Extending the Lifetime of Media Recorders Constrained by Battery and Flash Memory Size *

Younghyun Kim, Youngjin Cho, and Naehyuck Chang †
Seoul National University, Korea
{yhkim, yjcho, naehyuck}@elpl.snu.ac.kr

Chaitali Chakrabarti
Arizona State University, AZ, USA
chaitali@asu.edu

Nam Ik Cho
Seoul National University, Korea
nicho@snu.ac.kr

ABSTRACT

The lifetime of a stand-alone media recorder is a function of both the battery size and flash memory size. In this paper, we present a power management framework for media recorders that significantly enhances their lifetime while minimizing the flash memory usage and maintaining the same level of recording quality. This is achieved by implementing a mixture of encoding algorithms of different complexities that generate data with different compression ratios, and in turn balancing the energy consumption and the flash memory usage.

The proposed method can be effectively employed on a direct battery drive system which does not use a DC–DC converter. The gradual drop of the battery voltage of such system is compensated by operating algorithms of lower complexity more and more. For a speech encoding application where a mixture of ADPCM (low complexity) and MP3 (high complexity) is used, the proposed algorithm achieves 70% more lifetime than a DC–DC converter with a highest clock frequency, and 20% more lifetime than even a DC–DC converter with the optimal clock frequency.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems; C.5.3 [**Microcomputers**]: Portable devices (e.g., laptops, personal digital assistants)

General Terms

Design, Performance

Keywords

Dynamic voltage scaling, Passive voltage scaling, Multimedia

*This work is partly supported by the Brain Korea 21 Project and ETRI SoC Industry Promotion Center, Human Resource Development Project for IT SoC Architect. The ICT at Seoul National University provides research facilities for this study.

†Corresponding author

ISLPED'08, August 11–13, 2008, Bangalore, India..

1. INTRODUCTION

Media recorders have been integrated into many portable devices such as cell phones and PDAs as well as sensing devices used in security applications. These recorders typically consist of a low-power microprocessor (and/or a hardware encoder) and a flash memory for storage. The lifetime of these devices, or the total recording time, is determined by the battery and/or the flash memory size. In this paper, we show how judicious choice of encoding algorithms can be used to increase the lifetime of these devices. The encoding algorithms have various complexities; the more complex algorithms have higher energy consumption but generate smaller amount of encoded data, for instance. This feature can be used to effectively control the energy consumption of the microprocessor and the amount of data that is stored in the flash memory.

In general, the lifetime of portable devices can be increased by minimizing the energy which does not contribute to 'useful' work. The lifetime can be increased also by considering the characteristics of the battery [1, 2]. One of the energy consumptions that can be avoided is the DC–DC converter energy which is consumed to provide constant voltage to the microcontroller CPU (in spite of the decreasing battery voltage) [3, 4]. While a well-designed DC–DC converter can have up to 90% efficiency, its typical efficiency ranges from 70% to 85% depending on the load current [5]. In fact, the loss in DC–DC converters has prompted some sensor node designs that do not support DVS [6].

Now, if the CPU is directly driven by the battery, the power loss due to the DC–DC converter can be eliminated. We refer to this concept which is implemented in some systems such as Telos wireless sensor node [7] as *direct battery drive* (simply direct drive). Since the battery voltage drops with time, the CPU is operated at the lowest clock frequency and thereby delivers a constant performance. However, when the battery has large amount of residual charge, it is wasteful to not let the CPU do more useful work by running at a higher clock frequency.

We have developed another direct battery drive method, called passive voltage scaling (PVS) [8] that changes the clock frequency of the CPU according to the battery voltage. It is deployed in wireless sensor networks where the loss of performance of a single node can be compensated by cooperative processing. But PVS by itself cannot be applied to stand-alone devices such as media recorders. More recently, the direct drive approach was applied in multi-core applications where the number of active cores is increased as the frequencies of the individual cores drop with time [9]. However, like Telos, this method also does not effectively utilize the CPU performance.

