An Efficient Controller for Variable Supply-Voltage Low Power Processing

Vadim Gutnik† Anantha Chandrakasan
MIT Microsystems Technology Laboratories

Abstract: A chip has been designed to test the implementation of a PLL-based dynamic supply controller. Separating the rate transitions from V_T and temperature drift improves loop transient response. Buffering and explicit averaging can be used to smooth workload. Active damping minimizes loss in the supply filter due to voltage transitions.

Basic Considerations: There are many applications in signal processing and general purpose processing where the workload varies with time. An approach to reduce the energy consumption of such systems has been proposed which involves the dynamic adjustment of supply voltage based on processing load [1]. The idea is to slow computation and lower V_{DD} during reduced processing loads instead of working at a fixed high voltage and allowing the processor to idle. Even if workload is fixed, energy can be lowered by tracking the process and temperature variations, thus keeping voltage at the lowest level possible [2],[3].

Self-timed circuits have been suggested, but the switched capacitance overhead due to dual-rail coding can be substantial [1]. Our approach uses standard *synchronous* design with a ring-oscillator based clock and simple switching supply (Fig. 1). The clock frequency is controlled by a digital phase-locked loop with V_{DD} as the control signal for the ring-oscillator. If the ring oscillator is well matched to the circuits, logic should function properly, since logic transitions take place in nanoseconds, while voltage changes in microseconds for typical switching supplies.

Splitting the PLL: The LC output filter of the power supply makes the system PLL third-order, and since the input to the LC is pulse-width modulated, standard compensation schemes just lead to feed-through of the ripple. The PLL can be stabilized by cutting loop bandwidth, but system efficiency drops: power savings are possible only if the power supply bandwidth is comparable to that of the processing load. Fig. 2 shows *power-usage vs. update-time* simulation results for one scenario - a 16-element Poisson queue, with service rate (*i.e.* operating voltage) updated at rate $1/T$ by a controller that allows no more than a given rate of overflows. For large T, the processor cannot take advantage of reduced processing loads and works at fixed voltage; at small T, the processor changes levels optimally.

Since the PLL bandwidth constrains T, cutting bandwidth limits power savings. Our solution is to split the feedback loop into two parts: a *Load* part that handles large frequency steps when the computational load changes, and a *Tracking* part (much like the system suggested in [2],[3]) that adjusts the placement of the steps to compensate for process and temperature drift. A lookup table records the voltage needed for each processing rate, and the feedback slowly adjusts the entries in the table as needed. Thus the loop tracks process and temperature, but computation rate is changed open-loop by reading different table entries.

Number of Levels: Fortunately, implementing the lookup table is not difficult. First-order MOS models show that the energy to process a block of data is given by

$$E = CV_0^2 x\left(\frac{x}{2} + \frac{V_T}{V_0} + \sqrt{\left(\frac{x}{2}\right)^2 + 2x\frac{V_T}{V_0}}\right)^2$$

where C is the average total switched capacitance, x is the fractional workload for the block, and V_0 is a constant related to the supply voltage: $V_0=(V_{DD} - V_T)^2/V_{DD}$. (At fixed voltage, energy is directly proportional to workload). Fig. 3 shows a plot of E *vs.* x for four cases: fixed supply, infinite levels, and two plots for a four-level system. If exactly one block is processed per sample time, the processor will idle for part of the cycle unless the computation happens to fill it exactly - curve [c]. If the processing of one data block can cross sample-time boundaries (*i.e.* the data is buffered) the processor will never idle: whenever the actual workload happens to be between two fixed levels, some blocks will be processed at the higher rate and some at the lower, and the average power is given by the weighted sum of the two levels - curve [b]. It is clear that even rough quantization is sufficient to get power savings.

Controller Schematic: The lookup table is implemented as a 4 word * 4 bit register file, and its entries are duty cycles for the switching supply. The controller, shown in Fig. 4 is almost a phase-locked loop; in fact, it acts as four loops in parallel, and the desired workload determines which is active. Whichever of the effective loops is on locks to its desired rate.

