Cycle-Accurate Energy Consumption Measurement and Analysis: Case Study of ARM7TDMI[*]

Naehyuck Chang Kwanho Kim Hyung Gyu Lee
School of CSE, Seoul National University, Korea
naehyuck@snu.ac.kr khkim@cslab.snu.ac.kr hglee@cslab.snu.ac.kr

ABSTRACT

We introduce an energy consumption analysis of complex digital systems through a case study of ARM7TDMI RISC processor by using a new energy measurement technique. We developed a cycle-accurate energy consumption measurement system based on charge transfer which is robust to spiky noise and is capable of collecting a range of power consumption profiles in real time. The relative energy variation of the RISC core is measured by changing the opcode, the instruction fetch address, the register number, the register value, the data fetch address, and the immediate operand value in each pipeline stage, respectively. We demonstrated energy characterization of a pipelined RISC processor for high-level power reduction.

1. INTRODUCTION

Power consumption analysis is the basis of high-level power reduction techniques because they do not rely on actual physical design. High-level power reduction of microprocessor-based systems saves power consumption by changing *energy-sensitive factors* such as instruction fetch addresses, opcode encoding, register encoding, data fetch addresses, immediate operands, *etc.* Some of the energy-sensitive factors have great degrees of freedom while others are more restrictive. Under certain circumstances, even data and instructions can be changed as far as the original semantic is preserved. Consequently, it is important to be informed of power consumption variations with respect to the energy-sensitive factors for setting up proper power reduction strategies. Previous power analyses, however, are not suitable for inspiring various high-level power reduction techniques. Rather, they have been mainly for estimation purposes. Consequently, power estimation has been used for performance evaluation of predefined power reduction schemes.

Power analysis can be performed by simulation-based or measurement-based approaches. Simulation-based power analysis is convenient as far as a simulation model is available because it does not necessitate a prototype. Simulation is preferable to avoid system dependent bias as power consumption is also variable to bus configuration and peripheral devices. Related studies built high-level processor simulators to replace low-level simulators and estimated average power consumption at reasonable complexity [1, 2, 3, 4]. Low-level power simulation often backs up the high-level simulation [5]. On the other hand, a black box model was introduced to overcome the availability of simulation models for peripheral devices [4]. For the most part, they do not furnish explicit information for high-level power reduction techniques: energy-sensitive factors versus energy consumption.

Measurement-based power consumption analysis is sometimes more feasible due to the availability of existing models even if a prototype is necessary. Even with a prototype, correct measurements are not easily obtainable because digital systems consume power in a spiky manner with over hundreds MHz in the power spectrum [6]. DMMs (digital multimeters) [7, 8] inform only average power due to the limited bandwidth. The oscilloscope overcomes this drawback [9], but the power calculation procedure is invariably error prone. They often measure power consumption of working prototype systems, which may bias the power consumption due to system-dependent peripheral devices. Most of all, these standard equipment-based methods are greatly time consuming thereby restricting the number of experiments and thus sufficient sample space for characterizing the power consumption.

Some previous work organized power consumption for high-level power reduction; the results are in the form of *instruction base cost* and *inter-instruction cost* [7, 8, 5]. Power reduction techniques in a DSP application has been demonstrated [8], which is simple, regular, and restrictive. Although this scheme is useful for average power estimation, it does not afford many alternative plans in power reduction. The average base cost and the average inter-instruction cost do not inform the power consumption variation due to major energy-sensitive factors such as addresses, data, register encoding, immediate operands, *etc.* An intensive simulation study introduced limited analysis of average power variation due to addressing modes and data bus activities [5]. Operand-dependent power analysis has been introduced with power cost of representative components [1]. This work is limited in respect to many significant components, and additionally the results associate different costs with the same components by instructions, which is invariably difficult to conform. There exist different abstractions of systems which can be useful for hardware designers [10, 11], and for higher level software such as power management [12, 13].

