A Backlight Power Management Framework for Battery-Operated Multimedia Systems

Hojun Shim
School of CSE
Seoul National University
Korea 151-010

Naehyuck Chang*
School of CSE
Seoul National University
Korea 151-010

Massoud Pedram
Department of EE
University of Southern California
USA CA 90089

Abstract

Dynamic Luminance Scaling (DLS) reduces the backlight power consumption of a color TFT LCD while maintaining the image brightness or contrast. Extended DLS (EDLS) is a framework for backlight power management of transflective LCD panels. It switches between DLS and Dynamic Contrast Enhancement (DCE) depending on the panel mode, which may be transmissive or reflective, and the remaining battery energy. The benefit is an average 25% power saving in the backlight system for still and moving images, without any modification of existing applications.

1 Introduction

Color TFT (Thin-Film Transistor) LCD (Liquid Crystal Display) panels enable battery-operated hand-held embedded systems to support full-featured multimedia and have replaced monochrome STN (Super Twisted Nematic) LCD panels in most applications. Most importantly, a TFT LCD panel does not illuminate itself, but filters a backlight source, and this backlight is a primary power consumer in most systems. Thus, reducing backlight power consumption is one of the primary ways to extend battery life in battery-powered electronic devices. Most existing power reduction techniques are based on power management during idle or slack times, and are therefore difficult to apply to display panels, which have no idle time as long as they are turned on. Simply dimming or turning off the backlight results in appreciable degradation of the legibility of the display.

Recently, a power reduction technique was introduced [1] which maintains either the brightness or the contrast of the LCD panel when the backlight is dimmed down. Appropriate image compensation techniques preserve either the brightness or the contrast of the original image, at the expense of minor image distortion, which does not seriously affect the legibility of the display. Because the CCFL (Cold Cathode Fluorescent Lamp) backlight is usually considered to have a slow response, a feedback control circuit has been proposed [2] to enable the backlight luminance to change fast enough to support movie streams. This technique is known as dynamic luminance scaling (DLS) of a backlight. Human understanding of a display is affected by both the brightness of the LCD panel and the ambient luminance, which has motivated another approach involving backlight autoregulation in the context of ambient luminance [3]. Simultaneous brightness and contrast scaling [4] enhances image fidelity with a dim backlight, and thus permits an additional reduction in backlight power. In an MPEG-1 video streaming application, this approach (including the necessary image processing) has been implemented [5] by adaptive middleware, to avoid an extra burden on the streaming clients.

In this article, we introduce a backlight power management framework for color TFT LCD panels, as used in battery-operated multimedia applications. We extend DLS to cope with transflective LCD panels, that operate both

*Corresponding author

with and without a backlight, depending on the remaining battery energy and the ambient luminance. These transflective LCD panels dominate the battery-operated electronic systems, and the reason for their popularity is that the image is still visible without the backlight, even though the quality may be poor. We name our new backlight power management framework extended DLS (EDLS). It compensates for loss of brightness when there is a rich or moderate power budget and it compensates for loss of contrast when the power budget is poor.

The need for modification of existing applications limited the usage scope for using DLS because previous implementation required source code modification of existing multimedia application [2]. In developing EDLS, we have pursued an application-transparent approach intercepting the frame buffer contents and performing image compensation. We expect that this will contribute to the wide adoption of DLS. Our system is an application-transparent, pure hardware EDLS implementation with a 16-bit color depth and 640×480 screen resolution. It exhibits a 25% power reduction while maintaining acceptable image quality. The hardware overhead, in terms of both area and power is moderate.

2 Theory of operation

2.1 Transflective LCD

There are three kinds of TFT LCD panels [6]. In transmissive LCDs, the pixels are illuminated from behind (*i.e.* opposite the viewer) using a backlight. Transmissive LCDs offer a wide color gamut and high contrast, and are typically used in laptop personal computers. Transmissive LCDs give their best performance under lighting conditions ranging from complete darkness to an office environment. Reflective LCDs are illuminated from the front (*i.e.* the same side as the viewer). Reflective LCD pixels reflect incident light originating from the ambient environment or from a frontlight. Reflective LCDs can offer very low power consumption (especially without a front light) and are often used in small portable devices such as handheld games, PDAs or instrumentation. Reflective LCDs perform best in a typical office environments or in brighter lighting. Under dim lighting conditions, reflective LCDs require a frontlight. Transflective LCDs are partially transmissive and partially reflective, so they can make use of environmental light or a backlight. Transflective LCDs are often used in devices that will be used under a wide variety of lighting conditions from complete darkness to sunlight.

