Instruction Level Power Profile for the PowerPC Microprocessor

Oscar Acevedo Patiño
Advisor: Dr. Manuel Jimenez

Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez Campus
Mayagüez, Puerto Rico 00681-5000
oscara@ece.uprm.edu

Abstract

Power consumption is an important aspect in the design of modern embedded systems. The embedded system hardware and software should be evaluated in term of power metrics to verify if a design meets its specified power constraints. An instruction level power profile will help to determine the program power consumption and what consideration should be done for power reduction. This paper presents the methodology adopted to generate the instruction power profile for the PowerPC 603e microprocessor, which will be used to generate software techniques to reduce power consumption.

1. Introduction

Power reduction is a requirement in the design of modern applications principally to improve portable equipment autonomy and circuits efficiency.

Microprocessor power reduction can be achieved by low-power circuitry design (low level), and using optimized software compilation techniques (high level). Each instruction program activates specific parts of the microprocessor, which generates different power consumption based on the instruction type and the amount of instructions executed. Therefore, good instruction management can achieve power reduction. This is the premise for the software power reduction techniques.

To generate software techniques for power reduction, it is necessary a tool that estimates the microprocessor power consumption based on the program-executed instructions. This tool requires a power profile, where a power consumption value is assigned to each microprocessor instruction.

Two main methods have been reported to generate the instruction power profile. The first method, proposed by Tiwari et al. [Tiwari94], directly measures the current drawn by the processor as it repeatedly executes each instruction. The second method is based on a simulation of the microprocessor under test. The effect of the instruction set execution (power consumption) is then measured from the simulation model [Tiwari94]. A drawback of this method is that detailed information of the CPU circuitry must be available to do the simulation.

The Instruction power profile is a methodology used to generate average power consumption values for each instruction of a given microprocessor. It is a one-time process and requires a special test bench to generate the results. This paper presents the methodology adopted to generate the instruction power profile for the PowerPC 603e microprocessor, which is the first step to generate software power reduction techniques for this microprocessor.

2. Experimental Methodology

2.1 Background

The basic idea of generating the instruction power profile is to measure the current drawn by the microprocessor when it executes a specific instruction. Current measurement is enough to determine the power consumption since it is directly related to power by the equation:

$$P = V_{cc} * I_{micro} \quad (1)$$

where *Vcc* is the supply voltage (constant), and *I* is the current drawn by the microprocessor (measured). The instruction execution time is too short for a direct current measurement; therefore, it is necessary several instances of the instruction under test enclosed in a

loop to assure an accurate current measurement [Tiwari94].

2.2 Test Bench

The test bench is composed by the hardware and the equipment required to generate the instruction power profile. It includes a test board with the microprocessor under test, power supplies, an ammeter and a computer for the ammeter readings. Figure 1 shows a basic schematic for the test bench.

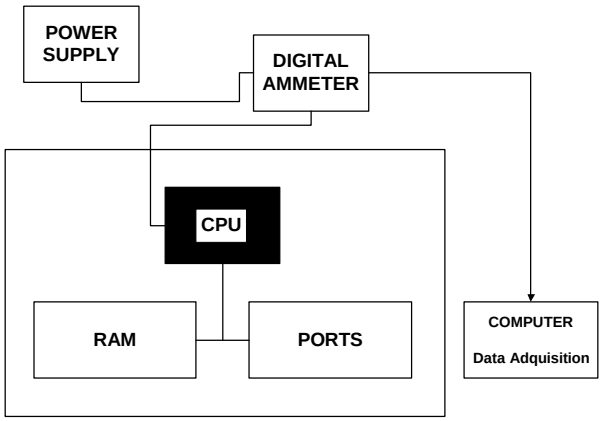

Figure 1. Test Bench Schematic

To realize the measurement, a standard ammeter was used which, includes an RS232 protocol for remote control and reading, which is used for data acquisition with a computer and a control program. The hardware includes the microprocessor under test and its support devices (memory, controllers, ports, etc.).

2.3 Test Program

The test program is code downloaded to the microprocessor under test that assures the execution of a unique instruction for current measurement. The algorithm used for the test program includes a random argument selection for each instruction, and a loop for the instruction under test [Russel98]. Several instances of the same instruction enclosed in a loop generate a stable signal to be read by the current meter. Furthermore, it assures each pipeline stage executes the same instruction, obtaining a current measurement only by the processing of the instruction under test [Tiwari94]. Unfortunately, some instructions require a variation in the test program in order to generate variation in their arguments. Figure 2 presents the general algorithm for the test program [Sridhar97] [Chakrabarti99].

The test program should be the only one executed by the microprocessor. This is because other instructions may introduce deviations for the power consumption, introducing errors in the measurements.

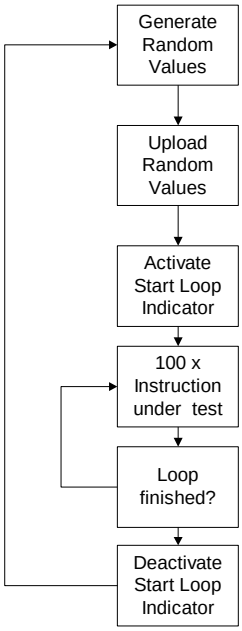

Figure 2. General Test program Algorithm

2.4 Profiling Algorithm

An application program executes different instruction combinations to generate a specific result. This instruction mixture generates conditions that change the microprocessor current consumption. These conditions, as the cache miss, stall, switching activity, etc. should be measured in order to generate an accurate power prediction [Tiwari95].

