Energy-Efficient Bus Encoding for LCD Displays

Alberto Bocca* Sabino Salerno* Enrico Macii* Massimo Poncino†

* Politecnico di Torino
Torino, ITALY

† Università di Verona
Verona, ITALY

ABSTRACT

This paper presents a low-power bus encoding technique suitable for the digital interface to a Liquid Crystal Display (LCD). In particular, we focus on interfaces that are compliant to the Digital Visual Interface (DVI) standard, in which the three color channels are serially transmitted to achieve high bandwidth.

The proposed technique exploits the well-know inter-pixel correlation that exists in typical images by serially transmitting an encoded representation of the difference between adjacent pixels. The encoding is based on the principle of clustering the 1's in the code towards either ends of the pixel data, in such a way that serial transmission of a code yields at most 1 transition per pixel.

The application of the encoding to a series of standard images resulted in energy savings of around 60% on average, with respect to a plain transmission of 8-bit pixel data.

Categories and Subject Descriptors: B.4.2: Input/Output and Data Communications: I/O Devices

General Terms: Algorithm, Experimentation.

Keywords: LCD Displays, Digital Display Interfaces, Low-Power Bus Encoding.

1. INTRODUCTION

In spite of the high computational power of the processor cores and the huge amount of DRAM memory, the Liquid Crystal Display (LCD) system is still one of the most energy-demanding components of modern multimedia devices. Although most of the power is consumed by the LCD panel (e.g., it can easily exceed the 1 Watt mark [10]), the LCD takas a significant fraction of the total power budget (typically around 10%). This is because this bus is usually implemented through a flat cable whose unit capacitance per line is orders of magnitude larger than that of on-chip buses. This power demand cannot be easily tackled as for other resource, for instance, through some form of power management. The LCD is intrinsically a non-power-manageable resource; it must be continuously refreshed, and shutting it down cannot be done without significant penalty in performance and image quality.

In this scenario, design techniques that facilitate the reduction of the switching activity of digital LCD buses assume higher importance than in the cases, widely addressed by the literature on low-power bus encoding, of parallel, memory-to-processor buses (see [1] for a survey).

Of particular interest, in the context of LCD interfaces are encoding solutions targeting serial communication channels. In fact, most LCD interfaces use serial connection to minimize the electrical effects occurring during the transmission of data on a flat cable at frequencies in the range required by typical LCD displays (in the order of few hundred of MHz). Cheng and Pedram [2] have recently proposed an encoding schemes, named *Chromatic Encoding*, which reduces the switching activity on each color channel based on the exploitation of the concept of tonal locality, that is, on the fact that pixels that are adjacent in an image tend to have large similarity in their RGB color decomposition.

In this paper, we improve over the work of [2], by proposing an encoding solution which exploits inter-pixel correlation in a different way. In particular, we specifically target the encoding function to the serial nature of the transmission; the basic idea of the method is to minimize intra-word transitions by clustering the 1's in the encoded values towards one end of the word. The proposed method provides comparable savings (about 60% on average), yet with a much simpler encoder structure, and without any kind of approximation (such as overflow or non-representable values).

GLSVLSI'04, April 26–28, 2004, Boston, Massachusetts, USA.

2. BACKGROUND AND PREVIOUS WORK

Figure 1 shows the organization of a typical digital LCD subsystem. In particular, the graphic controller, which contains the *frame buffer* transmits to the *LCD controller*, through a digital interface, the pixel data.

Unlike its analog counterpart (for which VGA and successors are an universal standard), the digital interface the graphics controller and the LCD has long been a matter of many controversies among LCD producers and PC makers; as a matter of fact, no official standard has emerged yet, and no solution has even managed to achieve widespread acceptance.

