

A MULTI-CORE SOC DESIGN FOR ADVANCED IMAGE AND VIDEO COMPRESSION

*A. Dehnhardt, M. B. Kulaczewski, L. Friebe,
S. Moch, P. Pirsch*

Institut für Mikroelektronische Systeme
Universität Hannover
Appelstr. 4, 30167 Hannover, Germany

H.-J. Stolberg, C. Reuter

videantis GmbH
Schneiderberg 32
30167 Hannover, Germany

ABSTRACT

A flexible SoC architecture and its hardware implementation targeting advanced MPEG-4 video coding and region-of-interest detection (ROI) is presented. The multi-core architecture integrates three fully programmable processors cores and various interfaces onto a single chip, all tied to a 64-Bit AMBA AHB bus. The processor cores are individually optimized to different computational characteristics, complementing each other to deliver high performance levels with high flexibility at reduced system cost. The SoC is fabricated in a 0.18 μm 6LM standard-cell technology, occupies about 82 mm², and operates at 145 MHz. A surveillance application example includes a MPEG-4 Simple Profile encoder with preceding ROI detection for superior compression results in full TV resolution.

1. INTRODUCTION

The tremendous progress in VLSI technology allows the integration of an ever-increasing number of transistors on a single chip. Likewise, continuous improvements in algorithm research lead to increasingly sophisticated multimedia signal processing applications, demanding a steadily rising amount of processing power. Due to the high innovation rate in this field, the development and standardisation process of these applications is characterized by rapid changes in the algorithms and tools used. A good example is the MPEG-4 video coding standard [1] of the Moving Picture Experts Group (MPEG), which has been introduced in 1999 with the Simple Profile, followed by—among others—the Advanced Simple Profile (ASP) in 2001, and its successor, MPEG-4 part 10 [2] or Advanced Video Coding (AVC), in 2003. In addition to these progresses in standardization, specialized schemes come into view to further improve compression efficiency for certain application domains, e.g., ROI detection for surveillance and closed-circuit television (CCTV).

While the technological progress generally offers the potential to keep pace with the growing processing demands,

the suitability of an implementation for advanced multimedia processing is determined by the architectural concept employed, i.e., how the transistors are actually spent on the chip. In the era of system-on-chip (SoC), multiple processing units can be integrated together with an extensive choice of interface modules on a single chip. In order to meet the demands of multimedia signal processing applications, however, an SoC must provide, in addition to a high level of arithmetic processing power, a sufficient degree of flexibility, integrate a powerful on-chip communication structure, and employ a well-balanced memory system to account for the growing amount of data to be handled when targeting higher-quality applications, e.g., in the area of video.

Existing approaches are either narrowly focused on a specific set of algorithm options, such as dedicated chips for the MPEG-4 Simple Profile [3], or consist of a very general DSP processing core [4] without specialization towards particular properties of the targeted algorithm class, potentially lacking processing performance for schemes with special processing demands. An extension of a programmable core with dedicated modules, as, e.g., in the Trimedia [5], does not help when the functions that have been hard-wired change in a new version of a multimedia standard or in cases when non-standard functionality is demanded.

The HiBRID-SoC multi-core architecture, developed at the University of Hannover, combines high processing power for advanced MPEG-4 coding with high flexibility due to full software programmability. With three programmable cores specifically adapted to different classes of multimedia processing types on a single chip, various on-chip memory modules, and a 64-Bit AMBA AHB system bus, the HiBRID-SoC provides a versatile solution for stationary or mobile multimedia applications such as MPEG-4 video up to full TV resolution or MPEG-4-based surveillance applications. The SoC implementation has already been presented in [6]. This paper adds implementation results for advanced image and video compression applications, utilizing the architectural features of the HiBRID-SoC.

In the following section, the architecture of the programmable multi-core SoC is briefly reviewed, including an over-

view of the three programmable cores and the hardware implementation results. Performance results for advanced MPEG-4 coding schemes with ROI detection are presented in Section 3, and Section 4 concludes the paper.

2. HIBRID-SOC ARCHITECTURE AND IMPLEMENTATION

With the rapid development cycles in today's multimedia algorithm research, programmability is a key requirement for a versatile platform designed to follow new generations of applications and standards. With programmable cores, several different algorithms can be executed on the same hardware, and the functionality of a specific system can be easily upgraded by a change in software.

