Quality-Driven Proactive Computation Elimination for Power-Aware Multimedia Processing [*]

Shrirang M. Yardi, Michael S. Hsiao, Thomas L. Martin and Dong S. Ha
{*yardi, mhsiao, tlmartin, ha*}@*vt.edu*
The Bradley Department of Electrical and Computer Engineering, Virginia Tech
Blacksburg, VA, 24061, USA

Abstract

We present a novel, quality-driven, architectural-level approach that trades-off the output quality to enable power-aware processing of multimedia streams. The error tolerance of multimedia data is exploited to selectively eliminate computation while maintaining a specified output quality. We construct relaxed, synthesized power macro-models for power-hungry units to predict the cycle-accurate power consumption of the input stream on the fly. The macro-models, together with an effective quality model, are integrated into a programmable architecture that allows both power savings and quality to be dynamically tuned with the available battery-life. In a case study, power monitors are integrated with functional units of the IDCT module of a MPEG-2 decoder. Experiments indicate that, for a moderate power monitor energy overhead of 5%, power savings of 72% in the functional units can be achieved resulting in an increase in battery life by 1.95×.

1 Introduction

The main challenge facing designers of battery-operated devices is guaranteeing battery-life for processor-intensive applications like portable video, audio, streaming multimedia and speech synthesis while providing the required Quality of Multimedia Data (QoMD) to the end-user. Both general-purpose and application-specific portable devices are becoming increasingly popular mainly due to their ability to support such applications. Some of the characteristics exhibited by such workloads include: (i) heavy, iterative computation using power-hungry arithmetic units, (ii) need for real time processing of the input streams, and (iii) tolerance to errors in computation that go unnoticed by human visual/auditory systems.

The property of error tolerance has been widely exploited for power and area optimization of multimedia processing hardware. Examples include use of fixed point arithmetic units [1], reduced bit-width floating-point units [2], application-specific enhancements to the Discrete Cosine Transform (DCT) and Inverse DCT (IDCT) implementations [3] and reformulation of canonical transform algorithms [4]. Recently, battery-driven power management techniques were proposed in [5, 6] that specifically target battery lifetime extension in addition to reducing average power consumption. Most of these techniques provide a trade-off between the QoMD and power consumption by exploiting the limitations of human sensory systems.

1.1 Our Contributions

In this paper, we propose an architectural-level methodology for energy-efficient processing of multimedia streams by dynamically adjusting the number of computations performed during decoding. We take advantage of the error tolerance of multimedia streams to proactively eliminate some computations, thus *intentionally introducing errors* during decoding, but without heavily degrading the output quality. Our key observation is that, unlike conventional techniques, relaxing the guarantee of accurate computation leads to higher power savings by trading off the QoMD. We outline a novel technique in which the power consumption of the input stream is dynamically predicted on a per-computation basis by monitor circuits, termed as *power monitors*. We use *synthesized and relaxed power macro-models* of selected power-hungry functional units as their power monitors. The degradation in playback quality is controlled by a multimedia-specific quality model designed to ensure reduced power consumption while allowing graceful degradation in quality. The power monitor and the quality model are integrated into a programmable architecture that enables the system to proactively eliminate computations in a controlled manner that is transparent to the user. Opportunities for such proactive adaptation exist in scenarios where accuracy may not be of the utmost importance when either the battery is running low or when the channel quality is high for streaming multimedia.

We explore the power-performance trade-offs of our technique by designing power monitors for a single-precision floating point unit (FPU) and integrating them in the IDCT module of a MPEG-2 decoder. Although we consider only a 32-bit FPU as a case study, our technique is general enough to be applied to other functional units like fixed-point or reduced precision FPUs in both, application-specific and general-purpose processors. We evaluate the energy- and battery-efficiency of our approach by experimenting with different power monitor and output quality settings. Our experiments indicate that power savings up to 72% can be achieved in the IDCT module for a power monitor overhead of only 5% in the total energy. We also demonstrate that these savings are complementary to techniques like dynamic

