Embedded Power Supply for Low-Power DSP

Vadim Gutnik and Anantha P. Chandrakasan, *Member, IEEE*

***Abstract*—The use of dynamically adjustable power supplies as a method to lower power dissipation in DSP is analyzed. Power can be reduced substantially without sacrificing performance in fixed-throughput applications by slowing the clock and lowering supply voltage instead of idling when computational workload varies. This can yield a typical power savings of 30–50%. If latency can be tolerated, buffering data and averaging processing rate can yield power reductions of *an order of magnitude* in some applications. Continuous variation of the supply voltage can be approximated by very crude quantization and dithering: a four-level controller is sufficient to get within a few percent of the optimal power savings. Significant savings are possible only if the voltage can be changed on the same time scale as the variations in workload. A chip has been fabricated and tested to verify the closed-loop functionality of a variable voltage system. The controller takes only 0.4 mm^2 and draws a maximum of 1 mW at 2 V with a 40 MHz clock. The control framework developed is applicable to generic DSP applications.**

***Index Terms*—Low-power DSP, variable supply voltage, workload averaging.**

I. Introduction

A. Motivation

MOST techniques to lower power consumption of integrated circuits (IC's) assume static behavior; that is, circuit and system parameters are chosen at design time to minimize power dissipation. In fact, in some applications adjusting the circuit *during operation* could save more power.

The number of operations performed per sample in many digital signal processor (DSP) systems can be minimized dynamically by exploiting time-varying signal statistics. MPEG video is one such example. Since frames of video are highly correlated to each other, most digital processing begins by comparing consecutive frames and processing only the differences. The amount of computation necessary to process the differences depends on how much the image changes from frame to frame. Typically, each frame is divided into small blocks and only those that change more than a threshold are processed. Thus, the amount of computation per frame in an MPEG encoder varies dramatically between a scene change (the entire frame encoded) and most other incrementally different frames. A more generic application is described in [1]: the number of taps of an FIR filter is varied based on the power of the out-of-band noise. The idea is to keep just enough taps in the FIR such that the stopband energy in the output is below a specified limit. Power-down techniques can be used to make power dissipation directly proportional to the computational workload per sample (henceforth simply called workload).

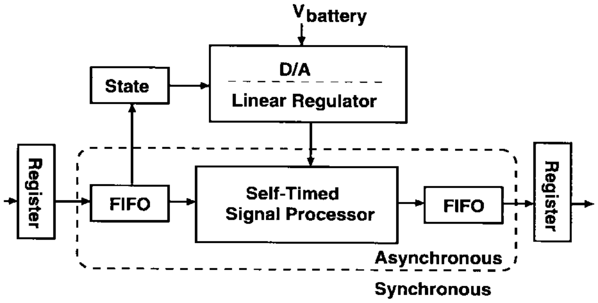

Fig. 1. Self-timed variable voltage system [2].

The power can be reduced further if a variable power supply is used in conjunction with a variable clock-speed processor. The basic idea is to lower supply voltage and slow the clock during reduced workload periods instead of working at a fixed speed and idling. The analysis and design of such a variable supply-voltage system will be presented in this paper.

B. Proposed System

Self-timed systems have been suggested to take advantage of data dependencies. A small, self-timed variable voltage-supply system has been demonstrated by Nielsen *et al.* [2]. The system of Fig. 1 consists of a ripple-carry adder and synchronization buffers at input and output; the fullness of the input buffer is used to control the supply voltage through a linear regulator. Although such systems track delays very well, the overhead to generate completion signals for each logic operation can be substantial, in terms of both delay and switching activity.

We propose using a synchronous design instead. A synchronous system cannot take advantage of bit-level dependencies in the computation (e.g., there is no way to exploit the data-dependent delay variation in an adder). However, algorithmic variations in workload (e.g., variations in the number of additions to be done per sample) can still be exploited to lower power. Like the self-timed variable-voltage system, some buffering would be needed on the input and output in the general case for synchronization, but a synchronous system could be designed *without* the completion generation signals, using static CMOS or any other logic families, and thus with very low overhead. A block diagram of the system developed in this work is shown in Fig. 2. Input data are buffered in the FIFO; the buffered data can be used to estimate the minimum sufficient processing rate. A loop around the voltage regulator

Manuscript received September 8, 1996; revised May 15, 1997. This work was supported by DARPA under Contract DAAL01-95-K-3526. V. Gutnik was supported by an NDSEG Fellowship.

The authors are with the Massachusetts Institute of Technology, Cambridge MA 02139 USA.

Publisher Item Identifier S 1063-8210(97)09204-4.

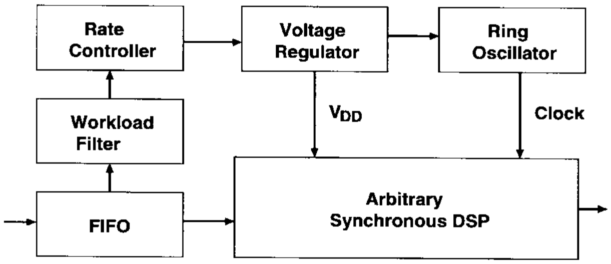

Fig. 2. Proposed synchronous, workload-dependent variable voltage system.

and the ring oscillator establishes a supply voltage at which the critical path just meets timing requirements.

II. Variable Supply Fundamentals

Since the goal of a variable supply system is to minimize power, it is best to start by considering what sets the power dissipation of a digital system. If both the technology and architecture are fixed, the determining factor in power dissipation is timing: at higher voltages, the circuits operate faster but use more energy.

A. Expected Power Savings

A convenient formalism is that digital signal processing deals with data as discrete *samples*. If, as in most cases, the samples come at fixed time intervals, the average power and computation for the processor can be related to the energy per sample and the computation performed therein.

We derive $E(r)$, the energy per sample as a function of normalized processing rate, from the assumption that to work at rate r, the gate delays and clock period can be $1/r$ times longer than nominal times, and that voltage can be lowered correspondingly. First-order CMOS delay models (i.e., ignoring subthreshold and velocity saturation effects) [3] can be manipulated to yield the energy dissipation per sample [4]

$$E(r) = CV_0^2 T_s f_r r \left[\frac{V_t}{V_0} + r/2 + \sqrt{r\frac{V_t}{V_0} + (r/2)^2}\right]^2 \qquad (1)$$

where C is the average switched capacitance, V_t is the device threshold voltage, $V_0 = (V_{\max} - V_t)^2/V_{\max}, T_s$ is the sample period, f_r is the maximum clock speed, (i.e., at $V_{DD} = V_{\max}$), and r is the *normalized sample processing rate*, or the clock speed normalized to f_r. For comparison, conventional digital logic works at a fixed voltage and idles if the computation finishes early, so $E_{\text{fixed}}(r) = E(1)r$.

Fig. 3 shows a plot of (1), for $V_{\max} = 2$ V and $V_t = 0.4$ V along with the fixed-supply line. The ratio of the two curves—($E(r)/E(1)r$)—gives the energy savings ratio per block for any given r, but the ratio changes with r. At high rates the power is the same because the voltage is the same; at very low rates the variable voltage approaches V_t so the ratio is $(V_t/V_{DD})^2$. The area between the curves is a convenient measure of the energy savings achievable by varying supply voltage.

