# Practical Methods in Coverage-Oriented Verification of the Merom Microprocessor

Alon Gluska

IBM Verification conference, Oct 2006

# Agenda

- Merom microprocessor and its verification
- Coverage-Driven Verification (CDV) and its drawbacks
- Merom Coverage-Oriented Verification
- Practical methods in Merom coverage
- Results and summary

# Agenda

- Merom microprocessor and its verification
- Coverage-Driven Verification (CDV) and its drawbacks
- Merom Coverage-Oriented Verification
- Practical methods in Merom coverage
- Results and summary

# Merom Microprocessor

- Intel's Core$^{TM}$ Duo 2 Mircroarchitecture
  - New foundation for desktop, mobile and server multi-core processors
  - Designed by Mobile Microprocessor team in Haifa, Israel
- Energy-efficient performance
  - 64-bit architecture, wider pipeline, instruction fusion, improved vector parallelization, incremental CPU power states
- Significantly higher performance and lower power than competition
- Dual and Quad core, with shared second-level caches
- Design and shipment ahead of schedule

# Merom Verification at a Glance

- Modular verification
  - Design & verification divided into 6 clusters
  - Most verification, and most coverage, at the cluster level
    - Cluster level testbenches
    - Scalable checking, monitoring, functional coverage
  - Verification at Full-Chip level
    - Architectural compliance and coverage
    - Compensate for cluster interface weaknesses
  - Verification at platform level
    - Co-simulation with chipset
- Formal verification
  - Applied selectively according to complexity and capacity
  - Reduced the need for coverage-based quality indicators

# Agenda

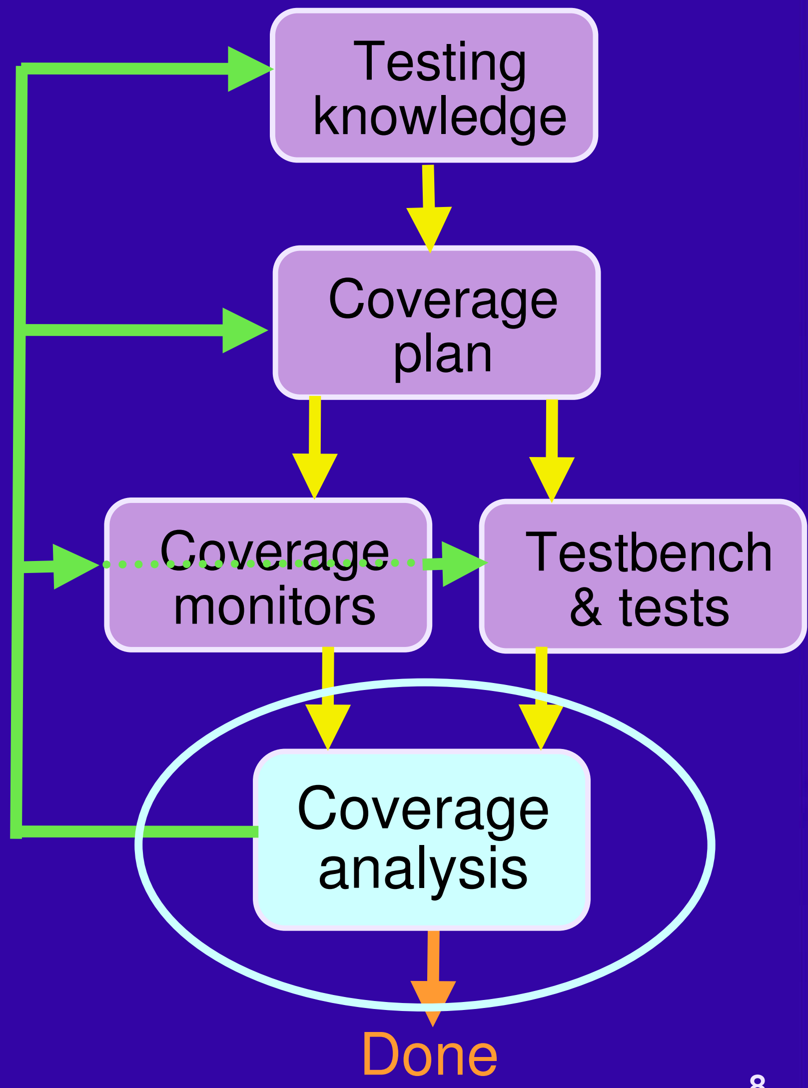
- Merom microprocessor and its verification
- Coverage-Driven Verification (CDV) and its drawbacks
- Merom Coverage-Oriented Verification
- Practical methods in Merom coverage
- Results and summary