In general, different approaches exist to accelerate execution on programmable processors. In most cases, some kind of parallelization technique is employed on instruction level (e.g., very long instruction word, VLIW), data level (e.g., single instruction multiple data, SIMD), or on task level (e.g., simultaneous multithreading). Another very powerful means to accelerate multimedia processing is to adapt programmable processors to specific algorithms by introducing specialized instructions for frequently occurring operations of higher complexity [7].

The HiBRID-SoC multi-core architecture comprises three programmable cores: The HiPAR-DSP core, the Macroblock Processor (MP) core, and the Stream Processor (SP) core, as shown in Fig. 1. All three cores have been specifically optimized towards a particular class of algorithms by employing different architectural strategies.

The HiPAR-DSP, previously developed at the University of Hannover [8], is a 16-data-path SIMD processor core controlled by a four-issue VLIW and is particularly optimized towards high-throughput two-dimensional DSP-style processing, such as FFT-intensive applications or filtering. The core consists of 16 identical 16-Bit data paths and a global control unit. Each data path has its own local cache memory for autonomous random access of local data.

A shared on-chip memory, called Matrix Memory, allows concurrent accesses of all data paths in matrix-shaped access patterns. This memory concept provides an easy data exchange between the data paths, which is required for many filter and image processing algorithms. An autonomously operating DMA unit serves all cache misses and performs data prefetch transfers to the matrix memory.

The MP core has been designed specifically for the efficient processing of data blocks or macroblocks that are typical for many video coding schemes. It has a heterogeneous data path structure consisting of a 32-Bit scalar and a 64-Bit vector unit controlled by a dual-issue VLIW, and contains instruction set extensions for typical video processing computation steps. The 64-Bit-wide arithmetic execution units in the vector path, e.g., MUL/MAC or ALU, incorporate

SIMD-style subword parallelism by processing either two 32-Bit, four 16-Bit, or eight 8-Bit data entities in parallel within a 64-Bit register operand. Conditional execution on subword level is supported.

Instructions and data are supplied to the MP via local memories, which are accessible within a single clock cycle. Transfers between external memory and local memories are performed in the background through programmed DMA as the program execution continues.

The SP core, finally, consists of a scalar 32-Bit RISC architecture that is more optimized towards control-dominated tasks such as bitstream processing or global system control with a particular focus on high-level language programmability.

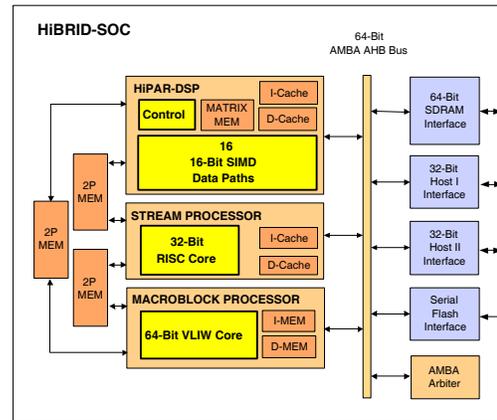


Fig. 1. HiBRID-SoC multi-core architecture.

A 64-bit AMBA AHB system bus [9] connects all cores to off-chip SDRAM memory via a 64-Bit SDRAM interface, to two versatile 32-Bit host interfaces for access, e.g., to a host PC via PCI, and to serial flash memory for stand-alone applications. While the system bus operates at full internal clock frequency, the SDRAM and host interfaces support a programmable internal-to-external clock ratio in order to facilitate adaptation to various system environments. The direct exchange of data and control information between the programmable cores without placing a burden on the system bus is supported by three dual-port shared memories.

The HiBRID-SoC has been implemented in a 0.18 μm 6LM standard-cell CMOS technology and integrates about 14 million transistors on chip [6]. Figure 2 shows the chip photo. In total, the HiBRID-SoC occupies about 82 mm^2 , with more than half of the area consumed by the HiPAR-DSP and its memories. MP and SP core including memories account together for about 30 % of the area, and the rest is occupied by the dual-port memories and interfaces. The chip operates at a frequency of 145 MHz with a power consumption of 3.5 W.

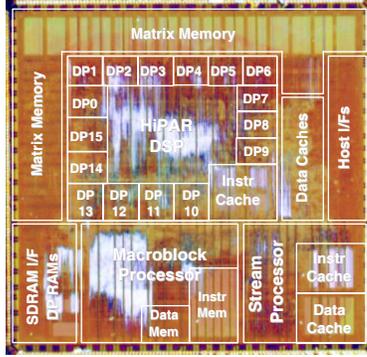


Fig. 2. Chip photo of the HiBRID-SoC.

