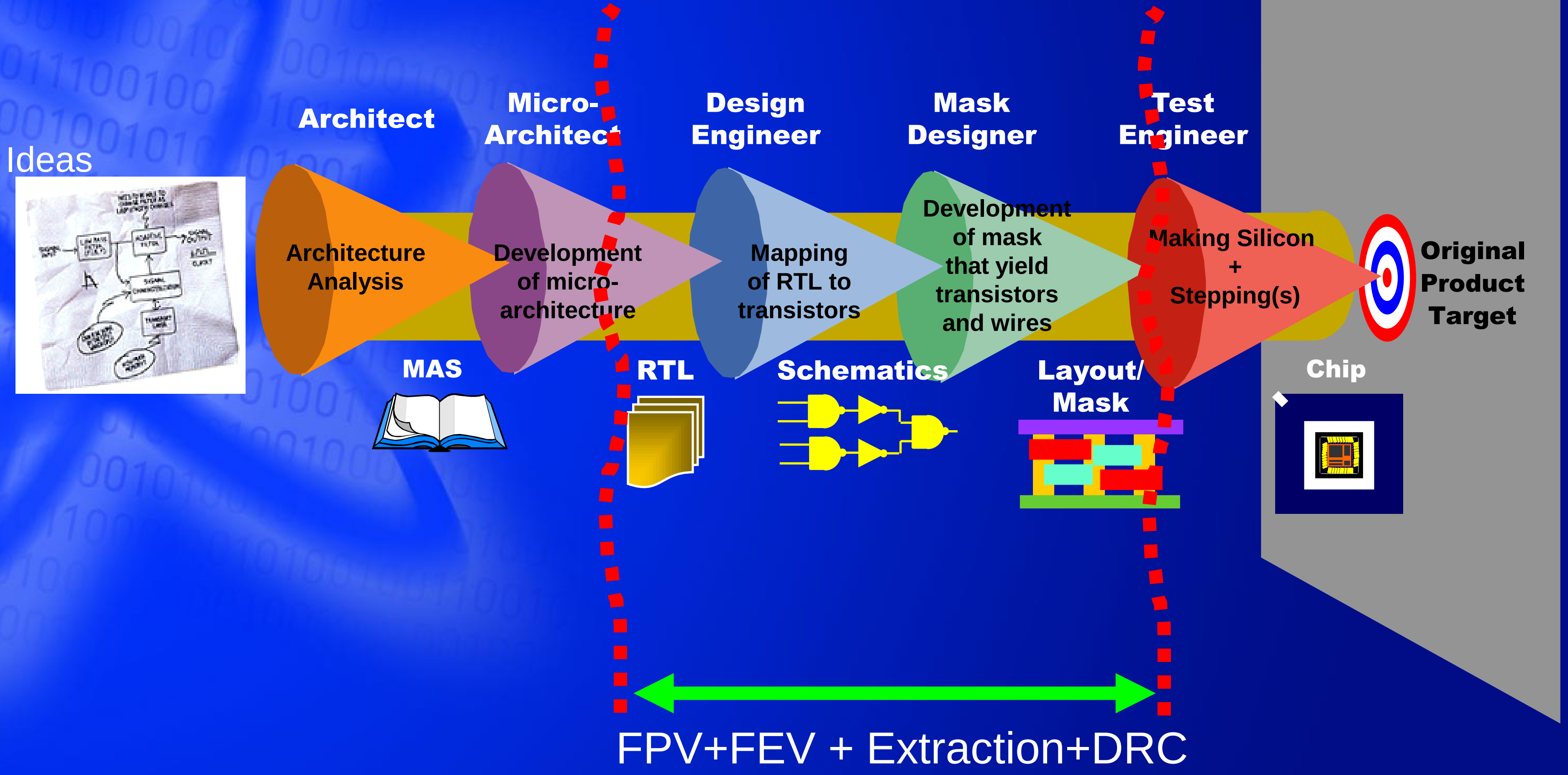
Formal Property Verification

- Symbolic Trajectory Evaluation (STE), a form of symbolic simulation, are today used to formally verify very large computation units/blocks
  - Complete formal property verification of all (>3,000) uops in the execution cluster of Intel processors is now routinely done
    - Includes all control, clock gating logic, test features etc. as well as the actual data-path computations
    - FPV is primary pre-Si verification for this unit
- Combining STE with theorem proving increases the quality of specification
  - Floating point spec is mathematical statement of IEEE standard
- Symbolic model checking is seeing more wide spread use
  - Early architecture exploration/validation
  - Control intensive designs
  - Design driven early exploration