[*]supported in part by NSF under grants 0219801 and 0121416

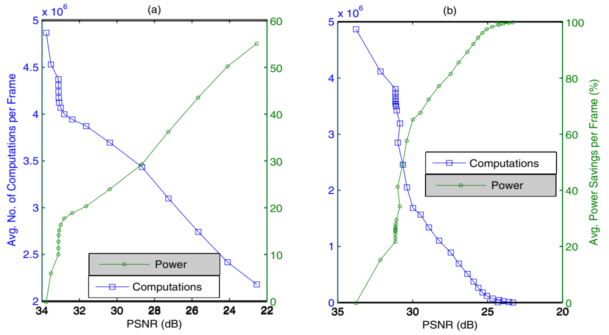

Figure 1. Effect of computation skipping on PSNR and Power (a) I-Frames, (b) P-Frames

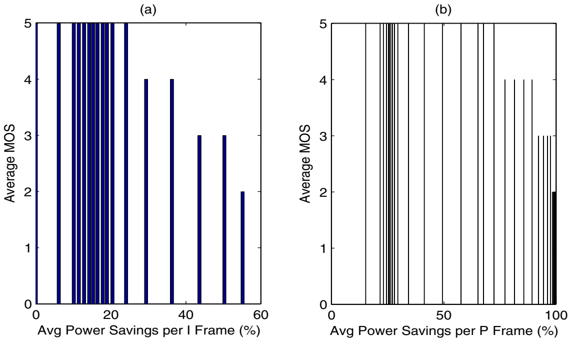

Figure 2. Effect of computation skipping on MOS (a) I-Frames, (b) P-Frames

voltage scaling (DVS) and dynamic power management (DPM).

1.2 Related Work

Low power design for multimedia systems is currently a very active field of research. In addition to circuit-level optimizations, techniques like DVS [7–9], DPM [10] and power-aware scheduling of multimedia tasks [11] have also been proposed.

Architectural enhancements for multimedia processing include use of execution caches [12], memoing [13] and architectural adaptations for general-purpose systems [14]. Execution caches and memoing techniques take advantage of the data value locality in multimedia streams to use previously computed results that are stored in a caching structure. When the input operands exactly match those seen previously, the result from the cache is used instead of computing it. Although our technique bears some similarity to this approach, the key differences are (i) instead of a caching structure, we use a power monitor that predicts the power consumption rather than a numerical value, (ii) we relax the guarantee of accurate computation while maintaining a pre-defined output quality, (iii) we use relaxed power macro-models as monitors that have a smaller overhead than the direct-mapped caches proposed in [12].

Micro-optimization of FPUs was proposed in [15] as a method to reduce the operation count by breaking the floating-point operations into their constituent micro-operations and optimizing the resulting code by eliminating, combining or overlapping different micro-operations. On the other hand, our technique provides a way to *turn-off the entire FPU* as opposed to some micro-operations thus providing greater power savings.

Recently, DPM policies that reduce output quality to specifically target improvement in battery lifetime were proposed in [5]. However, their method is a reactive one as it triggers a reduction in quality only after the battery voltage falls below a certain threshold. On the other hand, our method tries to proactively regulate the system power profile by using the information provided by power monitors. Our results show that such an *early action* approach leads to higher battery-efficiency.

2 Motivation

An MPEG video stream consists of a series of frames that are of three types: I-Frames (intra-coded), P-Frames (Predictive-coded) and B-frames (Bi-Directional coded). I-frames can be decoded independently while P- and B-frames require either the previous, future or both frames for decoding. Of the several steps required to decode a frame, the IDCT and Reconstruction steps are the most time-consuming and IDCT is the most computationally intensive [16]. The output quality of a MPEG decoder is measured objectively using the Peak-Signal-To-Noise-Ratio (PSNR), calculated with respect to reference frames at the encoder input, and subjectively using the Mean-Opinion-Score (MOS) where quality is assigned a score between 1-5 by human subjects. MOS of 4-5 represents “GOOD”, 2-3 means “FAIR” and a score of 1 is “POOR” or unacceptable.

Figures 1 and 2 show the relationship between the average number of computations performed by the IDCT unit *per frame* with the PSNR and MOS for 100 frames. Note that skipping is performed at the operation granularity rather than at frame granularity. Figure 1(b) shows that, when 88% of the computations are skipped in a P-Frame, the PSNR falls to roughly 24dB which corresponds to a MOS value of 3 (from Figure 2(b)). This indicates that the output is still of acceptable quality. This is also true of I-Frames where 55% computations can be skipped before MOS drops to 3 (Figure 1(a), 2(a)). Since the functional units are gated off when computations are skipped, this provides a power savings of 77% while maintaining acceptable quality.

