

*Egbert G.T. Jaspers*¹, *Peter H.N. de With*²

¹ Philips Research Labs., Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands.

² University of Mannheim, Fac. Computer Engineering, 68131 Mannheim, Germany.

Abstract— In this paper, we present a chip-set for digital video processing in a consumer television receiver or set-top box. A key aspect of the chip-set is a high flexibility and programmability of multi-window features with multiple Teletext (TXT) pages, Internet pages and up to three live video windows. The chip-set consists of a microcontroller featuring pixel-based graphics (GFX), and a video processor containing a number of flexible coprocessors for horizontal and vertical scaling, sharpness enhancement, adaptive temporal noise reduction, blending of graphics, mixing of multiple video streams, and 100 Hz up-conversion.

Keywords— Video processing architecture, multi-window TV, HW/SW co-design, programmable hardware, dynamic data-flow.

I. INTRODUCTION

Developments in high-quality television (TV) applications clearly need to envision the consumption of multiple simultaneous information channels, as opposed to the watching of a conventional single-channel broadcast. The increased demand for parallel processing and the large range of display formats is additionally fuelled by the strongly growing use of information systems like the Internet access, Teletext (TXT), and electronic help manuals. Similar to TXT on a conventional TV set and help wizards on a computer screen, the user wants to watch such information simultaneously in addition to real-time video channels. The presented ICs support these requirements with flexible control of the picture quality. Furthermore, the ICs allow the applicant to extend the functionality in hardware, by using additional coprocessors, and/or in software by installing new functions.

Another trend, which is relevant for TV processing architectures, is the gradual shift from dedicated hardware applications towards software-based (or controlled) functions. This is amongst others enabled by the increased computing power of programmable general-purpose hardware. It can also be noticed that new digital video communication standards, such as MPEG-based coding, exploit software control for improving the flexibility of hardware-based processing.

A third aspect is the continuous growth of applications in the TV domain. Upcoming features are more channels, introduction of digital TV broadcasting and reception, electronic program guides, pixel-based graphics (GFX), signal processing for improved quality, and so on. A new chip-set should accommodate to these new application domains. With such a high number of functions, it becomes attractive to pursue re-use of hardware functions and the corresponding memory, where it is appropriate to reduce system costs.

The aforementioned discussion can be translated into a list of system requirements, which are summarized below.

- reprogrammability of existing hardware for different functionality;
- sharing of (overall) memory capacity ;
- increased use of SW-based control or implementations;
- open for accommodating new functions;
- low system costs for TV market.

This paper is organized as follows. Section II briefly outlines the architecture of the new chip-set. In the

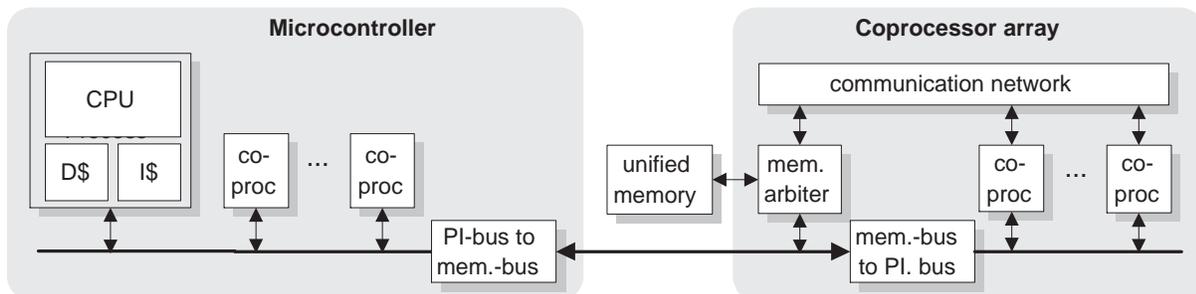


Fig. 1. The two-chip architecture of the proposed digital video platform.

TABLE I
CHARACTERISTICS OF THE CHIP SET.

	Units	Microcontroller	Coprocessor Array
Process	μm	CMOS 0.35	CMOS 0.35
Die size	mm^2	80	150
Data clock	MHz	48	64 (96 to memory)
Package		SBGA352	SBGA352
Dissipation	W	1.5	5 (estimation)
Transistor count	$\times 10^6$	2	6 (estimation)
Interfaces		JTAG, 2 \times UART, 2 \times I ² C, SDRAM Serial Interconnect Bus (SIB) remote control software AD converter general I/O Graphics OUT: RGB α /FB, H+V sync	JTAG, 2 \times I ² C, SDRAM 3 \times Video IN: YUV 4:2:2 (up to 60 MHz) 2 \times Video OUT: YUV 4:2:2/4:4:4 or RGB

following section an overview of possible communication and integrated video functions enabled by the chip-set are presented. Section IV deals with an example application discussing the large flexibility of the chip-set and the variety of functions that can be implemented. Section V gives an overview of the application domains (communication, video, etc.) and the individual functional possibilities in those domains. In Section VI the conclusions are presented.

II. ARCHITECTURE

In [3], the necessity of a modular parallel processing architecture for TV applications is motivated and a preliminary application is discussed. In-depth details of the architecture are presented in a corresponding paper [8]. In this paper, we will focus on the functional possibilities.

