

# Quantifying and Enhancing Power Awareness of VLSI Systems

Manish Bhardwaj, Rex Min, and Anantha P. Chandrakasan, *Member, IEEE*

**Abstract**—An increasingly important figure-of-merit of a VLSI system is “power awareness,” which is its ability to scale power consumption in response to changing operating conditions. These changes might be brought about by the time-varying nature of inputs, desired output quality, or just environmental conditions. Regardless of whether they were engineered for being power aware, systems display variations in power consumption as conditions change. This implies, by the definition above, that all systems are naturally power aware to some extent. However, one would expect that some systems are “more” power aware than others. Equivalently, we should be able to re-engineer systems to increase their power awareness. In this paper, we attempt to quantitatively define power awareness and how such awareness can be enhanced using a *systematic* technique. We illustrate this technique by applying it to VLSI systems at several levels of the system hierarchy—multipliers, register files, digital filters, dynamic voltage-scaled processors, and data-gathering wireless networks. It is seen that, as a result, the power awareness of these preceding systems can be significantly enhanced leading to increases in battery lifetimes in the range of 60–200%.

**Index Terms**—Energy aware, low energy, low power, metrics, power aware, scalable.

## I. INTRODUCTION

**L**OW-POWER system design, assuming a worst-case power dissipation scenario, is being supplanted by a more comprehensive philosophy variously termed power-aware, energy-aware, or energy-quality scalable design [1]. The basic idea behind these essentially identical approaches is to allow the system power to scale with changing conditions and quality requirements.

There are two main views to motivating power-aware design and its emergence as an important paradigm. The first view is to explain the importance of power awareness as a consequence of the increasing emphasis on making systems more *scalable*. In this context, making a system scalable refers to enabling the *user* to tradeoff system performance parameters as opposed to hard-wiring them. Scalability is an important figure-of-merit since it allows the end user to implement operational policy, which often varies significantly over the lifetime of the system.

Manuscript received November 12, 2000; revised June 15, 2001. This work was supported by the Defense Advanced Research Projects Agency (DARPA) Power Aware Computing/Communication Program and the Air Force Research Laboratory, Air Force Materiel Command, USAF, under Agreement F30602-00-2-0551. The work of M. Bhardwaj was supported by an IBM Research Fellowship. The work of R. Min was supported by an NDSEG Fellowship.

The authors are with the Department of EECS, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 USA (e-mail: manishb@mit.edu).

Publisher Item Identifier S 1063-8210(01)08002-7.

For example, consider the user of a portable multimedia terminal. At times, the user might want extremely high performance (say, high video quality) at the cost of reduced battery lifetime. At other times, the opposite might be true, i.e., the user might want bare minimum perceptual quality in return for maximizing battery lifetime. Such tradeoffs can only be optimally realized if the system was designed in a power-aware manner. A related motivation for power awareness is that a well designed system must *gracefully* degrade its quality and performance as the available energy resources are depleted [2]. Continuing our video example, this implies that as the expendable energy decreases, the system should gracefully degrade video quality (seen by the user as increased “blockiness,” for instance) instead of exhibiting a “cliff-like,” all-or-none behavior (perfect video followed by no video) [2], [3].

While the view above argues for power awareness from a user-centric and user-visible perspective, one can also motivate this paradigm in more fundamental, system-oriented terms. With burgeoning system complexity and the accompanying increase in integration, there is more diversity in the operating scenarios than ever before. Hence, design philosophies that assume the system to be in the worst-case operating state most of the time are prone to yield suboptimal results. In other words, even if there is little explicit user intervention, there is an imperative to track operational diversity and scale power consumption accordingly. This naturally leads to the concept of power awareness. For instance, the embedded processor that decodes the video stream in a portable multimedia terminal can display tremendous workload diversity depending on the temporal correlation of the incoming video stream. Hence, even if the *user* does not change quality criteria, the processor must exploit this operational diversity by scaling its power as the workload changes.

Since low energy and low power are intimately linked to power awareness, it is important and instructive to provide a first-cut delineation of these concepts even at this introductory stage. This is to convince the reader that power awareness as a metric and a design driver *does not* devolve to traditional worst-case centric low-power/low-energy design. As preliminary evidence of this, consider the system architect faced with the task of increasing the power awareness of the portable multimedia terminal mentioned above. While the architect can claim that certain engineering techniques reduce worst-case dissipation and/or overall energy consumption of the terminal and so on, these traditional measures still fall short of answering the related but different questions:

How *well* does the terminal scale its power with user or data or environment dictated changes? What prevents it from being

arbitrarily proficient in tracking operational diversity? How can we *quantify the benefits* of such proficiency? How can we *systematically enhance* the system's ability to scale its power? What are the *costs* of achieving such enhancements?

In this paper, we attempt to formally answer these questions. We initiate the process of formally understanding power awareness by using a multiplier as a simple but pedagogic example. This is followed by a more rigorous presentation of these concepts in the form of a metric which is shown to be fundamentally linked to the overall battery lifetime of a system (Section II). Methods to enhance power awareness are discussed in Section III. Section IV demonstrates the efficacy of the proposed metric and enhancement techniques using register files, digital filters, dynamic voltage scaling systems, and data-gathering networks as examples. Section V summarizes the paper.

## II. QUANTIFYING POWER AWARENESS

### A. Preliminaries

In this section, we develop the basic power-awareness formalisms using a simple system—a  $16 \times 16$ -bit array multiplier [4]—as an example. This will allow us to elucidate the essence of our arguments without getting bogged down by detail.

Consider a given system  $H$  that performs a certain set of operations  $F$ , while obeying a set of constraints  $C$ . For the illustrative system,  $H$  would be the given implementation of a  $16 \times 16$  bit array multiplier. While the set  $F$  would ideally contain all  $m$ -bit  $\times$   $n$ -bit multiplications, where  $m, n \in [1, 16]$ , we restrict  $F$  to be set of all  $m$ -bit  $\times$   $m$ -bit multiplications instead. We shall see the value of this restriction in the following discussion. Finally, the constraint might be simply one of fixed latency (i.e.,  $H$  cannot take more than a given time to perform  $F$ ).

Given this information, we ask:

1) *The Power-Awareness Question:* How well does the energy of a system,  $H$ , scale with changing operating scenarios?

Note that we use energy and not power in the statement above because energy allows us to seamlessly include latency constraints later on. Next, observe that our understanding of power awareness can only be as exact as our understanding of “operating scenarios.” As one might expect, these scenarios can be characterized with arbitrarily high detail. For instance, in the case of the multiplier, we can define the scenario by the precision of the current multiplicands or the multiplicands themselves or even the current multiplicands and the *previous* multiplicands, since the power dissipation is a function of those too. In the interests of simplicity, we choose to characterize the set of scenarios  $S$  by the precision of the multiplicands. Normally, this would need a two-tuple since there are two multiplicands. But, by our choice of  $F$ , only one number (the precision of the two multiplicands with identical bit widths) characterizes the scenario. Hence,  $H$  can find itself in one of 16 scenarios. Henceforth, we denote a scenario by  $s$  and the set of 16 scenarios by  $S$ .

Having defined scenarios, we take the first step toward characterizing the power awareness of  $H$  by tracing its energy behavior as it moves from one scenario to another. For a 16-bit multiplier, we would do this by executing a large number of different scenarios and measuring the energy consumed by each

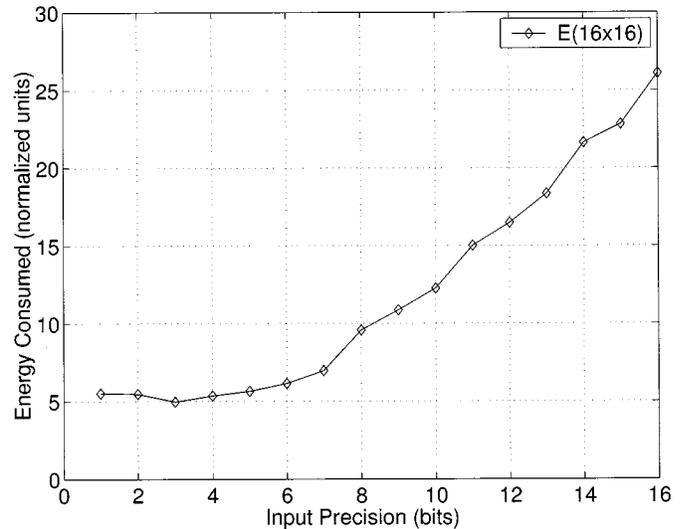


Fig. 1. Multiplier energy as a function of input precision.

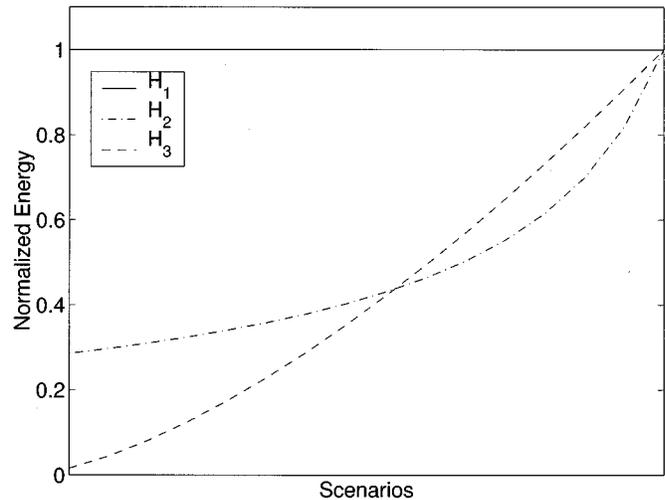


Fig. 2. Energy curves of three different hypothetical systems.

scenario.<sup>1</sup> Henceforth, we call these energy *versus* scenario curves of  $H$  simply as the “energy curves” of  $H$ .

Fig. 1 shows the energy curve of our  $16 \times 16$ -bit array multiplier over 16 scenarios (which represent the precision of the multiplication). Note that the multiplier has a natural degree of power awareness even though it was not explicitly designed for it. This is easy to understand since lower precision vectors lead to less switched capacitance than higher precision ones.

An energy-curve like the one in Fig. 1 is the first step to answering the power-awareness question. However, at this stage, it is difficult to answer the “how well” in the question by looking at a system by itself. Hence, instead of a single energy curve, we look at a few curves together to get a better understanding of the desirable properties of energy curves.

