System-Level Energy Modeling for Heterogeneous Reconfigurable Chip Multiprocessors

Xiaofang Wang[1], Sotirios G. Ziavras[2], and Jie Hu[2]

[1]*Dept. of Electrical and Computer Engineering*
Villanova University
Villanova, PA 19085
xiaofang.wang@villanova.edu

[2]*Dept. of Electrical and Computer Engineering*
New Jersey Institute of Technology
Newark, NJ 07102
{ziavras, jhu}@njit.edu

Abstract—**Field-Programmable Gate Array (FPGA) technology is characterized by continuous improvements that provide new opportunities in system design. Multiprocessors-on-a-Programmable-Chip (MPoPCs) represent the recent trend in this arena; they integrate the advantages of both software programmability and hardware reconfigurability. However, FPGAs consume more energy than ASICs. The lack of powerful tools and models to estimate and verify the energy consumption in the early stages of the design cycle exacerbates this problem. In this paper, we propose a system-level energy estimation model to accompany our design methodology for HERA (HEterogeneous Reconfigurable Architecture), a versatile reconfigurable MPoPC that we have implemented on Xilinx FPGAs. The model utilizes both physical-level measurements from a hardware component library and application statistics. Experiments with the parallel LU factorization of large sparse matrices show an average error in energy estimation of about 5.17%. We also demonstrate performance-energy trade-off test cases that incorporate this model into HERA's design methodology to satisfy the real needs of application-system pairs.**

Index Terms—**Chip multiprocessor, energy modeling, FPGA, SIMD/MIMD mixed-mode computing.**

I. INTRODUCTION

Reconfigurable chip multiprocessors integrate the advantages of reconfigurable computing and the flexibility of software programming. FPGA-based reconfigurable systems have shown impressive benefits for several classes of applications [1] and increasingly appeal more to system designers than ASICs of ever-increasing complexity and cost. Traditionally, most FPGA-based designs are *application-specific programmable circuits* (ASPCs) that are not user programmable and require extensive hardware expertise to achieve high performance. Large complex applications may exceed the FPGA capacity, resulting in numerous device reconfigurations for ASPCs. Each reconfiguration often consumes more than 100 mA and lasts tens to hundreds of milliseconds. MPoPCs can eliminate forced reconfigurations due to their programmable function units (FUs); also they are more friendly to the average user without hardware expertise.

Moreover, our MPoPCs carry the advantage of customizing the architecture at static and run times to match the diverse computation and communication characteristics of tasks. HERA is a mixed-mode reconfigurable MPoPC implemented on Xilinx Virtex II FPGAs [2]. It targets floating-point (FP) data-parallel applications and groups of processors can work simultaneously in a variety of independent or cooperating parallel computing modes, such as SIMD (Single-Instruction, Multiple-Data), multiple-SIMD and MIMD (Multiple-Instruction, Multiple-Data). Their group assignment and mode of computation are user programmable. HERA comprises heterogeneous PEs (Processing Elements) synthesized with selected FP FUs (e.g., adders, multipliers) from an in-house developed *parameterized hardware component library* (PHCL). The selection is based on various performance-energy objectives.

Unfortunately, SRAM FPGAs are more energy hungry than their ASIC counterparts due to higher routing capacitance and less efficient transistor utilization. Compared with processors implemented with fixed logic, soft IP (Intellectual Property) processors in FPGAs typically consume more energy even at a lower frequency [3]. Energy consumption, a concern for all digital designs, is becoming more of an issue for FPGA-based systems targeting high performance. It is especially devastating to battery-powered systems, where FPGAs are promoted to have more advantages over ASICs. The increasing role of energy constraints as performance reducers may force architects to consider energy consumption early in the design process. This is of utmost importance for FPGA-based systems with a short turnaround time or the need for runtime reconfiguration decisions. Early design decisions can have the greatest influence on energy consumption [4].

A. Related work and contributions

System-level power/energy modeling techniques for processor-centric systems can generally be classified as either instruction-oriented [5-6] or component-oriented [7-8]. The hindrance of the former approach lies in modeling inter-instruction impacts [5] (e.g., data dependencies) on the consumption. Also, such processor-dependent results do not provide much information on the energy distribution of its individual components to aid the task of resource management; this information, however, is prudent to use in architecture design and optimization, especially when

1-4244-9707-X/06/$20.00 ©2006 IEEE

focusing on hardware reconfigurability. Component-oriented modeling is time-consuming but tends to be more accurate. A hybrid technique has been applied to extensible processors [9], where instruction-oriented energy modeling for the basic processor is augmented with component-based analysis for the custom extensions.