To determine the amount of current generated by these states, three different tests are required: The first was described in the section 2.3, generates the current base cost; the current consumed by the execution of a specific instruction. The second test combines two different instructions in order to measure the current variation due to the execution of different instructions. This test is known as overhead state, and it is based on which states was the microprocessor before the instruction execution. The third test generates cache misses, stalls and other states to measure the current variation caused by these microprocessor states [Tiwari95].

The final result is an average current base for each instruction, a current offset for each pair of instructions

and a current offset for the different microprocessor states described before. The sum of each component generates the average current value for each microprocessor instruction. Only average current values are used due to real data and address can be known only at run-time.

2.5 Software Calculator

The software calculator group the results obtained by the different test program. This information is used to analyze a given program in order to determine its power consumption.

3. Results

This section presents current measurement tests done to some integer arithmetic instructions. The results presented here correspond to the current base cost, which is the first test to determine the current consumption without considering other effects.

Table 1 presents the current variation versus the register value variation. Based on the measurements, a linear relation is observed between the amount of ones in the registers rA & rB and the microprocessor current consumption. Table 2 presents the equation generated (based on information in Table 1), and the maximum error obtained.

Table 1. Current consumption (mA) per argument

	Amount of "ones"		
Instruction	**0**	**32**	**64**
AND rD,rA,rB	746.5	715.6	684.2
XOR rD,rA,rB	750.7	719.0	686.5
OR rD,rA,rB	715.4	719.1	688.6
ADD rD,rA,rB	815.0	782.0	749.0
SUBF rD,rA,rB	747.4	718.7	683.2

The results presented in Table 2 show a similar equation form for the instructions presented. The error generated is lower than 1%, which indicated that presume a linear variation is a good choice for the current variation.

Table 2. Equation generated for current consumption (mA) versus amount of "ones" (B).

Instruction	Equation	Error (%)
AND rD,rA,rB	I=-0.97344*B+746.5	0.08
XOR rD,rA,rB	I=-1.003*B+750.7	0.14
OR rD,rA,rB	I=-0.98125*B + 751.4	0.12
ADD rD,rA,rB	I=-1.0312*B+815	0.2
SUBF rD,rA,rB	I=-1.003*B+747.4	0.4

Table 3 presents the variation in current consumption when the registers used in the instruction are changed. The results observed indicate that when the registers used are different, the current consumption is approximately constant, but when the same register is used as argument and destination, the current consumption tends to grow. However, the current variation is not considerable except for the ADD instruction, where the variation is larger than 5%.

Table 3. Average current consumption (mA) per register variation

Instruction	rD?rA?rB	rD=rA?rB	rD=rA=rB
AND rD,rA,rB	697.6	707.5	711.0
XOR rD,rA,rB	702.6	790.7	749.6
OR rD,rA,rB	701.5	711.7	715.0
ADD rD,rA,rB	758.1	857.6	800
SUBF rD,rA,rB	696.1	728.5	746.4

Table 4 present the error obtained when the average value in table 3 is taken as reference value for different measurements. The variation is short, therefore, to assume these values as average is a good choice.

Table 4. Maximum error presented by register variation

Instruction	rD?rA?rB	rD=rA?rB	rD=rA=rB
AND rD,rA,rB	<2%	<2%	<2%
XOR rD,rA,rB	<2%	<2%	<2%
OR rD,rA,rB	<2%	<2%	<2%
ADD rD,rA,rB	<2%	<2%	<2%
SUBF rD,rA,rB	<2%	<2%	<2%

4. Future Work

The first step is to generate the instruction power profile for the complete PowerPC instruction set. Once this step is finished, a power calculator will be created in order to estimate the power consumption of a given program without physical measurement.

The information obtained will be used to generate software power reduction techniques, which will be applied to a compiler, in order to generate optimized programs for the PowerPC microprocessor.

5. Summary

The power consumption is an important constrain in the microprocessor system design. The Instruction Power Profile is a tool that enables to generate power measurements for the application programs and will give "clues" to optimize the programs done for the PowerPC microprocessor, in order to reduce the its power consumption.

References

[Chakrabarti99] Chakrabarti, C.; Gaiton de, D. “Instruction level power model of microcontrollers”. . Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, ISCAS '99. Volume: 1 , Page(s): 76 -79 vol.1

[Russel98] Russell, J.T.; Jacome, M.F. “Software power estimation and optimization for high performance, 32-bit embedded processors”. Proceedings, 1998 International Conference on Computer Design: VLSI in Computers and Processors. ICCD '98. Page(s): 328 –333

[Sridhar97] Sridhar, R.; Schindler, K. “Instruction level power model and its application to general purpose processors”. Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers, 1997., Volume: 1. Page(s): 753 –756.

[Tiwari94] Tiwari, V.; Malik, S.; Wolfe, A , " Power analysis of embedded software: a first step towards software power minimization ," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , Volume: 2 Issue: 4 , Dec. 1994 Page(s): 437 –445.

[Tiwari95], V.; Tien -Chien Lee, M. “Power analysis of a 32-bit embedded microcontroller”. Proceedings of the IFIP ASP-DAC '95/CHDL '95/VLSI. Asian and South Pacific Design Automation Conference, 1995. Page(s): 141 –148.