A Power Optimized Display Memory Organization for Handheld User Terminals

Lieven Hollevoet, Andy Dewilde, Kristof Denolf, Francky Catthoor*, Filip Louagie
IMEC
Kapeldreef 75
B3001 Heverlee, Belgium
lieven.hollevoet@imec.be

Abstract

Today's handheld devices become more and more multimedia capable. One subsystem of a multimedia terminal that accounts for a considerable amount of the total power consumption is the display unit. The backlight is the major culprit there. As new display units without backlights emerge, the data transfers required to put data on the screen start using up an increasingly important part of the platform's power.

We have examined a novel system view that allows for power savings by decreasing the required number of memory accesses to put a frame on the screen. A two-step optimization method for existing platforms is presented. Measurements on a multimedia application show that, on average, power savings of 72% can be obtained on the display related memory accesses.

For the proposed optimizations methods to work, it is important that both hardware and software designers become aware of the impact their design-time decisions have on the final power consumption of a system.

1. Introduction

The market for handheld devices has continued to grow ever since the breakthrough of cellular phones. Personal Digital Assistants (PDA's), mobile phones and portable multimedia players are just a few examples of battery powered devices that have a display unit as their main user interface.

On platforms equipped with standard color Liquid Crystal Displays (LCD's), the backlight unit consumes a large part of the total power budget (typically around 30% [1]). Recent developments in displaying techniques give rise to displays that can do without a backlight. In so-called Organic Light Emitting Diode (OLED) displays, the pixels itself generate light [2] [3].

With the removal of the backlight, other parameters become important when determining the power consumption of a display unit. One of the more important factors is the number of memory accesses that are performed to put data on the screen. This is explained in the next paragraphs.

In nowadays systems, the display unit is seen as a separate building block of a multimedia terminal. From the application point of view, the display is considered to be a black box. To put an image on the display, typically data is first written into a so-called video buffer or frame buffer. This buffer is located in the main memory of the platform [4] [5] or in a memory unit contained within the LCD controller [6]. The frame buffer is used as the data source for the display driver unit. The driver unit scans the frame buffer and generates the control signals for the display panel.

The memory accesses and bus traffic involved in the display update process consume a considerable amount of power. For every frame update, the complete content of the frame buffer is transferred from the frame buffer through a bus to the display unit. In a typical system where the frame buffer is located in main memory, this part of the system accounts for 28% of the total power [1].

Using a frame buffer allows designers to make abstraction of the display unit. In such a system view, the LCD controller takes care of the hardware specific actions required to put an image on the screen. As long as energy consumption or memory footprint are not an issue, this allows for a simpler design procedure. But in battery powered embedded systems, the overhead incurred is quite large. In such applications designers can obtain important energy savings by using a power optimized display solution.

We have applied a two-step optimization method for building a multimedia terminal that eliminates redundant memory accesses when interfacing to the display unit. In the first step, the frame buffer access is optimized. Only parts of the screen that are modified between frames are written to the frame buffer. This optimization reduces the

* Also professor at K.U.Leuven, Belgium

1530-1591/04 $20.00 (c) 2004 IEEE

number of memory accesses between processor and frame buffer. Hence, the power consumption of the system is reduced. In a second step, a new hardware architecture that eliminates the frame buffer is presented. This second step enables a further reduction of the power consumption. In the most optimal case of a static image between consecutive frames, all memory accesses are eliminated.

The proposed setups are evaluated on a test platform of which the power consumption can be measured. Both dedicated test software and an existing compressed video decoder application are used to evaluate the impact of the optimizations.

This paper is further organized as follows: first, in Section 2, the commonly used architectures are presented and optimizations are proposed. Section 3 shows the measurement setup and the results. In Section 4, the software optimization of a compressed video decoding client is presented. Section 5, finally, concludes the paper.

2. Architecture study and proposal

A handheld multimedia terminal is a portable, battery powered device with a display as main user interface. This section first presents the existing interface methods between a system and its display. Then, it describes a two-step optimization method to obtain a maximal power gain.