Minuscule system-level energy modeling efforts have appeared for chip multiprocessors and FPGA-based systems. Previous efforts for FPGAs [12-14] have focused on lower levels which may be very helpful for ASPCs. They often involve sophisticated capacitance models and assume detailed low-level design information at the gate or register-transfer level. Continuous increases in chip density and gate count make low-level tools very slow and impractical in architecture exploration. Instruction-level rapid energy estimation for soft IP microprocessors on FPGAs is presented in [10]; a processor is treated as a black box and the impact of inter-instruction interaction is ignored. Due to major architectural differences between FPGA- and SoC-based designs, our MPoPC analysis cannot rely on previous results for fixed logic. For example, ref. [11] simulates a shared-memory, bus-interconnected homogeneous ARM-based on-chip multiprocessor to find out that the main consumers of power are the caches. Our experiments and related work at the physical level [12-13] show that the logic and the interconnect network are the main energy consumers in state-of-the-art FPGAs.

We propose here a system-level, component-oriented energy estimation model for HERA which provides a quantitative basis for performance-energy trade-offs during system synthesis and runtime reconfiguration. The model employs physical-level measurements for the FUs in the PHCL and application statistics. Device-based physical power data of the FUs are measured only once at static time to dramatically reduce the simulation time associated with component-oriented models during architecture exploration. The activity cycles of various components are measured by application profiling using embedded monitoring hardware. Our experiments show that the PEs are the main contributors to energy and, hence, they become our focus in energy modeling. The on-chip memory of PEs utilizes BlockRAM blocks in Xilinx FPGAs and exhibit different characteristics from those in [11]. The basic HERA framework targets matrix-oriented applications and its knowledge before customization provides an accurate starting point in energy modeling.

The organization of this paper is as follows. Section II presents a brief description of HERA's architecture and design methodology. In Section III, we show how to characterize the power consumption of the FUs in the PHCL. The system-level energy model for HERA is presented in Section IV. Experimental results are shown and discussed in Section V. Finally, Section VI presents our conclusions.

II. Overview of HERA's Organization and Design Methodology

As the focus of this paper is energy modeling, we present here only an overview of HERA and its design methodology. More details are available in [2, 19]. Fig. 1 shows the general organization of our HERA MPoPC with its PEs interconnected via a 2-D mesh. We employ fast, direct NEWS (North, East, West, and South) connections between nearest neighbors. The computing fabric is controlled by the system *Sequencer* that communicates with the host via the PCI bus for data I/O. The *global control unit* (GCU), included in the system Sequencer, fetches instructions from the *global program memory* (GPM) for PEs operating in SIMD. A PE of HERA is an in-house designed pipelined RISC processor that consists of an integer FU, one or more pipelined FP FUs (from the PHCL), other *custom function blocks* (CFBs), and dual-port *local data memory* (LDM) and *local program memory* (LPM). Every PE includes a small amount of control logic, and can realize both the MIMD and SIMD modes of execution. In addition to the NEWS interconnect, HERA also has a hierarchical bus system. Every PE is connected to a column *Cbus* and all the *Cbus*es are connected to the *Column Bus* for broadcasting SIMD instructions and their immediate data. SIMD instructions and data are transferred via the *Dbus* in a pipelined fashion.