Fig. 2 plots the energy curves of three hypothetical systems  $H_1$ ,  $H_2$ , and  $H_3$  executing a certain, identical set of scenarios. If

<sup>1</sup>All multiplier energy curves that we discuss in this paper were derived by extensive (>1000 vectors) *PowerMill* simulations of multiplier SPICE netlists in a  $0.35\mu\text{m}$  process.

we had to judge the power awareness of these systems from their energy curves, we would intuitively classify  $H_1$  as the system that is the most *unaware* of the executing scenario. Such an undifferentiated energy curve might be expected if, for instance, these systems were implementing multiplication and  $H_1$  was a 32-bit RISC processor (since the energy taken by the other parts would be so great that the actual precision of multiplication would have insignificant impact).  $H_2$ , on the other hand, definitely displays more energy differentiation than  $H_1$  and is intuitively “more scalable.” Furthermore, since the energy of  $H_2$  is strictly less than  $H_1$ , it seems unequivocally better. Similar arguments can be applied while comparing  $H_3$  to  $H_1$  and we conclude that  $H_3$  is more scalable than  $H_1$ . However, these intuitive arguments break down when we try comparing  $H_3$  with  $H_2$ . On the one hand,  $H_3$  displays better scalability than  $H_2$ . On the other, its energy dissipation exceeds that of  $H_2$  over a certain interval. For this reason, at this point in our development, it is unclear whether we should pick  $H_2$  or  $H_3$  as the more power-aware system. To help answer that question, it might help to think of the energy curve of the *most* desirable system, say  $H_{\text{perfect}}$  executing the same operations under the same constraints as the three systems discussed above. In a second step, we could potentially compare the curves of  $H_2$  and  $H_3$  to that of  $H_{\text{perfect}}$  to decide which is more power aware. It helps to state the following.

2) *The Perfectly Power-Aware System (I)*: A system  $H_{\text{perfect}}$  is defined as the most power-aware system iff for every scenario in  $S$ ,  $H_{\text{perfect}}$  consumes only as much energy as its current scenario demands.<sup>2</sup>

It is clear from the above statement that we need to formally capture the concept of “only as much energy as a scenario demands.” To derive this energy for a given scenario, say  $s_1$ , we consider constructing a system  $H_{s_1}$  that is designed to execute this and only this scenario. The reasoning is that we should not hope that a given system  $H$  can ever consume lesser energy in a scenario compared to  $H_{s_1}$ —a *dedicated* system which was specially designed to execute *only* that scenario. We often refer to the  $H_{s_i}$ s as “point” systems because of their focused construction to achieve low energy for a particular scenario (or point) in the energy curve. Hence, in the context of power awareness, the energy consumed by  $H_{s_1}$  is, in a sense, the lower bound on the dissipation of  $H$  while executing scenario  $s_1$ . The following generalizes this statement.

3) *Bounds on Efficiency of Tracking Scenarios*: The energy consumed by a given system  $H$  while executing a scenario  $s_i$  cannot be lower than that consumed by a dedicated system  $H_{s_i}$  constructed to execute only that scenario  $s_i$  as efficiently as possible.

This leads to our next definition of  $H_{\text{perfect}}$ .

4) *The Perfectly Power-Aware System (II)*: The perfect system,  $H_{\text{perfect}}$ , is as energy efficient as  $H_{s_i}$  while executing scenario  $s_i$  ( $\forall s_i \in S$ ).

<sup>2</sup>More formally,  $H_{\text{perfect}}$  is the most power-aware system iff for every scenario in  $S$ ,  $H_{\text{perfect}}$  consumes only as much energy as demanded by its current operation in  $F$  executing in the current scenario under constraints  $C$ . In our multiplier example, we have chosen to construct  $S$  such that it has a one-one correspondence with  $F$  and, hence, it makes sense to talk about the “energy of a scenario” executing on  $H$ .

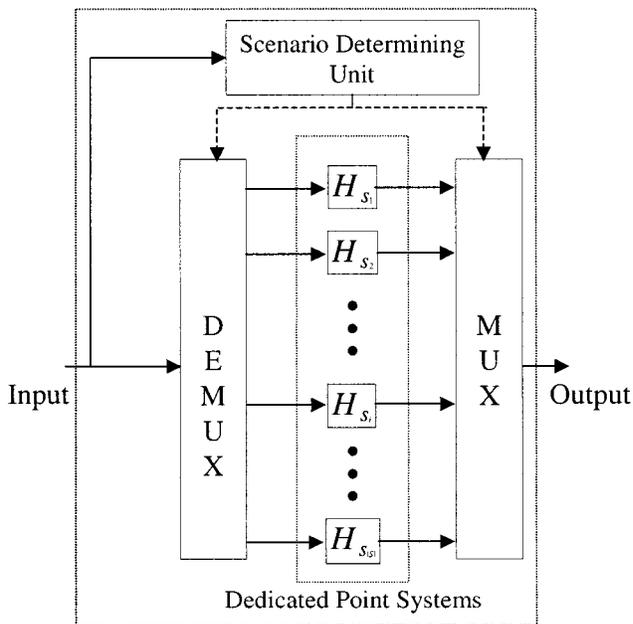


Fig. 3. The perfect system ( $H_{\text{perfect}}$ ) can be viewed as an ensemble of point systems.

We denote the energy curve of the perfect system by  $E_{\text{perfect}}$ . As defined above, the perfect system behaves as if it is composed of dedicated point systems—one for each scenario. When  $H_{\text{perfect}}$  has to execute a scenario  $s_i$ , it routes the scenario to the point system  $H_{s_i}$ . After  $H_{s_i}$  is done processing, the result is routed to the common system output. This abstraction of  $H_{\text{perfect}}$  as an *ensemble of point systems* is illustrated in Fig. 3.

The task of identifying the scenario by looking at the data input is carried out by the *scenario determining* block. Once this block has identified the scenario, it configures the *mux* and *de-mux* blocks such that data is routed to, and results routed from, the point system that corresponds to the current scenario. Note that if the energy costs of identifying the scenario, routing to and from a point system and activating the right point system are zero, then the energy consumption of  $H_{\text{perfect}}$  will indeed be equal to that of  $H_{s_i}$  for every scenario  $s_i$ . Since these costs are never zero in real systems, this implies that  $H_{\text{perfect}}$  is an abstraction and does not correspond to a physically realizable system. Its function is to provide a nontrivial lower bound for the energy-curve.

To construct the  $E_{\text{perfect}}$  curve for our 16-bit multiplier, we emulated the ensemble of points construction outlined above. The point systems in our example were 16 dedicated point multipliers— $1 \times 1$ -bit,  $2 \times 2$ -bit,  $\dots$ ,  $16 \times 16$ -bit—corresponding to  $H_{s_1}$  to  $H_{s_{16}}$ . When a pair of multiplicands with precision  $i$  came by, we diverted them to  $H_{s_i}$  (i.e., the  $i \times i$ -bit multiplier). Since we are deriving  $E_{\text{perfect}}$ , only the energy consumed by the  $H_{s_i}$ s was taken into account. The  $E_{\text{perfect}}$  curve thus derived is plotted in Fig. 4, where the energy-curve of a single  $16 \times 16$  multiplier is repeated for comparison.

Note that  $E_{\text{perfect}}$  scales extremely well with precision since the scenarios are being executed on the best possible point systems that we could construct. Before we indulge in a more detailed comparison of the two curves, it is essential to note that

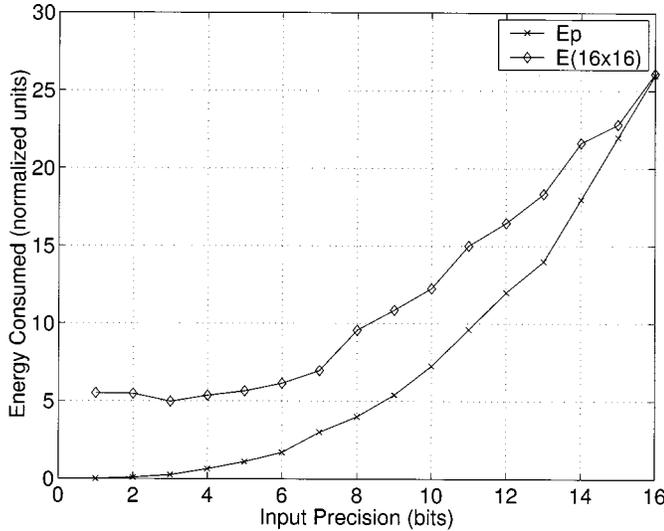


Fig. 4. Comparing the  $16 \times 16$  multiplier curve to the “perfect” curve ( $E_{\text{perfect}}$ ) denoted by  $E_p$  in the plot.

$E_{\text{perfect}}$  really depends on the kind of “point” systems we allow. In the case of the multiplier, we allowed any  $i \times i$ -bit multiplier. The set of point systems we allow is henceforth denoted by  $P$ . This set captures the resources available to engineer a power-aware system. Like the scenario and constraint sets, it can be specified with increasing rigor and detail. This new formalism ( $P$ ) has two key purposes. First, it gives a more fundamental basis to  $E_{\text{perfect}}$ . While it is not possible to talk about the “best possible energy curve,” it is indeed possible to talk about the “best possible energy curve for a specified  $P$ .” Second,  $P$  is also important when we discuss enhancing the power awareness of  $H$ . In that context,  $P$  specifies exactly which building blocks are available to us for such an enhancement.

To quantify how power aware our multiplier is, we plot the scenario efficiency ratio

$$\eta_i = \frac{E(H_{\text{perfect}}, s_i)}{E(H, s_i)} \quad (1)$$

in Fig. 5.<sup>3</sup>

A  $\eta_i$  value of unity indicates that the system under consideration is as power aware as it can be for that scenario. The smaller the  $\eta_i$  value, the worse the system’s awareness of scenario  $s_i$ . In the case of the multiplier, note that  $H$  tracks  $H_{\text{perfect}}$  fairly closely for higher precisions. This is to be expected since a  $16 \times 16$  bit multiplier would be very efficient for scenarios, where the operand precision is close to 16. For lower precision scenarios,  $H$  loses its ability to track as well as  $H_{\text{perfect}}$  and can dissipate up to two orders more energy than  $E_{\text{perfect}}$ . This is a recurring theme in system design. There are energy costs to pay when a single system ( $H$ ) is used over diverse operating conditions and the  $\eta$  curve above quantifies those costs.

### B. Defining Power Awareness

In the preceding section, we quantified the power awareness of a system on a scenario-by-scenario basis using the  $\eta$  func-

<sup>3</sup>The notation  $E(H, s_i)$  denotes the energy consumed by a system  $H$  while executing scenario  $s_i$ .

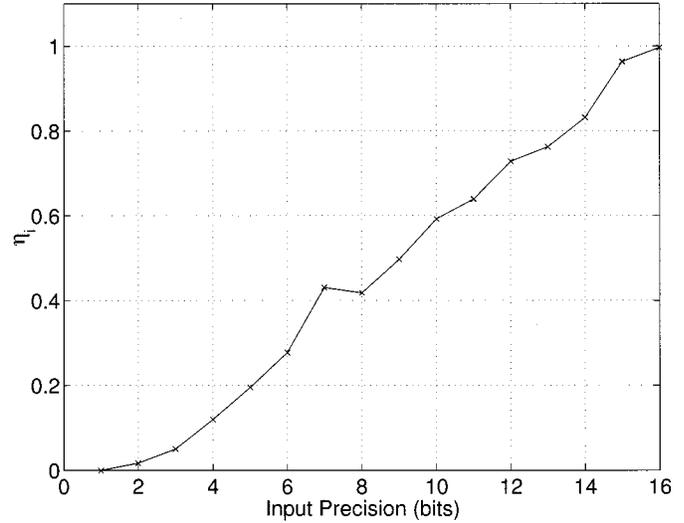


Fig. 5. The scenario efficiency or awareness ( $\eta_i$ ) of a  $16 \times 16$  bit multiplier.

tion. Hence, we have partially answered the power-awareness question posed earlier. The  $\eta$  curve is a partial answer because it still does not help us resolve the other question we posed in the last section—given the energy curves of two systems, can we determine which of the two is more power aware? It is clear that to answer this question, we need to develop a measure of power awareness that distills the *entire*  $\eta$  curve.

