TP 14.8 Heterogeneous Multi-processor for the Management of Real-time Video & Graphics Streams

Marino Strik, Adwin Timmer, Jef van Meerbergen, Erwin Waterlander,
Françoise Harmsze, Ad Vaassen, Leo Sevat, Marcel Oosterhuis,
Gert Jan van Rootselaar , Harry van Herten, Egbert Jaspers, Johan Janssen,
Gerben Essink, Jeroen Leijten

Philips Research Laboratories, Eindhoven, The Netherlands
Eindhoven University of Technology, Eindhoven, The Netherlands

A chip for the *concurrent* processing of many *real time* multi-media streams has 3 *independent and un-correlated* video input channels (Figure 14.8.1). They are stored in and retrieved from an external SDRAM to be displayed on a TV set using PC-like multiple windows. Graphics data generated by an external CPU is read from SDRAM to be blended with the composition of the video.

Functions like internet access, electronic program guide, E-commerce are added to the classical TV functions while maintaining a high display quality and a simple user interface. The chip processes up to 34 video streams of 16MHz pixel rate.

After a analysis of the applications a common set of high-level functions is defined: RGB graphics blending, horizontal sample rate conversion, vertical sample rate conversion, noise reduction, 2D sharpness enhancement and video juggling. Different applications can be specified as different flowgraphs in which the nodes are the above functions. Figure 14.8.4 shows 2 different application examples to illustrate the diversity. The flowgraphs show 'picture in picture' replay and a 'magnify glass' (Figure 14.8.5) in which the accesses to SDRAM are represented explicitly.

Classical programmable solutions fail because of performance and a too low intrinsic computational efficiency (ICE) [1]. Assuming a stream with 16MHz pixel rate and 50OPS/pixel, 800MOPS/stream is needed. This would require about one state of the art processor to process one stream. This is not cost effective for consumer applications for two reasons. First, the ICE can be improved by two orders of magnitude by application specific implementation for the different functions. Second, programmable solutions use coarse level synchronization, typically fields or frames, which leads to an explosion of the memory cost.

Therefore a different approach is taken, which sticks to the dataflow paradigm for the implementation. The application domain analysis really calls for a *reconfigurable* solution in which the kernels are the functions defined above. They are implemented in different processors each tuned to a particular function. Changing to a new application is done by reconfiguring the on-chip communication architecture and via new parameters for the weakly programmable processors. This must be done without artifacts on the display. The timing of the switching (the decision when to switch) can only be at runtime since different streams are uncorrelated, i.e. the field transitions are not synchronized. These problems are solved by storing and using two configurations simultaneously when in a transition state and by controlling the gradual switching at runtime. [2] (Figure 14.8.3)

This results in a heterogeneous multi-processor architecture are shown in Figure 14.8.2. The chip contains the following autonomous processors for high quality image processing: input(3x), output YUV, output with graphics blending RGB, horizontal sample rate converter (3x), vertical sample rate converter (2x), spatial & temporal noise reduction (2x), 2D sharpness enhancement, and video composition juggler (2x).

The architecture is completed with blocks necessary for the communication infrastructure: a configuration manager, a DMA stream processor (12x from mem, 8x to mem), global controller, CPU and memory interface. The number between brackets equals the number of different video streams which can be handled in parallel.

The interface between processors and the communication network is implemented with FIFOs. These fifos are present at the input and output of every processor and de-couple the processing from the communication. The processing of data is data driven. The processor executes when data is available in the input fifos and when storage locations are available in the output FIFOs. Otherwise the local processor clock will be gated and all processing activity will be halted for the related task without having consequences for the other task on the multitasking processors. Context switches can be made in a single cycle.

The most critical resource on the chip is the bandwidth to external memory. The video processing requires a high but predictable and periodic bandwidth, the video processing requires a lower bandwidth however with a short latency. The management scheme used provides priority for the CPU as long as the data processing meets its deadline. This is evaluated during a programmable number of clock cycles [3]. This scheme requires buffering on the video processing side but in return it maximizes CPU performance.

The application-specific approach means that backend design is totally different from general-purpose CPU design which is characterized by custom layout and careful optimizations to reach the high clock frequencies. Here, the interest is in reuse of the IP blocks in other applications or in future processes. Therefore reliance is on commercial RT level synthesis and on in-house high level synthesis tools (Phideo) and on a place and route at the top level which is easily reproducible and robust. Therefore the clocking strategy uses multiple clock domains where PLLs synchronize to a 16MHz reference clock. One PLL is used for every block in the layout hierarchy. This is done to make block design of all 9 layout blocks independent with respect to clock tree insertion delay. The oscillator in the PLL produces an 192MHz signal which is divided to 16/32/48/64/96MHz. The memory interface uses the highest clock frequency. All inter-block communication is performed at 64 MHz. A small delay line is used to tune the clock tree insertion delay for different clocks inside a layout block. A mismatch of up to 750ps can be trimmed to less than 20ps. Clock skew analysis is with static timing analysis after final routing.

References:

[1] Claasen, T., 'High Speed: Not the Only Way to Exploit the Intrinsic Computational Power of Silicon', ISSCC Digest of Technical Papers, pp 22-25, Feb. 1999.

[2] Leijten, J. A. J. et al., 'Prophid: A Data-Driven Multi-Processor Architecture for High-Performance DSP', Proceedings of the 1997 European Design & Test Conference, Paris, France, March 1997.

[3] Hosseini-Khayat S., A.D. Bovopoulos, 'A Simple and Efficient Bus Management Scheme that Supports Continuous Streams', ACM Transactions on Computer Systems, vol 13, no. 2, pp 122-140, May 1995.