The rest of the paper describes the power management frame-

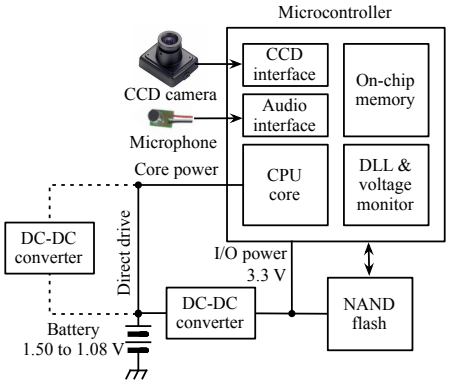


Figure 1: Architecture of a video/audio recorder with a direct drive.

work for implementing direct battery drive CPU for media recording applications. Like PVS, the CPU is operated at the maximum possible clock frequency. As time elapses, the CPU clock frequency decreases, and the reduction in the CPU performance is compensated by implementing a mixture of a high-complexity and low-complexity data encoding algorithms. Specifically, when the battery has large amount of residual energy, the CPU executes a more complex algorithm that achieves high compression ratio. As the battery voltage decreases, the CPU has to operate at a lower frequency and is now only capable of executing a simpler algorithm with a lower compression ratio.

A consequence of this is that the amount of compressed data that is stored in the flash memory increases. We propose a scheme that uses a mixture of low and high complexity algorithms to utilize the available CPU cycles with minimum flash memory usage.

The contribution of this paper is summarized as follows.

- Development of a media recorder based on direct drive CPU that maintains the same encoding quality during its lifetime.
- Introduction of a power management framework that uses a mixture of high and low complexity algorithms to trade off CPU energy consumption with flash memory usage.
- Utilization of this framework to determine the mixture of high and low complexity algorithms which maximizes the lifetime.

We have demonstrated this idea for a speech recorder using a mixture of ADPCM (low complexity) and MP3 (high complexity) encoding algorithms. When the flash memory size is 1 GB and the battery is an LR6 alkaline battery, the proposed algorithm achieves lifetime enhancement of 48% more than a DC–DC converter powered conventional system, 20% more than the optimized DC–DC converter system for the given flash memory size, and 70% more than a direct drive with a fixed clock frequency.

2. SYSTEM ARCHITECTURES

Typical standalone media recorders can be implemented with a low-power microcontroller along with NAND flash memory. Such systems are often powered by alkaline batteries, typically an LR6 AA type whose capacity is 2,500 mAh [10]. Atmel AT91SAM9261 [11] is a low-power microcontroller that can be used for such applications. It is based on supply voltage scalable ARM926 RISC microprocessor core and consumes around 90 mW at 240 MHz with all the peripherals turned on. The ARM926 CPU core can be driven i) with a 1.2 V fixed supply voltage and 240 MHz full speed clock frequency, referred to as *DC–DC converter with a full speed* (DCF), and ii) with an adjustable voltage in the range of 1.08 V to 1.20 V, referred to as *DC–DC converter with DVS* (DCV).

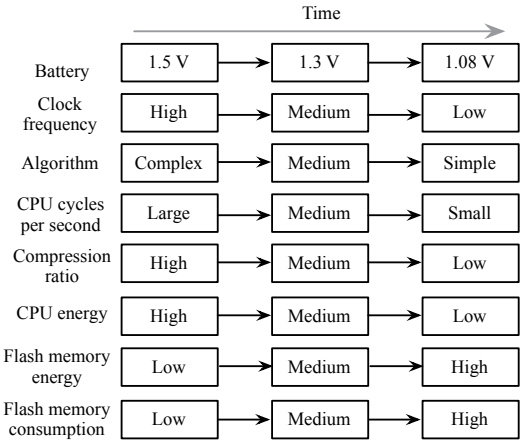


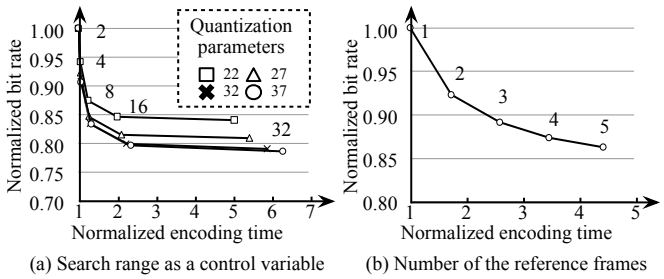
Figure 2: Summary of the DDV operation that maintains the same recording quality during the lifetime of the device.

The NAND flash memory uses 3.3 V supply voltage, and so a typical design uses two DC–DC converters: one for the 3.3 V I/O power and the NAND flash, and the other for the 1.08 V to 1.20 V core power. There are several choices for the boost DC–DC converter that is required to supply the I/O and flash memory power. An example is the TI TPS61070 boost converter that allows an input voltage range of 0.9 V to 5.5 V, and adjustable output up to 5.3 V. Its efficiency is around 70–80% when the input voltage and output voltage are 1.2 V and 3.3 V, respectively, for around 100 mA load current.

