Unified Architecture Level Energy-Efficiency Metric

Victor Zyuban
IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY
zyuban@us.ibm.com

ABSTRACT

The development of power-efficient microprocessors presents the need to consider power consumption at early stages of design, particularly at the ISA and microarchitecture definition stages, where the potential for power savings is more significant than at lower-level stages, and the opportunity for making power-performance tradeoffs is the largest. Design modifications to the ISA and microarchitecture, however, affect most (if not all) parameters of the design, including architectural speed, code density, clocking rate and power. A reliable metric is required to make knowledgeable power-performance tradeoffs in this multi-dimensional space. This paper derives a unified energy-efficiency metric for evaluating ISA and microarchitecture features, which subsumes other commonly used power-performance metrics as special cases of a more general equation. This new metric is derived based on an analysis of a multi-dimensional power optimization problem, and the resulting formula involves only relative changes in the characteristics of a processor, enabling its application at the early stages of the design.

Categories and Subject Descriptors

C.1.0 [**Processor Architectures**]: General; C.5.3 [**Microcomputers**]: Microprocessors; B.7.1 [**Types and Design Styles**]: Microprocessors and microcomputers,VLSI; C.0 [**General**]: Modeling of computer architecture; C.1.3 [**Other Architecture Styles**]: Pipeline processors

General Terms

Design, Performance

Keywords

Energy, power, performance, energy-efficiency, metric, architecture, microarchitecture

1. INTRODUCTION

The opportunity for power-performance tradeoffs is the largest at the early stages of microprocessor development, particularly at the instruction set and microarchitecture definition stages. At this level, even minor modifications to the design may result in significant changes to the power-performance characteristics of the processor. To draw a conclusion about the effectiveness of some existing or proposed architectural feature, one needs to evaluate its effect on the architectural speed of the processor (IPC), its power, maximum clocking rate and cost. Certain architectural features that improve the architectural speed, may be very costly in terms of power dissipation, whereas others may impact the clocking rate. A proper power-performance metric is needed to combine all these effects. In order to be useful at early design phases, such a power-performance metric has to deal with *relative* changes in the architectural performance of the processors, such as IPC and dynamic instruction count, and physical characteristics, such as the clocking rate and power dissipation. If an architectural feature under evaluation improves the power-performance metric, it is considered *energy-efficient* according to this metric; that is, it results in a better design point in the power-performance optimization space.

A number of power-performance metrics have been proposed [5, 7, 6, 2, 11, 12, 9, 3, 8], and some of them have been used to compare different products on the market. The "MIPS per Watt" metric, which can be reduced to the reverse of "energy-per-operation" [4], has been used for comparing low-end products. It has also been used as a power performance metric in the "fixed throughput" mode [4]. This paper shows that, depending on certain factors, metric "MIPS per Watt" may or may not lead to a power-optimized design for the "fixed throughput" mode. Furthermore, we show in section *2.1.1* that "MIPS per Watt" is a special case of a more general formula, derived in this work, that covers both the "fixed throughput" and "fixed power" modes.

Sometimes the "MIPS per Watt" metric is also used for analyzing high performance processors, when such a processor cannot be set to operate at its full speed because its power exceeds the power-dissipating capabilities of the package. In this case, however, the power-performance metric can be more accurately expressed as "MIPS at maximum power" which is substantially different from "MIPS per Watt", as will be shown in Section *2.1.2* of this paper.

The energy-delay product, whose inverse can be reduced to the "MIPS square per Watt", is a more reasonable metric [7] for comparing a midrange class of microprocessors. Formulas placing more emphasis on performance by raising the exponent of MIPS have also been used for comparing high-end server microprocessors; metric $\frac{\text{MIPS}^3}{\text{Watt}}$ is an example of this [2].

All the metrics mentioned above are difficult to use for evaluating the energy efficiency of architectural features at early stages of design, for two reasons. First, absolute power and performance data are typically unavailable. Second, it is hard to reach an agree-

GLSVLSI'02, April 18-19, 2002, New York, New York, USA

ment between architects and circuit designers on the appropriate value of γ in the power-performance formula $\frac{\text{MIPS}^\gamma}{\text{Watt}}$ [13].

In this paper, we derive a new metric that combines relative changes in the architectural speed, dynamic instruction count, average energy dissipated per executed instruction, and maximum clocking rate of the processor, resulting from design modifications at the architectural and microarchitectural levels. This metric will allow designer to evaluate the energy-efficiency of architectural features before making them part of the design, and to compare architectural alternatives in the power-performance design space.

