

# Design and Analysis of a Low Power VLIW DSP Core

Chan-Hao Chang  
Industrial Technology Research Institute  
chanhao@itri.org.tw

Diana Marculescu  
Carnegie Mellon University  
dianam@ece.cmu.edu

## Abstract

*Power consumption has been the primary issue in processor design, with various power reduction strategies being adopted from system-level to circuit-level. In order to develop a power efficient system, architecture design, compiler optimization, as well as user evaluation must be employed in a unified framework. This paper presents an architecture-level power/performance simulator for a VLIW DSP processor core. Relying on parameterized power models and cycle accurate simulation, it provides fast and accurate power estimation for architecture exploration. Furthermore, the proposed modeling methodology can be used with minimal changes in the evaluation of other VLIW processor cores or for characterizing the efficiency of compiler-driven power efficient transformations.*

## 1. Introduction

Power-related issues have been the primary concerns for battery operated devices, such as communication systems and multimedia systems. A portable device needs to rely on a long battery life to enable the convenience leaded by mobility. To reduce the energy dissipation, processor design is especially important in low power/energy computing. Some well-known power reduction techniques such as resource scaling [2] and clock gating have been adopted for low power architectures, while, in recent years, voltage scaling and multiple voltages [5] have been extensively considered to save the unnecessary energy dissipation. In order to optimize architectures for power budgets and meet performance requirements, designers have to investigate the trade-off between various low power architectures and, by doing so, rely on tools that need to provide fast, but sufficiently accurate estimates for performance, as well as power consumption. Frequent iterations among RTL design, synthesis, gate-level simulation, as well as power analysis is prone to lengthening the design cycle and possibly violating

time-to-market deadlines. In addition to architecture exploration, compiler designers need to rely on a rapid cycle, accurate framework for compile-time optimizations. For a VLIW digital signal processor (DSP) which runs statically scheduled instructions, the scheduler needs to perform instruction ordering, resource scheduling and insert specialized power down instructions to reduce power consumption. Therefore, a framework with fast power estimation characteristics has to be adopted in support of such compiler optimizations. Moreover, user evaluation tools are always essential for processors. Evaluating the power consumption for battery operated applications is especially important.

In this paper, we present an architecture-level power estimation simulator, which could be employed in exploring architecture tradeoffs, compiler optimizations as well as user evaluations of a given architecture. The power estimation engine of this simulator is based on parameterized power models and cycle accurate simulation, while power reduction strategies such as voltage scaling and clock gating are considered to provide various design alternatives. Furthermore, this simulator is also configurable to deal with various instruction set architectures (ISAs), while scalable power models can be employed to estimate the power consumption of soft cores, which could be synthesized according to specific applications.

## 2. Prior work

In recent years, several simulation-based power analysis engines at instruction-level, RT-level and architecture-level have been proposed. Some of the recent power modeling techniques which play an important role on the accuracy of simulators are described in the sequel.

Power analysis at RT-level based on an application driven methodology [7] has been proposed. Instead of using macro modeling, this work relies on program profiling to extract the parameters from applications. For instruction-level, the power model is derived through the simulation of various instruction sets

[11,14]. However, in general, if the full spectrum of transitions between instructions is considered, the complexity of instruction-level power model increases exponentially with the number of instructions. Moreover, power models derived for specific instruction set architectures do not provide flexibility for varying different architecture parameters.

More recently, some ideas related to architecture-level power estimation have been presented. For example, Wattch [4] and SimplePower [12] provide cycle accurate framework to collect switching activities and employ unit power models for power estimation. Their power models are configured by width and input transitions, but this may not be sufficient to be used for various microarchitectures since the load capacitance may need to be adjusted according to the delay budget used for optimizing individual modules. In power modeling and pre-characterization, several techniques have been introduced, such as RT-level Power Modeling [8,1,17], structure-oriented technique [13] and Dual Bit Type model [9]. In contrast with using the random input assumption, the structure-oriented method considers all possible input transitions and further reduces the complexity of state transition graph (STG) by finding the minimum number of compatible pattern set. The work proposed in the Dual Bit Type model organizes an operand into sign region and uniform region due to their different transition behavior.