# Functional coverage

- Widely known as a means of measuring the quality of verification
- Derived manually from logic specifications
- Systematically create a comprehensive list of conditions
  - Verify each is hit during simulation
- Steer test generation towards holes
- Quantitative way of measuring the progress of verification
- A means for quality, not a goal
  - Bugs may exist outside the coverage space

# Coverage-Driven Verification (CDV)

- Coverage is the primary driver of verification
  - Adopted from day one
  - Main metric for completion
- Drawbacks in early adoption
  - Focus should be on finding bugs
  - Lack of detailed knowledge
  - Instability of design and uArch spec
  - Incompleteness of coverage space



# Agenda

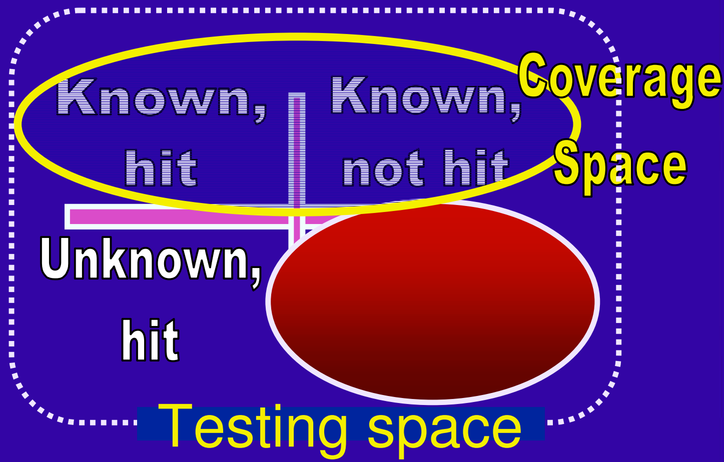
- Merom microprocessor and its verification
- Coverage-Driven Verification (CDV) and its drawbacks
- Merom Coverage-Oriented Verification
- Practical methods in Merom coverage
- Results and summary

