RESEARCH ON A LOW-POWER MCD TECHNIQUE BASED ON EPIC

Rong Ji, Liang Chen, Yongwen Wang, Xianjun Zeng, Junfeng Zhang
School of Computer Science
National University of Defense Technology
Changsha, Hunan Prov. China, ZIP:410073
E-mail: rongji@nudt.edu.cn

KEYWORDS

EPIC, MCD, GALS, low-power, clock network

ABSTRACT

With the development of very large scale integration (VLSI), the clock frequency of microprocessors rapidly increases, which brings the significant challenge for the microprocessors' power. The multiple clock domain (MCD) technique is a new technique to compromising between synchronous systems and asynchronous systems to reduce the power. Most present studies of MCD are only based on superscalar architectures. In this paper, a MCD technique based on explicitly parallel instruction computing (EPIC) is designed and implemented. Furthermore, a series of experiments on our design have been done to evaluate it. The result of the experiments show that, an EPIC microarchitecture based on MCD technique with a fine-grained adaptive dynamic adjustment algorithm, can effectively decrease the microprocessor power by 40%, compared with the conventional EPIC processor with only one clock domain.

1 INTRODUCTION

With the development of VLSI, the clock frequency of microprocessors rapidly increases, which brings significant challenge for the microprocessors' power. The clock network can dissipate 20–50% of the total power on a chip [1].Thus the optimization of the clock network becomes one of the important issues of high-performance microprocessor design.

At present, most microprocessors are implemented by using fully synchronous mode. The clock distribution network is designed very carefully to meet the constraints of clock skew. It contributes to the complexity of clock interconnection and the significant increase of microprocessor power. So the designers present asynchronous system which need not clock. But there are so many difficulties in design of the complete asynchronous signals. Globally Asynchronous Locally Synchronous (GALS) [2], which is a compromise between asynchronous systems and synchronous systems, has been focused on.

Almost current researches on GALS are based on superscalars [3,4,5,6,7,8], and not been applied to EPIC architecture, because there are much difficulties in its implementation. For example, the uncertainty of asynchronous communication brings significant challenges for data dependence and instruction completion in order in EPIC.

This paper describes the research on MCD based on EPIC is done with GALS style, which includes the implementation of it, and experimental simulation to evaluate it. Section 2 presents the conventional EPIC architecture, and analyzes what limits power optimization of microprocessor. Section 3 describes the design and implementation of the MCD based on EPIC. Section 4 details the simulation framework and experimental setup. The results of all the tests are given and analyzed in Section 5. Conclusions and future work are in Section 6.

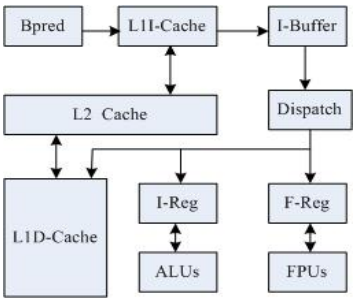
2 THE CONVENTIONAL EPIC ARCHITECTURE

2.1 The Conventional EPIC with One Clock Domain

An EPIC architecture emphasizes co-operating compiler with hardware to exploit instruction-level parallelism (ILP). It combines the advantages of superscalar architectures and the advantages of very long instruction word (VLIW) architectures [9]. Itanium 2 [10] is a classical processor based on EPIC architecture. Thus the EPIC architecture of Itanium 2 processor is a comparable architecture to our design.

Figures 1 shows a conventional EPIC architecture with single clock domain. The front-end of the microprocessor obtains the addresses of fetching instructions, by means of Bpred predicting branch objects. The instructions are fetched from instruction Cache (I-Cache), and reserved into instruction buffers (I-Buffer). They are decoded by the dispatch logic (Dispatch), and sent to corresponding execution units. ALUs and FPUs can execute multiply independent integer instructions and floating-point instructions per cycle, respectively. The operands are respectively given by integer registers (I-Reg) and floating registers (F-Reg). The results are reserved in corresponding registers. Load/store operations will access Level 1 data Cache (D-Cache) and level 2 Cache (L2 Cache). There are abundance of functional units and Cache levels to provide adequate hardware supports for exploiting ILP.

