Balancing hardware intensity in microprocessor pipelines

V. Zyuban
P. N. Strenski

The 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 illustrate the use of these equations in simple examples. Then we establish a relation between the architectural energy-efficiency metric and hardware intensity, and we derive expressions for evaluating the effect of modifications at the microarchitectural level on processor frequency and power, assuming the optimal tuning of the pipeline. These relations will guide the architect to achieve an energy-optimal balance between architectural complexity and hardware intensity.

Introduction

As power becomes an increasingly important constraint, it is necessary to include circuit power implications to evaluate correctly the impact of architectural changes. In previous works [1–8], this has been attempted with broad metrics combining global power and performance. For example, maximizing MIPS^n/watts and minimizing power $\times$ delay^n are expressed as goals. Arguments are made that $n = 0$ (power per operation) and $n = 1$ (energy per operation) are inadequate for evaluating tradeoffs, and $n = 2$ (energy–delay product) is commonly used. Attention to supply-voltage scaling [8] gives $n = 3$ as more appropriate in some domains. This reference also provides a good overview and a more refined version of the metric.

Some issues are common to these approaches. First, the exponent is global and typically integer, while the power–performance tradeoffs at the circuit level are generally local and continuous. Second, correct evaluation of the terms is often difficult because important side effects are easily neglected. For example, using MIPS^n/watts, it is straightforward to estimate changes in instructions executed per cycle (IPC); however, MIPS and watts also include changes in frequency that result from added logic (significant and often neglected), or operation near the power limit. Changes with marginal IPC improvement are much more likely to be accepted when such circuit power implications are overlooked.

With these issues in mind, we propose a metric of hardware intensity η, useful for evaluating issues which 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. We next derive relations between η and supply voltage v for a power-efficient design under progressively more general conditions. Specifically, we examine a single pipeline stage, multiple independent stages, and sequences within a stage. These relations allow the metric to overcome the first issue above. Hardware intensity is continuous, and can be used in both global and local

0018-8646/03/$5.00 © 2003 IBM

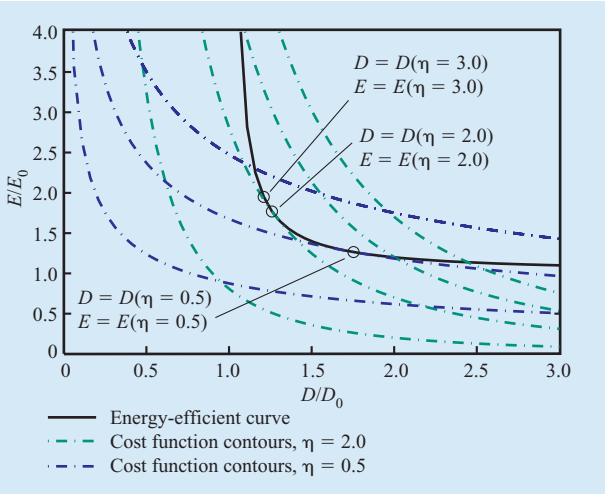

Figure 1

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

contexts as appropriate. No assumptions concerning technology voltage scaling are required. All introduced parameters have clear physical meanings and a method for measuring them.

After this we show how hardware intensity can be incorporated into an existing architectural metric. Some of the terms can be straightforwardly measured or estimated. For others we derive equations relating them to expressions involving hardware intensity and calculable terms. We show how, under simplifying assumptions, the more common metrics can be derived. These equations explicitly capture the effects which are often neglected in the second issue above. We conclude with a brief summary and a list of possible extensions of the ideas.

Hardware intensity

In the design of pipelined processors, the hardware in each stage is optimized by restructuring logic and tuning transistor sizes to meet the cycle requirement. The tighter the delay budget, the greater the parallelism required at the gate level and the larger the transistor sizes needed, which leads to higher power. To quantify 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:

$$F_{\text{cost}}(E, D) = (E/E_0)(D/D_0)^{\eta} \qquad \eta \geq 0, \tag{1}$$

where D is the critical path delay through the circuit, E is the average energy dissipated per cycle, and 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

$$\frac{\partial F_{\text{cost}}}{\partial D} \Big/ \frac{\partial F_{\text{cost}}}{\partial E} = \eta \frac{E}{D}, \tag{2}$$

which makes it useful as a common language in circuit and architectural communities, as is apparent in the following sections. Cost functions of form (1) have been used in previous work [3, 4, 6, 7, 9, 10] with fixed or variable η to optimize or compare hardware implementations 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 [11].

A notion of the *energy-efficient family* was introduced in [7], [9], and [10] 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.

Under a very general assumption that the curvature of the energy–delay curve is higher than the curvature of the contour of the cost function (1) at any point at which the two touch,

$$\frac{D^2}{E}\frac{\partial^2 E}{\partial D^2} > \eta(\eta + 1),$$

we can show that for any power-supply voltage v, every point on the energy–delay curve 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 viewed as a parameterized curve: $D = D(\eta, v)$, $E = E(\eta, v)$.

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

$$\left.\frac{\partial E}{\partial D}\right|_v = \frac{\partial E(\eta, v)}{\partial \eta} \Big/ \frac{\partial D(\eta, v)}{\partial \eta} = -\frac{\partial F_{\text{cost}}}{\partial D} \Big/ \frac{\partial F_{\text{cost}}}{\partial E} = -\eta \frac{E}{D}. \tag{3}$$

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

$$\eta = -\left.\frac{D\partial E}{E\partial D}\right|_{v}, \tag{4}$$

or

$$\eta = -\frac{D}{E}\frac{\partial E/\partial \eta}{\partial D/\partial \eta}. \tag{5}$$

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 % energy per % performance for an energy-efficient design:

$$\eta = -\left.\frac{\%E}{\%D}\right|_{\text{through retuning}}. \tag{6}$$

Figure 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-μm 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-based 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 load capacitances. The adder curves were obtained using a formal static tuner, EinsTuner [12], 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. The 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 [7] predicts the dependence $E = E(D)$ as $(E - E_0)(D - D_0) = E_0D_0$, plotted as a dashed curve. Our results in Figure 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: $(E - E_0)(D - D_0) = \gamma E_0D_0$, 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 the 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

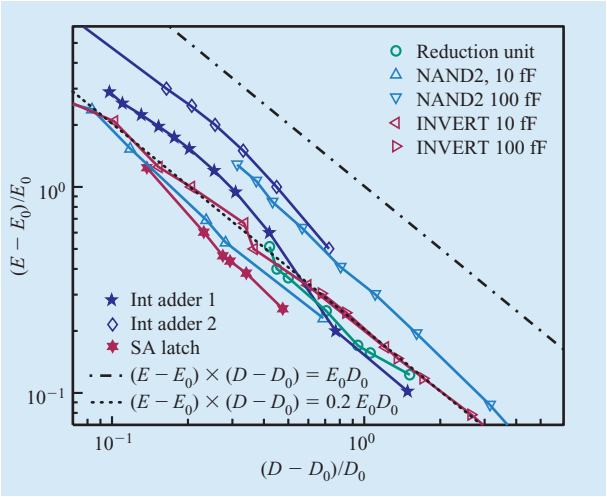

Figure 2

Energy-efficient curves for various hardware blocks built in 0.13-μm technology.