### 3. Framework

This proposed simulator is a cycle accurate design exploration environment based on a VLIW infrastructure and cycle-by-cycle simulation. In order to support multiple instruction issue processing, we treat each instruction way as an independent cluster to provide the flexibility for various architectures, with different number of instruction ways. Moreover, the architecture parameters in this simulator can be adjusted according to architecture specifications. For instance, examples of parameterized units include: the processing elements (PEs) or functional units, memory organization, or the number of pipeline stages. The flexible infrastructure enables designers to explore various design architectures.

The organization of simulator is illustrated in Figure 1. The core of the VLIW simulation engine is the pipeline operation composed of seven stages (where each stage could be skipped or extended to multi-cycle as well). The pipeline operation is implemented in software, so as to accomplish the work of each stage. In each stage, ways are executed sequentially, although the simulator emulates the actual parallel execution

taking place in hardware. Therefore, hazards occurring in shared register files and data memory are correctly detected in this simulator. For instance, there are two instructions issued at the same cycle for loading and storing data to the same memory address respectively, with the load instruction being completed prior to the store instruction. During the pipeline operation, the simulator calculates the transition probabilities of operands and uses pre-characterized power models for obtaining cumulative power numbers. In addition, the switching activity of pipeline registers is also accounted for. In order to simplify the usage of simulator and offer complete cycle accurate information for analysis and debugging, the loader in this simulator is responsible for program loading, while the debugger allows user to trace the state of operation through user interface. To launch the simulator, ISA parameters and benchmarks must be specified to configure memory organization, instruction path and corresponding power models. The operating conditions for frequency, voltage and synthesis frequency can also be selected independently for further exploration. In addition, the simulator can be used to figure out the optimal operation frequency/voltage point according to a given performance constraint. The simulator relies on fast simulation speed, which is more than 1000 times the simulation time of commercial tools. With such design exploration speed-up enabling various ISA evaluations, the extensive architecture analysis covers most useful strategies of power reduction.

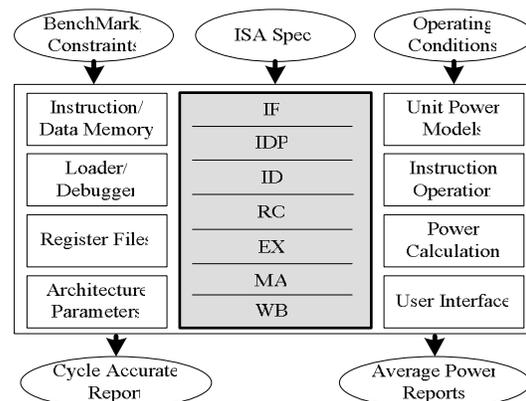


Figure 1. Block diagram of proposed simulator.

### 4. Power modeling

Our power modeling methodology is based on using Hamming distance between two consecutive input operands for determining switching activity and on the pre-characterization of main RT-level modules. Hamming distance has been extensively applied in dynamic power analysis [6], but it just examines a

portion of input vectors since many identical top bits caused by sign extension, and multi-function units increase the complexity of modeling [15]. Therefore, we analyze processing elements further to establish practical and parameterized power models. The parameters for the proposed parameterized power models are the bitwidth of PE, transition probability of input operands, operation voltage, as well as delay budget allocated to each PE. To truly support various architectures, the power models must be configurable for different timing constraints. Since most soft cores can be synthesized according to the characteristics of the target application, this simulator can also be used for soft processor evaluation as described next.

#### 4.1. Parameterized power models

We sort RT-level modules into two classes, combinational and array based components. Combinational components include PEs, fetch, dispatcher and decoder. We extract their power models based on input transition characterization. As shown in Figure 2, we illustrate our analysis methodology through a DesignWare adder [18]. The curves represent different widths for the adder, and each point is derived by simulating 1000 input patterns characterized by a specific transition probability. In fact, each curve could be a look-up table characterizing the power cost for certain bitwidths of the adder. For components that can have variable bitwidths (such as adders, etc.), we employ one curve and the ratio of average power (Figure 3) to determine the power tables for various bitwidths. To be able to reuse these power models, we consider the following factors. In an integrated system, the same type of components may be used to implement units which are characterized by different timing constraints. Furthermore, different architectures and user-defined synthesis speed targets for soft cores may also affect the average load capacitance according to the imposed timing constraints. Therefore, we analyze the influence of timing constraints on load capacitance. In Figure 4, we show the relation between average load capacitance and synthesis-driven timing constraint. The analysis provides not only the effect on the time budget of components but also the limit on the minimum time budget. With this practical information, the simulator could find the maximum operation frequency according to ISA specifications, and our power models could be scaled for various delay budgets.