Figure 1 shows the hardware architecture of the programmable video processor chip-set. The system consists of two chips: a microcontroller which has been presented in [7] and a coprocessor array. The former is for teletext (TXT), graphics generation, system and chip control and various communication features, whereas the latter executes the necessary video tasks in software and in weakly programmable hardware. Both ICs are autonomous units which communicate with each other via a central bus. Internally, this bus operates according to the Peripheral Interconnect (PI) bus protocol [6], whereas the external part of the bus uses a synchronous DRAM (SDRAM) memory protocol. This enables stand-alone operation of both ICs, while for the case that both ICs are interconnected, they make use of one single unified memory.

The novelty of the system for TV applications is the concept to enable a large range of features with high flexibility. The signal flow-graph, i.e. the order of signal processing functions through all coprocessors, is programmable by means of a flexible communication network [5]. The processing of data by several coprocessors is achieved without the need to ac-

cess (bandwidth-limited) memory for communication purposes. The coprocessors, which are interconnected via a communication network, are synchronized upon data-packet reception and scheduled by means of data-driven processing of the pixels [4]. This implies that each task in a data flow-graph (resembling the order of functions) is executed when data is present at the input, while the subsequent receiving coprocessor(s) can absorb the output data. This self-scheduling mechanism provides autonomous processing without interference of a microcontroller or a hardware scheduler.

Let us now discuss how the aforementioned architectural properties are exploited for flexibility and new features. Two examples are illustrated briefly. An overview is provided in the next section.

- The two chips are sharing the background memory. This enables the possibility to assign the available memory space to applications in a dynamical way. For example, if extra GFX data are generated for user control interface, the quality of one of the video applications, such as noise reduction, can be scaled down temporarily to free resources.
- The order in which video functions are carried out (the flow-graph) is programmable. Furthermore, most coprocessors can be accessed more than one time. This means that if functions are addressed in a different way, the quality of the resulting image can be optimized differently, depending on the application. For example, for two video windows, the noise reduction can be carried out for the window where it is needed most.

III. OVERVIEW OF HARDWARE FUNCTIONS

In this section we briefly describe the hardware blocks integrated in the microcontroller and the video processor chip.

The microcontroller contains a 32-bit R3000 reduced instruction set computer (RISC) core for control of the video coprocessor array, and TV-set control. Moreover, blocks are integrated for graphics generation, teletext

TABLE II
HARDWARE FUNCTIONS IN THE MICROCONTROLLER CHIP.

Category	Hardware support
Software execution	R3000 RISC with I- and D-cache, interrupt control, timers, watchdog timer
Teletext	Input for TXT front-end IC SAA5284
Control communication	2 × high-speed UART (230 kbit/s) 2 × I ² C, IR connection
Data communication	Serial Interconnect Bus for UCB100H modem IC
Memory connection	SDRAM controller for 16 Mbit and 64 Mbit external chips, general-purpose ROM interface
Testing, miscellaneous	EJTAG interface, general I/O pins, SADC pins (keypad)

slicing and decoding, and modem functionality for Internet connection. Some of the coprocessors in the microcontroller, as depicted in Figure 1, support the RISC core directly, such as an interrupt controller, and a number of standard communication modules (UARTS, Infrared support for remote control, JTAG for testing, etc.). The controller chip also contains an I²C block for generic control of neighbouring chips. Important is the graphics output processor which enables pixel-based graphics with a resolution above the conventional TXT images. Finally, an SDRAM memory controller supports the connection of a large external SDRAM memory for executing all software tasks and exchanging data with the coprocessor array. Table II gives an overview of the various functions.

The video coprocessor array performs all video signal-processing functions of the chip set. It contains a set of coprocessors for TV functions which are typical for a high-end TV set or set-top box. For video image resizing, it contains a horizontal scaler and a vertical scaler with field rate up-conversion capability. The latter reduces the line flicker when using interlaced video signals. The input signal quality may be improved using the integrated adaptive temporal noise reduction, which analyzes both the video signal noise and the motion. The sharpness may be augmented with the contrast-adaptive local sharpness enhancement. There is also support for mixing several video streams. Firstly, a graphics blender features "alpha" blending of graphics and full-motion video. Secondly, a mixer can combine up to three moving video streams. Finally, a color-space converter converts YUV to RGB signals. Most coprocessors are implemented only once, but have capabilities for parallel processing of multiple tasks. In this flexible approach, each task is independently programmable, e.g. the scalers can compress or expand an image to any arbitrary size with arbitrary filter coefficients.

TABLE III
HARDWARE FUNCTIONS IN THE COPROCESSOR ARRAY.

Category	Hardware support
Signal quality	temporal noise reduction, locally adaptive Y peaking, Transient Improvement for UV signals
Scaling	horizontal scaling (up to 3 streams) vertical scaling (two streams)
Graphics, mixing	alpha blending, resolution > TXT, up-conversion of TXT signals, mixing up to 3 signals
Video Interfaces	2 YUV/RGB inputs, CCIR-656 support, RGB out
Memory connection	SDRAM controller for 16 Mbit to 64 Mbit/s external chips
Testing, miscellaneous	EJTAG interface, I ² C for standalone operation

IV. APPLICATIONS

Because functionalities of TV and PC are merging increasingly, it is likely that the user-interfaces of these devices are going to show similarities. For the TV, this implies a multi-window environment with multi-tasking capabilities. The flexibility of the described system enables this feature and provides a broad range of applications which are new in a consumer TV environment. From the previous sections, it can be derived that programming of an application is straightforward and is limited to programming of the signal flow-graph and the individual setting of the tasks. In the following subsection, the functionality and the settings of the individual coprocessors are described. This together with some application examples, gives an impression of the application possibilities.

