BANDWIDTH REDUCTION FOR VIDEO PROCESSING IN CONSUMER SYSTEMS

Egbert Jaspers[1] and Peter de With[2]

[1]Philips Research Labs, Eindhoven, The Netherlands

[2]CMG Eindhoven / University of Technology, Eindhoven, The Netherlands

ABSTRACT

This paper presents a strategy for substantially improving the memory (SDRAM) bandwidth in MPEG-type video-processing consumer systems. A new pixel-mapping strategy is proposed that halved the memory bandwidth for MPEG decoding using a sophisticated memory device (for example DDR SDRAM).

INTRODUCTION

The architecture of present video processing units in consumer systems is usually based on various forms of hardware, including application-specific processors (ASIP) communicating via an off-chip SDRAM (see Figure 1). Examples of such consumer video systems are MPEG encoders and decoders, and high-end TVs. A problem of such systems is that the computational power of consumer systems increases faster over time than the available bandwidth to off-chip memory. Consequently, data communication to and from the memory has become the bottleneck in overall system performance. This paper presents a strategy for mapping pixels into the memory for MPEG-type applications, leading to a minimization of the transfer overhead between memory and video processing. A 25% reduction of the memory bandwidth was obtained in an MPEG decoding system. For double-data-rate SDRAM (DDR SDRAM), the proposed mapping strategy reduces the bandwidth by as much as 50%. This substantial performance improvement can readily be used for extending the quality or the functionality of the system.

PIXEL MAPPING INTO MEMORY BANKS

Consumer systems with off-chip SDRAM communicate bursts of video data for a good utilization of the memory bandwidth. Let us first briefly explain the origin of the communication problem. The above-mentioned bursts of data represent non-overlapping video blocks in the memory which can only be accessed as single entities, referred to as *data units*. Consequently, the access to a group of pixels (e.g. a macroblock – MB – in MPEG), may be distributed over several data units, resulting in transfer of all required data units. Hence, significantly more pixels are transferred than required (pixel overhead). This is particularly true if the data units are relatively large compared to the requested group of pixels.

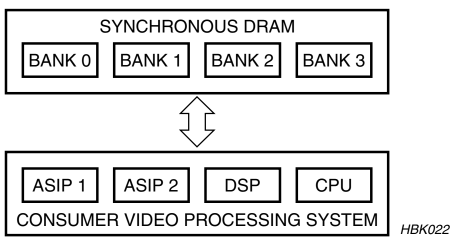

Fig.1 Consumer system architecture.

The increasing demand for memory bandwidth in multi-media consumer systems is expanding both the memory bus width and the length of the data bursts. This leads to larger data units (e.g. a 64-bit bus and a burst length of eight words, leading to 64 bytes per burst). This increasing burst size in turn leads to a much higher pixel overhead.

The optimal use of memory for MPEG applications has been studied before. In [1], a mapping of video data units into the memory relates to analyzing the application software model only, without considering data dependencies such as the set of requested data blocks and their probability of occurrence. For example, the type of data blocks that are fetched for motion compensation in an MPEG decoder strongly depends on the motion-estimation strategy applied by the encoder. In this paper, we determine an optimal mapping of video into the memory by analyzing the actual memory accesses, such that data dependencies are taken into account. Another issue that is important for bandwidth efficiency is the organization into memory banks, which is provided in all modern memory devices. We defined a template for mapping the data units

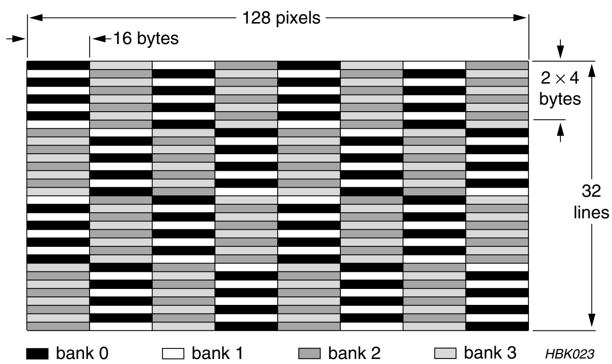

Fig.2 An example mapping template for both interlace and progressive data-block requests.

onto the different memory banks, without fixing the dimensions of the data units, thereby enabling further optimization. For example if an MB of 16×16 samples is requested, data units of 64×1 result in more overhead than data units of 8×8. Moreover, efficient communication can only be achieved when the memory banks are accessed alternately. Consequently, groups of adjacent pixels (e.g. 16×4) are mapped into single data units and successive data units are distributed in the memory banks in a fixed sequence. In addition, successive odd and even lines of an interlaced video signal are mapped in different banks of the memory to prevent additional overhead when interlaced video-data is requested by the video processing. The resulting mapping strategy, evaluated for an MPEG-2 decoder using 16×4 data units, is shown in Figure 2.