In this paper, we introduce a power analysis of microprocessors based on a new measurement method. We take into account all the factors that can be controlled by high-level power reduction tech-

[*] This work was supported by the Brain Korea 21 Project.

ISLPED '00, Rapallo, Italy.
Copyright 2000 ACM 1-58113-190-9/00/0007…$5.00.

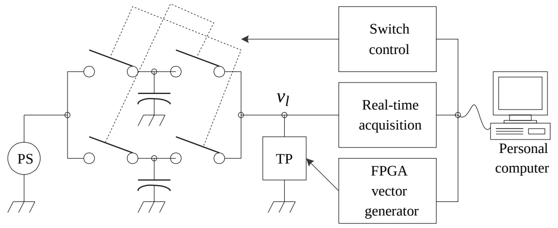


Figure 1: Real-time cycle-accurate energy measurement system. (PS: power supply, TP: target procesor)

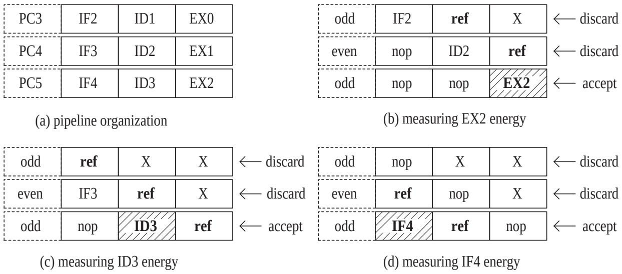
niques as energy-sensitive factors. We analyzed the energy consumption variation of each instruction per cycle with respect to the factors in each pipeline stage. We discovered the fraction of energy consumption that can be changed by the energy-sensitive factors as well as characterization of these factors. Our energy model describes not only the base and the inter-instruction costs but their composition with the functions of the Hamming distance and the weight of the energy-sensitive factors, while previous results inform their average values. Our result is a useful guideline for high-level power reduction techniques such as opcode or register re-encoding, address relocation, instruction re-scheduling, *etc*. We demonstrated the analysis method with a case study of ARM7TDMI RISC core. We built a testbed and performed a sufficient number of measurements in a short period; a real-time cycle-accurate energy consumption measurement system made it possible.

2. REAL-TIME CYCLE-ACCURATE ENERGY CONSUMPTION MEASUREMENT

2.1 Principle of operation

Synchronous state machines include most microprocessor-based systems. The state is a useful abstraction of the behavior and thus a base unit of energy consumption. A time unit shorter than the clock cycle is not capable of furnishing more useful information for architectural or behavioral level power analysis of synchronous state machines.

In this paper, we measure cycle-accurate energy consumption of synchronous state machines using switched capacitors [14]. The switch pairs (connected to dash-lines) alternatively repeat an on/off action. Each capacitor charges for one clock and discharges for another. The energy consumption for a clock cycle is given by $\frac{1}{2}\max(v_l)^2 - \frac{1}{2}\min(v_l)^2$. Figure 1 illustrates the measurement system. The real-time acquisition unit samples the $\max(v_l)$ and the $\min(v_l)$ after every clock transition and sends them to a personal computer for further analysis. It has many advantages over the previous methods. We can measure the exact energy consumed for a clock cycle of CMOS circuits with one sampling per clock cycle because the v_l remains stable when the circuit becomes stable, finishing the transition propagation. The existing methods measure the voltage across a series resistor in the power supply line. The power spectrum of the voltage across the resistor is dominant up to $\frac{1}{2t_f}$ where t_f is the shortest fall time of the signal and is often 2ns or less [6]. Thus one must sample the voltage in a very high sampling rate for reasonable accuracy. This dramatically increases the analysis time as well as the measurement time. In contrast, our method is robust to dynamic current change because v_l goes to a stable state when the acquisition unit samples it [14].