A. Functionality of the coprocessors

The coprocessor array (CPA), which performs the computationally expensive regular video processing, contains coprocessors with carefully selected functionalities. Some coprocessors contain more than one type of functionality, because flexibility in the configuration of the signal flow at this point would not make sense. In the following, the functionality of the individual coprocessors is described.

Vertical sampling rate converter (VS) for expansion and compression of images in vertical direction to any arbitrary size. It is based on a 6-tap 32-phase polyphase filter and has a median filter to perform de-interlacing (optional). The number of applied filter coefficients is programmable as well as the magnitude of the coefficients. The de-interlacing is optional. Also programmable are the scaling factor (-4 .. 64) for compression and expansion and the resolution of the input image. A memory cache for eight full-colour video lines is available and may be divided over the several vertical

scaler-tasks which can be executed simultaneously.

Horizontal sampling rate converter (HS) for expansion and compression of images in horizontal direction to any arbitrary size. This scaler is based on a 6-tap 64-phase polyphase filter and can be switched in a transposed mode. In this mode, only one single Nyquist filter has to be selected to compress the video pictures to any arbitrary size with high picture quality. The scaling factor for horizontal expansion is variable and according to a programmable parabolic curve as a function of the pixel position. This enables the commercially available "super zoom" option for the aspect-ratio adaptation. A more detailed description of the used sampling-rate conversion technique can be found in [1].

Advanced dynamic noise reduction (NR) to do temporal noise reduction, adaptive to the amount of noise and the amount of motion in the picture. A special low-pass filter ensures that noise reduction is performed in the frequency area where the human visual system is most sensitive to the noise, while high-frequency detail is preserved. The strength of the noise reduction and the motion adaptivity is programmable by means of a programmable look-up table.

Adaptive sharpness enhancement (SE) for subjective sharpness improvement of the luminance as well as the chrominance. For the luminance signal a 2D high-pass filter is used to create an enhancement signal which is added to the original. A large extent of programmable control logic provides suppression of the enhancement on those places where edges are absent and noise is visible. Suppression also takes place on image locations where enhancement would introduce aliasing artifacts and edges are already sharp, e.g. synthetically generated graphics in the video. More detailed information about the adaptive peaking function can be found in [2]. For the chrominance signal, a non-linear algorithm is implemented that increases the steepness of the edges by means of pixel translations.

Graphics (GFX) with video blending for graphics formats up to 16-bit resolution. The formats are converted to a 24-bit RGB format and the video is converted via a programmable colour-space matrix from a YUV 4:2:2 or 4:4:4 format to a RGB or YUV 4:4:4 format. For higher video resolutions, it is possible to up-convert the GFX with pixel repetition with a factor 1–8 to limit the CPU load for the GFX generation. This preserves the unlimited graphics bandwidth and gives subjectively the best algorithm. The fully digital blending of the GFX into the video is performed with an alpha factor (α) which describes the fraction of the

GFX in the output signal. For mapping GFX pixels into the signal components ($RGB\alpha$), a color look-up table (CLUT) is included. This allows the following GFX formats: CLUT8 (8 bits/pixel), $RGB\alpha$ 4:4:4 (α through LUT), $RGB\alpha$ 5:5:5:1 (α through LUT) and $RGB\alpha$ 5:6:5:0. Furthermore, the blender enables color keying for the video and/or the graphics.

Input processor for retrieval of three real-time video sources at resolutions up to HDTV and VGA. The processor supports several signal protocols for synchronization, e.g. with H- and V-pulse, with active video identifiers or according to the CCIR-656 recommendation. The digital input signal may have an 8-bit or 16-bit bus depending on a programmable time-multiplex mode. In addition, the input processor may select a capture window in the video to select the pixels to be processed by the rest of the system. This could be useful to reduce the bandwidth when only a part of the video is necessary, e.g. in the case that a part is zoomed in by the scalers.

Output processor to output one or two video signals with similar properties as the input processor. Additionally, the output processor also contains a multiplex mode to display a 24-bit YUV or RGB signal.

A memory input port and output port to store, buffer or shuffle video data at multiple positions in the video-signal flow-graph. It can access a programmable cyclic memory block in a sequential order and has the additional option to skip video lines in order to access a progressive frame in an interlaced manner.

An advanced address generator (juggler) to write video data into the memory at any position with an arbitrary shape, e.g. to create a circular Picture-in-Picture (PiP) (or alternative shapes).

B. Resource requirements

To create an application with several tasks, the functions as mentioned above, have to be executed in parallel. Because some tasks in the application may even have the same functionality, each function should be able to execute more tasks simultaneously. As a consequence, the data rate in the coprocessors and the data bandwidth to the memory increases for more complex applications and is limited by the physical clock rate of the coprocessors (64 MHz) and the memory. It can be concluded that the limitations of an application are determined by the task resources of the individual coprocessors and the memory capacity and bandwidth. The task resources of the available coprocessors are shown in Table IV (gross signal rates). The current top high-end TV-sets show features like Picture-in-

TABLE IV
TASK RESOURCES OF THE COPROCESSORS.