Most processors provide specific registers for accumulating results. For example, the 40 bit-registers in a 32-bit processor provide 256 accumulation operations without precision penalty. However, a 40-bit adder must be employed to perform the operation.

But the smaller utilization causes that the extra 8 bits to be generally the sign extension (which is all zero or all one). For a two-operand component, we consider the switching activity of most significant bits (MSB's) in three situations: one sign bit transition, two sign bit transitions, and no sign bit transition. In other words, we separate the operands into MSB's and uniform part, and establish the power table for each situation.

The array based components triggered with clock signal, include pipeline registers, memory and register files. We consider the power consumption contributed by clock signal in each array based power model. For instance, a single register with synchronous reset signal is analyzed to estimate the power consumption of pipeline registers. The power model of single register is composed of three coefficients, leakage power, dynamic power and idle power. The dynamic power represents the average power that a register latches different value every rising edge of clock signal, in contrast with dynamic power, the idle power consider only clock swing and no switching on latched bit. The power consumption when there is no switching on input and clock signals determines leakage power. For register files, the models are established in terms of the data switching on cells and read-ports. The power consumption of a register file could be modeled in the following equation:

$$P_{rf} = P_{idle} + \sum_{1 \leq i \leq n} P_{wd} + N_{cell,i} P_{cb} + \sum_{1 \leq i \leq n} P_{rd} + N_{r,i} P_{rb}$$

where  $P_{idle}$  is the idle power,  $P_{wd}$  is the average power of write-decoder while one write-port is active,  $N_{cell,i}$  denotes the number of changed bits by a write-port,  $P_{cb}$  is the average power associated with one changed cell bit,  $P_{rd}$  is the average power of read-decoder while one read-port is active,  $N_{r,i}$  is the Hamming distance of two successive vector on a read-port and  $P_{rb}$  is the average power associated with one changed bit on a read-port.

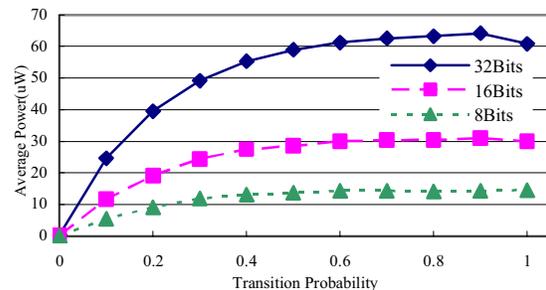


Figure 2. Average power as a function of input transition probability of Synopsys DesignWare adder.

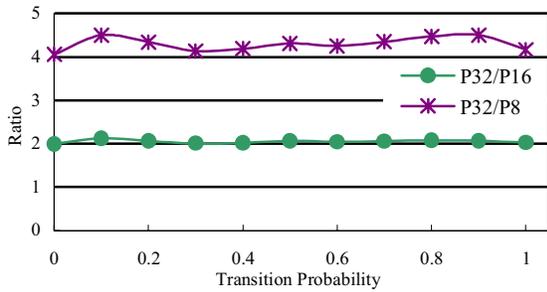


Figure 3. Ratio between the average power of 32-bit, 16-bit and 8-bit adder as a function of input transition probability.

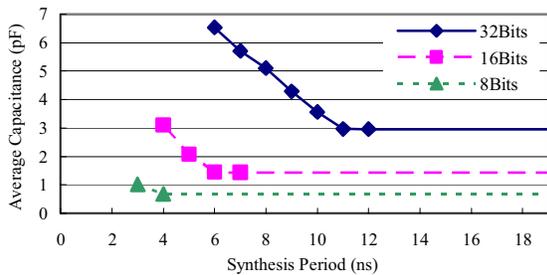


Figure 4. Average capacitance as a function of synthesis specified timing constraint.

## 4.2. Multi-function components

As they need to rely on resource sharing, existing processors typically use multi-function processing elements to perform various types of operations, while targeting full hardware utilization. In the case of multi-function processing elements, each operation mode may be characterized by different power values, especially when the control signals have transitions. Therefore, using a single set of switched capacitance models may be not sufficient to accurately model a multi-function component. We analyze input operands as well as control signals. The change in control signals makes a multi-function component switch the function from previous instruction to next one. For the purpose of computing switching activities, we treat both directions of transition between two types of functions as one transition. Figure 5 illustrates the average power for DesignWare Adder-Subtractor. The curves represent successive addition, successive subtraction and the switching between addition and subtraction. The average power numbers of addition and subtraction are quite close, but the curve add/sub which represents the switching between addition and subtraction contributes more power consumption. In this case, we establish two power tables for the Adder-Subtractor, one is the mean of curve add and curve sub, and the other one is exactly the curve add/sub. In fact, some functions of a multi-function component could

be incorporated into one type according to the power analysis. Doing so significantly decreases the size of power look-up tables.

