

# Mapping of a Film Grain Removal Algorithm to a Heterogeneous Reconfigurable Architecture

Sean Whitty, Henning Sahlbach, Rolf Ernst  
Institute of Computer and Communication Network Engineering  
Technical University of Braunschweig  
38106 Braunschweig, Germany  
{whitty | sahlbach | ernst}@ida.ing.tu-bs.de

Wolfram Putzke-Röming  
Deutsche Thomson OHG, Germany  
30625 Hannover, Germany  
wolfram.putzke-roeming@thomson.net

**Abstract**— Despite recent advances in FPGA, GPU, and general purpose processor technologies, the challenges posed by real-time digital image processing at high resolutions cannot be fully overcome due to insufficient processing capability, inadequate data transport and control mechanisms, and often prohibitively high costs. To address these issues, we proposed a two-phase solution for a real-time film grain noise reduction application. The first phase is based on a state-of-the-art FPGA platform used as a reference design. The second phase is based on a novel heterogeneous reconfigurable computing platform that offers flexibility not available from other computing paradigms. This paper introduces the heterogeneous platform and briefly reviews our previous work with the application in question, as well as its implementation on the FPGA demonstration board during the first phase. Then we present a decomposition of the application, which allows an efficient mapping to the new heterogeneous computing platform through the use of its diverse reconfigurable computing units and run-time reconfiguration.

## I. INTRODUCTION

Real-time processing of high-resolution digital film data places extraordinary demands on current processing platforms, due to the computation- and data-intensive nature of the applications in this domain. For example, processing large image sizes of 2K (2048x1556 pixels) requires data rates of approximately 233 MiByte/s; at 4K resolution (4096x3112 pixels), which is quickly becoming the standard, data rates increase dramatically to 1147 MiByte/s. Classical approaches based on powerful Application Specific Integrated Circuits (ASIC) or large Field Programmable Gate Arrays (FPGA) using the latest technology are often capable of meeting these data rate and processing requirements, but at the same time present several disadvantages that prompt further research into new solutions. ASIC disadvantages include extremely high one-time research, design, and testing costs (non-recurring engineering costs) that can potentially turn a promising idea into an outdated design. Another disadvantage, especially for FPGA designs, but also common to ASIC platforms, is the necessity for low-level programming, which reduces programming productivity and limits the reuse of existing functionality on other platforms.

Reconfigurable architectures offer an interesting approach to high performance processing platforms, often combining a General Purpose Processor (GPP) and/or configuration manager with reconfigurable processing units whose behavior can be controlled, or reconfigured. These designs offer significant

increases in hardware reuse, allowing different applications to run on limited resources in a time-sharing fashion, as well as the ability to modify control flow and data paths. One such reconfigurable architecture has been developed during the Multi-purpose Dynamically Reconfigurable Platform for Intensive Heterogeneous Processing (MORPHEUS) project. To evaluate the MORPHEUS architecture and approach to application development, a film grain noise reduction algorithm was selected to be mapped to the platform as a case study. The mapping is the focus of this paper, with empirical performance results to follow in later research.

This paper is organized as follows. After a summary of related research, the heterogeneous computing platform that is the basis for this work is presented. Next, Section III presents the film grain noise reduction algorithm and outlines the requirements in this application domain. The development and mapping process, along with results, are described in Section IV. Finally, Section V concludes the paper and presents an outlook for future research.

### A. Related Work

The application examined in this paper was originally designed during the course of the FlexFilm project for an FPGA-based architecture, which is programmed using a custom component-based library called the **Flexible Weakly-programmable Advanced Film Engine (FlexWAFE)** [1] (in press). Despite their increased cost-efficiency over ASIC platforms when dealing with a relatively small product demand, FPGA-based designs are often not the ideal solution due to technology dependence. Many FPGA-based architectures offer libraries of HDL components supplied by the hardware vendors [2], [3], which assists design optimization but increases the dependence on a specific technology and software, thereby producing a significant reduction in flexibility.