Horizontal scaler	$1 \times < 64\text{Mpixels/s}$ or $2 \times < 32\text{Mpixels/s}$ or $2 \times < 16 + 1 \times < 32\text{Mpixels/s}$ or $1 \times < 16 + 1 \times < 48\text{Mpixels/s}$
Vertical scaler	$2 \times < 32\text{Mpixels/s}$ or $1 \times < 16 + 1 \times < 48\text{Mpixels/s}$
Sharpness enhancement	$1 \times < 32\text{Mpixels/s}$
Noise reduction	$1 \times < 16\text{Mpixels/s HQ}$ or $2 \times < 16\text{Mpixels/s MQ}$
Color conversion, GFX expansion, GFX blending	$1 \times < 64\text{Mpixels/s}$
Video Inputs	$2 \times < 64\text{Mpixels/s}$
Video Outputs	$2 \times < 64\text{Mpixels/s}$
Memory Inputs	8 inputs + 12 outputs
Memory Outputs	total $< 192\text{Mpixels/s}$

Picture (PiP), dual-screen and Internet TV. However, these features are just a subset of the application space which is available with the introduced chip-set. Let us start with a straightforward example and consider a PiP application. Figure 2 shows the signal flow-graph of this application. For wide-screen standard-definition (SD) signals in the PiP application, the task resources as mentioned above are more than sufficient. To calculate the memory bandwidth, it is assumed that the data rate is $f_i = 16$ MHz. The system memory device runs at a clock rate of 96 MHz and has a bus width of 32 bits. For 16-bit pixels, this means a total memory bandwidth of 384 MByte/s. Since all communication between the CPU and the coprocessors is performed via memory, a part of the memory bandwidth is reserved and thus cannot be used for video processing. Assuming 30 MByte/s of memory bandwidth for control and communication between the CPU and the coprocessors, a bandwidth of 354 MByte/s remains for video processing. For the simple PiP application, only memory access for the mixing is necessary, thus the amount of available memory bandwidth is only used for a small part. This mixing or juggling of the video streams is

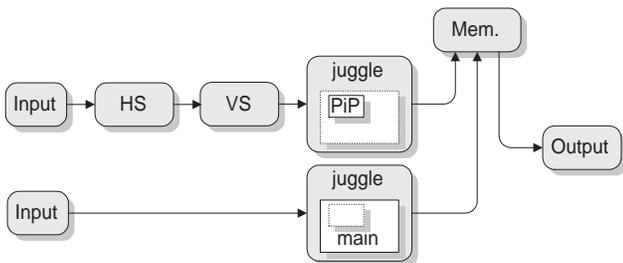


Fig. 2. The signal flow-graph of a PiP application.

designed such that it requires a minimum amount of memory bandwidth. In the background memory, two field blocks (for interlaced video) are allocated to con-

struct the mixed images. These memory blocks are filled with respectively the odd and even fields of the picture, except for the pixel positions where the PiP window is located. This unfilled area in the memory is used by the second video path to write the PiP window. Therefore, the total amount of data stored is equal to the data of one complete picture and similarly, the total required bandwidth equals the bandwidth for writing one complete video stream. With 2 Byte/pixel the amount of memory becomes 0.98 MByte and the bandwidth 64 MByte/s (reading and writing).

Since the two input video sources are generally not synchronous, the output image should be synchronized to one of the input video sources by means of a parameter setting. For the PiP application, the output video signal is usually synchronized with the input video signal of the main background picture. As a result, the input video signal of the PiP is written asynchronously into the field memories.

C. Application example I

A more advanced example of an application is a PiP with a video background that contains a zoomed-in part of the PiP. The part of the PiP to be zoomed in can be selected by the consumer by means of a small graphics square that can be moved and resized. In addition, an Internet browser, executed on the CPU, is shown on the display (see Figure 3). For generation



Fig. 3. A multi-window application with video and Internet.

of the graphics in the picture, no task resources others than the CPU and a memory output port are necessary. Therefore, the following calculations consider the video signal processing only and assume that the graphics are available in the background memory.

The signal flow-graph of the application is shown in Figure 4 and contains noise reduction (NR), scaling (HS, VS), mixing (juggle), sharpness enhancement (SE) and graphics blending prior to picture display. Table V presents an overview of all memory accesses, including the required memory capacity and the number

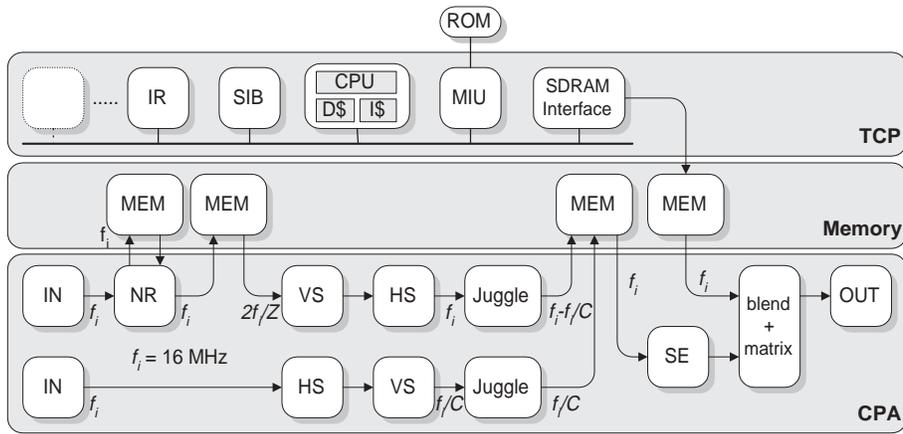


Fig. 4. Signal flow-graph of the multi-window application with Internet.

of inputs and outputs to and from memory. First, the

TABLE V

MEMORY REQUIREMENTS FOR THE MULTI-WINDOW APPLICATION.