capacitances, C_{ld} is the load capacitance, and $\gamma = RC_{cell}/D_0$. The value of γ is approximately constant for every type of cell across a range of power levels, because the output resistance R of a cell is roughly inversely proportional to the sizes of transistors used in the cell, and thus, inversely proportional to C_{cell}, $R \sim 1/C_{cell}$. For standard cells in a 0.13-μm 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 proportional to the sum of the cell capacitance and the load capacitance, $E \sim (C_{cell} + C_{ld})$. If $C_{cell} \ll C_{ld}$ for the minimum-size cell, the expression can be further approximated as $(E - E_0)/E_0 = C_{cell}/C_{ld}$, where E_0 is the energy dissipated by the minimum-size cell. Multiplying the expressions for energy and delay, we arrive at $(E - E_0)(D - D_0) = \gamma E_0D_0$. The dashed curve in Figure 2 that corresponds to $\gamma = 0.2$ is in much better agreement with the experimental results.

The formula for the energy–delay curve can also be derived using the logical effort delay model [13] as follows: $D = \tau(gh + P)$, where τ is the intrinsic delay of an inverter, g is the logical effort of the gate, h is the ratio of load capacitance to input capacitance, $h = (C_{out}/C_{in})$, and τP is the delay of the gate driving zero load, $\tau P = D_0$. Then

$$\frac{D - D_0}{D_0} = \frac{g}{P}\frac{C_{out}}{C_{in}}.$$

Approximating energy for a fixed output load as

$E \sim C_{in} + C_{out}$ and

$E_0 \sim C_{out}$, we get $\dfrac{E - E_0}{E_0} = \dfrac{C_{in}}{C_{out}}$,

hence

$$\frac{D - D_0}{D_0}\frac{E - E_0}{E_0} = \frac{g}{P} = \gamma.$$

The range of values of $\gamma (0.2 < \gamma < 0.4)$ that we measured is consistent with data reported in [13] for an 0.18-μm technology.

Through the remainder of the work, we assume that all implementations of any hardware belong to the energy-efficient family; however, none of the results depend on any analytical formula for the shape of the energy–delay curves.

Voltage intensity

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, referred to as *voltage intensity:*

$$E_v = \frac{v}{E}\frac{\partial E}{\partial v}, \qquad D_v = -\frac{v}{D}\frac{\partial D}{\partial v}, \qquad \theta = \frac{E_v}{D_v}. \tag{7}$$

Thus, the voltage intensity is the ratio of the relative increase in energy to the corresponding reduction in delay achievable locally through varying the power supply at a fixed hardware intensity:

$$\theta = \left.\frac{\%E}{\%D}\right|_{\substack{\text{through}\\ \text{varying } V_{dd}}}.$$

Theoretical formulas could be used to predict 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 **Figure 3** simulation results for a chain of XOR gates and a 32-bit adder implemented in a 0.13-μm 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, run for 200 cycles. The PathMill** static timer was used for delay analysis.

For all of the blocks, the value of E_v is higher than the value of 2 that corresponds to the $E = CV^2$ dependence. This superquadratic dependence of energy on the supply voltage is explained by short-circuit power that grows faster than the square of v [14], and by the higher glitching activity in large blocks of logic at higher supply voltages that we observed in our experiments. Although curves for different circuits in Figure 3 are very close to one another, we observed higher variation for hardware blocks designed in different circuit styles or using different design flows [11]. According to the experimental results in Figure 3, the *voltage intensity* θ grows almost linearly with the power supply v.

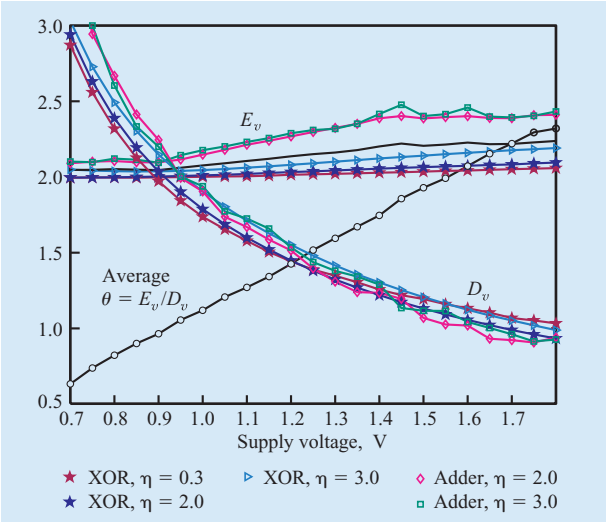

Figure 3

Simulation results for E_v, D_v, and θ.

Balance between hardware intensity and voltage intensity

Typically, the cycle-time requirement can be met at different combinations of hardware intensity η and power supply 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 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 multistage nonuniform pipeline. We also show how to abstract an aggregate hardware intensity η_{ag} for nonuniformly optimized pipelines to be used in the microarchitecture-level power optimization that follows.

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 as a function of two variables η and v, $E(\eta, v)$,

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

$$\frac{\partial D}{\partial \eta}\frac{\partial E}{\partial v} = \frac{\partial D}{\partial v}\frac{\partial E}{\partial \eta}. \quad (8)$$

Using (4) and the definition for the voltage intensity θ in (7), we arrive at

$$\eta = \frac{E_v}{D_v} = \theta(v): \quad (9)$$

Hardware intensity must equal voltage intensity. 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 the cost of a given relative increase in energy due to an increment in the supply voltage must equal the relative gain in performance achieved at the cost of a given relative increase in energy due to an increase in the hardware intensity.

With the help of (9), an optimal value for η can be determined for every value of v. For example, if $D_v = 1$ and $E_v = 2$ for a given power-supply voltage and technology, then, according to (9), for the optimal balance the hardware intensity must be set to $\eta = 2$, so that 1% gain in the critical path delay, achieved by retuning the circuit, costs 2% in the energy increase.

Relation (9) disproves the common misconception that the lowest power can 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 $E_v = 2$ ($v = 1.6$ V), and the circuit is optimized for $\eta = 4$ instead of $\eta = 2$, the balance between power supply and hardware intensity is not optimal. It is easy to calculate for the circuit in Figure 1 that by retuning the circuit for $\eta = 2$ and increasing the power supply appropriately for an unchanged performance, power reduction close to 10% will be achieved.

Multistage pipeline

The simple case of an isolated hardware macro considered above can be applied only to an ideally uniform pipeline. In real designs, different stages of the pipeline usually have different amounts of complexity, and it would be incorrect to tune all of them for the same value of hardware intensity. In this subsection we derive an optimality criterion for nonuniform pipelines.

