

Unified Methodology for Resolving Power-Performance Tradeoffs at the Microarchitectural and Circuit Levels

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

Philip Strenski
IBM T.J. Watson Research Center,
Yorktown Heights, NY
strensk@us.ibm.com

ABSTRACT

Evaluation of architectural tradeoffs is complicated by implications in the circuit domain which are typically not captured in the analysis but substantially affect the results. We propose a metric of hardware intensity (η), which is useful for evaluating issues that affect both circuits and architecture. Analyzing data for actual designs we show how to measure the introduced parameters and discuss variations between observed results and common theoretical assumptions. For a power-efficient design we derive relations for η and supply voltage V under progressively more general situations, and incorporate η into a prior art architectural energy-efficiency criterion. Then, a more general relation is derived for the optimal balance between the architectural complexity, hardware intensity and power supply. Modified forms for these relations are obtained in special cases where the supply voltage is constrained or when clock gating is disallowed.

Categories and Subject Descriptors

B.2.4 [High-Speed Arithmetic]: Cost/performance; B.2.1 [Design Styles]: Pipeline; B.6.1 [Design Styles]: Combinational logic, Parallel circuits; B.6.3 [Design Aids]: Optimization; B.7.1 [Types and Design Styles]: Microprocessors and microcomputers, VLSI; C.5.3 [Microcomputers]: Microprocessors; C.0 [General]: Modeling of computer architecture

General Terms

Design, Performance

Keywords

Energy, power, energy efficiency, hardware intensity, metric

Introduction

As power becomes an increasingly important constraint, it is necessary to include circuit power implications to evaluate correctly the impact of architectural changes. With this in mind, we propose a metric of hardware intensity η , useful for evaluating issues which

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'02, August 12-14, 2002, Monterey, California, USA.
Copyright 2002 ACM 1-58113-475-4/02/0008 ...\$5.00.

affect both circuits and microarchitecture. In the first section, we define η and other related parameters and illustrate how they can be measured with actual design data. Next we derive relations between η and supply voltage v for a power-efficient design under progressively more general conditions. Compared to the prior art [2, 6, 5], relations derived in this work are more general since they do not rely on simplifying assumptions about circuit characteristics. All introduced parameters have a clear physical meaning and a method for measuring them. Specifically we examine a single pipeline stage, multiple independent stages, and sequences within a stage. Finally we add hardware intensity to a formulation of architectural decision-making and show how a previous result [8] can be re-derived in a more general context. Special cases of this result are also produced for constrained supply voltage and absence of clock gating. The derived criterion subsumes other commonly used power-performance metrics [4, 1, 8] as special cases of a more general equation.

1. HARDWARE INTENSITY

In the design of pipelined processors the hardware in each stage is optimized through logic restructuring and tuning transistor sizes to meet the cycle requirement. The tighter the delay budget the more parallelism at the gate level is required and the larger transistor sizes are needed, which leads to higher power. To allow a mathematical approach to the analysis of these speed-power tradeoffs, we introduce a notion of *hardware intensity*, and a variable η associated with it. We define the physical meaning of η as a parameter in the cost function for optimizing hardware:

$$F_c = (E/E_0)(D/D_0)^\eta \quad 0 \leq \eta < +\infty, \quad (1)$$

where D is the critical path delay through the circuit, E is the average energy dissipated per cycle, D_0 and E_0 are the corresponding lower bounds that can be achieved through tuning and logic restructuring for a fixed supply voltage. Many types of functions can be used as a cost function. This particular form (1) was chosen because of the property:

$$\frac{\partial F_c}{\partial D} / \frac{\partial F_c}{\partial E} = \eta \frac{E}{D} \quad (2)$$

which makes it useful as a common language in circuit and architectural communities, as will become apparent in the following sections. Cost functions of form (1) have been used in previous works [4, 1, 9, 10, 6, 5] with fixed or variable η to optimize or compare hardware implementation in the power-performance space. In this paper we relate η to the power supply voltage in energy-efficient designs, and link it to the architectural energy efficiency criterion derived in [8].

A notion of the *energy-efficient family* was introduced in [9, 10] and later in [5] as a set of implementations of a given hardware function, each of which results in the highest performance among all possible configurations dissipating the same power. If plotted in the energy-versus-delay coordinates, the *energy-efficient* configurations form a *convex hull* of all possible implementations of a given hardware function. It is easy to show [5] that for any power supply voltage v , every point on the energy-efficient family corresponds to a certain value of the hardware intensity η , $0 \leq \eta < +\infty$. Then, the energy-efficient curve in the energy-versus-delay coordinates can be viewed as a parameterized curve: $D = D(\eta, v)$, $E = E(\eta, v)$.