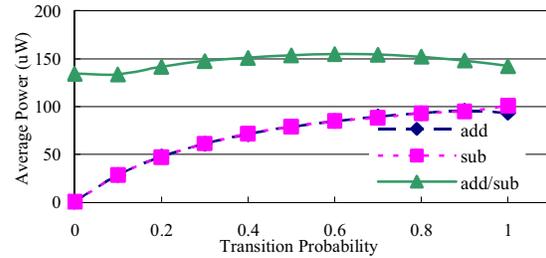


Figure 5. Average power as a function of input transition probability of multi-function component.

## 5. Experimental results

In this section, we describe how the simulator has been completely validated. Our experimental architecture is the PAC VLIW DSP v1.0 (Table 1) explored by Industrial Technology Research Institute, and the benchmark suite includes a series of DSP algorithm kernels including BDTI benchmarks [3]. The validation has been performed for all modules and the entire processor in terms of detailed power distribution, as well as the influence of synthesis timing constraints. The experiments are performed on a SparcII 500Mhz machine running SunOS 5.9 operating system. ModelSim SE 5.8b and Synopsys PrimePower 2002.05 are employed on gate-level simulation and power analysis respectively. Our simulator is more than 1000 times faster than the combination of PrimePower and ModelSim. Table 2 shows the comparison of average power between our simulator and the gate-level power analysis flow for entire DSP core. The DSP core is synthesized at 100MHz. The error is within 10% in all cases. In addition, we extract the power numbers of each component type. Figure 6 illustrates the percentage of register files which consists of accumulation registers, address registers as well as data registers. The average power consumption is about 28% of entire DSP core. We illustrate also processing elements in Figure 7. The power consumption depends on the active elements and input vectors. In Figure 8, we show the power consumption of pipeline registers.

For a soft core, the timing constraints provided for synthesis could be specified according to the target applications. Moreover, the effects of tight timing constraint can be reflected on the critical path. We demonstrate the modeling of MAC element which is on the most critical path of PAC VLIW DSP under different synthesis frequencies in Table 3. The target frequency for synthesis to be used in the simulator is set to 66Mhz, 83MHz and 100MHz. We validate these

benchmarks which have MAC element related instructions. As shown in Table 3, the results show a reasonable accuracy for different timing constraints.

TABLE 1. Configuration of the target architecture

VLIW Processor Core	
Fetch width	> 4 instructions/cycle
Decode width	5 instructions/cycle
Issue width	5 instructions/cycle
Instruction buffer	768 bits
Data path	2 cluster, 1 scalar unit
Functional units	2 ALUs/cluster 2 MACs/cluster 1 MUL/cluster 2 Shifters/cluster
Memory Hierarchy	
Instruction mem	256 KB, 2 ports
Data mem	256 KB, 4 banks
Max. Load/Store	5x32 bits/cycle

TABLE 2. Overall power consumption (TSMC 0.13um)

Benchmark	Cycles	Sim. (uW)	Prime. (uW)	Error (%)
dct	77	10505	11024	-4.7
rand1	80	7162	7360	-2.7
rand2	89	15748	16718	-5.8
fft64	2293	10018	10936	-8.4
bkfir	1779	9602	10655	-9.9
conv	3365	10526	10728	-1.9
iirm	4816	10619	10696	-0.7
fir2d	29964	10553	10663	-1.0
comult	280	10676	10714	-0.3
dot	338	10414	10572	-1.5
m1x3	1609	9826	10106	-2.8
comud	499	10256	10642	-3.6

TABLE 3. Results for various synthesis frequencies

Bench- mark	66MHz(uW)		83MHz(uW)		100MHz(uW)	
	Sim.	Prime.	Sim.	Prime.	Sim.	Prime.
dct	241	213	265	248	338	348
rand1	260	265	286	246	364	345
rand2	817	944	899	1035	1144	1139
fft64	112	108	123	121	157	141
bkfir	321	311	353	356	449	479
conv	8.9	9.8	11.4	11.2	13.9	13.3

