RC22621 (W0211-014) November 1, 2002
Computer Science

IBM Research Report

Experimental validation of relative accuracy limits and sensitivities in power-performance simulators

David Brooks[+] , Pradip Bose[*] , Margaret Martonosi[**]

[+] Dept. of Computer Science
Harvard University

[*] IBM T.J. Watson Research Center

[**] Dept. of Electrical Engineering
Princeton University

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Experimental Validation of Relative Accuracy Limits and Sensitivities in Power-Performance Simulators

David Brooks†, Pradip Bose‡, Margaret Martonosi††

† Dept. of Computer Science, Harvard University ‡ IBM T.J. Watson Research Center †† Dept. of Electrical Engineering, Princeton University

***Abstract*— Microprocessor research and development increasingly relies on detailed simulations to make design choices. As such, simulator accuracy is of increasing importance to the field. This paper breaks down accuracy, a broad term, into two main types: *relative* and *absolute*. We then discuss typical abstraction errors in power-performance simulators and show when they do (or do not) affect the design choices a user of those simulators might make. By performing this validation study using the Wattch simulator, we feel that the work helps to address validation issues both broadly and in the specific case of a widely-used simulator.**

I. Introduction

Because the computer systems we build today are so complex, they are difficult to reason about; as a result, detailed simulations have become essential both for designing real systems and for evaluating research ideas in our field. As systems grow more complex, simulator validation becomes increasingly crucial. Validation not only checks for bugs, but also helps to quantify a model's accuracy and applicability in different parts of the design space; users need to know how accuracy is affected by model abstractions used to provide superior simulation speed, to improve design space flexibility, or to speed model construction.

In this paper we discuss model accuracy within the context of Wattch, a power-performance simulator [2]. We show how different types of possible modeling abstractions and errors can affect the design choices one uses a simulator to evaluate. Our results distinguish cases when *absolute accuracy* is required, versus cases when the easier-to-achieve standard of *relative accuracy* is sufficient. This analysis can help power-performance simulator writers and users focus on areas that improve the likelihood of good design choices.

II. Types of Modeling Error

At the highest level, a model or simulator has metrics that the simulation is intended to produce. These may be aggregate numbers like execution cycles or total energy requirements for a program run; or the numbers may be more fine-grained, such as distributions of the number of instructions ready to issue each cycle, the maximum and minimum instantaneous power, etc. We use the term absolute accuracy to refer to a simulator's ability—for a particular metric—to closely track the value measured by the 'real system' or a better model for that same metric. Relative accuracy, on the other hand, reflects that a simulator produces a range of results that properly reflect relative changes with design parameters, even if the absolute value of the result may not be perfect.

Achieving relative accuracy is much easier than achieving absolute accuracy, especially during the early-stages of the design process. This is because relative accuracy can be maintained despite errors in low-level technology parameters, incorrect assumptions about circuit-design styles, clocking network design methodologies, etc. Absolute accuracy is degraded due to all of these conditions.

A simulator with good relative accuracy provides quite a bit of useful information to an architect. For example, design tradeoff studies with the goal of choosing architectural parameters to achieve an optimal power-performance efficiency can easily be performed. This is not to say that absolute accuracy is not important at all. For example, determining the true maximum wattage of a particular chip requires good absolute modeling accuracy. In contrast, relative accuracy can help designers deduce the design point that will produce the maximum wattage, but may not predict the actual wattage with sufficiently small error. In some cases, however, good relative accuracy combined with bounding techniques can help CPU designers with problems requiring some degree of absolute accuracy.

Previous work in energy model validation has mainly looked at validating models against more detailed information derived from lower-level tools. Comparing low-level capacitance values is one precise means of validating energy models within a power simulator. This method of validation has shown the models to be accurate within 10-15%, which is similar to what has been reported by the CACTI authors for analytical delay models [5] and later for analytical power models [4]. Amrutur and Horowitz have also studied analytical power and delay models for SRAMs [1].

III. Robustness of Relative Accuracy

While simulator error is never a good thing, it is important to understand how different types of error influence (or not) the design process. Understanding the effects of different types of error gives guidance for how to interpret simulator results. These results give some insight into the robustness of the relative accuracy of the power models and demonstrate the extent to which a design tradeoff study can withstand error in the low level power models.