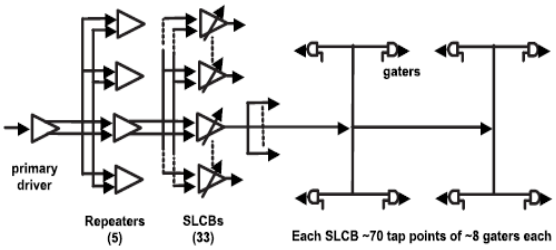
Proceedings 21st European Conference on Modelling and Simulation
Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007
ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD)



Figures 1: A Conventional EPIC Architecture

2.2 The Analysis What Limits Power Optimization of the Conventional EPIC Microprocessors

Currently the microprocessors based on the conventional EPIC architecture often use the globally synchronous clock strategy, such as the clock distribution network of the Itanium 2 processor described as Figures 2. The clock signal is generated by a clock generator, and is sent to each unit within the microprocessor by the long wires and a lot of buffers. All units synchronously operate under control of this global clock signal. The clock distribution network is usually designed carefully to meet the constraints of clock skew. It contributes to the complexity of clock interconnection and the significant increase of microprocessor power. Thus, it is necessary for EPIC architecture to implement MCD with GALS style, in order to decrease the microprocessor power.



Figures 2: The Clock Distribution of Itanium 2

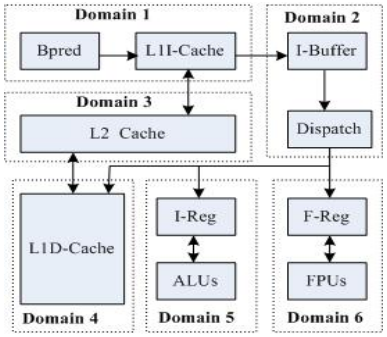
3 THE DESIGN AND IMPLEMENTATION OF MCD BASED ON EPIC

A MCD architecture based on EPIC, which need no global clock, is called MCDE (multiple clock domain based on EPIC). The MCDE microprocessor can be divided into several clock domains. Each domain uses an independent clock. Among different domains, there is asynchronous communication which can effectively save the power of microprocessors.

3.1 The Partition of Clock Domains

In MCDE, the entire chip can be divided into several domains according to the EPIC architecture characteristic of the microprocessor. There are two rules in the partition: 1) this partition can't change the organization structure of the microprocessor pipelines too much; 2) the domain boundaries are set between the components having a loose coupling with each other to the best of our abilities. The components having a close coupling with each other are placed in the same domain to reduce communication operations among different domains

The EPIC microprocessor is partitioned to six clock domains according to the rules described above: a fetching instruction domain (Domain1), a dispatch domain (Domain2), a L2 Cache domain (Domain 3), a load/store domain (Domain4), an integer domain (Domain5), and a floating-point domain (Domain6). Figures 3 shows the partition detail. The function of Domain1 contains branch prediction, instruction address generation, and I-Cache read. Domain2 accomplishes dispatch of instructions. Domain3 includes L2 Cache read/write operation. L1D-Cache read/write is completed in Domain4. Domain5 completes the load/store operation of integer operands, and execution of arithmetic logic. Domain6 consists of the load/store operation of floating-point operands, and execution of floating-point computing. In our design, Domain1 and Domain2 are called front-end. Domain3, Domain4, Domain5, and Domain6 are called back-end. Each domain has its own local clock. The units within same domain operate in synchronous mode. The asynchronous communication is used among different domains.