PC3	IF2	ID1	EX0
PC4	IF3	ID2	EX1
PC5	IF4	ID3	EX2

(a) pipeline organization

odd	IF2	**ref**	X	⟵ discard
even	nop	ID2	**ref**	⟵ discard
odd	nop	nop	▨ **EX2**	⟵ accept

(b) measuring EX2 energy

odd	**ref**	X	X	⟵ discard
even	IF3	**ref**	X	⟵ discard
odd	nop	▨ **ID3**	**ref**	⟵ accept

(c) measuring ID3 energy

odd	nop	X	X	⟵ discard
even	**ref**	nop	X	⟵ discard
odd	▨ **IF4**	**ref**	nop	⟵ accept

(d) measuring IF4 energy

Figure 2: Pipeline set up for measuring each pipeline stage (PC independent).

2.2 Experimental setup for ARM7TDMI

The target processor, in this paper, is an ARM7TDMI [15] test chip[1] manufactured for experimental purpose. Conventional processor boards may have differently loaded memory buses though the target processors are the same. This may result in measuring power consumption variation mainly in the bus rather than by processor. In this case, each instruction may show a distinct average power consumption with small variance. One may think the experiment is successful, but the measurement data exaggerates the effect of instruction encoding and the fetch addresses.

Like other recent microprocessors, there are separate power supply pins for the core enabling measurement free from system dependent bias. Our measurement tool is also designed to minimize the bus effect equipping bus switches and an FPGA vector generator in case the target processor does not have separate power supply pins. The address, the data, and all the control pins are connected to the FPGA vector generator that is capable of controlling the target processor with a great degree of freedom; *e. g.*, we can make the processor perform continual branch instructions to arbitrary address locations. We cross-compile an ARM7 program for a proper pipeline setup (Figure 2) and download the binary image to the FPGA vector generator. With only a few mouse clicks, we can upload the power consumption profile from the measurement system. We analyzed power consumption by the use of a spread sheet with user programed macro functions.

2.3 Energy measurement of pipelined microprocessors

Common RISC processors, including ARM7TDMI, have an *opcode*, (a) *source register number(s)*, a *destination register number*, and an *immediate operand value* in their instruction formats. Low-energy software reduces power consumption by controlling these factors. The internal state of the processor (*data stored in registers*) affects the power consumption of the datapath components, the instruction fetch, and the *load*/*store* operations. Figure 2 illustrates the pipeline setup for measuring energy consumption in each pipeline stage. We measure the *energy variation* of the shaded part changing the above factors with respect to various *reference* instructions. We can easily control the weight of the current instruction. We control the Hamming distance with careful adjustment of the reference and the current instructions.

[1] manufactured by EPSON.

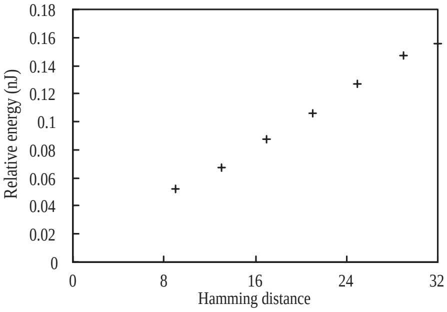


Figure 3: Energy consumption by the instruction fetch address (PC stage).

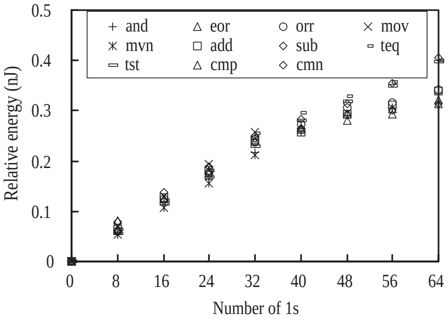


Figure 4: Energy consumption by the operand value (EX stage).

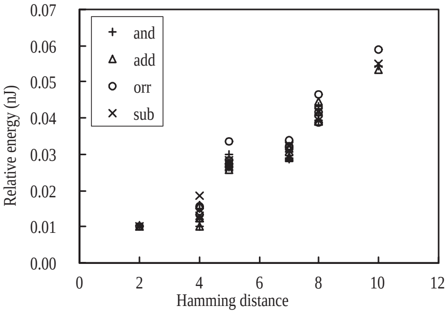


Figure 5: Energy consumption by the register number (EX stage).