When performing a design tradeoff study, a methodology must first be established for deciding when to choose a particular design point over another design point. When viewing design tradeoff curves visually, we often would like to choose the "knee" of the curve so as to pick the point that is close to optimal without pushing too deeply into a region of diminishing marginal returns. While this process is intuitive to designers, this paper quantifies it in order to be able to characterize whether a chosen design point is or is not acceptable. In particular, we propose the *acceptable range window* as a method to quantify the selection of design points from raw power/performance data.

The experiments in this paper quantify the amount of acceptable error within a power/performance simulator that can be tolerated before different design points are chosen. The acceptable range window forms a group of points which meet the criteria for selection. Generally, we choose the lowest cost point within the acceptable range window for implementation.

Two different definitions of the acceptable range window are considered:

- 1) +/-R1% of target metric at optimal choice (*range1*). This chooses candidate design points that are within a percentage of the target metric.
- 2) +/-R2% of (worst_choice - optimal_choice) for this design study (*range2*). This is geared to be more selective in cases when the design tradeoff curve is so flat that many choices might satisfy range1. This definition adjusts for the fact that when optimal and non-optimal options are close together, the true range of acceptable designs may be narrower.

In this paper we use Wattch (with the aggressive clock gating mode) as the baseline simulator to perform three representative studies and we consider acceptable range windows of Defintion 1) with R1 equal to 5%. While all of the types of error that we consider disturb the absolute accuracy of the simulator, this study quantifies the effect on the relative accuracy of the simulator by investigating two design tradeoff study scenarios. These design tradeoff studies investigate energy-delay product for the number of RUU-entries and the size of the L1 Instruction Cache. We have also done additional experiments both with Wattch for the L1 Data Cache as well another power simulator, PowerTimer, for both definitions of the acceptable range window [3]. In this paper, we show plots for the *vortex* application which tended to have the most interesting results (the most deviations) due to its sensitivity to the studied parameters. In the end of each subsection, we summarize the results for 5 of other the SPECint95 applications – *compress*, *gcc*, *go*, *ijpeg*, and *m88ksim*.

For each design tradeoff experiment, we check for overlap between the acceptable range windows of the baseline simulator and the modified simulator. Agreement implies that relative accuracy was maintained, and the correct design choice would be selected despite simulator error. We define a *critical deviation* if it would cause the designer to choose a point other than the least cost design point in the baseline acceptable range window – we will highlight these cases when they occur.

Example 1: Error in an Independent Unit: Consider first a simple scenario in which a designer is using a simulator to make a sizing choice about one of the hardware units, say the register file or the L1 caches. This experiment considers what happens if the designer uses a simulator which has error in the power estimate for a unit that is independent of the units for which design decisions are being made. For example, error in the ALU power model or the global clock power, is mostly independent of the power model for the RUU or the L1 caches. While the absolute accuracy of the model suffers quite a bit under these conditions, the relative accuracy of the model for a particular design study is typically less severely affected.

Figures 1 and 2 show two graphs each for the *vortex* application while varying the number of RUU entries and the I-Cache size. In each of the graphs there are five curves showing the power and energy-delay product trends while varying the microarchitectural parameters. These trends are impacted both by the energy model for the structures as well as the clock gating potential of the microarchitecture. For example, machines with larger I-Caches consume more power in the I-Cache hardware in addition to having a busier downstream pipeline. The five lines labeled -.2x through .2x refer to the amount of error (additional power dissipation) inserted into the model. The amount of power added/subtracted is equal to the ratio given multiplied by the total unconstrained chip power of the baseline case with an 80-entry RUU, and 64KB D- and I-Caches.

The first graph in each figure shows the power dissipation while varying both conditions. Since the additional power dissipation added in this experiment is independent of the RUU or cache power models, it does not affect the relative accuracy of this curve and only shifts the curves up and down by the corresponding amounts.