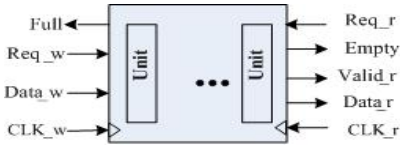


Figures 3: The Clock Domain Partition of EPIC

3.2 Communication Mechanisms among Domains

Within the microprocessor, the queue structures are synchronous points of the different domains. In this case, when the queues are not full and not empty, the latencies of synchronous overhead can be hidden. The design of the queue structures is based on the hybrid clock FIFO, which is shown as Figures 4. The FIFO queue has two ports: input and output. The function of input is to write data to the FIFO, which is controlled by *CLK_w*. *Req_w* is the control signal of write requirement. *Data_w* is the input data bus. When the FIFO is full, the signal *full* is set one. The function of the output of the FIFO is to read data from the FIFO, which is controlled by *CLK_r*. *Req_r* is the control

signal of read requirement. *Data_r* is the output data bus. *Valid_r* is the valid bit of output data.



Figures 4: The Hybrid Clock FIFO

The process of data stored at input is done as follows: When the rising edge of *CLK_w* arrives, *Req_w* that is the requirement signal and the data in *Data_w* are sampled. The data items will be written into the queue at the next clock period. If the *FIFO* is full, the signal *Full* is set before the next clock period arrives, and receiving the data at the input is prohibited.

The process of reading data from the output is described as follows: When the rising edge of *CLK_r* arrives, *Req_r* that is the requirement signal is sampled. The data items will be put on *Data_r* bus, and the valid bit *Valid_r* is set. If the *FIFO* is empty at this clock period, the signal *Empty* is set, and reading operation at the output is prohibited until the *FIFO* is not empty.

3.3 A Dynamic, Adaptive Control Algorithm of Clock Domain's Frequency

By analyzing processor resource utilization, a correlation is revealed, over an interval of instructions, between the valid entries in the input queue and the desired frequency for the domain. Queue utilization is thus an appropriate metric for dynamically determining the desired domain frequency. The dynamic, adaptive control algorithm of clock domain's frequency is based on this idea to reduce the power consumption of the clock network.

The dynamic, adaptive control algorithm of clock domain's frequency is described as follows: The MCDE architecture uses the attack/decay algorithm independently in each back-end domain. When the entries in the domain issue queue is in excess of 10,000-instruction interval, the hardware counts. Using the number and the corresponding number from the previous interval, the algorithm determines whether there is a significant change that threshold is 1.7 percent, in which case the algorithm uses the attack mode: The frequency changes by 7 percent. If no significant change occurs, the algorithm uses the decay mode: It reduces the domain frequency slightly by 0.17 percent.

But when the instructions per cycle (IPC) changes by more than a certain threshold that is 2.3 percent, the frequency remains unchanged for that interval. This rule is used to identify natural decreases in performance that are unrelated to the domain frequency, and to prohibit the algorithm from reacting to them.

4 THE SIMULATION FRAMEWORK AND EXPERIMENTAL SETUP

4.1 The Power Consumption Evaluation Model

To evaluate our MCDE architecture, we use basic technology parameter of CACTI 0.8μm, and scaling down method to implement power consumption evaluation model [11].The delay components that make up a general system are composed of the following three individual subsystems:

- **Memory Storage Elements**: Those elements include the Cache, the registers, all kinds of queue, and buffers in the processor. It is divided into five parts: the address decoders (*dec*), the bit lines (*bl*), the word lines (*wl*), the sense amplifiers (*amp*), and the route components (*rt*) of address and data bus. Thus, P_{Mem}, the peak power of memory is the sum of the maximum power of these five parts:

$$P_{Mem} = P_{dec} + P_{bl} + P_{wl} + P_{amp} + P_{rt}$$

Assume that, C represents the capacitance of each kinds of memory; V and f denote the working voltage and frequency, respectively; V_{dd} is the voltage supply; the maximum toggle rate of the decoder signal is 0.3, that of other parts is 1.