2.1. Exploration of existing systems

In traditional systems, the memory accesses cause a significant power consumption, both in the memory chips themselves as in the busses that have to be driven to transfer the data between the chips. However, both hardware designers and software programmers are still not concerned enough about the implications that memory accesses have on the power budget.

An application that generates screen data writes its output in the so-called frame buffer (Figure 1, transfer A). This buffer is located in main memory of the platform or inside the LCD controller unit.

On systems where the LCD controller is located in the processor (e.g. StrongARM and XScale processor based systems), the frame buffer is integrated in main memory. Refreshing the LCD panel requires transferring the data from the main memory buffer through the memory bus to the display controller (Figure 1, transfers B and C). This is done using a dedicated Direct Memory Access (DMA) controller contained within the display controller. The content of the frame buffer has to be transferred to the LCD controller at a speed that is determined by the refresh rate of the LCD (e.g. 60 Hz [7] or 75 Hz [6]). This traffic places a heavy burden on the processor bus, and hence, on the power consumption of the platform. The advantage of using a processor-integrated LCD controller is the reduction of the chip count of a system, an important factor in the design of portable devices. Disadvantages are the load on the main memory bus and the power consumption caused by the bus accesses.

Other systems use an LCD controller with integrated frame buffer [6]. In such a system, the LCD controller is a separate building block containing a frame buffer that is accessible to the processor. This setup removes the load involved in refreshing the LCD panel from the main memory and system bus, since the data traffic is routed from the internal frame buffer to the LCD through a separate bus. The advantage of this way of working is that the main memory bus is free for other subparts of the system and that the power consumption of the bus accesses, required to refresh the LCD panel, is reduced. The dedicated bus is shorter and less complex than the system bus and hence, it will pose a smaller load on the system's power consumption.

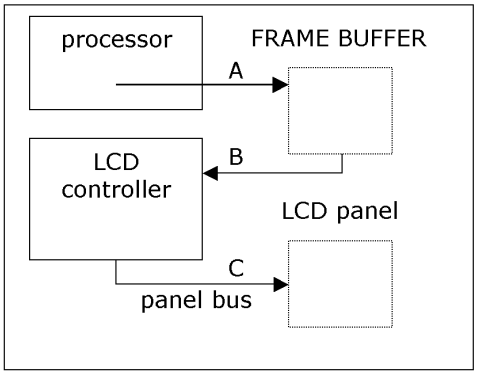

Figure 1. Block scheme of a traditional system with LCD display

Various methods of reducing the power consumption caused by frame buffer related memory accesses have been proposed, such as variable-duty-ratio refresh and dynamic color depth control [1]. Those methods reduce the power consumption by exploiting specific characteristics of LCD's with relation to the human eye. They leave the frame buffer present as a double or sometimes even triple buffered scheme. We present a method for reducing the number of memory accesses that is taking advantage of the content of the data that is displayed on the screen. This is explained in the next section.

2.2. Content dependent optimizations

It seldom occurs that, between two frames, every pixel of the new frame differs from the pixel at the same position in the previous frame. If the system can only update those pixels that have changed when comparing the new frame to the current frame, it can avoid many unnecessary memory accesses.

A broad range of applications can take advantage of selective screen updating. Video decoding applications can be optimized to only update those parts of the screen that have changed. Applications requiring user input, such as calendars, spreadsheets, and word processors can take advantage of the fact that user input is most of the time limited to a small part of the screen area.

In the first optimization step, we propose a modification to the software that generates the data for the frame buffer. Instead of writing the entire buffer from the application, the software should only update the pixels that have changed. This approach yields a substantial reduction of the number of memory operations.

In order to apply this optimization to existing systems, it has to be taken into account that it will only work correctly on systems that use a single frame buffer. Systems that use a double frame buffer to avoid display artifacts on the LCD first will have to be modified so that a single frame buffer can be used. This requires some kind of synchronization between the application and the controller that generates the signals for the LCD panel. In practice the production of new data in the data source should never be slower than the read by the display controller to the display unit. A detailed description of processor to display synchronization solutions is out of the scope of this paper.

