TAM Design and Test Data Compression for SoC Test Cost Reduction

Julien Dalmasso, Marie-Lise Flottes, Bruno Rouzeyre

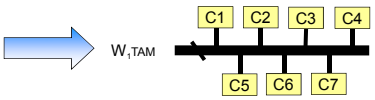
TAM design and Test Data Compression for SoC Test Cost Reduction

Julien DALMASSO, Marie-Lise FLOTTES, Bruno ROUZEYRE
LIRMM, UMR5506 CNRS/Université Montpellier II
161 rue Ada, 34932 Montpellier cedex 5, France
tel: (33)467418525, fax: 33)467418500
{dalmasso, flottes, rouzeyre}@lirmm.fr

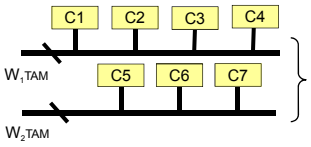
Framework

Motivations

- Number of Cores ↗
- Complexity of cores ↗
  ⇒Test Time of SoCs ↗
- **Goal: Test Time reduction**
- **How: Test Parallelism ↗**
  - ❑ Standard Solution: ATE costs ↗
  - ❑With Compression: ATE costs constant

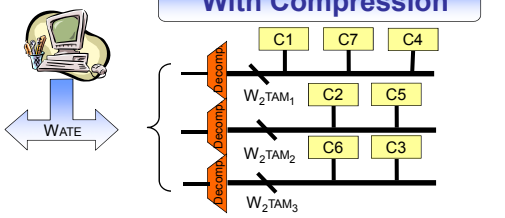


Standard Solution



$W_1TAM + W_2TAM = WATE$

With Compression



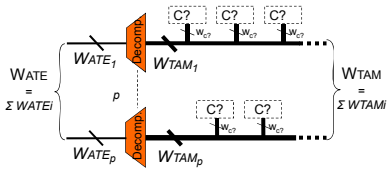
$WATE < \sum W_iTAM \Rightarrow$ Test Parallelism ↗

Remarks on compression

- Must follow SoC test paradigm:
  - Circuit netlist independent
  - Test data independent
  - No Specific tool
  - Low hardware decompressor
- No Impact on fault Coverage
- Allow Sharing of decompressor among cores
- May increase individual cores test time

Compression Technique

- **Compression Technique used here:**
  - [1] Julien Dalmasso, Marie-Lise Flottes, Bruno Rouzeyre: Fitting ATE Channels with Scan Chains: a Comparison between a Test Data Compression Technique and Serial Loading of Scan Chains - **DELTA 2006**: 295-300
- **Note**: Any orher technique meeting these requirements can be used in this framework



Goal: Minimize Test Time

- How: by determining:
  - Number of Buses (p)
  - Compression ratio for each bus ($WATE_i$ / $WTAM_i$)
  - Assignment of cores to buses

Test Architecture Exploration

Partition algorithm

1. For all ATE channels partitions into p parts
2. For each compatible TAM partition into p parts
3. Find the best assignment of the cores to the p lines (that minimize TAT)
   If this assignment reduces the global TAT, memorize this assignment and ATE/TAM architecture

Test Application Time

$$TAT = \max(TAT_i, i = 1,...,\#\text{buses})$$

$$\text{and } TAT_i = \sum_{\text{cores c assigned to i}} t_{c,\rho_i}$$

Core Assignment algorithm

// Initial Solution
- Assign each core on the smallest possible bus
- Compute TAT

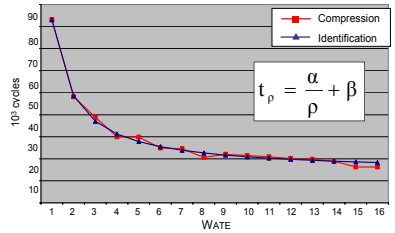
// Improvement of the solution
- While TAT is reduced
  - Find the line i with the highest TAT_i
  - For each core c assigned to i,
    - For all other lines k (k ≠ i)
      - Move core c from i to k
      - Compute new_TAT and memorize i, k, j, and new_TAT
      - Move back core c from k to i
  - Move core c from i to k such that:

1/ the smallest TAT has been obtained
2/ the number of useless bits on k is minimized
3/ the standard deviation between TAT_i of all lines is maximized

Test times interpolation

- Each Test Time / core / compression ratio must be known
  - IMPOSSIBLE: Too many configurations to be computed

⇒ For each WTAM, Identification of test time
⇒ Only 2*WTAM computations instead of $(WTAM)^2$



S38417 from ISCAS'89 benchmarks with 16 scan chains test times identification

Experimentations

Results

-SoC with 16 cores
-32 ATE channels
-TAM bitwidth: 64

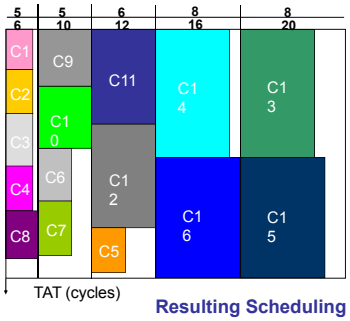
# bits	TAT	Compression parameters (W_{ATE-i} / W_{TAM-i})	#bits used on TAM	# config.
2	89892	(16,16) / (16, 48)	28	522
3	63441	(5,13,14) / (9,14,41)	36	44639
4	50249	(3,7,7,15) / (10,14,15,25)	48	1345142
5	42513	(5,5,6,8,8) / (6,10,12,16,16)	60	18605924
6	42513	(1,4,5,6,10,10) / (4,6,10,12,16,16)	60	142238520
7	44090	(1,1,1,1,6,7,15) / (5,5,6,6,12,14,16)	64	648424515

Gains

ATE Interface: 32 channels		
Test Application Time		Gain
No Compression 32 bits / 2 buses	With Compression 32 → 64 bits / 5 buses	
89892	42513	+ 53 % test time 0 % ATE Channels

TAM: 64 bits		
Test Application Time		Gain
No Compression 64 bits / 6 buses	With Compression 32 → 64 bits / 5 buses	
35402	42513	-17% test time +50 % ATE channels

Resulting Test Architecture and Scheduling



Resulting Scheduling (5 buses)

Resulting Architecture (5 buses)

Conclusion

- Compression increases test parallelism without any ATE Cost
  ⇒Thus it decreases Test Application Time

Note: Any compression technique compliant with SOC test paradigm can be used