Assume that there are N stages in a pipeline which are different in the amount of logic and time slack available. Each stage consists of a single block of logic followed by a latch, both tuned for one value of energy weight w_i and hardware intensity η_i as shown in **Figure 4**. Then, to achieve the optimum in the power–performance

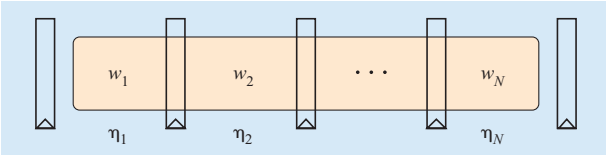

Figure 4

Simple pipeline consisting of N stages with energy weights w_i and hardware intensities η_i.

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: $\eta_1, \cdots \eta_N$, and a single power supply, v.

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

$$E(\eta_1, \cdots \eta_N, v) = \sum_i E_i(v, \eta_i), \quad (10)$$

subject to N constraints

$$D_i(\eta_i, v) = D \qquad i = 1, \cdots, N. \quad (11)$$

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

$$\sum_i w_i \eta_i = \theta(v), \quad (12)$$

where $w_i = (E_i/E)$ are the energy weights of the pipeline stages, $\Sigma_i\, w_i = 1$. 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 (12) together with the cycle time requirement conditions (11) allows 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 (12) is not satisfied, this indicates that power can be reduced without performance loss by changing voltage and retuning circuits. This information can then be used as feedback to reevaluate the choice of the power-supply voltage and the clock-cycle target, and possibly the partitioning of the pipeline into stages.

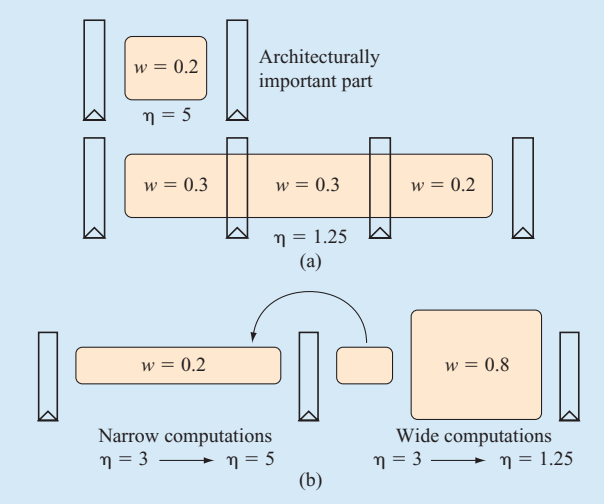

Figure 5

Balancing η in simple pipelines: (a) Two pipelines with different architectural costs of increasing the latency; (b) reduction in the average hardware intensity (η) by moving a block of logic across a stage boundary.

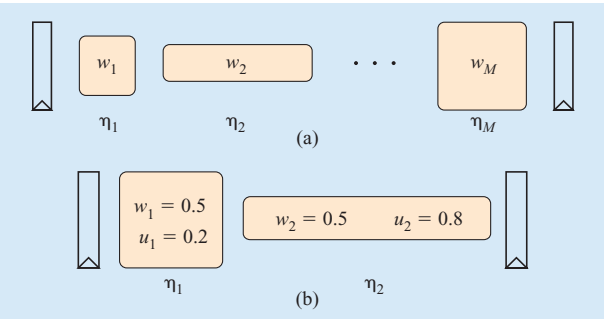

Figure 6

Balancing η in pipelines with composite stages: (a) Composite pipeline stage consisting of m blocks with energy weights w_i and hardware intensities η_i; (b) pipeline stage consisting of two blocks, latch and logic, designed independently.

As an example application of this second relation, consider the system illustrated in **Figure 5(a)**, consisting of two pipelines of one and three stages. Suppose that there is a large architectural cost to lengthening the first pipeline, but that reducing the second pipeline to two stages would have only a negligible impact on the architectural performance. The indicated allocations (again for $E_v = 2$, $D_v = 1$) show how an overall target of $\eta_{ag} = 2$ is obtained by using high-hardware-intensity ($\eta = 5$) circuitry in the architecturally important cycle and balancing that with lower $\eta = 1.25$ in the less critical pipe. Note that we neglect the important effect of changing latch count in this example.

A second example [**Figure 5(b)**] shows how the movement of logic between adjacent cycles can also be used to facilitate power efficiency. Again the overall target is $\eta = 2$, but the initial partitioning has $\eta = 3$ in both cycles. However, perhaps because of large differences in the sizes of logic cones between the cycles, most of the power is burned in the second stage. By moving a portion of the second-stage logic to the first cycle (again ignoring any changes in latch count), and by resizing the two cycles, the η values are made more unbalanced but the overall weighted aggregate is reduced to the target value.

For the higher-level microarchitectural analysis of energy–performance tradeoffs, it is useful to abstract a single aggregate quantity for hardware intensity η_{ag} that represents the whole pipeline, such that

$$\eta_{ag} = -\left.\frac{D\partial E}{E\partial D}\right|_v ,$$

where D is the clock period of the pipeline and E is the average energy dissipated per cycle, $E = \Sigma E_i$. To derive an expression for η_{ag}, notice that increasing the clock cycle time by dD through retuning the circuits in all stages of the pipeline increases the total energy of the pipeline by

$$dE = \sum dE_i = -\sum \frac{E_i}{D_i}\eta_i\, dD,$$

where the summation is performed over all stages of the pipeline. Since (11) is satisfied,

$$\frac{dE}{E} = -\frac{dD}{D}\sum w_i\eta_i,$$

which means that the aggregate hardware intensity for a multistage pipeline is expressed through the hardware intensities of individual stages η_i as

$$\eta_{ag} = \sum_i w_i\eta_i\,. \qquad (13)$$

Then (12) is identical to (9), with $\eta = \eta_{ag}$.

Composite pipeline stage

Pipeline stages usually consist of multiple blocks that are designed and optimized independently. In any conventional pipeline, at least two independent blocks (latches and logic) can be distinguished, and these 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 [**Figure 6(a)**]. Then, there are $M + 1$ independent variables corresponding to the hardware intensities in the M blocks of a pipeline stage: $\eta_1, \cdots \eta_M$, and the single power-supply voltage v. The goal is to find

a relation between $\eta_1, \cdots \eta_M$ and v that leads to the minimum energy

$$E(\eta_1, \cdots \eta_M, v) = \sum_i E_i(v, \eta_i), \tag{14}$$

subject to the total delay requirement D_r; disregarding interblock delay coupling effects, this relation can be written as

$$D(\eta_1, \cdots \eta_M, v) = \sum_i D_i(v, \eta_i) = D_r. \tag{15}$$

Solving this optimization problem, we arrive at M expressions,

$$\frac{w_i}{u_i}\eta_i = \theta(v) \qquad 1 \le i \le M, \tag{16}$$

where u_i is the delay weight of block i, $u_i = (D_i/D)$, and w_i is the corresponding energy weight, $w_i = (E_i/E)$, calculated taking into account the activity factors in clock-gated designs. Note that within a single pipeline stage this implies

$$\frac{w_i}{u_i}\eta_i = \frac{w_j}{u_j}\eta_j \qquad 1 \le i, j \le M. \tag{17}$$

Thus, in a pipeline stage that consists of multiple blocks designed independently, 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. This balance equation is immediately useful in describing the relation between latches and logic. Again, use target values of $E_v = 2$, $D_v = 1$ and suppose that all of the pipeline stages are similar, with latches using 20% of the cycle delay but 50% of the power, as in **Figure 6(b)**. Equation (16) can then be used to determine the optimal hardware intensity for the latches at $\eta_{\text{latch}} = (0.2/0.5)2.0 = 0.8$, and for the logic at $\eta_{\text{logic}} = (0.8/0.5)2.0 = 3.2$. Thus, for these assumptions logic must be optimized much more aggressively than latches.