Although there are infinitely many possibilities, we will describe those that have a useful system-level meaning and can be practically employed by system architects. A definition that reflects the *average-case* power-aware behavior of the system is its *expected* ability to track scenario changes

$$\phi_1 = \mathbf{E}[\eta] = \frac{\sum_{i=1}^{|S|} \eta_i}{|S|} \quad (2)$$

where  $|S|$  is the cardinality of set  $S$  and  $\mathbf{E}[\cdot]$  is the expectation operator and should not be confused with energy. The physical interpretation of  $\phi_1$  is that if *all scenarios were equally likely*,  $H$  would track the scenario changes with an expected efficiency of  $\phi_1$ . For the  $16 \times 16$  multiplier,  $\phi_1 = 0.501$ .

Clearly, we can refine the definition of  $\phi_1$  to be more realistic if we had a sense of the *likelihood* that the system will reside in a particular scenario rather than just assuming all scenarios to be uniformly likely. For instance, Fig. 6 charts the probability that a multiplier will be in a certain precision scenario when it is filtering a typical speech signal [5].

We call the curve in Fig. 6 a *scenario-distribution* curve (henceforth, we use  $d$  to denote scenario-distributions and  $d_i$  to denote the probability of occurrence of scenario  $s_i$ ). We can now factor in  $d$  to arrive at a more reasonable value of the *expected* power awareness of  $H$

$$\phi_d = \mathbf{E}(\eta) = \sum_{i=1}^{|S|} \eta_i d_i. \quad (3)$$

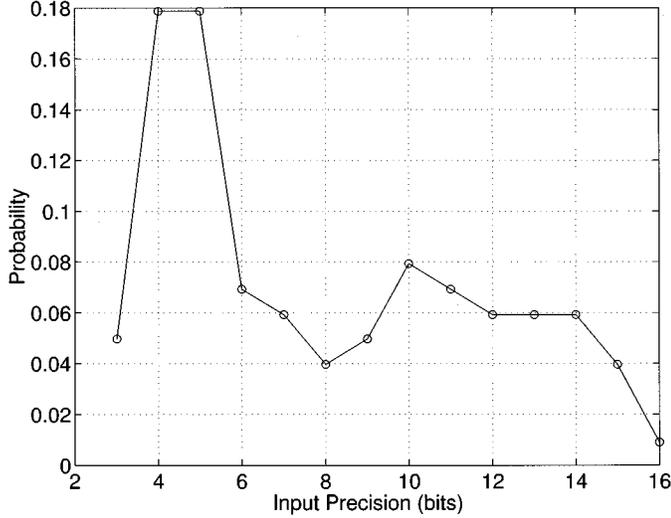


Fig. 6. Typical multiplier usage pattern in speech filtering applications.

For the case of the multiplier for the distribution above, this turns out to be close to 0.42.

While a scenario's frequency of occurrence is a fair indicator of its importance, its not the only one. For instance, a scenario might have a low probability of occurrence, but when it *does occur*, the architect *might want* the system to track the change well. If we plug in this importance ( $m$ ), we arrive at a generalized version of (3)

$$\phi_{d-m} = \frac{\sum_{i=1}^{|S|} \eta_i d_i m_i}{\sum_{i=1}^{|S|} d_i m_i}. \quad (4)$$

A very useful application of power awareness as defined by (4) is in predicting and enhancing battery lifetime of the system  $H$ . In the context of maximizing battery lifetime, the importance,  $m_i$ , of a scenario is simply the energy dissipated by that scenario

$$m_i = E(H, s_i).$$

Plugging in this definition of importance into (4) and simplifying using (1), we get

$$\begin{aligned} \phi &= \frac{\sum_{i=1}^{|S|} E(H_{\text{perfect}}, s_i) d_i}{\sum_{i=1}^{|S|} E(H, s_i) d_i} \\ &= \mathbf{E}(\text{Normalized Battery Lifetime}). \end{aligned} \quad (5)$$

The interpretation above is important enough that we do not attach any subscript and consider it *the* default definition of power awareness. It is one of the most useful interpretations of power awareness since it directly equates the metric to the *expected battery lifetime of the system normalized to the lifetime of the perfect system*. To see why this is so, note that the denominator is a summation of the expected energy consumption per scenario and, hence, equal to the expected energy consumed

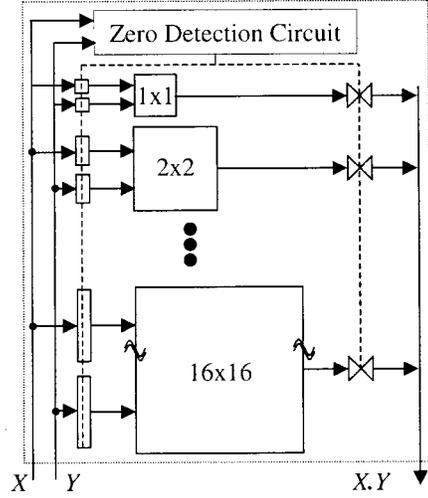


Fig. 7. The  $\hat{H}_{\text{perfect}}$  system mimics the abstract  $H_{\text{perfect}}$  system by using an ensemble of 16 dedicated point multipliers and a zero-detection circuit as the scenario detector.

by the system  $H$  displaying a scenario distribution  $d$ . Similarly, the numerator is the expected energy dissipation of the perfect system. Since battery lifetime is inversely proportional to the energy consumed,  $\phi$  as defined by (5) represents the normalized battery lifetime of the system  $H$ . For our 16-bit multiplier, it turns out that  $\phi = 0.57$ , which implies that in a speech filtering application, this multiplier will have half the battery lifetime of a perfectly power-aware system. Note that in equating  $\phi$  to the expected normalized lifetime of the system, we ignore second-order effects like the dependence of the battery capacity on the discharge pattern [6].

Coming back to our original motivation—resolving which of  $H_2$  or  $H_3$  is more power aware—we see that the question cannot be answered in the battery lifetime sense without specifying a scenario distribution. We can unambiguously answer which of  $H_2$  or  $H_3$  is more efficient *for any specified*  $d$ .

### III. ENHANCING POWER AWARENESS

#### A. Motivation

Enhancing the power awareness of a system is composed of two well-defined steps:

- 1) Engineering the best possible point systems.
- 2) Engineering the desired system using the point systems constructed in step 1) such that power awareness is maximized.

In the context of a power-aware multiplier, the first task is understood easily. It involves engineering  $1 \times 1$ ,  $2 \times 2$ , ...,  $16 \times 16$ -bit multipliers that are as efficient as possible while performing  $1 \times 1$ ,  $2 \times 2$ , ...,  $16 \times 16$ -bit multiplications respectively. The second task—that of engineering a system *using point systems*—is illustrated by the multiplier shown in Fig. 7.

Note the overall similarity between this figure and the abstraction of  $H_{\text{perfect}}$  in Fig. 3. The ensemble of point systems was used as an abstract concept in the context of explaining  $H_{\text{perfect}}$ 's energy curve. In the present context, however, we are illustrating an *actual physical realization* ( $\hat{H}_{\text{perfect}}$ ) of a system based on this concept. The basic idea is to detect the

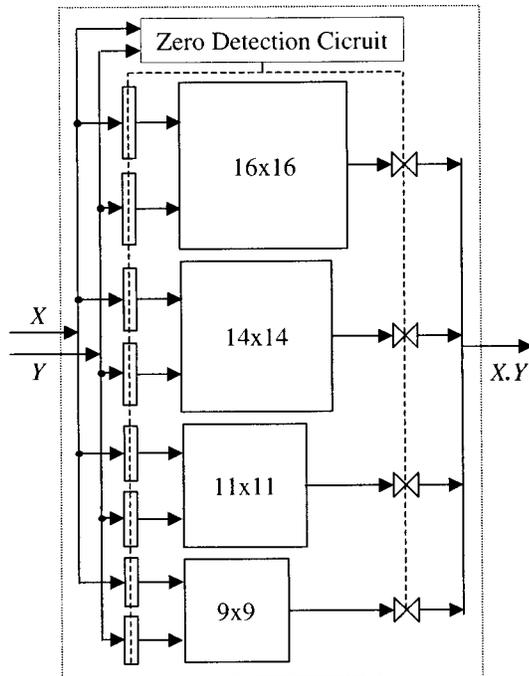


Fig. 8. The four-point ensemble multiplier system.

precision of the incoming operands using a zero detection circuit and then route them to the most suitable point system. In the case of  $H_{\text{perfect}}$ , the matching is done trivially—multiplier operands which need a minimum precision of  $i$  bits are directed to a  $i \times i$ -bit multiplier. Similarly, the output of the chosen multiplier is multiplexed to the system output. As we might expect though,  $\hat{H}_{\text{perfect}}$  has significant overheads. Even if we were to ignore the area cost of having 16 point multipliers and focus solely on the power awareness, the energy curve of  $\hat{H}_{\text{perfect}}$  would not be the same as  $E_{\text{perfect}}$ . While the scenario execution itself is the best possible, the energy costs of determining the scenario (the zero detection circuit), routing the multiplicands to the right point system and routing the result to the system output (the output mux) may be nonnegligible.

A system that uses a less aggressive ensemble in an effort to reduce the energy overhead of assembling point systems is shown in Fig. 8.

The basic operation of this multiplier ensemble is the same. The precision requirement of the incoming multiplicand pair is determined by the zero detection circuitry. Unlike the previous 16-point ensemble, this four-point ensemble is not complete and hence mapping scenarios to point systems is not one–one. Rather, precision requirements of the following:

- 1)  $\leq 9$  bits are routed to the 9-point multiplier;
- 2) 10, 11 bits are routed to the 11-point multiplier;
- 3) 12–14 bits are routed to the 14-point multiplier;
- 4) 15, 16 bits are routed to the 16-point multiplier.

Similarly, the results are routed back from the activated multiplier to the system output. While scenarios are no longer executed on the best possible point systems (with the exception of 16, 14, 11, and 9 bit multiplications), this ensemble has the advantage that energy overheads of routing are significantly reduced over  $\hat{H}_{\text{perfect}}$ . Also, while the scenario to point system

mapping of the four-point ensemble is not as simple as the one–one mapping, it is important to realize two things. First, the energy dissipated by the extra gates needed for the slightly more involved mapping in the four-point ensemble is low relative to that dissipated in the actual multiplication. Second, only four systems have to be *informed* of the mapping decision compared to 16 earlier. This reduction further offsets the slight increase in scenario mapping.