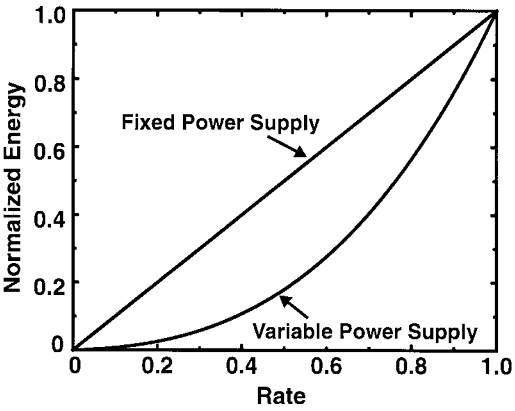

Fig. 3. Energy versus rate for variable and fixed supply systems.

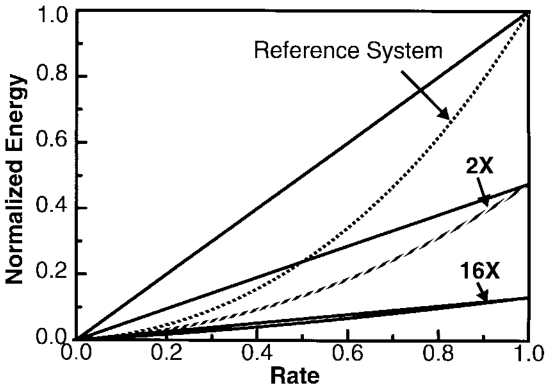

Fig. 4. Effect of parallelization on power.

B. Parallelization

The plot of energy versus rate is a convenient tool to analyze the influence of different architectures on power. In particular, the effect of parallelization is easy to show. By copying a circuit block N times and dividing the system clock by N, throughput is maintained. Since the effective critical path is clocked N times slower, supply voltage can be lowered, and the system runs at lower power [3]. Algebraicly, the parallel system has an energy–rate relation given by

$$E_{\text{par}}(r; N) = NE(r/N) + \text{Overhead}(N). \qquad (2)$$

The overhead consists of the extra routing and multiplexing necessary for a parallelized system. Fig. 4 shows an E-r plot of (2) for $N = 1$ (the reference system) and for $N = 2$ and 16, again with the fixed-supply lines for comparison, ignoring overhead.

There are two important observations to be made from the plot. First, note that at relatively low rates the curves are coincident; if typical workload is a small fraction of the maximum workload, the variable-voltage system performs as well as the parallelized, but has a much smaller area. The second is that the E-r curve has a smaller curvature for the higher Ns. Equivalently, the area between the constant supply and variable supply curves is relatively smaller, so the power savings achievable by varying voltage diminish for higher Ns. This is exactly the same effect that limits the energy savings from further parallelization: in both cases the energy savings arise because energy per clock cycle increases with speed. As V_{DD} approaches V_t circuit delay increases rapidly, so slowing the clock allows only small changes in V_{DD}, and if voltage does not change, the energy per clock cycle does not change with rate. In fact, due to subthreshold operation, (1) does not model device performance well if V_{DD} approaches V_t. However, gate delays increase so rapidly near the threshold voltage that the result is nearly correct.

III. Rate Control

The equations in the previous section define the static relationship between power and processing rate; in other words, the instantaneous power as a function of rate. The dynamic behavior, or the average power as a function of a sequence of rates, is equally important.

A. Averaging Rate

There is a subtle but significant distinction between the processing rate r and what will be called the *computational workload,* denoted w. The rate r is the processing speed, or more simply the system clock frequency, while w is a measure of how much processing needs to be done on the incoming blocks of data. For example, a digital video application may have a maximum clock rate of 50 MHz and computation dependent on what fraction of the image changed. Half of the image changing on a certain frame corresponds to $w = 1/2$. If the clock frequency happens to be 30 MHz during the computation of that frame, $r = 0.6$; as long as both r and w are normalized, the computation on that sample will finish for $r \geq w$. Even in the cases where it is not necessary, it is often advantageous to buffer the workload so that r need not follow w exactly. Why incur the overhead of a buffer when r can be set equal to w?

1) Motivation Consider a compressed digital video sequence where about one of out of every 10 frames is a scene change while the rest of the frames are differentially coded and need only incremental updates. So w as a function of frame number is

$$w[n] = \begin{cases} 1 & n = 0, 10, 20, 30 \cdots \\ 0.1 & \text{otherwise.} \end{cases}$$

If we schedule the processing for each frame independently, then $r[n] = w[n]$. If the energy to process a full frame of video is normalized to one, a fixed supply system would use $(1 \times 1) + (9 \times 0.1) = 1.9$ normalized energy units. With a variable supply system, the less-than-full frames take $E(0.1)/E(1)$ energy, (which, for typical parameters is 0.016), so the energy falls to $(1 \times 1) + (9 \times 0.016) = 1.1$, a 40% savings.

The 1.1 energy unit figure still assumes that each frame is processed independently. If, instead, the computation for each 10-frame long sequence were done at the average speed, (so that $r[n] = 1 * 1 + 9 * 0.1/100 = 0.19$) the total energy would be $10 * E(0.19) = 0.42$ normalized energy units, which is an energy savings of $5\times$! This assumes, of course, that the extra latency can be tolerated and there is a buffer to hold the nine extra frames of data. Note that the amount of processing done remained the same; only the scheduling of the computation changed.

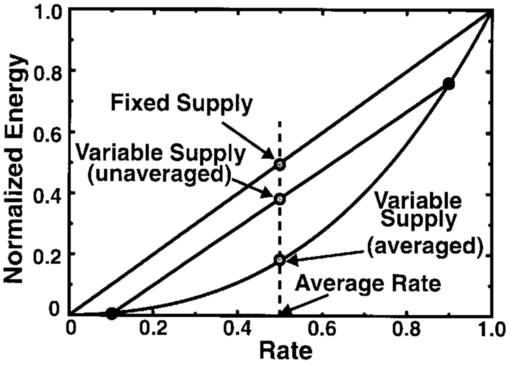

Fig. 5. Averaging example.

2) Convexity and Jensen's Inequality: Power dissipation falls when processing rate is averaged because energy is a *convex* (or "concave up") function of rate. Since both the energy per block and the number of operations done per block add linearly, the point describing the average rate and dissipated power for any two samples lie along the chord connecting their respective operating points; if the rate goes from 0.1 to 0.9, for example, the average power lies on the line connecting the points $[0.1, E(0.1)]$ and $[0.9, E(0.9)]$ as shown in Fig. 5. But since the Er curve is convex, any such chord always lies above the curve! So, if we average the workload over two sample periods and work at a fixed rate, the power will be lower. That is exactly what happened in the example—the operating point that corresponds to working at the average rate is much lower in power than the average of the high and low processing rate points. This result is summarized in *Jensen's inequality* [5].

$$\text{for convex } E(\cdot)\text{:} \quad \overline{E(r)} \geq E(\overline{r}). \tag{3}$$

In short, averaging the rate lowers power. It is this result that makes filtering the rate important for variable-supply systems; for comparison, note that the Er curve for fixed-supply systems is straight, so averaging the rate does not save power there.