To derive an expression for the aggregate hardware intensity of a composite pipeline stage, notice that increasing the clock cycle time by $dD = \Sigma\, dD_i$ through retuning the circuits in all blocks of the pipeline stage increases the total energy by

$$dE = \sum dE_i = -\sum \frac{E_i}{D_i}\eta_i\, dD_i,$$

or

$$\frac{dE}{E} = -\sum \frac{w_i}{u_i}\eta_i \frac{dD_i}{D}.$$

If (17) is satisfied, the expression reduces to

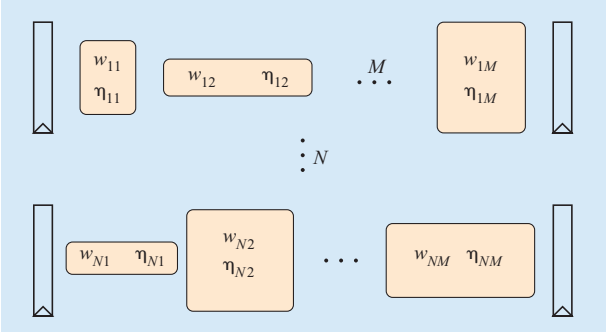

Figure 7

Multistage pipeline with composite stages.

$$\frac{dE}{E} = -\frac{w_k}{u_k}\eta_k \sum \frac{dD_i}{D} = -\frac{w_k}{u_k}\eta_k \frac{dD}{D},$$

where k is *any* sub-block in the pipeline stage. Thus, the aggregate hardware intensity η_{ag} for a composite stage is expressed through the hardware intensities of individual sub-blocks η_i as follows:

$$\eta_{ag} = \frac{w_k}{u_k}\eta_k, \tag{18}$$

where k is any sub-block in the pipeline stage.

Multistage pipeline with composite stages

Now we derive the optimality relation for a more general case, representative of a realistic microprocessor, in which the pipeline consists of N stages and there are at most M sub-blocks in each pipeline stage that are designed independently of one another (**Figure 7**). 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, $1 \le i \le N$, $1 \le j \le M$. The goal is to minimize the total energy in the space on $N \times M + 1$ variables,

$$E(\eta_{11} \cdots \eta_{1M}, \cdots, \eta_{N1} \cdots \eta_{NM}, v) = \sum_{ij} E_{ij}(v, \eta_{ij}), \tag{19}$$

subject to the N constraints

$$\sum_j D_{ij}(v, \eta_{ij}) = D_r \qquad 1 \le i \le N. \tag{20}$$

Solving this problem, we arrive at $N(M-1)$ relations, $M-1$ relations for every pipeline stage i, which are similar to (17):

$$\frac{w_{ij}}{u_{ij}}\eta_{ij} = \frac{w_{ik}}{u_{ik}}\eta_{ik} \qquad 1 \le j, k \le M, \tag{21}$$

and one expression similar to (12):

$$\sum_{i=1}^{N} \frac{w_{ik}}{u_{ik}} \eta_{ik} = \theta(v), \tag{22}$$

where index ik refers to *any* sub-block k within pipeline stage i, u_{ij} is the delay weight of sub-block j in pipeline stage j, $u_{ij} = (D_{ij}/D)$, and w_{ij} is the corresponding energy weight, $w_{ij} = (E_{ij}/E)$, calculated taking into account the activity factors.

To derive an expression for the aggregate hardware intensity of a multistage pipeline with composite stages, notice that increasing the clock cycle time by $dD = \Sigma\, dD_i$ through retuning circuits in all pipeline stages increases the total energy by

$$dE = \sum_{ij} dE_{ij} = -\sum_{ij} \frac{E_{ij}}{D_{ij}} \eta_{ij}\, dD_{ij},$$

or

$$\frac{dE}{E} = -\sum_{ij} \frac{w_{ij}}{u_{ij}} \eta_{ij} \frac{dD_{ij}}{D}.$$

If (21) is satisfied, the expression reduces to

$$\frac{dE}{E} = -\sum_{i} \frac{w_{ik}}{u_{ik}} \eta_{ik} \sum_{j} \frac{dD_j}{D} = -\frac{dD}{D} \sum_{i} \frac{w_{ik}}{u_{ik}} \eta_{ik},$$

where index ik refers to *any* sub-block k within pipeline stage i. Thus, the aggregate hardware intensity η_{ag} for a pipeline with composite stages is expressed through the hardware intensities of individual sub-blocks η_{ij} as

$$\eta_{ag} = \sum_{i=1}^{N} \frac{w_{ik}}{u_{ik}} \eta_{ik}, \tag{23}$$

where index ik refers to any sub-block k within pipeline stage i. Notice that the optimality relation (22) is equivalent to (9), with $\eta = \eta_{ag}$.

Using the expression for the aggregate hardware intensity within pipeline stages (18), relation (22) can be rewritten as

$$\sum_{i=1}^{N} w_i \eta_{ag\,i} = \theta(v), \tag{24}$$

where w_i is the total energy weight of pipeline stage i and $\eta_{ag\,i}$ is the aggregate hardware intensity in pipeline stage i.

Relation to the architectural metric

So far the paper has focused on balancing performance and power at the circuit level. It was shown in [15] that the concept of hardware intensity is closely related to the architectural energy-efficiency metric. To achieve the energy-optimal design in the global architecture-circuit space, architectural choices must be balanced with circuit-level decisions. We next present a methodology that allows architects to optimize the architecture in the global energy-performance space by balancing the architectural complexity with the aggressiveness of the design at the implementation level.

To derive the architectural energy-efficiency criterion, we introduce a discrete variable ξ that represents the architectural complexity of a processor [11, 16], and we express the average power W[1] and performance P of a processor as functions of three variables: architectural complexity ξ, power-supply voltage v, and aggregate hardware intensity η,[2] as follows:

$$P(\xi, \eta, v) = \frac{f(\xi, \eta, v) I(\xi)}{N(\xi)}; \tag{25}$$

$$W(\xi, \eta, v) = f(\xi, \eta, v) I(\xi) E(\xi, \eta, v), \tag{26}$$

where I is the average number of instructions executed per cycle (IPC), which is a measure of the architectural speed, N is the dynamic instruction count, f is the maximum clock frequency, and E is the average energy dissipated per executed instruction, measured on the same set of benchmarks as the IPC. The architectural characteristics N and I do not depend on η and v, whereas f and E depend on all three design variables.