In order to mitigate design complexity, a coarse-grained overlay [4] and a corresponding design flow for FPGAs have been introduced. However, as the data width of the coarse-grained Processing Elements (PE) can only be changed between 8 and 16 bits, there is a design complexity vs. flexibility trade-off. This results in a significant area overhead when processing uncommon data widths like 10 bit color components, making strictly coarse-grained solutions not ideal for image processing applications.

Specifically in the digital image processing field, target platforms other than FPGAs are feasible. Several ASIC [5], [6] implementations exist, however each is specialized for a specific application and offers restricted hardware reuse.

Many of these implementations suffer from slow or poorly implemented intra-chip communication, which quickly creates a performance bottleneck even in the presence of massively parallel PEs, large memory resources, and hand-optimized algorithms. To improve intra-chip communication, a Network-on-Chip (NoC) based approach is presented in [7] and introduces a flat control hierarchy while omitting local high-speed control. The MORPHEUS architecture also utilizes an NoC as the backbone of the communication network between processing units, memories, and configuration managers.

An additional issue with many of these implementations, especially when high-end imaging applications are concerned, is data storage and transport. On-chip memories are rarely sufficient and therefore external solutions are required. A variety of memory controllers exist to address this issue, such as controllers by Lee, Lin, and Jen [8] and Whitty and Ernst [9].

Finally, recent flexible graphics processors [10] and the Cell Broadband Engine [11] have entered the image processing segment. They combine impressive computational power and fast memory accesses resulting from high clock frequencies with software-based high-level development environments, presenting serious alternatives for digital signal processing. Compared to the FlexWAFE and MORPHEUS architectures, however, they both lack a flexible and composable control hierarchy, and their granularity is limited to a fixed word width.

## II. HETEROGENEOUS RECONFIGURABLE PLATFORM

The MORPHEUS project is a European collaboration (IST027342) whose main goal is to provide a flexible heterogeneous platform for HW/SW co-design via a unique architecture, composed of reconfigurable computing units of varying granularity. This allows high computation density common to coarse-grained reconfigurable architectures, optimal hardware structure like that found in many System-on-Chips (SoC), and the flexibility and programmability of GPPs, while at the same time attempting to minimize the disadvantages of each platform. Another goal is to provide an integrated toolset to easily map and implement target applications, allowing shorter development times typical for FPGAs.

The architecture is based on an ARM9 processor and three heterogeneous reconfigurable engines, each targeting different types of computation:

- The coarse-grain PACT XPP provides high computation density for stream-based algorithms with deterministic control and dataflow [12].
- The medium-grain DREAM targets computation intensive applications that can iteratively run on small local data memories [13].
- The fine-grain M2000 embedded FPGA is suited for control-dominated tasks and variable data path widths [14].

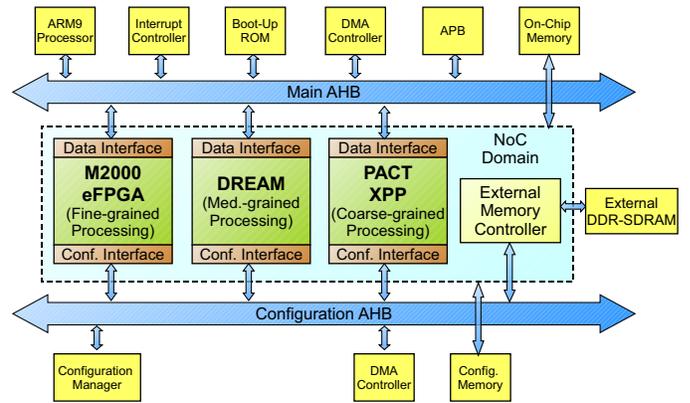


Figure 1. MORPHEUS architecture

The toolset uses C-based high-level descriptions for design entry to the platform and is composed of three modules:

- The retargetable compilation component begins with an application described in pragma-annotated C and produces ARM binary code. Additionally, using the Molen paradigm [15], calls to operations to be implemented on the reconfigurable units are replaced with the appropriate system calls.
- The dynamic control module intercepts calls to hardware and schedules them with the assistance of an RTOS. Together with the hardware configuration manager, it also manages the dynamic reconfiguration of the heterogeneous processing engines.
- The spatial design module performs the actual synthesis of the various operations destined for the reconfigurable hardware units, including memory-oriented task mapping, architecture synthesis and physical synthesis.

