Power and Performance Simulator:ESP and its Application for 100MIPS/W Class RISC Design

Toshinori SATO†, Masato NAGAMATSU‡, Haruyuki TAGO†

†Toshiba Research and Development Center, Kawasaki, Japan
‡Toshiba Semiconductor Device Engineering Lab., Kawasaki, Japan
{tsato,tago}@srd.ull.rdc.toshiba.co.jp
nagamatu@sdel.toshiba.co.jp

Abstract

A new power simulator, ESP, Early design Stage Power and performance simulator, is presented. With ESP, it becomes possible to do more precise optimization between power and performance in an early design stage. We applyed it to a low-power RISC design and proved it useful for lower power design.

1 Introduction

With the emergence of battery-operated devices including notebook PCs and PDAs, low-power high-performance microprocessors have recently been developed toward 100MIPS/W platform[1].

To aid in energy efficient design, several gate/transistor-level dynamic power simulators have been developed[2]. These tools require a gate/transistor-level net list. However, because design decisions at architectural level often have an bigger impact upon power and performance than gate/transistor-level optimization, the estimation of power consumption during architecture/pipeline level would be much more useful to the LSI designer.

For this reason, we have developed an architecture/pipeline-level power and performance simulator, ESP, to assist in making design decisions at the architectural level.

2 ESP

ESP is a cycle-based simulator with the aim of power and performance estimation. The target system of ESP is a RISC microprocessor, and is divided into several components according to hardware implementation.

As shown in Figure 1, ESP reads object code and parameters about cache configuration and low-power techniques utilized. ESP executes the object code, and at each clock cycle ESP determines active components, and sums up the current for them. When a simulation is finished, ESP reports average current and the number of clock cycles executed.

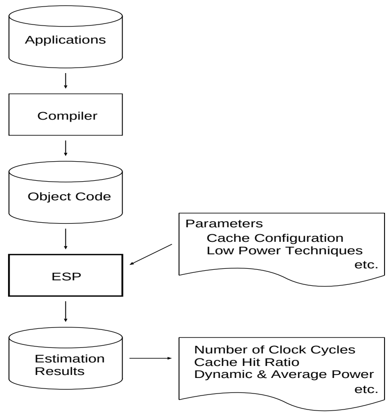

Figure 1: ESP

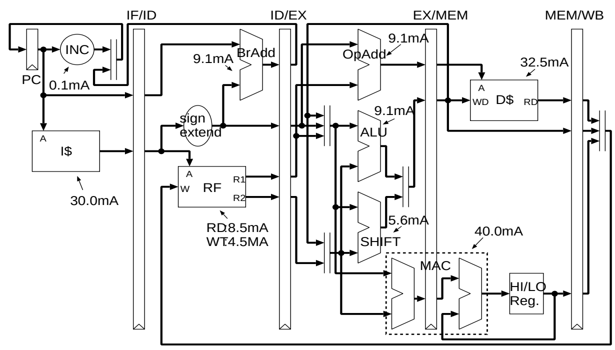

Figure 2: RISC pipeline

ESP has been experimentally applied to a RISC microprocessor as shown in Figure 2. The static current for major components are also indicated, including an instruction cache(I$), an incrementer(INC), a branch adder(BrAdd), a register file(RF), an address calculator(OpAdd), an ALU, a shifter(SHIFT), a data cache(D$), and a multiply-add unit(MAC).

Figure 3 shows the process of a simulation. For example, at clock #3, the active components are ALU(Inst #1), RFread(Inst #2), and BrAdd(Inst #2), and total current is the sum of the component currents.

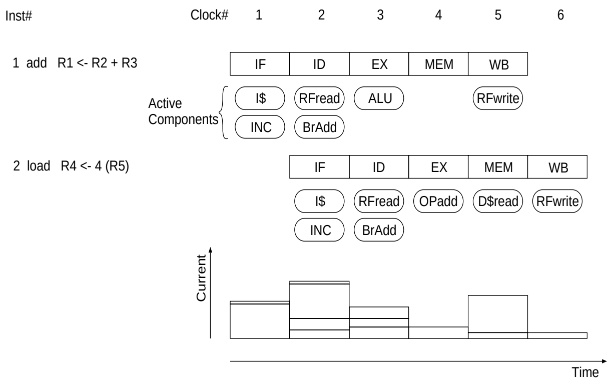

Figure 3: Simulation Example

3 Power and Performance Analysis

When Dhrystone ver.2.1 benchmark is executed on ESP, the active rate of the major components are shown in Table 1 and average current is 161.7mA at 3.3V.

Table 1: Current and Active Rate of each Component

Component		Active Rate
Name	Current(mA)	
I$	30.0	99.6%
BrAdd	9.1	99.6%
INC	0.1	99.6%
RF	RD:8.5,WT:4.5	97.4%
ALU	9.1	59.0%
D$	32.5	47.8%
OpAdd	9.1	30.1%
SHIFT	5.6	6.4%
MAC	40.0	0.2%

This result implies that I$, D$ and RF dominate the total power, especially I$ with a 99.6% active rate at 30.0mA. By decreasing the power or active rate of this component, total system power can be significantly reduced.

It is possible to decrease its active rate by using an instruction queuing technique. Instead of reading one instruction per clock, 4 instructions are read in parallel every 4 clocks and stored in a queue as shown in Figure 4. This scheme could lower the active rate of I$ by as much as a factor of 4.

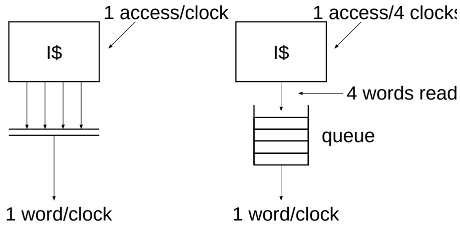

Figure 4: I-cache

The experimental results show that the active rate of I$ is reduced to 39.3%, and the total average current is reduced to 143.6mA. Figure 5 shows dynamic current on same period for the 2 simulation results.

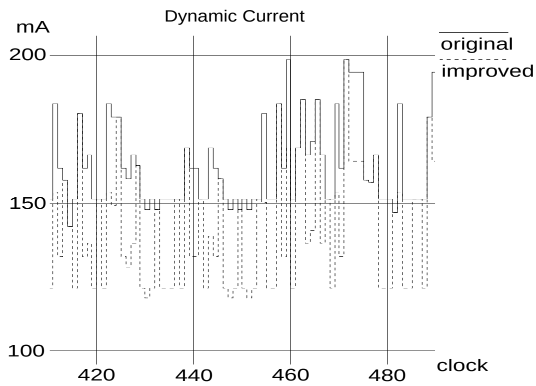

Figure 5: Dynamic Power Consumption

With this technique, the total current is reduced by 11.1%, at the expense of extra hardware and control logic for the queue. Whether to adopt this I$ design depends on the design target. Thus, ESP is useful for evaluating design trade-offs between power consumption and chip area at the architectural level.

4 Conclusion

We have developed an architectural level power and performance simulator, ESP, which provides LSI designers with effective design tool for low-power designs. We have applyed ESP to low-power RISC design, and found it useful for architectural level optimization.

References

[1] Yeung,N.K.,et al.,"The Design of a 55SPECint92 RISC Processor under 2W,"ISSCC Digest of Technical Papers, pp.206-207,Feb.,1994.

[2] EPIC Design Technology Inc., "PowerMill User Manual Release 3.0," Oct., 1993.