Among the various proposals for a digital LCD interface (Plug and Display [4], Digital Flat Panel (DFP) [5], OpenLDI [6]) DVI (Digital Visual Interface) [7] has lately appeared as the solution that satisfies most of the contrasting requirements. It has been proposed by Digital Display

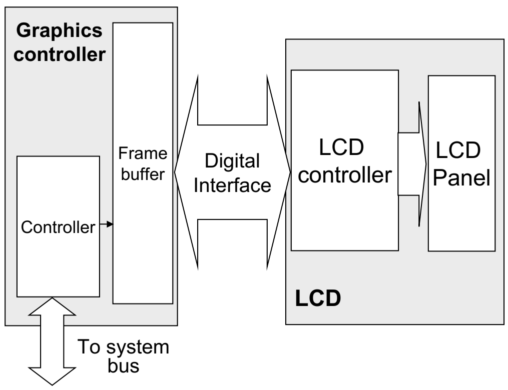

Figure 1: Organization of a Typical LCD System.

Working Group (DDWG), driven by Intel and includes companies such as HP, Fujitsu, IBM and NEC.
Like its competitors, DVI is based on a specific signaling technology called TMDS (Transition-Minimized Differential Signaling) to send data. This interface is implemented using an encoder and serializer to format the digital RGB data, which is then serially transmitted on three twisted pairs, with an additional twisted pair for the clock signal. Each 8-bit pixel data is encoded and serialized into a 10-bit value. TMDS achieves low transition counts by using *transition encoding*, where transitions are encoded rather than values. The precise encoding depends on the value of the first redundancy bit. When 1, transitions are encoded with Boolean XOR (i.e., 1 = transition, 0 = no transition); conversely, when 0, Boolean XNOR is used. The second redundancy bit is used to selectively invert (i.e., complement) the word to be transmitted so as to achieve DC balancing.
The most appealing feature of TMDS is that the clock is only used as a frequency reference, and not to latch data, which provides excellent clock skew insensitivity.
In spite of its name, TMDS has not been devised with energy consumption in mind. The minimization of transitions has been mostly sought to avoid excessive electromagnetic interference (EMI) levels on the cable.
As a matter of fact, energy efficiency techniques for LCD systems have been limited so far to the design individual component for the display systems [8, 9] or for specific control mechanism inside the LCD controller [10, 11]. The only approach that has dealt with the issue of energy efficiency during the transmission of data on the digital LCD interface is the one proposed by Cheng and Pedram in [2]. In their work, they use the same DVI-compliant interface (the 10-bit serial link) to transmit properly encoded pixel data by exploiting the correlation existing between adjacent pixels. The encoding, called *chromatic encoding*, provides significant savings, but it presents several corner cases in which pixels cannot be encoded (overflow conditions, or situations in which the encoding must switch to plain TMDS).
The encoding proposed in this work is similar in scope to that of [2], in that it assumes the availability of a DVI-compliant physical connection. With respect to that solution, however, we provide comparable energy savings, with no approximation or corner cases, and with a simpler encoder/decoder implementations.

3. LOW-POWER ENCODING

3.1 Motivation

The motivation behind the proposed encoding comes from a widely used property of digital images, that is, the *high correlation* existing between adjacent pixels. Figure 2 shows the distribution of the distance between adjacent pixels for the three (R,G,B) channels; the various curves are averaged over a sample of ten images. The distribution is roughly Gaussian, with average value approximately 0 (i.e., identical adjacent pixels) and very small variance (0.0031) for the depicted curves.

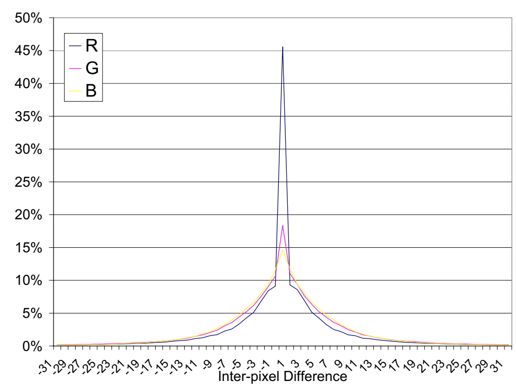

Figure 2: Distribution of Pixel Difference.

This correlation is called *tonal locality* in [2]. Such spatial correlation of images is exploited in most image compression algorithms, which rely on a *differential* representation, where differences are stored rather than full information.