The organization of the paper is as follows: Sections 2 derives the energy-efficiency metric for three types of processor implementations: ideal clock gating, free running clock, and partial clock gating. Section 3 considers the effect of technology characteristics and circuit style on the derived metric. Section 4 gives examples of applying the metric to evaluate the energy efficiency of some architectural features. Section 5 discusses the limitations of the derived metric and summarizes the paper.

2. POWER-PERFORMANCE OPTIMIZATION

Consider the problem of optimizing the power-performance characteristics of a processor in the space of two variables: architectural complexity and power supply voltage. To allow a mathematical analysis of the problem, we introduce a discrete variable ξ that represents a measure of the architectural complexity of a processor. The domain of this variable can be defined by ordering all possible architectural alternatives, and assigning a numeric value to each of them. Then, any architectural modification to the processor results in an increment or decrement in the value of ξ. Examples of variations in architectural complexity include the addition of instructions to the ISA, modifying the definitions of existing instructions, or, at the microarchitecture level, changing the pipeline latency, adding or removing hardware functionality such as bypasses, functional unit, access read or write ports to various structures, changing the width of the datapath, and so on. We will treat the architectural complexity as an independent variable in the optimization process.

Power supply voltage v will be treated as the second independent variable in the optimization process, based on the assumption that, to achieve the desired power and performance characteristics, the power supply voltage can be set to any value from the range for which the technology is qualified. Then, the performance and power characteristics of a processor can be viewed as functions of the independent variables ξ and v, where v is a continuous and ξ is a discrete variable:

$$\begin{aligned} &\text{dynamic instruction count} && N = N(\xi) \\ &\text{architectural speed (IPC)} && I = I(\xi) \\ &\text{maximum clocking rate} && f = f(\xi, v) \\ &\text{energy per istruction} && E = E(\xi, v) \end{aligned} \quad (1)$$

In these and all following formulas, N is the total number of dynamic instructions executed on a given benchmark suite; I is the average number of instructions completed per clock cycle by the processor, calculated on the same benchmark suite; E is the average energy per instruction, calculated as $E = \sum_i w_i E_i$, where E_i is the average energy dissipated on the execution of instruction i from the instruction set, and w_i is the normalized dynamic frequency of the corresponding instructions in the benchmark suite. To a first approximation, N and I depend only on the architectural complexity ξ, and are independent of the supply voltage. The clocking rate, f, and the average energy per instruction, E, depend both on the architectural complexity ξ and the supply voltage v. Then, the processor performance P on the given benchmark suite can be expressed as follows:

$$P(\xi, v) = \frac{f(\xi, v) I(\xi)}{N(\xi)}. \quad (2)$$

The expression for power dissipation $W(\xi, v)$ depends upon the implementation details of the processor. We will consider two extreme cases: ideal clock gating and free-running clock implementations, and a more realistic case of partial clock gating.

2.1 Ideal Clock Gating

Under an ideal clock gating model, the only resources that dissipate power are those accessed by executed instructions, and all unused hardware is gated-off, using the finest-grain clock gating mechanism or some sort of transition barrier mechanism[1], or a combination of both. In this case, the average power is directly proportional to the average number of instructions executed per cycle and the average energy dissipated per completed instruction:

$$W(\xi, v) = f(\xi, v) I(\xi) E(\xi, v), \quad (3)$$

wherein E is the average energy per executed instruction, as defined above. Notice that if expression 3 is applied to a speculative issue processor, then the energy dissipated by instructions from mispredicted paths that are fetched, and possibly executed but not committed, has to be included in E.

2.1.1 Constant-Performance Optimization

In this subsection we consider the problem of minimizing the average power dissipation, given a performance requirement, $P = \text{const}$. The designer is allowed to modify the architecture (both ISA and microarchitecture) and adjust the clocking rate of the processor, by changing the power supply voltage within certain limits, to satisfy the performance requirement at minimum power dissipation. This sort of optimization problem is typical for the design of low-power microprocessors, application specific, real time processors and DSPs. In mathematical terms, the problem of power minimization can be reduced to the problem of minimizing the function $W(\xi, v)$ in the space of two design variables ξ and v, under the constraint $P(\xi, v) = \text{const}$. If we use finite difference notation for the discrete variable ξ,

$$\left.\frac{\triangle F(\xi, v)}{\triangle \xi}\right|_v = \frac{F(\xi + \triangle\xi, v) - F(\xi, v)}{\triangle \xi}, \quad (4)$$

wherein $F(\xi, v)$ is any function of variables ξ and v, involved in the analysis, and neglect the second-order terms, then the constraint condition can be expressed in differential form as

$$\left.\frac{\triangle P}{\triangle \xi}\right|_v \triangle\xi + \frac{\partial P}{\partial v}\triangle v = 0, \quad (5)$$

where $\triangle v$ is the adjustment in the supply voltage needed to compensate for performance loss or gain, resulting from the architectural modification $\triangle\xi$.