Next, we consider a direct drive where the core power comes directly from the battery rather than from the DC–DC converter as shown in Figure 1. In this case, the CPU core voltage varies from 1.50 V to 1.08 V depending on the state of charge of the battery. Clearly, the CPU clock frequency should not exceed the maximum possible value allowed by the battery voltage. In Telos sensor node, the CPU clock is fixed at the minimum value (corresponding to the lowest battery voltage) and thus the available CPU performance is not effectively utilized. From now on, we call the operation setup like Telos as *direct drive with a fixed clock frequency* (DDF).

In contrast, PVS [8] and the gradual wake up of a multi-core system [9] modify the CPU clock frequency according to the battery voltage. We call the setup like these as *direct drive with variable clock frequency* (DDV). In this paper, we apply the concept of DDV to stand-alone applications. Here the loss in CPU performance is compensated by migrating to a lower complexity algorithm when the battery voltage degrades.

Figure 2 shows how the operation mechanism of DDV maintains the same recording quality during the entire lifetime of the recording device. In the beginning, the battery is fully charged and the CPU core is able to operate at the full speed. This enables the CPU core to execute a complex algorithm that results in the high compression ratio. Although the CPU energy consumption is high, the high compression ratio results in less data being stored in flash memory and low flash programming energy. As time elapses, the battery voltage decreases. Now the CPU can run only at medium speed and is capable of executing a medium complexity algorithm. Such algorithm results in lower compression ratio and thus higher flash programming energy and higher flash memory usage. When the battery discharges further, the CPU operates at the lowest clock frequency. Then the flash programming energy and the memory usage increase even more because only a simple algorithm with low compression ratio can be used.



(a) Search range as a control variable (b) Number of the reference frames

Figure 3: Encoding time versus compression ratio in H.264 based video recorder applications.

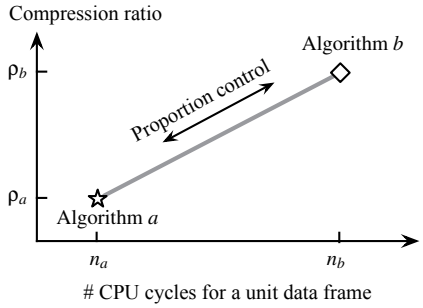


Figure 4: Power management framework using the trade-off between computation time and compression ratio.

In the next two sections, we show how DDV performs significantly better than DCF and DCV in terms of recording lifetime, and DDF in terms of flash memory requirement. We also show that DDV is superior to DCF, DCV and DDF when there is a flash memory size constraint.

3. POWER MANAGEMENT FRAMEWORK

3.1 Proposed Framework

Media recording are built on a rich set of encoding algorithms with various levels of computation complexity. A more complex algorithm achieves higher level of compression requiring more number of CPU cycles. In video encoding schemes such as MPEG-1, -2, -4, H.263, and H.264 [12, 13], motion estimation parameters such as search range or search resolution can be used to trade off complexity with compression ratio as shown in Figure 3(a). Wider and finer search results in reduced size of encoded data while increasing computations to meet the given quality constraint. In the case of H.264, greater compression can be achieved by searching over multiple frames as shown in Figure 3(b).

Such trade-off between the computation time and the compression ratio exists not only for video but also for speech and audio applications. By changing the algorithm or using a mixture of low and high complexity algorithms, the computation time changes but the image or speech quality can be maintained at the same level. Figure 4 illustrates how two encoding algorithms, Algorithm a and Algorithm b that differ in the parameters and thereby complexity can be combined to achieve different compression ratios. So if a large number of CPU cycles is available, then the more complex algorithm (Algorithm b) with higher compression ratio is used more.

Let the number of CPU cycles for each algorithm to process unit data by Algorithms a and b be denoted by n_a and n_b (cycle/byte), respectively. We denote the corresponding compression ratio as ρ_a and ρ_b, respectively. Let the size of incoming data per second be

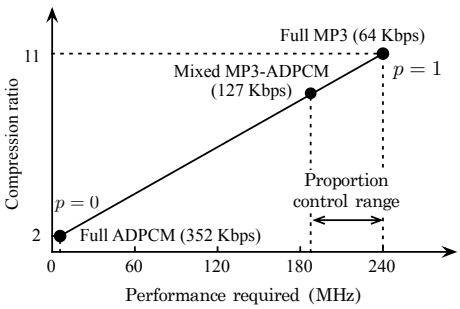


Figure 5: Trade-off between computation time and compression ratio using a mixture of MP3 and ADPCM encoding algorithms. The values correspond to an ARM9 implementation.