The power of the address decoder P_{dec} is given by

$$P_{dec} = 0.3C_{dec}V^2 f$$

The power of the word line P_{wl} is given by

$$P_{wl} = C_{wl}V^2 f$$

The power of the bit line P_{bl} is given by

$$P_{bl} = C_{bl}V^2 f$$

The power of the sense amplifier P_{amp} is computed by the empirical formula.

$$P_{amp} = 0.5C\frac{V_{dd}}{8}\times 10^{-3}$$

The power of the route component P_{rt} is given by

$$P_{rt} = C_{rt}V^2 f$$

- **Logic elements**: Those elements include such units as ALU and floating-point unit. For the power evaluation of the logic elements, we first use the Wattch power model [12], and the capacitance value of the basic device and wire under technology parameter of CACTI 0.8μm [13] to compute the reference power P_P, on condition that reference voltage is V_P and reference frequency is f_P. Assume that, the characteristic

size of the actual technology of the microprocessor is f_s. According to the scaling down theory of CMOS [14], for the given voltage and frequency, the power of the logic element is computed by

$$P_{\log ic}(f_s, V, f) = \frac{V^2 f}{\frac{0.8}{f_s} V_P^2 f_P} P_P$$

- **Clock network**: Assume that, the microprocessor is divided into M domains; For the i*th* domain, the capacitance of the local clock wire is $C_w(CL_i)$, the load capacitance of the local clock is $C_t(Cp_i)$, the load capacitance of the local clock buffer is $C_t(B_i)$, the load capacitance among the local pipelines is $C_t(P_i)$, the working voltage and frequency is V_i and f_i, respectively. The peak power of the clock network is given by

$$P_{mclk}(f_s, V, f, M) = \sum_{i=1}^{M} (C_w(CL_i) + C_t(Cp_i) + C_t(B_i) + C_t(P_i)) V_i^2 f_i$$

If M = 1, it represents the peak power of the clock network with single clock domain.

After the peak power of each part is evaluated, the performance simulator is run to obtain dynamically the statistic of the access counts of each component per cycle. The dynamic power is computed according to the statistic and the peak power of each part.

4.2 The Simulation and Experimental Setup

Since IMPACT provides a comprehensive infrastructure for modeling and simulation of EPIC microarchitecture feature, the Lsim simulator [15] of the IMPACT compile framework is adopted as the simulation engine. Furthermore, to simulate the MCDE processor, the event-driven algorithm based on multi-clock domains is designed, described as Algorithm 1.

```
Init_event_queue();                    /*  initialize the event queue  */
while ((event_queue is not empty) && (simulation is not finished))
  Get the id number(id), cycle time(cycle),
                          and trigger_time(trigger_time);
  timer = trigger_time;    /*  push ahead with the global timer  */
  event_handler();         /*  call the the handler of this event  */
  next_time = timer +cycle;
                  /*  arrange next occurrence time of this event  */
  Attain the tail pointer pt of the current queue;
  while( (next_time < the occurrence time pressed toward by pt)
        || (the priority of the current event is more prior))
    pt = pt→prev_event;    /*  move the pointer ahead  */
  if (pt is empty)
     Take error alarm: the event occurrence time is passed;
  else
     Insert the current_event_node subsequent to the node
                                    pressed toward by pt.
```

Algorithm 1: The Event-driven Simulation Algorithm

To evaluate objectively the MCDE proposed, we design three groups of experiments:

- **SVF Strategy**: Each of clock domains works on condition that the voltage is same, and frequency is also same, called SVF for simple,which represents same voltage and frequency. That is to verify the performance and power consumption characteristic of MCDE relative to the EPIC.

- **DVF Strategy**: Each of clock domains works on condition that the voltage may be different, and frequency may be different, too. The frequency and voltage are not adjusted during the system running. This strategy is called DVF for simple, which denotes different voltage and frequency. It can verify advantages of fine-grained setting of voltage and frequency in MCDE. In this group of experiment, the voltage and the frequency of each of domains are set as Table 1, according to the architecture characteristic of Itanium 2.