The second graph in each figure shows the energy-delay product while varying the microarchitectural parameters and the amount of error. The energy-delay product factors in the IPC (performance) for the various microarchitectural choices. Because of this, the energy-delay product curves are skewed by the IPCs of the various design points. Each of the energy-delay product figures also has several highlighted (circled) data points. These points represent designs that fit within the acceptable range window (with the range1 criteria) for each curve. If the acceptable range window for the base case (without artificial error) matches the curves where artificial error exists, then we can say that relative accuracy was preserved. For example, in Figure 2 the same instruction cache would be chosen (64KB) even with the .2x and -.2x error conditions. On the other hand, an RUU of 16 entries would fit into the acceptable range window with -.1x and -.2x error conditions, whereas only a 32-entry RUU is in the acceptable range window with the base simulator.

For the other 5 SPECint95 applications that we considered, only *ijpeg* experienced a critical deviation in the acceptable range window. This occured for the RUU design study with -.2x error. When considering SPECint95 as an aggregate, there was no deviation from the acceptable range window with any of the error conditions that we considered.

Example 2: Error in Bitline Capacitance: A second major class of experimental inaccuracy in power models is error that occurs in a model that is used within many microarchitectural structures. For example, in Wattch, array structures such as the L1 instruction and data caches, the L2 cache, and the branch predictor tables are all modeled as instantiations of a single 'cache' power model. Error in the cache power model affects the power estimates for all of these units. In this example, we consider a specific scenario in which we have misestimated bitline capacitance. Since bitline capacitance estimates are used within the array structure models for caches and register files, an error in bitline capacitance affects both of the microarchitectural parameters under study, as well as several independent structures.

Figures 3 and 4 show the power and energy-delay product for the *vortex* application while varying the number of RUU entries and I-Cache size. The five curves shown are similar to the ones in the previous section, but each of these curves shows a different ratio for the bitline capacitance scaling that was used. Again, significant deviations are difficult to see from these curves even with 0.6x and 1.4x scaling of the bitline capacitance estimates. In fact, the acceptable range windows were identical for both the RUU and I-cache design studies under all error conditions and there were no critical deviations for any of the SPECint95 applications.

Example 3: Error in Dependent Unit Scaling Factors: As a third example, we consider the effect of an error solely within the unit focused on by design study. Such an error might arise

if, for example, we modeled a different circuit-design style than was actually used for that particular structure, or if an incorrect sub-banking scheme was assumed, etc. These experiments explore this error by scaling the power estimate for the individual structures (RUU and L1 Caches) by 1x through 2x.

Figures 5 and 6 show the power and energy-delay product for the *vortex* application while varying the number of RUU entries and I-Cache size. Each of the five curves shows the power or energy-delay product as that particular unit's power estimate scales by 1x through 2x. This type of error clearly affects the design tradeoff study. As the amount of scaling increases, instead of just shifting the results, the curves begin to separate as more scaling is applied. The acceptable range windows begin to differ more for both the RUU and I-Cache design studies. For the RUU, the acceptable range window is between 16 and 64 entries, whereas with -.2x error it is between 32 and 80 entries and with .2x error it is between 16 and 48 entries.

The acceptable range windows highlight how and when this error disturbs the design tradeoff study. Even with 25% error, ie, a 1.25x scaling factor, there is very little change in the acceptable range windows for the three design tradeoff studies with these applications. However, *vortex* with the 1.5x and larger scaling factors resulted in critical deviation by choosing a design point with a smaller number of RUU-entries. There were no other critical deviations when considering the remainder of the SPECint95 applications.

Overall, the errors in this third example—those that specifically involve the unit under study— are more likely to change the design choices made. This is because the additional scaling on the microarchitectural structure, in the absence of the scaling in other independent units, causes the structure in the tradeoff experiment to become a larger share of the overall chip power dissipation 'pie'.

IV. Conclusion

This paper has considered the relative versus absolute accuracy of the architecture simulators we use. In particular, working with the Wattch power simulator, we have investigated some likely primary sources of error and demonstrated how design tradeoff studies can tolerate some error because *relative* inaccuracy need not affect the design point chosen.

When performing a design tradeoff study, it is most important to provide accurate power models for the unit under consideration in the study. Error in independent units tend not affect the study, and errors that can affect multiple units could also have small disturbances because relative accuracy is maintained. However, errors that affect only the unit under study can lead to errors in the relative accuracy of the power model and incorrect design choices in some cases.

References

[1] B. Amrutur and M. Horowitz. Speed and power scaling of SRAM's. *IEEE Journal of Solid-State Circuits*, 35(2):175–185, 2000.