B. Constraints

As argued above, the lowest possible power is achieved by averaging rate over all samples. In practice, latency and buffer size constraints limit the number of samples over which the computation can be averaged. Both constraints are invoked by assuming that there exists a finite buffer of size B that may not overflow as data arrives. Of course, the buffer cannot be allowed to underflow either—data cannot be processed before it arrives. Two more constraints need to be added to fully describe the problem: nonnegativity and causality. The buffer over- and under-flow constraints on the rate $r[n]$ given a workload $w[n]$ are

$$\forall n_0\text{:} \quad \sum_{n=-\infty}^{n_0} w[n] \geq \sum_{n=-\infty}^{n_0} r[n] \geq \sum_{n=-\infty}^{n_0-B} w[n]. \tag{4}$$

TABLE I
FIR MECHANICS

t	Data Buffer			Workload FIR			Avg.
0			0/6	6	0	0	2
1		0/3	2/6	3	6	0	3
2	0/21	0/3	5/6	21	3	6	10
3		0/12	6/21	12	21	3	12
4	0/27	0/12	18/21	27	12	21	20
5		0/9	5/27	9	27	12	16

Notation: done/total

The left inequality guarantees that no more data is processed than has arrived, i.e., that the buffer cannot underflow. The right one has the complementary role: to guarantee that all data that arrived B or more samples ago has been processed; that is, the buffer cannot overflow. Of course, in the long run, the average rate and the average processing load must be equal. This is consistent with the limits as stated; dividing the inequalities in (4) by $1/n_0$ and taking the limit as n_0 goes to infinity gives $\overline{r[n]} = \overline{w[n]}$.

As described by the constraints in (4), the optimum $r[n]$ depends on the statistics of the workload. Even for simple distributions the optimization rapidly becomes unmanageable. Furthermore, even when a functional form exists, the calculation of $r[n]$ from a $w[n]$ sequence could be prohibitive in terms of area and power. Fortunately, approximations exist.

1) FIR Filter of the Workload: When $w[n]$ is known *a priori*, a moving average of the workload is a simple, good approximation to the optimal $r[n]$. It can be shown that any weighted average satisfies the required constraints for any $w[n]$. This average can be written as an FIR filter with constraints as follows:

$$r[n] = \sum_{k=0}^{B-1} a_k w[n-k] \tag{5}$$

$$0 \leq a_k \leq 1 \qquad \sum a_k = 1. \tag{6}$$

The filter is causal by construction. It is also conveniently time-invariant and linear. Non-negativity of $r[n]$ is guaranteed by choosing the a_k as prescribed in (6). That this satisfies the over- and underflow constraints is shown below.

2) FIR Mechanics: Before starting the proof, the difference between the workload in the FIR and the sample in the input buffer should be stressed. The data samples in the input buffer are processed at varying speeds depending on the current value of r. Sometimes more than one sample is finished per sample period, sometimes less; the number of data samples in the input buffer (i.e., the frames of video or digitized speech or whatever else) varies with time. On the other hand, the FIR filter performs a weighted average of the workload of the last B data samples, whether or not they have already gone through the DSP.

The difference is illustrated in Table I. Entries in the "Data Buffer" represent data to be processed, with the denominator giving the total work needed to finish the sample and the numerator the work done so far. Note that the workloads here are not normalized to 1; the normalization is a matter of convenience. At $t = 3$, more than one sample is processed in one sample period. If only the workloads of the data samples in the data buffer were averaged, the rate would be lower than required by (5), and eventually the buffer would overflow.

3) FIR Avoids Buffer Over- and Underflow: The proof proceeds by direct substitution of (5) into (4)

$$\begin{aligned}\sum_{n=-\infty}^{n_0} r[n] &= \sum_{n=-\infty}^{n_0} \sum_{k=0}^{B-1} a_k w[n-k] \\ &= \sum_{k=0}^{B-1} a_k \sum_{n=-\infty}^{n_0} w[n-k] \\ &= \sum_{k=0}^{B-1} a_k \sum_{p=-\infty}^{n_0-k} w[p] \qquad p = n-k \qquad (7) \\ &\leq \sum_{k=0}^{B-1} a_k \sum_{p=-\infty}^{n_0} w[p] \\ &= \sum_{p=-\infty}^{n_0} w[p]. \qquad (8)\end{aligned}$$

The inequality shows that the buffer cannot underflow, and a similar sequence shows that it cannot overflow. Starting from (7) gives the following result:

$$\begin{aligned}\sum_{n=-\infty}^{n_0} r[n] &= \sum_{k=0}^{B-1} a_k \sum_{p=-\infty}^{n_0-k} w[p] \qquad p = n-k \\ &\geq \sum_{k=0}^{B-1} a_k \sum_{p=-\infty}^{n_0-B} w[p] \\ &= \sum_{p=-\infty}^{n_0-B} w[p] \qquad (9)\end{aligned}$$

which completes the proof. Intuitively, the buffers do not overflow because the system is linear and any single data sample is processed correctly. That is, if only one data sample required computation and all the others were zero, by the Bth sample time, exactly enough operations would have been completed to finish processing that sample. Since the $r[n]$ superpose, the system satisfies the processing constraints for all the data samples.

C. Update Rate

It is clear that data buffers allow r to be somewhat decoupled from w; as shown above, the rate can be averaged over several samples to lower power. It is a small step to push the buffering idea further: if the power (and hence the rate) are averaged over B samples, should not we only have to update r at $1/B$ of the sample frequency? As shown in Section IV, slowing the update rate can save power in the power supply. Unfortunately, this can have unwelcome repercussions in the power dissipated in the DSP. A didactic example, the quasi-Poisson queue, is analyzed below.

Poisson queues, or queues with exponentially distributed service and arrival times, may be used to model a variety of physical process, including some signal processing applications. For example, a packet-switched network may be modeled as a Poisson queue: the packets arrive independently, and are decoded and routed faster if they have fewer errors.

Because of the nature of the model is impossible to completely avoid buffer overflows; however, this model allows us to examine the power versus overflow–probability tradeoff. We may be interested in finding the processing rate at which the probability P_L of losing packets due to buffer overflow is less that some constant.

1) Constant Processing Rate: The probability of overflow can be derived easily by treating this as a Markov process [6]. If λ is the arrival rate and μ the service rate, P_L is

$$P_L = \frac{\left(\frac{\lambda}{\mu}\right)^{n-1}}{\sum_{k=0}^{n-1}\left(\frac{\lambda}{\mu}\right)^{k}}. \tag{10}$$

The higher the processing rate, the lower the probability of losing a packet; for a buffer of length 4 and $P_L = .1\%, (\mu/\lambda) \geq 9.6$. This means that to keep the overflow probability sufficiently low, the processor has to be able to process packets almost ten times as quickly as they arrive.