A common feature of all techniques mentioned in Section 1.2 is that they guarantee accurate computation for every operation (and instruction). In our approach, we relax this guarantee and instead, intentionally introduce errors by selectively skipping computations. To prevent excessive quality degradation due to the accumulative nature of such errors over successive frames, we bind the number of skipped computations by a quality model. The proposed quality model controls the error propagation by quantifying the playback quality and using this value to direct the power monitors to either skip or execute a particular computation. The key difference between our approach and previous techniques is that instead of trading off power with a fixed, low quality level, our proposed architecture can be *dynamically tuned* to fit **both** the quality and power requirements of the system.

3 Methodology

Our methodology consists of three phases: (i) design of an effective power macro-model, with small area and power overhead when synthesized, to predict the power on a per-computation ba-

Table 1. Comparison of Full and Relaxed Models of the Power Monitors

	Full Model			Relaxed Model			
Circuit	No. of Terms	CPE (%)	R	No. of Terms	CPE (%)	R	Power Savings(%)
c2670	233	14.37	0.69	16	15.13	0.65	90.34
c5315	178	9.84	0.81	16	11.31	0.73	87.60
c7552	207	10.32	0.94	16	10.86	0.93	89.23
FP Mult	50	17.76	0.47	20	18.26	0.44	45.47
FP Add	58	29.21	0.89	28	31.70	0.88	52.65

sis, (ii) design of a quality model that quantifies the output quality of the target application and controls on/off status of the monitored units, and (iii) a programmable architecture that integrates the power monitor and the quality model to enable dynamic control of the number of computations. In what follows, we describe each phase in detail.

3.1 Macro Model Construction

The application of a power macro-model as a small synthesized circuit presents unique challenges for its design. Specifically: (i) the model should be able to track the changes in power consumption in a cycle-accurate manner with low area and power overhead, (ii) the only information available to the model is the transition activity on the inputs of the selected components, and (iii) the operation of the model should be controllable by the quality model. For design of cycle-accurate macro models, we use the approach outlined in [17] in which the *transitive fanout* correlated input switching activity is used to construct a regression model for power estimation. However, unlike [17], we assume that all the inputs in the model are uncorrelated to reduce the model complexity. Further, we use a backward elimination algorithm as opposed to the forward elimination procedure proposed in [17].

The macro-model approximates the power consumption, P, as a linear function of the Hamming distance between a pair of input vectors, I_{c-1} and I_c, in clock cycles $c-1$ and c respectively. The model is obtained using multiple regression over the *input transition vector*, in which a 1 represents a transition on the corresponding input pin. The model thus obtained is termed as the *full model* and contains $k+1$ terms where k is equal to the number of primary inputs (PIs) of the selected component.

For large circuits, the synthesized full model results in a circuit with a large area and power overhead as compared to the monitored component itself. To reduce this overhead, we *relax the full model* by eliminating those terms from the full model that have the *least impact* on the power estimate in any given clock cycle. The relative impact of each term on the power estimate is calculated by the F^*-test statistic [19] based on hypothesis testing. We use a procedure called *backward elimination* that starts with the full model as input and determines the *level of significance* of every term, obtained from the F^*-distribution, for a given confidence level (95% in our case). The terms that have a F^* value below some threshold are eliminated and the model is re-fitted with the remaining terms. In our case, the threshold was selected so as to sacrifice some accuracy in order to decrease the size of the macro model. The algorithm terminates when the goodness of fit, R, for the remaining terms falls below a certain value or the per-cycle estimation error, *CPE* [17], increases above a threshold. On termination, the remaining inputs are weighted with their final F^* values.