It is not difficult to see the basic tradeoff at work here. Increasing the number of point systems decreases the energy needed for the scenario execution itself but increases the energy needed to coordinate these point systems. Hence, it is intuitively reasonable to assume the existence of an *optimal* ensemble of point systems which strikes the right balance. Motivated by this possibility, we can now pose the problem of *enhancing* power awareness as follows.

1) *Determining the Most Power-Aware System Practically Realizable (I)*: Can we construct a system  $H_{\text{optimal}}$  as an ensemble of point systems drawn from  $P$  such that  $H_{\text{optimal}}$  is unconditionally more power aware than any other such constructed system?

It is not difficult to show that unconditional power awareness only leads to partial ordering. Hence, the existence of a unique  $H_{\text{optimal}}$  as defined above cannot be guaranteed. In other words, while it is possible to present a *set* of systems that are unconditionally more power aware than all other solutions, we cannot guarantee that this set will have only one member. In fact, this last condition is highly unlikely to occur in practice—unless routing costs are very low or very high compared to scenario execution costs (in which cases the optimal ensembles would be the complete and single-point solutions, respectively). Hence, in general, it is futile to search for an “optimal” ensemble of point systems that is unconditionally better than all other ensembles. Thus, we set our ambitions lower and ask a slightly different question.

2) *Determining the Most Power-Aware System Practically Realizable (II)*: Can we construct a system  $H_{\text{optimal}}$  as an ensemble of a point systems drawn from  $P$  such that  $H_{\text{optimal}}$  is more power aware than any other such constructed system for a specified scenario  $d_{\text{given}}$ ?

Since a specified scenario distribution  $d_{\text{given}}$  imposes a total ordering on the power awareness of all possible subsets of the power-set of  $P$ , it is easy to prove the *existence* of an optimal system. Note that the proof based on total ordering is nonconstructive—it only tells us that  $H_{\text{optimal}}$  exists but doesn’t help us determine what it is. This is unfortunate because a brute-force search of the optimal subset of  $2^P$  would require an exponential number of operations in  $|P|$ —a strategy that takes unacceptably long even for modestly large  $P$ .

To see if there are algorithms that can find  $H_{\text{optimal}}$  in non-exponential run-times we pose the problem more formally as follows.

### B. Formal Statement of the Power-Awareness Enhancement Problem

Given:

- 1)  $F$ : A system function to be realized.

- 2)  $S$ : A set of scenarios characterized by a *scenario basis*. For example, the basis in our multiplier example was the precision of the multiplicands.
- 3)  $P$ : A set of point systems available to realize  $F$ . Also, we denote the power-set of  $P$  i.e., the set containing all the subsets of  $P$  by  $\mathcal{P}(P)$ .
- 4)  $d$ : The scenario distribution

$$d: S \rightarrow \mathcal{R}^+$$

that obeys the additional constraint

$$\sum_{s_i \in S} d(s_i) = 1$$

expected of a distribution functions.

- 5)  $e$ : The energy function

$$e: S \times P \rightarrow \mathcal{R}^+.$$

In other words, for any given pair  $(s_i, p_i)$ ,  $e$  gives us the energy consumed when scenario  $s_i$  is executed on point system  $p_i$ . For instance, the energy taken by a  $4 \times 4$ -bit multiplication is different on a 4-bit multiplier than, say, a 9-bit multiplier. If the scenario  $s_i$  cannot be executed on the point system  $p_i$ , an infinite cost is assigned to the pair.<sup>4</sup>

- 6)  $w$ : The *energy overhead* cost function

$$w: \mathcal{P}(P) \rightarrow \mathcal{R}^+.$$

Hence,  $w$  maps every subset of  $P$  to the sum of all energy spent in coordinating the points in the subset ensemble (routing energy, determining the scenario, mapping the scenario, etc.).

- 1) *Form of the Solution*:

- 1) An ensemble of point systems  $h \in \mathcal{P}(P)$ .
- 2) A corresponding *mapping*

$$g: S \rightarrow h$$

i.e.,  $g$  maps each scenario to a point system in  $h$ . For instance, in the four-point multiplier example above,  $g$  would specify that scenarios 1–9 execute on the 9-bit multiplier, 10, and 11 execute on the 11-bit multiplier and so on.

*Measure of the Solution*: Since we are interested in the expected battery lifetime of a system, the measure of a proposed solution— $h$ ,  $g$ —is the expected energy consumption  $E(h)$  given by

$$E(h) = w(h) + \sum_{i=1}^{|S|} e(s_i, g(s_i)) d_i. \quad (6)$$

Note that like all models, the one for energy above can be made increasingly more precise. For instance, the interconnect energy will display some dependence on scenario distributions. Hence, the  $w$  function can take  $d$  as an argument and so on. However, we refrain from these refinements because our intent here is to use a realistic but simple model to analyze the complexity of finding a solution.

<sup>4</sup>This has the nice property of preventing any *infeasible* ensemble of point systems. For instance the ensemble {14-bit, 11-bit, 9-bit multiplier} is a subset of  $P$ , but is an infeasible solution since 15- and 16-bit multiplications cannot be executed on any of these point systems.

- 2) *Problem*: Determine a solution that minimizes the measure.

It seems likely that the problem of finding  $H_{\text{optimal}}$  as stated above belongs to the class of NP-complete problems. In other words, we cannot hope to determine the construction of  $H_{\text{optimal}}$  in polynomial time [7]. The proof of NP-completeness and suitable approximation algorithms to find  $H_{\text{optimal}}$  are beyond the scope of this paper. At this point, it suffices to say that we are currently working with heuristics to determine  $H_{\text{optimal}}$  and as the application examples in the next section show, these heuristics yield good results. Finally, it is important to note that the re-engineered system must not violate any constraints that the original system was expected to obey unless the constraints are relaxed explicitly for the sake of increasing power awareness.

### C. Reducing Area Costs Incurred in Enhancing Power Awareness

Our focus in the preceding discussion was maximizing power awareness without regard to implementation costs like area. While such an approach is acceptable for systems where power awareness must be increased at all costs, it might need to be reformulated for those with area constraints. In these latter cases, the problem would be to find the most power-aware ensemble for a specified distribution *while obeying specified area constraints*. If the area costs are significant enough, it is often beneficial to think of implementing an ensemble *temporally* rather than *spatially*. For example, instead of a spatial layout of four multipliers as illustrated earlier, we must imagine a temporal layout of these four multipliers. In other words, the same physical hardware is reconfigured to a 16, 14, 11, or 9-bit multiplier as desired. A possible solution is to selectively shut off the parts of a 16-bit multiplier and make it behave like smaller multipliers. While such a solution may or may not save any energy in the case of multipliers (due to the overhead of latches and the latch control network), it is an important illustration of the fact that spatial mappings are not the only means to implement ensembles. In fact, our discussion of power-aware processors in the next section is a real world example of a system where a purely *temporal* ensembles increase power awareness significantly.

If we reformulated the fitness measure of an ensemble to include its silicon real-estate costs, we can expect that the optimal ensemble might neither be totally temporal nor totally spatial, but a hybrid. Continuing our multiplier example, it might mean that we end up with, say three, point multipliers, one or more of which are reconfigurable to differing extents. To find such an optimal, possibly hybrid, solution, we must extend the spatial formulation of the problem (as stated in the last section) in two ways. First, we must allow new point systems that correspond to temporally reconfigurable ensembles. In the multiplier example, this means including point systems like a  $n \times n$ -bit multiplier that can be explicitly reconfigured as more efficient  $r_0 \times r_0, r_1 \times r_1, \dots, r_k \times r_k$ -bit multiplier where  $r_i \in [1, n]$ . Second, we must factor in the energy costs of *temporal reconfiguration*. In simple models, these costs could be factored into the scenario execution energy itself. Hence, the  $e$  function that maps  $(s_i, p_i)$  pairs to energy values would not only include the cost of

executing scenario  $s_i$  on point system  $p_i$ , but also the expected energy cost of possibly reconfiguring  $p_i$  to execute scenario  $s_i$ .

Finally, it is worth noting that although we motivated temporal and hybrid ensembles to reduce area costs, such ensembles might in fact *outperform* purely spatial ones in power awareness even if we allow unlimited area for both. In other words, one should not expect area saving temporal ensembles to be always inferior than the best possible, area-unconstrained spatial ensemble. With some thought, this should not be surprising because in moving from spatial to temporal ensembles, we *augment* our set of point systems, allowing temporal ensembles a larger solution space to pick from.

#### IV. PRACTICAL ILLUSTRATIONS OF ENHANCING POWER AWARENESS

It is amply clear from the previous section that enhancing power awareness by constructing ensembles of point systems carefully chosen from  $P$  is a general technique that can be used not just for multipliers but other systems as well. In this section, we shall illustrate how this ensemble idea can be applied to enhance the power awareness of multiported register files, digital filters, a dynamic voltage scaled processor and wireless sensor networks. In each case, we express the problem in terms of the framework we have developed above and characterize the power awareness of the system. Then we use an ensemble construction to enhance power awareness. It is interesting to note that these applications cover not just spatial ensembles, but purely temporal (processor example) and spatial-temporal hybrid ensembles (register files and adaptive digital filters) as well.

##### A. Power-Aware Register Files

1) *Motivation:* Architecture and VLSI technology trends point in the direction of increasing energy budgets for register files [8]. The key to enhancing the power awareness of register files is the observation that over a typical window of operation, a microprocessor accesses a small group of registers repeatedly, rather than the entire register file. This locality of access is demonstrated by the 20 benchmarks comprising the SPEC92 suite that were run on a MIPS R3000 simulator (Fig. 9). More than 75% of the time, no more than 16 registers were accessed by the processor in a 60-instruction window. Equally important, there was strong locality from window to window. More than 85% of the time, less than five registers changed from window to window.

If we think of the number of registers the processor typically needs over a certain instruction window as a scenario, the curves in Fig. 9 are simply scenario distributions. When a processor uses  $n$  registers over a window, we would want the file to behave as if it were a  $n$  register (i.e.,  $n$  word) file. This would lead to a register file architecture, which is significantly more power aware than one where the files always behaves as a, say, 32-register file. Smaller files have lower costs of access because the switched bit-line capacitance is lower. Hence, from a power awareness perspective, over any instruction window, we want to use a file that is as small as possible.

2) *Modeling the Problem:* We model the problem of increasing the power awareness of register files using the terminology developed in Section III.

