SYSTEM DESIGN TRICKS FOR LOW-POWER VIDEO PROCESSING

Jonah Probell
ARC International
San Jose, CA, USA
+1 (408) 887-3350
jonah.probell@arc.com

0. ABSTRACT

Low-power circuit design alone will not achieve the battery life requirements of future handheld and mobile multimedia devices. Algorithm and system design tricks must be employed, in software and hardware, to reduce energy consumption. A software programmable, configurable, video optimized processor is best suited for such low-power media system designs.

Some tricks are: invoking sleep mode after processing simple frames, using image enhancement algorithms, balancing on-chip SRAM with off-chip SDRAM accesses, sizing the SDRAM data bus width appropriately, including processor and SDRAM dice within a single package, and tiling data within reference frame buffers.

Contact the author to learn more about these and other power saving tricks for portable video systems.

1. INTRODUCTION

Low energy consumption, in order to extend battery life, is a top concern for handheld video product makers and hence so for the makers of video processor chips. As shown in equation 1, the most power consuming components in portable video products are displays, video processor chips, and SDRAM memories.

$$P_{system} \approx P_{display} + P_{video\_processor\_chip} + P_{SDRAM} \qquad (1)$$

The design of hardware and software within video processors affects the energy consumed by the processor chip as well as by SDRAM and the display.

Counts of processor cycles and SDRAM memory accesses were gathered for each frame of the decoding of several H.264 video test sequences. The sequences were provided by BDTI and decoded on an ARC Video FPGA system from ARC International. The power consumed by the video processor was determined using Synopsys Power Compiler with SDF annotation from a netlist layout done with Cadence tools and a SAIF switching activity file from simulation of H.264 video decode. The power consumption in the full video processor chip and SDRAM chip within a digital video system was modeled in a spreadsheet analysis that included parameters for processor and SDRAM clock rates, SDRAM interface data bus width, access latency, SDRAM architecture, RAS delay cycles, DMA controller data tiling, I/O voltage, inter-chip net capacitance, processor power consumption per cycle, processor chip die area, and process technology leakage. SDRAM power consumption models were provided by Micron. Display power was not modeled.

Section 2 discusses a method for reducing the power consumed within the video processor. Section 3 discusses a method for reducing the power consumed by the display. Sections 4 through 7 discuss methods for minimizing the power consumed due to memory accesses. Power savings are expressed as a percentage of that used for video decode, by the video processor chip and SDRAM, in a typical low-power system that does not incorporate the method described.

2. SLEEP MODE

Power consumption within a processor comes from the momentary short-circuit current between power and ground through a gate's transistors while it switches state, the charging (and discharging) of wire nets between gates, and a constant leakage between power and ground through every transistor in the chip.

Video is displayed at a regular frame rate, but some video frames require less processing than others. A processor can be programmed to enter a power-saving sleep mode after finishing an easy frame while waiting until it is time to process the next.

In sleep mode the clock to almost all registers within the processor is disabled. This prevents the toggling of almost all gates, which eliminates nearly all gate switching and net charging power within the processor.

In advanced chip designs, with multiple voltage islands, a deep sleep mode can be implemented in

which most of the processor is powered down to eliminate leakage power consumption.

Processors are generally woken from sleep mode by an interrupt. In a video system this can be done by an interrupt generated at the desired frame rate by a real-time clock.

By entering sleep mode after completing the decode of easy frames, idle cycles can be eliminated, saving 3% of system power for video decode at the expense of system design complexity for clock gating and/or power-down circuitry. Though hardwired logic can also sleep, the cost calculation of when to sleep is different for every system and can be optimized in a software programmable processor

3. IMAGE ENHANCEMENT

The component that consumes the most power in video systems is the display. It is also the component in which reducing power consumption is the most difficult without degrading the user experience.

Using digital processing to enhance the clarity and brightness of the image can enable an acceptable apparent quality with a lower intensity of illumination.

Brightness can be digitally enhanced by applying a transfer function, such as that shown in equation 2, to the luma value for each pixel to be displayed.

$$\text{luma}_{\text{disp}} = \text{luma}_{\text{enc}}\left(1 + \frac{(256 - \text{luma}_{\text{enc}})}{256}\right) \qquad (2)$$

This function increases the luma value of mid-intensity pixels while keeping black levels dark and whites bright. The transfer function is shown in figure 1.