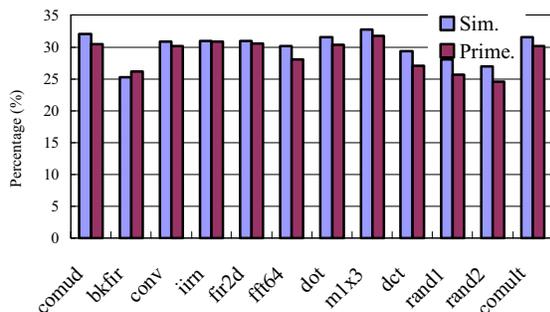


Figure 6. Percentage of register files power from total power budget.

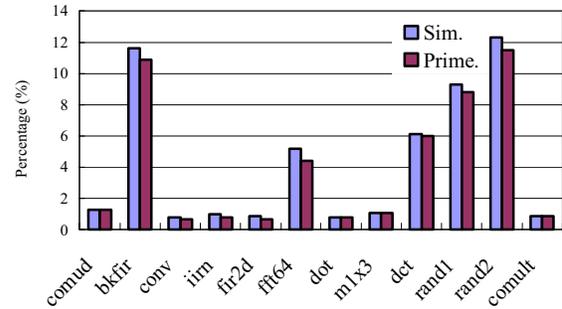


Figure 7. Percentage of processing elements power from total power budget.

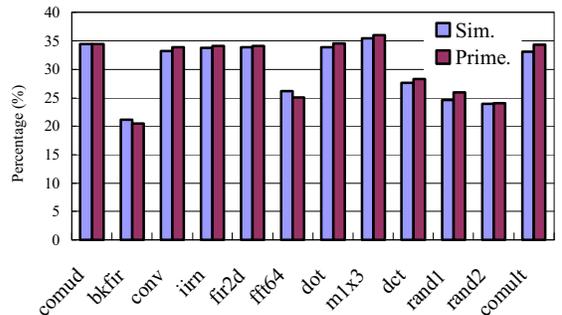


Figure 8. Percentage of pipeline registers power from total power budget.

## 5.1. Case study of clock gating

We have also performed a detailed analysis of the power distribution across various datapath modules (Figure 6-8). As seen in Figure 8, approximately 35% of the total power consumption is contributed by pipeline registers, while register files consume 30% of the total power budget (Figure 6).

For benchmarks that have fewer parallel instructions because of longer data dependency chains, many resources turn out to be redundant and consume power in multi-ways processors. Clock gating is a well-known technique for pipeline power reduction [16,10] especially for multi-issued processors. We employ this technique in our simulator to investigate the influence of clock gating on pipeline registers and register files (Figure 9). The three levels we consider are cluster base (CB), register file (RF) as well as pipeline register (PR). CB is a coarse-grain clock gating, the entire data path is isolated by clusters and one scalar unit, the clock signal of specific cluster is gated while there is no valid instruction traveling in it. RF focuses on register files. Each cluster has one address register file, one accumulation register file and two data register files. We detect the instructions in read-operand stages and write-back stages in order to determine the idleness of register file. Regarding PR,

each instruction in data path is analyzed to determine if there will be any valid instructions to be transferred through pipeline registers. For instance, the clock signal of the pipeline register between execution stage and memory-access stage is gated while the instruction in execution stage is invalid. In addition, the corresponding leakage power is considered when clock gating is used. This simulator provides valuable data through fast simulation. The estimates illustrate the percentages of power saving and the comparison of different strategies. Given the efficiency with which they have been obtained, they could determine alternatives of architecture optimizations without frequent gate-level analysis flow.

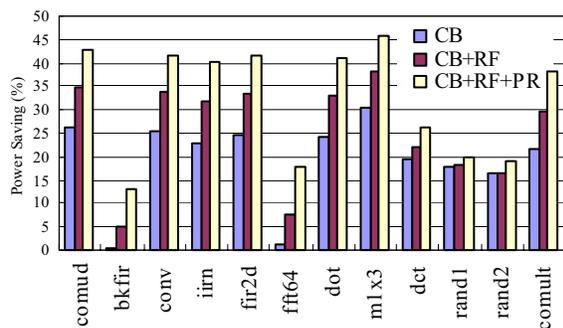


Figure 9. Power saving due to clock gating.