We pose the optimization problem as a problem of optimizing performance subject to a constant power budget.[3] In discrete terms, for an architectural feature $\Delta\xi$ under evaluation we will find a condition for which $\Delta P > 0$, assuming that the power-supply voltage v and the aggregate hardware intensity η are adjusted accordingly to satisfy the constraint $\Delta W = 0$.

To derive the criterion, we make an assumption that for every architectural configuration the processor pipeline is tuned according to the optimal balance between the aggregate hardware intensity and the power supply [Equations (9) and (23)], $\eta = \theta(v)$. Then, disregarding second-order terms, the increment in performance and the constraint of fixed power can be expressed as follows:

$$\frac{\Delta P}{\Delta \xi} = \left.\frac{\Delta P}{\Delta \xi}\right|_{\text{fixed } \eta v} + \left(\frac{\partial P}{\partial \eta}\frac{d\theta}{dv} + \frac{\partial P}{\partial v}\right) \left.\frac{\Delta v}{\Delta \xi}\right|_{\text{fixed } W}; \tag{27}$$

$$\left.\frac{\Delta W}{\Delta \xi}\right|_{\text{fixed } \eta v} + \left(\frac{\partial W}{\partial \eta}\frac{d\theta}{dv} + \frac{\partial W}{\partial v}\right) \left.\frac{\Delta v}{\Delta \xi}\right|_{\text{fixed } W} = 0. \tag{28}$$

[1] For designs in which the worst-case power is the main criterion or designs in which clock gating is not implemented, use should be made of a different form of expression for power which leads to a similar expression for the energy-efficiency criterion [11, 14].
[2] For compactness, the subscript "$_{ag}$" of η has been omitted in most of the formulas in this section.
[3] It was shown in [11] that the reciprocal problem of minimizing power subject to a constant performance requirement leads to the same result.

Substituting expressions (25) and (26) into these formulas and using notation from (7), we rewrite (27) and (28) as

$$\frac{1}{P}\frac{\Delta P}{\Delta\xi} = \frac{1}{f}\frac{\Delta f}{\Delta\xi}\bigg|_{\text{fixed }\eta v} + \frac{\Delta I}{I\Delta\xi} - \frac{\Delta N}{N\Delta\xi} + \left(\frac{D_v}{v} - \frac{1}{D}\frac{\partial D}{\partial\eta}\frac{d\theta}{dv}\right)\frac{\Delta v}{\Delta\xi}\bigg|_{\text{fixed } W} \tag{29}$$

and

$$\frac{1}{f}\frac{\Delta f}{\Delta\xi}\bigg|_{\text{fixed }\eta v} + \frac{\Delta I}{I\Delta\xi} + \frac{1}{E}\frac{\Delta E}{\Delta\xi}\bigg|_{\text{fixed }\eta v} + \left(\frac{D_v + E_v}{v} + \left(\frac{1}{E}\frac{\partial E}{\partial\eta} - \frac{1}{D}\frac{\partial D}{\partial\eta}\right)\frac{d\theta}{dv}\right)\frac{\Delta v}{\Delta\xi}\bigg|_{\text{fixed } W} = 0. \tag{30}$$

Using the definitions of the voltage intensity θ in (7) and aggregate hardware intensity η in (5), and the assumption about the optimal power–performance balance in the pipeline, $\eta = \theta$, we rewrite the constraint of fixed power (30) as follows:

$$\frac{1}{f}\frac{\Delta f}{\Delta\xi}\bigg|_{\text{fixed }\eta v} + \frac{\Delta I}{I\Delta\xi} + \frac{1}{E}\frac{\Delta E}{\Delta\xi}\bigg|_{\text{fixed }\eta v} + (\eta+1)\left(\frac{D_v}{v} - \frac{1}{D}\frac{\partial D}{\partial\eta}\frac{d\theta}{dv}\right)\frac{\Delta v}{\Delta\xi}\bigg|_{\text{fixed } W} = 0. \tag{31}$$

When expressions (29) and (31) are combined, the condition of an increase in performance $\Delta P > 0$ subject to the constant power constraint $\Delta W = 0$ leads to the following formula:

$$\eta\frac{1}{I}\frac{\Delta I}{\Delta\xi} - (\eta+1)\frac{1}{N}\frac{\Delta N}{\Delta\xi} > -\eta\frac{1}{f}\frac{\Delta f}{\Delta\xi}\bigg|_{\text{fixed }\eta v} + \frac{1}{E}\frac{\Delta E}{\Delta\xi}\bigg|_{\text{fixed }\eta v}. \tag{32}$$

In this formula $(\Delta f/f\Delta\xi)$, $(\Delta I/I\Delta\xi)$, $(\Delta E/E\Delta\xi)$, and $(\Delta N/N\Delta\xi)$ are relative increments in the processor frequency, architectural speed, average energy per instruction, and dynamic instruction count arising from a modification at the architectural or microarchitectural level, evaluated for a *fixed* hardware intensity η_{ag} and power supply v. Thus, all deltas in (32) have the meaning of partial derivatives with respect to the architectural complexity.

The terms $(\Delta I/I\Delta\xi)$ and $(\Delta N/N\Delta\xi)$ in (32) can be measured by running the benchmark suite on an architectural simulator. Next we present a methodology for estimating the two remaining terms, $(\Delta f/f\Delta\xi)$ and $(\Delta E/E\Delta\xi)$, and derive a new form of the energy-efficiency criterion that does not require estimating Δf.

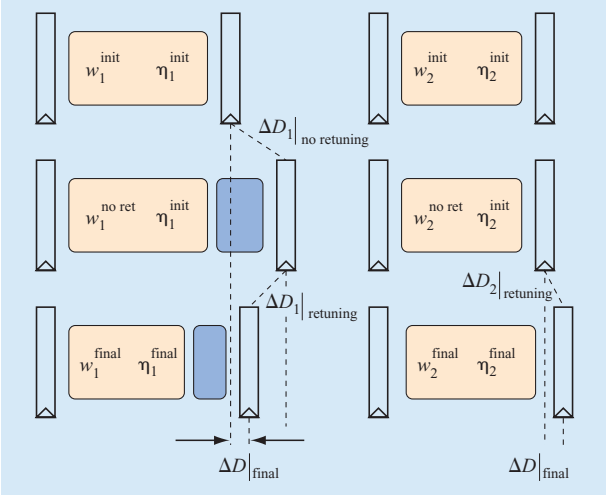

Figure 8

Retuning the pipeline after an architectural modification.

The key assumption in deriving the energy-efficiency criterion (32) was the assumption about the optimal tuning of circuits in every pipeline stage (21) and (22) for every architectural alternative, such that the aggregate hardware intensity of the processor η_{ag} (23) is unchanged between designs implementing architectural alternatives. This assumption imposes special rules on the calculation of $(\Delta f/f)$ and $(\Delta E/E)$; in particular, these relative increments must be calculated assuming that the processor pipeline is reoptimized after every modification to the microarchitecture to satisfy (21) and (22).