2) Variable Processing Rate: The high μ/λ ratio means that the packets must be serviced at a rate much higher than the average arrival rate, so the processor is idle nearly 90% of the time. This seems like an excellent application for a variable supply system: process quickly only when the queue starts to fill up.

Again solving the state equation we find that the steady state probabilities are

$$p \propto [\mu_2\mu_3\mu_4 \quad \lambda\mu_3\mu_4 \quad \lambda^2\mu_4 \quad \lambda^3]^T. \tag{11}$$

The expected power[1] is given by $[0, \mu_2^2, \mu_3^2, \mu_4^2] \cdot p$. Optimizing the μ_i for minimum power (with the same 0.1% overflow constraint) gives

$$\mu_2 \approx 3 \quad \mu_3 \approx 7.5 \quad \mu_4 \approx 32 \quad \text{and } \overline{E} \approx 5. \tag{12}$$

3) Sampled Queue: The system presented above requires the rate to change at arbitrary times—any time a sample arrives or is serviced, the rate changes. A better model for a realizable queue would have the rate changing at discrete times.

Evolution of the state of the queue from one sample to the next is derived in [4]; unfortunately, the optimization is nonalgebraic, so an approximate solution was obtained numerically. The minimum power solution for the length 4, 0.1% overflow case is plotted in Fig. 6 for a range of sample times.

For update times much faster than the arrival rate, the minimum power solution approaches the unconstrained minimum of (12). For times much slower than the arrival rate, the system cannot vary the rate as fast as the queue fills up, so the minimum is simply the fixed-rate minimum. The key observation is that for supply voltage variation to be effective, the designed voltage update time must be comparable to the characteristic workload variation time. In other words, the bandwidth of the power supply must exceed the rate at which data arrives.

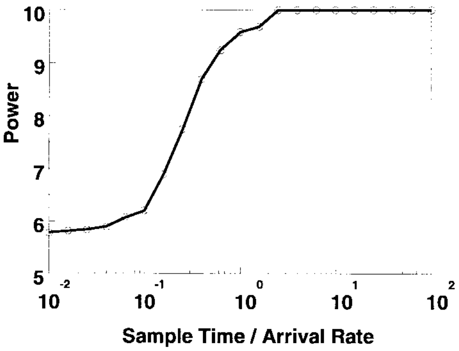

Fig. 6. Minimum power versus sample time.

IV. Variable Supply and Clock Generation

In variable supply systems, power savings are achieved by lowering the supply voltage as the system clock slows down; indeed, this is the *only* reason such a system saves power. In the context of this paper, the term "power supply" is used to mean the power converter that draws current from some battery or rectified source, filters and regulates the voltage level and outputs a supply voltage. The words "static" and "dynamic" will be used to describe the voltage level that is generated, *not* the internal operation of the converter; thus a switching converter that produces 5 V will be considered static.

A. Specifications

In the simplest static supply systems, the power supply is trivial—the battery voltage is used to power the chip directly. However, in most low power systems some regulation is required, and especially in the case of switching converters, low-pass filtering is needed as well. The output voltage of dynamic converters also needs to be filtered and regulated, but the performance criteria are somewhat altered from those for a conventional, static power supply.

First, the desired transient response is markedly different. Since the ideal static supply has no variations on the output, the low-pass filter cutoff frequency is designed to be as low as volume and cost constraints allow. A dynamic supply still needs a low-pass filter to attenuate ripple, but also needs fast step response to allow rate changes as described in Section III.

The second difference relates to the dc voltage level. A system with a static supply is typically designed to meet timing constraints at a specific voltage. At a lower level of abstraction, this means that feedback is established around the power converter to fix the output voltage as shown in the left half of Fig. 7.

A more efficient approach for fixed-rate systems was presented in [7], [8] where the feedback around the entire systems establishes a fixed circuit delay rather than a fixed voltage, as shown in the right half of Fig. 7.

Several systems have been designed to compensate for process and temperature variations [7]–[9]. In each case the circuit delays are measured indirectly by means of a ring oscillator with a period matched to critical path circuit delays. The ring oscillator is used as the VCO of a phase- or frequency-locked loop with power supply acting as control

[1] The energy terms should be $E[\mu_i]$, where $E(\cdot)$ is defined by (1) rather than by μ_i^2, but the optimization is much simpler with the simpler expression, and the results are essentially the same.

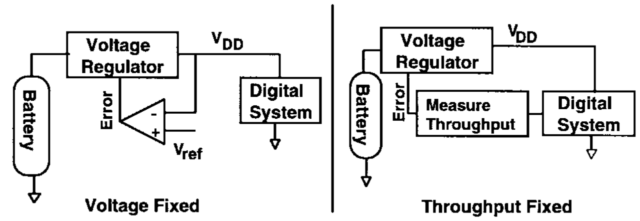

Fig. 7. Feedback loops around the power supply versus the system.

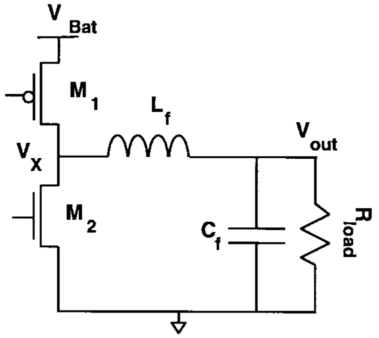

Fig. 8. Simplified buck converter schematic.

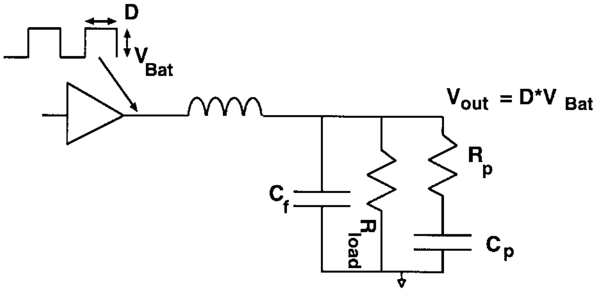

Fig. 9. Passive damping.

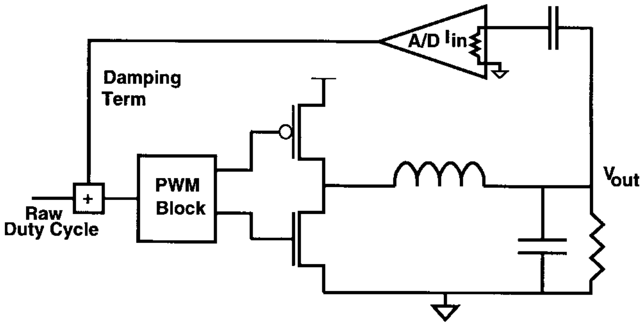

Fig. 10. Active damping proposed in this work.

voltage. When the loop settles, critical path delay matches the clock period.

The power and area overhead is small; a single loop may be sufficient for the entire chip. Stability concerns limit the speed of the loop, but since temperature changes slowly, the bandwidth is adequate.