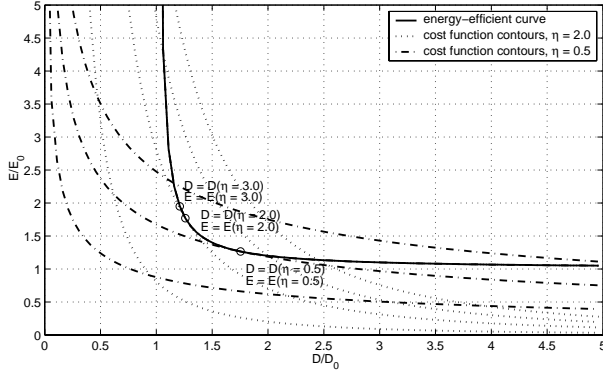


Figure 1: Typical energy-efficient curve and constant cost function contours for $\eta = 0.5$ and $\eta = 2.0$.

Figure 1 gives a graphical interpretation of the hardware intensity. The solid line plots a typical energy-efficient curve for some hardware function. Dotted lines show several contours of the cost function (1), for two values of the hardware intensity. Point (D, E) at which the energy-efficient curve touches the lowest of the contours ($F_c = A$ with the smallest value of A) corresponds to the energy-efficient implementation for this value of the hardware intensity. Using (2), the tangent to the energy efficient curve at this point can be expressed as

$$\left. \frac{\partial E}{\partial D} \right|_v = \frac{\partial E / \partial \eta}{\partial D / \partial \eta} = - \frac{\partial F_c / \partial D}{\partial F_c / \partial E} = -\eta \frac{E}{D}. \quad (3)$$

Then, we have the following property for the hardware intensity:

$$\eta = - \left. \frac{D \partial E}{E \partial D} \right|_v \quad (4)$$

Thus, the hardware intensity is the ratio of the relative increase in energy to the corresponding relative gain in performance achievable locally through logic restructuring and tuning at a fixed power supply voltage for a power-efficient design. Simply put, it is the value of % power per % performance for an energy-efficient design.

Fig. 2 shows on a logarithmic scale energy-efficient curves for two tuned adders, a vector reduction unit, a latch and several ASIC cells, all implemented in a 0.13 μ m technology (some in bulk, others in SOI). The energy-efficient curve for the latch was obtained by tuning several latches with a dynamic transistor-level Spice-base circuit tuner, run with different cost functions. The tuned points for all simulated latches were combined into a common energy-efficient family, as described in [10]. For ASIC cells, different power levels (from A to I) were used as points on the energy-efficient family, assuming that every ASIC cell is optimally tuned. Energy and delay values for the cells were looked up for various power levels directly from the design databook for the assumed

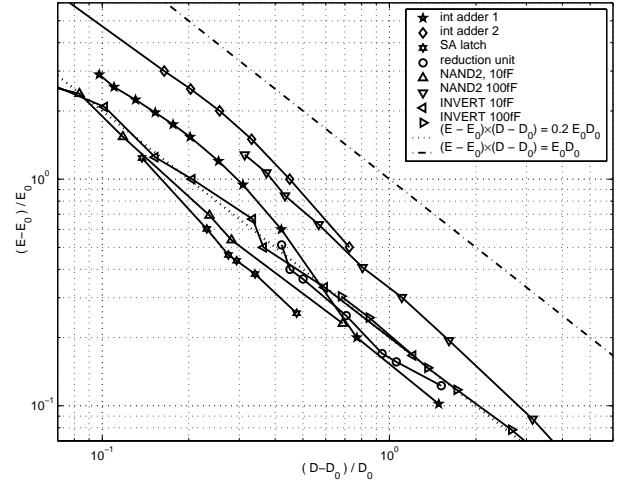


Figure 2: Energy-efficient curves for various hardware blocks built in 0.13 μ m technology.

load capacitances. The adder curves were obtained using formal static tuning EinsTuner [3] for a variety of targets for the total device width. The curve for the vector reduction unit was obtained using multiple ASIC synthesis runs for different frequency targets. IBM BooleDozer synthesis tool was used.

