

SoC Test Scheduling with Power-Time Tradeoff and Hot Spot Avoidance

James Chin and Mehrdad Nourani

Center for Integrated Circuits & Systems, Univ. of Texas at Dallas, Richardson, Texas 75083

{jchin,nourani}@utdallas.edu

ABSTRACT

We present a test scheduling methodology for core-based system-on-chips that can avoid hot spots and allows tradeoff between physical power dissipation and overall test time. A mixed integer linear programming formulation is presented to globally perform the power-time tradeoff, satisfy constraints, and produce the SoC test schedule.

I. INTRODUCTION

Using pre-designed cores to implement a system-on-chip can significantly shorten the design turnaround time and time-to-market. However, this advantage brings with itself many challenges for SoC testing. One of them is to find the test schedule with reasonable time under proper power constraints. The work in [1] presents a heuristic, based on the compatibility graph, to limit the power dissipation during test. In [2], an algorithm for power-constrained scheduling problem is presented. A method for determining preemptive and power-constrained schedules is discussed in [3]. Power profile and algorithms to reduce test time under power constraints are discussed in [4]. In [5] and [6], mixed integer linear programming (MILP) scheduling methods are presented to do power-time tradeoff for core-based systems without considering power distribution over physical areas.

The main contribution of our work is a test scheduling methodology for core-based SoCs that allows the user to avoid hot spots and perform power-time tradeoff. A mixed integer linear programming (MILP) formulation takes information about power distribution profile of non-embedded cores. This includes power profile over time as test patterns are applied and over area units, called *grids* (i.e. power distribution across physical area of layout). This method considers the total peak power consumption of the grids of the cores inside the selected areas *areas* and time/sequencing requirements as constraints.

II. POWER PROFILING OF NON-EMBEDDED CORES

A. Power Profiling over Time

We propose to approximate power analysis by putting some breakpoints in applying the patterns. These breakpoints effectively partition a test pattern set to some subsets and the formulation can look for the best timing to apply each. For *Core_j*, inserting $m - 1$ breakpoints partitions the set of n patterns to m subsets, e.g. $S_{j,1}, \dots, S_{j,k}, \dots, S_{j,m}$, such that $\sum_{k=1}^m |S_{j,k}| = n$. Without loss of generality, we assume that these breakpoints are drawn in such a way to create subsets with equal number of patterns, i.e. $|S_{j,k}| = |S| = n/m \forall k$. When the breakpoints are chosen, analytical methods or CAD tools can be employed to find various power values (e.g. peak, average or instantaneous) that eventually form the power profile of cores over time.

When multiple cores exist in the system, still one number, e.g. greatest common divisor, will be chosen for $|S|$. The basic test scheduling time step (time unit) will be equivalent to time

needed to apply $|S|$ test patterns. In this paper we refer to the time unit as *pattern set time step*. If the speeds of cores during test are different, we still use one parameter $|S|$ but it corresponds to different number of test patterns for different cores.

B. Power Profiling over Grids

In our work a *grid* is a physical or structural partition of a core for which power consumption can be measured. Such measurement can be done in two ways:

- 1) After the physical layout of a core is done, the layout can be partitioned into several small pieces by the user. Each piece forms a *grid*. The power distribution over the physical layout of the whole chip can be obtained with respect to a specific pattern set of a core. Some physical adjacent grids which may even belong to different cores can be grouped into an *area*. A user can define areas to avoid hot spots in selected regions.
- 2) A core can be partitioned into several sub-circuits. Each sub-circuit forms a *grid*. The power distribution of a sub-circuit can be obtained with respect to a test pattern set. With SoC floor plan information, some sub-circuits of different cores may be very close to save the chip space and cost. Similar to the previous case, those sub-circuits, or grids, can form an *area* for which formation of hot spot should be avoided.

In this paper, we use the first measurement of grid power. Note carefully that uniformity of grids is not a requirement in our method as long as power consumption of grids provided by the user corresponds to the same power density for user-defined hot spot areas.