2.3. Integrating a pixel-level memory display

The second step of the optimization focuses on the actual hardware architecture. In existing video interface methods, the refresh of the display generates a large part of the total memory traffic and bus usage. Refreshing a 640 x 480 pixel, 60 Hz display that uses 18 bits to represent one pixel, generates a 316 Mbits per second load in the memory units and bus between the frame buffer and the LCD panel. The consequences in power consumption are considerable: 6% of the total system power is needed to drive the LCD panel bus, the frame buffer accounts for 19% of the total power [1].

For displays that do not need such refreshing, an important power saving can be obtained. This is where the OLED displays can provide a solution.

Using novel integration techniques, researchers have developed OLED displays where memory is integrated at pixel level [8]. The status of pixels in the OLED panel is determined by the value that is written in a memory cell. In this way, the only video related memory that is required is the memory that exists on the display panel itself. Switching a pixel on or off occurs by writing a certain value to the corresponding memory location on the OLED panel. In this way, the frame buffer can be avoided and the refreshing of the pixels required for LCD panels is eliminated. This allows for important power savings. The high throughput bus between the frame buffer and the LCD panel is not required any more, and the frame buffer does not need to be allocated in main memory. Using less memory will become important in future systems, when the power required to keep a memory chip active will start to be dominant.

By combining the content dependent optimizations discussed earlier, with a pixel-level memory display, maximal power savings can be obtained. Not only are the frame buffer and the related memory access eliminated, but also the memory transfers to the on-display memory are reduced to the minimum, depending on the content of the screen. Only this systemwide approach allows to achieve the absolute minimum number of data transfers.

3. Measurements

Using a readily available prototyping platform, we were able to measure on real hardware the impact of our proposed optimizations. Although the proposed OLED display is not available to us, we were able to draw conclusions on the obtained measurement results by extrapolating them. This section describes the hardware and software involved in the measurement setup and presents the results of our measurements.

3.1. Hardware setup

The test platform is an XSCALE PXA250-based prototyping platform with 64 MB of SDRAM and a Tvia CyberPro 5205G-50 video processor. The video buffer of the Tvia is mapped in the memory space of the processor. This allows us to emulate an OLED display if we consider the Tvia video processor to be a display unit. The operating system of the setup is Linux. The Tvia video processor generates a PAL video output signal. We have used this signal to display the output of the test setup on a standard TV screen.

To determine the power consumption of the display related memory accesses, we have monitored the current drained by the entire platform at the power supply. The power required by the actual display unit (the TV screen) is not taken into account.

3.2. Test software

Three different software applications are used. The first two are dedicated test applications. The third application is an MPEG-4 video decoder that has been developed in our research group. This decoder allowed us to verify the predicted gain in a real-life application and is described in detail in Section 4.

The test applications are written to be able to measure the impact of every step of the proposed optimization. In a first test case, the influence of optimizing the frame buffer access is measured. The second test case allowed us to verify the direct video memory access optimization.

In the first test case, we measured the influence of the content dependent optimizations with the following setup. An application running on the processor of the test platform generates screen data. This data is written into a frame buffer that is allocated in main memory of the system (Figure 2, transfer A). To emulate the frame buffer to LCD panel transfers, the application copies the frame buffer content to the Tvia video controller (Figure 2, transfers B and C). Eight frames per second are generated and written to the frame buffer (640 x 480 pixels, 16 bits per pixel). The frame rate has been selected in order not to overload any hardware system limits, such as bus capacity or processor load. The transfers from the frame buffer to the Tvia also occur eight times per second. The impact of the memory transfers on the power usage is evaluated in five different cases, ranging from updating the entire display to doing no update at all.

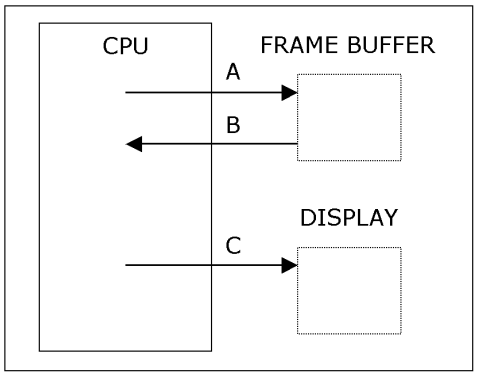

Figure 2. Block scheme of the test setup with frame buffer

In the second test case, the possible energy savings by integrating an OLED display with per-pixel memory are evaluated on the same hardware setup. The frame buffer is eliminated and the test application writes its data immediately to the Tvia video processor. Again, the impact of the display related memory accesses on the power usage is evaluated in five different cases, ranging from a full screen update to doing no update at all. The block scheme of the test software shows that the only display related transfer is the one from the CPU to the Tvia video processor (Figure 3).