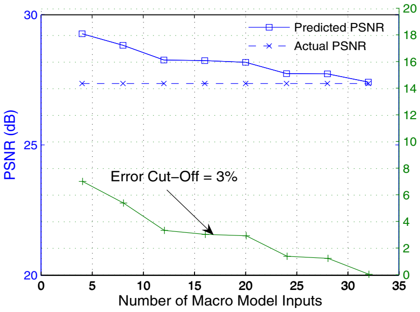

Figure 3. Prediction Error vs. Macro Model size

We tested the models on some large benchmark circuits and the mantissa datapaths of 32-bit floating-point multiplier and adder. We did not target the exponent datapath since their small bit-widths resulted in macro-models with excessive overhead even after relaxation. The regression coefficients were calculated using a training set of 5,000 vectors extracted from gate-level simulation of each circuit. For complex designs with multiple components like the mantissa datapaths, gate-level simulation was performed in context of the RTL simulation of the entire FPU, as outlined in [18], allowing us to extract application-specific vectors. The fit was validated using 50,000 randomly generated vectors for the benchmark circuits and from actual MPEG-2 bitstreams for the FPU. Actual power consumption values were obtained using a gate-level power estimator. For the elimination algorithm, we selected $R = 0.4$ and $CPE = 35\%$ as the terminating conditions by trial-and-error.

Table 1 provides a comparison between *CPE* and *R* values for the full and relaxed models. The increase in *CPE* values over the full model was observed to be less than 3% while the power consumption as compared with the full model was reduced by as much as 90%. The key observation is: given a Hamming distance threshold, the relaxed models can be used to *differentiate vectors that consume higher or lower power than the threshold by monitoring transitions on the selected PIs* even for complex designs. We found that the false negatives and positives during this differentiation were less than 4% for all the circuits. Since the ability of the macro-model to differentiate the input vectors directly affects the output quality, we explored the effect of different macro-model sizes on the PSNR of several decoded videos. Figure 3 compares the actual measured PSNR with that obtained by skipping computations as directed by macro-models of different sizes for the FPU. Based on this, a macro-model size that yielded a prediction error of less than 3% was selected as a trade-off between prediction accuracy and area/power overhead.

3.2 Selection of a Quality Model

To prevent a runaway degradation in output quality due to error propagation over successive frames, a quality model is needed that binds the degradation in quality (PSNR/MOS in our case) with a specific number of skipped computations. Further, the model should be able to provide directives to the power monitor to guide differentiation of input vectors. In this sub-section,

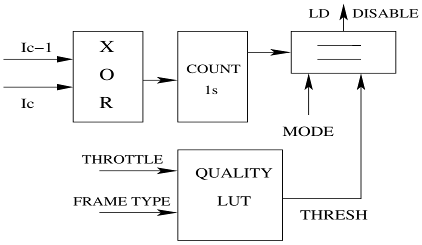

Figure 4. Power Monitor Architecture

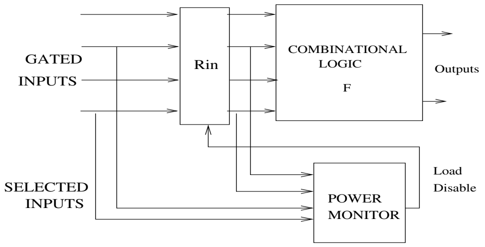

Figure 5. System Model with Power Monitor

we outline the design of such a model with MPEG-2 decoding as the target application.

Previously in Figure 1, we illustrated the relationship between PSNR and the number of computations performed in the IDCT unit. Based on this observation, we constructed a regression model of the PSNR as a function of the number of computations performed. The goodness of fit, R, for the model was observed to be 0.91 for I-frames and 0.89 for P-frames respectively. Further, as explained in Section 3.1, the classification of input vectors by the macro-model is programmable by a threshold value of the Hamming distance between successive input vectors. Hence, the number of computations, and consequently the quality can be controlled by using different thresholds.

Table 2. Quality Model for FP MULT: P-Frames

Threshold (%)	PSNR Difference(dB)	Computations Skipped(%)	MOS
10	0.62	15.4	5
20	0.64	26.6	5
30	0.92	41.5	5
40	2.85	72.5	4
50	5.84	92.3	3

We have designed such a quality model as a small look-up table (LUT) through extensive experimentation using our software prototype. Table 2 provides an example of a LUT for P- and B-frames. The first column lists the output of the model, *i.e.* the threshold values, as a fraction of the total Hamming distance of the monitored component. The second column lists the loss in quality, in terms of the difference in PSNR with and without skipping, when input vectors below the specified threshold were skipped. The third column lists the fraction of input vectors for which the monitored unit can be disabled and the last column lists the average MOS values for the decoded streams. The strength of this simple yet effective quality model lies in the fact that it allows the quality to degrade very slowly even for a large number of skipped computations. Further, in our experiments, we found that the computational workloads vary according to the frame types. Hence, different quality LUTs are required for I-frames and P-, B- frames. However, as will be seen in Section 3.4, the size of each table is small enough to be included in hardware with small area and power overhead.