[2] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, June 2000.

[3] D. M. Brooks. *Design and Modeling of Power-Efficient Computer Architectures*. PhD thesis, Princeton Univ., Nov. 2001.

[4] G. Reinman and N. Jouppi. CACTI 2.0. In *WRL Research Report*, 1999.

[5] S. Wilton and N. Jouppi. An Enhanced Access and Cycle Time Model for On-chip Caches. In *WRL Research Report 93/5, DEC Western Research Laboratory*, 1994.

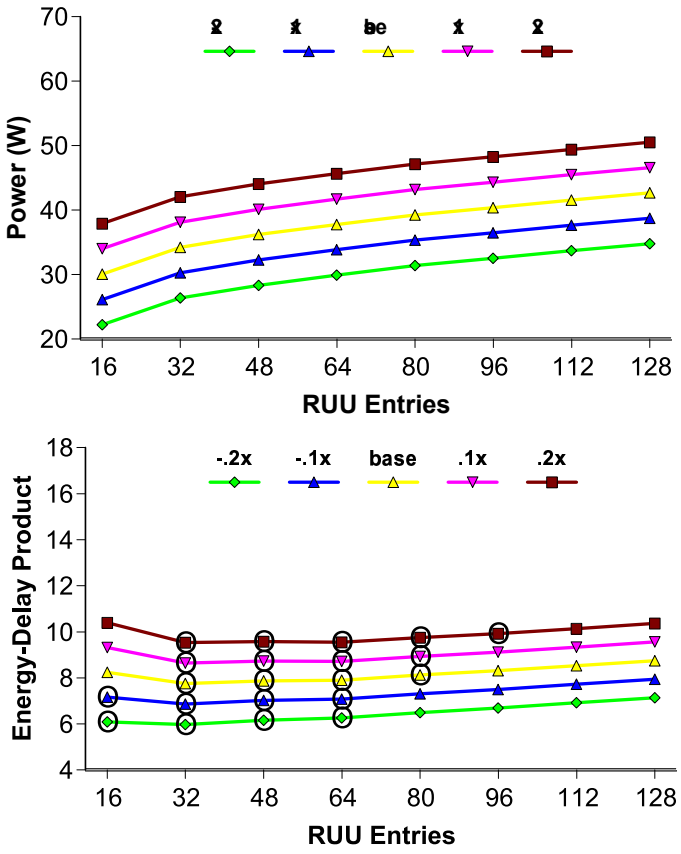

Fig. 1. Power and EDP for *vortex* varying indep. unit and RUU entries.

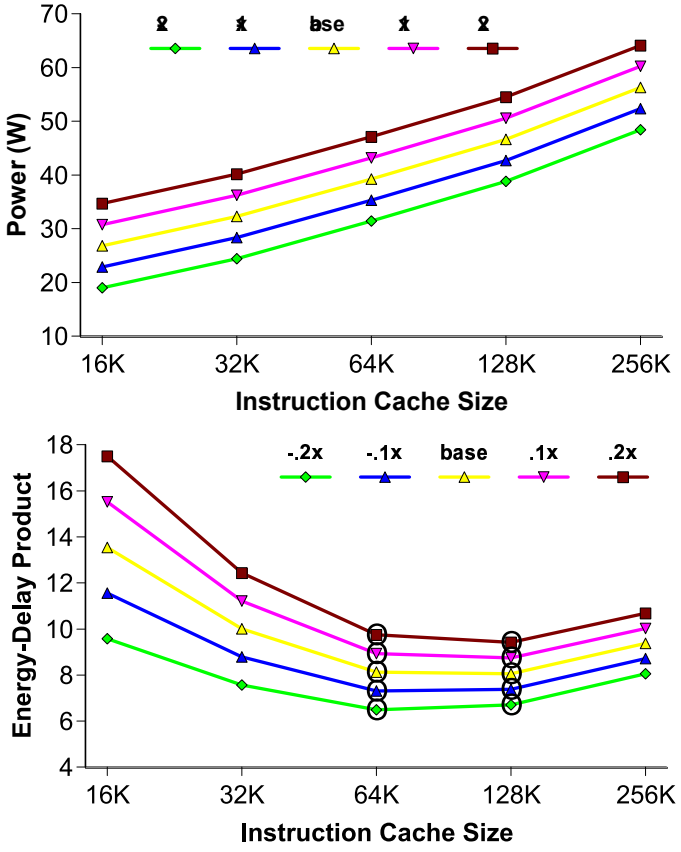

Fig. 2. Power and EDP for *vortex* varying indep. unit and I-Cache size.