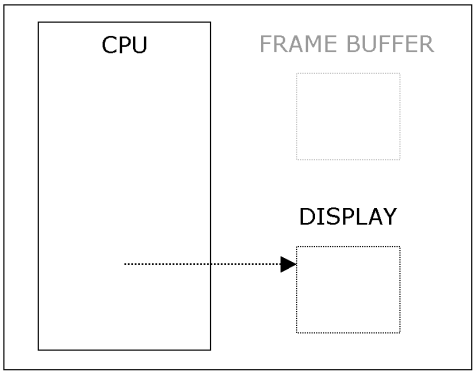

Figure 3. Block scheme of the test setup with per-pixel memory display emulation

3.3. Test results

A comparison of the measurement results between the non-optimized setup (with frame buffer and full screen update) and the setup where the per-pixel memory is emulated (no frame buffer and a static image) shows a difference in power consumption of 770 mW. As an indication, the idle power consumption of the platform is 5.3 W. It is important to understand that, as low power design techniques get more and more accepted, the CPU and its peripherals will start consuming a smaller amount of the total power. The memory related power consumption is not expected to drop that fast. This means that the impact of the memory operations on the power consumption will even become larger in future systems.

In order to represent the measurement results, all further figures are expressed as a percentage of the maximal difference in power consumption, 770 mW.

For the first test case, using the optimized frame buffer access, the possible gain in power consumption ranges between 0% (when all pixels are updated) and 71% (when no pixels are updated) compared to the original power consumption of the display related memory accesses. On average, 36% power is saved. (Figure 4).

The second test case, with per-pixel memory display and optimized access, allows for power savings between 45% (the entire screen is updated) and 100% (in case of a static

image). On average, a platform based on this architecture uses 72% less power for the display related memory accesses. (Figure 4).

3.4. Interpretation of the test results

For the interpretation of the results considering existing LCDs, it has to be taken into account that the refresh rate of our "display unit" (the Tvia) is lower than what is used in current systems. The effect of that is that the absolute measured gain by introducing the new display will be even larger in existing systems. The high throughput on the LCD panel bus due to the high refresh rate will cause a higher power drain than the energy consumption that occurs in our test setup. The absolute value of the power saving in a device also depends on the frame size, the frame rate and the system architecture (e.g. the type of RAM, the length of the system busses, ...).

Assuming that on average 50% of the pixels need to be refreshed between two frames, the obtained power savings are 277 mW (36%) for the content dependent optimization (first test case) and 554 mW (72%) for the version with the per-pixel memory display (the second test case).

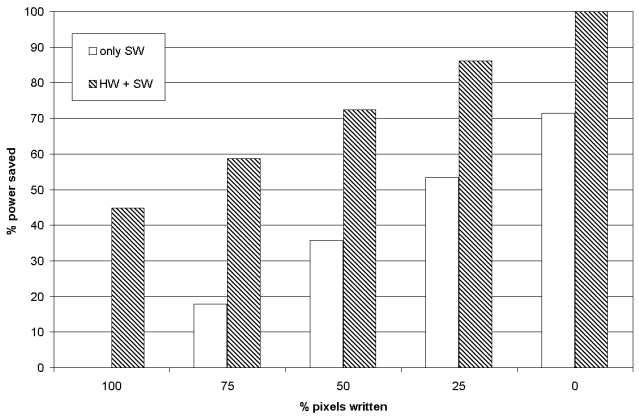

Figure 4. Power gained in the display related memory accesses

4. Optimization of a video decoder

Measurements on a realistic MPEG-4 video decoder application that we have run on the XScale platform confirm the above results. The proposed testbed (Table 1) contains video at different bitrates and with varying degrees of motion: from *mother and daughter*, a head and shoulders type sequence to *calendar and mobile*, a highly complex sequence with multiple, heterogeneous motion.