	Connect. to/from Mem	Memory (MByte)	Memory bandwidth (MByte/s)
NR (expansion)	1/1	0.49	64
VS (expansion)	1/1	<0.98	<96
Juggling (write/ read, worst case)	2/1	0.98	64
graphics	0/1	0.49	32
Total (worst case)	4/4	2.94	256(218)

noise-reduction (NR) coprocessor accesses the memory to use a field delay for advanced adaptive temporal Infinite-Impulse-Response (IIR) filtering. For an SD image of 288 lines \times 854 pixels with 2 Byte/pixel, the required amount of memory equals 0.49 MByte. For a pixel rate of 16 Mpixels/s, the total memory bandwidth for writing and reading is 64 MByte/s. The memory requirements for the access prior to vertical scaling are different. The image is written with 16 Mpixels/s, but is read at $2 \times 16/Z$ Mpixels/s for interfield processing, with Z being the expansion factor. Because $Z > 1$, the required memory bandwidth is smaller than 96 MByte/s. If intrafield processing is used for vertical scaling, the data rate is even less than $16/Z$ Mpixels/s. The computation for the amount of buffering is less straightforward. If interfield processing is used, a complete field of L_f lines has to be written in the memory and cannot be overwritten, because it has to be read out two times. Therefore, the required amount of memory in the progressive video part that is expanded equals:

$$Buf_{inter} = 2 \times \frac{L_f}{Z} B_l,$$

where B_l denotes the number of Bytes per video line. For intrafield scaling, buffering is only necessary to

compensate for the rate difference between the writing to and reading from the memory. In this case, the time for writing one field is equal to the time for reading one field. Writing of the video lines to be expanded is done at a higher rate than reading. The maximum distance in the memory between the read and write pointer is equal to the memory space that has to be buffered for the rate difference. This maximum distance is reached when the write pointer has just finished the field. The part of the field that has been read at that time is $1/Z$. Therefore, the part that is not read yet equals $1 - 1/Z$. As a result, the buffering that is required to deal with the rate difference equals

$$Buf_{intra} = \frac{L_f}{Z} \left(1 - \frac{1}{Z}\right) B_l.$$

Since it is desired to have a static memory allocation, the maximum buffering can be found as follows:

$$\frac{d}{dZ} (Buf_{intra}) = -\frac{L_f}{Z^2} \left(1 - \frac{2}{Z}\right) B_l = 0 \Rightarrow Z = 2,$$

$$Buf_{intra} = \frac{L_f}{Z} \left(1 - \frac{1}{2}\right) B_l = \frac{L_f}{4} B_l.$$

For $L_f = 288$ and $B_l = 1708$ the amount of required buffering is $Buf_{intra} = 0.12$ MByte.

Finally, the mixing or juggling of the video streams for image composition is performed. As explained in the previous subsection, the amount of data stored is equal to one frame and the required bandwidth equals to the bandwidth for writing one complete video stream.

For generation of the graphics in the background memory, a field or frame memory could be used depending on the desired quality. When a field memory is used and the content is read for both odd and even fields, the amount of memory is reduced at the cost of some vertical resolution. Since synthetically generated graphics may contain high spatial frequencies,