An interesting observation is that energy-efficient curves for widely different hardware functions, obtained using different methods are remarkably similar. A recent theoretical work [5] predicts the dependence $E = E(D)$ as $(E - E_0)(D - D_0) = E_0 D_0$, plotted as a dashed line. Our results in Fig. 2 show a substantial deviation from this prediction even for simple gates. However, the expression above can be modified to fit the experimental data as follows: $(E - E_0)(D - D_0) = \gamma E_0 D_0$, where $0 < \gamma < 1$.

To explain this form of the dependence, let us rewrite the expression $D = D_0 + RC_{ld}$, used for calculating delays of ASIC cell, as follows: $(D - D_0)/D_0 = \gamma C_{ld}/C_{cell}$, where C_{cell} is the sum of the cell input and internal capacitances, and $\gamma = RC_{cell}/D_0$ is approximately a constant value for every cell. For standard cells in a 0.13 μ m technology, the value of γ is in the range from 0.2 to 0.4, depending on the cell type. The expression for energy can be roughly approximated as $(E - E_0)/E_0 = C_{cell}/C_{ld}$. Multiplying the expressions for energy and delay, we arrive at $(E - E_0)(D - D_0) = \gamma E_0 D_0$. The dotted line in Fig. 2 that corresponds to $\gamma = 0.2$ is in much better agreement with the experimental results.

Through the remainder of the work we will only be interested in those implementations of any hardware that belong to the energy-efficient family.

2. DEPENDENCE OF ENERGY AND DELAY ON THE POWER SUPPLY

For the energy-efficiency analysis that follows it is useful to introduce the dimensionless derivatives of the delay and energy with respect to the power supply voltage, and their ratio:

$$E_v = \frac{v}{E} \frac{\partial E}{\partial v} \quad D_v = - \frac{v}{D} \frac{\partial D}{\partial v} \quad \theta = \frac{E_v}{D_v}. \quad (5)$$

Theoretical formulas could be used to determine D_v , E_v and θ as functions of v . Alternatively, a more practical way to calculate the values of these coefficients is to simulate representative circuits over a range of v .

For a fixed logic style, and a fixed technology we observed a close resemblance between the dependencies $E_v(v)$ and $D_v(v)$ for different functional units, and for hardware blocks optimized for different values of hardware intensity η .

As an illustration we plotted in Fig. 3 simulation results for a chain of XOR gates, and a 32-bit adder implemented in a 0.13um technology, tuned for several values of η . For the energy analysis PowerMill was used with random patterns at the inputs with a switching factor of 0.3, for 200 cycles. PathMill static timer was used for delay analysis.

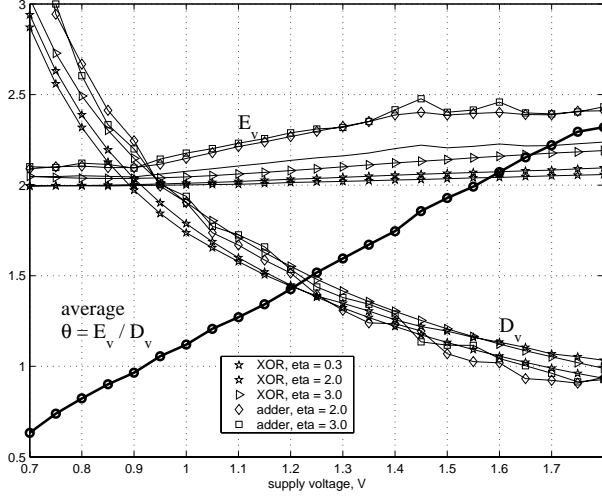


Figure 3: Simulation results for E_v , D_v , and θ .

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 explained by short circuit power which grows faster than the square of v [7], and higher glitching activity in large blocks of logic at higher supply voltages that we observed in our experiments. Although, curves for different circuits in Fig. 3 are very close to each other, we observed higher variation for hardware blocks designed in different circuit styles, or using different design flows [8].

3. BALANCE BETWEEN HARDWARE INTENSITY AND POWER SUPPLY

Typically, the cycle time requirement can be met at different combinations of η and v . In this section we derive a condition for the optimal balance between v and η , such that for a given critical path delay requirement $D = D_r$, the energy reaches its minimum over the two-dimensional space (η, v) . We will derive optimality relations for progressively more general assumptions about the pipeline, starting with a single-stage assumption, and ending with a general case of a multi-stage non-uniform pipeline. We also show how to abstract an aggregate η for non-uniformly optimized pipelines to be used in the microarchitecture level power optimization that follows.