The main advantage of the fixed-throughput approach is that the safety margin delay, the extra time allocated to make sure the chip meets timing constraints, needs only to account for small intradie process variations since the interdie delay variations are measured and compensated.

1) Switching Regulators and Active Damping: Switching voltage regulators are usually preferred where low power is a concern, because unlike the linear regulators, the losses approach zero with ideal elements, and efficiencies of over 90% have been reported [10].

A representative switching converter called a *buck converter* is shown in Fig. 8. When transistor M_1 is on and M_2 off, voltage V_x rises to V_{DD}. Conversely, with M_2 on and M_1 off, V_x approaches ground. The amplitude and duty cycle of the resultant square wave on V_x determine the dc level of the output voltage, and the ripple is attenuated by an LC filter.

To prevent ringing on the output of the switching converter, the LC filter should be damped. In most cases, the series resistance of the switches or the parallel resistance of the load presents enough damping. If that is not the case, extra damping needs to be added to limit oscillations.

A simple approach that works well for fixed supplies is to add a resistor R_p in parallel with the load; to avoid the dc dissipation, a large capacitor can be added in series with R_p, as shown in Fig. 9.

This turns out not be an efficient method for variable supplies because the parallel resistor dissipates energy every time the voltage changes. It is possible to avoid this by *actively damping* the loop; that is, introducing feedback into the system to emulate a resistor. The transfer function from the input to the output with a forward gain a and feedback gain f is $A = a/(1+af)$. The forward path is a second-order filter, so a has the form $1/(s^2+2\alpha s+\omega_0^2)$ where α is a damping term and $\omega_0 = \sqrt{1/(LC)}$ is the resonance frequency. Hence, the total response is

$$\begin{aligned} A &= \frac{1/(s^2+2\alpha s+\omega_0^2)}{1+f/(s^2+2\alpha s+\omega_0^2)} \\ &= \frac{1}{s^2+2\alpha s+\omega_0^2+f}. \end{aligned} \tag{13}$$

To increase the damping, the feedback needs to be of the form βs for some constant β; the current through a capacitor is of this form. A capacitor cannot be used directly because the feedback signal must interface to the digital part of the signal, so an A/D converter is needed. Fig. 10 shows the block diagram of an actively damped supply. The current through the capacitor is digitized and passed back to the pulsewidth modulator. The power cost of this scheme is in the A/D, but only a very low resolution and slow sample rate is needed, so a small, low-power converter can be used.

Simulated waveforms are shown in Fig. 11. Without any damping the voltage would oscillate at each step; with active damping the circuit behaves just as it would if it were passively damped, except that the power dissipation can be lower.

B. Loop Stability and Step Response

1) Open Loop: Two methods to establish a voltage appropriate for a given rate have been mentioned in the previous subsection. The first is an extension of the static supply approach; for every desired rate an adequate duty cycle level

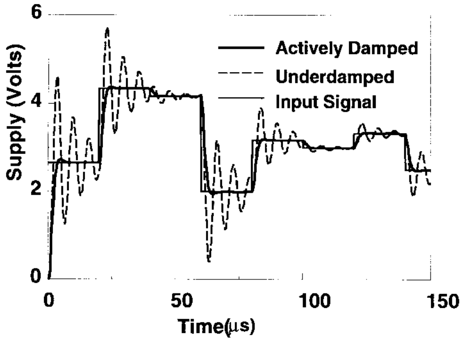

Fig. 11. Active damping waveforms.

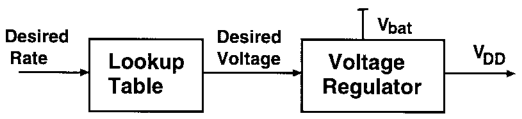

Fig. 12. Open-loop variable supply system.

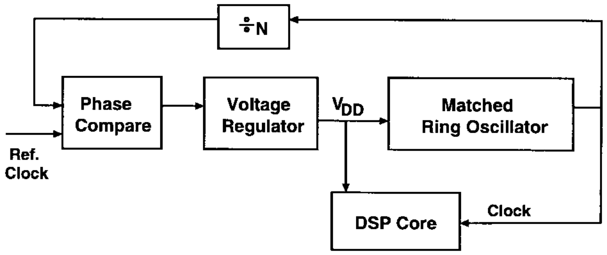

Fig. 13. Block diagram of a phase-locked loop.

would be read out of a ROM and passed to the voltage regulator, as shown in Fig. 12. When desired rate increases, the voltage goes up first to ensure that timing constraints will be met, and then the clock speeds up; when rate falls, the clock slows down and then the voltage drops. This has the advantage that changes in rate yield the fastest possible transitions on the output voltage, but as mentioned above, process and temperature variations would force the voltage to be high enough that delays meet worst-case, rather than actual, timing constraints.

2) Closed Loop—PLL: The phase-locked loop approach avoids this problem. Rather than having a separate clock and power supply, both the clock and power converter are part of a feedback loop, and the system clock is based on chip delays. A system inspired directly by [7] is shown in Fig. 13.

Just as in the case of the fixed supply system, the PLL adjusts supply voltage to the lowest possible level compatible with the required number of gate delays between registers. When the desired rate changes, the reference clock that the PLL sees changes and the loop relocks. The drawback is in the time constant of the changes.

The time constant is determined by the bandwidth of the power supply, the loop filter, and the extra pole introduced by the integration of frequency to phase. With no loop filter, the feedback pole at dc combines with the second-order pole from the power supply to give a peaked response loop gain. A loop filter with a bandwidth significantly less than that of the power supply output filters can be added, but then this bandwidth limits how fast the loop locks.

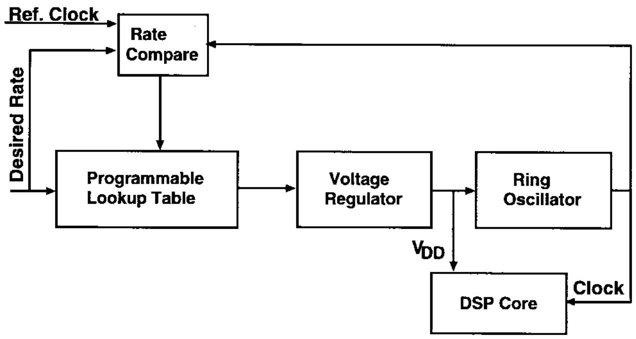

Fig. 14. Hybrid system proposed in this work.

A similar approach is presented by [8], who show that by incorporating a PID controller it is possible to circumvent some of the problems with the straightforward PLL. However, the resulting controller is more complicated and dissipates significantly more power than the one proposed here.

3) Our Proposed Hybrid Approach: The two previous schemes, the PLL and the lookup table, adjust the output voltage at different rates because they were originally intended for different functions. The PLL does best in tracking process variations where its low bandwidth is sufficient, and the lookup table is well suited to making fast voltage steps to predefined level. In fact, the characteristics are not mutually exclusive; it is possible to merge the look-up table approach with the phase-locked loop to get fast voltage steps and process tracking at the same time.