Transmissive and transflective LCD panels use very bright backlight sources that emit more than 1000 cd/m^2. However, the transmittance of the LCD, ρ_T, is relatively low, and thus the resultant maximum luminance of the panel is usually less than 10% of the backlight luminance. Theoretically, the backlight and the ambient light act in an additive manner. However, once the backlight is turned on, a transflective LCD panel effectively operates in the transmissive mode because the backlight source is generally much brighter than the ambient light.

The brightness in transmissive mode is proportional to the product of the transmittance, ρ_T, and the backlight luminance, L_B [7]. Similarly, the brightness in the reflective mode is proportional to the product of the reflectance, ρ_R, and the ambient luminance, L_A. The reflectance of an LCD panel is even lower than its transmittance. The transmissive mode is significantly superior to the reflective mode in terms of both brightness and contrast. For example, the NEC6448BC33-50 LCD panel exhibits a contrast ratio of 300:1 and 8:1 in transmissive and reflective modes, respectively.

2.2 The EDLS framework

The principle of DLS is to reduce the luminance of the light source but to compensate brightness by allowing more light to pass through the screen by enhancing the image luminance [1, 2]. The viewer should perceive little change. DCE (Dynamic Contrast Enhancement) also enhances image quality under a dimmed backlight, but by increasing the contrast of the image. While DLS preserves the original colors, DCE can result in a noticeable change to the original colors in pursuit of higher contrast and improved legibility. DCE must therefore be considered as a very aggressive power management scheme for transmissive LCD panels. This clearly differentiates the DLS and DCE strategies.

The EDLS framework (Fig. 1) achieves a harmonious combination of DLS and DCE. The EDLS interface is a simple slider knob similar to a brightness control knob on a monitor. The EDLS knob controls the trade-off between energy consumption and image quality, and not simply the brightness of the backlight. It provides users with a power management scheme which can be used to extend the battery life at the cost of whatever display degradation they are

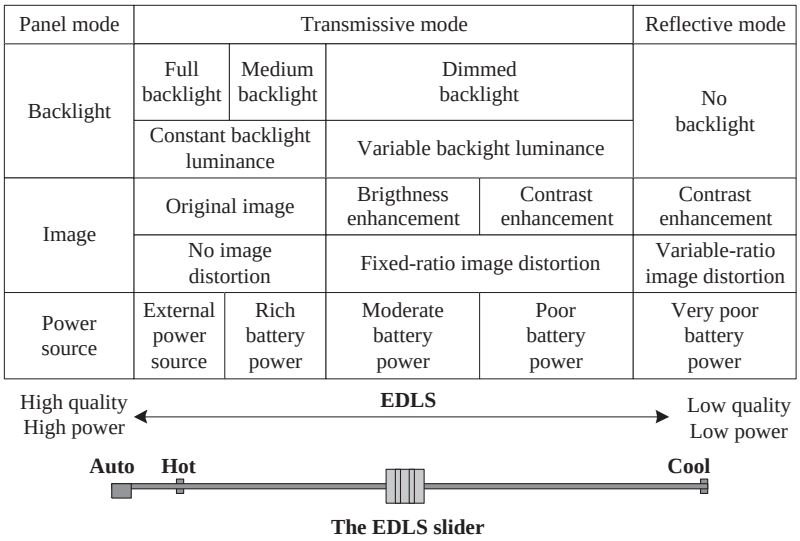


Figure 1: The EDLS framework.

prepared to accept. There is also an automatic mode which changes the power management setting depending on the remaining battery energy.

When the system is connected to an external power source, the backlight will be fully turned on and exhibit its maximum luminance. In this case, no backlight power management is desirable, so that users may enjoy the best image quality. When the system is battery-powered, users may want to extend the battery life for future use even when the battery is fully charged, but they are generally not ready to sacrifice picture quality appreciably at that stage. As the remaining battery energy decreases, users may become more willing to compromise image quality to improve the battery life. At this point, EDLS applies DLS.