Here, and in the remainder of the paper, we neglect the second-order terms of the form $\frac{\partial^2 F}{\partial v^2}(\triangle v)^2$ and $\frac{\triangle \partial F}{\triangle\xi \partial v}\triangle\xi\triangle v$, where F is any function involved in the analysis, such as W, P, f, I, N. Thus, all formulas and conclusions in this section are only valid for 'small'

[1]Transition barriers are placed before functional units (FU) to prevent switching in unused FUs, or portions of FUs without the overhead of duplicating the operand latches.

variations to the architecture, such that the resulting relative increments in all involved functions, and in their derivatives, are small ($\frac{\triangle F}{F} \ll 1$, $\frac{\triangle F'}{F'} \ll 1$) and relative changes in the supply voltage, v, needed to compensate for the performance loss or gain, resulting from architectural modifications $\triangle\xi$, are also small, ($\frac{\triangle v}{v} \ll 1$). Implications arising from these assumptions are considered in section 5.

Under the above assumptions, the problem of establishing the energy efficiency of a particular modification to the architecture, $\triangle\xi$ can be reduced to that of finding a relation between relative changes in processor characteristics in (1) for which

$$\left.\frac{\triangle W}{\triangle\xi}\right|_{P=\text{const}} = \left.\frac{\triangle W}{\triangle\xi}\right|_{v} + \frac{\partial W}{\partial v}\left.\frac{\triangle v}{\triangle\xi}\right|_{P=\text{const}} < 0. \quad (6)$$

Using (2) and (3) and the assumptions stated above, we can calculate the finite differences and partial derivatives in the constraint formula (5) as follows:

$$\left.\frac{\triangle P}{\triangle\xi}\right|_{v} = \frac{I}{N}\left.\frac{\triangle f}{\triangle\xi}\right|_{v} + \frac{f}{N}\frac{\triangle I}{\triangle\xi} - \frac{fI}{N^2}\frac{\triangle N}{\triangle\xi}, \quad (7)$$

$$\frac{\partial P}{\partial v} = \frac{I}{N}\frac{\partial f}{\partial v} = \frac{IfF_v}{Nv}, \quad (8)$$

where F_v is the dimensionless partial derivative of the maximum clocking rate with respect to the supply voltage,

$$F_v = \frac{v}{f}\frac{\partial f}{\partial v}. \quad (9)$$

The value of F_v can be estimated empirically for a selected technology, supply voltage and the selected circuit style. To evaluate it, the designer can simulate the dependence of the delay through the hardware blocks that are expected to be on the critical path upon the supply voltage. Examples of the evaluation of F_v are considered in the next section.

Substituting expressions (7) and (8) into the constraint condition (5), we arrive at the following expression for the ratio of finite differences $\triangle v$ and $\triangle\xi$ subject to the constraint $P(\xi, v) = \text{const}$:

$$\left.\frac{\triangle v}{\triangle\xi}\right|_{P=\text{const}} = -\frac{v}{F_v f}\left.\frac{\triangle f}{\triangle\xi}\right|_{v} - \frac{v}{F_v I}\frac{\triangle I}{\triangle\xi} + \frac{v}{F_v N}\frac{\triangle N}{\triangle\xi}. \quad (10)$$

The remaining terms in the energy-efficiency formula (6) are calculated as follows:

$$\left.\frac{\triangle W}{\triangle\xi}\right|_{v} = IE\left.\frac{\triangle f}{\triangle\xi}\right|_{v} + fE\frac{\triangle I}{\triangle\xi} + fI\left.\frac{\triangle E}{\triangle\xi}\right|_{v}, \quad (11)$$

$$\frac{\partial W}{\partial v} = \frac{IEf}{v}(E_v + F_v), \quad (12)$$

where E_v is the dimensionless partial derivative of the average energy dissipated per instruction with respect to the supply voltage,

$$E_v = \frac{v}{E}\frac{\partial E}{\partial v}. \quad (13)$$

The value of E_v for CMOS circuits is typically close to 2, since the energy of the charged capacitance is proportional to the square of the supply voltage, $E = \frac{CV^2}{2}$. A more accurate estimate for the value of E_v for a selected technology and circuit style can be obtained by simulating representative circuits over a range of supply voltages. Examples of the evaluation of E_v are given in the next section.