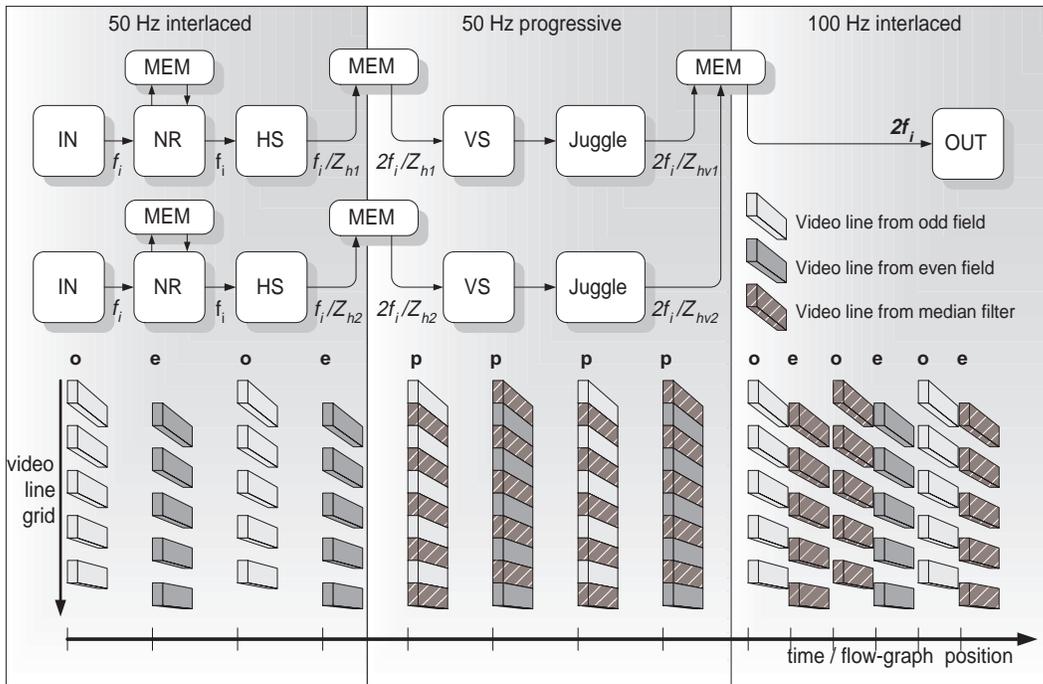


Fig. 5. An example application with 100-Hz conversion.

the use of a frame memory may result in annoying line flicker, when the content of the memory is displayed in interlaced mode. Therefore, a field memory is most attractive for 50-60 Hz interlaced video, whereas for field rates higher than 70 Hz a frame memory could be used for high-resolution GFX.

To Summarize, the total amount of applied video memory is less than 3 MByte and the maximum memory bandwidth is 256 MByte/s. This required bandwidth is only used during the transfer of the active pixels. At the blanking times, no data is transferred, thereby decreasing the bandwidth significantly. In order to decrease the required peak bandwidth, the data transfer rate can be equalized over time. To do this, the read and write tasks of the CPA have the ability to spread the transfer of an active video line over the time of a complete video line including the horizontal blanking. Typical video signals contain 15 % horizontal line blanking time, so that the total amount of bandwidth can be reduced by 15 %. This leads to a net total memory bandwidth of 218 MByte/s.

D. Application example II

Because the system is mainly limited by the throughput bandwidth of the coprocessors and the memory, a large range of resolutions and frame rates can be generated. It may even provide 50-100 Hz conversion making advanced use of the memory. Figure 5 shows a simple dual-screen application which also provides 50-100 Hz conversion. The total signal flow-graph can be divided into several independent subgraphs, separated by mem-

ory accesses. Because temporal scaling requires the use of field and/or frame memories, this can only be provided by the intermediate memory accesses. Therefore, it is not possible to perform temporal scaling within a subgraph. Only spatial scaling with some local buffering can be applied. The subgraphs that contain the input processors (IN) should operate at the field rate of the (50-Hz) input signals, due to the data-driven concept. The bottom part of Figure 5 illustrates the position of the vertical video lines as a function of time. After temporal noise reduction (NR) and horizontal scaling (HS), the interlaced fields are written into the memory. In the succeeding subgraphs, the video data are read from memory again and scaled down in the vertical dimension to obtain the correct aspect ratio of the input images. As was mentioned in the previous subsection, the vertical scaler may read the data from the memory in a progressive way to enable high-quality scaling. The vertical scaling is then applied to progressive video and is interlaced again at the output stage of the scaler. For this mode of operation, this interlacing at the output is not used and the video is written into the memory again in a progressive-scan format. In the bottom part of Figure 5, it is shown that all missing lines of the interlaced fields are filled with video lines from the median filter. If further vertical processing of progressive video would be desirable (e.g. 2D sharpness enhancement), it would be obvious to perform it in this subgraph. The last part of the video signal flow-graph contains the subgraph that reads the 50-Hz progressive frames in memory with an interlaced scan to do the

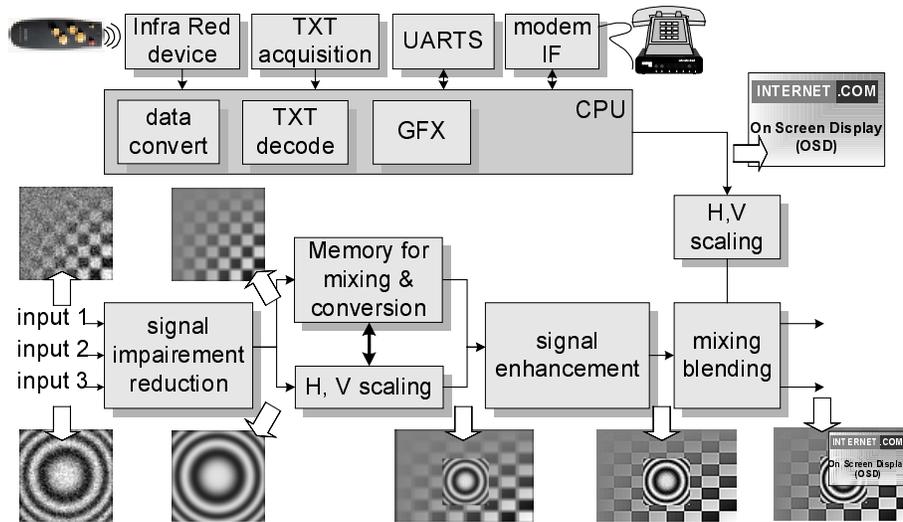


Fig. 6. Feature overview of the system.

field-rate up-conversion and to create the 100-Hz output signal. The figure shows a pair of subsequent fields containing both original video lines or video lines from the median filter. This type of 50-100 Hz conversion is commercially available in some TV sets and known as "digital scan". Let us finally consider the memory requirements for this 100-Hz dual-screen application. The necessary bandwidth equals 300 MByte/s and the amount of memory used is 2.94 MByte. These numbers are computed using similar assumptions as in the previous subsection, and include a 100-Hz graphics signal in the final picture.