3. REGION-BASED MPEG-4 ENCODING ON THE HIBRID-SOC

The fully programmable architecture of the HiBRID-SoC facilitates the efficient implementation of a non-standard region-based video encoder. Fig. 3 shows the basic idea of the algorithm. Intended for a surveillance system using a static camera and a very low bitrate transmission channel, the current image is analyzed and regions of interest (ROIs) are determined. These regions are used to control a standard MPEG-4 encoder such that ROIs are coded with a higher resolution than regions that are considered background. The available bandwidth is, therefore, allocated mostly to the ROIs.

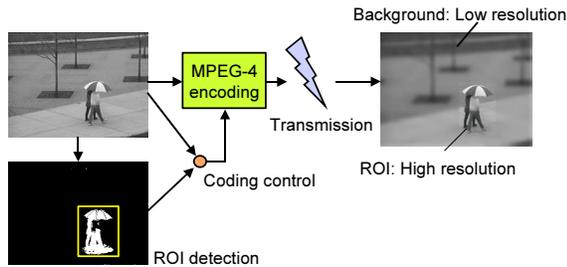


Fig. 3. Region based MPEG-4 encoding.

The tasks of this application are partitioned onto the cores according to their computational characteristics. The HiPAR-DSP core performs the complete ROI detection step, whereas the MPEG-4 encoder is mapped onto the combination of MP and SP.

3.1. ROI Detection

For the ROI detection step, the HiPAR-DSP first detects object pixel candidates by performing a threshold operation against the static background image. The threshold

operation takes into account luminance and chrominance information of each pixel. The result of this processing step is a binary image with object pixel candidates. In order to delete false object pixels due to camera noise, the morphological erosion operator is applied to the binary image. Subsequently, the dilation operator is used to fill gaps in object pixels segments. In the current implementation, both operators use 3×3 neighborhoods. After application of the dilation operator, the binary object pixel image is partitioned into 16×16 blocks, and for each block it is determined whether it contains an object pixel or not. This information is passed on to the MP core via the dual-ported memory between MP and HiPAR-DSP. In addition, the memory address of the currently processed input image is transmitted.

The HiPAR-DSP core with its special memory architecture is highly suited for 2D-image processing applications. The implementation results for the subtasks mapped on the HiPAR-DSP are shown in Table 1. In this implementation, the basic processing unit is a block of 64×64 pixels. This size allows an efficient double-buffering in the HiPAR-DSP's internal memories and, therefore, to completely hide data transfer latencies. At 145 MHz clock frequency and a resolution of $720 \times 576 @ 25\text{Hz}$, the HiPAR-DSP core is utilized only by 23%. More complex object pixel detection algorithms are, therefore, possible while still achieving real-time processing.

Table 1. Region-based MPEG-4 encoder performance on HiPAR-DSP and MP, $720 \times 576 @ 25\text{Hz}$ (worst-case numbers)

<i>HiPAR-DSP</i>			
<i>ROI detection subtask</i>	<i>Cycles/64×64</i>	<i>MHz</i>	<i>%</i>
Object pixel detection	5510	14.87	44.3
Erosion (3×3)	3640	9.82	29.2
Dilation (3×3)	3300	8.91	26.5
Total HiPAR-DSP		33.60	100.0
<i>MP</i>			
<i>Encoder subtask</i>	<i>Cycles/8×8</i>	<i>MHz</i>	<i>%</i>
Motion Estimation	99	32.08	22.4
Motion Compensation	110	26.00	18.1
DCT	138	33.53	23.4
(Inverse-) Quantization	128	31.10	21.7
Full IDCT	190	11.60	8.1
IDCT DC only	21	0.60	0.4
IDCT skipped	8	1.50	1.0
Reconstruction	29	7.05	4.9
Total MP		143.46	100.0

3.2. MPEG-4 Encoding

An MPEG-4 encoder covering all tools from the Simple Profile has been implemented on the MP and SP combination with support for full TV resolution. The quantization control at macroblock level as well as the motion estimation has been adapted to the ROI detection performed on the HiPAR-DSP core.

For predicted video object planes (VOPs), an efficient block matching motion estimation algorithm has to be employed. For each motion vector candidate describing the displacement between the current macroblock and its corresponding reference block, a sum of absolute pixel differences (SAD) has to be calculated. Approaches like diamond or even full search require many candidates to find the best match motion vector leading to the minimum SAD.

To achieve real-time performance, an extended version of the 3D recursive search block matching algorithm (3DRS) has been implemented [10]. This algorithm reduces the number of motion vector candidates to five while the resulting minimum SAD is comparable to other approaches. The ROI information provided by the HiPAR-DSP core is used to restrict the search areas to detected objects. The background area is not searched for motion, which keeps the computational complexity low.