The lookup table should still be used to get the fastest possible voltage changes; however, if the voltage levels are stored in a RAM instead of ROM they can updated to track process and temperature. A schematic is shown in Fig. 14.

The rate comparison and updates can be done very slowly compared to the bandwidth of the power supply. For example, if a buck converter switches at 1 MHz, the output filters can have a bandwidth of ≈100 kHz. For comparison, temperature compensation can be done at the frequencies below 1 kHz. Thus, the dynamics of the power supply are insignificant in the feedback loop so there are no instabilities.

a) Quantization and dithering: Both the open-loop and the hybrid implementation have a lookup table to translate from rate to voltage. Since the overhead of the controller scales with the number entries in the lookup table, a smaller table is preferable to a larger one. Fig. 15 shows the Er curve for four-level voltage quantization. The lowest curve is the theoretical minimum E-r as predicted by (1); the area between it and the fixed-supply line is a measure of the power savings achievable by varying power.

If each sample must be processed at a fixed voltage and in one sample period (i.e., without buffering), the rate must be the next highest available rate. So, if the available rates are 0.25, 0.5, 0.75, and 1 and the sample workload is 0.6, the controller would have to choose the 0.75 rate and idle for part of the cycle. This gives the "stair-step" curve in Fig. 15.

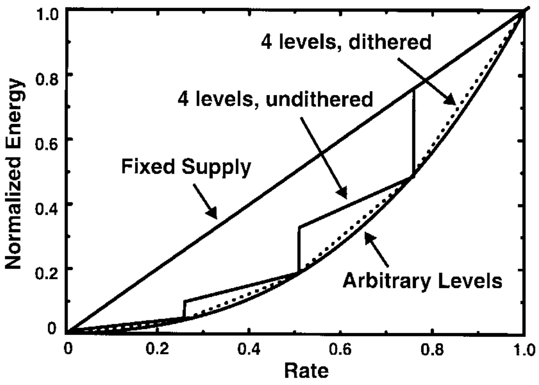

Fig. 15. Quantization effect.

If the voltage can be changed during the processing of one sample, the voltage can be dithered. In the example above, by processing for 40% a sample time at rate of 0.75 and 60% at 0.5, the average rate can be adjusted to the 0.6 that is needed. This *dithering* leads to an Er curve that connects the quantized (E, r) points. As the figure shows, even a four-level lookup table is sufficient if dithering is used.

If the sample period is too short to allow dithering within the sample, the same effect can be achieved by allowing processing of one sample to extend beyond one sample period; in fact, this was the assumption made in the FIR filter analysis of Section III-B1). If one sample is processed at a rate higher than its workload because of quantization, the next will be processed at a lower rate.

V. Implementation and Testing

A. Chip Design

A chip was designed to test the stability of the feedback loop and to verify that timing constraints are met as the modified phase-locked loop changes the clock and supply voltage. Since the focus is on the variable supply voltage control, only a token amount of processing is done by the DSP subsection, but it can be reconfigured to emulate applications with long sample periods (i.e., computationally intensive cases like video processing) as well as applications with shorter sample periods. Similarly, the rest of the circuitry is designed for flexibility rather than efficiency.

1) Block Diagram: The block diagram of the chip is shown in Fig. 16. At the highest level of abstraction, the test chip consists of four blocks. The FIFO and the DSP comprise the datapath, while the supply controller and ring oscillator are part of the control loop that generates both the supply voltage and the clock for the circuit.

During operation, input data are buffered in the FIFO until it is needed by the DSP. The control loop controls the processing rate to avoid queue overflow and underflow: as the queue fills up the clock speed increases to cope with the higher workload, and as the queue empties the clock slows. Thus, the FIFO acts as both a buffer for the data and as the workload-averaging mechanism. This control loop, which keeps the FIFO from overflowing by varying the processing rate forms the "outer" control loop.

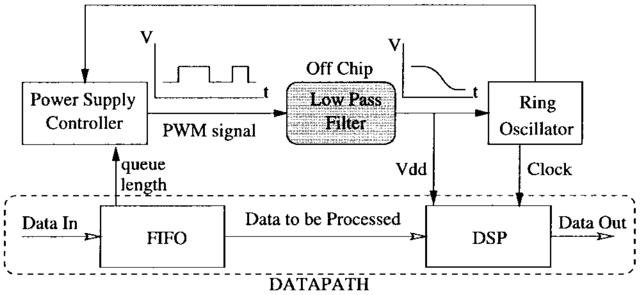

Fig. 16. Dynamic supply voltage chip block diagram.

A second or "inner" control loop is formed between the power supply controller and the ring oscillator. This is the quasi-PLL that adjust the voltage steps in the controller to match processing rates. It is in this loop that temperature and processing variations are measured and corrected.

2) Supply Control: The supply controller has several functions: it assigns an operating rate based on workload; it translates that desired rate into a digital word proportional to the voltage level; and finally it generates a pulse width modulated signal that is filtered and buffered off chip and fed back as the supply voltage for the DSP and ring oscillator.

a) Setting the desired rate and voltage: The desired rate is periodically updated based on the number of words in the queue. The clock speed is averaged implicitly by considering the queue length in the FIFO as the processing rate.

The desired processing rate is translated to a supply voltage by means of a lookup table. The desired rate forms the address, and the 4 bit output gives the fraction of the battery voltage needed to achieve the desired rate. The current table entry is updated at the end of each sample period by comparing the number of actual clock edges to the desired number, so the voltage levels track temperature and battery voltage (as well as process variations).

The 4 bit word output from the register file is converted into a pulse width by means of a counter. A minimum dc level is established by using a 5 bit counter so that on every cycle the output pulse would be high for several clock periods plus the number of periods specified in the lookup table.

3) DSP: The only function of the DSP on the test chip is to emulate the throughput and timing constraints of a general variable-workload signal processor. This functionality was implemented as two separate parts.

Any algorithm that terminates on a data-dependent condition generates a variable workload. For example, an algebraic approximation iteration that terminates when the desired precision is reached, or a video decompression procedure that processes data until it finds an end-of-frame marker; both appear as variable workload algorithms. This chip uses a counter: it counts from zero until the count matches the input data. When the count matches, the data word is considered processed, and another word is fetched from the FIFO. To verify that the right words are being fetched and executed, the counter state is an output of the chip.

The block diagram of the DSP is shown in Fig. 17.

The critical path consists of strings of inverters placed between the counter and output latches. The four delay lines

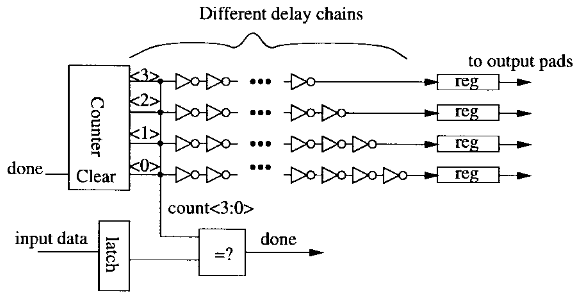

Fig. 17. DSP block diagram.