III. THE MILP FORMULATION

A. Constants

W_t	Optimization weight for total test scheduling time
W_p	Optimization weight for total power consumption
N^{CORE}	Total number of cores
N^{AREA}	Total number of areas
N_j^{SET}	Total number of test pattern subsets for <i>Core_j</i>
N_j^{GRID}	Total number of grids in <i>Core_j</i>
P_i^{AREA}	Peak power allowed for <i>Area_i</i>
$P_{j,h,k}$	Peak power dissipation at <i>Grid_h</i> of <i>Core_j</i> when tested with subset S_k
$A_{i,j,h}$	Binary constant showing if <i>Area_i</i> includes <i>Grid_h</i> of <i>Core_j</i>

We also define T^{MAX} as the upper bound for pattern set time steps (or *time step*) required for scheduling. Assuming one time step needed per test pattern set, the sum of the length of the pattern sets of such cores is the upper bound T^{MAX} :

$$T^{MAX} = \sum_{j=1}^{N^{CORE}} R_j * N_j^{SET} \text{ where } R_j = \begin{cases} 1 & \text{if } \sum_{i=1}^{N^{AREA}} \sum_{h=1}^{N_j^{GRID}} A_{i,j,h} > 0 \\ 0 & \text{otherwise} \end{cases}$$

B. Variables

- $total_time$ represents overall test scheduling time.
- p_i^{AREA} denotes the total peak power consumption of the area i at any pattern set time step.

■ We need an additional set of variables as defined below:

$$t_{j,k,l} = \begin{cases} 1 & \text{if } Core_j \text{ is scheduled to receive its test data} \\ & \text{subset } S_k \text{ at time step } l \\ 0 & \text{otherwise} \end{cases}$$

- For the purpose of simplicity in writing the equations and constraints, we define $c_{j,l}$ variables: $c_{j,l} = \sum_{k=1}^{N_j^{SET}} t_{j,k,l}$.
- For each $Area_i$ and each time step l , total power of the grids inside that area i is $w_{i,l}$ and we have $w_{i,l} \leq p_i^{AREA}$.

C. Complete MILP Formulation

We use five indices (h, i, j, k and l) to define the variables. Their ranges are: $1 \leq i \leq N^{AREA}$, $1 \leq j \leq N^{CORE}$, $1 \leq k \leq N_j^{SET}$ and $1 \leq h \leq N_j^{GRID}$ for $Core_j$, and $1 \leq l \leq T^{MAX}$. The complete formulation will be as follows:

Minimize $(W_t \cdot total_time) + (W_p \cdot \sum_{i=1}^{N^{AREA}} p_i^{AREA})$
subject to:

- (1) $t_{j,k,l} \leq 1 \quad \forall j, k, l$
- (2) $c_{j,l} \leq 1 \quad \forall j, l$
- (3) $total_time \leq T^{MAX}$
- (4) $c_{j,l} = \sum_{k=1}^{N_j^{SET}} t_{j,k,l} \quad \forall j, l$
- (5) $l * c_{j,l} \leq total_time \quad \forall j, l$
- (6) $\sum_{l=1}^{T^{MAX}} c_{j,l} = N_j^{SET} \quad \forall j$
- (7) $p_i^{Area} \leq P_i^{Area} \quad \forall i$
- (8) $\sum_{j=1}^{N^{CORE}} \sum_{h=1}^{N_j^{GRID}} \sum_{k=1}^{N_j^{SET}} A_{i,j,h} * P_{j,h,k} * t_{j,k,l} = w_{i,l} \quad \forall i, l$
- (9) $w_{i,l} \leq p_i^{Area} \quad \forall i, l$
- (10) $\sum_{l=1}^{T^{MAX}} t_{j,k,l} = 1 \quad \forall j, k$
- (11) $\sum_{l=1}^{T^{MAX}} l * t_{j,k,l} < \sum_{l=1}^{T^{MAX}} l * t_{j,k+1,l} \quad \forall j, k$

The objective function is to minimize a linear function of weighted test time and power consumption. Selecting normalized weights for W_t and W_p imply the importance of these factors in the optimization process from the user point of view. Constraints 1 through 10 form a complete (mandatory for convergence) set for peak power and scheduling time control. Constraint 11 controls the order of patterns within the pattern set for sequential circuits.

If T^{MAX} is large we can tradeoff accuracy with faster MILP search time by considering larger size of test pattern subsets. The number of constraints is bound to $O((N^{AREA} + N^{CORE} + T^{MAX}) \cdot T^{MAX})$. Such complexity makes this formulation of a manageable size such that almost any MILP software package can solve it.