BANDWIDTH OPTIMIZATION

We have constructed a bandwidth calculation model satisfying all memory constraints. This model was applied in a simulation environment of the communication system between the memory and video-processing units. This bandwidth calculation model depends on the dimensions of the data units ($M \times N$), but also on system parameters such as the memory burst length and the width of the data bus. Moreover, the model is data dependent. These data-dependent parameters are as follows.

- *The dimensions of the requested data blocks.*
  MPEG-2 decoding contains a large variety of different data-block accesses: for interlaced and progressive video, field and frame prediction, luminance and chrominance data and for the sub-pixel accurate motion compensation.
- *The interlace factor of the requested data blocks.*
  Progressive data blocks require accesses in pairs of data units in the vertical direction. Consequently, the smallest data entity to access is two data units. Hence, the calculations are different.
- *The probability of the data-block occurrence.*
  For example, if only 16×1 data blocks are accessed (100% probability), the optimal data-unit dimension will also be very much horizontally oriented. Obviously, the probability of each data-block type depends very much on the application.
- *The probability distribution of data-block positions.*
  If a requested data block is aligned with the boundaries of the data units, it can be accessed with relatively low pixel overhead. Data blocks that overlay many data units cause much pixel overhead. Note that the 16×16 macroblock grid for MPEG and the high probability of the zero-motion vectors have a positive effect on the pixel overhead.

Our experiments used an MPEG-2 decoder implementation to statistically determine the four above-mentioned parameters. Subsequently, the pixel overhead was minimized as a function of the data-unit dimensions ($N \times M$), thereby achieving the minimum bandwidth requirements.

RESULTS AND CONCLUSIONS

Several bit streams, originating from different encoders have been decoded with our system to optimize the memory efficiency. Table 1 shows the final bandwidth results for 32 and 64-byte data units and includes the writing of the motion-compensated MBs and the reading of block-based video data ($N \times M$) for display. From the table it can be concluded that for 32-byte data units the mapping of 8×4 results in the minimum overhead. For larger data units (64 bytes), representing more sophisticated memory devices such as double-data-rate SDRAM, it was found that 16×4 is optimal.

Table 1 Bandwidth results for 32 and 64-byte data units.

	data unit dimensions	requested[1] data [%]	transferred[1] data [%]
32-byte data	(32 × 1)	100	100 + 101
	(16 × 2)	100	100 + 55
	(8 × 4)	100	**100 + 49**
	(4 × 8)	100	100 + 65
64-byte data	(64 × 1)	100	100 + 241
	(32 × 2)	100	100 + 130
	(16 × 4)	100	**100 + 72**
	(8 × 8)	100	100 + 81

1. 100% equals 22 Mbyte/s for 50 Hz CCIR 601 video.

For conventional systems (e.g. [2]) where no block-based mapping strategy is used, the bandwidth requirements are equal to the results of the 32×1 mapping, as shown in the first row of the table. Hence, the overhead is halved and an overall bandwidth improvement of 25% is realized. For systems with 64-byte data units, the amount of bandwidth is even reduced by 50%. Baskett and Hennessy [3] indicate that the gap between processing power and available memory bandwidth is increasing rapidly, suggesting the main bottleneck in overall system performance for years to come. The improvement of our proposal is so substantial that it can easily be exploited to narrow this gap or extend the quality or functionality of the MPEG system. Although we performed our optimization for an MPEG-2 decoder only, the approach can also be successfully applied in other consumer video processing systems based on a shared memory concept.

REFERENCES

[1] H. Kim and I.C. Park, "Array address translation for SDRAM-based video processing applications", in *Proc. of SPIE: Vis. Comm. and Image Proc.*, June 2000, vol. 4067, pp. 922-931.

[2] S. Rathnam an G. Slavenburg, "Processing the new world of interactive media", *IEEE Signal Processing Magazine*, vol. 15, no. 2, pp. 108-117, March 1998.

[3] F. Baskett and J.L. Hennessy, "Microprocessors: From desktops to super computers", *Science*, vol. 261, pp. 864-871, Aug. 1993.