The advanced 3DRS algorithm is divided into two tasks and mapped onto SP and MP causing a well balanced load on the two cores. The SP derives all five motion vector candidates from previous results and purges identical candidates to further reduce the complexity. To keep the DMA data transfer for the reference block pixel data low, all motion vector candidates pointing to overlapping segments are combined to a computational set, and only one DMA transfer is initiated for each set of motion vector candidates. The motion vectors are passed to the MP using a double buffer in the dual-ported memory. Therefore, MP and SP can operate in parallel on different macroblocks with minimal synchronization overhead. While the SP generates the bitstream for the previous macroblock and derives vector candidates for the next, the MP performs SAD calculation for each candidate of the current macroblock. The best match is written back to the SP.

The residual pixel differences for predicted VOPs or the input image pixel blocks for intra coded VOPs are transformed using the discrete cosine transform (DCT). The DCT is efficiently implemented on the MP with its four 16-Bit MACs operating in parallel. After DCT, the quantization step significantly affects the target bitrate. The quantization parameter is chosen according to the position of the current macroblock and the ROI information provided by the HiPAR-DSP core. In most cases, the background macroblocks not belonging to the ROI can be coded as skipped macroblock, which further reduces the bitrate.

The MP transfers the quantized DCT coefficients macro-

block-wise to the dual-ported on-chip memory and signals their availability to the SP for further processing. The SP performs all bitstream related tasks like motion vector coding, DC/AC prediction and variable-length coding (VLC). The extensive parallel processing capabilities of the MP core would be wasted here.

The final MPEG-4 encoded bitstream is transferred via the host interface to a host system, where it can be written to harddisk or transmitted over any network. For the intended surveillance application, the ROI based MPEG-4 encoding scheme provides an average bitrate of 150 to 300 kBit/s for a full TV image. The resulting bitstream is still MPEG-4 compliant, therefore any standard MPEG-4 decoder is able to decode the recorded or transmitted bitstream data.

4. CONCLUSIONS

The HiBRID-SoC provides a powerful and versatile system-on-chip solution for advanced MPEG-4 coding with preceding ROI detection for very high compression rates. With its three programmable cores adapted to different classes of algorithms, different standard and non-standard applications can efficiently be mapped onto this architecture. The full software programmability of all three cores facilitates to keep pace with the rapid algorithm developments in this field.

5. REFERENCES

- [1] ISO/IEC JTC1/SC29/WG11 N4668, "Overview of the MPEG-4 Standard," Jeju, March 2002.
- [2] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, "Study of Final Committee Draft of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC)," Awaji, Dec. 2002.
- [3] M. Takahashi, T. Nishikawa, M. Hamada, T. Takayanagi, *et al.*, "A 60 MHz 240-mW MPEG-4 videophone LSI with 16-Mb embedded DRAM," *IEEE Journal Solid-State Circuits*, vol. 35, no. 11, pp. 1713–1721, Nov. 2000.
- [4] N. Seshan, "High Velocity Processing," *IEEE Signal Processing Mag.*, pp. 86–101, March 1998.
- [5] Philips, *TriMedia TM-1300 Media Processor Data Book*, Sep. 2000.
- [6] H.-J. Stolberg, S. Moch, L. Friebe, A. Dehnhardt, M. B. Kulaczewski, M. Berekovic, P. Pirsch, "An SoC with Two Multimedia DSPs and a RISC Core for Video Compression Applications," *2004 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 330-331, 531, February 2004.
- [7] M. Bereković, H.-J. Stolberg, M.B. Kulaczewski, P. Pirsch, H. Möller, H. Runge, J. Kneip, B. Stabernack, "Instruction set extensions for MPEG-4 video," *Journal VLSI Signal Processing Systems*, vol. 23, pp. 27–50, Oct. 1999.
- [8] J. P. Wittenburg, W. Hinrichs, H. Lieske, H. Kloos, L. Friebe, P. Pirsch, "HiPAR-DSP—a scalable family of high performance DSP-cores," *Proc. of the 13th Annual IEEE International ASIC/SOC Conference*, pp. 92–96, Sep. 2000.
- [9] ARM Ltd., *AMBA Specification Rev. 2.0*, www.arm.com, May 1999.
- [10] G.de Haan and P.W.A.C.Biezen, "Sub-pixel motion estimation with 3-D recursive search block-matching," *Signal Processing:Image communication*, vol. 6, no. 3, pp.229–239, June 1994.