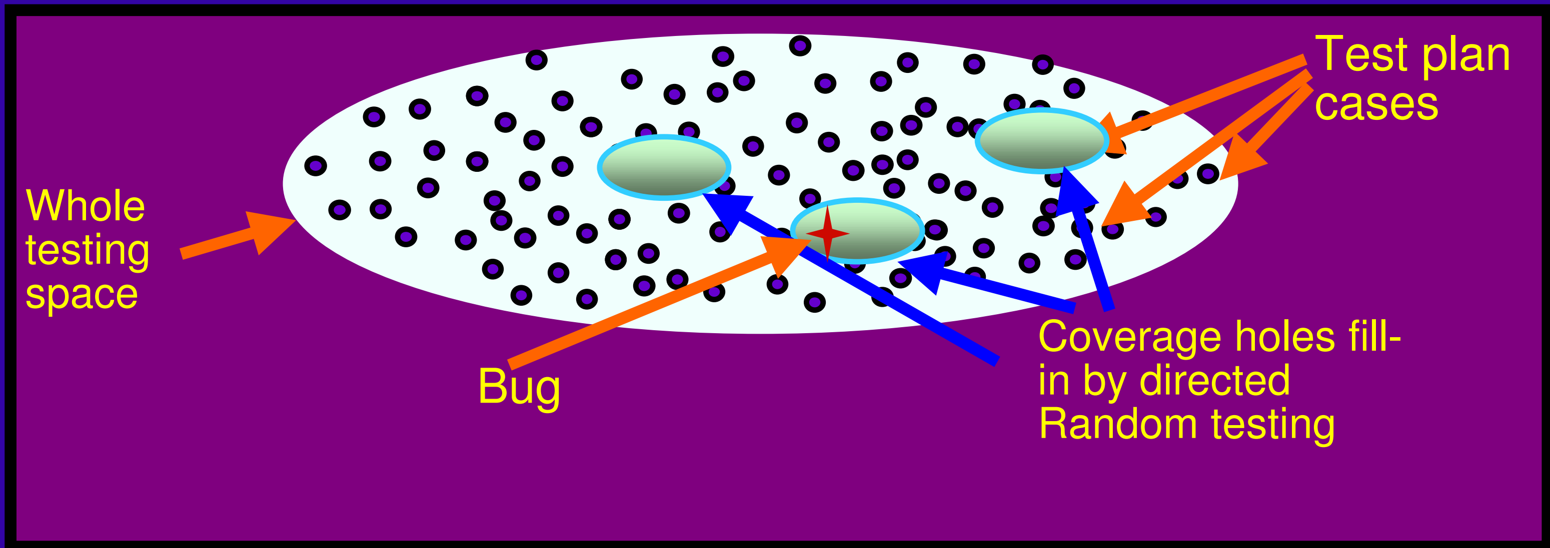
# Merom Coverage-Oriented Verification

- Practical trade-offs for highest return on investment
  - Reduced scope and target high risk areas
- Delayed applying coverage to the latter stage
  - Stage I: basic cleanup, kept full-chip model functional
  - Stage II: development and analysis of functional coverage
- Overcomes major drawbacks of CDV
  - Basic bugs have already been flushed out
  - Design and uArch specs have stabilized
  - Engineers have acquired fundamental knowledge

# Coverage Guidelines (1)

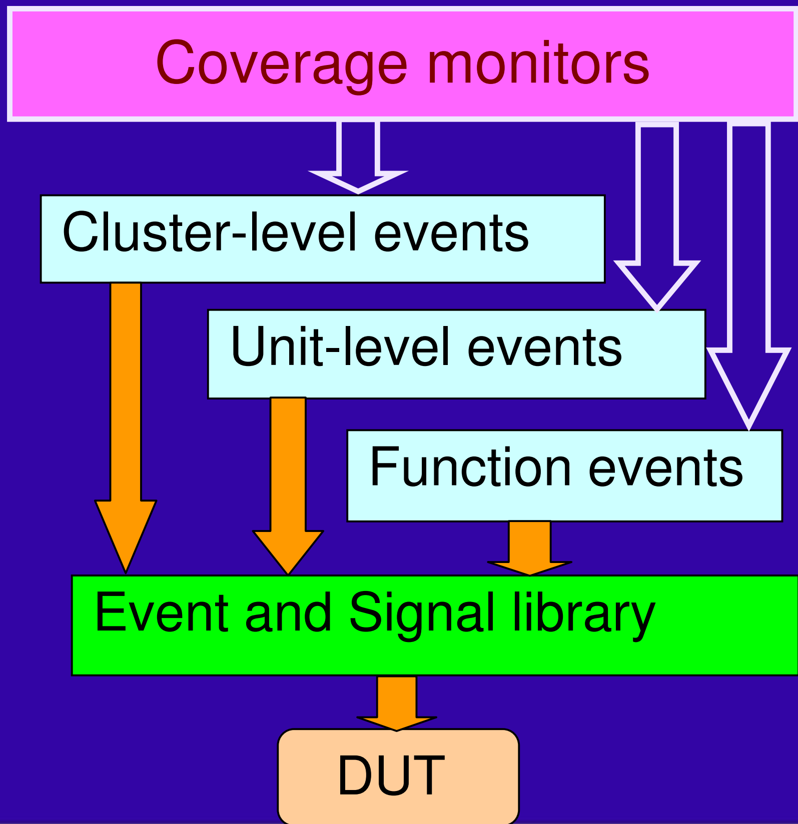
- Invest in random capabilities to fill in testing space
  - Random testing enables hitting Unknown cases



