COTEST

Gert Jervan
ESLAB/LiU

Position Statement

- From contract - what are the goals
- Workpackage by workpackage
- Vision and goals
- Current status and work
- Further goals
- Open issues

Main goal - testability improvements

- Design modifications
- Test generation
- Testability metrics issue
- Goal first. Step-by-step how to reach there...

Project Summary

- COTEST - Testability support in a Co-design Environment
- The main goal: To provide support in test issues at the system level and in the hardware/software co-design environment

Workplan

- WP1 : Benchmark identification
- WP2 : Test sequence generation (PdT)
- WP3 : Test-oriented system modifications (LiU)
- WP4 : Final evaluation of project results

Test-oriented System Modifications

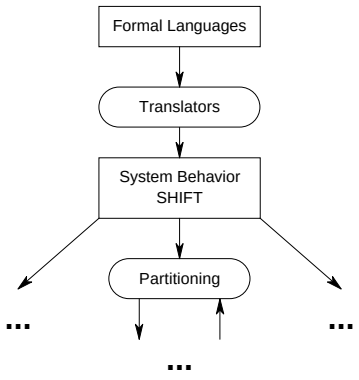
- We need to have testability metric (fault model)!!!
- Testability analysis to identify hard to test parts
- DFT strategy selection and modifications
  - Modification of the existing modules
  - Insertion of additional (test oriented) modules
- Evaluation of the results

Previous Work in the Area

- Modelling of the systems at the behavioral level using Decision Diagrams (DD)
- Test generation at the behavioral level targeting code coverage metric's and mutation fault model

Modeling – Co-design Environment

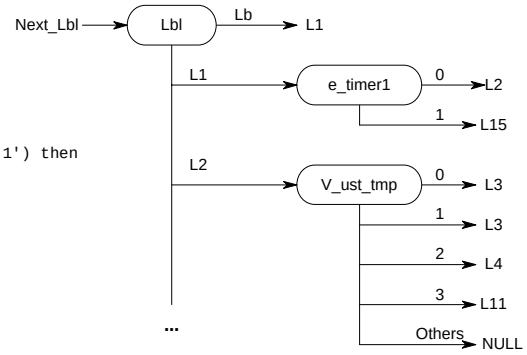
- **In POLIS - system is represented as a network of interacting Co-design Finite State Machines (CFSMs)**
- **Test environment should have the same view**
- **Translation of the codesign model into the DD-s**



Modeling – DDs

```
case Lbl is
when LB =>
      v_uv_0 <= v_uv_0_tmp;
      v_uv_1 <= v_uv_1_tmp;
      v_ucount_2 <= v_ucount_2_tmp;
      v_ucount_3 <= v_ucount_3_tmp;

-- sample input events
      e_timer1:= e_timer1_start;
      e_msec_tmp := e_msec_to_z_timer_0;
      Next_Lbl := L1;
when L1 =>
      if (e_timer1_e_start_timer_tmp = '1') then
            Next_Lbl := L15;
      else
            Next_Lbl := L2;
      end if;
when L2 =>
      case v_ust_tmp is
      when 0 =>
            Next_Lbl := L3;
      when 1 =>
            Next_Lbl := L3;
      when 2 =>
            Next_Lbl := L4;
      when 3 =>
            Next_Lbl := L11;
      when others => null;
      end case;
```



Testability Evaluation

- Need for an appropriate fault model - so far an open issue...
  - Code coverage metric's
  - Observability based metric's
  - New “supermetric’s”
    - Bit-coverage metric (statement, branch, condition coverage and partial path coverage)

Testability Evaluation

- Proposed metric's are mainly targeting functional verification - verification by simulation
- For HW - only up to the RT-Level
- No work targeting manufacturing faults

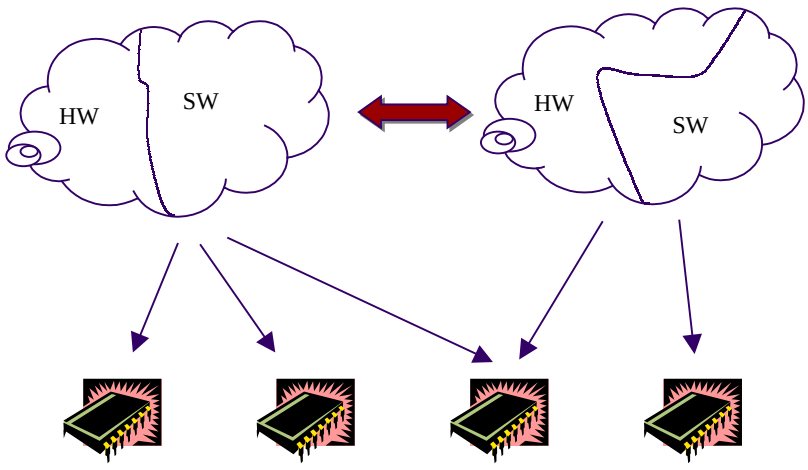
Bit coverage metric

- Behavioral error model
  - Bit failures
  - Condition failures
- Based on VHDL transformations
- Covers statement, branch, condition coverage and partially path coverage
- Gives accurate estimation of the RT- and gate-level fault coverage

Test Generation

- Test generation as a mean to detect hard to test parts
- Test generation at a high abstraction level for production faults
- Previous work targets code coverage + mutant faults

DfT Modifications



Next Steps

- To finalize the modeling
- Deeper investigation about the bit-coverage metric
- Adopt the existing test pattern generation algorithm for the bit-coverage metric
- DfT modifications