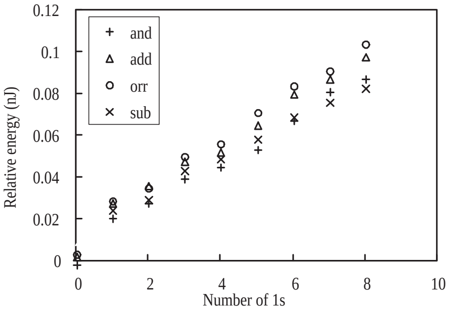


Figure 6: Energy consumption by the immediate value (EX stage).

The pipeline setup for the EX stage is the simplest. We chose a reference instruction for the ID stage keeping the same base cost in the EX stages or compensating the base cost. We set up the pipeline for the IF stage by filling nop instruction in the EX stage. The reference instruction needs to sustain the same ID stage energy. The Hamming distance between the instruction fetch address values also produces bias. We located the measuring points in the even address space (PC stage) fixing the Hamming distance to one. There is not an explicit PC stage in ARM7TDMI. But the instruction fetch address is issued at Phase 2 of the previous cycle, and thus we distinguished the PC stage from the IF stage during the measurement and the analysis. We go by the relative energy consumption in this paper. The smallest value is used as a base in each analysis. Relative power consumption is more important in RTL level design [16].

3. CASE STUDY OF ARM7TDMI CORE

3.1 PC stage

We measured the PC stage energy supplying the same instructions repeatedly to the processor. We found that the Hamming distance between previous and current instruction fetch address values is a major concern of the energy consumption. The maximum variation is up to 0.15nJ as shown in Figure 3.

3.2 EX stage

We first measured the energy variation due to the register values over 11 instructions. We found that the energy consumption is proportional to the number of 1's in the value. This is understandable because of the dynamic CMOS configuration of ARM7TDMI. Figure 4 illustrates that the energy consumption shows consisteny regardless of the opcodes. The variation is large: up to 0.4nJ. Secondly, we measured energy variation by the register numbers. We measured data processing instructions and found consistency in results. Figure 5 shows that the EX stage energy is proportional to the Hamming distance between the register numbers in previous and current instructions. Thirdly, we observed that the immediate operand value also affects the EX stage energy. The trend is similar to the register values as shown in Figure 6.

Finally, we measured the EX stage energy by each instruction keeping other factors the same. We repeated the measurement with four reference instructions. Unique base energy cost is associated with each instruction regardless of the reference instruction, and the portion is significant (Figure 7). As described in existing work, the base cost is useful for power estimation. However, for reduction purposes, it is less important because we have little alternatives with the base cost in high-level approaches than other factors of the EX stage.

3.3 ID stage

It is more difficult to find regularity in the ID stage. First, we observed that the register number significantly affects the energy consumption and that it is proportional to the Hamming distance between previous and current instructions as illustrated in Figure 8. We also measured the base cost of the ID stage energy with four reference instruction followed by other instructions. We had unique base costs of the ID stage energy by the opcodes (Figure 9). The values are not in accordance with those of the EX stage. The base costs are less important than energy variation due to other energy-sensitive factors in high-level power reduction. Figure 10 shows that the immediate operand value also affects the ID stage energy.

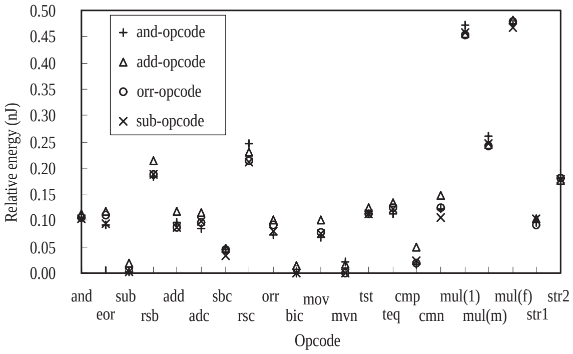


Figure 7: Energy consumption by the opcode (EX stage).