# Coverage Guidelines (2)

- Write coverage-friendly test plans
  - Easy and accurate translation into coverage monitors
  - Formal definition of coverage cases
- Develop coverage hierarchically
  - Higher abstraction of coverage monitors
  - Facilitates maintenance and reuse of lower level events



# Agenda

- Merom microprocessor and its verification
- Coverage-Driven Verification (CDV) and its drawbacks
- Merom Coverage-Oriented Verification
- Practical methods in Merom coverage
- Results and summary

# Merom Coverage Methods (1)

- Reduced coverage space based on prioritization
  - Complexity and risk
  - Controllability: reach-ability from testbench boundaries
  - Intensity of testing
  - Use of other techniques
  - Implemented only high-priority coverage
    - Major reduction in coverage space, low impact on quality
    - ~60% of test plan dropped from coverage space

# Merom Coverage Methods (2)

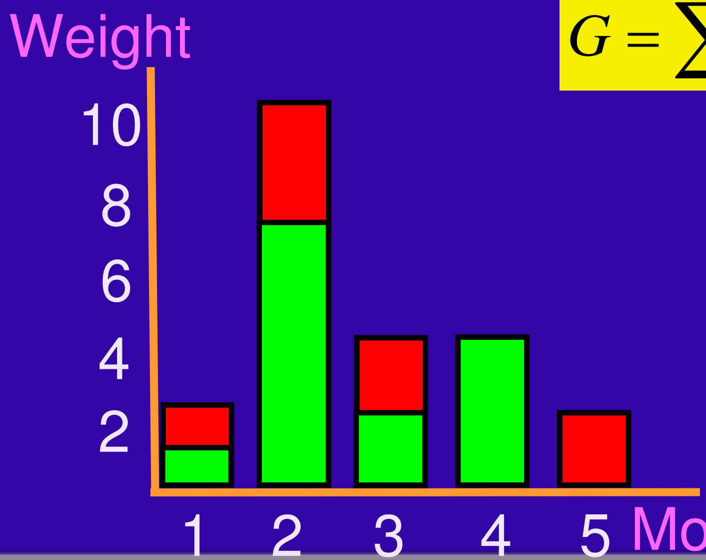
- Frequency coverage of Basic Events
  - Statistical approach to maintain balance between events
  - Simple, tool supported approach
- Automatic toggle coverage for Functional Boundaries
  - Simple, automatically generated for interface signals
- Coverage for for Clock Gating logic
  - Simple monitors, automatically generated from HDL
- Reduced space by merge of similar events
  - Merged coverage from identical components (e.g. decoders)

# Merom Coverage Methods (3)

- On-going use of fresh coverage data
  - Sliding Window – merge coverage from last 4 weeks
  - Eliminated use of stale data or need for resets
- Targets and weights defined per monitor
  - Used for hierarchical indicators
  - Visually reflected priority of coverage holes