With a poor power budget, the user's prime concern may well be to complete their current task within the remaining battery energy budget, even if the image quality decreases. This is the right time for EDLS to change from DLS mode to DCE mode. Although DCE may alter the original colors, a moderate degree of DCE does at least maintain a fixed distortion ratio. However, if the battery energy is nearly exhausted, the only remaining possibility is to turn off the backlight. Without the backlight, EDLS applies DCE to achieve the maximum possible contrast. In this case, EDLS cannot guarantee a fixed amount of image distortion, but it should still be possible to read the display, and finish the task.

2.3 Formulation of DLS and DCE

To build the EDLS framework, we borrow the principles of DLS for brightness compensation and of DCE for contrast enhancement [2]. The EDLS process starts by building a RGB histogram of the image to be displayed. The EDLS slider determines the panel mode (transmissive or reflective), the image processing algorithm (DLS or DCE), and the maximum allowed percentage of saturated pixels, S^R, after image processing. Note that S^R is a given input parameter determined by user preference. Then the EDLS process derives upper and lower thresholds, T_H and T_L, from S^R and the histogram, and calculates a scaling factor that controls the amount of backlight dimming as shown in [2]. Let C denote the current color value. After brightness compensation, the new color value, C', is given by

$$C' = \min(2^n - 1, S_{BC} \times C), \tag{1}$$

where n is the color depth of each color component, and S_{BC} is the brightness compensation factor, which is equal to $\frac{2^n - 1}{T_H}$. Similarly, after contrast enhancement, the new color value, C', is given by

$$C' = \min(2^n - 1, S_{CE} \times \max(0, C - T_L)), \tag{2}$$

where the contrast compensation factor, S_{CE}, is equal to $\frac{2^n - 1}{T_H - T_L}$. After image compensation, we reduce the backlight luminance, L_B, so that $L'_B = \frac{L_B}{S_{BC}}$ or $L'_B = \frac{L_B}{S_{CE}}$, depending on the EDLS mode. The power reduction ratio of the backlight is $1 - \frac{L'_B}{L_B}$ [1].

3 Trade-offs in the EDLS implementation

3.1 The EDLS design space

Fig. 2 summarizes how to add an EDLS capability to typical multimedia applications. Such applications draw images in the frame buffer, and the backlight luminance is fixed by user preference (Fig. 2 (a)). Because there are many ways to add EDLS to an application, we need to consider application transparency and hardware-software partitioning. In addition, we need to optimize the energy reduction of the backlight, energy overhead and performance penalty of the EDLS process itself, and the resulting image quality.

The gray boxes represent functional blocks to be added to an existing implementation. Our first approach is to embed EDLS in an application program (Fig. 2 (b)). The advantage of this approach is that we will have many opportunities to reduce the EDLS overhead. For example, we may construct an approximate histogram in a compressed domain for an MPEG decoder. Sometimes, we can obtain the histogram before rasterization and thus avoid additional frame buffer accesses. The first DLS implementation* falls into this category; it was highly coupled with the application. However, such optimizations are *ad hoc*, and the necessary changes to the application discourage developers from using EDLS, due to the heavy porting burden.

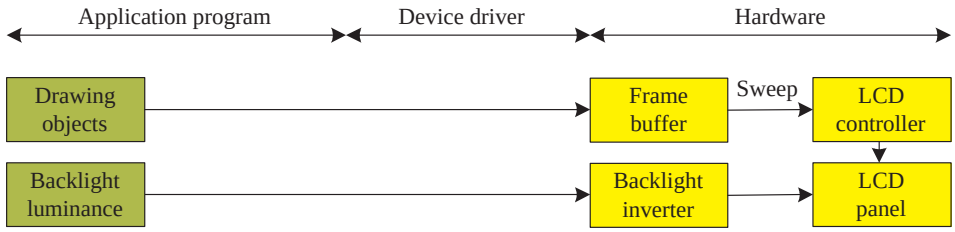
Our second DLS implementation introduced a standard API (Application Program Interface) at the window management level [2]. A standard EDLS API makes porting systematic [2], but this approach still involves source code modification. In many cases, EDLS developers simply cannot access the source code of an existing application. Even though this approach has limited portability, it maximizes energy reduction and image quality because it can use the application context [2].