3.1 Single pipeline stage

Consider an ‘ideal’ system in which the hardware is evenly distributed among multiple identical stages, which means that the same value of the hardware intensity η applies to all stages. By solving the problem of minimizing the energy function $E(\eta, v)$, subject to

the constant delay constraint $D(\eta, v) = D_r$, we arrive at:

$$\frac{\partial D}{\partial \eta} \frac{\partial E}{\partial v} = \frac{\partial D}{\partial v} \frac{\partial E}{\partial \eta}. \quad (6)$$

Using (4) and the definition for θ in (5), we arrive at:

$$\eta = \frac{E_v}{D_v} = \theta(v). \quad (7)$$

This formula can be interpreted as follows: for an optimal balance between the power supply voltage and the hardware intensity, the relative gain in performance achieved at a cost of a given increase in energy due to an increment in the supply voltage must equal the relative gain in performance achieved at a cost of a given increase in energy due to an increase in the hardware intensity.

With the help of (7), an optimal value for η can be determined for every value of v . For example, if for a given power supply voltage and technology $D_v = 1$ and $E_v = 2$, then, according to (7), for the optimal balance the hardware intensity must be set to $\eta = 2$, so that 1% gain in the critical path delay, achieved by re-tuning the circuit, costs 2% in the energy increase.

Relation (7) disproves the common misconception that the lowest power can be achieved by building the fastest circuit and then reducing the power supply to the lowest value for which the clocking rate requirement is still satisfied. For example, if $D_v = 1$ and $E_v = 2$ ($v = 1.6V$), and the circuit is optimized for $\eta = 4$ instead of $\eta = 2$, then the balance between power supply and hardware intensity is not optimal. It is easy to calculate for the circuit in Fig. 1 that by re-tuning the circuit for $\eta = 2$ and increasing the power supply appropriately for an unchanged performance, close to 10% power reduction will be achieved.

3.2 Multi-stage pipeline

Assume there are N stages in a pipeline which are different in the amount of logic and time slack. Then to achieve the optimum in the power-performance characteristics of the whole pipeline, the values of hardware intensity for different stages may be different. There are $N + 1$ independent variables corresponding to the hardware intensities in the N pipeline stages: η_1, \dots, η_N , and a single power supply, v .

Since all stages are optimized for the same clocking rate, $D_1 = D_2 = \dots = D_N$. Then, the problem is reduced to minimizing the function

$$E(\eta_1, \dots, \eta_N, v) = \sum_i E_i(v, \eta_i), \quad (8)$$

subject to N constraints

$$D_i(\eta_i, v) = D, \quad i = 1, \dots, N \quad (9)$$

Solving the optimization problem, and taking advantage of the earlier discussed property that E_v and D_v for all stages of the pipeline are equal, we arrive at:

$$\sum_i w_i \eta_i = \theta(v), \quad (10)$$

wherein $w_i = \frac{E_i}{E}$ are the energy weights of the pipeline stages, $\sum_i w_i = 1$. In the presence of clock gating the weights of those pipeline stages that are not activated every cycle are scaled down by the corresponding activity factors.

The optimality criterion (10) together with the cycle time requirement conditions (9) allow us to derive the optimal values for the hardware intensity at different stages of the pipeline as functions of the supply voltage. It can also be used to calculate the optimal value for the power supply voltage, after a preliminary version

of the pipeline is designed, by summing (with energy weights) the values of hardware intensities that were needed to meet the clock cycle target for every pipeline stage. If (10) is not satisfied, this indicates that power can be reduced without performance loss, by changing voltage and re-tuning circuits. Then this information can be used as feedback to re-evaluate the choice of the power supply voltage and the clock cycle target, and possibly the partitioning of the pipeline into stages.

It is easy to show that if (10) is satisfied, then the aggregate hardware intensity η_{ag} for the whole multi-stage pipeline optimized as a flat circuit, is related to the hardware intensities of individual stages η_i as follows:

$$\eta_{ag} = \sum_i w_i \eta_i. \quad (11)$$

Then (10) is identical to (7), with $\eta = \eta_{ag}$.