sequence	bitrate (kbps)	blocks (%) intra	inter coded	inter skipped
mother & daughter	101	1.0	21.5	77.5
mother & daughter	287	1.0	30.4	68.6
foreman	334	1.4	69.3	29.3
foreman	935	1.4	79.9	18.8
calendar & mobile	1641	1.0	92.9	6.0
calendar & mobile	3998	1.0	95.5	3.5

Table 1. The amount of blocks requiring a screen update varies with the complexity of the video sequence

All sequences are compressed using an MPEG-4 Simple Profile [9] coder which uses a hybrid video block-based algorithm exploiting temporal and spatial redundancy in subsequent frames. Three types of 8x8 blocks can occur in a frame: (i) intra blocks contain only independent texture information, (ii) inter blocks use motion compensated prediction and (iii) skipped blocks, a special kind of inter blocks, in which none of the pixels change compared to the previous frame. Hence these last blocks do not need a screen update and can exploit the two proposed optimizations.

Table 1 lists the relative amount of intra, inter and skipped 8x8 blocks in the MPEG-4 Simple Profile video sequences. The amount of skipped blocks varies from virtually zero for highly complex to almost 80 % for low motion video. For an average sequence (*Foreman*), around 75 % of the image is updated. According to Figure 4, this leads to respectively 18 % and 59 % power saving for the two optimizations.

5. Conclusions

Display related memory accesses consume an important part of the power budget on portable devices. In future systems, their influence is only expected to increase.

The proposed two-step optimization enables valuable energy savings, increasing the battery autonomy. The measurements show that, using this optimization, average energy savings can be obtained up to 72% of the power that is used in traditional setups for the display related memory accesses. Applications on handheld devices that require user input are expected to allow for even better results. In such an application, only a fraction of the screen is updated between frames.

Elimination of the frame buffer also reduces the required main memory size of a platform. This enables designers

to switch off memory banks that are not in use, which reduces the required memory footprint and further optimizes the power consumption.

New display types with per-pixel memory will enable integration of the discussed techniques in future multimedia terminals. Careful design of both the hardware architecture of a system and the software that runs on the platform is required to benefit from the proposed optimizations.

6. Acknowledgements

We would like to thank Lode Nachtergaele and our team members for the interesting discussions and valuable remarks.

References

[1] I. Choi, H. Shim, and N. Chang. Low-Power Color TFT LCD Display for Hand-Held Embedded Systems. In *ISPLED*, Aug. 12-14, 2002.

[2] http://www.kodak.com/US/en/corp/display/overview.jhtml.

[3] G. Rajeswaran, M. Itoh, M. Boroson, S. Barry, T. Hatwarand, K. Kahen, K. Yoneda, R. Yokoyama, T. Yamada, N. Komiya, H. Kannoand, and H. Takahashi. Active Matrix Low Temperature Poly-Si TFT/OLED Full Color Displays: Development Status. In *Society for Information Display*, May 2000.

[4] Intel PXA26x Processor Family Developer's Manual, Mar. 2003. http://www.intel.com.

[5] Intel StrongARM SA-1110 Processor Family Developer's Manual, Oct. 2001. http://www.intel.com.

[6] S1D13704 Embedded Memory Color LCD Controller Technical Manual, 2002. http://www.erd.epson.com.

[7] LG PHILIPS LP064V1 Liquid Crystal Display Specifications, May 2001. http://www.lgphilips-lcd.com.

[8] J. L. Sanford, E. S. Schlig, O. Prache, D. B. Dove, T. Ali, and W. E. Howard. Technology and design of an active matrix OLED on crystalline silicon direct view display for a wrist watch computer. In *International Symposium on Optical Science and Technology*, Aug. 2001.

[9] ISO/IEC JTC1/SC29WG11 14496-2, Information technology - Generic coding of audio-visual objects - Part 2: Visual Amendment 1: Visual Extensions, Dec. 1999. N3056, Maui.