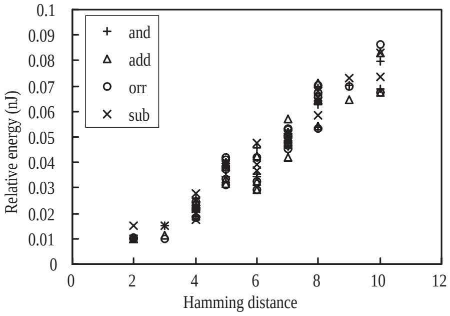


Figure 8: Energy consumption by the register number (ID stage).

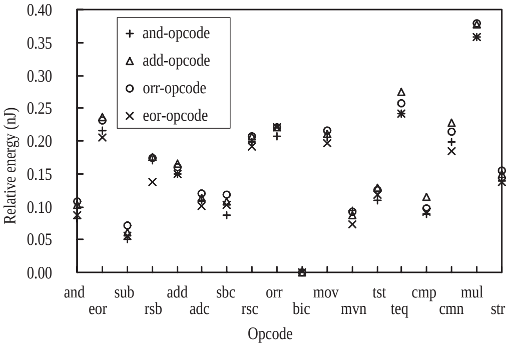


Figure 9: Energy consumption by the opcode (ID stage).

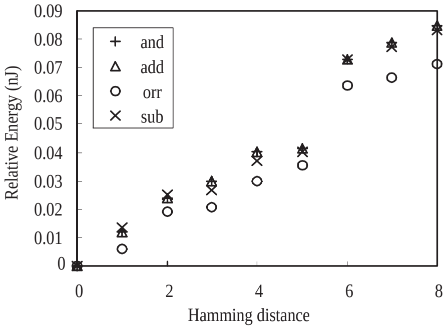


Figure 10: Energy consumption by the immediate operand value (ID stage).

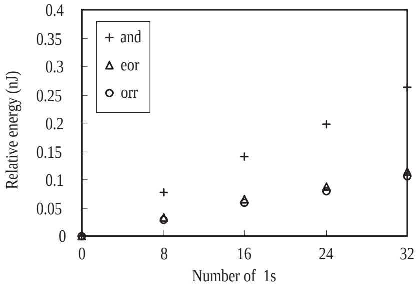


Figure 11: Energy consumption by the register value (A port registers, ID stage).

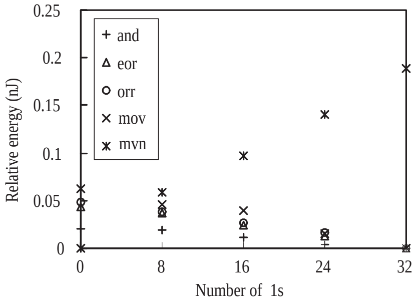


Figure 12: Energy consumption by the register value (B port registers, ID stage).

It is proportional to the Hamming distance between previous and current instructions.

We observed an energy characteristic that is not in accordance with the literature describing ARM7TDMI. The ID stage energy is significantly affected by the register values. The energy consumption is proportional in *A bus* and mostly inversely proportional in *B bus* to the number of 1's in the value as shown in Figures 11 and 12.

3.4 IF stage

We observed that the opcode encoding affects the IF stage energy as shown in Figure 13. Four instructions were used as reference in measuring the energy difference by opcodes. The IF stage energy is proportional to the Hamming distance between the opcodes but not significantly.

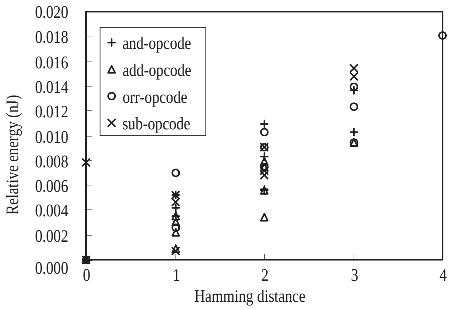


Figure 13: Energy consumption by the opcode encoding (IF stage).

