Coverage Metrics for Functional Validation of Hardware Designs

Serdar Tasiran, Kurt Keutzer

IEEE, Design & Test of Computers, 2001

Presenter: Guang-Pau Lin
What’s the problem?

- What can ensure optimal use of simulation resources, measure the completeness of validation, and direct simulations toward unexplored areas of the design?

Coverage Metrics!
Outline

- Background
- Classification of coverage metrics
  - Code coverage
  - Metrics based on circuit structure
  - Metrics defined on FSM
  - Functional coverage
- Conclusions
Background

Write function spec.

Write technical spec.

Develop area/timing/power constraints

Write RTL
Run lint

Develop testbench

Synthesize

Simulate

Measure verification coverage

Write creation guide
Pass - ready for integration

Meet all specified constraints

Not enough
Background (Cont.)

Coverage Metrics

- Code coverage
- Circuit structure
- Metrics on FSM
- Functional coverage
Code coverage

- Derived from metrics used in S/W testing
- Identify which structure in the HDL code to exercise during simulation
  - Classify into two categories:
    - A sequence of lines with no control branches
    - Constitute the branching points
Example of code coverage

```verilog
case (oneHot)
    3'b001:  z <= a;
    3'b010:  z <= b;
    3'b100:  z <= c;
    default:  z <= 1’ bx;
endcase
if (a | b)
    d = d1 + d2 ;
else
    d = d1 - d2 ;
end
end
```

Path coverage的問題

三種case都要測到

這兩行都要模擬到
Strengths and weaknesses

- **Strengths:**
  - Little overhead
  - Easy to interpret coverage results

- **Weakness:**
  - Complete code coverage is a minimum requirement
  - 「Necessary but not sufficient」
  - No concurrency
Background (Cont.)

Coverage Metrics

Code coverage
Circuit structure
Metrics on FSM
Functional coverage
Circuit structure

- Identify physical portions of the circuit that are not exercised
- Separate circuits into data path and control path
  - In the data path:
    • Notice registers (initial, load, read, all paths from register to register)
    • Notice counter (reset, min value, max value)
  - In the control path:
    • Exercise all combinations of assignments to the signals between the two circuit parts
Example of circuit structure metric
Strengths and weaknesses

- **Strengths:**
  - Lower bound on the amount of simulation (vs. complete code coverage)
  - Easy to interpret coverage results

- **Weakness:**
  - Exercise certain structure or combinations of signals might not be possible
  - Eliminate false negatives is challenge
  - Sequential behavior detect is limited
Background (Cont.)

Coverage Metrics

- Code coverage
- Circuit structure
- Metrics on FSM
- Functional coverage
Metrics on FSM

- These metrics require state, transition, limited path coverage on a FSM
  - Find out primary control state
  - Choose acceptable path length

- Increasing the amount of detail in the FSMs increases the coverage metric’s accuracy but makes interpreting the coverage data more difficult
Example of metrics on FSM
Strengths and weaknesses

**Strengths:**
- Can detect many difficult-to-find bugs
- Should used to test certain sequential behavior

**Weakness:**
- Difficult to design a compromising abstract FSM
- Some rare bugs may be overlooked
- Maintain it as the design evolves takes big effort
Coverage Metrics

- Code coverage
- Circuit structure
- Metrics on FSM
- Functional coverage
Functional coverage

- Refer directly to the computation performed rather than its structure.
  - Each scenario and functionality fragment must be exercised
  - Monitor during simulation

- **Snapshot tasks** and **Temporal tasks**
  - **State** in the abstract machine specifies a snapshot task
  - **Path** constitutes a temporal task
Strengths and weaknesses

**Strengths:**
- Reusable for family design
- Catch more of the difficult-to-find bugs than most other approaches

**Weakness:**
- Designer must thoroughly understand the design to define effective metrics
- The effort involved in generating tests is daunting
  - Overcome by developing test pattern while design
- Weak for directing test generation toward unexpected corner cases
Observability of simulator

\[
i = j + k; \\
x = c * i; \\
\text{if ( } a > 0 \text{ )} \\
\quad m = x; \\
\text{else} \\
\quad m = 0; \\
\]

如果 \( i \) 的值錯了，而剛好 \( x = 0 \) 或 \( a < 0 \)，則 \( m = 0 \)，不管此時 \( i \) 的值正確與否。
Conclusion

- If you want only HDL code verification, choose code coverage.
- If you want H/W verification and hate overhead, choose circuit structure metrics.
- If you want to pay attention to sequential behavior verification, try FSM metrics.
- If the entry is a family design, and you have a complete spec and function test pattern, use functional coverage metrics.
- Above all, combined them is commended.