N_I (byte/sec). Suppose we encode p ($0 \le p \le 1$) portion of the incoming data with Algorithm a, and the rest $1-p$ portion with Algorithm b. The required CPU clock frequency f_{cpu} (cycles/sec) to process the whole data is

$$f_{cpu} = N_I\,(n_a p + n_b(1-p))\,. \tag{1}$$

Therefore, if the CPU clock frequency is given, p is obtained as

$$p = \frac{f_{cpu}/N_I - n_b}{n_a - n_b}. \tag{2}$$

Note that both $N_I n_a p$ and $N_I n_b(1-p)$ should be integers for a feasible implementation.

The power consumption of the CPU, P_{cpu}, is denoted by

$$P_{cpu} = P_0 \frac{V_{DD}^2}{V_0^2} \frac{f_{cpu}}{f_0} + I_{leakage} \cdot V_{DD}, \tag{3}$$

where $I_{leakage}$ is the leakage current of the CPU, V_{DD} is the supply voltage to the CPU, and P_0 is the dynamic power consumption of the CPU when $V_{DD} = V_0$ and $f_{cpu} = f_0$, as described in the manufacturers' datasheet. We consider a CPU fabricated in a 90 nm technology, and assume the CPU leakage power to be 15% to 21% of the total CPU core power.

The amount of data that is processed and stored in the flash memory, N_f (byte/sec), is

$$N_f = N_I \left(\frac{p}{\rho_a} + \frac{1-p}{\rho_b} \right). \tag{4}$$

Let the energy required to program the unit data size be E_f (J/byte). This can be derived from the programming delay, power, and page size given by the manufacturers. Then the flash power P_{flash} (J/sec), is given by

$$P_{flash} = E_f N_f. \tag{5}$$

The total system power, P_{system}, for DCF and DCV is denoted by

$$P_{system} = \eta_{dc}(P_{cpu} + P_{flash}), \tag{6}$$

where η_{dc} is the DC–DC converter efficiency, and V_{DD} is the minimum voltage required to operate at the specified clock frequency. For DDF and DDV, where the CPU is directly connected to the battery without DC–DC converter,

$$P_{system} = P_{cpu} + \eta_{dc} P_{flash}. \tag{7}$$

The system energy is the integral of P_{system} over time.

In the rest of this paper, we consider a speech compression application which uses a mixture of MP3 and ADPCM, running on an ARM926 RISC core. A detailed listing of the parameters for the

Table 1: Parameters used for the energy and timing models

Symbol	Value	Unit	Symbol	Value	Unit
η_{dc}	75	%	P_0	76	mW
V_0	1.2	V	$I_{leakage}$	12	mA
N_I	704	Kbps	E_f	23.5	nJ/byte
n_a	2.7	Kcycle/byte	n_b	27	cycle/byte
ρ_a	11	-	ρ_b	2	-
Q_{batt}	2500	mAh			

energy and timing models is shown in Table 1,assuming constant compression ratios and throughputs of the encoding algorithms. Figure 5 illustrates the trade-off between the computation time and the compression ratio using measured values. A full MP3 encoding requires the highest clock frequency of 240 MHz and achieves a bit rate of only 64 Kbps. On the other hand, a full ADPCM encoding requires 100 times less number of CPU cycles, but the resultant bit rate is 352 Kbps. When the CPU operates between 188 MHz to 240 MHz, the parameter p can be varied from $0.781 \leq p \leq 1$.

3.2 Determination of p

Figure 6 illustrates the CPU and flash power consumption as a function of the CPU clock frequency. The power consumption of contemporary NAND flash is 61 mW when the duty ratio is 100%. The speech encoding application produces only a low output bit rate such that duty ratio is less than 0.1%. So, in the range of interest in Figure 6, the flash power consumption is less than 0.6 mW and is not noticeable.

Figure 6 also shows the output bit rate, i.e., the flash memory consumption rate, as a function of the CPU clock frequency. As the CPU clock frequency decreases, the bit rate increases accordingly. But still, the flash memory contributes very little to the total power consumption. However it acts as a strong constraint in terms of the memory size cost.