Operation is simple -- the counter counts for a time given by an external *Ref_CLK* signal; that count is subtracted from the desired frequency (i.e. workload), and the error is converted to an error voltage. If the desired frequency and *Ref_CLK* are scaled appropriately, A can be set to unity, as is done in the design.

Buffering: As previously stated, buffering the data improves performance by allowing voltage level dithering. Buffering over multiple samples also smooths the computational load. This improves power supply performance, and lowers the energy required by the processor for the same reason that varying voltage saves power -- working at an average rate is more efficient than speeding up and slowing down. In cases where the computation required for buffered blocks is known, any FIR filter (with coefficients summing to 1) can be used to smooth the load. Fig. 5 shows voltage requirements for a variable-voltage MPEG decoder with and without buffering. Although buffering lowers the power, and there are some cases where buffering is necessary (e.g. variable-rate packet arrival over a network), buffering is not inherent in this synchronous scheme the way it is in the self-timed approach.

Active Damping: The damping of the supply output filter presents a problem for variable-voltage supplies not present in fixed voltage systems. In low-power applications, volume constraints preclude impedance-scaling the inductor to match the load. A resistor can be inserted in series with the capacitor C_F to provide damping, but it can burn significant energy ($.236\ C*\delta V^2$) at voltage transitions. Active damping can eliminate this power loss at the expense of a slow, low-precision A/D (Fig. 6). Simulation shows that even very simple active damping will be sufficient (Fig. 7).

Simulation Results. A chip has been designed for a MOSIS 1.2μm process to test the controller; mixed-signal simulation results are shown in Fig. 8. The supply voltage waveform shows the start-up transients (since the register file is cleared at start-up) and the levels settling to steady state. This chip uses the top two bits of the queue length to choose the processing rate. In general, the optimal workload filter is nonlinear and depends on the processor. Regardless of the filter, however, the controller described above will adaptively generate the correct voltages for the prescribed workload.

References.

[1] L Nielsen, C. Niessen, J. Sparso, K. van Berkel, "Low-Power Operation Using Self-Timed Circuits and Adaptive Scaling of Supply Voltage," IEEE Transactions on VLSI systems, December 1994, pp 391-397.
[2] P. Maken, M. Degrauwe, M. Van Paemel, H. Oguey, "A Voltage Reduction Technique for Digital Systems," ISSCC February, 1990.
[3] M. Horowitz, "Low Power Processor Design Using Self-Clocking," Workshop on Low-power Electronics, 1993.

† This research funded by ARPA Contract # DAAL01-95-K-3526. Vadim Gutnik is supported by an NDSEG fellowship.

