Design Verification

Lecture 21 - Simulation-Based Verification III

1. Branch coverage makes sure each branch of a condition is taken
   $\longmapsto$ However, there may exist multiple ways to make the condition evaluate to true (false)
   $\longmapsto$ need a way to cover exercising of conditions more thoroughly
   $\longmapsto$ thus, even if all branch directions taken, and all statements have been exercised, and that not all paths can be exercised, we'll still be missing many possible control flow paths that are critical to exercising the code

2. If the condition on the branch depends on only 1 variable, branch coverage sufficient. However, if condition depends on multiple expressions, we need to consider more possibilities to fully exercise the conditional node

3. Modified Condition Coverage (MCC)

   - keep a table of exercised conditions
   - keep also the excitation propagation of conditions to registers
   - problem: total number of conditions exponential and could be large

   Example 1

4. Extended Condition Coverage (ECC)

   - Instead of accounting all possible condition combinations, just track all variables in condition expression
   - the number of test cases less than exponential
     $\longmapsto$ if condition a simple conjunction of n variables, then there is only $n+1$ test cases
     $\longmapsto$ if condition contains disjunction involving n variables, then there are $O(n)$ test cases
   - for each variable in the condition, need to identify condition pairs

   Example 2

5. Enhancing coverage

   - Although ECC can cover all branching conditions, it does not necessary take into account if the condition expression itself is erroneous
   - In other words, do the ECC test cases cover missing variable in condition, additional variable in condition, etc?

   Example 3

Example 3 (continued)

6. Add test cases to cover more design errors
   - which additional test cases to add such that most uncovered cases will be covered?

Example 4

7. Recap: ECC coverage measures to be reasonable
   - not as low as path coverage - too pessimistic
   - not as high as statement+branch coverage - too optimistic
   - want to be a good tracking measure for true error coverage
8. Implications on testing software / software-like code
   - testing software in hardware-software co-design environment
   - checking specification written in C/C-like program
   - generate test cases directly from spec?
   - want: high statement + branch coverage
   - want: observability included in coverage metric
   - want: reasonable path coverage
   - problem: pointers hard to track
     - pointer is a variable that holds the address of another variable in mem
     - an error in pointer value means it points to a different location
     - result: the value read by pointer becomes incorrect

Example 5

Example 6

9. Cross-product of events
    - Generate a list of monitored signals/buses/variables
    - Construct a cross-product of all monitored events
    - Goal: increase the coverage of as many combinations of events as possible.
    - : Need: feedback control mechanism to direct the generation of new stimuli to exercise those combinations that haven't occurred.
        - Bayesian networks
        - Markov models
        - These are used to correlate coverage of events with characteristics in the input stimuli