3.3 Analysis of Power Savings

In this subsection we provide a quick analysis of the potential power savings due to the proposed architecture by using a floating-point multiplier as an example. The aim is to provide the designer with some parameters to explore the design space for the selected component and application.

The total power consumption due to the addition of a power monitor can be modeled as,

$$P_T = P_{mon} + (1 - R_s) \times P_{mul} \tag{1}$$

where, P_T represents the total power, P_{mon} is the power overhead of the monitor circuit, P_{mul} is the power consumed by the multiplier and R_s is the *skip-rate* for the multiplier, defined as the fraction of time that the monitor circuit is able to disable inputs to the multiplier. Note that R_s depends on the threshold value provided by the quality model and the ability of the macro-model to correctly differentiate the inputs. As we address only dynamic power consumption in this analysis, the savings in power consumption is given by,

$$P_s = [R_s - (P_{mon}/P_{mul})] \times 100\% \tag{2}$$

The above equation can be used to explore the design space for both the monitor circuit and the quality model. Specifically, a quality model that allows substantial number of computations to be skipped (high R_s) for only a small quality degradation is preferable. From Table 2, it can be seen that our quality model satisfies this criteria. Similarly, a monitor circuit with a low P_{mon}/P_{mul} is desirable. It should be noted that these two objectives are conflicting: a smaller monitor circuit will have higher *CPE* which may lead to imbalances in the quality model and a restrictive quality model may result in a large monitor circuit. Hence, trade-offs can be examined depending upon the selected component and application using equation 2.

3.4 Monitor Circuit Architecture

We integrate the macro-model and quality model designs of Sections 3.1 and 3.2 into a programmable architecture for the power monitor. Figure 4 illustrates the architecture while Figure 5 shows how the monitor can be integrated in a complex system.

A bank of XOR gates is used to calculate the transition activity vector among participating bits between I_{c-1} and I_c. The Hamming distance is then determined by a small 1's-counter circuit. The monitor can be programmed using the signals *MODE*, *FRAME TYPE* and *THROTTLE*. Specifically, *MODE* is used to select between skipping of predicted low power and high power vectors. The *FRAME TYPE* and *THROTTLE* signals are used to index the proper quality model LUTs that yield the value of *THRESH*. The calculated Hamming distance is compared with this value and, depending upon the *MODE*, the decision to skip or execute the computation is made by driving the *LD DISABLE* signal to the proper value. If the computation is to be skipped, then the inputs to the monitored unit are disabled to save power and previously computed outputs are maintained.

The *THROTTLE* signal is used to control the skip-rate by switching between different quality levels. This can be a function of one or several system parameters. In Section 4, we provide an example where this signal is derived from the sensed battery voltage. As battery voltage decreases by a certain amount, *THROTTLE* is used to decrease the quality to provide a graceful degradation while prolonging the battery life.

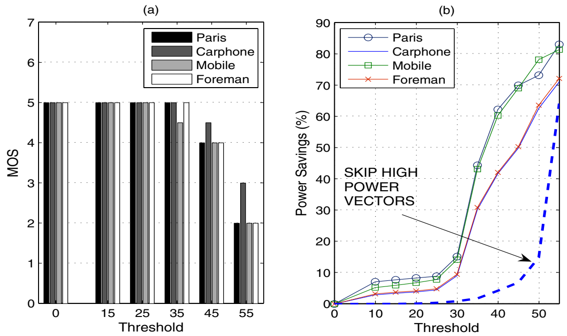

Figure 6. Energy-Performance Trade-Off

The *MODE* signal is used to provide a fine-grained control over both the power and output quality. We consider a case where a system is under a denial-of-service power attack [20] that is trying to rapidly drain the battery. In such a case, the *MODE* can be set to skip all the high power vectors so as to prolong battery life as much as possible.