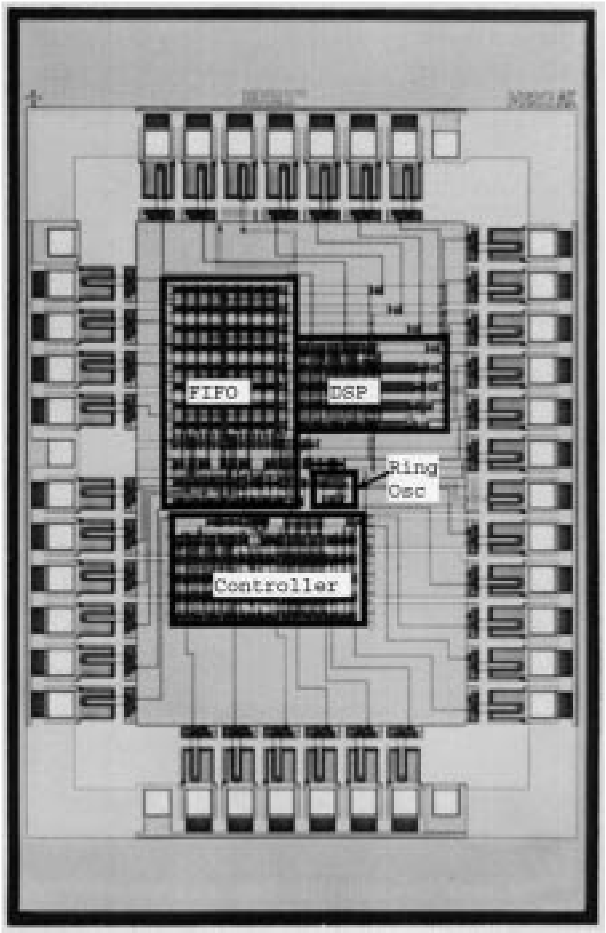

Fig. 18. Variable supply controller chip plot.

actually have different numbers of inverters. The difference allows calibration of the safety margin available in the clock cycle: if all delay lines shorter than a cutoff length consistently latch correctly and all longer lines do not, the clock is correctly tuned to model delays of the cutoff length. If errors are intermittent, the delay is not well matched.

B. Test Results

The chip was fabricated in a 0.8 μm CMOS process. The final layout consumed 2.2 mm × 3 mm; the area was determined by the 40-pad pad ring. The chip photo is reproduced in Fig. 18.

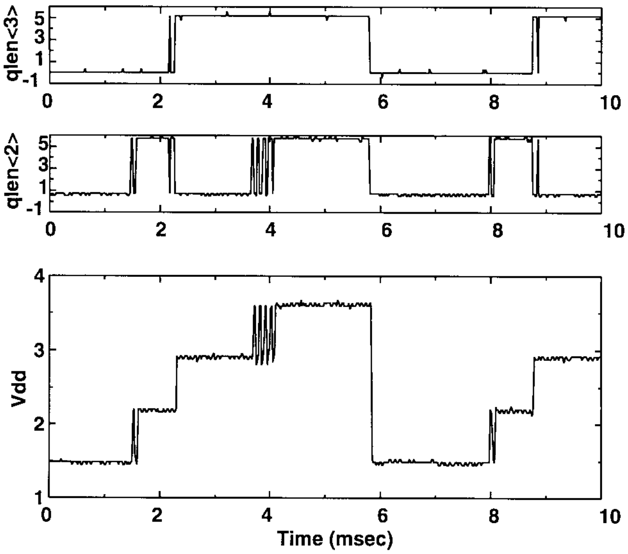

Fig. 19. Measured voltage levels.

The basic functionality of both the DSP and the control loop has been verified. Fig. 19 shows measured waveforms for the two bits of the queue length and the supply voltage. The top two bits of the queue length are used to determine what rate to process at. As the figure shows, when the queue is full, the controller raises voltage to a level determined by the delays, and as the queue empties the voltage drops in discrete levels.

A timing diagram is shown in Fig. 20. In this case the **dataout** bus shows the un-delayed outputs of the DSP counter and the **lowbits** bus shows the contents of the registers that latch the same output after the delay chains. The key observation is that the highest bit of **lowbits**, the one that goes through the longest delay chain, fails to meet the timing requirements—each of its transitions is clocked one cycle late. The three other bits are correctly latched every time, even as V_{DD} rises and the clock speeds up; clearly, the ring oscillator tracks the delay correctly.

C. Overhead

Of course, the power savings possible by varying the supply voltage must be compared to the overhead, in terms of both power and area, introduced by the controller. Since this controller was designed to be dynamically reconfigurable to test the theory presented above, the power and area were not fully optimized. It uses standard cells to minimize design time. Nevertheless, because of the very coarse quantization needed and the simplicity of the controller, the overhead is quite small. The entire controller fits into 0.4 mm^2, with the pulse width modulator taking roughly 0.09 mm^2 of that in a $1u$m-drawn process.

Power numbers are similarly small. Even when running at 40 MHz and 2 V the controller draws just under 0.5 mA, for a tear loss of about 1 mW. A modern, low-power DSP core like the one presented by TI at ISSCC ‘97 [11] may run at power levels as low as 15–20 mW, so the controller would increase power by 5%; more traditional DSP’s, running at 1 W or higher power levels, would hardly notice the 0.1%

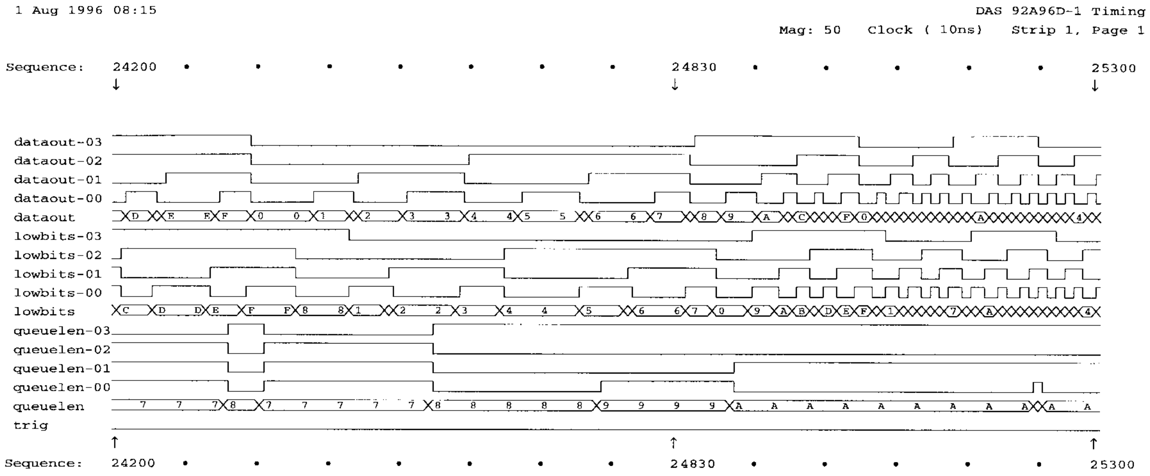

Fig. 20. Measured timing verification.

increase. In both cases, even a modest amount of workload variation would overwhelm the loss.