 0-7803-3339-X/96/$5.00©1996 IEEE

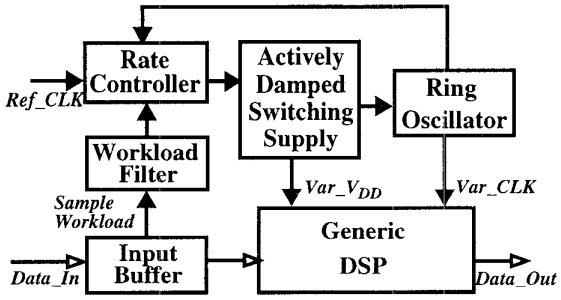

Figure 1: System Control and Data Flow Diagram

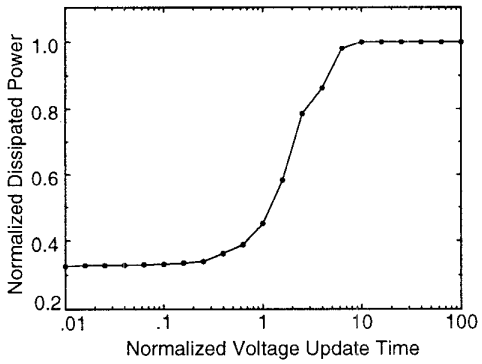

Figure 2: Processor Power vs. Voltage Update Time

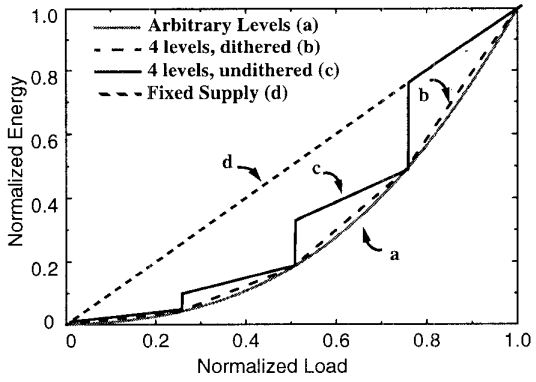

Figure 3: Energy Usage vs. Load

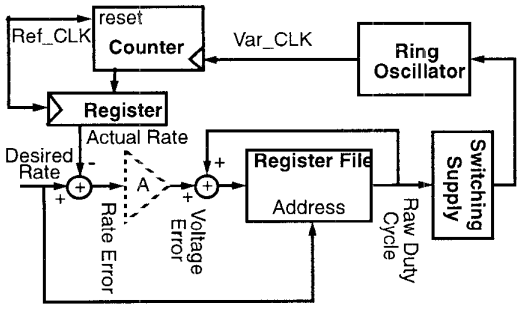

Figure 4: Controller Schematic

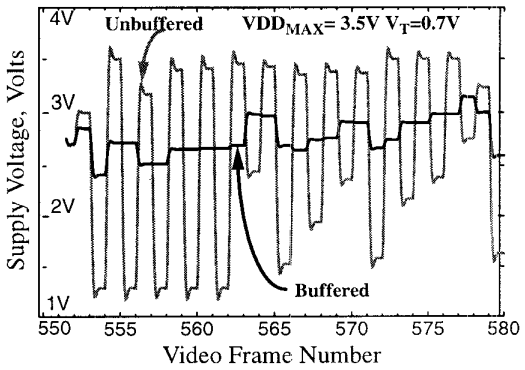

Figure 5: V_{DD} vs. Frame # for MPEG2 Decoder

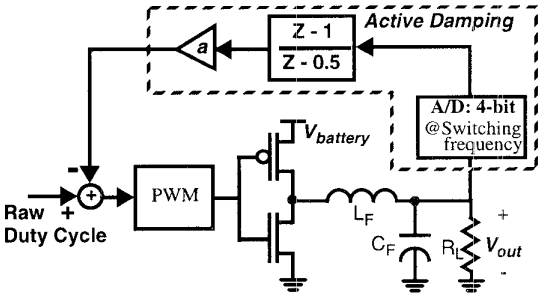

Figure 6: Active Damping

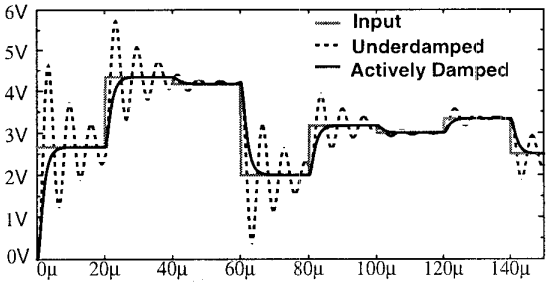

Figure 7: Simulated Supply Transient Response

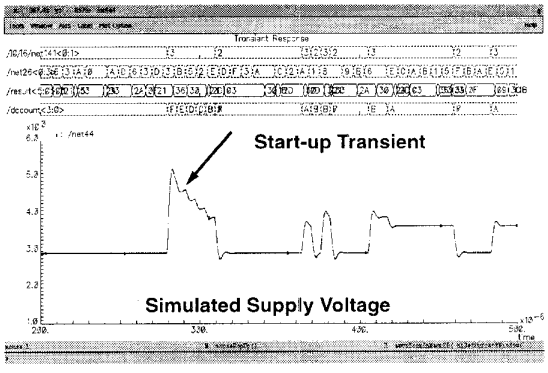

Figure 8: Mixed Signal Simulation Results