The alternative approach is to implement EDLS functionality outside the applications, as shown in Fig. 2 (c) and (d). This offers an application-transparent EDLS implementation because modification of the existing application is not needed. Instead, we simply redirect the frame buffer address pointer using a new device driver. The EDLS functional blocks then periodically read the temporary frame buffer and rebuild the histogram. However, visual artifacts may occur due to improper synchronization between the application and the EDLS functional blocks. Fig. 2 (c) shows how all the EDLS functional blocks are implemented in a frame buffer device driver. Because the application directly accesses the frame buffer memory to draw the cursor, menus, pictures and so on, oversampling is the only way to synchronize an application with the EDLS functional blocks. The overhead for refreshing the histogram is potentially much higher than that of an application-embedded implementation because the EDLS functional blocks must read the entire frame buffer in every refresh period.

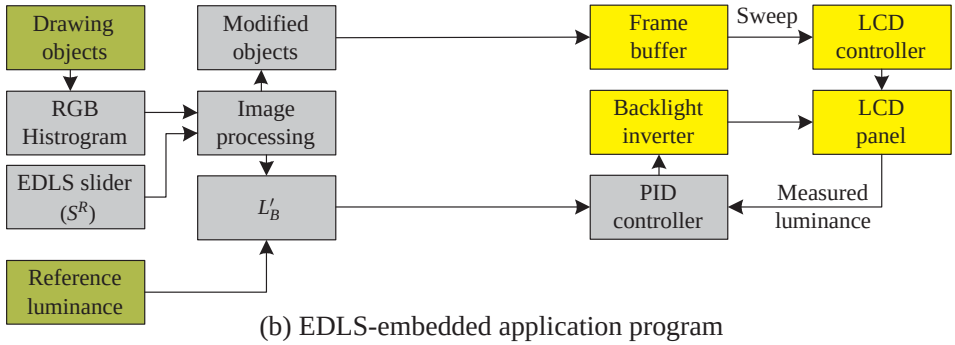
An appropriate way to partition hardware and software may be to embed the EDLS functional blocks in the LCD controller (Fig. 2 (d)). In that case, synchronization of the EDLS functional blocks with an application does not cause visual artifacts, since the LCD controller sweeps the LCD panel every 16.67ms†, and application-transparent hardware EDLS updates the histogram whenever the sweep operation occurs. We have added extra comparators and counters to a standard LCD controller to construct the histogram. Image processing requires additional datapath resources such as multipliers, adders and comparators, as defined by Eqs. 1 and 2, but the hardware EDLS approach does not involve any additional frame buffer accesses. Image processing is performed on-the-fly before issuing the RGB color data to the LCD panel, while the frame buffer always contains the original image.

*A demonstration was performed at the SIGDA University Booth of the 2000 Design Automation Conference and at the Design Contest of the 2002 International Symposium on Low Power Electronics Design.

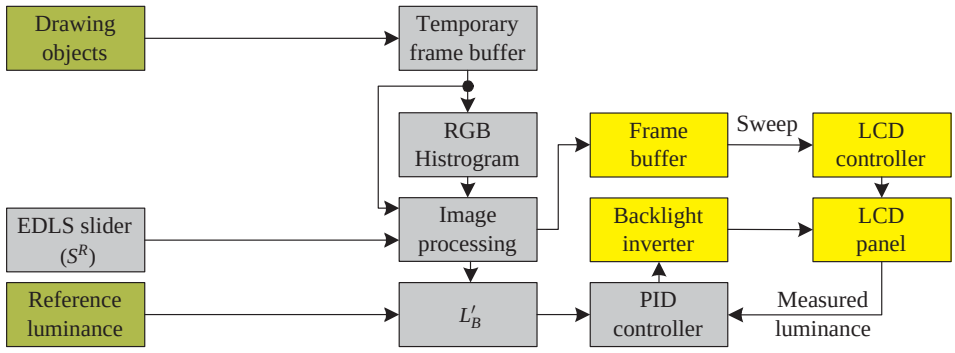
†LCD panels are commonly designed for 60Hz referesh.