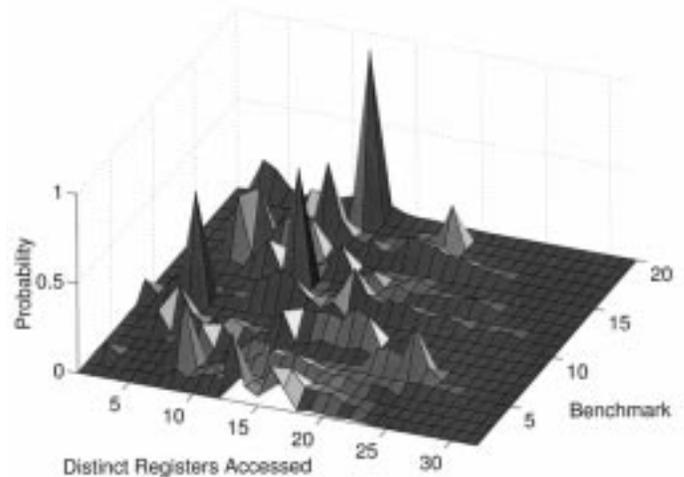


Fig. 9. Distribution showing number of distinct registers accessed in a 60-instruction long window for the MIPS R3000 processor executing 20 benchmarks.

- 1) Function to be realized ( $F$ ): A  $2^n$ -word  $\times$   $m$ -bit register file with  $r$  read ports and  $t$  write ports.
- 2) Set of scenarios ( $S$ ): We use the number of registers accessed in an instruction window of length  $k$  to characterize scenarios. In picking  $k$ , one must remember that the longer the window, the larger the number of accessed registers, leading to less differentiation. A smaller window needs frequent scenario to point-system mapping changes which has energy costs too. In this paper, we choose  $k = 60$ .
- 3) Point systems available ( $P$ ): We assume the availability of  $1, 2, 4, \dots, 2^n$  word  $\times$   $m$ -bit register files with  $r$  read ports and  $t$  write ports. Hence, the number of words is the only degree of freedom allowed. While it is possible to have more exotic point systems (different read and write ports, bit widths, etc.), our choice is reasonable and works well when practically implemented.
- 4) Scenario distributions ( $d$ ): The 20-register access profiles in Fig. 9 are the scenario distributions.
- 5) Energy function ( $e$ ) and overhead energy ( $w$ ): All register file results were obtained by generating layouts using a custom-written program, extracting the layouts into SPICE netlists, and simulating the netlists in *PowerMill* with test vectors. The register files themselves were implemented using NAND-style row decoding in dynamic logic with precharged address decoding lines, and use a standard cross-coupled inverter pair for static storage. The file that we use to illustrate power-aware engineering is a  $32 \times 4$  bit, 3 read, 2 write port file. We chose  $m = 4$  although, as long as  $m$  is not unreasonably large, it does not affect the results in any material way. This is because the bit-line switched capacitance is essentially independent of  $m$ .

3) *Results:* A monolithic 32-word file has an awareness  $\phi$  varying between 0.2 and 0.3 for the different distributions. Using a (16,8,4,4) ensemble as shown in Fig. 10 we increase  $\phi$  to between 0.5 and 0.8 for the different distributions.

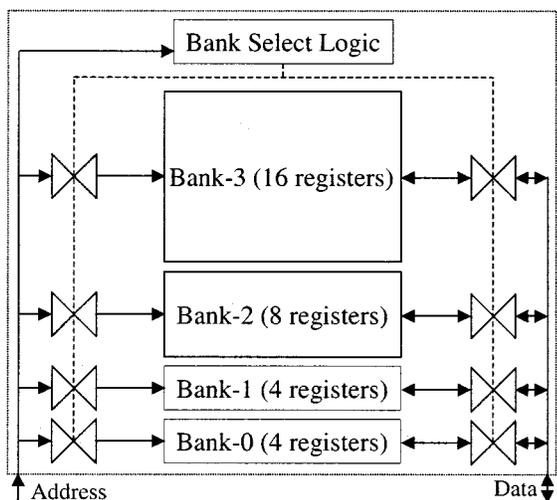


Fig. 10. This (16,8,4,4) ensemble of four register files increases power awareness by two times for the  $d_1$  scenario distribution and about 2.5 times for the  $d_2$  distribution.

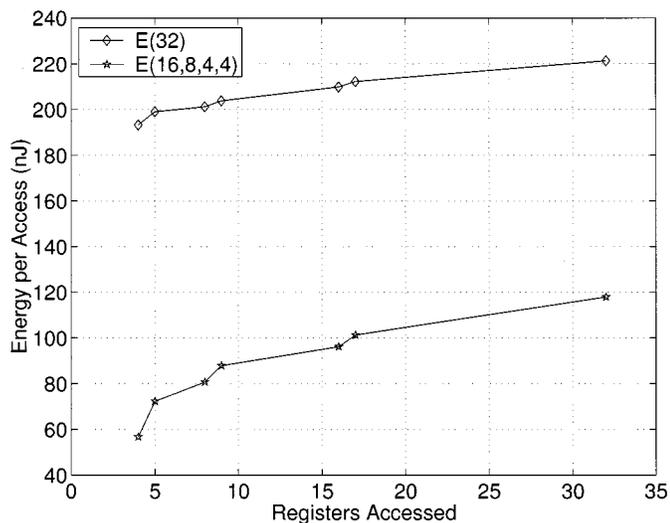


Fig. 11. Energy curves of a monolithic 32-word file and the nonuniform four-point ensemble (16,8,4,4).

The energy curves of the single point solution (a 32-word file) and the four-point (16,8,4,4) ensemble are plotted in Fig. 11. Interpreted in terms of lifetime increase, the nonuniform four-point ensemble increases lifetime by between 2 and 2.5 times for the twenty distributions used.

## B. Power-Aware Filters

1) *Motivation*: There are significant motivations for investigating power-aware filters. As an example, consider the adaptive equalization filters that are ubiquitous in communications ASICs. The filtering quality requirements depend strongly on the channel conditions (line lengths, noise, and interference), the state of the system (training, continuous adaptation, freeze, etc.), the standard-dictated specifications and the quality of service (QoS) desired. All these considerations lead to tremendous scenario diversity which a power-aware filtering system can exploit [9].

## 2) Modeling the Problem:

1) Function to be realized ( $F$ ):

$$y[n] = \sum_{k=1}^{\text{Number of Taps}} h[k]x[n-k].$$

We have chosen a 64-tap, 24-bit filter.

2) Set of scenarios ( $S$ ): We use the basis  $\langle \text{Number of taps}, \text{Precision} \rangle$  to characterize the operational state that the system is in. The precision refers to both the data and coefficients.

3) Point systems available ( $P$ ): We assume the availability of all possible  $\langle \text{Number of taps}, \text{Precision} \rangle$  filters. We pick distributed arithmetic (DA) filters as described in [10] because they allow the energy to scale with both taps and desired precision. A four-tap DA filter is shown in Fig. 12. In each step, incoming and delayed data bits with the same weights are used to access a memory that has precomputed combinations of coefficients  $h[n]$ . This pre-computed value is then either added to or subtracted from a partially accumulated sum  $y[n]$ . The number of cycles needed is the same as the precision of the multiplicands. Thus, filters that have to manage precisions lower than the maximum can scale their voltages lower and still meet deadlines. Hence, this filter architecture allows extremely fine-grained control over energy dissipation and is highly aware. The problem is that since it relies on lookups, it has to resort to partitioning and hybrid schemes to remain feasible as the number of taps grows [10].

4) Scenario distributions ( $d$ ): We model the desired filtering quality using a synthetic distribution centered around a  $\langle 16\text{-taps}, 8\text{-bit} \rangle$  scenario. Such a distribution might prevail, for instance, when the system is in the freeze mode with a high line quality and/or low signal-to-noise ratio (SNR) requirements.

5) Energy function ( $e$ ) and overhead energy costs ( $w$ ): We parametrically model the filters described since the nature of the DA architecture lends itself to a reasonably accurate energy model [5]. The energy curve that results from this model is shown in Fig. 14. Note that while energy scales about linearly with the number of taps, it scales in a quadratic manner with precision. This is because lower precision filters can scale their voltage.

3) *Results*: Before we illustrate suitable ensemble constructions that enhance power awareness, it is instructive to look at the energy characteristics of the perfect system. Fig. 15 plots the product of scenario energy and scenario probability for the perfect system (which would be an ensemble of  $64 \times 24 = 1536$  point systems). The scenario energy-probability product curve shows the energy consumed as a function of scenarios. Note that although energy consumption around the (16-tap, 8-bit) scenario is clearly prominent in Fig. 15, some high-precision high-tap scenarios also account for significant contributions to the overall energy consumed. This is easily understood because although they occur infrequently (as seen in the distribution plot in Fig. 13), they consume significant energy when they do occur (as seen in the energy plot in Fig. 14).

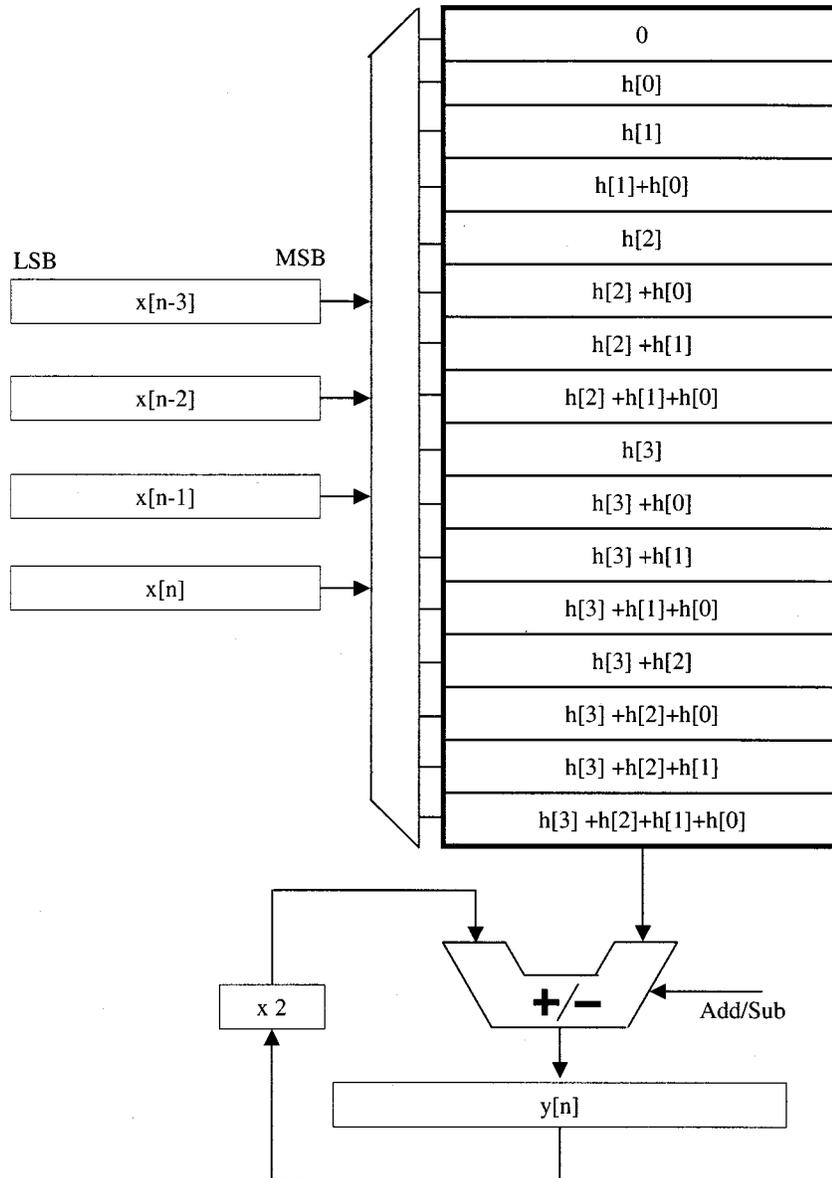


Fig. 12. A four-tap distributed arithmetic (DA) filter architecture.