3.3 Composite pipeline stage

Pipeline stages usually consist of multiple blocks that are designed and optimized independently. At least two independent blocks can be distinguished in any conventional pipeline: latches and logic that are usually designed and tuned independently of each other. Consequently, different blocks in the same pipeline stage may have different values for the optimal hardware intensity. Then, there are $M + 1$ independent variables corresponding to the hardware intensities in the M blocks of a pipeline stage: η_1, \dots, η_M , and the single power supply voltage v . The goal is to find a relation between η_1, \dots, η_M and v , that leads to the minimum energy

$$E(\eta_1, \dots, \eta_M, v) = \sum_i E_i(v, \eta_i), \quad (12)$$

subject to the total delay requirement D_r which, disregarding inter-block delay coupling effects, can be written as:

$$D(\eta_1, \dots, \eta_M, v) = \sum_i D_i(v, \eta_i) = D_r \quad (13)$$

Solving this problem we arrive at

$$\frac{w_i}{u_i} \eta_i = \theta(v), \quad 1 \leq i \leq M, \quad (14)$$

where u_i is the delay weight of block i , $u_i = \frac{D_i}{D}$, and w_i is the corresponding energy weight, $w_i = \frac{E_i}{E}$, calculated taking into account the activity factors in clock-gated designs. If (14) is satisfied, then the aggregate hardware intensity η_{ag} for a composite stage optimized as a flat circuit, is related to the hardware intensities of individual sub-blocks η_i as follows:

$$\eta_{ag} = \frac{w_i}{u_i} \eta_i, \quad 1 \leq i \leq M \quad (15)$$

Thus, in a pipeline stage that consists of multiple blocks designed independently, those blocks that have lower energy weight and higher delay weight should be designed more aggressively than blocks with lower delay weight and higher energy weight. For example, suppose, $E_v = 2$ and $F_v = 1$. Consider a pipeline in which every stage consists of a block of latches and a cloud of logic. Assume that the latch delay budget is 20% of the cycle time and the one for the logic is 80%. Furthermore, assume that latches are responsible for 60% of the total power. Then, using (14), the optimum hardware intensity for latches is $\eta_1 = \frac{0.2}{0.6} 2.0 = 0.67$, and that for the logic is $\eta_2 = \frac{0.8}{0.4} 2.0 = 4.0$. Thus, for these assumptions logic must be optimized much more aggressively than latches.

3.4 Multi-stage pipeline with composite stages

Suppose the pipeline consists of N stages, and there are at most M sub-blocks in each pipeline stage that are designed independently of each other. Let E_{ij} be the energy dissipated in sub-block j of pipeline stage i , D_{ij} be the corresponding critical path delay, and η_{ij} be the corresponding hardware intensity, $1 \leq i \leq N$, $1 \leq j \leq M$. The goal is to minimize the total energy in the space on $N \times M + 1$ variables:

$$E(\eta_{11} \dots \eta_{1M}, \dots, \eta_{N1} \dots \eta_{NM}, v) = \sum_{ij} E_{ij}(v, \eta_{ij}), \quad (16)$$

subject to the N constraints:

$$\sum_j D_{ij}(v, \eta_{ij}) = D_r, \quad 1 \leq i \leq N \quad (17)$$

Solving this problem we arrive at

$$\sum_{i=1}^N \frac{w_{ij}}{u_{ij}} \eta_{ij} = \theta(v), \quad 1 \leq j \leq M, \quad (18)$$

where u_{ij} is the delay weight of sub-block j in pipeline stage i , $u_{ij} = \frac{D_{ij}}{D}$, and w_{ij} is the corresponding energy weight, $w_{ij} = \frac{E_{ij}}{E}$, calculated taking into account the activity factors.

If (18) is satisfied, then the aggregate hardware intensity η_{ag} is expressed through the hardware intensities of individual sub-blocks η_{ij} as follows:

$$\eta_{ag} = \sum_{i=1}^N \frac{w_{ij}}{u_{ij}} \eta_{ij}, \quad 1 \leq j \leq M. \quad (19)$$

Using this definition, the optimality relation (18) is equivalent to (7), with $\eta = \eta_{ag}$.

4. MICROARCHITECTURE - HARDWARE INTENSITY BALANCING

We now introduce a third variable into the analysis, called the architectural complexity ξ which, unlike v and η , is discrete [8]. 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.