Table 1: Different Voltage and Frequency Setting

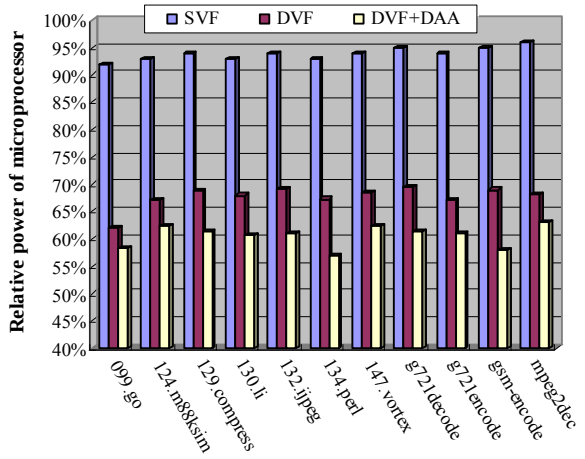
Domain	Frequency (MHz)	Voltage (V)
Domain1	1000	1.5
Domain2	1000	1.5
Domain3	100	0.8
Domain4	500	1.1
Domain5	500	1.1
Domain6	0	0

- **DVF+DAA Strategy**: Each clock domains works on condition that voltage and frequency maybe different with each other. In this group of experiment, the voltage and the frequency of each domain is also set as Table 1. Furthermore, during the system running, the frequency of each back-

end domains is adjusted dynamically and adaptively as the control algorithm described above. This strategy is called DVF+DAA, which is an abbreviation for Different Voltage and Frequency + Dynamic Adaptive Adjustment Strategy. This group of experiment is used to verify the potentiality for fine-grained dynamic adaptive frequency adjustment decreasing the power of the microprocessor in MCDE.

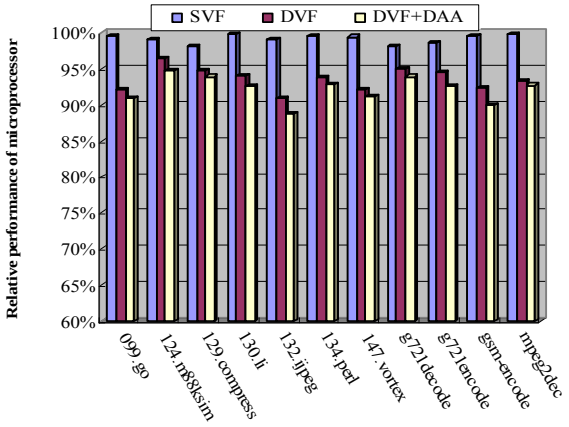
5 THE ANALYSIS OF EXPERIMENT RESULTS

Figures 5 shows the power consumption of MCDE processors relative to the basic EPIC processor under the three groups of experiment described above. The results of the experiments reveal that all three MCDE strategies can decrease the microprocessor's power consumption. But the power decrease of SVF is small, which is only 6 percent. After DVF used, the processor's power is decreased by 32 percent owing to further exploiting the advantage of MCDE. Comparing with SVF and DVF, Using DVF+DAA strategy can more effectively decrease the power consumption of the microprocessor, which decreases the power by 40 percent, as a result of using the fine-grained dynamic adaptive frequency adjustment.