The architecture and toolset are detailed in [16].

## III. APPLICATION DOMAIN

The film grain noise reduction algorithm mapped to the heterogeneous computing platform is part of the high-resolution digital image processing application domain. This domain has specific requirements, especially when designing real-time applications.

### A. Application Requirements

Real-time processing of high resolution film or images requires extremely capable processing architectures, as mentioned in Section I. The most common processing tasks are encoding, decoding, and post processing of digital film data, tasks which traditionally have not been executed in real time. With the increasing popularity of video distribution and broadcast methods such as video on demand, however, real-time encoders have become a necessity. Professional post-processing systems are also beginning to demand real-time computation because it allows immediate evaluation of results.

These processing tasks are covered by a broad range of algorithms, many of which perform the same function but have widely variable requirement profiles. For example, some applications operate efficiently in a block-based manner (e.g. MPEG2), while others are primarily frame-based (e.g. fast

Fourier transformations) and some a combination of both (e.g. the application examined in this paper). Consequently, data storage, access, and transport requirements will vary significantly when both small image blocks and complete frames have to be moved and stored.

Ideally, a processing platform provides the flexibility to execute each of these algorithms efficiently.

An additional key requirement for applications like the one examined in this paper is the ability to support an accelerated design flow. This proved to be a weakness of the FlexFilm project, which was only partially addressed by its component-based FlexWAFE library (see Section IV-A).

As with most FPGA-based platforms and many ASIC designs, application development moves at a relatively slow pace because the programmer is forced to work very close to the hardware level. While this can offer increased control over design optimization and therefore speed, in many situations it is not ideal. The problem can be complicated when platforms include heterogeneous computing units without offering additional techniques for application development beyond individually programming each coprocessor. Often a much more efficient overall development process can be achieved when sufficient control and data-flow information for a given application is available to the designer.

### B. A Film Grain Noise Reduction Application

Film grain noise reduction has an important application in the digital cinema market, allowing reduction and even complete removal of unwanted noise from digital film data. For digital cinema, lossless noise reduction is a requirement, making algorithms such as 3DRS [17] that are less bandwidth intensive than full-search algorithms not applicable. The theory behind the algorithm used in this paper is presented in detail in [18].

As clearly shown in Fig. 2, the algorithm consists of three distinct modules: a bidirectional Motion Estimation (ME) unit, a bidirectional Motion Compensation (MC) unit, and a 2.5 dimensional Discrete Wavelet Transformation (DWT). Each element has unique data requirements and processing behaviors. The block-based ME unit begins by comparing a given image to its preceding and succeeding images in a video sequence and subsequently calculating the detected motion vectors. It performs block matching between these images using an exhaustive search algorithm. A series of comparisons and arithmetic operations are executed in parallel on input image blocks to compute the sum of absolute differences (SAD) values of all possible motion vectors. This produces very predictable, content independent memory access patterns in the form of data streams.

The block-based MC unit uses the block motion vectors produced by the ME unit to construct an image that is visually similar to the current image, but contains only pixels extracted in a blockwise manner from the previous or next images, depending on what the unit selects as the best match. Input data is therefore taken from the framebuffers and the ME unit directly, making the MC memory-intensive. Computations are comparison-based.