Suppose that an architectural feature under evaluation introduces an additional complexity into several (or all) stages of the pipeline, leading to increments $\Delta D_i|_{\text{no retuning}}$ in critical path delays in the corresponding pipeline changes, assuming that no retuning is done to recover the clock frequency. Suppose that the corresponding increments in average energies are $\Delta E_i|_{\text{no retuning}}$. The increments $\Delta D_i|_{\text{no retuning}}$ and $\Delta E_i|_{\text{no retuning}}$ should be evaluated consistently with the initial hardware intensities of the corresponding stages. For example, logic added to stage i should be tuned (or assumed to be tuned) according to Equation (17). Then, after the logic is added, the aggregate hardware intensity (18) in pipeline stage i will not change. The delay and energy increments may be either positive or negative, and in those pipeline stages that are unaffected by the architectural modification, the delay and energy increments are zero, $\Delta D_i|_{\text{no retuning}} = 0$, $\Delta E_i|_{\text{no retuning}} = 0$, as shown in **Figure 8**.

Circuit designers usually have no difficulties estimating the "nonretuned" increments in delay and energy. For example, adding an execution bypass in the 10FO4

pipeline results in increments in the critical path delay and average energy of the execution stage of the pipeline which are approximately

$$\left.\frac{\Delta D_{\mathrm{EX}}}{D}\right|_{\text{no retuning}} = 0.2 \text{ and } \left.\frac{\Delta E_{\mathrm{EX}}}{E_{\mathrm{EX}}}\right|_{\text{no retuning}} = 0.02,$$

whereas adding an extra read port to a multiported register file may result in

$$\left.\frac{\Delta D_{\mathrm{RF}}}{D}\right|_{\text{no retuning}} = 0.1 \text{ and } \left.\frac{\Delta E_{\mathrm{RF}}}{E_{\mathrm{RF}}}\right|_{\text{no retuning}} = 0.2,$$

with no impact on other stages of the pipeline.

To recover the clock frequency, circuits in those stages of the pipeline that are negatively affected by the architectural modification must be tuned up for a higher hardware intensity. To restore the energy optimal balance in the pipeline (24), circuits in all remaining stages must be tuned down for a lower hardware intensity, so that

$$\Delta\eta_{\mathrm{ag}} = \sum_i \eta_i \Delta w_i + \sum_i w_i \Delta\eta_i = 0, \tag{33}$$

where $\Delta\eta_i$ is the increment in the aggregate hardware intensity in stage i as a result of retuning ($\Delta\eta_i = \eta_i^{\text{final}} - \eta_i^{\text{initial}}$), as illustrated in Figure 8, whereas Δw_i is the net increment in the corresponding energy weight as a result of both adding hardware and subsequent retuning [$\Delta w_i = (\Delta E_i/E) - w_i(\Delta E/E)$].

We designate by $\Delta D_i|_{\text{retuning}}$ and $\Delta E_i|_{\text{retuning}}$ the increments in delay and energy in the pipeline stage i as a result of retuning the processor, whereas by $\Delta D_i = \Delta D$ and ΔE_i we designate the net increment in delay and energy in pipeline stage i as a result of *both* modifying the function and subsequent retuning:

$$\Delta D = \Delta D_i|_{\text{no retuning}} + \Delta D_i|_{\text{retuning}}; \tag{34}$$

$$\Delta E_i = \Delta E_i|_{\text{no retuning}} + \Delta E_i|_{\text{retuning}}. \tag{35}$$

Thus, the net delay and energy increments in every pipeline stage consist of increments due to a change in the functionality resulting from a microarchitectural modification, and additional increments as a result of retuning the circuits. The net delay increment ΔD does not require any index because all pipeline stages are assumed to have the same delay before and after the retuning, $D_i = D$. The relative increment in the maximum clock frequency is related to ΔD as

$$\left.\frac{\Delta f}{f}\right|_{\text{fixed } \eta v} = -\frac{\Delta D}{D}. \tag{36}$$

Assuming small changes in hardware intensities in all pipeline stages and neglecting second-order terms, the increments in energies $\Delta E_i|_{\text{retuning}}$ as a result of the retuning can be expressed through the corresponding increments in delays $\Delta D_i|_{\text{retuning}}$ as follows:

$$\Delta E_i|_{\text{retuning}} = -\eta_i \frac{E_i}{D} \Delta D_i|_{\text{retuning}}. \tag{37}$$

By using (34) and (35), the final increments in energies can be expressed as

$$\Delta E_i = \Delta E_i|_{\text{no retuning}} - \eta_i \frac{E_i}{D} (\Delta D - \Delta D_i|_{\text{no retuning}}). \tag{38}$$

The total increment in energy of the whole pipeline, $\Delta E = \Sigma\, \Delta E_i$, is calculated by summing expressions (38) over all pipeline stages and taking advantage of (13) and (36):

$$\left.\frac{\Delta E}{E}\right|_{\text{fixed } \eta v} = \sum \left.\frac{\Delta E_i}{E}\right|_{\text{no retuning}} + \sum \eta_i w_i \left.\frac{\Delta D_i}{D}\right|_{\text{no retuning}} + \eta_{\mathrm{ag}} \left.\frac{\Delta f}{f}\right|_{\text{fixed } \eta v}. \tag{39}$$

Substituting this expression into the derived energy-efficiency criterion (32), we notice that the term $(\Delta f/f)$ cancels out, since in both expressions it has the same meaning of a partial derivative with respect to architectural complexity ξ. Then, dropping $\Delta\xi$ in the denominators of all terms, we arrive at the form of the energy-efficiency criterion that does not require estimating the increment in frequency:

$$\eta \frac{\Delta I}{I} - (\eta + 1) \frac{\Delta N}{N} > \left.\frac{\Delta E}{E}\right|_{\text{no retuning}} + \sum_i \eta_i w_i \left.\frac{\Delta D_i}{D}\right|_{\text{no retuning}}, \tag{40}$$

where

$$\left.\frac{\Delta E}{E}\right|_{\text{no retuning}} = \sum \left.\frac{\Delta E_i}{E}\right|_{\text{no retuning}}$$

is the total increase in average energy dissipated per instruction, assuming that no retuning is done, summation being done over all stages in the pipeline affected by the architectural modification.

Expression (40) is a more convenient form of the energy-efficiency criterion than (32). According to (40), in order to evaluate the energy efficiency of some architectural feature, the architects must supply the relative gain (or loss) in the architectural performance ($\Delta I/I$) and relative change in the dynamic instruction count ($\Delta N/N$) that result from this feature. These estimates can be obtained by running an architectural simulator, or timer, such as Turandot [17, 18]. The second term, ΔN, is nonzero if changes to the instruction set architecture (ISA) are considered, or compiler

optimizations are analyzed for energy efficiency. It may also be nonzero if microarchitectural changes that affect the average number of instructions executed from mispredicted paths are considered in a speculative issue processor.