If we used a single, 64-tap, 16-bit filter (i.e., a one point ensemble), the resulting energy–probability product curve turns out to be the one plotted in Fig. 16. A rough comparison of the energies consumed by different scenarios in this system to that in the perfect system shows that the former is significantly nonoptimal. In fact, the power awareness  $\phi$  of the single point system is only 0.17.

To find more optimal ensembles, we programmed a brute-force exhaustive search algorithm that could find the best four-point ensemble. Due to exponential timing requirements, it was retired beyond four points and a greedy heuristic used instead. The optimal four-point ensemble turns out to be  $((64, 24), (64, 15), (58, 20), (51, 10))^5$  as shown in Fig. 17.

Its energy–probability curve is plotted in Fig. 18. Note that although not quite optimal, it has a power awareness of 0.52, which is over three times better than the single point ensemble.

<sup>5</sup>(64, 24) stands for 64-tap, 24-bit precision, etc.

Interestingly, our greedy heuristic revealed that if we include four more points—(30,17), (43,23), (64,7), (43,13)—in the above ensemble, we can increase the power awareness to 0.64.

### C. Power-Aware Processors

1) *Motivation:* Having looked at three examples of power-aware subsystems (multipliers, register files and digital filters), we illustrate power awareness at the next level of the system hierarchy—a power-aware processor that scales its energy with workload. Unlike previous examples, however, this one illustrates how an ensemble can be realized in a *purely temporal* rather than a spatial manner.

It is well known that processor workloads can vary significantly and it is highly desirable for the processor to scale its energy with the workload. A powerful technique that allows such power awareness is dynamic frequency and voltage scaling [11]. The basic idea is to reduce energy in nonworst-case workloads

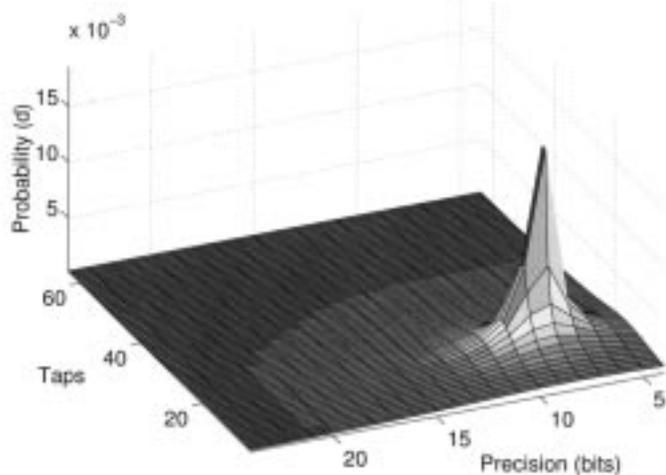


Fig. 13. Probability distribution of anticipated filter quality.

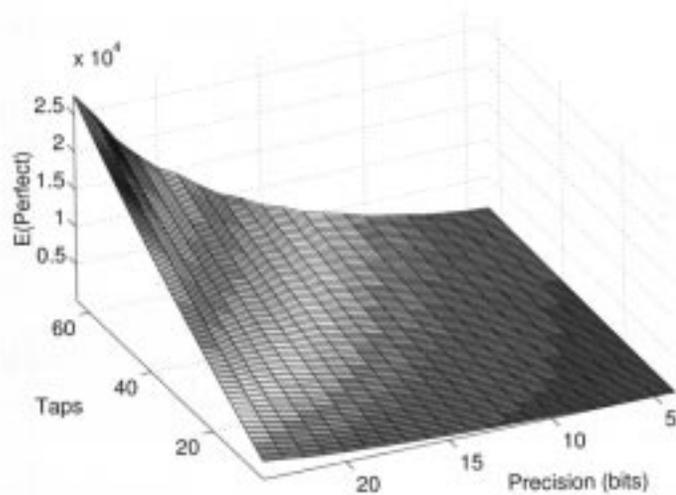


Fig. 14. The “perfect” energy curve for 64-tap, 24-bit DA-based filtering.

by extending them to use all available time, rather than simply computing everything at the maximum clock speed and then going into an idle or sleep state. This is because using all available time allows one to lower the frequency of the processor, which, in turn, allows scaling down the voltage leading to significant energy savings [11]–[13].

In terms of the power-awareness framework that we have developed, a scenario would be characterized by the workload. The point systems would be processors designed to manage a specific workload. As the workload changes, we would ideally want the processor designed for the instantaneous workload to execute it. It is clear that implementing such an ensemble spatially is meaningless and must be done temporally using a dynamic voltage scaling system. Before we look at such a system, we state the problem more concisely.

## 2) Modeling the Problem:

- 1) Function to be realized ( $F$ ): Any workload running on a given processor. In this case, the processor we use is the Intel StrongArm SA-1100. The workload variation comes from a variable tap filter running on the SA-1100 (the reader is referred to [12] for details of the actual setup).

- 2) Set of scenarios ( $S$ ): We use the workload  $\in [0, 1]$  as a basis (with 0 for no workload to 1 for a completely utilized processor). Note that the workload requirement has a one–one mapping to a frequency and voltage requirement.
- 3) Point systems available ( $P$ ): A point system in this case would refer to the SA-1100 designed for a specific workload. Since we are interested in achieving power awareness through voltage scaling, this corresponds to a SA-1100 with a dedicated voltage and frequency (which are the minimum possible to achieve the workload). Also, due to an infinite number of scenarios, there are infinite number of point systems—one for every workload between zero and one. Equivalently, in terms of voltages, there are an infinite number of point systems between zero and  $V_{dd}^{\max}$ , the latter being the highest voltage the SA-1100 can run at, which also corresponds to its highest frequency and a workload of unity.
- 4) Scenario distribution ( $d$ ): We assume, for simplicity, that all workloads are equally probable. As we see below, such an assumption is pessimistic and in real applications, we can expect to see even better numbers for power awareness.
- 5) Energy function ( $e$ ) and energy overhead ( $w$ ): The energy dissipated by the SA-1100 was physically measured.

3) *Results:* We now analyze an *actually constructed system* that recently demonstrated this power-awareness concept [12]. The overall setup is summarized in Fig. 19 adapted from [12]. The basic idea is that a power-aware operating system ( $\mu$ -OS) running on the SA-1100 determines the current workload, scales the frequency accordingly and then instructs a switched regulator supply to scale the voltage accordingly. Again, the reader is referred to [12] for the details of the setup and the dynamic voltage circuitry.

The DVS system uses a *temporal ensemble* of 32-point systems with voltage levels almost uniformly distributed between zero and  $V_{dd}^{\max}$ . The energy-curves of a nonaware (fixed) voltage system and the implemented dynamic voltage system are plotted in Fig. 20.

For uniform workload distributions, power awareness improves from 0.63 for a fixed voltage system to 1.0 for the implemented dynamic voltage system. Note that although the 32-point ensemble is by no means perfect, it was chosen as a reference to define the power awareness (since the ratio of the power awareness of one system to the other is independent of the perfect system). Hence, for uniform load distributions, DVS leads to battery lifetime increases of about 60%.

## D. Power-Aware Data Gathering Wireless Networks

1) *Motivation:* Increasing levels of integration and advanced low-power techniques are enabling *ad hoc*, wireless networks of microsensor nodes. Each node is composed of a sensor, analog preconditioning circuitry, A/D, processing elements (DSP, RISC, FPGA, etc.) and a radio link, all powered by a battery. Replacing high quality macrosensors with such networks has several advantages—robustness and fault tolerance, autonomous operation for years, enhanced data quality, and optimal cost performance [14], [15]. Such data

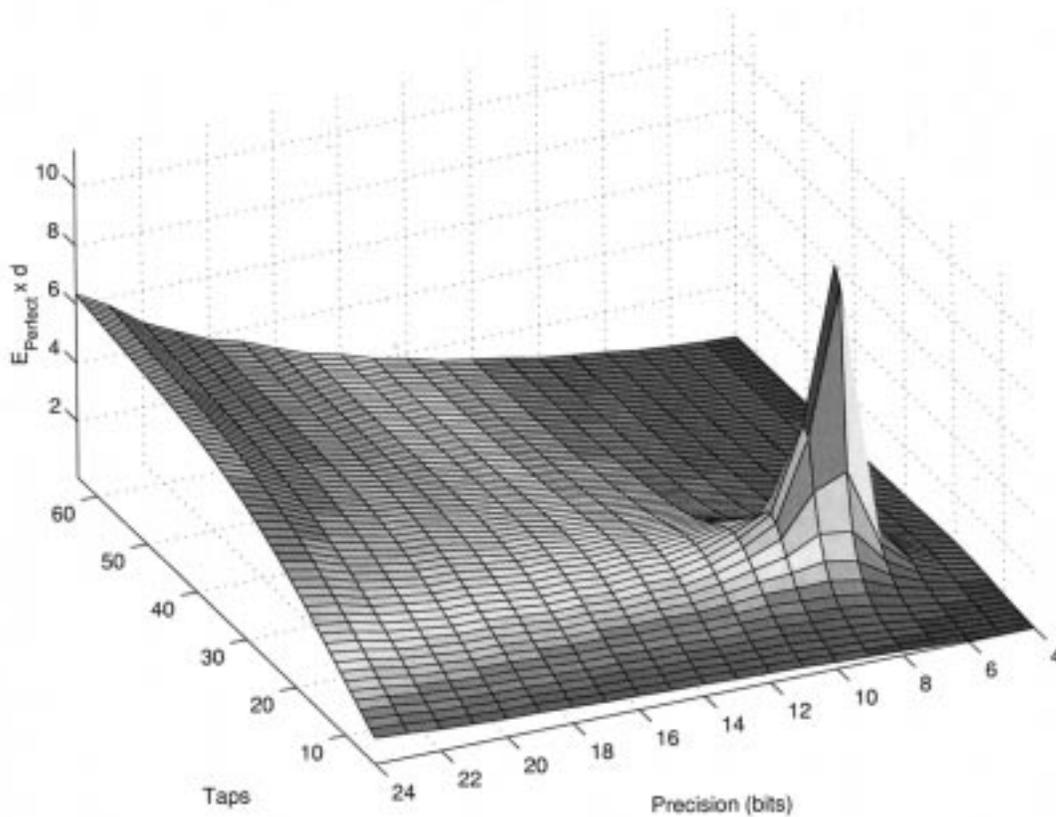


Fig. 15. The perfect energy-probability curve for 64-tap, 24-bit DA-based filtering.