Good News / Bad News

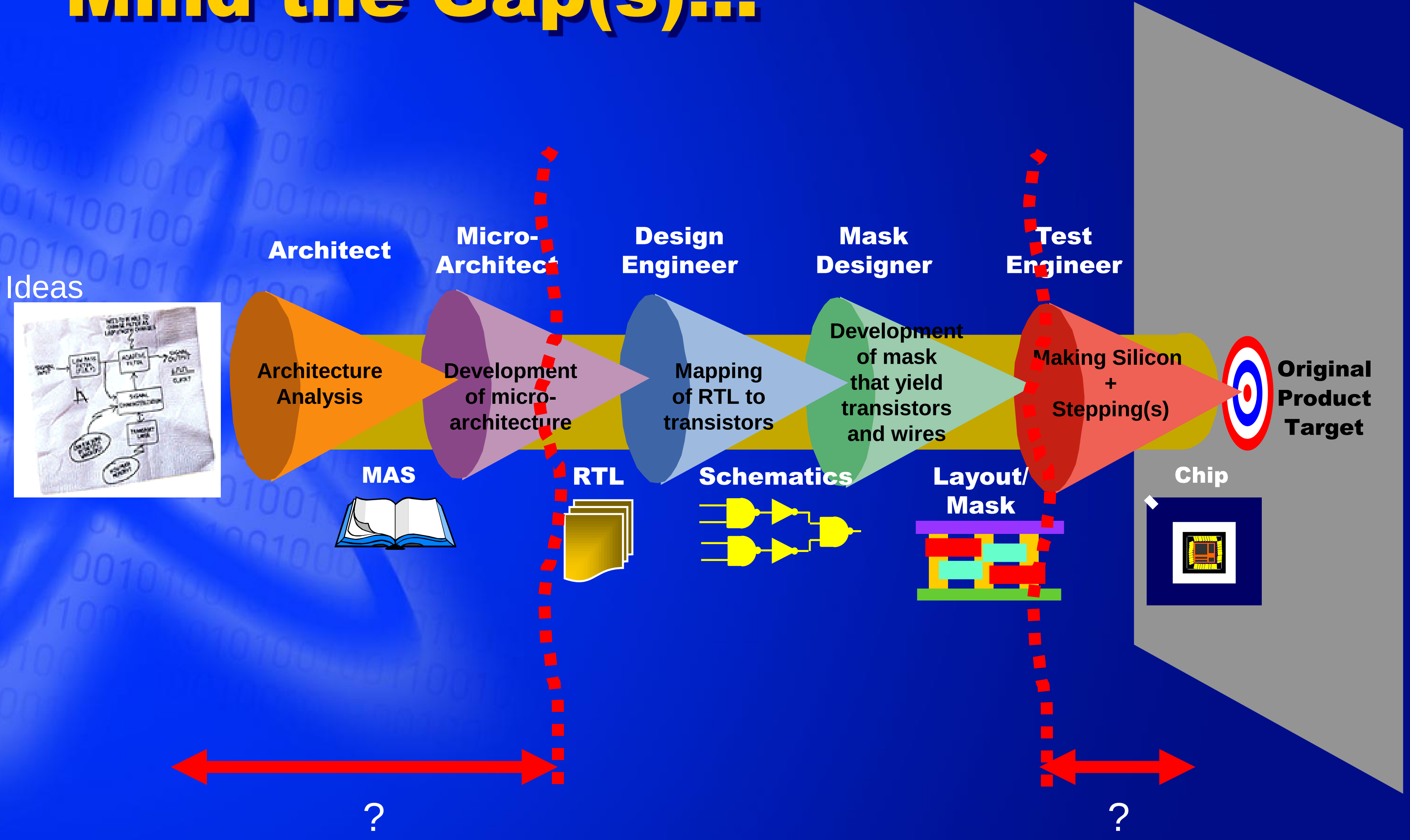
- Good news:
  - Formal verification can guarantee the correctness of extremely large and complex hardware
  - The verification programs allow continuous regression runs, thus preventing bugs from re-appearing
  - The verification specifications and verification scripts can often be re-used for new designs
- Bad news:
  - Difficult to capture control aspect accurately & robustly
  - Knowledge intensive activity to create initial specs and verification scripts
  - FV capacity not growing as fast as design size/complexity.
  - Structural verification decompositions are very fragile

Solid Formal Link with Good Return of the Investment



FPV+FEV + Extraction+DRC

Mind the Gap(s)…



Summary

- Formal HW Verification
  - Relies heavily on Computer Science research:
    - Finite state machines; “everything is an FSM”
    - Lattices & Galois connections
    - Data structures for representing very large circuits and Boolean functions
    - Advanced algorithms for symbolic state machine traversal, SAT solving, etc.
  - Is deployed widely in industry and is now usable by most designers.
  - Is likely to have even wider use as industry is converging on System-on-Chip designs with re-usable IP blocks
- How to leverage FV technology at higher level of abstraction and in mixed HW/SW system is a major research problem.

Finding a Needle in a Haystack vs Finding a HW bug

vs.

Finding a single pair of values for a double precision floating point divide operation that fails.

For probability to be the same, how big should the haystack be? (Assume half-sphere haystack)

Answer: Radius ~550 light years!