Substituting (11), (12) and (10) into (6), and grouping terms in front of the partial derivatives, we arrive at the following criterion for energy efficiency:

$$-\frac{E_v}{F_v}\left.\frac{\triangle f}{f\triangle\xi}\right|_{v} - \frac{E_v}{F_v}\frac{\triangle I}{I\triangle\xi} + \left.\frac{\triangle E}{E\triangle\xi}\right|_{v} + \frac{F_v + E_v}{F_v}\frac{\triangle N}{N\triangle\xi} < 0 \quad (14)$$

The increments of all quantities in (14) appear in relative form and, thus are dimensionless. This feature makes this formula easy to use as a negotiation basis between architects and circuit designers. For example, if $E_v = F_v = 2$, then if some microarchitectural enhancement (say adding a bypass) increases the average energy per instruction by 5%, and potentially increases the delay on the critical path by 2%, without any effect on the dynamic instruction count, then it will be energy efficient only if the resulting increase in the architectural speed I is at least 7%. More examples on using the derived energy efficiency criterion are given in Section 4.

For some combinations of the values of E_v and F_v, the derived energy-efficiency criterion can be viewed upon as a differential form of one of the conventional power-performance metrics. For example, if $E_v = 2$ and $F_v = 1$, then (14) is reduced to

$$-2\frac{\triangle f}{f} - 2\frac{\triangle I}{I} + \frac{\triangle E}{E} + 3\frac{\triangle N}{N} < 0, \quad (15)$$

which is a differential form of the well-known "MIPS-cube per Watt" formula, $S = \frac{P^3}{W} = \frac{f^2 I^2}{N^3 E}$, assuming relations (2) and (3) for performance and power hold true. Indeed, according to the "MIPS-cube per Watt" metric, processor A with the "MIPS-cube per Watt" rating $S = S_A$ is considered a better design point than processor B with the "MIPS-cube per Watt" rating $S = S_B$ if and only if $S_A - S_B > 0$. If we denote $\triangle f = f_A - f_B$, $\triangle E = E_A - E_B$, $\triangle N = N_A - N_B$, and $\triangle I = I_A - I_B$, then the inequality can be re-written as

$$\left(1 + \frac{\triangle f}{f_B}\right)^2 \left(1 + \frac{\triangle I_B}{I_B}\right)^2 > \left(1 + \frac{\triangle N}{N_B}\right)^3 \left(1 + \frac{\triangle E}{E_B}\right) \quad (16)$$

If all $\triangle$'s are sufficiently small, then the above expression is equivalent to (15) to the accuracy of the second-order terms.

Similarly, if $E_v = 2$ and $F_v = 2$, then the energy-efficiency criterion (14) is reduced to

$$-\frac{\triangle f}{f} - \frac{\triangle I}{I} + \frac{\triangle E}{E} + 2\frac{\triangle N}{N} < 0, \quad (17)$$

which, in a similar way, can be shown to be equivalent to the differential form of the "MIPS-square per Watt" metric, provided that all assumptions stated earlier hold.

Finally, if $F_v \gg E_v$, (14) is reduced to

$$\frac{\triangle E}{E} + \frac{\triangle N}{N} < 0, \quad (18)$$

which, under the same assumptions, is equivalent to the differential form of the "MIPS per Watt" metric. Therefore, the "MIPS per Watt" metric that is commonly used for power analysis under the "fixed throughput" mode [4] leads to an energy-optimized design only if $F_v \gg E_v$.

Thus, the "MIPS per Watt", "MIPS-square per Watt", "MIPS-cube per Watt", and other similar "MIPS to the power of γ per Watt" metrics are special cases of the energy-efficiency criterion, derived in this paper. Advantages of the new metric are its generality and the ability to calculate the parameter γ for every particular case, taking into account technology and circuit characteristics.

2.1.2 *Constant-Power Optimization*

The energy-efficiency formula (14) appears to be also valid for the reverse problem of performance maximization, subject to the

constant power constraint, $W = \text{const}$. To show this, let us assume that, similarly to the previous case, the designer is allowed to change both the architectural complexity ξ and the power supply voltage v to achieve the maximum performance, while keeping the average power at the required level. This optimization goal is typical of the high-performance microprocessor design targeted at achieving the highest performance, without exceeding the power budget set by packaging.