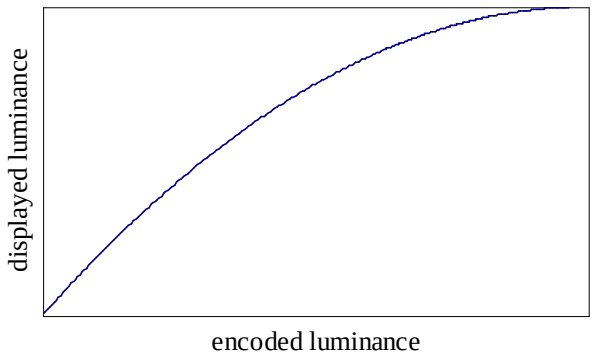


Figure 1: A luma transfer function for digitally enhanced brightness.

Applying a sharpness or local contrast enhancement algorithm to video display frames can also improve the perceived clarity of display images and allow lower illumination intensity.

Performing an edge strengthening transformation also improves visual clarity with low intensity of illumination, especially for text. A Sobel gradient or Prewitt algorithm can be used to detect the direction of the edge, if one is present, at each pixel.[1] Based on the difference between luma and chroma values across the edge in the normal direction, the difference can be increased to strengthen the edge.

$$\begin{aligned} \text{diff}_{\text{disp}} = \; & \textit{if}\,(\text{diff}_{\text{enc}} < \text{T})\{ \quad \text{diff}_{\text{enc}}\} \\ & \textit{else} \qquad\quad \{\text{S} \cdot \text{diff}_{\text{enc}}\} \end{aligned} \qquad (3)$$

In equation 3, T is a threshold value, below which the difference across an edge is unlikely to represent a real edge in the encoded video material, and S is a scaling factor. Equation 3 is an edge strengthening algorithm with an edge difference response as shown in figure 2.

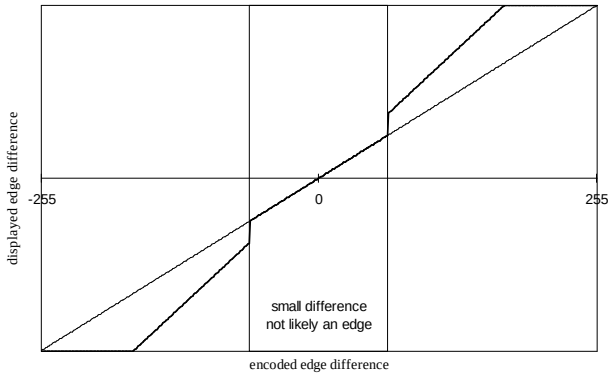


Figure 2: Edge difference transfer function for digital edge enhancement.

Using digital enhancement of brightness, contrast, sharpness, and edges can improve the subjective clarity of a video image, allowing acceptable apparent quality at reduce intensity of illumination in the display. This decrease in display power is achieved at the expense of a smaller increase in power consumed by the video processor to execute the image enhancement algorithm.

For video products based on MPEG and other standards, image enhancement algorithms distinguish one product from another. Product differentiation through proprietary algorithms is only possible with a software programmable video processor.

4. SRAM VS SDRAM

Static RAM (SRAM) is used on processor chips for caches and other local data storage. Data that does not fit into on-chip SRAMs in the processor chip must be

stored in off-chip synchronous dynamic RAM (SDRAM). Increasing the size of on-chip SRAMs improves the ability for the video processor to store required data structures in local buffers thereby decreasing the number of off-chip SDRAM accesses. In video processing algorithms, many small data structures, such as pointers and semaphores, tend to be accessed most frequently and certain large data structures, such as reference frame buffers, are accessed less frequently. In figure 3 the bars represent the total size of one or more data structures in local SRAM on the video processor chip for each of six hypothetical scenarios. The line represents the frequency of SDRAM accesses for each scenario.

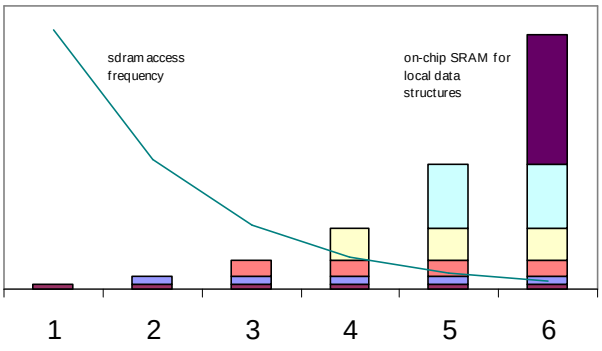


Figure 3: Scenarios of on-chip data storage vs SDRAM bandwidth.

SDRAM accesses consume much more power than on-chip SRAM accesses. Furthermore, SDRAM accesses take more time than on-chip SRAM accesses and therefore increase the number of cycles required for processing each frame, which increases the required processor clock rate, which increases the video processor power consumption.