V. OVERVIEW OF TV APPLICATIONS

The flexibility of the described system enables a broad range of features which are new in a consumer TV environment. Examples are given in Figure 6, which provides an overview of the application possibilities. Some of the features will be discussed briefly here.

The top half of Figure 6 shows the application area of the telecommunication and control processor (TCP). An infrared (IR) device accepts user control commands from the remote control. Furthermore, acquisition and decoding of Teletext information is carried out and data ports such as UARTS are available. New are the integrated modem interface enabling glueless connection to an existing modem IC and the generation of pixel-based graphics (GFX). The latter also supports Internet applications as is indicated. The processor also provides control of the special video hardware, which is shown at the bottom half of Figure 6 and control of external additional hardware. This flexibility is possible, because the video processing does not require a large cycle budget of the CPU. Control of the video processing is discussed later in this section. Let us first focus

on the aforementioned features.

In the field of Teletext, On-Screen-Display (OSD) and GFX processing, the TCP has to decode Teletext information and generate the GFX in the memory, without severe real-time constraints. However, for GFX refresh rates higher than 10 Hz, the load on the CPU cycle budget becomes significant. The following features could be provided and are supported by the hardware:

- high-quality pixel-based 2D graphics;
- multiple Teletext pages simultaneously and programmable fonts;
- all-page storage for Teletext;
- integration of GFX with photos or images;
- user-defined GFX environment;
- electronic TV program guide application;
- automatic TV controller with user-dependent setting and support via messages and menus.

For the modem functionality of the TCP, the real-time constraints are much more demanding, since the majority of the modem data conversion is executed as a SW program on the CPU. This may result in less performance of the graphics, depending on the application and the software architecture. The modem extension in the TV system offers a new range of telecommunication features, such as

- fax-message and data-file reception, storage and display;
- Internet connection for o.a. program data retrieval;
- interactive TV communication;
- downloading of photo's and images;
- execution of various WinCE applications.

In the bottom half of Figure 6, the application domain of the video coprocessor array (CPA) is shown. The design of the architecture (for an IC floor plan, see Figure 7) is such that video processing can take place with-

out continuous set control. Monitoring is performed on a periodical basis. At the input stage in the coprocessor array (bottom left), video signals may be noise reduced. The video scalers are fully programmable and can be used for compressing or zooming of full-motion video signals. This enables virtually any type of scaling function with a large dynamic range, which results in a very flexible multi-window TV. The setting and control may also be defined by the individual user. The signal quality can be optimized over the several windows. The multi-signal processing capability is very important for composing pictures of various size and contents. In all of these modes, the TCP generates the high-level commands for programming and setting of the CPA coprocessor hardware, thereby enabling for example:

- aspect-ratio conversions (panorama, side-panel, wide-screen);
- PiP, dual-screen, multi-window (arbitrary sizes);
- PiP record and PiP playback;
- mosaic screen for visual channel selection;
- flexible matching to various input/output resolutions;
- high-quality sharpness improvement;
- dynamically moving of video, menu's and graphics.

Finally, as indicated in the Figure 6, graphics and video are blended in full digital form. For this purpose, some of the graphics can be up-converted to a higher resolution, if required. For more details about applications and the quality of the individual coprocessors, the reader is referred to [3].

The most flexible and interesting features are enabled by the configuration that both chips are connected to each other with sufficient SDRAM and the modem function is activated. Whilst looking to a TV program, an image can be retrieved from the Internet and the TV may signal the recovery completion to the consumer. If extra memory is needed temporarily for Internet communication, some memory may be taken from the video processing (e.g. the quality of the 100Hz conversion), in order to boost the microcontroller performance. With existing chip sets for TVs, such a flexibility is unknown to the manufacturer and the consumer.

It should be obvious that the chip-set could also be used in other consumer products than TV set, such as the display signal part of a set-top box, featuring MPEG decoding, electronic programming guide, and telecommunication via telephone for retrieval of a descrambling key. Furthermore, the chip-set could be used as an unified display driver which converts standard-definition signals or VGA-resolution video to a format for Cathode-Ray-Tubes (CRTs), computer monitors, plasma/plasma-addressed-liquid-

crystal (PALC) displays or LCD displays. It can be concluded that the programmable concept behind the applications and the cost-efficient and modular architecture of these ICs give a high degree of applicability for various systems in the consumer market.