VI. Application Example: Video

A. Slowly Varying Scene

Digital video is in popular signal processing. Since the video is coded differentially, one measure of the computation required by a chip decoding the video stream is the number of bytes sent per frame. Fig. 21 shows a graph (normalized to full frame = 1) of the number of bytes per frame. Notice the initialization frame at the beginning of the sequence. For a fixed supply system, the energy to process the sequence is just the sum of the normalized workloads, which comes out to 66.4 for this sequence. An (unbuffered) variable supply system would give an energy of 29.7. In this case, with relatively rare updates and a high fraction of the energy going to average frames, little more can be saved by averaging; in fact, even if the entire sequence could be stored and computed at an average rate, the power is not much lower: 28.3.

B. Compressed, Bursty Data

Video compression algorithms use a host of signal processing tricks to reduce the average number of bits that must be sent to encode an image. Fig. 22 shows the output of one such algorithm operating on another video sequence. Note the frequent re-initialization frames required by the algorithm, and the much lower average workload. The linear sum of the workloads is 32, while the variable-voltage sum is 20.2. If the sequence is averaged, however, the total energy falls to just 4.6, a factor of seven smaller!

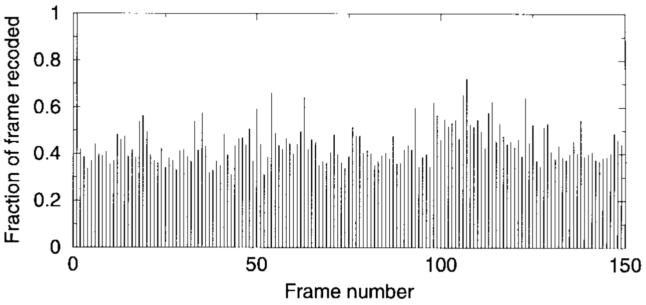

Fig. 21. Workload versus frame number, MPEG.

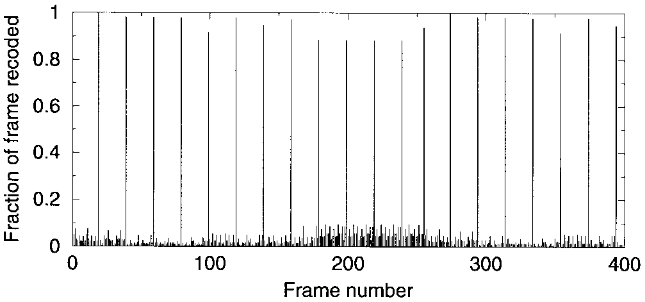

Fig. 22. Workload versus frame number, forced initializations, and compressed video.

VII. Conclusions

For applications where workload changes with time, the power consumed by the DSP can be lowered by varying supply voltage. Video and communications IC's are particularly likely to benefit from variable voltage supplies. The possibility of varying voltage dynamically may allow some fixed-workload algorithms to be replaced by variable-workload algorithms with lower mean computation requirements.

Where latency and extra buffering can be tolerated, averaging workload can lower power even further. When the workload can be predicted *a priori* explicit low-pass filtering can be done; otherwise, the data can be queued and processing adjusted on the basis of queue length. Optimum filtering algorithms are difficult to find and dependent on the statistics of the input data, but easily computed approximations exist.

Continuous variation of the supply voltage can be approximated by very crude quantization and dithering. As long as the power supply is damped correctly the voltage transitions do not cause extra dissipation. Since the power and area of the supply control circuitry scales with the number of voltage levels, quantization to only a few bits minimizes overhead.

Finally, the control circuitry developed is applicable to general DSP circuitry. No assumptions have been made about the DSP in the development of the control; the DSP is synchronous. All that is required for the control is that the critical timing path is known.

REFERENCES

[1] J. Ludwig, H. Nawab, and A. P. Chandrakasan, "Low power digital filtering using approximate processing," *IEEE J. Solid-State Circuits*, pp. 395–400, Mar. 1996.

[2] L. S. Nielsen *et al.*, "Low-power operation using self-timed circuits and adaptive scaling of the supply voltage," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 391–397, Dec. 1994.

[3] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–483, Apr. 1992.

[4] V. Gutnik, "Variable supply voltage for low power DSP," M.S. thesis, Massachusetts Inst. Technol., Cambridge, 1996.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley Series in Telecommunications, 1991.

[6] A. W. Drake, *Fundamentals of Applied Probability Theory*. New York: McGraw-Hill, 1988.

[7] P. Macken *et al.*, "A voltage reduction technique for digital systems," in *Tech. Dig. Papers, IEEE Int. Solid-State Circuits Conf.*, Feb. 1990, pp. 238–239.

[8] G. Wei and M. Horowitz, "A low power switching power supply for self-clocked systems," in *Proc. Int. Symp. Low Power Electron. Design*, 1996, pp. 313–318.

[9] V. R. von Kaenel *et al.*, "Automatic adjustment of threshold and supply voltages for minimum power consumption in CMOS digital circuits," in *IEEE Symp. Low Power Electron.*, 1994, pp. 78–79.

[10] A. Stratakos, S. Sanders, and R. Brodersen, "A low-voltage CMOS DC-DC converter for a portable low-powered battery-operated system," in *Proc. PESC*, 1994.

[11] W. Lee *et al.*, "A 1 V DSP for wireless communications," in *IEEE Int. Solid-State Circuits Conf.*, 1997, pp. 92–93.

Vadim Gutnik received the B.S. degree in electrical engineering and materials science from University of California at Berkeley, Berkeley, in 1994, and the S.M. degree in electrical engineering from the Massachusetts Institute of Technology (M.I.T.), Cambridge, in 1996. He received an NDSEG fellowship in 1994, and the Intel Foundation fellowship in 1997. He is currently working toward the Ph.D. degree at M.I.T., investigating high-speed, low-skew clock distribution.

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 at Berkeley, Berkeley, in 1989, 1990, and 1994, respectively.

Since September 1994, he has been the Analog Devices Career Development Assistant Professor of Electrical Engineering at the Massachusetts Institute of Technology, Cambridge. His research interests include the ultra lower power implementation of custom and programmable digital signal processors, wireless sensors and multimedia devices, emerging technologies, and CAD tools for VLSI. He is a coauthor of the book titled *Low Power Digital CMOS Design* (New York: Kluwer Academic).

Dr. Chandrakasan received the NSF Career Development Award in 1995, the IBM Faculty Development Award in 1995, and the National Semiconductor Faculty Development Award in 1996. He received the IEEE Communications Society 1993 Best Tutorial Paper Award for the IEEE Communications Magazine paper titled, "A Portable Multimedia Terminal." He has served on the technical program committee of various conferences including ISSCC, VLSI Circuits Symposium, DAC, ISLPED, and ICCD. He is the Technical Program CoChair for the 1997 International Symposium on Low-power Electronics and Design and for VLSI Design '98.