SRAMs require more die area per stored bit than DRAMs, adding to processor chip area and increasing leakage power consumption and chip manufacturing cost. Standard (6T) SRAM technology uses 6 transistors per bit. For large on-chip SRAM capacities significant die area and SRAM power consumption on the processor chip can be significantly reduced by using 1 transistor (1T) SRAM, though this will increase chip manufacturing cost.

The optimal trade-off between on-chip SRAM and off-chip SDRAM accesses depends on the type of SDRAM used, the SDRAM controller design, and the technology used for the processor chip. Making the optimal trade-off for a system requires a configurable processor. Using 1T SRAMs on the processor chip to hold all video processing data structures except for video decoder reference frame buffers can save 15% of system power for video decode.

5. DATA BUS WIDTH

When the processor in the video chip accesses the SDRAM, it might have to stall for an earlier transfer to finish before being granted bus access. Stall cycles increase the number of cycles required to process each video frame and therefore increase the clock rate at which the processor must run in order to complete each frame during the frame period.

Power is consumed within the SDRAM chip during each data transfer cycle. By doubling the width of the SDRAM data bus, the number of data transfer cycles is approximately halved, yielding lower SDRAM power consumption. Furthermore, the number of stall cycles within the video processor chip is reduced, yielding a lower clock rate requirement for the processor and further saving power at the expense of increased chip packaging and board design expense.

The correct choice between a 16-bit and a 32-bit SDRAM interface can save 15% of system power for video decode. The freedom to make this trade-off is only available in a configurable video processor.

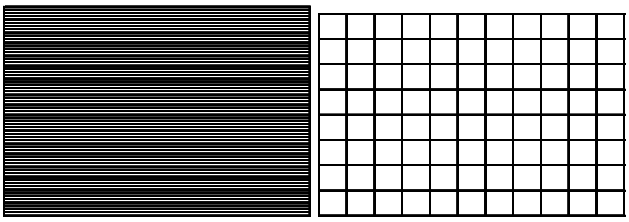
6. SYSTEM IN PACKAGE

Optimal manufacturing process technologies for SDRAM are different than for processor logic and the two are almost always manufactured as separate chips. However, both chips can be placed into the same package. This reduces the amount of effective capacitance in trace metal to be charged for the toggling of signals between chips, which reduces chip I/O power consumption. This benefit is achieved at the expense of greater packaging complexity and cost. Systems in package or stacked die packaging may still yield a total system cost saving by reducing the number of pins coming out of the package, the number of packages to be soldered to the board, and the number of traces on the board.

Because of the reduced wiring capacitance, a wider SDRAM data bus and higher clock speed can be accommodated in a system in package design (SiP) than in a design with discrete components. To optimize a processor for an SiP requires a configurable processor. A system in package can save 8% of system power for video decode over a design based on discrete memory and processor components.

7. FRAME BUFFER TILING

Video reference frame buffers are typically too large to store within the video processor chip and are stored in SDRAM instead. Performing motion estimation or compensation block reads requires accessing many different lines of video data from reference frames. If reference frame data is stored in SDRAM in raster scan order, as shown in frame a of figure 4, then the lines of video data will tend to be stored in different SDRAM rows. Reading a block of data would require many switches of active SDRAM rows, each of which consumes power. Furthermore, SDRAM active row switching adds to memory access delay, thereby increasing the processor clock rate requirement and video processor power consumption.



(a) (b)

Figure 4: Raster and tiled frame buffer order.

If reference frames are stored in SDRAM in rectangular tiles, as shown in frame b of figure 4, such that each SDRAM row contains as large of a tile as it can fit then each reference frame block access will require accessing data locations in as few SDRAM rows as possible. This minimizes SDRAM row switching and its inherent delay and power consumption.[2]

Tiling frame data would require many extra clock cycles if performed in software, but can be implemented in hardware with no extra cycles required. A video-optimized processor with a two-dimensional DMA controller can minimize the number of software instructions required to locate blocks of data within reference frame buffers in memory. This yields power savings in software cycles and even greater power savings in the reduced number of SDRAM accesses at the expense of only a few thousand extra gates.

In a typical low-cost SDRAM memory subsystem, storing the pixels of reference frame buffers in tiled order will reduce reference frame read cycles by more than 10%. This saves several milliwatts, or a few percent, of system power for video decode at the expense of negligible extra processor chip logic.

8. CONCLUSION

System power consumption is an equation of many variables that must be solved for its minimum in order to create an optimal portable video device. The ideal solution can only be achieved with the flexibility of a software programmable, configurable, video optimized processor.

9. REFERENCES

[1] Wikipedia articles on “Prewitt” and “Sobel” edge detection algorithms.

[2] J. Probell, “Architecture Considerations for Multi-Format Programmable Video Processors”, *Picture Coding Symposium*, Beijing, China, April 24-26, 2006.