Determination of p for DCF. In DCF, the value of p is determined in a straight-forward manner. The CPU is set to provide enough performance to support 100% operation (corresponding to $p = 1$) of the complex algorithm. As shown in Figure 6, DCF results in the CPU core operating at the maximum clock frequency. In the speech encoding application, the resultant total power consumption reaches 120 mW due to very high CPU frequency for executing of MP3.

Determination of p for DCV. In Figure 6, the operating point of DCV can be any point on the dashed line representing the total power consumption. The proposed power management framework determines the value of p that maximized the lifetime, given the flash memory size constraint. Note that if the flash memory size is not considered, p is likely to correspond to the simplest algorithm which requires a very large amount of flash memory.

For a simple battery model, if Q_{batt} is the battery energy capacity, the expected lifetime T_{rec} can be estimated as

$$T_{rec} = \frac{Q_{batt}}{(P_{cpu} + P_{nand})/\eta_{dc}}, \tag{8}$$

Let the given flash memory size be S_f. Then from (4),

$$\begin{aligned} S_f &= T_{rec} N_f \\ &= \frac{Q_{batt}}{(P_{cpu} + P_{nand})/\eta_{dc}} N_I \left(\frac{p}{\rho_a} + \frac{1-p}{\rho_b} \right). \end{aligned} \tag{9}$$

Note that P_{cpu} is a function of p by (1) and (3), if the relationship between V_{dd} and f_{cpu} follows the alpha power delay model, $\frac{1}{f_{cpu}} \propto \frac{V_{DD}}{(V_{DD} - V_t)^\alpha}$, and $\alpha \approx 1$ for a 90 nm technology. Thus (9) is a polynomial to p and is easily solvable by numerical analysis.

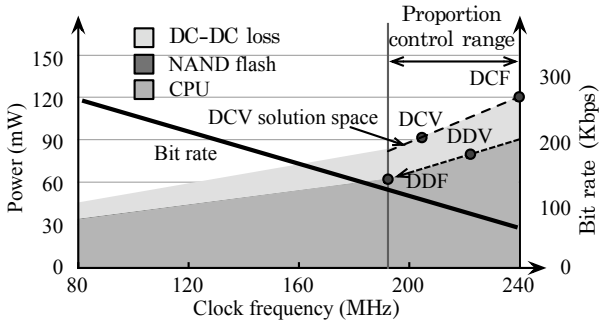


Figure 6: CPU and flash power consumption, and DC–DC converter loss as a function of the CPU clock frequency. Note that flash power (in the middle) is not noticeable.

In practice, user operation patterns such as recording and erasing affect the resultant lifetime. However, every time that the constraints such as residual battery and memory size change, a new value p can be determined to meet the new constraints.

Determination of p for DDF. DDF is designed to operate at the lowest clock frequency and shows the lowest power consumption because the flash power consumption is negligible compared to the CPU power consumption. In the speech encoding example, DDF uses $p = 0.781$, which is the smallest value among those chosen by the other systems.

Determination of p for DDV. DDV is not capable of actively adjusting p. Instead, the value of p in DDV is changed passively by the battery voltage drop. In this example, the operating point of DDV traverses from 240 MHz to 188 MHz along the dashed line, which corresponds to p varying in the range of 1 to 0.781.

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup

The energy and performance model parameters of the CPU and NAND flash memory are borrowed from the datasheets of Atmel AT91SAM9261 [11] and Samsung K9G4G08U0M 4 Gbit MLC NAND flash, respectively. For the battery, we have used the discrete battery model of 2,500 mAh LR6 AA alkaline battery which considers the rate capacity and recovery effects [14]. The audio source is a mono 16-bit 44 KHz stream. We assumed that the sampled data and the compressed data are transfered by the DMA controller without CPU intervention, and DMA transfer rate is independent of the CPU clock frequency. The performance requirements and compression ratios are estimated by running an MP3 encoder and an ADPCM encoder on an ARM9 processor. The proposed configuration DDV (direct drive with variable frequency) is compared against three other competing configurations, DCF (DC–DC converter with full frequency), DCV (DC–DC converter with DVS), and DDF (direct drive with fixed clock frequency).