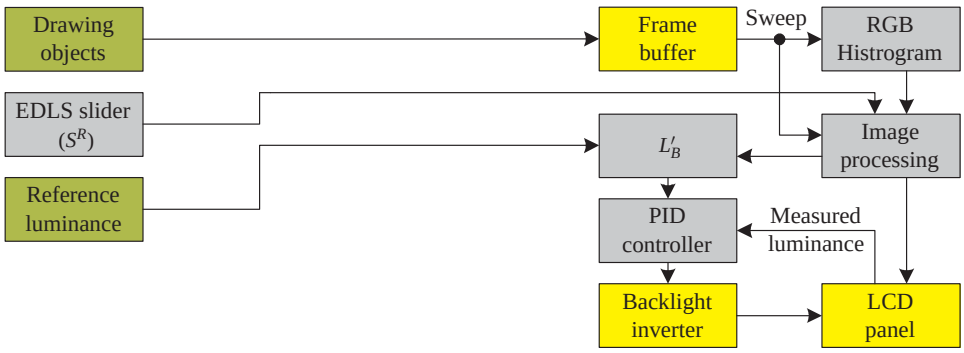
(a) Conventional application program



(b) EDLS-embedded application program



(c) Application-transparent software EDLS



(d) Application transparent hardware EDLS

Figure 2: Porting the EDLS capability.

3.2 Compact EDLS

The area complexity of application-transparent hardware EDLS explodes as the color depth increases. More precisely, the area overhead increases exponentially with the color depth, n, because 2^n counters and comparators are necessary to construct the histogram. However, we can approximate the EDLS algorithms to reduce the area complexity. Our test platform requires 64 19-bit counters and 6-bit comparators to construct a full-resolution histogram at 640×480 resolution display and 16-bit color depth. The area explosion corresponding to the color depth affects both the cost and energy overhead. We have forged a compromise between the energy saving from the backlight and the area complexity of the EDLS-enabled LCD controller, called compact EDLS. We use acronym EDLS-d, in which d signifies that a d-digit histogram, where $d \leq n$ is used. More precisely, we truncate the color values to d-digit numbers and compose a d-digit histogram. Using EDLS-d, a 2^n-level histogram is approximated by a 2^d-level histogram. This imposes restrictions on the value of T_H and T_L, and reduces the area complexity of application-transparent hardware EDLS from 2^n to 2^d.

EDLS-d may achieve lower power savings from the backlight compared to EDLS because it may result in reduced brightness compensation or a smaller contrast enhancement factor. Note that the energy reduction achieved by backlight dimming is roughly equal to $\frac{1}{S_{BC}}$ or $\frac{1}{S_{CE}}$, depending on the EDLS mode. The worst-case threshold, T'_H, and the brightness compensation factor, S'_{BC}, of EDLS-d are calculated as

$$T'_H = \lceil T_H, \frac{2^n}{2^d} \rceil = T_H + \frac{2^n}{2^d}, \tag{3}$$

and

$$S'_{BC} = \frac{2^n - 1}{T'_H} = \frac{2^d(2^n-1)S_{BC}}{2^n S_{BC} + 2^d(2^n-1)} < \frac{2^d S_{BC}}{S_{BC} + 2^d}, \tag{4}$$

where T_H and S_{BC} are the threshold and the brightness compensation factor used by EDLS. Thus the actual saturation ratio, S'^R is determined by T'_H. Since $T'_H \leq T_H$ and thus $S'^R \leq S^R$, EDLS-d achieves a smaller power reduction than EDLS. Note that $\lceil A, B \rceil$ is a ceiling function of number A, which rounds up A to the nearest multiple of significance, B. The difference in the power reduction between EDLS and EDLS-d in the DLS mode is equal to $\frac{P_B}{S_{BC}} - \frac{P_B}{S'_{BC}}$, and thus it is bounded by $\frac{2^n P_B}{2^d(2^n-1)}$, where P_B is the original backlight power consumption.