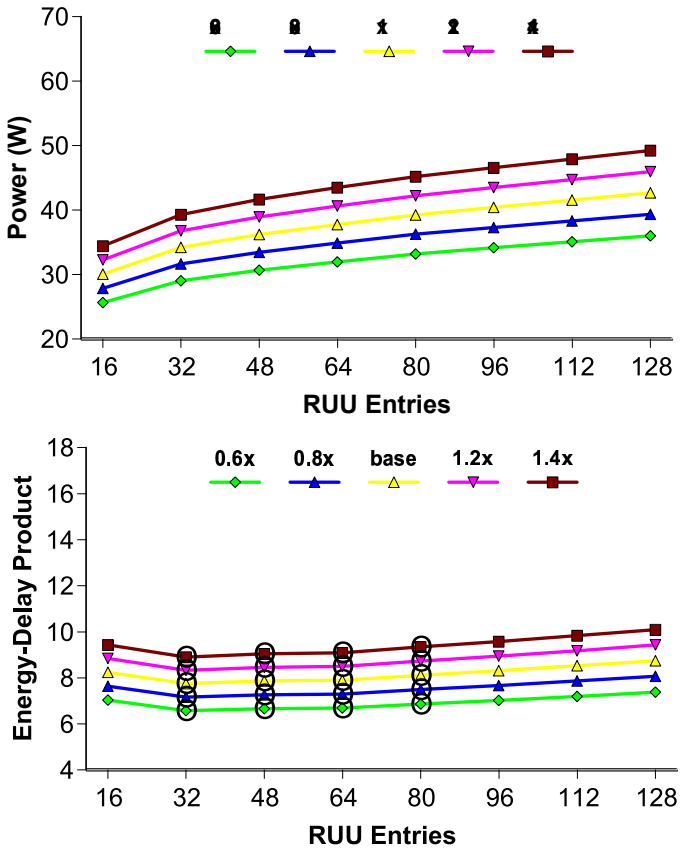

Fig. 3. Power and EDP for *vortex* varying bitline error and RUU entries.

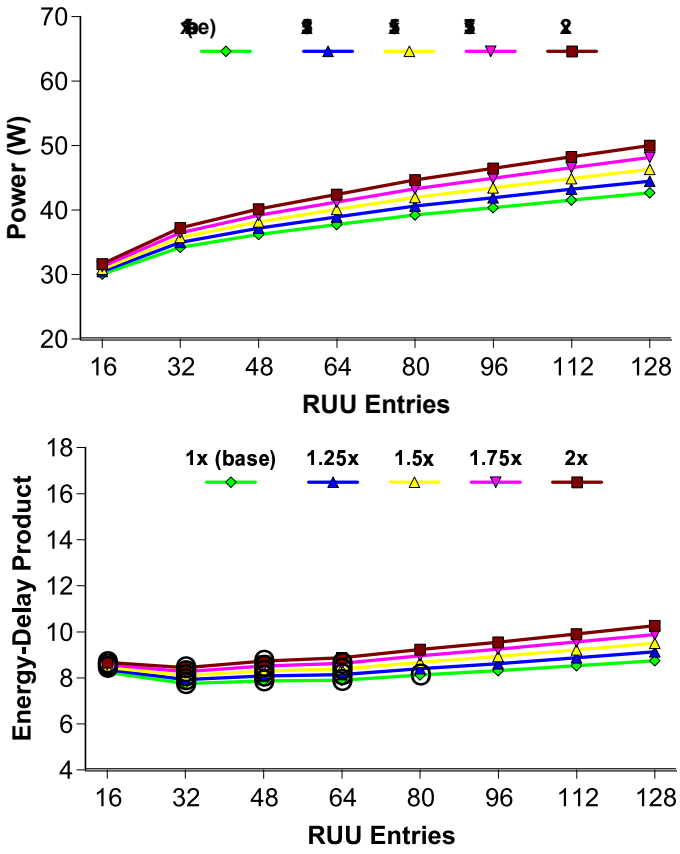

Fig. 5. Power and EDP for *vortex* varying RUU-scale-factor and RUU entries.