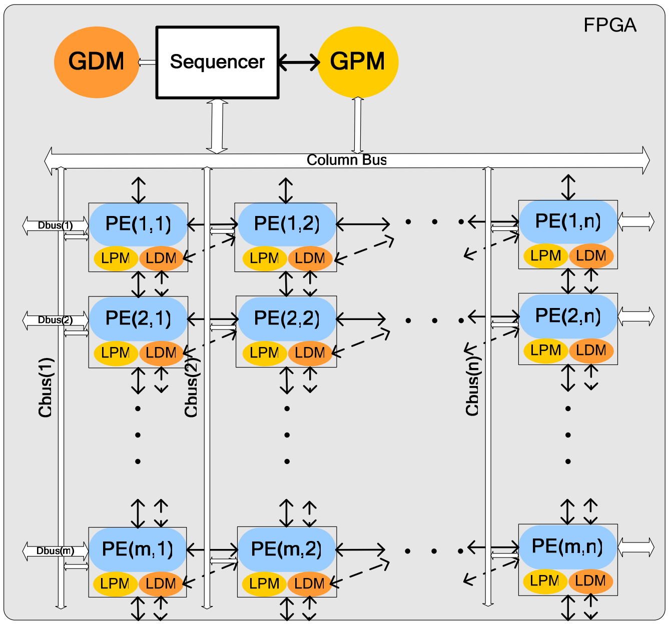

Fig. 1. HERA organization.

Our design methodology [19] starts with a task flow graph with SIMD and MIMD tasks, the PHCL, and an FPGA with limited resources in terms of logic (slices), on-chip memory blocks (BlockRAM), embedded DSP blocks, etc. FP FUs are very resource expensive in FPGAs. Not all FP operation types are needed all the time by all the tasks in the given application. Our implementation results in Table I show that a single-precision IEEE-754 FP divider consumes about 2 or 6 times more resources than an FP adder or multiplier of the same precision, respectively. By removing unneeded FP FUs from the PEs to run a task, we can potentially increase the number of PEs assigned to the task and thus improve the performance. It is then beneficial to employ a dynamically changing HERA architecture where the functionality of PEs and their number vary as needed by different tasks. Also, different application-system pairs may have different performance-energy objectives. To this extent, the PHCL contains diverse FU types for system synthesis.

III. POWER CHARACTERIZATION OF FUS

The major parameterized components in the PHCL include: variant precision pipelined FP FUs, HERA system and PE templates, memory blocks of various sizes and port populations, CFBs, etc. For each operation type (+, -, *, /, $\sqrt{\ }$, …) of a given precision, there are many choices in terms of latency, resource requirements, frequency, and power consumption. The components are designed in VHDL, and placed and routed for the target FPGA device. We first conducted a set of experiments to evaluate the effect of various factors on FU power dissipation before deciding on appropriate system-level energy modeling. ISE 7.1 and XPower 7.1 from Xilinx, ModelSim SE 6.0, and Synplify Pro 8.0 are our power analysis tools. ModelSim takes the placed/routed design from ISE and produces annotated gate-level simulation results, which are then used by XPower to produce a power report. The power dissipation in SRAM-based FPGAs can be broken down into the *static*, *dynamic*, and *configuration* (not considered in this paper) components. We assume typical static power values that can be determined by using vendor power analysis tools at static time. The impact of run-time temperature variations is not considered. We approximate an FU's contribution based on its corresponding resource consumption. Table I shows the resource usage and total power of the single- and double-precision FUs in the PHCL on an XC2V6000-5 FPGA. The frequency is set to 100MHz and the average input activity rate is 20%. "S_" and "D_" stand for single- and double-precision, respectively. The dynamic power of all the FUs dominates the total power. Table I also shows that double-precision FUs require much more dynamic power than their corresponding single-precision counterparts due to more resource requirements. Fig. 2 shows the dynamic power per slice for these designs. All the double-precision FUs, except the divider, require slightly smaller power per slice than their single-precision counterparts.

Contributors to the dynamic energy consumption of an FPGA are the device core, and the auxiliary and I/O blocks. The latter two parts are related to the board implementation, so we are only interested here in the first factor. The dynamic power of an FU increases linearly with its frequency, the average number of its activated switches per clock cycle, and the switch capacitance. Dynamic power is primarily affected by the resource utilization, implementation, and circuit switching activity [12-13]. Fig. 3 shows the impact of the average input activity rate on the dynamic power of single-precision FUs. The variances due to different rates are much less significant compared to the big gap between the idle (activity rate of 0%) and active states. The results for other precision FUs show a similar pattern. Hence, we distinguish among four power states for an FU: *active*, *idle*, *standby*, and *sleep*. An FU consumes both static and dynamic power in the *active* and *idle* states, and only static power in the *standby* state. All consumptions are eliminated by shutting down the power supply to an FU (*sleep* state). The consumption of an FU in the *idle* state is due only to clock activities. An FU is put into *standby* by disabling its clock signal.