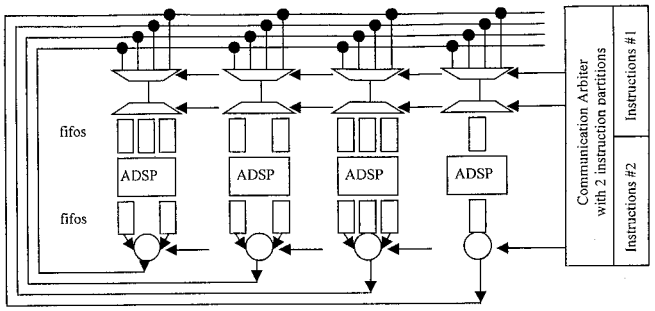

Figure 14.8.4: Video stream high-performance communication structure.

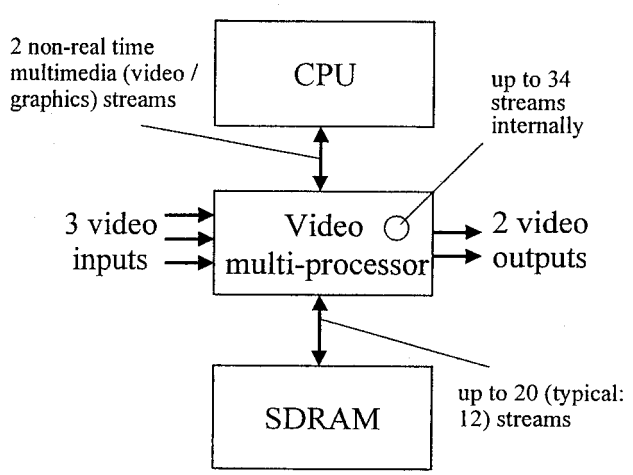

Figure 14.8.1: System concept.

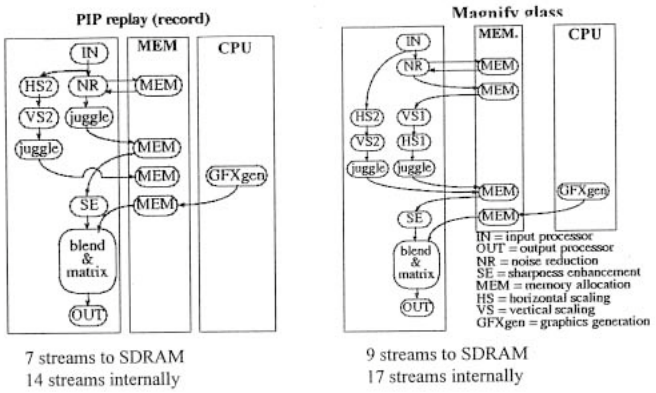

Figure 14.8.2: Two example applications.

Figure 14.8.3: Image result from magnify example.

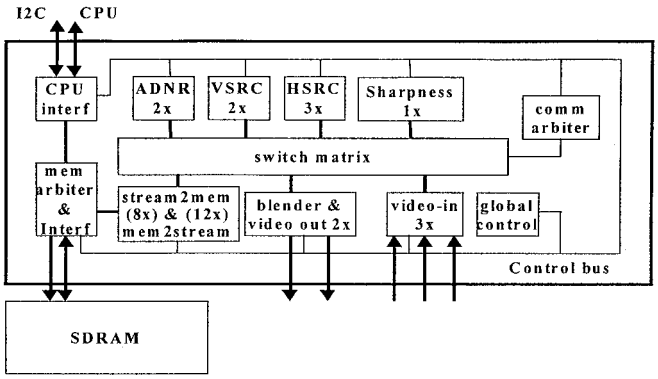

Figure 14.8.5: Heterogeneous multiprocessor.

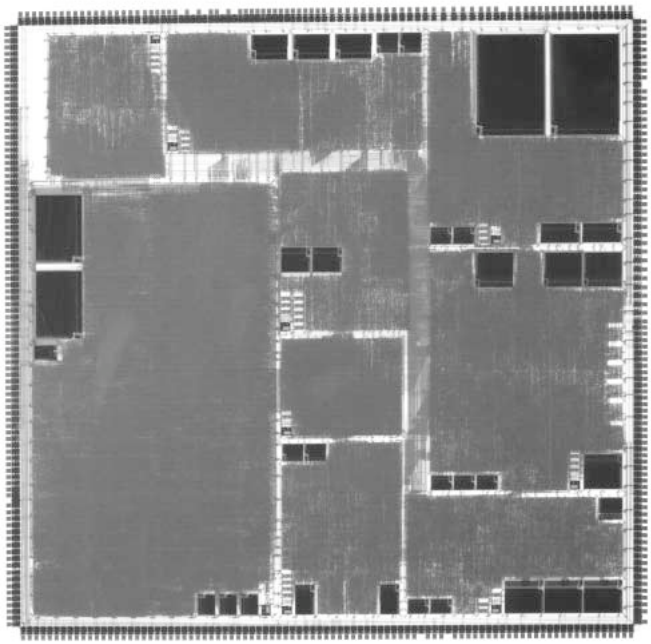

Figure 14.8.6: Chip micrograph.

Technology	0.35µm CMOS
Clock frequency	Noise reduction = 16MHz
	Sharpness enhancement =32MHz
	Horizontal zoom=64MHz
	Vertical zoom=64MHz
	Memory interface=96MHz
	CPU interface=48MHz
Power supply	3.3V
Transistor count	7.6M
Package	352 SBGA
Processing power	>10GOPS
Testability	Full scan
Debug	Full internal state access via JTAG Hardware breakpoints
Die size	13mm x 13mm

Figure 14.8.7: Device specifications.