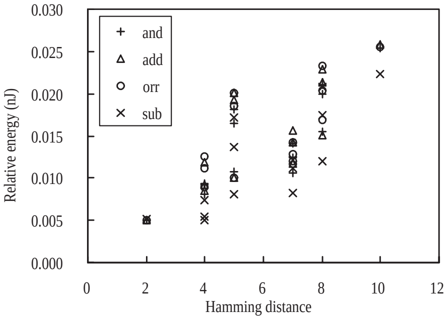


Figure 14: Energy consumption by the register number (IF stage).

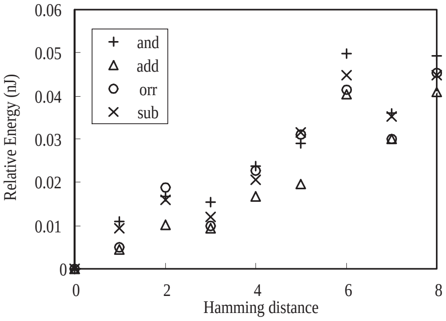


Figure 15: Energy consumption by the immediate operand value (IF stage).

Figure 14 illustrates the energy variation due to the Hamming distance between the register numbers in previous and current instructions. The amount is not significant. Figure 15 shows that the immediate operand value affects the IF stage energy. We observed that the cost is marginally proportional to the Hamming distance between current and previous instructions, but this influence is minor.

3.5 Multi-cycle instructions

Multi-cycle instructions occupy more than two EX stage cycles while causing other stages to stall. Figure 7 shows the base cost of `str` and `mul` instructions, for the first, middle (one or more), and the last cycles. The first EX cycle of `str rd,(rs1),rs2` instruction transfers the effective memory address to the *address register*, and the energy cost obeys Figure 4 by the values of `rs1` and `rs2`. The second EX stage performs a memory operation, and the cost obeys Figure 3 by the Hamming distance between the effective address value and the previous instruction fetch address. The number of EX cycles of `mul` instruction is dependent on the data (Booth algorithm). Although the EX stage energy of `mul` is not significantly variable, it does not agree with Figure 4.

3.6 Example of energy consumption modeling

While characterization plays an important role in the existing power analysis work [17, 18, 19, 20], a simple characterization method is often suitable for complex systems including microprocessors. We observed that the power consumption is proportional or inversely proportional to the Hamming distance between previous and current values, or the number of 1's in the current value. We introduce an example of energy model that characterizes the power consumption by first order linear functions. We took the dynamic CMOS into account and thus the Hamming distance together with the number of 1's in the current value. Our results are reasonable because each factor is characterized with each unique function for all the instructions except for the base cost of opcodes in the ID and the EX stages.

Table 1: Relative base cost of instructions.

opcode	ID stage (nJ)	EX stage (nJ)
and	0.10	0.10
eor	0.22	0.1
sub	0.06	0.02
rsb	0.17	0.20
add	0.15	0.10
adc	0.10	0.10
sbc	0.10	0.03
rsc	0.19	0.23
orr	0.22	0.08
bic	0.00	0.00
mov	0.21	0.09
mvn	0.09	0.02
tst	0.12	0.11
teq	0.26	0.12
cmp	0.09	0.03
cmn	0.20	0.23
mul (1st)	0.38	0.46
mul (middle)	N/A	0.25
mul (lat)	N/A	0.48
str (1st)	0.15	0.10
str (last)	N/A	0.18

Table 2: Relative energy consumption model (h: Hamming distance, w: weight).

factor	IF stage		ID stage		EX stage	
	E (pJ)	%	E (pJ)	%	E (pJ)	%
opcode	4.5**h**	1.58	Tab. 1	33.3	Table 1	40.4
reg. #	2.5**h**	2.63	7.5**h**	7.9	5.4**h**	5.7
reg. val	0	0	Fig. 11 and 12		6.7**w**	37.6
IF addr	5.3**h**	14.9	0	0	0	0
DF addr	0	0	Fig. 11 and 12		5.3**h**	14.9
IMM val	6.2**h**	5.4	1.0**h**	7.0	1.13**w**	7.9