4.2 Variation of f_{cpu}, P_{cpu}, and N_f as a Function of Time

Figure 7(a) shows the variation in CPU clock frequency as a function of time. Here DCF is set to 240 MHz, DCV to 188 MHz

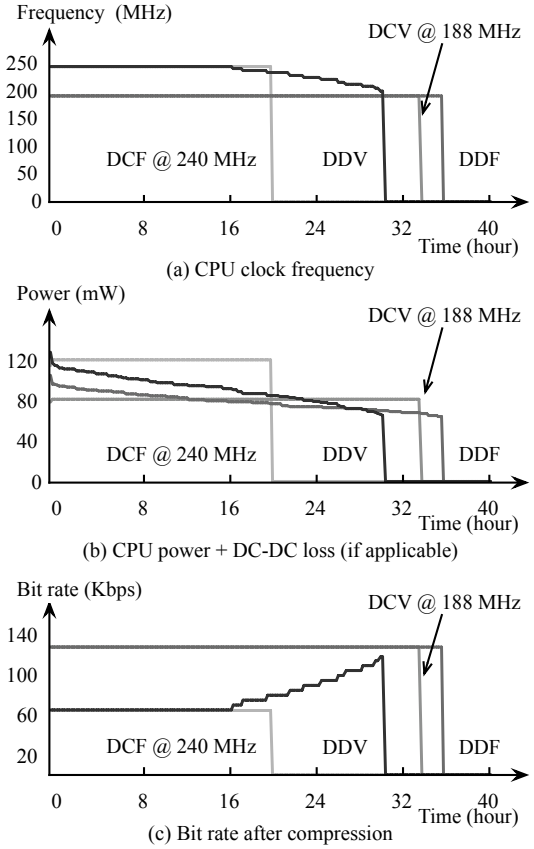


Figure 7: CPU clock frequency, CPU power and bit rate variation with time.

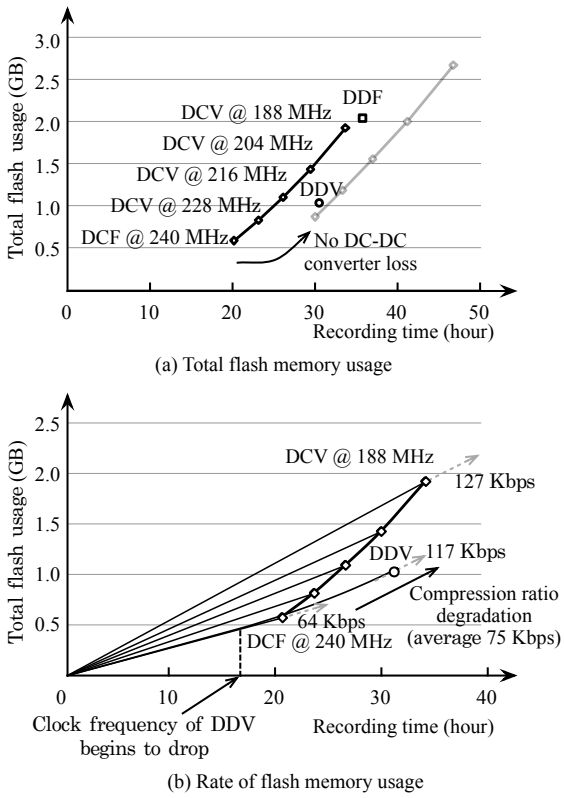


Figure 8: Comparison of recording time and (a) total flash memory usage, (b) rate of flash memory usage.

and DDF to 188 MHz. While these three maintain the same clock frequency at all times, the CPU clock frequency in DDV drops when the battery voltage drops below 1.2 V. However, DDV has a CPU clock frequency similar to that of DCF till 18 hours when DCF goes out.

Figure 7(b) shows the variation in CPU power consumption over time. The power consumption of DCF at 240 MHz is the highest, which explains its shortest lifetime. The highest power consumption is caused by the high dynamic power of the CPU and the DC–DC converter loss. While DDF always keeps the same CPU clock frequency, its actual dynamic power consumption changes over time because its supply voltage, which comes from the battery unregulated, changes over time. As the CPU power consumption of DDV is always higher than that of DDF, the lifetime of DDV is shorter than that of DDF.

Figure 7(c) shows the bit rate for compressed data as a function of time. Note that the bit rate is inversely proportional to the compression ratio. The bit rate of DCF at 240 MHz is the lowest, while that of DDF and DCV at 188 MHz is the highest. In the beginning, DDV has the same low bit rate as that of DCF at 240 MHz. But the bit rate of DDV begins to increase when the clock frequency drops, as expected.