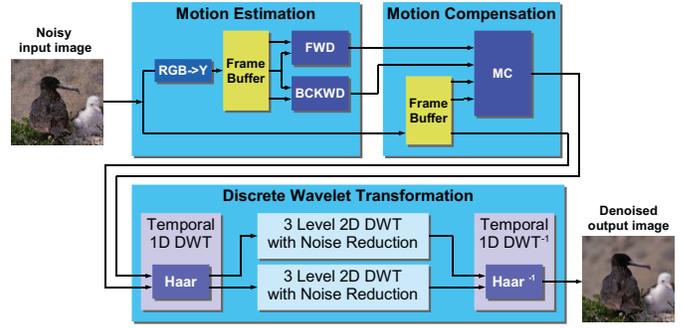


Figure 2. Advanced noise reduction algorithm

The frame-based DWT unit has a very different function. It transforms an input signal into a space where the base functions are wavelets, similar to the way a Fourier transformation maps signals to a sine-cosine based space. It uses a series of Finite Impulse Response (FIR) filters in the horizontal and vertical directions to output a high-pass and a low-pass image stream. Finally, lower pixel values, which represent image noise, are eliminated using a user-specified threshold. Computations range from multiplication, addition, and shift-add operations and vary in width from 10 to 30 bits. Because of the numerous transformation and filter iterations, the DWT is computation intensive.

The complete film grain noise reduction algorithm implementation is presented in detail in [19]. However, this brief overview illustrates that the processing blocks have very different requirements and therefore are best-suited to an architecture composed of heterogeneous processing elements.

## IV. APPLICATION DEVELOPMENT

In the MORPHEUS project, a two phase development approach was chosen. In the first phase, the complex application for reducing film grain noise in real-time from a digital film source was realized on a reference design to verify the algorithmic concept and demonstrate the capabilities of the selected algorithms (see Section IV-A). This implementation is used as a reference design and is compared to the implementation on the heterogeneous platform in phase two (see Section IV-B). An evaluation of the first phase also produced new conclusions regarding the application's characteristics and data access patterns, leading to important new ideas for the mapping to the heterogeneous processing engines.

### A. Mapping to Preliminary Demonstrator Board

The initial implementation on the reference design is based on research conducted during the FlexFilm [19] project using the weakly-programmable FlexWAFE [1] (in press) library. The application was decomposed and mapped to a powerful multi-FPGA hardware/software architecture as shown in Fig. 3. The processing platform is based on three Xilinx Virtex-II Pro XC2VP50-6 FPGAs. Here, the goal was not to use the latest technology, but to utilize an established platform as a reference.

The FPGAs contain the reconfigurable image stream processing data path and are supported by large external SDRAM

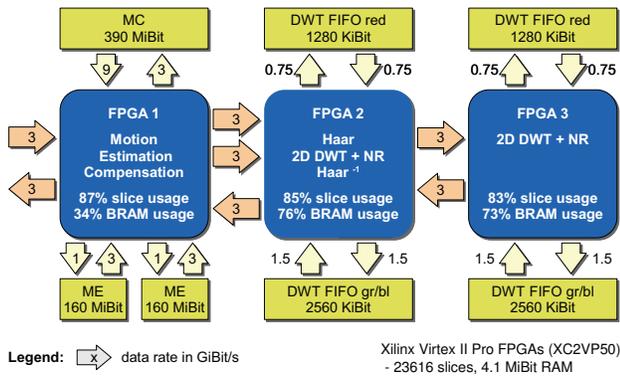


Figure 3. Mapping to demonstrator board

memories for multiple frame storage, as well as a PCI-Express (PCIe) communication backbone network.

The bidirectional ME is realized as a systolic array of  $2 \cdot 256$  processing elements and shares one FPGA with the MC component. The other two FPGAs are consumed by the DWT implementation, as the data path width grows to 30 bit per color component and results in a large chip area requirement. Data transport was realized in a stream-oriented fashion, using a simple three signal protocol with back-pressure to connect processing elements along the data path. For inter-chip communication, a TDMA-based interconnect was designed allowing the multiplexing of several streams within a single channel.