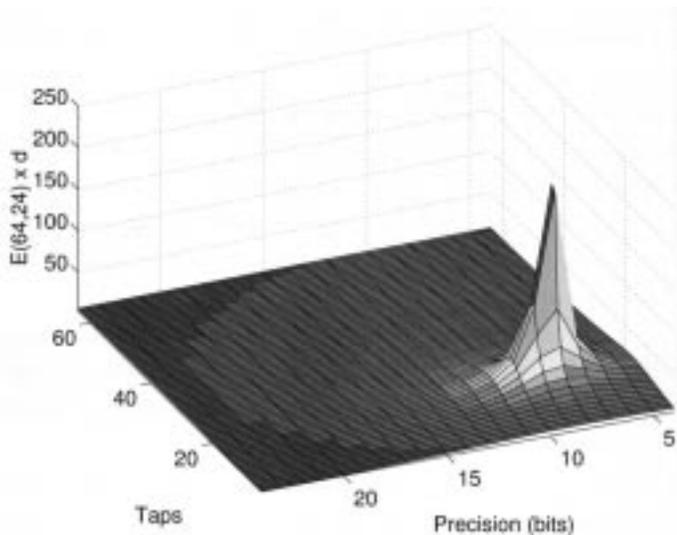


Fig. 16. The energy-probability curve for a single 64-tap, 24-bit DA-based filter.

gathering networks are expected to find wide use in remote monitoring applications, intrusion detection, smart medicine, etc. An illustrative data gathering network is shown in Fig. 21. The network is live as long as it can guarantee that any source in region  $R$  will be sensed and the data relayed back to a fixed base station. To accomplish this objective, different nodes take on different roles over the lifetime of the network as seen in the figure. A noteworthy point is that nodes must often

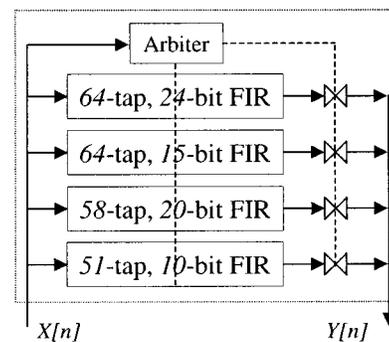


Fig. 17. This four-point ensemble of DA filters improves power awareness by over three times over a single point system.

change roles even if the source does not move. This is to enable energy drain to be spread throughout the network which leads to increased lifetimes. An assignment of roles to nodes that leads to data gathering is termed a *feasible role assignment*.<sup>6</sup> A data-gathering strategy or *collaborative strategy* can be completely characterized by specifying a sequence of feasible role assignments and the time for which the assignment is sustained.

A key challenge in unlocking the potential of data-gathering networks is attaining long lifetime despite the severely energy constrained nature of the network. For example, networks composed of ultra-compact nodes carrying less than 2 J of battery

<sup>6</sup>We also require feasible role assignments to be nonredundant, i.e., data from a sensor should not be routed via multiple links.

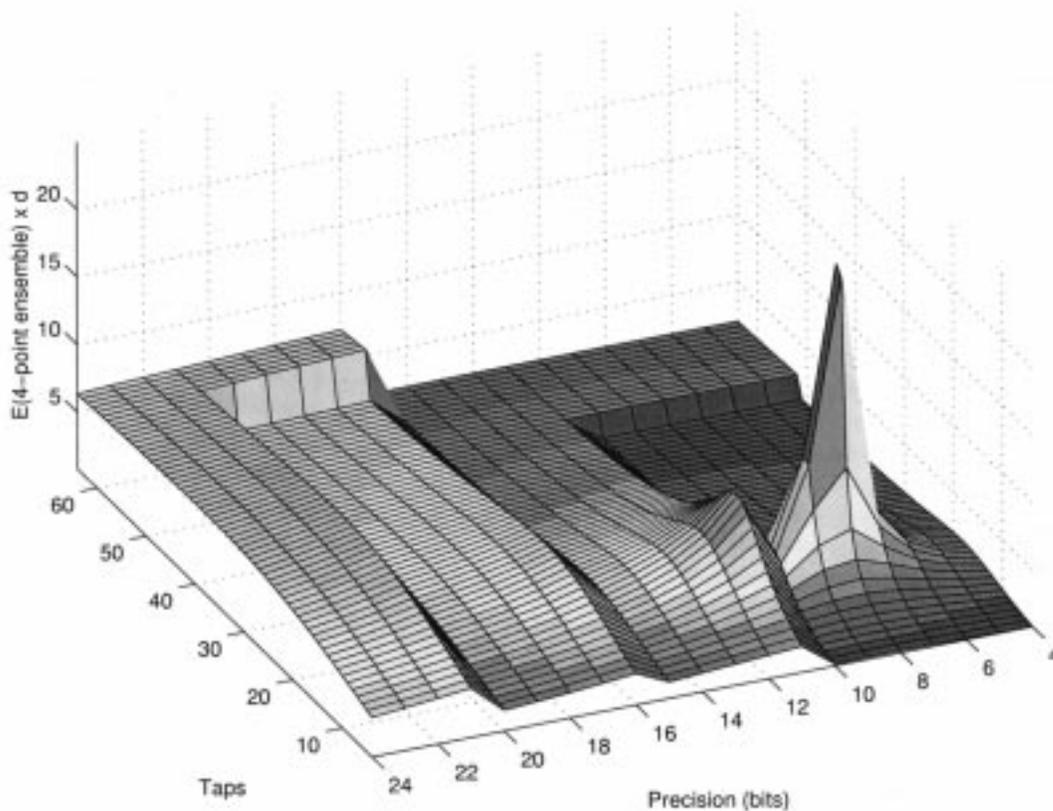


Fig. 18. The energy–probability curve for the four-point ensemble in Fig. 17. Note the similarity to the “perfect” curve in Fig. 15.

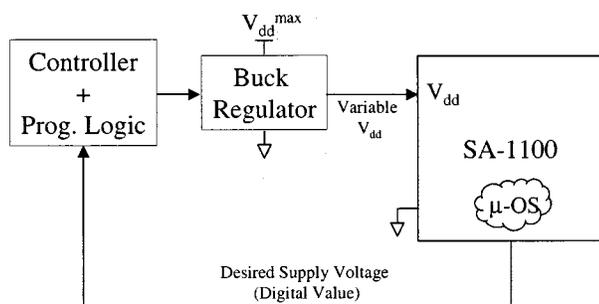


Fig. 19. The dynamic voltage scaling (DVS) system used to enhance power awareness.

energy might be expected to last for five to ten years [16]. It is possible to address these challenges by power-aware design. Data-gathering networks can be aware to the desired quality of gathered data, to changing source behavior, to the changing state of the network and, finally, to the environment in which they reside. In this section, we focus on this last aspect and tackle the problem of designing a power-aware data gathering network that tracks changes in the environment to maximize energy efficiency. It is well known that the transmit power can be scaled with changing noise power to maintain the same SNR and, hence, the same link performance. A more holistic approach is to view environmental variations as affecting changes in the energy needed to process a bit (i.e., carry out some computation on it) versus the energy needed to communicate it. A power-aware network is then simply one that can track changes

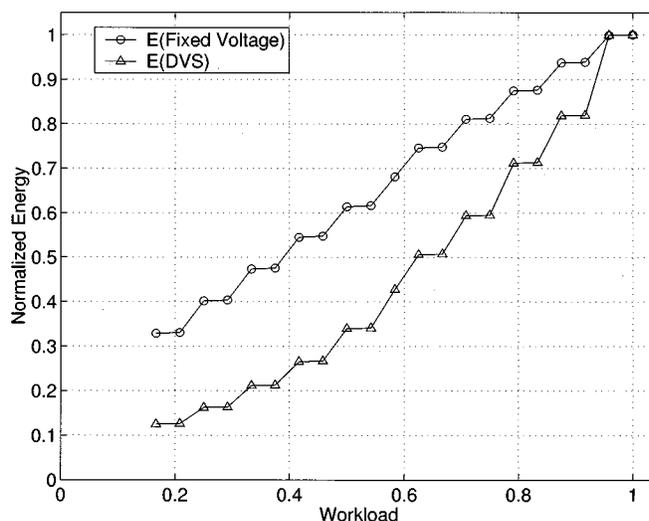


Fig. 20. The energy curves of the fixed voltage and DVS system. Both curves have been normalized with respect to the maximum load case.

in the *computation-to-communication energy ratio*. For large ratios or equivalently high computation costs, the network will favor unaggregated or raw sensor streams. Conversely, for low ratios, i.e., high communication costs, aggregation will be favored. Hence, the challenge in power-aware data gathering is to determine and execute the collaborative strategy that assigns roles optimally for a specified computation to communication energy ratio.

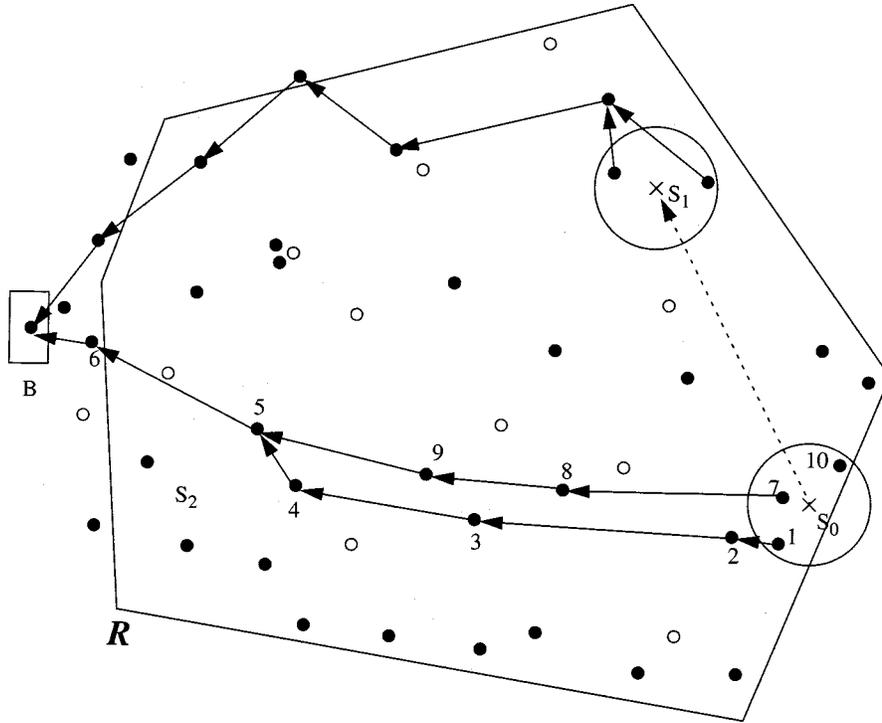


Fig. 21. A sensor network gathering data from a circularly observable source (denoted by a  $\times$ ) residing in the shaded region  $R$ . Live nodes are denoted by  $\bullet$  and dead ones by  $\circ$ . The basestation is marked  $B$ . In this example, we require that at least two nodes sense the source. When the source is at  $S_0$ , nodes 1 and 7 assume the role of sensors and nodes  $2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$  form the relay path for data from node 1 while nodes  $7 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 6$  form the relay path for data from node 7. Data might be aggregated into one stream at node 5. This is not the only feasible role assignment that allows the source to be sensed. For instance, node 10 could act as the second sensor instead of node 7 and  $10 \rightarrow 7 \rightarrow 8 \rightarrow 4 \rightarrow 5 \rightarrow 6$  could form the corresponding relay path. Also, node 6 might aggregate the data instead of node 5, etc. Finally, note how the sensor, aggregator and relay roles must change as the source moves from  $S_0$  to  $S_1$ .