To achieve this goal, the designer needs to evaluate if a particular modification to the architecture will result in higher performance, assuming that the clocking rate will be adjusted to meet the power budget and the power supply voltage will be adjusted accordingly, to enable the processor hardware to operate at the desired clocking rate. Then, the optimization problem can be formulated in mathematical terms as the problem of maximizing the function $P(\xi, v)$ in the space of two design variables ξ and v, under the constraint $W(\xi, v) = \text{const}$ which, under the assumptions stated earlier, can be expressed in the finite difference form as

$$\left.\frac{\triangle W}{\triangle \xi}\right|_v \triangle \xi + \frac{\partial W}{\partial v} \triangle v = 0. \tag{19}$$

Determining the energy efficiency of a particular modification to the architecture can then be reduced to finding a condition for which

$$\left.\frac{\triangle P}{\triangle \xi}\right|_{W=\text{const}} = \frac{\triangle P}{\triangle \xi} + \frac{\partial P}{\partial v} \left.\frac{\triangle v}{\triangle \xi}\right|_{W=\text{const}} > 0. \tag{20}$$

Substituting (11) and (12) into (19), we derive the following expression for the ratio of finite differences $\triangle v$ and $\triangle \xi$, under the constraint (19):

$$\left.\frac{\triangle v}{\triangle \xi}\right|_{W=\text{const}} = \frac{-v}{F_v + E_v} \left(\frac{1}{f} \left.\frac{\triangle f}{\triangle \xi}\right|_v + \frac{1}{I}\frac{\triangle I}{\triangle \xi} + \frac{1}{E} \left.\frac{\triangle E}{\triangle \xi}\right|_v \right). \tag{21}$$

Substituting (7) and (8) into the energy-efficiency equation (20), we arrive at (14). Thus, the energy-efficiency criterion (14) is also valid for the alternative formulation of the power-performance optimization problem, where the goal is to maximize performance without exceeding the power budget. Therefore, metric (14) should be used (instead of "MIPS per Watt") for optimizing high performance clock gated processors, when such a processor cannot be set to operate at its full speed because of the power constraint. The next subsection derives an energy-efficiency metric for processors that do not use any clock gating.

2.2 Worst-Case Power Analysis

The energy-efficiency criterion (14) deals with the average power of a processor, assuming that the average power is proportional to the weighted average number of instructions executed per cycle. In this subsection we derive a special version of the energy-efficiency criterion tailored for processors that do not use any clock gating.

The power-performance optimization analysis in this special case follows the same path as in case of ideal clock gating. The expression for the average power in the absence of clock gating is written as

$$W(\xi, v) = f(\xi, v) E(\xi, v), \tag{22}$$

where E is the average energy dissipated *per cycle*. The expression for performance (2) holds. Consequently, we only need to re-write formulas involving the power term, (11) and (12), as follows:

$$\left.\frac{\triangle W}{\triangle \xi}\right|_v = E \left.\frac{\triangle f}{\triangle \xi}\right|_v + f \left.\frac{\triangle E}{\triangle \xi}\right|_v, \tag{23}$$

$$\frac{\partial W}{\partial v} = \frac{E f}{v}(E_v + F_v). \tag{24}$$

Repeating the analysis for the constant-performance power optimization in subsection *2.1.1*, we arrive at the following energy-efficiency criterion:

$$-\frac{E_v}{F_v} \left.\frac{\triangle f}{f \triangle \xi}\right|_v - \frac{F_v + E_v}{F_v}\frac{\triangle I}{I \triangle \xi} + \left.\frac{\triangle E}{E \triangle \xi}\right|_v + \frac{F_v + E_v}{F_v}\frac{\triangle N}{N \triangle \xi} < 0. \tag{25}$$

It is easy to verify that, for the free-running clock implementation, the constant-power optimization, described in subsection *2.1.2*, leads to the same formula (25).

Compared to the corresponding expression for the ideal clock gating implementation (14), formula (25) has a larger weight in front of the term $\frac{\triangle I}{\triangle \xi}$. This is a consequence of the assumption that the average power is independent of the number of instructions executed per cycle.

Notice that expression (25) also holds for the *worst-case* power analysis in clock-gated microprocessors, if E is interpreted as *worst-case* energy dissipated *per cycle*. Therefore, if a processor is constrained by the worst-case sustained power that may be dissipated during the execution of a loop of power-intensive instructions with high degree of ILP, combined with high switching factors in the data bits, then metric (25) should be used for both clock-gated and non-gated implementations of the processor.

It is easy to show, following the reasoning in the previous subsection, that the "MIPS per Watt", "MIPS-square per Watt", "MIPS-cube per Watt", and other "MIPS-to-the-power-of-γ per Watt" metrics are special cases of the energy-efficiency criterion (25), written in the integral form. For example, $E_v = 2, F_v = 1$ leads to "MIPS-cube per Watt"; $E_v = 2, F_v = 2$ leads to "MIPS-square per Watt"; $F_v \gg E_v$ leads to "MIPS per Watt", while $E_v = 2, F_v = 0.5$ leads to "MIPS-power-5 per Watt".

