Synthesis of Power- and Area-Efficient Binary Machines for Incompletely Specified Sequences

Nan Li and Elena Dubrova
Royal Institute of Technology (KTH), Sweden
Outline

- Binary machine
- Logic built-in self test
- Specifying don’t-care bits
- Switching activity analysis
- Experimental results
- Conclusion
Binary Machine

- Autonomous state machine

Finite state machine (Moore)

Binary machine
Binary Machine (cont.)

- Can be synthesized to generate any given sequence of bit vectors
  - Elena Dubrova, Synthesis of Parallel Binary Machines, ICCAD'2011.

```
001
010
111
110
...
```

![Diagram](attachment:image.png)
Logic Built-In Self Test (LBIST)

- Test patterns generated on-chip
  - Pseudo-random patterns – LFSR
  - Deterministic patterns – binary machine
- Requirements for binary machines
  - Small area overhead
  - Low power consumption
Deterministic Test Patterns

- Generated by automatic test pattern generation (ATPG) algorithms
- Incompletely specified
  - Over 90% don’t-care bits
- How to deal with don’t-care bits?
  - Specify don’t-care bits
  - Synthesize BM for completely specified sequence
How to Specify Don’t-Cares

- Random fill
  - Specify to 0 or 1 with equal probability
- Constant fill
  - Specify all to 0 (or 1)
- Inherit fill
  - Specify to be equal to the bit at the same position in the previous vector
- Others
  - Interleaving, pattern, inverse-inherit
Advantages of Inherit Fill

- Minimal number of register toggles
  - Minimal dynamic power dissipation
- No additional randomness introduced
  - Small area overhead
- Efficient implementation
Example

Random Fill

- x0x0
- 010x
- 1xxx
- 01x1
- 101x
- 1xx0
- 0x01
- xx1x
- 1xx0
- 0x01

Const-0 Fill

- 1000
- 0100
- 1010
- 0111
- 1111
- 1110
- 0001
- 0011
- 1100
- 0001

Inherit Fill

- 0000
- 0100
- 1100
- 1010
- 0110
- 1001
- 0001
- 0010
- 0011
- 0001
Example (cont.)

Random fill
$N_{\text{toggle}} = 24$

Const-0 fill
$N_{\text{toggle}} = 19$

Inherit fill
$N_{\text{toggle}} = 17$

Winner
Switching Activity Analysis

Toggle Probability

Fraction of specified bits

- Random fill
- Const-0 fill
- Inherit fill
Experimental Results (1)

- Dynamic power dissipation
- 16-bit vector sequences
- 10% specified (randomly to 0 or 1)
- Mapped to 90nm ASIC library
Experimental Results (1)

Dynamic Power Dissipation / uW/MHz

Sequence length (Number of 16-bit vectors)

- **Random fill**
- **Const-0 fill**
- **Inherit fill**
Experimental Results (2)

- Circuit area after mapping
- 16-bit vector sequences
- Mapped to mcnc.genlib
Experimental Results (2)

Area after mapping for sequence length 4096

- Random fill
- Const-0 fill
- Inherit fill

Percentage of specified bits
Experimental Results (2)

Area after mapping for 10% specified sequences

Sequence length

Random fill
Const-0 fill
Inherit fill
Conclusion

- We compared different methods for specifying don’t-care bits
- Inherit fill outperforms other methods
  - Smaller circuit area
  - Less power dissipation
- Potential use in embedding deterministic patterns on-chip