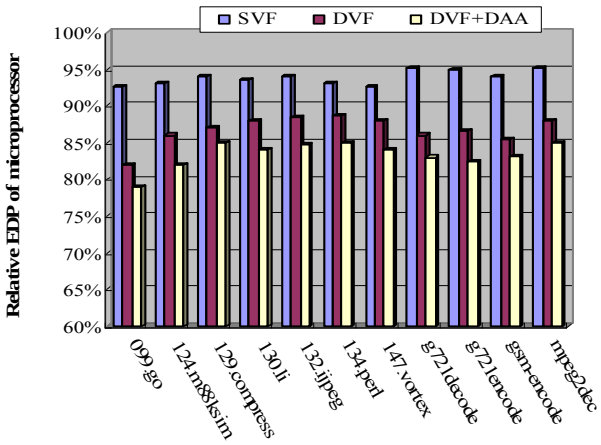
Figures 5: The Power of MCDE Relative to EPIC

Figures 6 shows that using the different MCDE strategy has an impact on the performance of the micro-processor. The results of the experiment reveal that all these three MCDE strategies can lead to performance degradation relative to EPIC, since they use the asynchronous communication mechanism. The SVF strategy results in a slight degradation, within 1 percent. Comparing with SVF, DVF and DVF+DAA result in more performance degradation owing to clock frequency being decreased. For DVF, the average performance degradation is approximately 6.5 percent. For DVF+DAA, the average performance degradation is about 7.3 percent.



Figures 6: The Performance of MCDE Relative to EPIC

When designing architecture oriented towards power consumption, it is improper that only performance or only power consumption is considered. We should comprehensively consider the performance and the power consumption, and compromise between them. Energy delay product (EDP) is a popular metric to evaluate comprehensively the performance and the power consumption. Figures 7 shows the EDP of MCDE relative to EPIC, corresponding to three MCDE strategies. The results of the experiment indicate that, for DVF+DAA, although the performance degradation is slightly significant, a significant overall EDP improvement is achieved, about 17 percent, owing to immensely exploit the potential for MCDE reducing the power consumption of the microprocessor. In addition, for DVF, the average EDP improvement is approximat-ely 14 percent. Finally, for SVF, although the performance degradation is very slight, comparing with DVF and DVF+DAA, the power consumption optimization brought by SVF is not significant, only 7 percent, as a consequence of the conservative strategy. But comparing with EPIC, all three MCDE strategies are obvious to improve the EDP. Thus, there is a significant advantage of decreasing the power consumption by using MCDE, comparing with EPIC.



Figures 7: The EDP of MCDE Relative to EPIC

6 CONCLUSIONS AND FUTURE WORK

The MCD is a novel technique that combines the advantages of synchronous systems and the advantages of asynchronous systems to resolve the problems of high-performance microprocessors, such as the significant power, and the complex clock distribution network. At present, the studies of MCD are almost based on superscalar. In this paper, MCDE, a MCD technique based on EPIC, is implemented, and evaluated comprehensively. The experimental results show that, the MCDE has a significant potential for reducing the power consumption of the high-performance EPIC microprocessor. For example, using the DVF+DAA strategy, the power consumption of the EPIC microprocessor can be reduced by 40%, and about a 17 percent EDP improvement is achieved.

In future work, we will continue to develop more effective fine-grained dynamic, adaptive voltage and frequency adjustment mechanisms to attain more significant power consumption decrease at the cost of slight performance degradation.

ACKNOWLEDGMENT

This work was supported by NSFC (No.60676016) in China.