The core difficulties of this implementation were the control and dataflow logic, as FPGAs do not offer predefined structures for data and communication synchronization. As all parts of the application have different data transport and memory access requirements, custom solutions for each algorithm part were created using sophisticated memory access patterns provided by the FlexWAFE library. These mandatory tasks, together with the fine granularity of the FPGAs, led to a lengthy development cycle.

On the other hand, the fine-grained structure of the FPGA enabled the development of an extremely optimized implementation. This led to performance figures reaching 26 FPS for 2K images, which corresponds to 170 GOPS. Another key advantage of the chosen FPGA board is the large amount of on-chip (4.1 MiBit/FPGA) and external memory (512 MiByte/FPGA), which is accessed via several instances of the custom designed memory controller and guarantees an external data rate of 26 GiBit/s per FPGA, thereby avoiding memory bottlenecks.

### B. Mapping to Heterogeneous Platform

Compared to the homogeneous FPGA solution, the heterogeneous MORPHEUS platform offers different processing engines whose characteristics can be exploited for the mapping of the application, which also has a heterogeneous structure. The properties of each processing unit were carefully examined with respect to the application structure before making preliminary mapping decisions. In order to obtain

the optimal mapping of each part of the algorithm, all major processing steps (ME, MC, DWT) have been implemented on two processing engine simulators (PACT XPP, ST DREAM). This provides vital feedback for the final mapping decision presented in the following sections.

1) *Mapping of Algorithm Modules:* The largest processing engine, the PACT XPP, offers powerful 4D DMA engines and implements a Kahn graph-based streaming protocol for data transport [20]. The DMA engines allow the generation of a sliding window memory access pattern, which is essential for implementing the block-based memory accesses required by the ME and MC units. Furthermore, the DMA engines are able to modify the image orientation from a row- to a column-wise pixel representation, which is a requirement for the ME implementation. The XPP's streaming concept is also suitable for the ME implementation, due to its regular memory accesses that allow the composition of a gapless image stream. On the other hand, the XPP uses a fixed word width of 16 bits, which is problematic for the DWT, whose word width varies from 10 to 30 bits per color component.

The fine-grained M2000 FPGA has a variable bit width, which satisfies the DWT's structural demands. However, it is very limited in terms of chip area and, consequently, logic blocks. Therefore, this processing engine cannot be selected for the area demanding DWT or ME implementations without requiring a significant reconfiguration overhead. However, it is an appropriate target device for a smaller part of the application, such as the RGB2Y conversion, which changes the data path width from 32 bit RGB to 10 bit luminance values.

Finally, the DREAM engine provides a mid-grained computation array with extremely fast reconfiguration capabilities (see Section IV-B2). This feature can be exploited for the DWT, which is implemented as a large number of FIR filters placed consecutively in a row. Another advantage of this processing engine is its flexible word width, which is accomplished by combining multiple 4-bit cells into a larger data word. However, this processing engine is also limited in size. One configuration can contain only a single filter stage. Therefore, massive use of the run-time reconfiguration paradigm is necessary, since three stages of direct and inverse filters are required. On the other hand, the DREAM only provides 2D DMA engines, which are not optimized for the block-based frame accesses or data reorganizations that are required by the ME or MC components. This analysis and the results of the simulator implementations led to a decomposition of the application that uses all available reconfigurable units and is depicted in Fig. 4.

2) *Application and Reconfiguration Control:* In phase one, reconfiguration was solely used for run-time programmability of the application. Parameters such as image size and filter coefficients can be modified to adapt the application to a new image format or environment. A complete reconfiguration of the FPGA at run time was not necessary as sufficient chip area was available.