We defined a function **h** as the Hamming distance between current and previous values, and a function **w** as the number of 1's in the current binary number. We formulated a hypothesis that the power consumption of each pipeline stage is given by $\alpha\mathbf{h}+\beta\mathbf{w}+\gamma$ where α, β and γ are non-negative real numbers. We include the PC stage in the IF stage for convention. We ignore γ, in this paper, because relative energy consumption is meaningful in high-level power reduction. Table 1 shows the base costs of the ID and the EX stages. These values do not change by the instructions in the other pipeline stages. However, the same instruction may consume different energies according to Table 2. We can explain that the *inter-instruction costs* are determined by Table 2. The order of the table size is lower than the existing inter-instruction approaches. However, it offers much more information for various software-level power reduction techniques because each cost is not a constant but a function of the Hamming distance or the weight.

Multi-cycle instructions have different base costs for each cycle. Other pipeline stages are stalled during the middle and the last cycles, but still consume significant amount of energy. Consequently, it would be better to regard the actual energy as the base cost plus the average energy consumption by the entire pipeline stages[2].

4. DISCUSSION

Conventional processor boards are composed of memory subsystems and many other peripherals. The best way to remove system dependent bias during measurement is to use a processor board solely composed of a microprocessor core. This is almost impossible in real systems, but we can set up a test environment as presented in this paper.

Table 2 shows that inter-instruction cost is caused by various energy-sensitive factors and the resulting amount is large. The average inter-instruction costs, introduced in previous work, take into account the effect of the opcode only, which mainly affects the IF stage energy variation and occupies under 2 % out of total variation. Other factors are orthogonal to the instruction and much more significant. This shows that the average inter-instruction cost is not a suitable arrangement.

Our power consumption model informs us of various software power reduction schemes. For example, the power consumption model explains that the instruction fetch energy can be optimized by reducing the Hamming distance between the address values. We are also able to see that the reduction amount will be 5.3pJ per one Hamming distance. In addition, the address bus encoding only affects the IF stage energy. The possible energy reduction in the best case will be 15 % of the core energy. The PC and the IF stage energy becomes more significant in system-level power consumption because of bus and peripheral devices. Their energy characteristic may be different from that of the processor. We can also estimate the effectiveness of the register re-encoding scheme. The register ID may change the power consumption of the IF, ID and EX stages up to 2.63%, 7.9% and 5.7%, respectively. With 30% Hamming distance reduction, we can achieve 5% reduction of the CPU core power. A simple calculation shows that power consumption of the same instruction may differ up to 120 %. Let us assume that there is little degree of freedom in changing data in software power reduction; there still remains up to 80 % power.

5. CONCLUSION

We analyzed energy consumption of the ARM7TDMI core in terms of opcodes, register numbers, register values, instruction fetch addresses, data fetch addresses, and the immediate operands in each pipeline stage, respectively. Most of them are dependent on the Hamming distance between the values in current and previous cycles or the number of 1's in the current value. We also observed that each instruction has a base energy cost in the ID and the EX stages, which is not variable to the previous pipeline status. Generally the base cost does not give enough degree of freedom to low-power software designers because they have little alternatives in most cases. We characterized the power consumption variation with respect to the factors that are dependent on the Hamming distance and the number of 1's, introducing substantial power reduction guidelines.

The real-time cycle-accurate energy consumption measurement technique has made it possible to discover the energy consumption with a large number of input vectors in a short period. Future work will include analysis of DSPs and static CMOS processors. Our measurement technique does not limit energy characterization to the example in this paper. And, we are developing various applications of the measurement technique.

[2] We let this value 1.144nJ: the average of the entire measurement data.

