

An Architectural Leakage Power Simulator for VHDL Structural Datapaths

Chandramouli Gopalakrishnan and Srinivas Katkoori
Department of Computer Science and Engineering,
University of South Florida, Tampa, FL 33620.
{cgopalak, katkoori}@csee.usf.edu

Abstract

We present a fast RTL leakage power simulator for datapaths described hierarchically in VHDL. Only the leaf-cells such as full adder, NAND gate etc., are characterized for leakage power. At the bit-slice level, exhaustive characterization can be performed in reasonable time. We observed that in the transient state, the leakage power is dependent on the previous input as well. This dependence is also incorporated into the leakage model. Using the characterized bit-slice cell library and a given set of inputs, the total leakage energy dissipated in a given datapath is estimated. Compared to HSPICE estimates, the average percentage error for three datapath-intensive designs is 1.38%. The estimation times are reduced by 4-5 orders of magnitude.

1. Introduction

Aggressive scaling down of supply and threshold voltages results in increased leakage power. As detailed leakage estimation using SPICE takes long times (hours to days), several leakage estimation models have been proposed at gate-level and transistor levels of abstraction [1, 2, 3]. We propose a fast leakage power simulation approach for hierarchical RTL VHDL structural datapaths. Katkoori and Vemuri [5] present a similar power simulator approach for dynamic power estimation.

Halter and Najm [4] establish that the leakage power in a CMOS circuit is dependent only on the current input to the circuit. On exhaustive simulation of the leaf cells, we observe that, this is true only after a certain period of time. For accurate characterization, we divide the time into the transient period and the steady-state period. During the transient period, leakage is dependent on the previous value of inputs to the circuit. During this transient time interval, drain-source voltages in the transistors change with the new input. During this settling time, the leakage power dissipated by the transistors (which is related exponentially to V_{ds} [7]), is a function of the previous input and the current input to the

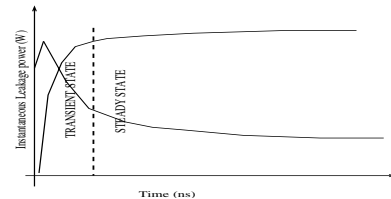


Figure 1. Typical Leakage Power Profile

leaf cell. Fig. 1 shows two typical leakage power profiles for a leafcell for a (previous input, current input) combination.

Each leafcell is characterized for all possible input pairs. The threshold time that separates the transient period and the steady-state period is identified empirically. The average leakage power values during the two intervals are measured using HSPICE and entered in a look-up table. As the number of input combinations to a leaf-cell is small (eg., 64 in case of a full adder), detailed bit-slice characterization can be done quickly.

Leakage power using HSPICE is determined as follows: The input transistor netlist to HSPICE is modified by adding commands to print: (1) voltage at each terminal of each transistor, and (2) instantaneous power of each transistor at 1ns interval. The terminal acting as source may change due to the voltage difference between the terminals. The state of the transistor is determined by comparing the V_{gs} and the threshold voltage of the technology. If the transistor is in cut-off state, then the instantaneous power dissipated is accumulated. This procedure has been coded in UNIX SHELL script and run on the raw data generated by HSPICE.

2 Estimation of Component and Data-path Leakage Power

The module library consists of storage units such as registers, functional units (FUs) such as adders and multipliers, and interconnect units such as multiplexers. There exists two architectures for each of the modules - the 1-bit architecture and the n-bit architecture. The n-bit architecture is obtained using the `generate` statement, depending on the

parameters (usually the bit width) supplied to it.

The 1-bit architecture itself consists of two parts: (1) A behavioral process which implements the functionality of the leaf cell; (2) A process which accumulates the leakage energy dissipated by the leaf cell. This process is sensitive to the inputs to the leaf cell.

Due to the bit-sliced design of these components, simulating the top level component VHDL file would result in simulating the leafcells, with appropriate inputs. This is taken care of by the VHDL simulator (NCLAUNCH in our case). The data-path in turn consists of instances of the n-bit architectures connected appropriately. Hence simulating the top level VHDL datapath would *percolate* the logic values on the primary inputs all the way down to the leafcells (see Fig. 2). Hence it is enough to incorporate our leakage power measuring logic only at the leafcell level.