In the same way, the worst-case upper and lower thresholds, T'_H and T'_L, and the worst-case contrast enhancement factor, S'_{CE}, of EDLS-d in the DCE mode are calculated as

$$T'_H = \lceil T_H, \frac{2^n}{2^d} \rceil = T_H + \frac{2^n}{2^d}, \tag{5}$$

$$T'_L = \lfloor T_L, \frac{2^n}{2^d} \rfloor = T_L - \frac{2^n}{2^d}, \tag{6}$$

and

$$S'_{CE} = \frac{2^n - 1}{T'_H - T'_L} = \frac{2^d(2^n-1)S_{CE}}{2^{n+1}S_{CE} + 2^d(2^n-1)} < \frac{2^d S_{CE}}{2S_{CE} + 2^d}, \tag{7}$$

where T_H, T_L are the upper and lower thresholds, and S_{CE} is the contrast enhancement factor used by EDLS. Note that $\lfloor A, B \rfloor$ is a floor function of number A, which rounds down A to the nearest multiple of significance, B. The difference in the power reduction achieved by EDLS and EDLS-d in the DCE mode is equal to $(\frac{P_B}{S_{CE}} - \frac{P_B}{S'_{CE}})$, and thus it is bounded by $\frac{2^{n+1} P_B}{2^d(2^n-1)}$. Note that the backlight power consumption penalty due to the approximation made by EDLS-d is usually much less than the worst-case value, we have just calculated.

Once the parameters of hardware EDLS-d have been determined for a given display specification, we can make a further compromise between the energy reduction from the backlight and the area overhead for the EDLS functional blocks. Hardware EDLS-d slightly reduces the energy saving achieved by backlight dimming, and results in some minor inconsistency in the inter-frame saturation ratio in video applications, where d is smaller and the area saving is

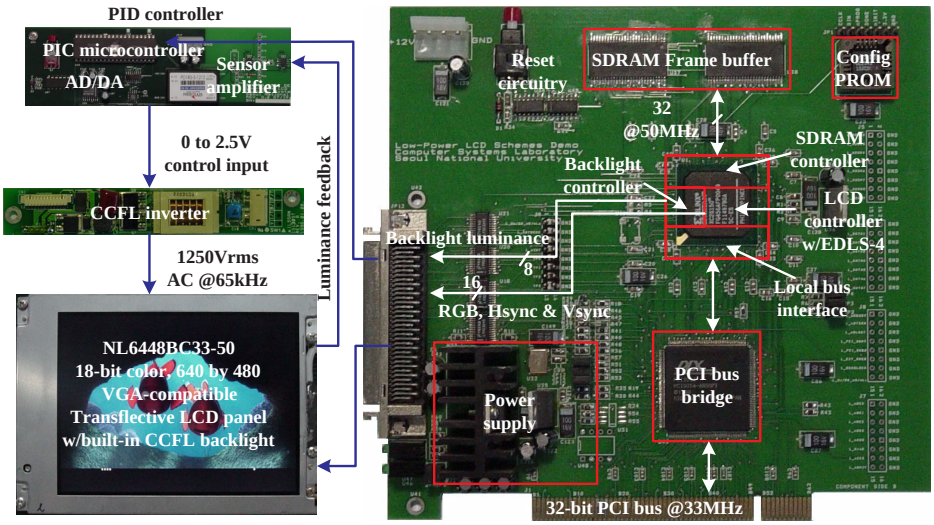


Figure 3: Block diagram of an application-transparent EDLS prototype.

large. Note that EDLS-*d* is also applicable to the software-oriented approach, but we would expect no enhancement in either the energy or time overhead because all the operations must be performed by the same datapath resources in the CPU.

4 Experimental results

4.1 Implementation

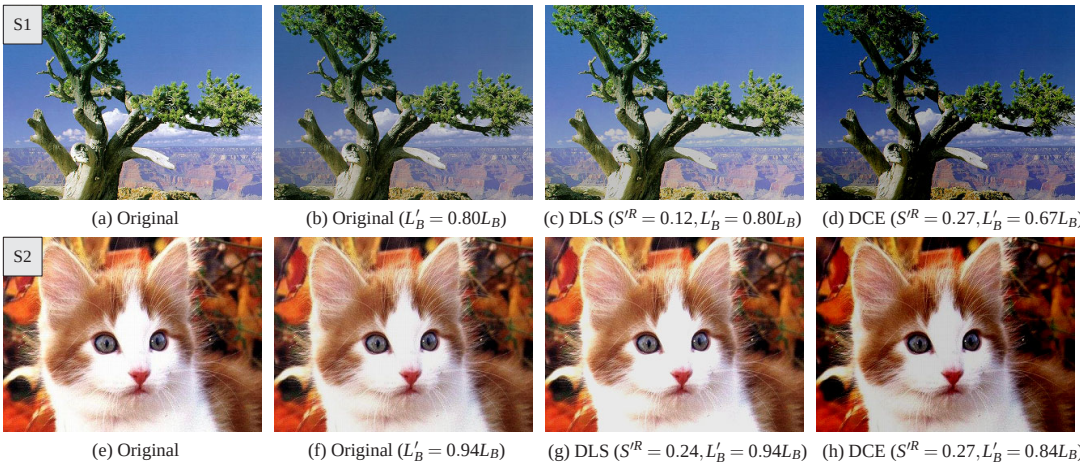
We have implemented a VGA (Video Graphics Adapter) compatible LCD controller with application-transparent hardware EDLS-4 at a 640×480 screen resolution and 16-bit color depth. It outperformed a software EDLS which we also implemented for comparison.