## 6. Conclusion and future works

The processor design is getting more complex, with energy consumption being now the primary design issue for portable devices. Iterating between synthesis, gate-level simulation and power analysis is very expensive and may lead to time-to-market violations. We have presented a simulator, which is at the heart of an architecture-level power/performance estimation framework providing fast simulation and reasonable accuracy. For user evaluation, our simulator relies on the parameterized power models to enable hard core and soft core power/performance evaluation.

Future work for this simulator will include power analysis for the memory subsystem. In general, the memories in processors are macro blocks, thus our memory models will be established according to the characteristic of memory macros. Moreover, our power models will be augmented to include information about more aggressive technologies (90 and 65nm).

## 7. References

[1] A. Bogliolo, R. Corgnati, E. Macii and M. Poncino, "Parameterized RTL Power Models for Soft Macros," *IEEE Transactions Very Large Scale Integration Systems*, vol. 9, pp. 880-887, Dec. 2001.

[2] A. Iyer and D. Marculescu, "Power Aware Microarchitecture Resource Scaling," *Design Automation and Test in Europe Conference*, pp. 190-196, Mar. 2001.

[3] Berkeley Design Technology Inc., "Evaluating DSP Processor Performance," <http://www.bdti.com>.

[4] D. Brooks, V. Tiwari and M. Martonosi, "Wattch: A Framework for Architecture-Level Power Analysis and Optimizations," *Proceedings of the 27th International Symposium Computer Architecture*, pp. 83-94, Jun. 2000.

[5] D. Marculescu, "Power Efficient Processors Using Multiple Supply Voltages," *Workshop on Compilers and Operating Systems for Low Power*, Oct. 2000.

[6] G. Jochens, L. Kruse, E. Schmidt and W. Nebel, "A New Parameterizable Power Macro-Model for Datapath Components," *Design, Automation and Test in Europe Conference and Exhibition*, pp. 29-36, Mar. 1999.

[7] H. Cheng-Ta and M. Pedram, "Profile-Driven Program Synthesis for Evaluation of System Power Dissipation," *Proceedings of the 34th Design Automation Conference*, pp. 576-581, Jun. 1997.

[8] J. Costa, J. Monteiro, L. M. Silveira and S. Devadas, "A Probabilistic Approach for RT-Level Power Modeling," *Proceedings of 6th IEEE International Conference Electronics, Circuits and Systems*, vol. 2, pp. 911-914, 1999.

[9] J. Y. Lin, W. Z. Shen and J. Y. Jou, "A Power Modeling and Characterization Method for Macrocells Using Structure Information," *IEEE/ACM International Conference Computer-Aided Design*, pp. 502-506, Nov. 1997.

[10] L. Hai, S. Bhunia, Y. Chen, T. N. Vijaykumar and K. Roy, "Deterministic Clock Gating for Microprocessor Power Reduction," *The Ninth International Symposium on High-Performance Computer Architecture*, pp. 113-122, Feb. 2003.

[11] M. Sami, D. Sciuto, C. Silvano and V. Zaccaria, "An Instruction-Level Energy Model for Embedded VLIW Architectures," *Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, pp. 998-1010, 2002.

[12] N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim and W. Ye, "Energy-driven integrated hardware-software optimizations using SimplePower," *Proceedings of the 27th International Symposium Computer Architecture*, pp. 95-106, Jun. 2000.

[13] P. E. Landman and J. M. Rabaey, "Activity-Sensitive Architectural Power Analysis," *IEEE Transactions Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, pp. 571-587, Jun. 1996.

[14] P. Stanley-Marbell and M. S. Hsiao, "Fast, Flexible, Cycle Accurate Energy Estimation," *International Symposium Low Power Electronics and Design*, pp. 141-146, Aug. 2001.

[15] S. Haga, N. Reeves, R. Barua and D. Marculescu, "Dynamic Functional Unit Assignment for Low Power," *Design, Automation and Test*, pp. 1052-1057, 2003.

[16] S. Manne, A. Klauser, and D. Grunwald, "Pipeline Gating: Speculation Control for Energy Reduction," *Computer Architecture*, pp. 132-141, Jul. 1998.

[17] S. Theoharis, G. Theodoridis, P. Merakos and C. Goutis, "Accurate data path models for fast RT-level power estimation," *IEE Proceedings Computers and Digital Techniques*, vol. 147, pp. 209-214, Jul. 2000.

[18] Synopsys, Inc., <http://www.synopsys.com>.