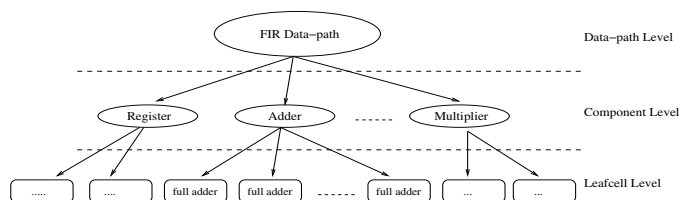


Figure 2. Hierarchy in Bit-Sliced Designs

The estimated leakage power profile is compared to that generated by HSPICE. The structural VHDL generated by AUDI [6] is translated into the Structural Description Language (SDL) format. The LAGER IV Silicon Compiler synthesizes a MAGIC layout of the design. Synopsys Nanosim is used to profile the input data streams through the HSPICE netlist and obtain input traces for all instances in the datapath. The leakage power for each of the instances are obtained using HSPICE and the post processor described in Section 1.

3 Experimental Results

We present results for three well known HLS benchmarks implemented in 100nm technology. The designs were synthesized using As-Soon-As-Possible schedule. HSPICE simulations were carried out for 100nm technology obtained from MOSIS.

Table 1 presents the leakage power estimates obtained using HSPICE and the proposed approach. Each benchmark is simulated with nine input vectors. We observe that the maximum percentage error in the leakage power for the datapath is 3.12%. We also report the leakage estimates at the component level. We see that the absolute error at the component level is in the range of 1.50% – 16.56%. The execution times for the proposed approach is the user time reported by the VHDL NCLaunch simulator. All time measurements

are taken on SUNSparc Ultra-2 Workstation with 256MB of memory. We observe a speedup of 4-5 orders of magnitude.

We conclude that *fast* and *accurate* leakage power estimation can be carried out by exploiting the hierarchy in the VHDL descriptions.

Module	HSPICE (uW)	Ours (uW)	% Estmn. Error
IIR Filter			
Adders	0.893	0.812	9.07
Registers	4.586	4.310	6.01
Multipliers	19.663	20.427	-3.88
Multiplexers	2.716	2.266	16.56
Datapath	27.858	27.816	0.15
Execution Time	2hr 47min	2.3s	
FIR Filter			
Adders	0.411	0.411	0.0
Registers	4.486	4.345	3.14
Multipliers	19.654	20.434	-3.96
Multiplexers	2.190	1.893	13.56
Datapath	26.740	27.083	1.28
Execution Time	2hr 34min	1.8s	
EWV Filter			
Adders	-0.816	0.849	-4.04
Registers	3.287	2.912	11.40
Multipliers	1.885	1.909	-1.27
Multiplexers	8.090	7.968	1.50
Datapath	14.079	13.638	3.132
Execution Time	36min	1.7s	

Table 1. Experimental Results

References

- [1] Z.Chen, L.Weil, M.Johnson, and K.Roy. "Estimation of standby leakage power in CMOS circuit considering accurate modeling of Transistor Stacks". In *Proceedings of the ISLPED*, pages 239–244, 1998.
- [2] R. Kumar and C. P. Ravikumar . "Leakage Power Estimation for deep sub-micron circuits in an ASIC design environment". In *Proceedings of the Fifteenth International Conference on VLSI Design, 2002*, pages 45–50, 2002.
- [3] J. A. Butts and G. S. Sohi . "A static power model for architects". In *Proceedings of the 33rd IEEE/ACM Annual Symposium on Microarchitecture (MICRO33)*, pages 191–200, 2000.
- [4] J. Halter and F. Najm. "A gate-level leakage power reduction method for ultra-low-power CMOS circuits". In *Proceedings of the CICC*, pages 475–478, 1997.
- [5] S. Katkooori and R. Vemuri. "A Power Simulator for VHDL Structural Descriptions". In *Proceedings of the VIUF Fall Conference*, pages 4.17–4.25, 1995.
- [6] C.Gopalakrishnan and S.Katkooori. "Behavioral Synthesis of Datapaths with Low Leakage Power". In *Proceedings of ISCAS*, pages 699–702, 2002.
- [7] K. Roy. "Leakage Power Reduction in Low-Voltage CMOS Design". In *Proceedings of the IEEE International Conference on Circuits and Systems*, pages 167–173, 1998.