2.3 Partial Clock Gating

In the design of real processors, clock gating may be applied to only some portion of the processor resources, or the granularity of clock gating may be coarser than assumed in subsection 2.1. Then, a linear combination of the energy-efficiency criteria (14) and (25), derived under the assumptions of the ideal clock gating and zero clock gating, leads to:

$$\begin{aligned} -\frac{E_v}{F_v}\frac{1}{f} \left.\frac{\triangle f}{\triangle \xi}\right|_v - \left(\frac{F_v + E_v}{F_v} - \kappa\right)\frac{1}{I}\frac{\triangle I}{\triangle \xi} \\ + \frac{1}{E} \left.\frac{\triangle E}{\triangle \xi}\right|_v + \frac{F_v + E_v}{F_v}\frac{1}{N}\frac{\triangle N}{\triangle \xi} \quad < \quad 0, \end{aligned} \tag{26}$$

where κ is the power-weighted portion of hardware covered by the clock gating, $0 < \kappa < 1$.

3. EFFECT OF CIRCUIT AND TECHNOLOGY CHARACTERISTICS

The proposed energy-efficiency metric is dependent on the characteristics of technology and circuits, through the parameters F_v and E_v, defined in (9) and (13). As shown in the previous section, different combinations of values of F_v and E_v may lead to different conclusions about the effectiveness of the same architectural features.

Theoretical formulas could be used to determine F_v and E_v. Alternatively, a more practical way to calculate the values of these coefficients is to simulate representative circuits over a range of power supply voltages. For the evaluation of F_v, it is important to select

functional block that can potentially be on the critical path, on the other hand, for the evaluation of E_v the most significant power consumers should be simulated.

As an illustration, a representative set of blocks in a typical microprocessor was selected, including an inter-unit star-connect data bus; a synthesized ASIC 32-bit integer adder; a full-custom 16-bit multiplier; the critical read path of the 4read/4write - port full custom register file (just simulation), described in [1]; and a 2read - 2write 16-entry semi-custom register file built of latches and multiplexors, all implemented in a 0.13um technology. For the energy analysis, all blocks were simulated with PowerMill, applying random patterns to the inputs with a switching factor of 0.3, for 200 to 500 cycles (depending on the size of the circuit). A clocking rate of 100MHz was used in power simulations for all values of Vdd. PathMill static timer was used for delay analysis. All derivatives were calculated by the 3-point formula.

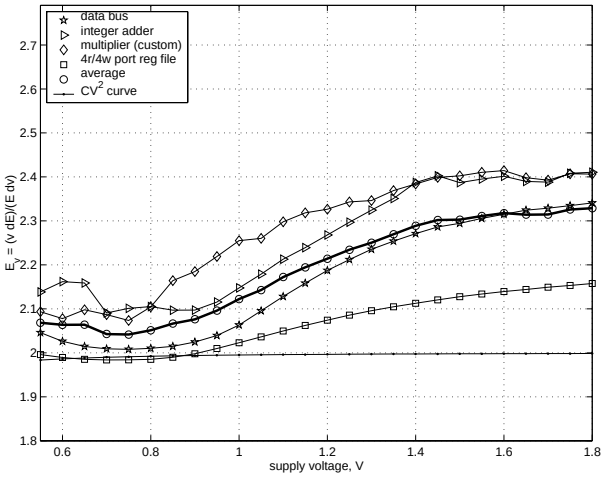

Figure 1: Simulation results for $E_v = \frac{v}{E}\frac{\partial E}{\partial v}$.

Fig 1 shows simulation results for E_v. The curves on the graph correspond to the blocks described above. A curve, corresponding to the $E = CV^2$ dependence is also plotted, as a reference. Fig. 1 shows that, for all the blocks, the value of E_v is higher than the value of two that corresponds to the $E = CV^2$ dependence. This super-Vdd-square dependence of energy on the supply voltage is partially explained by short circuit power which grows faster than the square of v [10], and higher glitching activity at higher supply voltages. Those blocks that have more significant glitching factors also demonstrate higher values of E_v, especially at high supply voltages. Detailed discussions of the factors affecting the dependence of E_v on v are beyond the scope of this paper.

Fig 2 shows simulation results for F_v. The curves on the graph correspond to the previously described blocks. For all blocks, F_v increases rapidly for low values of Vdd, especially as Vdd approaches the transistor threshold voltage. For high values of Vdd, F_v drops below unity because of the velocity saturation effect. For custom-designed blocks, F_v tends to be smaller than for ASIC-synthesized blocks, especially at low values of Vdd, because of the (selective) use of low-threshold devices in custom circuits, and low-voltage circuit styles (e.g. smaller transistor stacks).