According to the previous section, in the most general case of an N -stage pipeline, where each stage is composed of up to M individually designed blocks, there are up to $N \times M$ independent hardware intensity variables, η_{ij} , $0 \leq i \leq N$, $0 \leq j \leq M$. We will replace all these variables with a single hardware intensity variable for the whole processor η , defined as (19), assuming that hardware intensities in sub-blocks of individual pipelines are related by (17) and (18).

Then, the performance and power characteristics of a processor can be viewed as functions of the independent variables ξ , η and v :

$$\begin{aligned} \text{dynamic instruction count} & N = N(\xi) \\ \text{architectural speed (IPC)} & I = I(\xi) \\ \text{maximum clocking rate} & f = f(\eta, \xi, v) \\ \text{energy per instruction} & E = E(\eta, \xi, v) \end{aligned} \quad (20)$$

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, calculated on the same benchmark suite; E is the average energy per

instruction, calculated as $E = \sum_i n_i E_i$, where E_i is the average energy dissipated on the execution of instruction i from the instruction set, and n_i is the normalized dynamic frequency of the corresponding instructions. Then, the processor performance P on the given benchmark suite can be expressed as follows:

$$P(\xi, \eta, v) = \frac{f(\xi, \eta, v)I(\xi)}{N(\xi)}. \quad (21)$$

The expression for power dissipation $W(\xi, \eta, v)$ depends on the implementation details of the processor. We will consider cases of ideal clock gating and free-running clock implementations.

4.1 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. 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:

$$W(\xi, \eta, v) = f(\xi, \eta, v)I(\xi)E(\xi, \eta, v). \quad (22)$$

If expression (22) 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 .

Let us consider the problem of minimizing the average power dissipation, given a performance requirement, $P = P_r$. The designer is allowed to modify the architecture (both ISA and microarchitecture) and adjust the clocking rate of the processor, by changing the hardware intensity and power supply voltage to satisfy the performance requirement at minimum power dissipation. Then the problem of power minimization can be reduced to the problem of minimizing the function $W(\xi, \eta, v)$ in the space of the three design variables ξ , η and v , under the constraint $P(\xi, \eta, v) = P_r$. If we use finite difference notation for the discrete variable ξ ,

$$\frac{\Delta F(\xi, \eta, v)}{\Delta \xi} \Big|_{\eta v} = \frac{F(\xi + \Delta \xi, \eta, v) - F(\xi, \eta, v)}{\Delta \xi}, \quad (23)$$

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

$$\frac{\Delta P}{\Delta \xi} \Big|_{\eta v} \Delta \xi + \frac{\partial P}{\partial v} \Delta v + \frac{\partial P}{\partial \eta} \Delta \eta = 0, \quad (24)$$

where $\Delta \eta$ and Δv are adjustment in the hardware intensity and supply voltage needed to compensate for performance loss or gain, resulting from the architectural modification $\Delta \xi$. Here, and in the remainder of the paper, we neglect second-order terms. All formulas and conclusions in this section are only valid for ‘small’ variations to the architecture, such that the resulting relative increments in all involved functions, and in their derivatives, are small ($\frac{\Delta F}{F} \ll 1$, $\frac{\Delta F'}{F'} \ll 1$) and relative changes in the supply voltage v and the hardware intensity η , needed to compensate for the performance loss or gain, resulting from architectural modifications $\Delta \xi$, are also small, ($\frac{\Delta v}{v} \ll 1$).

Under the above assumptions, the problem of establishing the energy efficiency of a particular modification to the architecture, $\Delta \xi$ can be reduced to that of finding a relation between relative changes in processor characteristics in (20) for which

$$\frac{\Delta W}{\Delta \xi} \Big|_P = \frac{\Delta W}{\Delta \xi} \Big|_{\eta v} + \frac{\partial W}{\partial \eta} \frac{\Delta \eta}{\Delta \xi} \Big|_P + \frac{\partial W}{\partial v} \frac{\Delta v}{\Delta \xi} \Big|_P < 0. \quad (25)$$

Using (21) and (22) and the assumptions stated above, we can calculate the finite differences and partial derivatives in the constraint formula (24) as follows:

$$\frac{\Delta P}{\Delta \xi} \Big|_{\eta v} = \frac{I}{N} \frac{\Delta f}{\Delta \xi} \Big|_{\eta v} + \frac{f}{N} \frac{\Delta I}{\Delta \xi} - \frac{fI}{N^2} \frac{\Delta N}{\Delta \xi}, \quad (26)$$

$$\frac{\partial P}{\partial v} = \frac{IfD_v}{Nv}, \quad \frac{\partial P}{\partial \eta} = -\frac{If}{ND} \frac{\partial D}{\partial \eta}. \quad (27)$$