TABLE I
RESOURCE USAGE AND POWER DISSIPATION (mW) FOR IEEE-754 SINGLE- AND DOUBLE-PRECISION FP FUS

FU	*Slices*	*Dynamic*	*Static*	*Total*
S_ADD	343	247.5	4.1	251.6
S_MUL	119	75.9	1.4	77.3
S_DIV	731	559.9	8.7	568.6
S_SQRT	666	435.9	8.0	443.9
D_ADD	745	472.5	8.9	481.4
D_MUL	836	493.8	10.0	503.8
D_DIV	3089	2526.5	36.9	2563.4
D_SQRT	2757	1409.0	33.0	1442.0

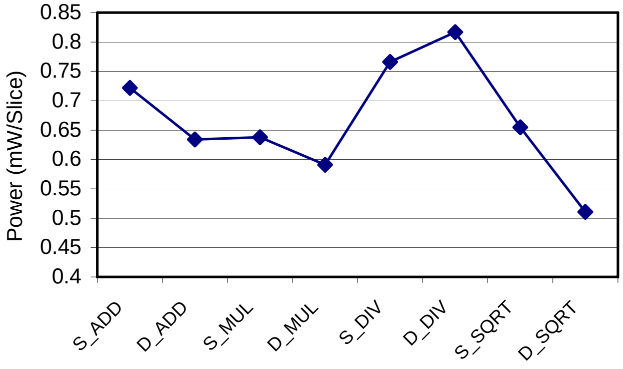

Fig. 2. Dynamic power dissipation (per slice) for the single- and double-precision FP FUs.

Let FU$_{j,k}$ denote the k^{th} realization in the PHCL of an FP FU capable of the operation type j, where $j \in \{+, -, *, /, \sqrt{\ }, \ldots\}$. The required power for FU$_{j,k}$ in the *active*, *idle*, and *standby* states is represented by $P_{j,k}^{active}(F)$, $P_{j,k}^{idle}(F)$ and $P_{j,k}^{stdby}$, respectively, where F is the implemented system frequency. They grow almost linearly with F and can be obtained from experiments. We approximate the dynamic power part of $P_{j,k}^{active}(F)$ by a linear function of the input activity rate, as suggested by Fig. 3. Exhaustive simulation to get the average activity rate for an application is impractical; vendors suggest an average activity rate of 12-24% [15]. Given an application, we obtain a typical rate for each task through simulation with ModelSim; XPower then produces $P_{j,k}^{active}(F)$. Fig. 3 also implies that for the same clock frequency and zero activity rate different designs consume almost the same power.

Similar parameters for the BlockRAM blocks, buses and Sequencer are obtained with similar experiments. For the BlockRAM memory, the power variation between reads and writes is very small. The variance in the activity rate due to different data and addresses has very little impact on the consumption whereas the clock activity and the number of accesses are the main contributing factors. Hence, the clock of all the BlockRAM blocks in HERA is controlled by a glitch-free enable signal provided with the memory blocks. Due to limited space, we do not show here corresponding experimental results.

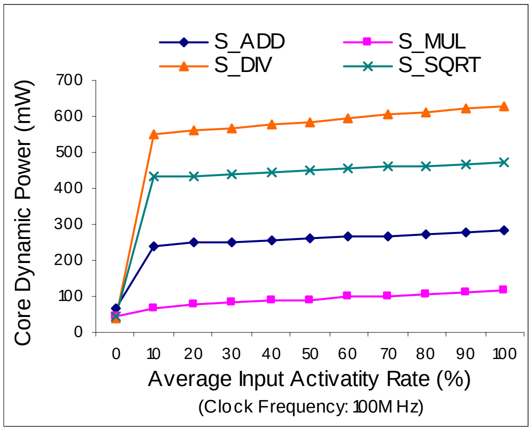

Fig. 3. Impact of the average input activity rate on dynamic power for the single-precision FP FUs.