3.2 Differential Bar-Encoding

Our encoding exploits inter-pixel correlation in the simplest possible way, that is, by transmitting *pixel differences* rather than their actual values. However, this differential encoding does not guarantee by itself any reduction of the number of transitions on the channels. To achieve that, a suitable encoding of the difference is required.
In our specific context, the vast literature on bus encoding for reduced switching activity is not of help, since all these methods target parallel buses. In the case of a serial bus, as for our links, a different solution must be devised. The serial requirement, in fact, rules out many of the principles used for reducing transitions on parallel buses, namely, spatial redundancy and code mappings. In our case we can only play with (i) the temporal redundancy, and (ii) the temporal sequence of the transmitted bits. The former is defined by the constraints of using TMDS links, that is, we are allowed two redundancy bits per symbol (for a total of 10 bits). The latter is thus the only degree of freedom available.
The problem we are trying to solve can be formulated as follows: *Given a n-bit word to be sent onto a serial connection, choose a proper temporal order of the bit transmission so that the resulting number of transitions is minimized.*
The above problem has a straightforward solution, that is, to reorder the bits so as to *pack all the 1's towards the beginning or the end* of the word.This order guarantees that the serial transmission of a n-bit word can be done with just 1 transition.

In our case, the "words" to be transmitted are the inter-pixel differences; since each pixel of each RGB channel is encoded as a 8-bit unsigned value, the difference may range from -255 to $+255$. In principle, encoding this range would then require one extra bit with respect to the 8-bit RGB data. In practice, however, this is not an issue, because of how the codewords are built.

In fact, following the principle of optimizing the *common case*, and exploiting the inter-pixel locality, we choose to encode only a small subset of the possible pixel difference values, and in particular the most typical ones. From Figure 2 these happen to be those with the smallest difference values. All other difference values are not encoded, and the original pixel value (rather than its difference) is transmitted.

Figure 3 summarizes the complete encoding scheme. In the pseudo-code, `map` denotes the actual mapping function which translates the pixel difference *Diff* into a transition-minimized code.

```
Encode(w_t, w_{t-1}) {
    Diff = w_t - w_{t-1};
    if (Diff is within the range)
        Code_t = map(Diff);
    else
        Code_t = w_t;
    return Code
}
```

Figure 3: Encoding Algorithm.

A precise definition of this conditional encoding requires to identify:

1. *Which and how many difference values to encode.* In principle, since there are 8 bits available, we could encode 2^8 difference values (e.g., in sign and magnitude, from -127 to $+127$). However, as discussed above, the reduction of the number of transitions requires to have the 1's clustered towards either ends of a word; this constraint on the encoding reduces the number of usable bit patterns, and automatically determines how many difference values can be encoded.

2. *The use of redundancy bits.* The conditional nature of the encoding requires one redundancy bit to signal whether the encoding is applied or not to a received pattern. However, to ensure a fair comparison with the TMDS-encoded links, which use two redundancy bits, we used another redundant bit that can be used for the encoding.

One code mapping which addresses the two above issues, and which is used in our encoding, is shown in Table 1.

We can notice the differently skewed representation of positive (from LSB to MSB) and negative (from MSB to LSB) difference values. We call this code the *Differential Bar Encoding (DBE)*, because it is reminiscent of a bar-code, in each value is identified by a number of bars.

As the table shows, the code is able to cover a limited number of values (namely, 18). The bounds of the covered range (-9 and $+8$) are somehow arbitrary; however, the semi-Gaussian distribution of inter-pixel differences suggests a placement of this range symmetrically around 0. From the plots of Figure 2, we observed that 79.9% of bit differences fall into the $-9, \ldots, +8$ range, averaged over a set of images and over the three RGB channels.

Value	*Code*	*Value*	*Code*
0	00 00000000	-1	11 00000000
+1	00 00000001	-2	11 10000000
+2	00 00000011	-3	11 11000000
+3	00 00000111	-4	11 11100000
+4	00 00001111	-5	11 11110000
+5	00 00011111	-6	11 11111000
+6	00 00111111	-7	11 11111100
+7	00 01111111	-8	11 11111110
+8	00 11111111	-9	11 11111111

Table 1: Table of Codewords.