4.3 Comparison of Recording Time and Flash Memory Usage

Figure 8(a) compares the lifetime of the voice recorder and the flash memory usage of the four competing operations. DCF setup achieves the least flash memory usage. However, due to the high power consumption by the CPU and the DC–DC converter loss, its lifetime is the shortest. DCV can adjust the parameter p (the proportion of MP3 and ADPCM described in Section 3) and operate at one of the multiple frequencies as shown in Figure 8(a). More ADPCM encoding, which is represented by a smaller value of p, results in less CPU power consumption and so longer lifetime, but higher flash energy and flash usage. DCV at 188 MHz CPU clock extends the recording time by 67% from 20 to 34 hours compared to DCF.

Next we illustrate the inefficiency introduced by the DC–DC converter. The solid gray line in Figure 8(a) shows how the lifetime of DCF and DCV can be enhanced if there were a 100% efficient DC–DC converter, which is not possible in practice. For instance, DCF could operate for 30 hours (instead of 20 hours) if there was no DC–DC converter loss. The p value is fixed, determined by the lowest clock frequency, in DDF but can adjustable in DCV. In spite of that, DDF at 188 MHz is alive for two hours more than DCV at 188 MHz due to the elimination of the DC–DC converter loss. DDV lasts for 10 hours more than DCF at 240 MHz, and its performance is very close to the gray curve corresponding 100% DC–DC converter efficiency.

Figure 8(b) explains the superiority of DDV in more detail. Note that both Figures 8(a) and 8(b) plot flash usage and recording time, but their annotations are different. Figure 8(b) shows the flash consumption rate of each setup explicitly. Every DCV setup corresponds to one specific value of p that is fixed for the entire duration. On the other hand, DDV keeps changing the values of p during the recording lifetime. Thus the slope of DDV changes over time according to the drop of battery voltage. DDV has the same rate of flash usage as that of DCF until the clock frequency begins to drop. DCF goes out after 10 hours, but DDV is still operative up to 30 hours.

4.4 Recording Time with Flash Memory Size as a Constraint

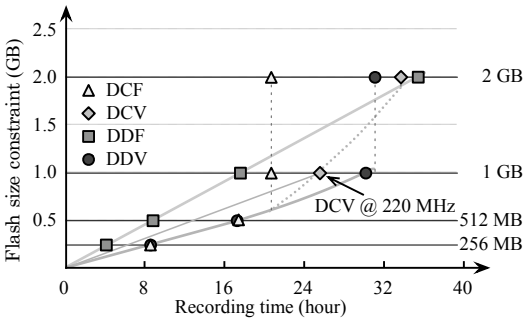


Figure 9: Recording time with flash memory size constraint.

The flash memory capacity is an important design constraint due to its effects on the cost and size of the voice recorder. Figure 9 shows the recording time when the flash memory size is 256 MB, 512 MB, 1 GB and 2 GB.

In Figure 9, DDF shown with shaded rectangles lies on the line with the stiffest slope. The DDF lifetime continues to increase as the flash size increases until the flash size is 2.04 GB. This implies that up to 2.04 GB flash size, the lifetime of DDF is determined not the battery, but by the flash memory.

The lifetime of DCF, on the other hand, is dictated by the battery capacity. When the battery goes out, DCF has used no more than 580 MB of flash memory. Therefore, if the flash memory is larger than this size, the lifetime of DCF is determined by the battery capacity.

Flash memory size plays a very important role in DCV. Since DCV is able to control the parameter p, in this experiment, we choose the optimal p value for each memory constraint. The optimal p value is the value that makes the battery capacity budget and the memory capacity budget to expire at the same time (refer to (9)). For each flash memory constraint, the optimal DCV setup is obtained with a different CPU clock frequency. For instance, the best DCV setup for 1 GB flash memory is to set the CPU clock to 220 MHz.

DDV uses 1.03 GB until the battery goes out, thus if the memory size increases to 2 GB, the battery runs out prior to the memory, and the lifetime is determined by the battery capacity.

Next we show how the value of p can be determined for a system with flash memory size constraint. For instance, for the DCV method, the value of p can be obtained using (9). The solution can also be obtained by the intersection of the dotted curve with the 1 GB flash line in Figure 9. This corresponds to the time when the battery and the memory run out at the same time.