IV. HERA ENERGY ESTIMATION MODEL

The major components of HERA are the PEs, their LDM and LPM, buses, NEWS interconnect, Sequencer, GDM and GPM, and system template. Let *p* be the total number of PEs. The total energy consumption, E_{sys}, of HERA with *p* PEs for a given application can be represented by:

$$\begin{aligned} E_{sys} &= \sum_{i=1}^{p} E_{PE(i)} + E_{MEM} + E_{seq} + E_{bus} + E_{NEWS} + E_{sys}^{tpt} \\ &= \sum_{i=1}^{p}\{[\sum_{j}\sum_{k}(E_{i,j,k}^{FU} * \gamma_{i,j} * \gamma_{j,k}) + \sum_{m} E_{m}^{CFB} * \gamma_{i,m}] + \\ &\quad E_{PE}^{tpt}\} + E_{MEM} + E_{seq} + E_{bus} + E_{NEWS} + E_{sys}^{tpt} \end{aligned} \tag{1}$$

where

$$\gamma_{i,j} = \begin{cases} 0 & \text{if PE}\,(i)\text{ does not support the FP operation type } j \\ 1 & \text{if PE}\,(i)\text{ supports the FP operation type } j \end{cases}$$

$$\gamma_{j,k} = \begin{cases} 0 & \text{if PE}\,(i)\text{ does not include an FU}_{j,k} \\ 1 & \text{if PE}\,(i)\text{ includes an FU}_{j,k} \end{cases}$$

$$\gamma_{i,m} = \begin{cases} 0 & \text{if PE}\,(i)\text{ does not include a CFB}_{m,}\text{ i.e., the } m^{th} \text{ type of CFB} \\ 1 & \text{if PE}\,(i)\text{ includes a CFB}_{m} \end{cases}$$

$E_{PE(i)}$, E_{MEM}, E_{seq}, E_{bus}, E_{NEWS}, E_{PE}^{tpt}, E_{sys}^{tpt}, and E_{m}^{CFB} represent the energy consumption of PE(*i*), system memory, Sequencer, bus, NEWS, PE template, system template, and the m^{th} CFB, respectively. The energy consumption $E_{i,j,k}^{FU}$ of $FU_{j,k}$ in PE(*i*) is determined by:

$$\begin{aligned} E_{i,j,k}^{FU} &= E_{j,k}^{active} * C_{i,j,k}^{active} + E_{j,k}^{idle} * C_{i,j,k}^{idle} + E_{j,k}^{sdby} * C_{i,j,k}^{stdby} \\ &= P_{j,k}^{active}(F_i) * \frac{1}{F_i} * C_{i,j,k}^{active} + P_{j,k}^{idle}(F_i) * \frac{1}{F_i} * C_{i,j,k}^{idle} + \\ &\quad P_{j,k}^{static} * \frac{1}{F_i} * (C_{i,j,k}^{active} + C_{i,j,k}^{idle} + C_{i,j,k}^{stdby}) \end{aligned} \tag{2}$$

where F_i is the clock frequency of PE(*i*), and $E_{j,k}^{active}$, $E_{j,k}^{idle}$, and $E_{j,k}^{stdby}$ represent the energy consumption per clock cycle of $FU_{j,k}$ in the respective states. $C_{i,j,k}^{active}$, $C_{i,j,k}^{idle}$, and $C_{i,j,k}^{stdby}$ are the consumed clock cycles in the corresponding states in PE(*i*). Although the PE or system template consumptions differ, we consider a constant energy value for different configurations because the FUs consume most of the transistors in a PE. In HERA, a PE template uses less than 10% of a PE's logic resources. The templates, Sequencer, and the CFBs are treated as FUs and their consumptions are evaluated by similar equations as for $E_{i,j,k}^{FU}$ except that they are in either the active or idle state (never in the standby or sleep state).

The energy consumption of all the BlockRAM blocks is counted into E_{MEM}. Based on our experiments in Section III, we identify energy consumption for three states: *idle*, *one-port access* (*acc*_1), and *simultaneous dual-port access* (*acc*-2). The total memory consumption is:

$$E_{MEM} = \sum_{l=1}^{n_m}(E_{mem}^{idle} * C_{l}^{idle} + E_{mem}^{acc\_1} * C_{l}^{acc\_1} + E_{mem}^{acc\_2} * C_{l}^{acc\_2}) \tag{3}$$

where n_m is the total number of BlockRAM blocks in the given FPGA, and E_{mem}^{idle}, $E_{mem}^{acc\_1}$, and $E_{mem}^{acc\_2}$ are the energy consumptions per clock cycle of a block in the respective states.