The architecture of the prototype is shown in Fig. 3. The implementation includes an FPGA (Field-Programmable Gate Array) EDLS-enabled LCD controller, a PCI bus interface, a frame-buffer memory and a PID (Proportional, Integral and Differential)-controlled CCFL backlight inverter. The LCD controller is equipped with two Samsung K4S641632D SDRAM devices for the frame-buffer memory. The backlight system of an NEC6448BC33-50 10.4" TFT LCD panel consumes about 8.1W at its maximum luminance. Thanks to an effective compaction of the EDLS algorithms, it was possible to mount the EDLS-4 on a small and low-cost Xilinx Spartan-II FPGA, the XC2S-150FG456. Since the Linux operating system tends to have a slow response time due to its heavy locking mechanism, a 1ms timer interrupt to activate the PID controller is not feasible. Instead, we used a simple RISC microcontroller, the PIC16C74A from Microchip Technology. The EDLS-enabled LCD controller is supported by a VGA-compatible Linux driver (corresponding to the Linux kernel 2.4.19). The resulting platform is capable of utilizing the EDLS capability for all kinds of applications that use the LCD display, without the need for any modification.

4.2 Energy reduction and image quality

EDLS reduces the energy consumed by the backlight. We will now compare the power reduction achieved by software EDLS and hardware EDLS-4. There is no reason to use software EDLS-*d* where *d* is smaller than the original color depth, and hardware EDLS-*d* with $d = 4$ is a reasonable configuration considering hardware complexity. We expect more power reduction by software EDLS under fixed S^R because $S^R > S'^R$, and thus we may use dimmer backlight with software EDLS. In other words, EDLS-*d* produces an image quality no worse than EDLS, but achieves less power saving.

Fig. 4 shows image quality of EDLS-4. Fig. 4(a) and Fig. 4(e) are the original images and Fig. 4(b) and Fig. 4(f)



(a) Original (b) Original ($L'_B = 0.80L_B$) (c) DLS ($S'^R = 0.12, L'_B = 0.80L_B$) (d) DCE ($S'^R = 0.27, L'_B = 0.67L_B$)

(e) Original (f) Original ($L'_B = 0.94L_B$) (g) DLS ($S'^R = 0.24, L'_B = 0.94L_B$) (h) DCE ($S'^R = 0.27, L'_B = 0.84L_B$)

Figure 4: Still images before and after application-transparent hardware EDLS-4 ($S^R = 0.3$).

Table 1: Average and variance of the backlight power saving for a movie stream (%).

Power saving	EDLS (DLS)	EDLS-4 (DLS)	EDLS (DCE)	EDLS-4 (DCE)
Average	20.6	18.9	32.3	32.0
Variance	38.3	43.2	58.6	58.7

are unprocessed images with a dimmed backlight. We can see that the dimmed backlight reduces both brightness and contrast. Fig. 4(c) and Fig. 4(g) were produced by EDLS-4 in DLS mode with the same amount of backlight dimming. EDLS-4 restores both brightness and contrast to their original values. We can hardly see any image distortion although it is present. Finally, Fig. 4(d) and Fig. 4(h) are the results of using EDLS-4 in the DCE mode with more backlight dimming and hence reduced power consumption. Although their brightness is less than that of the original, the contrast has been recovered.