This simple architecture effectively integrates the macro-model and quality models to allow fine-grained architectural-level control over the power-hungry units. The architecture can be programmed using a very few signals and consequently, can be easily integrated into a general-purpose or application-specific system aimed at multimedia workloads. The monitor circuit was designed in VHDL and synthesized in Synopsys using the TSMC 0.18 μm standard cell library. A 20-entry quality LUT was used with 10 entries for each frame type. The area of the resulting circuit was 8715 μm^2 while the power consumption measured with typical MPEG-2 bitstreams was 144 μW. The delay in the critical path due to addition of the monitor was $2ns$. The overhead of the monitor circuit was 12% in terms of area and 5% in terms of power as compared to the FPU.

4 Experimental Validation

Setup: Matched software and hardware prototypes of the proposed architecture were designed in C and VHDL for single-precision floating point multiply and add units. All bit-level hardware operations were emulated in software. Input data for power estimation was obtained from benchmark MPEG-2 streams of varying bit rates and motion characteristics. Battery lifetime estimation was performed using the PSPICE model of a Lithium-Ion battery [21] with standard capacity 1.25 Ah and fully charged voltage of $4.1V$. Battery efficiency was measured in terms of the residual battery charge at the end of a particular simulation run. Current values required for battery simulation were obtained by dividing the power profile with the rated battery voltage and were averaged over a window of $10ms$ to filter out effect of noise. Energy-efficiency of the approach was measured in terms of the power savings obtained by varying the threshold settings. For quality assessment of the decoded stream, PSNR was calculated using the difference in pixel values with original benchmark video sequences obtained from [23] and MOS values were calculated as the mean of MOS values from 5 individuals, with $MOS = 3$ assumed to be the minimum acceptable quality. Although, both MOS and PSNR were used as quality metrics, we report only the MOS values here due to space constraints.

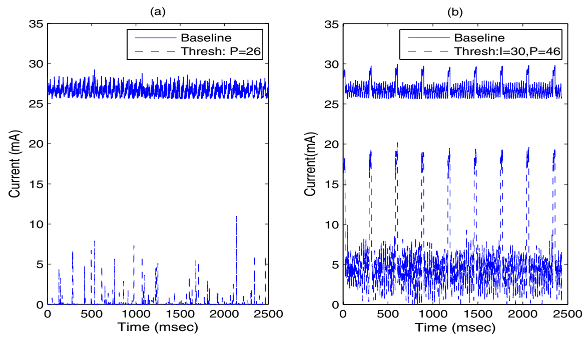

Figure 7. Discharge Current when skipping - (a) high power vectors, (b) low power vectors

Energy Efficiency and QoMD Trade-Off: As a case study, we integrated a single-precision FPU with its power monitors in the IDCT module of a hardware and software MPEG-2 decoder [24]. Instead of targeting a specific IDCT architecture, we focus on the power-hungry FPU since it can be used in either general-purpose or application-specific systems. Here, we present the results for a 32-bit FPU, but the technique is general enough to be applicable to fixed-point or reduced bit-width FPUs as well.

The software prototype was used to decode 4 benchmark videos and functional vectors to the FPU were extracted and provided to a RT-level power estimator. Figure 6 illustrates effect of skipping low power and high power vectors, using several threshold value settings, on the quality (Fig. 6(a)) and energy (Fig. 6(b)) of the decoded stream. The dotted line in Figure 6(b) represents the power savings when skipping high power vectors averaged over all the video streams and the bold lines illustrate the savings when low power vectors are skipped. Average power savings of 68.2% were observed with a gradual degradation in the quality. We observed that fewer high power vectors can be skipped since high switching activity implies more computation and consequently, a higher impact on the output quality. However, average power savings of 55% were observed as illustrated by the dotted line. Based on these experiments, we found that threshold values of 46 for P-Frames and 30 for I-frames resulted in maximum power savings of 72% for a MOS value of 3.

Battery Efficiency: It has been observed that available battery charge diminishes rapidly if the current drawn from the battery is consistently higher than its rated current due to rate capacity effects [6]. Hence, in addition to energy-efficiency, a reduction in current drawn from the battery is required to prolong the battery life. Figure 7 shows a snapshot of the current profile of the baseline and modified IDCT units when (a) skipping high power vectors, and (b) skipping low power vectors. When skipping high power vectors, the current peaks are flattened resulting in a decrease in average current by 98.3%. When skipping low power vectors, current peaks are observed mainly during decoding of I-frames and the average current drawn decreases by 79.2%.