Substituting (26) and (27) into the constraint condition (24), we arrive at the following expression for the ratio of finite differences $\Delta \eta$, Δv and $\Delta \xi$ subject to the constraint $P(\xi, v) = P_r$:

$$\frac{D_v}{v} \frac{\Delta v}{\Delta \xi} \Big|_P - \frac{1}{D} \frac{\partial D}{\partial \eta} \frac{\Delta \eta}{\Delta \xi} \Big|_P = \frac{\Delta N}{N \Delta \xi} - \frac{\Delta f}{f \Delta \xi} \Big|_{\eta v} - \frac{\Delta I}{I \Delta \xi}. \quad (28)$$

The remaining terms in (25) are calculated as follows:

$$\frac{\Delta W}{\Delta \xi} \Big|_{\eta v} = IE \frac{\Delta f}{\Delta \xi} \Big|_{\eta v} + fE \frac{\Delta I}{\Delta \xi} + fI \frac{\Delta E}{\Delta \xi} \Big|_{\eta v},$$

$$\frac{\partial W}{\partial v} = \frac{IEf}{v} (E_v + D_v), \quad \frac{\partial W}{\partial \eta} = IEf \left(\frac{1}{E} \frac{\partial E}{\partial \eta} - \frac{1}{D} \frac{\partial D}{\partial \eta} \right).$$

Substituting these expressions into (25) and taking advantage of property (3) for the hardware intensity, and, we arrive at the following relation for energy efficiency:

$$\frac{\Delta f}{f \Delta \xi} \Big|_{\eta v} + \frac{\Delta I}{I \Delta \xi} + \frac{\Delta E}{E \Delta \xi} \Big|_{\eta v} < (1 + \eta) \frac{\partial D}{D \partial \eta} \frac{\Delta \eta}{\Delta \xi} \Big|_P - (\theta + 1) \frac{D_v \Delta v}{v \Delta \xi} \Big|_P$$

If the processor is designed according to the optimal balance between the power supply and the hardware intensity (7) or (18) then $\eta = \theta$. Then, using the constraint formula (28) that relates the finite differences $\Delta \xi$, $\Delta \eta$ and Δv , the last expression is reduced to:

$$-\eta \frac{\Delta f}{f \Delta \xi} \Big|_{\eta v} - \eta \frac{\Delta I}{I \Delta \xi} + \frac{\Delta E}{E \Delta \xi} \Big|_{\eta v} + (\eta + 1) \frac{\Delta N}{N \Delta \xi} < 0 \quad (29)$$

Now, the increments of the architectural complexity $\Delta \xi$ can be omitted from the formula, as long as a fixed supply voltage and hardware intensity are assumed when calculating ΔE and Δf , and thus, the meaning of partial derivatives as defined in (24) is preserved. Then, a simplified form of the criterion can be used:

$$-\eta \frac{\Delta f}{f} - \eta \frac{\Delta I}{I} + \frac{\Delta E}{E} + (\eta + 1) \frac{\Delta N}{N} < 0, \quad \eta = \theta. \quad (30)$$

Thus, for a processor designed according to the optimal balance between hardware intensity and power supply ($\eta = \theta$), we were able to recover, under a much more general formulation of the optimization problem, the same expression for the energy efficiency as in [8]. Expression (30) not only allows the development of energy-efficient architecture, but also provides a basis for negotiations between architects and circuit designers, in terms that are well understood in both communities. It also shows that to achieve an energy-efficient design, architecture-level decisions must be balanced both with the choice of the power supply voltage and the hardware intensity needed to make the clock cycle. Since (30) involves only relative changes in the characteristics of the processor, it can be used even at early stages of the processor development. For those who prefer the integral metric of the form $\frac{MIPS^T}{Watt}$, expression (30)

provides a consistent and reliable method for calculating the power, $\gamma = \eta + 1 = \theta + 1$.