Redundancy bits are used as follows: 00 denotes a positive difference value, whereas 11 a negative difference value. The other two patterns (01 and 10) are used to tag non-encoded words. The choice between the two patterns depends on the MSB of the pixel value: If the MSB is 1, 01 is used, otherwise 10 is used.

This code guarantees that for any value transmitted (regardless of its difference with respect to the previous one), *only 1 inter-word transition is obtained*. It is important to emphasize that the encoding does not take into account inter-pixel transitions.

3.3 An Example

Consider the sequence of pixels (*Values*) reported in the table, which also reports the corresponding binary encodings.

Values	*Binary*	*Diff*	*Diff. Bar Code*	*TMDS*
155	10011011	–	0110011011	0010101101
156	10011100	+1	0000000001	0010100100
160	10100000	+4	0000001111	0011100000
155	10011011	-5	1111110000	0010101101
223	11011111	+68	0111011111	0001100001
224	11100000	+1	0000000001	0000100000
224	11100000	0	0000000000	0000100000
78	01001110	-146	1001001110	0011010010

The application of the DBE results in the sequence shown in Column *Diff. Bar Code*; the serial transmission of these patterns results in a total of 21 transitions.

Column *TMDS* reports the resulting sequence after application of the TMDS encoding. Without loss of generality, we have assumed XOR encoding (the 9-th of the codeword is always 0), and no inversion applied (the 10-th bit of the codeword is always 0). The TMDS encoding yields 37 transitions.

4. EXPERIMENTAL RESULTS

We have applied the proposed encoding on a set of images taken from the SIPI database [12], which contains several types images with different spectral characteristics, and compared to the results obtained with the application of the TMDS encoding, as described in the DVI standard specification [7].

The results of the transition activity for the three (R, G, and B) channels are summarized in Tables 2–4. For each base color we observe that the average saving is higher then 50%, with maximum higher than 70%. Interestingly, we note that, the transition-based encoding on which TMDS relies does not decrease transition activity significantly; conversely, in many cases, transition activity is increased. This fact actually depends on the specific patterns which are serially transmitted; specifically, the transition-based encoding will be effective when pixel values exhibit many intra-word transitions (e.g., pixel values like 01010101).

Image	*Orig*	*TMDS*	*Saving %*	*DBE*	*Saving %*
couple	204306	224226	-9.75	99565	51.27
girl1	239117	236365	1.15	112405	52.99
girl2	223446	227750	-1.93	93547	58.13
house	260464	244430	6.16	85025	67.36
jellybean	207713	225621	-8.62	73114	64.8
moon	235687	218358	7.35	139855	40.66
sailboat	1045538	957862	8.39	615730	41.11
splash	923976	989212	-7.06	256960	72.19
tiffany	438092	610659	-39.39	221375	49.47
tree	263813	243885	7.55	150013	43.14
Average			-5.01		56.25

Table 2: Transition Activity for Red Channel.

Image	*Orig*	*TMDS*	*Saving %*	*DBE*	*Saving %*
couple	185778	218023	-17.36	89581	51.78
girl1	221331	228523	-3.25	95793	56.72
girl2	227495	227179	0.14	99732	56.16
house	232612	244128	-4.95	89435	61.55
jellybean	230025	250423	-8.87	79082	65.62
moon	235687	218358	7.35	139855	40.66
sailboat	906941	971888	-7.16	652583	28.05
splash	816961	905722	-10.86	313436	61.63
tiffany	852263	933770	-9.56	440692	48.29
tree	227543	240927	-5.88	136871	39.85
Average			-4.97		51.71

Table 3: Transition Activity for Green Channel.

Image	*Orig*	*TMDS*	*Saving %*	*DBE*	*Saving %*
couple	183893	220690	-20.01	99395	45.95
girl1	214200	228752	-6.79	118756	44.56
girl2	262090	234601	10.49	100177	61.78
house	244711	228672	6.55	106170	56.61
jellybean	267714	238764	10.81	69801	73.93
moon	235687	218358	7.35	139855	40.66
sailboat	951201	990100	-4.09	583902	38.61
splash	855070	924644	-8.14	290617	66.01
tiffany	1029019	982397	4.53	412539	59.91
tree	253763	241244	4.93	148284	41.57
Average			4.06		56.79

Table 4: Transition Activity for Blue Channel.