The NEWS interconnect and buses are implemented mainly with global routing fabric. A large part of FPGA power needs is due to routing resources [12-13]. Local routing resources are mainly used by the PEs and are counted in their power needs. We distinguish between two power states for the NEWS connections and buses: *idle* and *active*, and their total energy consumption can be found in a similar approach as that for FUs. The clock counts are collected at runtime as each component is equipped with appropriate hardware. The counters are read and reset by the host by using the *Configure* instruction. The bus activity information is monitored by the bus controller in the Sequencer. Each PE counts its own NEWS requests. All the counts are also indexed by task information in order to analyze the energy distribution among various tasks in the given application.

V. EXPERIMENTAL RESULTS

The parallel LU factorization of sparse Doubly-Bordered Block Diagonal (DBBD) matrices is employed in our experiments to evaluate the effectiveness of our model and show its application to performance-energy trade-offs when synthesizing a HERA system. LU factorization is a widely applied direct method to solve a system of simultaneous linear equations expressed by *Ax* = *b*. *A* is an *N* x *N* nonsingular matrix, x is a vector of N unknowns, and b is a given vector of length N. For sparse matrices in applications such as VLSI place-and-route and power-flow analysis, a node-tearing technique [16] can permute A into the DBBD form, as shown in Fig. 4. In the DBBD form, the A_{ik}'s represent matrix sub-blocks and all the non-zero elements in the matrix appear only inside these sub-blocks. The blocks $\{A_{ii}, A_i^L, A_i^U\}$, where $i \in [1, n]$, are said to form a 3-block group. A_F is known as the last block. In general, the parallel LU factorization of sparse DBBD matrices involves four tasks [2, 18]: (1) *FAC*: Independent factorization of all the 3-block groups. (2) *MAC*: Independent multiplication of the factored border block pairs ($L_i^{'}U_i^{'}$) and local accumulation of the partial products. Every resulting product has the same size as A_F. (3) *PAC*: Parallel accumulation of the partial products in all the PEs after no more FAC or MAC tasks are left. (4) *LAST*: Parallel LU factorization of A_F upon finishing all the other tasks.

$$\begin{pmatrix} A_{11} & \cdots & 0 & A_1^U \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & A_{nn} & A_n^U \\ A_1^L & \cdots & A_n^L & A_F \end{pmatrix} \Rightarrow \begin{pmatrix} L_1U_1 & \cdots & 0 & U_1^{'} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & L_nU_n & U_n^{'} \\ L_1^{'} & \cdots & L_n^{'} & U_F^{'}U_F^{'} \end{pmatrix}$$

Fig. 4. LU factorization of sparse DBBD matrices.

Four benchmark matrices from the Matrix Market [17] are used in the experiments. We always try to maximize the usage of the available resources in the given FPGA since HERA targets data-parallel applications, where more PEs can potentially reduce the execution time. The characteristics of the matrices after permuting are shown in Table II. The matrix blocks in the DBBD matrix are still sparse and have various percentages of non-zero elements resulting in different activity rates. We first evaluate the accuracy of our energy model. A fixed HERA system with 25 single-precision PEs running at 125MHz was used for this purpose. The results calculated by our energy model are compared in Table III with the XPower-reported results. The activity files for XPower of each benchmark are generated by ModelSim based on the full-scale VHDL simulation of the algorithm. Only the dynamic power is shown in these results. The idle PEs are switched into standby during execution. The four tasks are coded with HERA instructions and no device reconfiguration is required when switching between tasks, regardless of the matrix size. This saves significant device configuration energy which would be unavoidable if the PEs were not programmable. The number of the diagonal blocks (n) and the size of A_F have a larger impact on the reported differences as they represent more inter-PE communications. The average error is about 5.17%, which is an acceptable rate for fast system-level estimation models. The reasons for the differences are: (1) Only one power value is assumed for the FUs in the active state; (2) The average activity rate varies for different sets of data; we used a fixed rate instead as discussed earlier. (3) We concentrate on the major components and neglect other miscellaneous logic for system and PE control. (4) The energy measurements for the system buses tend to be less accurate than for the FUs due to coarse-grain power modeling. These choices are justified by our objective to develop fast, yet useful models for exploring static- and run-time performance-energy optimizations without involving time-consuming low-level simulations.