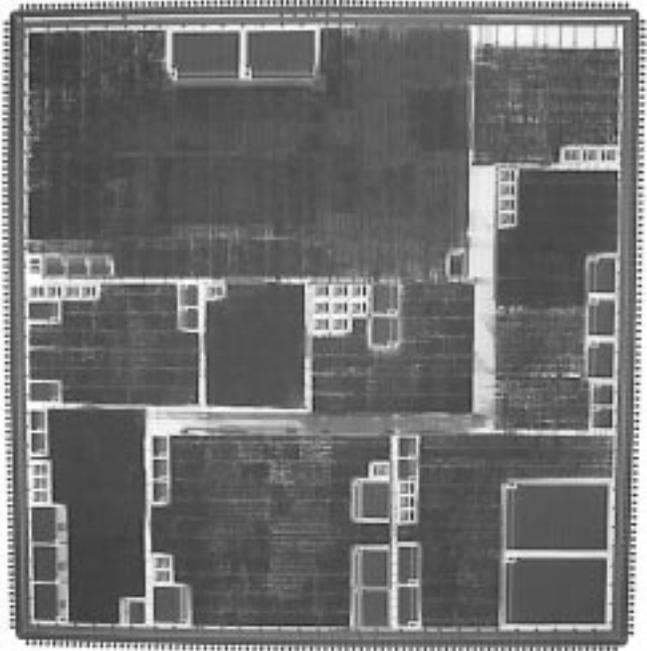


Fig. 7. Floorplan of the CPA design.

VI. CONCLUSIONS

We have presented a chip-set for high-end TV or set-top box, consisting of a microcontroller with a plurality of extensions and a video coprocessor array. The microcontroller consists of a RISC core with a number of peripherals to provide telecommunication features and high-quality graphics. The video coprocessor array offers a number of high-quality TV functions which are programmable and can be used in pluriform ways and for several video signals simultaneously.

The chips may be used in stand-alone operation in combination with existing off-the-shelf external SDRAM. Furthermore, the combination of the two chips results in a highly versatile package of TV applications in which video processing quality and external interactive data communication can be interchanged with great flexibility. User-defined interfaces regarding the use of several video windows combined with the display of side information can be optimized with respect to optimal quality in all circumstances.

The proposed chip-set enhances both quality and programmability of existing TV-sets and it is a key component for approaching the interactive Internet-based multi-window TV or set-top box of the near future.

REFERENCES

- [1] Janssen, J.G.W.M. , J.H. Stessen, P.H.N. de With, "An Advanced Sampling Rate Conversion Technique for Video and Graphics Signals", Proc. of the IPA 97 conf., July, 1997, Vol. 2, pp. 771-775
- [2] Jaspers, E.G.T, P.H.N. de With, "A Generic 2D Sharpness Enhancement Algorithm for Luminance Signals", Proc. of the IPA 97 conf., July, 1997, Vol. 1, pp. 269-273.
- [3] Jaspers, E.G.T., P.H.N. de With, J.G.W.M. Janssen, "A Flexible Heterogeneous Video Processor System for Television Applications", IEEE Trans. Cons. Electron., Febr. 1999.
- [4] Kahn, "The Semantics of a Simple Language for Parallel Programming", *Information Processing 74* (Proc. of the IFIP congress 1974), pp. 471-475, August 1974.
- [5] Leijten, J., *Real-Time constrained Reconfigurable Communication between Embedded Processors*, Eindhoven University of Technology, Ph.D. thesis, November 1998.
- [6] OMI/PI-Bus specifications, "OMI 324: PI-Bus Rev. 0.3d, 1994".
- [7] Steinfatt, O., P. Klapproth and H. Tichelaar. "TCP: A Next Generation for TV Control Processing", *Digest of the ICCE 99 Conference*, June. 1999.
- [8] With, P.H.N. de and E.G.T. Jaspers, "A Video Display Processing Platform for Future TV Concepts", IEEE Trans. Cons. Electron., submitted for Sept. 1999.

Egbert Jaspers was born in Nijmegen, The Netherlands, in 1969. He graduated in electrical engineering from the Venlo Polytechnical College in 1992 and subsequently, he joined Philips Research Laboratories in Eindhoven. For one year, he worked on video compression for digital HDTV recording. In 1993, he continued his education at the Eindhoven University of Technology, from which he graduated in electrical engineering (ir. degree) in 1996. In the same year, he joined Philips Research Laboratories Eindhoven, where he became a member of the TV Systems Department. He is currently involved in the research of programmable architectures and their implementation for TV and computer systems.

Peter H.N. de With was born in Lexmond, The Netherlands, in 1958. He graduated in electrical engineering from the University of Technology in Eindhoven. In 1992, he received the Ph.D. degree from the University of Technology Delft, The Netherlands, for his work on video bit-rate reduction for recording applications. He joined Philips Research Laboratories Eindhoven in 1984, where he became a member of the Magnetic Recording Systems Department. From 1985 to 1993 he was involved in several European RACE projects (a.o. 1001 and 2026) in which digital SDTV, HDTV and data recording was studied. In the early nineties he contributed as a video coding expert to the DV standardization committee. Since 1994 he became a member of the TV System group where he is working on advanced video processing architectures for various enhancements and compression systems. In 1996 he became senior TV systems architect and in October 1997 he was appointed as full professor at the University of Mannheim, Germany. Regularly, he is a teacher of the Philips Centre for Technical Training and for other public post-academic courses on digital video compression and recording and video processing issues. In 1995 he co-authored the paper that received the IEEE CES Transactions Paper award. In 1996, he received a company Invention Award. In 1997, Philips received the ITVA award for its contributions to the DV standard. Mr. de With is a senior member of the IEEE, and board member of the IEEE Benelux Chapter on Consumer Electronics and the Benelux working group for Information and Communication theory.