Another important measure which must be considered is the distribution of 0's and 1's in the transmitted sequence, whose number should be, in principle, as much similar as possible to guarantee perfect DC balancing. By construction, TMDS sacrifices some possible savings in transition counts by imposing a perfect 50% balancing. Our code does not explicitly consider DC balancing, yet it yields a 0/1 distribution which is not so far from 50% (36% on average). Notice, however, that, due to the limited length of the connections, DC balancing is not usually an issue. As a matter of fact, TMDS enforces DC balancing mostly for fiber-channel interconnect, rather than for standard cables. If DC balancing becomes an issue, the proposed encoding can be easily modified to tradeoff transition savings for DC balancing, by implementing the same "selective invert" scheme of TMDS.

4.1 Encoder Implementation

We have also implemented the proposed encoder for the evaluation of its power consumption. Although the latter may not be an issue, given the high capacitances of the flat cables used to connect the LCD panel to the graphics controller, we have evaluated it for the sake of completeness.
The encoder has been synthesized using Synopsys Design Compiler with a $0.13\mu m$ technology by STMicroelectronics. The power consumption is 44 μW, at a frequency of 100 MHz, using one of the images as testbench.
Notice that 165 MHz, according to the DVI standard, is the upper frequency bound for using a single TMDS channel; for higher frequencies, two TMDS channels are required.

5. CONCLUSIONS

We have presented a novel low-power encoding technique for DVI-compliant LCD interfaces. This technique can reduce transition activity significantly by exploiting the correlation existing between consecutive pixels. The proposed differential encoding can encode about 80% of the pixels with a codeword with only one transition, with no approximation whatsoever (for an average of about 60% transition saving). The hardware implementation of the encoder is simple, and it consumes only $100\mu W$ per channel, without requiring any redundant line. In addition, our encoding can be used to tradeoff energy savings with DC balancing, when this is an issue.

6. REFERENCES

[1] L. Benini, G. De Micheli, "System-Level Power Optimization: Techniques and Tools," *ACM Transactions on Design Autom of Electronic Systems*, Vol. 5, No. 2, pp. 115–192, Feb. 2000.

[2] W.-C. Cheng, M. Pedram, "Chromatic Encoding: a Low Power Encoding Technique for Digital Visual Interface," *DATE'03: Design, Automation and Test in Europe*, March 2003, pp. 694–699.

[3] K. Man, J. Norris, "Interface issues for LCD monitors," *Electronics Engineer*, December 1999.

[4] Plug and Display Standard, Version 1.0, VESA, June 11,1997.

[5] Digital Flat Panel Standard, Version 1.0, VESA, February 14, 1998.

[6] Open LVDS Display Interface (OpenLDI) Specification, Version 0.9, National Semiconductor, February 24, 1999.

[7] Digital Display Working Group, Digital Visual Interface, V1.0, `www.ddwg.org`.

[8] K. Meinstein. et al., "A Low-Power High-Voltage Column/Dot Inversion Drive System," *Society for Information Display International Symposium Digest of Technology*, Vol. 28. May 1997.

[9] B.-D. Choi, O.-K. Kwon, "Stepwise Data Driving Method and Circuits for Low-Power TFT-LCDs," *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 11, pp. 1155–1160, Nov. 2000.

[10] I. Choi, H. Shim, N. Chang, "Low-Power Color TFT LCD Display for Handheld Embedded Systems," *ISPLED'02: ACM/IEEE International Symposium on Low-Power Electronics and Design*, August 2002, pp. 112–117.

[11] F. Gatti, A. Acquaviva, L. Benini, B. Riccò, "Low Power Control Techniques For TFT LCD Displays," *CASES'02: International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, pp, 218–224, Dec. 2002.

[12] A. G. Weber, "USC-SIPI Image Database Version 5," USC-SIPI Report #315, Oct. 1997. `sipi.usc.edu/services/database/Database.html`.