To analyze the effect of our approach on the battery life, we looped a 5000-frame video several times for a total viewing time of 1.5 hours and provided the resulting current profile to the PSPICE battery model. Typical current values due to other units involved in MPEG-2 decoding like dithering, motion estimation and display were added to the current values of the IDCT unit

IEEE COMPUTER SOCIETY

to simulate a realistic multimedia workload [22]. The amount of charge that the battery can deliver is a function of the state-of-charge (SOC), discharge rate and frequency of the discharge current [21]. In [21], the SOC is modeled as voltage across a capacitor, C_{CAP}, where an initial voltage, $V(C_{CAP}) = 1V$ corresponds to a fully charged battery (100% residual charge). At the end of the simulation run, the baseline IDCT unit resulted in a residual charge of 82% ($V(C_{CAP}) = 0.82V$) while the residual charge due to the modified IDCT was 99.8% ($V(C_{CAP}) = 0.998V$): an increase in battery lifetime of 1.22×.

From these experiments, we can make the following observations: (i) relaxing the guarantee of accurate computations results in significant power savings; (ii) power savings are consistent over video streams with different motion characteristics; (iii) threshold values provided by our quality model offer a simple way to dynamically control the playback quality, power consumption and discharge current; (iv) the only system-level support required for our architecture is a stimulus for changing the thresholds during program execution. Hence, our technique is orthogonal to system-level techniques like DVS and DPM and the two can be applied together to further enhance the energy-efficiency. In practice, the quality levels can be decreased in response to values like the available battery SOC to prolong battery life while providing a graceful reduction in quality.

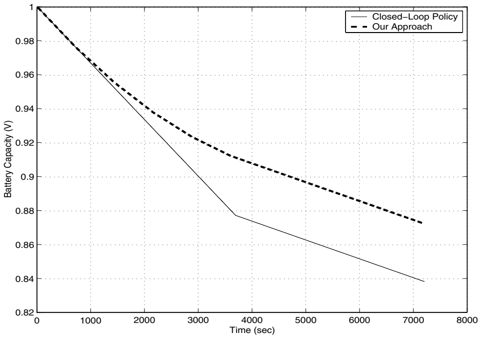

Figure 8. Comparison of trade-off characteristics with Battery-Driven DPM [5]

Comparison with other Power Management Techniques: We compare our approach with the closed-loop policy of the battery-driven DPM system outlined in [5] where the quality of an audio recorder is dynamically adjusted from *Fine Sound* to *Raw Sound* when the sensed battery voltage drops below a certain threshold, V_{th}. We adapt this scheme in our prototypes for comparison using MPEG-2 streams instead of audio. The battery efficiency of [5] depends on the amount of time for which the system can be run in *Raw* mode while providing a graceful quality degradation. This, in turn, depends on the length of time before the supplied voltage reaches V_{th}. Thus the battery-efficiency of the two approaches can be compared by observing the time required for the battery voltage to drop to V_{th}. For the experiment, we assume $V_{th} = 3.6V$, which corresponds to $V(C_{CAP}) = 0.87V$ in the model. The *Raw* quality setting is assumed to have a MOS value of 4. In our prototype, successive quality levels were selected when the sensed battery voltage reduced by $0.1V$. Figure 8 illustrates that proactive computation elimination is superior both in terms of quality and battery efficiency. Specifically, our approach caused V_{th} to be reached only after $7200sec$ as compared to $3690sec$ for the closed-loop policy resulting in a 1.95× increase in battery lifetime. Further, higher output quality can be maintained over a longer time using our approach since MOS decreases to 4 only after $7200sec$. We would like to emphasize that our technique can be applied in addition to the DPM policies like *Sleep* and *Idle* states mentioned in [5] resulting in higher battery-efficiency than these techniques.

5 Conclusion

In this paper, we presented a novel technique that uses dynamic, cycle-accurate power estimation of the input stream to selectively eliminate computation. We demonstrated how our approach can be used to achieve quality-driven, fine-grained control over critical, power-hungry units by using synthesized, relaxed power models as monitor circuits. Our experimental results with the IDCT unit of a MPEG-2 decoder indicate that this approach can successfully exploit the error tolerance of certain applications and dynamically regulate the application power profile to achieve energy- and battery-efficient operation.