4.2 Power supply-constrained optimum

In the design of high-performance microprocessor, a business decision may be made to deliver a higher clocking rate than that achievable at the point of the optimal balance between hardware intensity and power supply voltage (18). If the power supply voltage is raised to the limit set by the technology reliability $v = v'$, but circuits still do not deliver the required speed at the optimal value of the hardware intensity (18), then circuits may be optimized even more aggressively, which results in a higher-than-optimal value of η , $\eta = \eta' > \theta$. Then, the energy-efficiency relation (30) is not valid, since v is no longer an independent variable. The problem of power minimization is reduced to the problem of minimizing function $W(\xi, \eta, v = v')$ in the space of only two design variables ξ and η , under the constraint $P(\xi, \eta, v = v') = P_r$. Repeating the analysis above in the two-variable space, we arrive at:

$$-\eta' \frac{\Delta f}{f} - \eta' \frac{\Delta I}{I} + \frac{\Delta E}{E} + (\eta' + 1) \frac{\Delta N}{N} < 0, \quad \eta' > \theta \quad (31)$$

Compared to the corresponding expression for the optimally balanced power supply and hardware intensity (30), formula (31) has a smaller weight in front of the term $\frac{\Delta E}{E}$. Thus, under the described scenario the architectural energy-efficiency criterion will value improvements in the speed performance more than in case of an optimally balanced design. For example, if $E_v = 2$, $D_v = 1$, and $\eta' = 3$ expression (31) leads to “MIPS-power-4 per Watt”.

Although derived for the fixed performance assumption, relations (30) and (31) are also valid for the alternative formulation of the power-performance optimization problem, where the goal is to maximize performance without exceeding the power budget.

4.3 Worst-Case Power

The energy-efficiency criteria (30) and (31) deal with the average power. In the design of server-class processors the goal may be set to achieve the highest performance without exceeding a power limit even for the worst case instruction scheduling scenario. In this case the following expression for power should be used in place of (22)

$$W(\xi, \eta, v) = f(\xi, \eta, v)E(\xi, \eta, v), \quad (32)$$

where E is the *worst-case* energy dissipated *per cycle*. This expression also holds for the average power if E is interpreted as *average* energy dissipated *per cycle* in processors that do not use any clock gating. Repeating the analysis in the previous section we arrive at:

$$-\eta \frac{\Delta f}{f} - (\eta + 1) \frac{\Delta I}{I} + \frac{\Delta E}{E} + (\eta + 1) \frac{\Delta N}{N} < 0 \quad (33)$$

Compared to the corresponding expression for the ideal clock gating implementation (30), formula (33) has a larger weight in front of term ΔI .

5. CONCLUSIONS

The concept of hardware intensity leads to a number of quantitative relations which can be used to communicate information between circuit designers and architects. Circuit designers can use existing designs to provide typical hardware intensity values to architects for use in evaluating the power-efficiency of a starting design. Architects in turn can use these relations to provide guidance to the circuit designers on appropriate levels of power/performance to target. Note that the metric η can be used as a target for circuit tuning, or evaluated for a tuned circuit straightforwardly. The relations on η also provide guidance for choosing appropriate supply

voltage. Overall attention to these concepts insures a more power-efficient design.

Acknowledgment

The authors would like to thank P. Bose for valuable discussion, T. Fox for synthesizing some of the test unit and K. Warren and J. Moreno for the management support.

6. REFERENCES

- [1] 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.
- [2] A. Chandrakasan, S. Sheng, and R. Brodersen. Low-power CMOS digital design. *IEEE Journal of Solid-State Circuits*, 27(4):473–484, April 1992.
- [3] A. Conn et al. Gradient-based optimization of custom circuits using a static-timing formulation. In *Proceedings of Design Automation Conference*, pages 452–459, June 1999.
- [4] R. Gonzalez and M. Horowitz. Energy dissipation in general purpose microprocessors. *IEEE Journal of Solid-State Circuits*, 31(9):1277–1283, September 1996.
- [5] P. Penzes and A. Martin. Energy-delay efficiency of VLSI computations. In *Proceedings of the Great Lakes Symposium on VLSI*, pages 104–107, April 2002.
- [6] M. Stan. Low-power CMOS with subvolt supply voltages. *IEEE Transactions on VLSI Systems*, 9(2):394–400, April 2001.
- [7] 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.
- [8] V. Zyuban. Unified architecture level energy-efficiency metric. In *Proceedings of the Great Lakes Symposium on VLSI*, pages 24–29, April 2002.
- [9] 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.
- [10] V. Zyuban and D. Meltzer. Clocking strategies and scannable latches for low power applications. In *IEEE Symposium on Low Power Electronics and Design*, pages 346–351, August 2001.