REFERENCES

[1] Jatuchai Pangjun et al.. 2002. “Low-Power Clock Distribution Using Multiple Voltages and Reduced Swings.” *IEEE Trans.on VLSI ststems*, Vol.10, No.3(June), 309-318.
[2] D.M. Chapiro. 1984. “Globally Asynchronous Locally Synchronous Systems”. PhD thesis, Stanford Univ.
[3] A. Iyer and D. Marculescu. 2002. “Power and Performance Evaluation of Globally Asynchronous Locally Synchronous Processors”. *Proc. 29th Int’l Symp. Computer Architecture* (ISCA 02).IEEE, 158-170.
[4] G. Magklis et al.. 2003. “Profile-based Dynamic Voltage and Frequency Scaling for a Multiple Clock Domain Microprocessor”. *Proc. 30th Int’l Symp. Computer Architecture* (ISCA 03), ACM Press, 14-27.
[5] G. Semeraro et al.. 2002. “Energy-Efficient Processor Design Using Multiple Clock Domains with Dynamic Voltage and Frequency Scaling”. *Proc. 8th Int’l Symp. High-Performance Computer Architecture* (HPCA02), IEEE, 29-40.
[6] G. Semeraro et al.. 2002. “Dynamic Frequency and Voltage Control for a Multiple Clock Domain Microarchitecture”. *Proc. 35th Ann. IEEE/ACM Int’l Symp. Microarchitecture* (MICRO-35), IEEE, 356-370.
[7] Iyer A and Marculescu D. 2002. “Power efficiency of voltage scaling in multiple clock, multiple voltage cores”. In *Proceedings of the International Conference on Computer-Aided Design* (ICCAD 2002), San Jose, America, 379-386.
[8] Eswaran A and Chen S. 2003. “All-domain fine grain dynamic speed/voltage scaling for GALS processors”. URL http://www.ece.cmu.edu/~schen1/ece743/proposal_ up.pdf.
[9] Schlansker M.,Rau B..2000.“EPIC: Explicitly Parallel Instruction Computing”. *IEEE Computer*, Vol.33, No.2,37-45.
[10] Naffziger S., Colon-Bonet G., Fischer T. et al.. 2002. “The Implementation of the Itanium 2 Microprocessor”. *IEEE Journal of Solid-State Circuits*, Vol.37, No.11, 1448-1460.
[11] Wang Yongwen and Zhang Minxuan. 2004. “Microarchitecture-Level Power Modeling and Analyzing for High-Performance Microprocessors”. *Chinese Journal of Computers*, Vol.27, No.10(Oct.), 1320-1327.
[12] Brooks D., Tiwari V., Martonosi M.. 2000. “Wattch: A Framework for Architecture-level Power Analysis and Optimizations”. *In Proceedings of the 27th Annual International Symposium on Computer Architecture.* Vancouver, British, Columbia, Canada, 83-94.
[13] Reinman G. and Jouppi N.. 2000. “An Integrated Cache Timing and Power Model”. Technical report 2000/7, Western Research Laboratory, Palo Alto, California, USA.
[14] Gan Xue-Wen. 1999. “Digital CMOS VLSI Analysis and Design Basics”. Peking University Press. Beijing, Chnia.
[15] Chang P., Mahlke S., Chen W. et al.. 1991. “An Architectural Framework for Multiple-instruction-issue Processors”. *In Proceedings of the 18th Annual International Symposium on Computer Architecture*, Toronto, Ontario,Canada, 266-275.

AUTHOR BIOGRAPHIES

RONG JI received the M.S. degree in computer engineering from National University of Defense Technology, Changsha, China, in 2004, and is currently working toward the Ph.D. degree at this school. Her research interests include architecture,low-power design, and application analysis.
Her e-mail address is : rongji@nudt.edu.cn

LIANG CHEN received the M.S. degree in computer engineering from National University of Defense Technology, Changsha, China, in 2005, and is currently working toward the Ph.D. degree at this school. His research interests include architecture and low-power design.
His e-mail address is : liangchen@nudt.edu.cn

YONGWEN WANG is a associate professor in School of Computer Science at the National University of Defense Technology. His research interests include computer architecture, and low-power design. He has a Ph.D degree in computer engineering from National University of Defense Technology.
His e-mail address is : ywwang@nudt.edu.cn

XIANJUN ZENG is a professor in School of Computer Science at the National University of Defense Technology. His research interests include computer architecture, and low-power design. He has a Ph.D degree in electrical and computer engineering from National University of Defense Technology.
His e-mail address is : xjzeng@nudt.edu.cn

JUNFENG ZHANG received the M.S. degree in computer engineering from National University of Defense Technology, Changsha, China, in 2006, and is currently working toward the Ph.D. degree at this school. His research interests include low-power design, and application analysis.
His e-mail address is : jfzhang@nudt.edu.cn