6. REFERENCES

[1] Davide Sarta, Dario Trifone, and Giuseppe Ascia, "A data dependent approach to instruction level power estimation," in *Proceedings of IEEE Alessandro Volta Memorial Workshop on Low-Power Design*, 1999, pp. 182 –190.

[2] R. Yu Chen, M. J. Irwin, and R. S. Bajwa, "An architectural level power estimator," in *Proceedings of ISCAW*, June 1998.

[3] R. Yu Chen, R. M. Owens, M. J. Irwin, and R. S. Bajwa, "Validation of an architectural level power analysis techinque," in *Proceedings of 35th Design Automation Conference*, June 1998, pp. 242–245.

[4] Tajana Simunic, Luca Benini, and Giovanni De Micheli, "Cycle-accurate simulation of energy consumption in embedded systems," in *Proceedings of 36th Design Automation Conference*, June 1999, pp. 867–872.

[5] Peggy Laramie, "Instruction level power analysis and low power design methodology of a microprocessor," in *Master Thesis, U. C. Berleley*.

[6] Howard W. Johnson and Martin Graham, *High-Speed Digital Design a Hand Book of Black Magic*, Prentice-Hall Inc., 1993.

[7] V. Tiwary, S. Malik, and A. Wolfe, "Power analysis of embedded software: A first step towards software power minimization," *IEEE Transaction on VLSI systems*, vol. 2, no. 4, pp. 437–445, December 1994.

[8] Mike Tien-Chien, Vivek Tiwari, Sharad Malik, and Masahiro Fujita, "Power analysis and low-power scheduling techniques for embedded DSP software," in *Proceedings of the Eighth International Symposium on System Synthesis*, 1995, pp. 110–115.

[9] J. Russel and M. Jacone, "Software power estimation and optimization for high perfomance, 32-bit embedded processors," in *International Conference on Computer Design*, October 1998, pp. 328–333.

[10] Jon Bradley, "Calculation of TMS320C5x power dissipation application report," in *http://www.ti.com/sc/docs/psheets/abstract/apps/spra030.htm*, 1993.

[11] Tom Burd and Brad Peters, "A power analysis of a microprocessor: A study of an implementation of the MIPS R3000 architecture.," in *ERL Technical Report, http://bwrc.eecs.berkeley.edu/burd/gpp*, 1994.

[12] Jacob Lorch and Alan Jay Smith, "Energy consumption of apple macintosh computers," *IEEE Micro*, vol. 18, no. 6, pp. 54 – 63, November/December 1998.

[13] Andre Wolfe, "Opportunities and obstacles in low-power system-level CAD," in *Proceedings of the 33rd annual conference on Design automation conference*, 1996, pp. 15–20.

[14] Naehyuck Chang and Kwan-Ho Kim, "Real-time per-cycle energy consumption measurement of digital systems," to appear in *IEE Electronics Letters* (related technical report in http://www.power-reduction.com), 2000.

[15] Steve Furber, *ARM System Architecture*, Addition-Wesley, England, 1997.

[16] Alessandro Bogiolo, Luca Benini, and Giovanni De Micheli, "characterization-free behavioral power modeling," in *Design, Automation and Test in Europe*, 1998, pp. 767 –773.

[17] S. Gupta and F. N. Najm, "Energy-per-cycle estimation at RTL," in *International Symposium on Low Power Electronics and Design*, Aug 1999, pp. 121–126.

[18] Qing Wu, Qinru Qiu, Massoud Pedram, and Chih-Shun Ding, "Cycle-accurate macro-models for RT-level power analysis," *IEEE Transactions on VLSI Systems*, pp. 520 – 528, December 1998.

[19] Roberto Corgnati, Enrico Macii, and Massimo Poncino, "Clustered table-based macromodels for RTL power estimation," in *Proceedings of Ninth Great Lakes Symposium on VLSI*, 1999, pp. 354 –357.

[20] Muhammad M. Khellah and M.I. Elmassry, "Effective capacitance macro-modeling for architectural-level power estimation," in *Proceedings of the 8th Great Lakes Symposium on VLSI*, 1998, pp. 414 – 419.