Due to more limited computation resources of the heterogeneous platform, run-time reconfiguration of the various

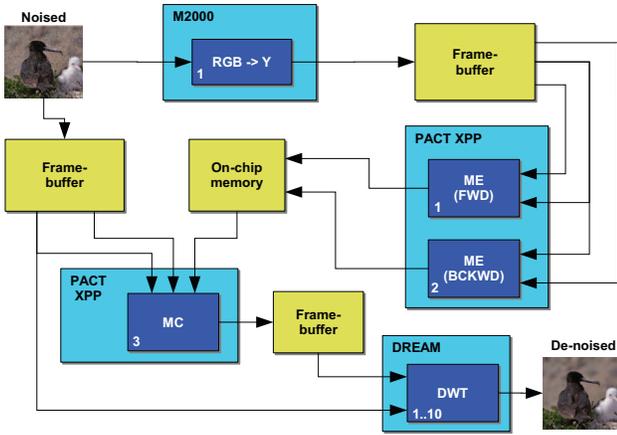


Figure 4. Mapping to heterogeneous platform

processing engines is required to support the application. In Fig. 4, consecutive numbers are assigned to the engines to depict different configurations for the application. The ME and MC components share the PACT XPP in a time-divisioned manner, resulting in three reconfigurations (ME FWD, ME BCKWD, MC) per image. The reconfiguration time of the XPP is approximately 1000 clock cycles. As each of the XPP configurations processes roughly one pixel per clock cycle, the configuration overhead becomes less and less significant with increasing image resolution (SD-TV: 0.2%; 2K: 0.03%). No reconfigurations are necessary for the M2000 FPGA, as it will only contain a single processing step with limited complexity. In contrast, massive run-time reconfiguration is used for the DREAM processor, where 10 configurations are required for each frame.

The exchange of different configurations is controlled by the ARM9 processor. It triggers an internal configuration manager, which is able to buffer several configurations in its dedicated memory. The configuration manager is able to preload configurations in the processing engine’s local configuration memories, effectively hiding the latencies required to fetch a new configuration. For example, in the DREAM unit up to four configurations can be buffered. These configurations can later be swapped within two clock cycles. Because of this feature, the DREAM is more suitable for the DWT than the XPP. Despite requiring only six configurations per frame, the XPP has a significant reconfiguration overhead.

3) *Memory Management*: Besides the mapping of the algorithm modules and its reconfigurations, memory management and data organization is a key design challenge for reconfigurable systems. In Fig. 3, the external data rates for 2K images of the phase one implementation are displayed, resulting in a total demand of 28 GiBit/s (19.5 GiBit/s read, 9.5 GiBit/s write) for the complete application, delivered by 7 memory controller instances.

Because the current incarnation of the heterogeneous platform is equipped with a single memory controller, it is obvious that these demands cannot be completely satisfied. Furthermore, compared to the FPGA board, the heterogeneous

platform offers less on-chip memory (256 KiByte), which prevents buffering of complete image lines and requires a different caching strategy. In order to mitigate the memory controller bottleneck while still allowing real-time processing, the image size has been reduced to a SD-TV (720x576 pixels) resolution. A decrease in the required frame rate below 24 FPS would have allowed processing of larger resolutions; however, emphasis was placed on the real-time requirements of the application.

Additional memories, such as the configuration manager memory, can be directly addressed via the system’s memory map and are converted to data buffers, increasing the overall amount of available memory. This trade-off reduces the amount of possible configurations that can be simultaneously buffered on-chip, but can enhance overall system performance. Results can be directly exchanged between application modules, bypassing slower external memories completely (see Fig. 4). On the other hand, the reduction of available configurations might induce a significant configuration penalty, which will be quantified in future experiments.

Finally, each processing engine has small internal memories, caches, and configuration memories that are used for buffering intermediate values, reordering data streams, and loading new configurations. The resulting two level memory hierarchy of on-chip and local heterogeneous memories increases complexity beyond that of the flat memory model of the FPGA platform’s distributed homogeneous on-chip memories. This necessitates a completely new implementation of the distinct application modules.