The thick lines on the graphs, marked with circles, represent the averages over all simulated blocks, calculated for unity weight factors. For the analysis of a real microprocessor, F_v and E_v of different blocks should be averaged with appropriate weights.

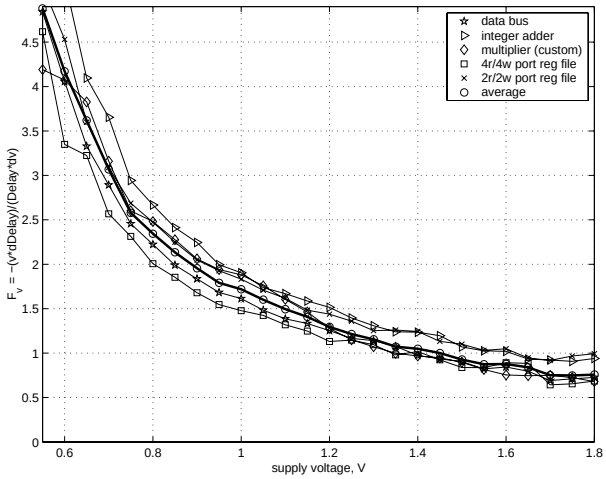

Figure 2: Simulation results for $F_v = \frac{v}{f}\frac{\partial f}{\partial v}$.

4. EXAMPLE OF USING THE ENERGY-EFFICIENCY CRITERION

In practice, a simplified form of equations (14), (25) and (26) can be used for comparing architectural alternatives, where $\triangle\xi$'s are omitted from the formulas. Then, for example, (14) is reduced to:

$$-\frac{E_v}{F_v}\frac{\triangle f}{f} - \frac{E_v}{F_v}\frac{\triangle I}{I} + \frac{\triangle E}{E} + \frac{F_v + E_v}{F_v}\frac{\triangle N}{N} < 0. \qquad (27)$$

It is important to note that, for calculating the finite increments $\triangle f$ and $\triangle E$, the meaning of partial derivatives with respect to the architectural complexity be preserved, as defined in (4). Particularly, the designer needs to assume a fixed supply voltage when calculating the increments in those quantities.

To illustrate the practical use of the energy efficiency criterion, let us consider examples of two hypothetical microprocessors: low-power microprocessor *A* that uses the fine-grain clock gating, covering close to 100% of the hardware, and high performance dynamic-issue microprocessor *B* that does not use any clock gating. Assume that microprocessors *A* and *B* are targeted to operate at Vdd=$1.0V$, and $1.7V$, respectively. Then, by looking at the curves for the average F_v and E_v in Fig. 2 and Fig. 1, we determine that $F_v = 1.72$, $E_v = 2.12$ for processor *A*, and $F_v = 0.75$, $E_v = 2.31$ for processor *B*.

As a first example, let us evaluate the energy efficiency of the execution bypass of the register file in processor *A*. Suppose that architectural-level simulation results show that, on a given set of benchmarks, the increase in the architectural speed (IPC) resulting from adding the bypass is $\frac{\triangle I}{I} = 7\%$. Moreover, suppose that hardware analysis reveals that the critical path delay increases by 5%, $\frac{\triangle f}{f} = -5\%$, because the register file read access happens to be on the critical path, and the average energy dissipated by instructions that read the register file increases 2% because of the bypass. If 80% of dynamic instructions read operands from the register file, then the average energy dissipated by an executed instruction increases $\frac{\triangle E}{E} = 1.6\%$. Since adding the bypass does not affect the dynamic instruction count, $\frac{\triangle N}{N} = 0$. Substituting these values, and the values of F_v and E_v, estimated above, into (14) or (27), we get $\frac{E_v}{F_v}(-\frac{\triangle f}{f} - \frac{\triangle \mathrm{IPC}}{I}) + \frac{\triangle E}{E} = 1.23 \cdot (0.05 - 0.07) + 0.016 < 0$. The energy-efficiency criterion indicates that for the stated assumptions, adding the bypass improves the energy efficiency of proces-

sor A. Notice, however, that the same feature would not be energy-efficient if processor A were targeted to operate at $Vdd = 0.7V$ or lower.