## 2) Modeling the Problem:

- 1) Function to be realized ( $F$ ): Gathering data from a specified point source using a specified network. The network is specified by its topology (including the location of the base station) and the initial energy in the nodes.
- 2) Set of scenarios ( $S$ ): A scenario is characterized by the ratio of computation to communication energy.
- 3) Point systems available ( $P$ ): A point system simply corresponds to a collaborative strategy as defined above.
- 4) Scenario distribution ( $d$ ): We will analyze the power awareness for uniform scenario distributions.
- 5) Energy function ( $e$ ) and energy overhead ( $w$ ): The energy needed for communicating a bit is modeled using  $d^n$  path-loss model as  $\alpha_1 + \alpha_2 d^n$  where  $n$  is typically between two to four [17]. The energy required to aggregate two bits into one is denoted by  $\alpha_3$ . Nominal values of  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  are typically 150 nJ/bit, 10 pJ/bit/m<sup>2</sup> and 50 nJ/bit [16]. The computation to communication energy ratio is defined as  $\alpha_3/\alpha_2$  because variations in channel noise mainly mandate changes in the power dissipation in the transmit amplifier.

3) *Results:* To illustrate power-aware data gathering, we simulated an eight-node network for a wide range of computation to communication ratios centered about the nominal ratio. Specifically, the ratio was varied in steps of 3 dB starting with 30 dB below nominal and ending at 24 dB above nominal. For each ratio, the optimal collaborative strategy was determined via linear programming. This strategy was executed by the

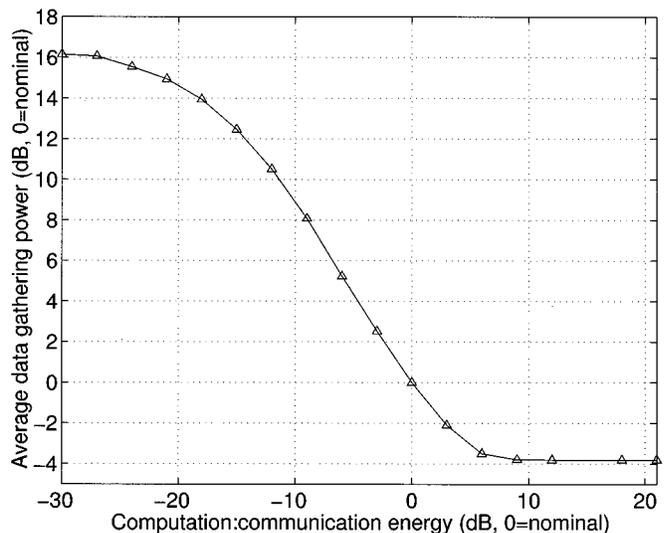


Fig. 22. The energy curve of a power-aware data gathering network.

network and the lifetime recorded. The inverse of the lifetime was used as a measure of the average data-gathering power dissipated in the network. Fig. 22 shows the variation in the data-gathering power with changing scenarios. The impact of adapting the data-gathering strategy to track the energy ratio is clear—the power-aware network displays close to two orders of dissipation diversity. Translated in terms of the proposed power-awareness metric, the temporal ensemble

of collaborative strategies is 3.22 times more power aware than an unaware network for a uniform scenario distribution. Rephrased, power-aware data-gathering can increase network lifetime by more than three times compared to an unaware network.

## V. CONCLUSION

In this paper, our objective was twofold: to *quantify* the increasingly important notion of power awareness of a VLSI system and having done that to propose a systematic technique to *enhance* this quality.

The first step in quantifying the power awareness of a general system was to develop the notion of a perfectly power-aware system ( $H_{\text{perfect}}$ ). The awareness of this system was shown to be an upper bound on practically achievable power awareness. In the next step, we proposed a power-awareness metric whose physical interpretation is the expected battery lifetime of a system normalized to the lifetime of the perfect system.

Next, the problem of enhancing power awareness was treated formally using the concept of *ensembles of point systems*. We showed that constructing systems by intelligently putting together dedicated point systems could significantly enhance power awareness. The basic factor that limited a monotonic increase in power awareness as more and more point systems were put together was the increasingly amount of energy spent in coordinating these point systems. Hence, the problem of finding an optimal subset of point systems that struck the right balance was formally proposed. While it seems unlikely that this optimal subset can be found using polynomial time algorithms, greedy heuristics were seen to work reasonably well.

The technique of ensemble construction was illustrated using five different applications—multipliers, register files, digital filters, dynamic voltage processors, and data-gathering networks. Significant power-awareness improvements leading to system battery lifetime improvements in the range of 60% to 200% were seen.

It is our sincere hope that the power-awareness metric proposed here will be used to quantify this important aspect of VLSI systems and that the proposed framework will be employed by system architects to engineer systems that scale their power and energy requirements with changing operating scenarios leading to significant improvements in overall battery lifetimes.

## REFERENCES

- [1] A. Sinha, A. Wang, and A. P. Chandrakasan, "Algorithmic transforms for efficient energy scalable computation," in *Proc. Int. Symp. Low-Power Electronics and Design*, Italy, Aug. 2000, pp. 31–36.
- [2] S. H. Nawab, *et al.*, "Approximate signal processing," *J. VLSI Signal Processing Syst. Signal, Image, and Video Technol.*, vol. 15, no. 1/2, pp. 177–200, Jan. 1997.
- [3] L. McMillan and L. A. Westover, "A forward-mapping realization of the inverse discrete cosine transform," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1992, pp. 219–228.
- [4] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A System Perspective*. Reading, MA: Addison-Wesley, 1994.

- [5] A. Sinha and A. P. Chandrakasan, "Energy efficient filtering using adaptive precision and variable voltage," in *12th Annu. IEEE ASIC Conf.*, Washington, DC, Sept. 1999, pp. 327–331.
- [6] T. Martin and D. Siewiorek, "A power metric for mobile systems," in *Proc. 1996 Int. Symp. Lower Power Electronics and Design*, Monterey, CA, Aug. 1996, pp. 37–42.
- [7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.
- [8] V. Zyuban and P. Kogge, "The energy complexity of register files," in *Proc. 1998 Int. Symp. Low-Power Electronics and Design (ISLPED)*, Monterey, CA, Aug. 1998, pp. 305–310.
- [9] C. J. Nicol *et al.*, "A low power 128-tap digital adaptive equalizer for broad-band modems," *IEEE J. Solid State Circuits*, vol. 32, pp. 1777–1789, Nov. 1997.
- [10] R. Amirtharajah, T. Xanthopoulos, and A. P. Chandrakasan, "Power scalable processing using distributed arithmetic," in *Proc. 1999 Int. Symp. Low-Power Electronics and Design*, San Diego, CA, Aug. 1999, pp. 170–175.
- [11] V. Gutnik and A. P. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 5, pp. 425–435, Dec. 1997.
- [12] R. Min, T. Furrer, and A. P. Chandrakasan, "Dynamic voltage scaling techniques for distributed microsensor networks," in *Proc. IEEE Computer Society Annual Workshop VLSI (WVLSI '00)*, Orlando, FL, Apr. 2000, pp. 43–46.
- [13] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," in *Proc. Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, Feb. 2000, pp. 294–295.
- [14] G. Pottie, "Wireless sensor networks," in *Proc. Information Theory Workshop*, Killarney, Ireland, June 1998, pp. 139–140.
- [15] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
- [16] W. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, Massachusetts Institute of Technology (MIT), Cambridge, MA, 2000.
- [17] T. Rappaport, *Wireless Communications: Principles & Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.



**Manish Bhardwaj** was born on February 16, 1976 in Delhi, India. He received the B.A.Sc. (honors) degree in computer engineering from the Nanyang Technological University, Singapore and the S.M. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1997 and 2001, respectively. He is currently working toward the Ph.D. degree at MIT.

From 1997 to 1999, he was a Digital Integrated Circuits Designer and later a Digital Signal Processing (DSP) Architect with the microelectronics research and development arm of Siemens Semiconductors (now Infineon Technologies), Singapore (1997–1998) and Villach, Austria (1998–1999). His research interests include very large scale integrated systems, signal processing, and information theory.



**Rex Min** received the S.B. and M.Eng. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1998 and 1999 respectively. He is currently working toward the Ph.D. degree at MIT.

His research interests center on power aware design and implementation of large-scale embedded systems, with an eye toward emerging wireless applications.

Mr. Min is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi, and has received the National Defense Science and Engineering Graduate Fellowship (NDSEG).



**Anantha P. Chandrakasan** (S'87–M'95) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994, respectively.

Since September 1994, he has been with the Massachusetts Institute of Technology (MIT), Cambridge, MA, where he is currently an Associate Professor of Electrical Engineering and Computer Science. He is coauthor of *Low Power Digital CMOS Design* (Norwell, MA: Kluwer, 1995) and

a coeditor of *Low Power CMOS Design* and *Design of High-Performance Microprocessor Circuits* (New York: IEEE Press, 1998). His research interests include the ultralow-power implementation of custom and programmable digital signal processors, distributed wireless sensors, multimedia devices, emerging technologies, and computer-aided design tools for very large scale integrated (VLSI) systems.

Dr. Chandrakasan has served on the technical program committee of various conferences including ISSCC, VLSI Circuits Symposium, DAC, and ISLPED. He has served as a Technical Program Co-Chair for the 1997 International Symposium on Low-Power Electronics and Design (ISLPED), VLSI Design 1998, and the 1998 IEEE Workshop on Signal Processing Systems, and as a general co-chair of the 1998 ISLPED. He served as an elected member of the Design and Implementation of Signal Processing Systems (DISPS) Technical Committee of the Signal Processing Society. He was the Signal Processing Subcommittee Chair for ISSCC from 1999 to 2001. He is Program Vice-Chair for ISSCC 2002. He held the Analog Devices Career Development Chair from 1994 to 1997. He was an Associate Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS from 1998 to 2001. He received the NSF Career Development Award in 1995, the IBM Faculty Development Award in 1995, and the National Semiconductor Faculty Development Award in 1996 and 1997. He has received several Best Paper Awards, including the 1993 IEEE Communications Society's Best Tutorial Paper Award, the IEEE Electron Devices Society's 1997 Paul Rappaport Award for the Best Paper in an Electron Devices Society publication during 1997 and the 1999 Design Automation Conference Design Contest Award.