Then the architect needs to consult circuit designers to estimate the impact of the architectural feature under consideration on the average energy dissipated per instruction and the critical path delay through every stage of the pipeline affected by this architectural feature. A significant advantage of the derived formula is that in estimating the relative changes in energy and critical path delays, the circuit designer does not need to worry about retuning the circuits to recover the frequency, or reducing the positive timing slack to save some power. Then, the relative increments in critical path delays are summed and multiplied by the appropriate energy weights and hardware intensities. The higher the energy weight w_i and the hardware intensity η_i of a part of the pipeline i affected by the architectural feature, the higher the weight of the increase in the critical path delay through this part of the pipeline.

Expression (40) is then evaluated. If the inequality holds, the architectural feature under evaluation is energy-efficient; that is, after adopting it, the processor will deliver higher net performance at the same power budget, after appropriate retuning and, possibly, adjustment in the power-supply voltage are done to meet the power budget.

The energy weights w_i in (40) are typically available as part of power budgeting at the early stages of the definition of the processor pipeline. The only additional data required in order to use the energy-efficiency criterion are hardware intensities η_i in all blocks of the processor. Those quantities can be measured by static tuning tools such as EinsTuner [12] on the basis of simulations of previous designs, or set as targets at early planning of the microarchitecture, in the same way the power targets are budgeted.

We refer the reader to [16] for realistic examples of using the derived energy-efficiency criterion, and a graphical interpretation of the iterative process of refining the architecture using the criterion.

Conclusions and future work

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 straightforwardly evaluated for a tuned circuit. The relations on η also provide guidance for choosing appropriate supply voltage. Overall attention to these concepts ensures a more power-efficient design.

We plan to extend this analysis in a number of ways. In the examples considered so far, the delay of a block has been considered as a single value. In practice, however, macros typically display irregular boundary conditions. There may be critical paths of greatly different lengths in a single macro. Fortunately, it is also possible to compute sensitivities of macro power with respect to various timing assertions and derive conditions for power optimality based on them. This work is underway. We have mentioned briefly the fact that moving latch boundaries generally results in changing the number of latches, which is significant because of the large contribution of latches to overall power. Such considerations generally push the analysis into the realm of microarchitecture, and collaboration is underway to extend the analysis in this way [11, 19]. For planning purposes, it is important to understand the values for hardware intensity implied by real designs. Existing circuits are being studied to understand the hardware intensities in practice.

The existing analysis asserts relatively rigid cycle boundaries, while in practice some degree of transparency or cycle stealing may be possible. There may be mixtures of clock domains of differing frequencies. Techniques for enhanced power efficiency such as multiple thresholds, oxides, or supplies should be considered. In addition to facilitating communication between circuits and microarchitecture, hardware intensity could also be used between circuits and technology. Decisions regarding FET parameters could be aided by considering the power–performance characteristics of such devices. The supply voltage for power limits may differ from the supply voltage used for nominal timing, and this can affect the scaling. As is clear from the above discussion, these ideas provoke a number of interesting extensions addressing practical issues.

Appendix

For completeness, we derive below a closed-form expression for the term

$$\left.\frac{\Delta f}{f}\right|_{\text{fixed } \eta v},$$

which appears in form (32) of the energy-efficiency criterion and relations for energy increments. The goal is to express this term through the easily measurable quantities

$$\left.\frac{\Delta D_i}{D}\right|_{\text{no retuning}}.$$

To derive the formula, we use N expressions (38) and the condition of the unchanged aggregate hardware intensity (33) which, using (12), can be expressed as

$$\sum_i \Delta E_i(\eta_i - \theta) + \sum_i E_i \Delta\eta_i = 0. \tag{41}$$

To close the system of equations, we express the increments in hardware intensities $\Delta\eta_i$, resulting from the retuning in expression (41), through the corresponding increments in delays as

$$\Delta\eta_i = \frac{\partial\eta_i}{\partial D_i}\Delta D_i\Big|_{\text{retuning}}.$$

The partial derivatives $(\partial\eta_i/\partial D_i)$ can be expressed through the second-order derivative of energy with respect to delay as

$$\frac{\partial\eta_i}{\partial D_i} = \frac{\partial}{\partial D_i}\left(-\frac{D_i}{E_i}\frac{\partial E_i}{\partial D_i}\right) = \frac{1}{D}\left(\eta_i + \eta_i^2 - \frac{D_i^2}{E_i}\frac{\partial^2 E_i}{\partial D_i^2}\right).$$

Using (34), the increments in hardware intensities $\Delta\eta_i$ can be expressed as

$$\Delta\eta_i = \left(\eta_i + \eta_i^2 - \frac{D_i^2}{E_i}\frac{\partial^2 E_i}{\partial D_i^2}\right)\left(\frac{\Delta D}{D} - \frac{\Delta D_i}{D}\Big|_{\text{no retuning}}\right). \tag{42}$$

Substituting N relations (38) for ΔE_i and N relations (42) for $\Delta\eta_i$ into Equation (41), we can express the final increments in the clock period ΔD and frequency Δf through "nonretuned" increments in energy and delay, $\Delta E_i|_{\text{no retuning}}$ and $\Delta D_i|_{\text{no retuning}}$ in pipeline stages $i = 1, \cdots N$ as

$$\frac{\Delta f}{f}\Big|_{\text{fixed }\eta v} = -\frac{\Delta D}{D}\Big|_{\text{fixed }\eta v}; \tag{43}$$

$$\frac{\Delta D}{D}\Big|_{\text{fixed }\eta v} = \frac{1}{\sum b_i}\sum_i b_i \frac{\Delta D_i}{D}\Big|_{\text{no retuning}} + \frac{1}{\sum b_i}\sum_i c_i \frac{\Delta E_i}{E_i}\Big|_{\text{no retuning}}, \tag{44}$$

where b_i and c_i are weighting factors expressed as

$$b_i = -w_i\eta_i(\theta + 1) + w_i\frac{D^2}{E_i}\frac{\partial^2 E_i}{\partial D_i^2}; \tag{45}$$

$$c_i = w_i(\eta_i - \theta), \tag{46}$$

where w_i are the energy weights of pipeline stages, as defined before. Thus, the relative change in the clock period is a weighted average of relative increments in critical path delays, calculated over all pipeline stages. The only new quantity in these expressions is the normalized second-order derivative of energy with respect to delay, which, like hardware intensity, can be obtained from energy–delay tradeoff curves.

It can be shown under very general assumptions about the shape of the energy–delay tradeoff curve, such as a uniform growth of hardware intensity as a function of the critical path delay $\eta = \eta(D)$, that the normalized second-order derivative of energy with respect to delay grows at least as a square of the hardware intensity,

$$\frac{D^2}{E_i}\frac{\partial^2 E_i}{\partial D_i^2} > \eta_i(\eta_i + 1),$$

and, as a consequence, $\Sigma\, b_i > 0$. Then, the higher the energy weight w_i and hardware intensity η_i of pipeline stage i, the higher the weight of the corresponding increase in the critical path delay b_i in the weighted average (43).