TABLE II
DBBD STATISTICS FOR BENCHMARK MATRICES FROM THE MATRIX MARKET

	IEEE1	*IEEE2*	*PSADMIT1*	*PSADMIT2*
N	118	300	494	1138
1*	3.42	1.24	0.68	0.31
n	11	18	27	67
2*	10	18	20	20
3*	5	8	11	4
F	18	31	45	100
4*	6x10**, 3x9, 1x8, 1x5	5x18, 6(17)**, 2x14, 4(11), 1x8	13(20), 8(15), 6x12, 1x11	22(20), 26(15), 18 (12), 1x4

**1: % of non-zero elements*
2: Dimension of the largest diagonal block
3: Dimension of the smallest diagonal block
4: Distribution of diagonal block sizes
**: 6x10 stands for 6 blocks of size 10 x 10
11(17) stands for 11 blocks of approximate size 17 x 17

We are also interested in the breakdown of the overall energy consumption among individual system components for different benchmarks. We focus in Fig. 5 on the normalized numbers for all the PEs, bus, NEWS connections, and the memory blocks when considering all the benchmarks. Such information can be used to optimize the architecture and algorithm. Note that our PEs are macro components which include significant interconnect resources. As previously, more diagonal blocks normally incur more communication that increases the significance of the energy consumed by the NEWS connections and bus.

TABLE III
COMPARISON OF THE MODELED AND XPOWER-REPORTED DYNAMIC ENERGY CONSUMPTIONS (mJ)

Benchmarks	*Modeled*	*XPower*	*Error (%)*
IEEE1	2.85	2.93	-3.66
IEEE2	15.67	16.40	-4.67
PSADMIT1	69.58	73.12	-5.10
PSADMIT2	196.25	210.50	-7.25

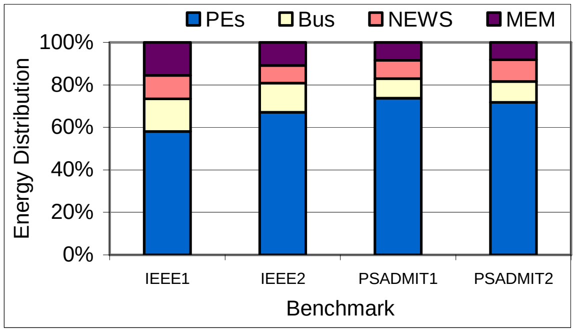

Fig. 5. Energy distribution among the major components in HERA.

An important advantage of our component-oriented energy model over instruction-oriented models is that it allows the exploration of architectural choices in meeting various performance-energy objectives during HERA's synthesis. We incorporated our energy model into our design methodology [19]; three realistic optimization scenarios are of particular interest here: (1) optimize the performance with no energy constraints; (2) optimize the performance with energy constraints; and (3) reduce the current energy cost for a given performance loss. Table IV shows performance-energy tradeoff results for *PSADMIT2*. In Scenario-II, we reduce the total energy consumption of Scenario-I by about 10.2% for a performance loss of 1.5%. An increase of 5.6% in the execution time is observed when reducing the energy consumption of Scenario-I by 20.3% (Scenario-III). In Scenario-IV, we relax the performance of Scenario-I by about 14.8%, which reduces the energy consumption by 28.3%. Generally, we achieve better performance for a larger input matrix because of more matrix blocks that can be manipulated in performance-energy tradeoffs.

TABLE IV
PERFORMANCE-ENERGY OPTIMIZATION FOR THE PSADMIT2 BENCHMARK

Scenario	*Objective*	*Constraints*	*Energy (mJ)*	*Execution Time (ms)*
I	Minimize T	None	196.25	12.67
II	Minimize T	E < 176.6	176.1	12.86
III	Minimize T	E < 157.0	156.4	13.38
IV	Minimize E	T < 14.57	140.7	14.55

VI. CONCLUSIONS

Reconfigurable chip multiprocessors exhibit distinct advantages and flexibility in matching their architecture with application idiosyncrasies at a very low cost. Ever-tightening energy budgets and the energy-hungry nature of FPGAs force us to take energy into consideration early in the design process. We have presented a system-level, component-oriented energy model for our HERA MPoPC. Both implementation measurements and application statistics are utilized by the model. Using a parameterized hardware component library with highly optimized FP FUs, our design exploration process saves costly physical-level simulation time associated with component-oriented models. Our model achieves an acceptable low error rate. As increased FPGA densities bring us close to high-performance designs, the importance of similar models will become more preeminent.