References

[1] D. Dobberpuhl, ‘The Design of High-Performance Low-Power Microprocessor”, *ISLPED*, 1996.

[2] F. Fang *et al.*, ‘Lightweight Floating-Point Arithmetic: Case Study of Inverse Discrete Cosine Transform”, *EURASIP J. Sig. Proc.*, 2002.

[3] N. J. August and D. S. Ha, ‘Low Power Design of DCT and IDCT for Low Bit Rate Video Codecs”, *IEEE Trans. Multimedia*, vol. 6(3), 2004.

[4] Z. Wang, ‘Fast Algorithms for the Discrete W transform and Discrete Fourier Transform”, *IEEE Trans. Acoustics, Speech and Sig. Proc.*, vol. ASSP-32(4), 1984.

[5] L. Benini *et al.*, ‘Battery-Driven Dynamic Power Management”, *IEEE Design and Test of Computers*, vol. 18, 2001.

[6] K. Lahiri, A. Raghunathan and S. Dey, ‘Communication Architecture Based Power Management for Battery Efficient System Design”, *DAC*, 2002.

[7] Z. Lu *et al.*, ‘Reducing Multimedia Decode Power Using Feedback Control”, *ICCD*, 2003.

[8] K. Choi *et al.*, ‘Frame-based Dynamic Voltage and Frequency Scaling”, *ICCAD*, 2002.

[9] Y. Lu, L. Benini and G. DeMicheli, ‘Dynamic Frequency Scaling with Buffer Insertion for Mixed Workloads”, *IEEE Trans. Computer-Aided Design of Integrated Circuits*, vol. 21(11), 2002.

[10] L. Benini and G. DeMicheli, *Dynamic Power Management: Design Techniques and CAD Tools*, Kluwer, 1998.

[11] H. Shaoxiong, Q. Gang, S. S. Bhattacharya, ‘Energy Reduction Techniques for multimedia applications with tolerance to deadline misses”, *DAC*, 2003.

[12] M. Azam, P. Franzon and W. Liu, ‘Low Power Data Processing by Elimination of Redundant Computations”, *ISLPED*, 1997.

[13] D. Citron, D. Fietelson and L. Rudolph, ‘Accelerating Multi-Media Processing by Implementing Memoing in Multiplication and Division Units”, *ACM SIGOPS*, 1997.

[14] C. J. Hughes *et al.*, ‘Saving Energy with Architectural and Frequency Adaptations for Multimedia Applications”, *34th MICRO*, 2001.

[15] W. J. Dally, ‘Micro-optimization of Floating Point Operations”, *ASPLOS*, 1989.

[16] K. Patel, B. Smith and L. Rowe, ‘Performance of a Software MPEG Video Decoder”, *ACM Int'l Conf. on Multimedia*, 1993.

[17] Q. Wu *et al.*, ‘Cycle-Accurate Macro-Models for RT-level Power Analysis”, *IEEE Trans. VLSI Systems*, vol. 6, 1998.

[18] A. Bogliolo, L. Benini, and G. DeMicheli, ‘Regression-based RTL Power Modeling”,*ACM Trans. Des. Aut. Electronic Sys.*, vol. 5, 2000.

[19] J. Neter, *Applied Linear Regression Models* , Irwin, 1996.

[20] T. Martin, M. Hsiao, D. Ha and J. Krishnaswami, ‘Denial-of-Service Attacks on battery-powered mobile computers”, *IEEE Int'l Conf. Pervasive Computing and Communications*, 2004.

[21] S. Gold, ‘A PSPICE macro-model for lithium-ion batteries”, *12th Annu. Battery Conf. Applications and Advances*, 1997.

[22] S. Chakraborty and David K. Y. Yau, ‘Predicting Energy Consumption of MPEG Video Playback on Handhelds”, *ICME*, vol. 1, 2002.

[23] *Video Traces for Network Perf. Evaluation*, http://trace.eas.asu.edu/

[24] *MPEG Software Simulation Group*, http://www.mpeg.org/MPEG/MSSG/