CoTest

# COTEST meeting

Paris March, 4th 2002

## Outline

- Experiment description
- Results
- Conclusions.

## Past Activities

- Different versions of the adopted benchmarks have been developed (using `Behavioral Compiler`)
- Prototypical environment supporting:
  - semi-automated fault list generation (supporting different fault models)
  - high-level fault simulation
  - gate-level fault simulation (using `faultsim`).

## Goal

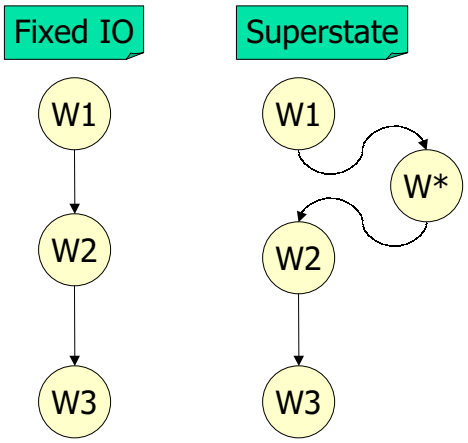
- The correlation between several high-level fault models with the gate-level stuck-at has been evaluated
- Different benchmark descriptions with different implementations have been considered.

## Assumptions

- Behavioral-compiler (BC) like model:
  - One entity
  - One process
  - Explicit clocking strategy via (multiple) `wait`
  - Global synchronous reset
- BC+DC Synopsys synthesis flow
- Simple gate-level library (and, or, ffd…).

## Scheduling mode

```
W1: wait on clock
  B <= f(A)
W2: wait on clock
  if B > 0 then
     C <= g(B)
  else
     C <= h(B)
W3: wait on clock
```



## Benchmarks

| Benchmark | Scheduling | VHDL lines | Gates | FFs |
|---|---|---|---|---|
| BIQUAD 1 | Fixed IO speed optimized | 93 | 6,043 | 125 |
| BIQUAD 2 | Fixed IO area optimized | 93 | 3,252 | 257 |
| FIR | Fixed IO | 53 | 5,553 | 141 |
| FIR | Superstate | 53 | 2,909 | 263 |
| TLC | Fixed IO | 168 | 241 | 23 |

## Input vectors

- Randomly generated
- Vector timing depends on adopted scheduling.

# Metrics

- Behavioral level:
  - Statement coverage: SC
  - Bit coverage: BC
  - Condition coverage: CC
- Gate level (GL):
  - Single permanent stuck-at: GL stuck-at

# Statement coverage

- Percent number of executed statements.

```
W1: wait on clock
  B <= f(A)
W2: wait on clock
  if B > 0 then
      C <= g(B)
  else
      C <= h(B)
W3: wait on clock
```

## Bit coverage

- Bit stuck-at: a bit in a variable/signal is stuck-at 1/0
- Percent number of detected bit stuck-at faults.

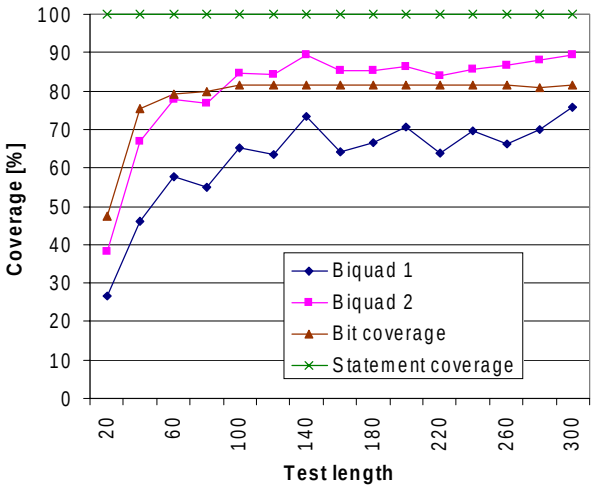
```
W1: wait on clock
  B <= f(A)
W2: wait on clock
  if B > 0 then
      C <= g(B)
  else
      C <= h(B)
W3: wait on clock
```

## Condition coverage

- Condition stuck-at: a condition is stuck-at true or false
- Percent number of detected condition stuck-at faults.

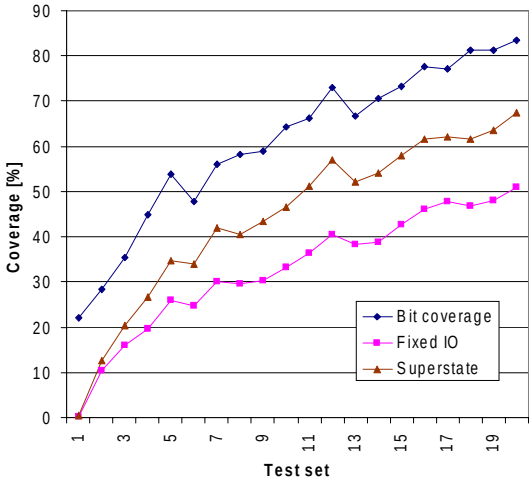
```
W1: wait on clock
  B <= f(A)
W2: wait on clock
  if B > 0 then
      C <= g(B)
  else
      C <= h(B)
W3: wait on clock
```

# BIQUAD



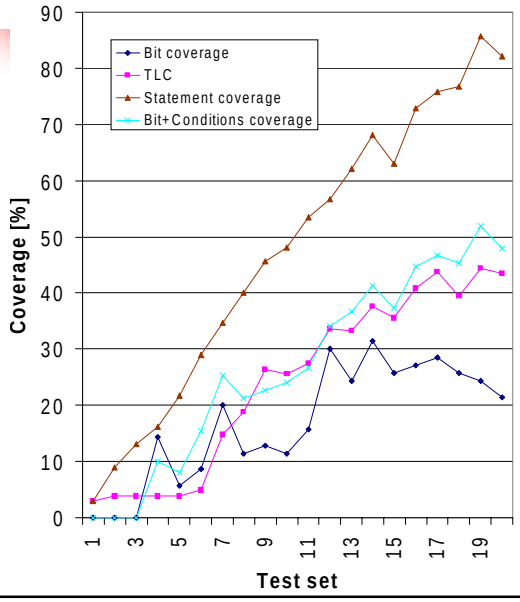

- CC is N.A. (no branches in the model)
- SC is useless (always 100%)
- BC has good correlation with the GL stuck-at for both the considered implementations

# FIR




- CC is N.A. (no branches in the model)
- SC (not shown here) is useless
- BC has very good correlation with GL stuck-at no matter which scheduling is considered.

# TLC




- SC is somewhat useful
- BC has poor correlation with GL stuck-at
- BC+CC has good correlation with GL stuck-at.

# Conclusions (I)

- Data-path oriented benchmarks (BIQUAD, FIR):
  - SC is useless
  - CC is not applicable (due to the benchmark nature)
  - Good correlation between BC and GL stuck-at.
  - Predictions about the testability are implementation-independent: same trends no matter the scheduling mode and the optimization constraints.

## Conclusions (II)

- Control-oriented benchmark (TLC):
  - SC is somewhat useful
  - BC has poor correlation with GL stuck-at
  - BC+CC has good correlation with GL stuck-at.

## Publications

- O. Goloubeva, M. Sonza Reorda, M. Violante, "Experimental analysis of fault models for behavioral-level test generation", IEEE DDECS 2002, Design and Diagnostics of Electronic Circuits and Systems Workshop

## Open questions

- Is it possible to further improve the correlation between high and gate levels ?
- What happen when pipelining is used ?
- Transition faults…
- More complex benchmarks…