$$G_i = \frac{e_i}{E_i} \frac{\min(p_i, P_i)}{P_i}$$

$$G = \sum W_i G_i$$

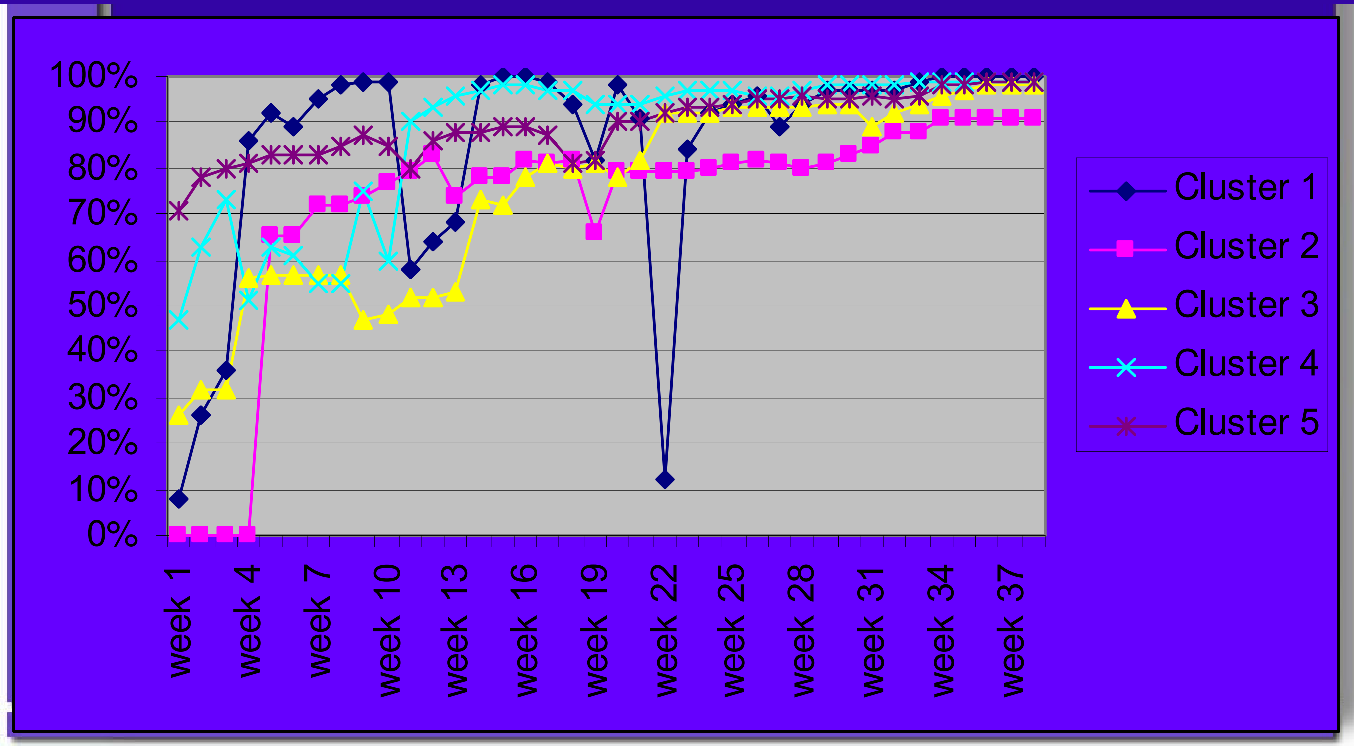


# Agenda

- Merom microprocessor and its verification
- Coverage-Driven Verification (CDV) and its drawbacks
- Merom Coverage-Oriented Verification
- Practical methods in Merom coverage
- Results and summary

# Coverage Tracking

- Coverage indicators climbed quickly
- Coverage drops when major changes are made to monitors



# Results

- Bugs found by coverage activities
  - ~80 bugs, ~8% of bugs in relevant period
    - 42 directly, not uniformly distributed
    - At least similar number found indirectly
      - Improved knowledge, fixed flaws in tests and testbenches
  - Most bugs involved temporal behavior or multi-cluster
  - >100 bugs in verification infrastructure
- Coverage perceived very important by engineers
  - Enforced learning of low-level details
  - Contributed to quality of testing

# Summary

- Applied Coverage Oriented Verification
  - Practical trade-offs, used in the latter stage of the project
- Improved random testing to fill-in coverage holes
- Multiple techniques to improve effectiveness
  - Prioritized coverage
  - Automatic monitors for logic boundaries
  - Automatic merge of similar events
  - Grading and visualization of holes
- Bugs detected by coverage activities
  - ~8% of RTL bugs, significant testbench enhancements
- Coverage contributed to very successful verification
  - Relatively few escapes found in silicon
  - Enabled pull-in of tape-out and production