IV. EXPERIMENTAL RESULTS

We used the ILOG CPLEX package from ILOG S. A., Inc. [8] to solve our MILP formulation. To challenge a large example, we selected the Intel 8051 microcontroller. Excluding the

TABLE I

POWER-TIME TRADEOFF WHEN $N_j^{GRID} = 8$ ($1 \leq j \leq 6$) AND $N^{AREA} = 4$.

Weights (W_t, W_p)	Peak[Average] Power				Time Step
	p_1^{AREA} [P1]	p_2^{AREA} [P2]	p_3^{AREA} [P3]	p_4^{AREA} [P4]	
(1,0)	44.4[33.8]	42.4[34.4]	42.6[33.7]	46.0[35.0]	3
(0.98,0.02)	42.6[33.8]	39.9[34.0]	44.1[33.5]	43.7[35.0]	3
(0.97,0.03)	31.5[26.1]	31.4[25.8]	33.5[25.3]	32.6[26.3]	4
(0.9,0.1)	26.8[20.3]	24.9[20.6]	23.7[19.2]	25.8[21.0]	5
(0.5,0.5)	22.2[16.9]	21.9[17.2]	23.7[14.3]	24.5[17.5]	6
(0.1,0.9)	22.2[20.3]	21.9[17.2]	23.7[16.9]	24.5[17.5]	6
(0,1)	22.2[14.5]	21.9[14.7]	23.7[9.2]	24.5[8.8]	12

oscillator circuitry and RAM in test mode, the six cores in 8051 microcontroller have overall 20 input ports. To be able to show the power-time tradeoff in our MILP formulation we assume no bottleneck exists on the core access mechanism and we can access multiple cores at the same time.

The cores of 8051 system are described in VHDL. Synthesis, test patterns generation power simulation, and profiling are performed using Synopsys design tool set [7] using its generic library. Then, the MILP formulation has been applied to do the power-time tradeoff. We assumed each core has 8 grids and 4 areas are selected. The peak power of $Area_i$ is defined as $\max\{w_{i,l}\} (1 \leq l \leq T^{MAX})$. The average power of $Area_i$ will be: $\sum_{l=1}^{T^{MAX}} w_{i,l} / (e_r - b_s + 1)$, where e_r and b_s are $e_r = \max\{e_j\} (\forall Core_j \in Area_i)$ and $b_s = \min\{b_j\} (\forall Core_j \in Area_i)$. b_j and e_j denote time steps in which $Core_j$ receives the first and last test pattern sets, respectively. The results are tabulated in Table I for different (W_t, W_p) weights. The $total_time$ is in pattern set time step and total power of different areas are in $mWatt$, respectively. When W_t increases, the scheduling time shrinks but peak and average area power increase as the formulation gives higher weight to time reduction. In general, effective power-time tradeoffs can be done by providing various time/power limit, choosing different areas of interest and trying different weights.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation CAREER Award #CCR-0130513.

REFERENCES

- [1] R. Chou, K. Saluja and V. Agrawal, "Scheduling Tests for VLSI Systems Under Power Constraints," *Trans. on VLSI*, vol. 5, no. 2, pp. 175-185, June 1997.
- [2] V. Muresan, X. Wang, V. Muresan, and M. Vladutiu, "List Scheduling and Tree Growing Technique in Power-Constrained Block Test Scheduling," in *Digest of Papers European Test Workshop*, pp. 27-32, 2000.
- [3] V. Iyengar and K. Chakrabarty, "Precedence-Based, Preemptive, and Power-Constrained Test Scheduling for System-on-a-Chip," in *Proc. of VLSI Test Symposium (VTS)*, pp. 368-374, 2001.
- [4] P.M. Rosinger, B.M. Al-Hashimi, N. Nicolici, "Power profile manipulation: a new approach for reducing test application time under power constraints," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 21, No. 10, pp. 1217-1225, Oct. 2002.
- [5] K. Chakrabarty, "Test Scheduling for Core-Based Systems Using Mixed Integer Linear Programming," in *IEEE Trans. on CAD*, vol. 19, pp. 1163-1174, Oct. 2000.
- [6] M. Nourani and J. Chin, "Power-Time Trade off in Test Scheduling for SoCs," in *Proceedings of Int. Conf. on Computer Design, 2003*.
- [7] Synopsys Design Analyzer, "User Manuals for SYNOPSIS Toolset Version 2000.05-1," Synopsys, Inc., 2000.