Changes in energies of pipeline stages due to architectural modifications also affect the increment in the clock period. According to (46), the higher the energy weight w_i and hardware intensity η_i of pipeline stage i, the higher the weight of the increase in the energy c_i in (43). Notice that $\Sigma\, c_i = 0$.

For architectural changes that uniformly affect all pipeline stages,

$$\Delta E_i|_{\text{no retuning}} = \Delta E_j|_{\text{no retuning}}$$

and

$$\Delta D_i|_{\text{no retuning}} = \Delta D_j|_{\text{no retuning}},$$

formula (43) is reduced to intuitive expressions

$$\frac{\Delta f}{f} = -\frac{\Delta D_i}{D}\Big|_{\text{no retuning}}.$$

The same simplification applies to uniformly tuned pipelines, with $b_i = b_j$, $\eta_i = \eta_j$, and $w_i = w_j$.

Acknowledgment

The authors would like to thank P. Bose for valuable discussions and J. Moreno and K. Warren for management support.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Synopsys, Inc.

References

1. J. Burr and A. Peterson, "Energy Considerations in Multichip Module-Based Multiprocessors," *Proceedings of the International Conference on Computer Design* (*ICCD*), 1991, pp. 593–600.
2. A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-Power CMOS Digital Design," *IEEE J. Solid-State Circuits* **27,** No. 4, 473–484 (April 1992).
3. T. Burd and R. Brodersen, "Energy Efficient CMOS Microprocessor Design," *Proceedings of the 28th Annual*

Hawaii International Conference on System Sciences, 1995, pp. 288–297.
4. R. Gonzalez and M. Horowitz, "Energy Dissipation in General Purpose Microprocessors," *IEEE J. Solid-State Circuits* **31,** No. 9, 1277–1283 (September 1996).
5. R. Gonzalez, B. Gordon, and M. Horowitz, "Supply and Threshold Voltage Scaling for Low Power CMOS," *IEEE J. Solid-State Circuits* **32,** No. 8, 1210–1216 (August 1997).
6. M. Stan, "Low-Power CMOS with Subvolt Supply Voltages," *IEEE Trans. VLSI Syst.* **9,** No. 2, 394–400 (April 2001).
7. P. Penzes and A. Martin, "Energy-Delay Efficiency of VLSI Computations," *Proceedings of the Great Lakes Symposium on VLSI*, April 2002, pp. 104–107.
8. K. Nowka, P. Hofstee, and G. Carpenter, "Accurate Power Efficiency Metrics and Their Application to Voltage Scalable CMOS VLSI Design," *IEEE Trans. VLSI Syst*., 2003, in press.
9. V. Zyuban and P. Kogge, "Optimization of High-Performance Superscalar Architectures for Energy Efficiency," *Proceedings of the IEEE Symposium on Low Power Electronics and Design*, August 2000, pp. 84–89.
10. V. Zyuban and D. Meltzer, "Clocking Strategies and Scannable Latches for Low Power Applications," *Proceedings of the IEEE Symposium on Low Power Electronics and Design*, August 2001, pp. 346–351.
11. V. Zyuban, "Unified Architecture Level Energy-Efficiency Metric," *Proceedings of the Great Lakes Symposium on VLSI*, April 2002, pp. 24–29.
12. A. R. Conn, I. M. Elfadel, W. W. Molzen, Jr., P. R. O'Brien, P. N. Strenski, C. Visweswariah, and C. B. Whan, "Gradient-Based Optimization of Custom Circuits Using a Static-Timing Formulation," *Proceedings of the Design Automation Conference*, June 1999, pp. 452–459.
13. Xiao Yan Yu, V. G. Oklobdzija, and W. W. Walker, "Application of Logical Effort on Design of Arithmetic Blocks," *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, 2001, pp. 872–874.
14. J. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits," *IEEE J. Solid-State Circuits* **19,** No. 4, 468–473 (August 1984).
15. V. Zyuban and P. Strenski, "Unified Methodology for Resolving Power–Performance Tradeoffs at the Microarchitectural and Circuit Levels," *Proceedings of the International Symposium on Low Power Electronics and Design*, August 2002, pp. 166–171.
16. J. H. Moreno, V. Zyuban, U. Shvadron, F. D. Neeser, J. H. Derby, M. S. Ware, K. Kailas, A. Zaks, A. Geva, S. Ben-David, S. W. Asaad, T. W. Fox, D. Littrell, M. Biberstein, D. Naishlos, and H. Hunter, "An Innovative Low-Power High-Performance Programmable Signal Processor for Digital Communications," *IBM J. Res. & Dev.* **47,** No. 2/3, 299–326 (2003).
17. M. Moudgill, P. Bose, and J. H. Moreno, "Validation of Turandot, a Fast Processor Model for Microarchitecture Exploration," *Proceedings of the IEEE International Performance, Computing, and Communications Conference* (*IPCCC*), February 1999, pp. 451–457.
18. M. Moudgill, J. D. Wellman, and J. H. Moreno, "Environment for PowerPC Microarchitecture Exploration," *IEEE Micro* **19,** No. 3, 9–14 (May/June 1999).
19. V. Srinivasan, D. Brooks, M. Gschwind, P. Bose, V. Zyuban, P. N. Strenski, and P. G. Emma, "Optimizing Pipelines for Power and Performance," *Proceedings of the 35th Annual International Symposium on Microarchitecture*, November 2002, pp. 333–344.

Received October 9, 2002; accepted for publication February 14, 2003

Victor Zyuban *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (zyuban@us.ibm.com).* Dr. Zyuban received his B.S. and M.S. degrees from the Moscow Institute of Physics and Technology in 1993 and 1995, respectively, and his Ph.D. degree in computer science and engineering from the University of Notre Dame in 2000. From 1995 to 1996, he worked in the Moscow Center for SPARC Technologies. He is currently a Research Staff Member at the IBM Thomas J. Watson Research Center. From 2000 to 2003, Dr. Zyuban was working on a low-power DSP research project, in which he was involved in ISA definition, microarchitecture, and physical design, and was leading the development of a semicustom eLite core test chip. Currently Dr. Zyuban is working on a high-performance general-purpose microprocessor core. His research interests include high-frequency, low-power circuitry, microarchitecture, and methodologies for low-power design.

Philip N. Strenski *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (strensk@us.ibm.com).* Dr. Strenski received his A.B. degree in mathematics in 1979 from Washington University, and his Ph.D. degree in physics in 1985 from Stanford University. That same year he joined IBM, engaging in research in design automation, including simulated annealing, parametric extraction, and circuit tuning. In the early 1990s he switched to microprocessor circuit design, over the years contributing to CPU designs in the mainframe, midrange, and workstation businesses. Dr. Strenski is currently involved in the design of the "Cell" system, a collaboration among Sony, Toshiba, and IBM; he is also actively involved in issues of power–performance efficiency.