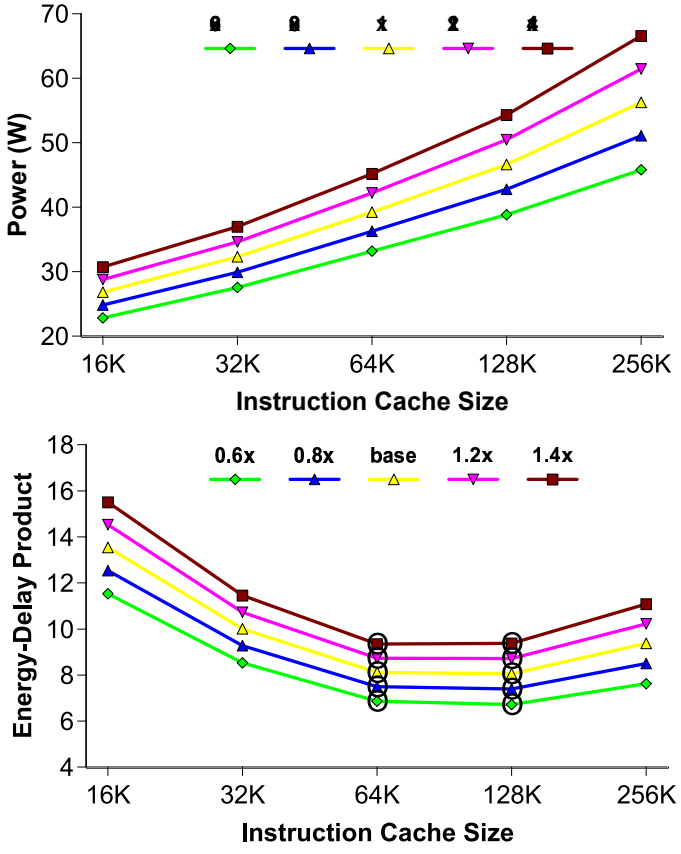

Fig. 4. Power and EDP for *vortex* varying bitline error and I-Cache size.

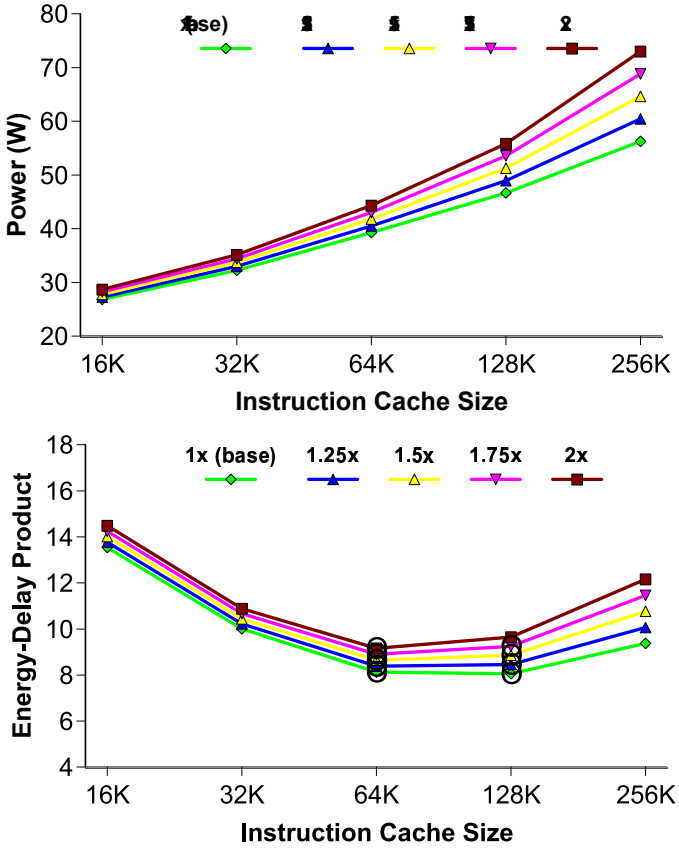

Fig. 6. Power and EDP for *vortex* varying ICache-sf and I-cache size.