As a second example, consider a proposal to add one extra read port to the multiported integer register file in processor B, which will remove some restrictions on the issue of store instructions in parallel with arithmetic instructions. Suppose that simulations showed that this feature would improve the architectural performance by 0.5%. Assume that the register file access in not on the critical path, so that adding an extra read port does not impact the clocking rate. Assume also that the increase in the power dissipated in the register file (which is not clock gated) is 10%, and the integer register file is responsible for 15% of the total CPU power. Then, the increase in the average energy dissipated per cycle is $\frac{\triangle E}{E} = 1.5\%$. Substituting these values, and the values or F_v and E_v, estimated above, into (25), we get $-\frac{(E_v+F_v)}{F_v}\frac{\triangle I}{I} + \frac{\triangle E}{E} = -4.08 \cdot 0.005 + 0.015 < 0$. Thus, according to the energy-efficiency criterion for microprocessors without clock gating, adding an extra read port improves the energy efficiency of processor B. The same feature, however, would not be energy-efficient if processor B were targeted to operate at $Vdd = 1.2V$, or below.

These examples demonstrate the usefulness and convenience of the proposed energy-efficiency metric. In both examples, the relative changes in the characteristics of the processors were small, so that the assumptions, for which the formulas were derived, were satisfied.

5. CONCLUSIONS

A new architectural-level energy-efficiency metric was derived that subsumes other commonly used power-performance metrics as special cases of a more general equation. An advantage of the derived metric is that it takes into account the characteristics of circuits and technology to draw a conclusion about the energy efficiency of an architectural feature. In spite of being very general, the new formula is easy to use because it only involves relative changes in the characteristics of the processor, which can be evaluated even at early stages of the processor development. For those who feel more comfortable using the integral metric of the form $\frac{\text{MIPS}^\gamma}{\text{Watt}}$, this work provides a consistent and reliable method for calculating parameter γ, $\gamma = \frac{E_v+F_v}{F_v}$. Examples have been provided that illustrate the application of the proposed metric to a low-end and a high-performance processors.

For the validity of the derived formulas, the relative differences in the processor characteristics, corresponding to architectural alternatives under evaluation must be small, a 10% limit can be used for most practical purposes. Special care is needed, if the criterion is to be used to evaluate the energy efficiency of architectural features that result in significant changes in processor characteristics, such as increasing the issue width, or changing the width of data. Also, the conclusion may be misleading, if the formulas are used to compare different products on the market, especially those built in different technologies. Another limitation of the derived criterion is that it does not consider other important factors, such as the code size, ease of programming, or compilability of an architecture. Also, in order to be useful for future technologies, the formulas need to be extended to take into account the leakage power. These are among the targets of our current and future work.

Acknowledgment

The author would like to thank his colleagues J. Moreno, A. Zaks, U. Shvadron, P. Bose and A. Kudriavtsev for useful discussions; and K. Warren for the management support.

6. REFERENCES

[1] A. Alvandpour et al. A low leakage dynamic multi-ported register file in 0.13um CMOS. *IEEE Symposium on Low Power Electronics and Design*, August 2001.

[2] D. Brooks, P. Bose, et al. Power-aware microarchitecture: Design and modeling challenges for next-generation microprocessors. *IEEE MICRO*, 20(6):26–44, November 2000.

[3] T. Burd. *Energy-Efficient Processor System Design*. PhD thesis, University of California, Berkeley, 2001.

[4] T. Burd and R. Brodersen. Energy efficient CMOS microprocessor design. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, pages 288–297, 1995.

[5] A. Chandrakasan, S. Sheng, and R. Brodersen. Low-power CMOS digital design. *IEEE Journal of Solid-State Circuits*, 27(4):473–484, April 1992.

[6] R. Gonzalez and M. Horowitz. Energy dissipation in general purpose microprocessors. *IEEE Journal of Solid-State Circuits*, 31(9):1277–1283, September 1996.

[7] M. Horowitz, T. Indermaur, and R. Gonzalez. Low-power digital design. *Proceedings of the IEEE Symposium on Low Power Electronics*, pages 8–11, October 1994.

[8] L Jia, Y. Gao, and H. Tenhunen. New metrics for architectural level power performance evaluation. In *IEEE International Symposium on Circuits and Systems*, pages 549–552, May 2000.

[9] V. Tiwari et al. Reducing power in high-performance microprocessors. *Proceedings of the Design Automation Conference*, pages 732–737, 1998.

[10] J. Veendrick. Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. *IEEE Journal of Solid-State Circuits*, 19(4):468–473, August 1984.

[11] N. Vijaykrishnan et al. Energy-driven integrated hardware-software optimizations using simplepower. In *Proceedings of 27th Annual International Symposium on Computer Architecture*, pages 95–106, 2000.

[12] V. Zyuban and P. Kogge. Optimization of high-performance superscalar architectures for energy efficiency. In *IEEE Symposium on Low Power Electronics and Design*, pages 84–89, August 2000.

[13] V. Zyuban and P. Kogge. Inherently lower-power high-performance superscalar architectures. *IEEE Transactions on Computers*, 50(3), March 2001.