Finally, we applied both software EDLS and hardware EDLS-4 to a movie clip, namely a trailer for the movie 'Bad Boys 2'. We summarize the average power reduction and variance, where $S^R = 0.2$, in Table 1. It shows that EDLS-4 produces significant results in a real situation.

4.3 Power, delay and area overhead

While EDLS significantly reduces power consumption by the backlight, it involves a power, delay and area overhead that takes place in other components. This overhead is primarily determined by the screen resolution, the refresh rate, and the color depth. Typically, EDLS needs to cope with a 30Hz refresh rate for quality movie streams.

Thus application-transparent software EDLS occupies a 36.9MB/s data bandwidth to refresh the histogram at 640×480 resolution, with a 16-bit color depth. Even though it is not an expensive setting in modern applications, application-transparent software EDLS requires over 300% utilization of a 733MHz XScale processor. That would imply a 240mW power overhead if it was feasible. This shows that application-transparent software EDLS is only applicable to small screen resolution.

On the other hand, power and area overhead for hardware EDLS-d are not sensitive to screen resolution; it is only affected by the value of d. Table 2 summaries the overhead for histogram construction, which is a primary concern of hardware EDLS-d. Image enhancement is not a serious overhead in hardware EDLS-d, but it takes most of the CPU and memory resources in software EDLS. Image processing requires additional datapath resources such as multipliers, adders and comparators; however, just three 13-bit precision integer multipliers can manage the image processing for 16-bit color. The multipliers for image processing require just 77 slices, which corresponds to an area overhead of 9%. Furthermore, the power consumption of the LCD controller prototype is only increased by 6mW due to image processing.

Table 2: Power and area overheads of EDLS-*d*.

EDLS-*d*	# slices	Equivalent gates	FPGA core (mW)	LCDC total (mW)
w/o EDLS	926	64,656	229	1,313
EDLS-1	1,033	66,596	239	1,328
EDLS-2	1,121	68,356	248	1,340
EDLS-3	1,266	71,372	260	1,361
EDLS-4	1,574	77,578	284	1,392

5 Conclusions

We have introduced extended DLS (EDLS) as a framework for backlight power management of transflective LCD panels for quality multimedia applications powered by batteries. We have explored the EDLS design space in which the application transparency and hardware-software partitioning exhibit trade-offs in terms of energy reduction, energy overhead, performance penalty and image quality. As there is a further trade-off between energy reduction and area overhead, we have proposed compact EDLS, called EDLS-*d*, which approximates the image processing algorithm used by EDLS and thus dramatically mitigates the explosion of gate count in the LCD controller, while achieving nearly the same backlight power savings as the exact EDLS system. Our hardware EDLS-*d* has perfect application transparency, and thus supports all kinds of multimedia applications running on Linux without any modification. We demonstrate that the EDLS-*d* with $d \geq 4$ results in significant power reduction without any appreciable degradation of image quality.

References

[1] I. Choi, H. Shim, and N. Chang, "Low-power color TFT LCD display for hand-held embedded systems," in *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 112 – 117, August 2002.

[2] I. Choi, H. S. Kim, H. Shin, and N. Chang, "LPBP: low-power basis profile of the java 2 micro edition," in *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 36 – 39, August 2003.

[3] F. Gatti, A. Acquaviva, L. Benini, and B. Ricco, "Low power control techniques for TFT LCD displays," in *Proceedings of International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 218 – 224, October 2002.

[4] W.-C. Cheng, Y. Hou, and M. Pedram, "Power minimization in a backlit tft-lcd display by concurrent brightness and contrast scaling," in *Proceedings of the Design, Automation and Test in Europe*, pp. 16 – 20, February 2004.

[5] S. Pasricha, S. Mohapatra, M. Luthra, N. Dutt, and N. Subramanian, "Reducing backlight power consumption for streaming video applications on mobile handheld devices," in *First Workshop on Embedded Systems for Real-Time Multimedia*, October 2003.

[6] *Display Modes (Transmissive/Reflective/Transflective)*. Sharp Microelectronics of the Americas, http://www.sharpsma.com/sma/Products/displays/AppRefGuide/DisplayModes.htm, 2002.

[7] T. Tsukuda, *TFT/LCD: Liquid-Crystal Displays Addressed by Thin-Film Transistors*. Taylor & Francis, June 1996.