ACKNOWLEDGMENT

This work was supported in part by the U. S. Department of Energy under grant DE-FG02-03CH11171. The authors would like to thank the anonymous reviewers for their constructive comments that helped improve the quality of this paper.

REFERENCES

[1] T. J. Todman, G. A. Constantinides, S. J. E. Wilton, O. Mencer, W. Luk, and P. Y. K. Cheung, "Reconfigurable computing: architectures and design methods," *IEE Proc. Computers Digital Techniques*, vol. 152, no. 2, pp. 193-207, March 2005.

[2] X. Wang and S. G. Ziavras, "Exploiting mixed-mode parallelism for matrix operations on the HERA architecture through reconfiguration," *IEE Proc. Computers Digital Techniques,* vol. 153, no. 4, pp. 249-260, July 2006.

[3] R. Lysecky and F. Vahid, "A study of the speedups and competitiveness of FPGA soft processor cores using dynamic hardware/software partitioning," *IEEE Design Automation and Test in Europe (DATE)*, pp. 18-23, March 2005.

[4] A. Raghunathan, N. K. Jha, and S. Dey. *High-Level Power Analysis and Optimization*. Kluwer Academic Publishers, 1998.

[5] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: A first step towards software power minimization," *IEEE Trans. VLSI Systems*, vol. 2, no. 4, pp.437-445, Dec. 1994.

[6] A. Sinha and A. P. Chandrakasan, "JouleTrack-A web based tool for software energy profiling," *IEEE Design Automation Conf.,* June 2001.

[7] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," *IEEE Intern. Symp. Computer Archi.*, pp. 83-94, June 2000.

[8] W. Ye, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "The design and use of SimplePower: A cycle-accurate energy estimation tool," *IEEE Design Autom. Conf.*, pp. 340-345, June 2000.

[9] Y. Fei, S. Ravi, A. Raghunathan, and N. K. Jha, "Energy estimation for extensible processors," *IEEE Design, Automation, and Test in Europe (DATE)*, March 2003.

[10] J. Ou and V. K. Prasanna, "Rapid energy estimation of computations on FPGA based soft processors," *IEEE Intern. SoC Conf.,* Sept. 2004.

[11] M. Loghi, M. Poncino, and L. Benini, "Cycle-accurate power analysis for multiprocessor systems-on-a-chip," *ACM Great Lakes Symp. VLSI*, pp. 401-406, Apr. 2004.

[12] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic power consumption in Virtex™-II FPGA family," *ACM/SIGDA Intern. Symp. Field-Program. Gate Arrays*, pp. 157-164, 2002.

[13] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power modeling and characteristics of Field Programmable Gate Arrays," *IEEE Trans. CAD Integr. Circuits Systems*, vol. 24, no. 11, pp. 1712-1724, Nov. 2005.

[14] J. H. Anderson and F. N. Najm, "Power estimation techniques for FPGAs," *IEEE Trans. VLSI Systems,* vol. 12, no. 10, pp. 1015-1027, Oct. 2004.

[15] Virtex II FPGA datasheet, http://direct.xilinx.com/bvdocs/publications/ds031.pdf.

[16] A. Sangiovanni-Vincentelli, L. K. Chen, and L. O. Chua, "An efficient heuristic cluster algorithm for tearing large-scale networks," *IEEE Trans. Circuits Systems*, vol. 24, no. 12, pp. 709-717, Dec. 1977.

[17] Matrix Market, http://phase.hpcc.jp/mirrors/MatrixMarket/index.html.

[18] D. P. Koester, S. Ranka, and G. C. Fox, "Parallel block-diagonal-bordered sparse linear solvers for electrical power system applications," *IEEE Scalable Parallel Libraries Conference*, 1994.

[19] X. Wang and S. G. Ziavras, "A framework for dynamic resource management and scheduling on reconfigurable mixed-mode multiprocessors," *IEEE Intern. Conf. Field-Program. Tech. (FPT)*, Singapore, pp. 51-58, Dec. 2005.