Although the proposed DDV cannot actively control p, it shows excellent lifetime. Even if a 512 MB flash is used, DDV shows 98% recording time compared to DDF. Note that DDF shows the longest recording time only when the flash is very large. For flash size of 1 GB, DDV shows the longest lifetime: 9.7 hours longer than DCF (48%), 4.9 hours longer than DCV (20%), and 12.3 hours longer than DDF (70%). Table 2 summarizes the numerical data.

In our experiments, the clock frequency is varied from 188 MHz to 240 MHz since in this range the supply voltage can also be scaled. Scaling the clock frequency lower than 188 MHz results in linear reduction of the CPU power consumption but considerable increase of the flash memory usage, which is not desirable. Only a slight energy saving is achieved as shown in Figure 6, since only frequency scaling is possible.

5. CONCLUSIONS

Media recorders are typically constrained by both the size of the battery and the size of the flash memory. In this paper, we presented a power management framework that can be used to trade-off energy consumption with flash memory usage. By using a mixture of high complexity (high energy consumption producing small volume of encoded data) and low complexity (low energy consumption producing large volume of encoded data) algorithms, the energy consumption and the flash memory usage can be controlled. When implemented on a dynamic voltage scaling (DVS) based system, this scheme achieves significant more lifetime than conventional DC–DC converter driven systems.

In order to further increase the lifetime, we implemented this scheme on a direct drive CPU. Although in such system, the CPU has a gradual drop in performance over time, use of the proposed framework allows us to migrate to a lower complexity algorithm thereby maintaining the same recording quality. We have demonstrated additional 20% more lifetime even over the optimized DVS, which is 48% more lifetime than a conventional DC–DC converter driven system and 70% more lifetime than a previous direct drive system.

Table 2: Recording time (in hour) for different flash memory sizes. Note that the frequency of DCV is optimized for the given flash size constraint.

Memory capacity	DCF	DCV	DDF	DDV
256 MB	8.7	8.7	4.4	8.6
512 MB	17.4	17.4	8.7	17.3
1 GB	20.1	24.9	17.5	29.8
2 GB	20.1	33.6	35.0	30.4

Out of flash Out of battery Out of the both

6. REFERENCES

[1] M. Pedram and Q. Wu, "Design considerations for battery-powered electronics," in *DAC '99*, pp. 861–866, 1999.

[2] D. Rakhmatov and S. Vrudhula, "Energy management for battery-powered embedded systems," *Trans. on Embedded Computing Systems*, vol. 2, no. 3, pp. 277–324, 2003.

[3] V. Kursun, S. G. Narendra, V. K. De, and E. G. Friedman, "Monolithic DC-DC converter analysis and MOSFET gate voltage optimization," in *ISQED '03*, p. 279, 2003.

[4] V. Kursun, S. G. Narendra, V. K. De, and E. G. Friedman, "Low-voltage-swing monolithic DC-DC conversion," *IEEE Trans. on Circuits and Systems*, pp. 241– 248, 2004.

[5] T. . Šimunić, L. Benini, and G. D. Micheli, "Energy-efficient design of battery-powered embedded systems," *IEEE Trans. on Very Large Scale Integration Systems.*, vol. 9, no. 1, pp. 15–28, 2001.

[6] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, pp. 40–50, March 2002.

[7] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *IPSN '05*, p. 48, 2005.

[8] Y. Cho, Y. Kim, and N. Chang, "PVS: passive voltage scaling for wireless sensor networks," in *ISLPED '07*, pp. 135–140, 2007.

[9] S. Kim, R. P. Dick, and R. Joseph, "Power deregulation: eliminating off-chip voltage regulation circuitry from embedded systems," in *CODES+ISSS '07*, pp. 105–110, 2007.

[10] D. Linden and T. B. Reddy, *Handbook of Batteries*. McGraw-Hill, 3rd ed., 2001.

[11] "Atmel AT91SAM9261 datasheet," 2008. http://www.atmel.com/dyn/resources/prod_documents/doc6062.pdf.

[12] L. . Harte, A. Wiblitzhouser, and T. Pazderka, *Introduction to MPEG; MPEG-1, MPEG-2 and MPEG-4*. Althos Publishing, 2006.

[13] I. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. Wiley, 2003.

[14] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "A discrete-time battery model for high-level power estimation," in *DATE '00*, pp. 35–41, 2000.