4) *Mapping Results*: The results of the mapping activities described above are summarized in Table I. In general, the heterogeneity of the MORPHEUS platform can be efficiently exploited by the different characteristics of the algorithm modules, leading to a complete mapping of the application. Some features, such as the XPP’s 4D-DMA engines and the DREAM’s fast reconfiguration capabilities, alleviate the need to implement specific algorithm parts, resulting in significantly reduced development effort.

On the other hand, the small amount of on-chip memory, limited data rates to external memories, and insufficient data transport mechanisms complicate certain development steps and reduce the performance of the implementation. Due to cost and complexity reasons, the MORPHEUS platform offers reduced memories and fewer caches, which are crucial for the targeted application domain.

Overall, a significant decrease of development time (3 years vs. 1 year) was achieved. Even greater improvements are expected following the availability of the complete toolset, which will eliminate the need to implement the non-trivial amount of software required to control the application and its necessary reconfigurations as well as memory management.

## V. CONCLUSION

In this paper, we have presented the mapping of a film grain noise reduction application onto a novel reconfigurable, heterogeneous architecture. First, the structures and characteristics of the architecture and relevant algorithms were

	Motion Estimation	Motion Compensation	Discrete Wavelet Transformation
<b>XPP</b>	suited: stream-oriented, 4D-DMA	suited: 4D-DMA	not suited: reconf. overhead, fixed width
<b>DREAM</b>	not suited: only 2D-DMA	not suited: only 2D-DMA	suited: fast reconf., variable width
<b>M2000</b>	not suited: too small, RGB2Y possible	not suited: too small	not suited: too small

Table I  
MAPPING RESULTS

described. Second, the mapping of each distinct algorithm unit to a demonstrator platform was explained. Finally, the heterogeneous platform was analyzed and the mapping of the complete application, including memory partitioning and reconfiguration control, was presented. Assets and drawbacks of the heterogeneous platform were identified and their impact on the application implementation was illustrated.

Data transport represented the largest difficulty and caused performance degradation compared to the demonstrator implementation. Despite this weakness, the heterogeneous platform has been shown to be a potentially promising alternative for high-end video applications, since the processing units can be exploited efficiently and the development effort is reduced significantly compared to other solutions.

As described previously, the image resolution requirement for the application was modified in order to maintain real-time processing support, which might give the impression that the implementation on the MORPHEUS platform is a step in the wrong direction. The first phase of application development demonstrated that a Virtex-II Pro-based FPGA board could support both higher resolutions and real-time processing simultaneously. This implies a board based on a single Virtex-5 FPGA from Xilinx or Stratix III/IV FPGA from Altera could also support these requirements through their enhanced processing capabilities and increased size.

The point of this work, however, was to approach the application from a completely different context and evaluate the MORPHEUS platform as a viable heterogeneous platform. The platform proved promising and did not present any weaknesses other than those already identified before the mapping process. It must also be noted that the research-based evaluation version of the MORPHEUS platform used for this work represents only a single instantiation of the architecture. Through its flexible nature, it can easily be expanded to include a larger number of the available heterogeneous processing elements and more importantly, additional on-chip memory and external memory controllers to reduce the memory bottleneck, creating the potential to push application performance to the original levels and beyond.

#### A. Future Research

As the implementation work is just now nearing completion, this paper focuses primarily on architectural principles and not on empirical results. In upcoming experiments, the performance of the new implementation will be evaluated and compared to the demonstrator implementation from phase one, and effects of the memory bottleneck and the configuration memory trade-off will be quantified. Finally, the overall performance of the heterogeneous platform will be measured and evaluated.

#### REFERENCES

- [1] A. do Carmo Lucas, H. Sahlbach, S. Whitty, S. Heithecker, and R. Ernst, "Application Development with the FlexWAFE Real-time Stream Processing Architecture for FPGAs," *ACM Transactions on Embedded Computing Systems, Special Issue on Configuring Algorithms, Processes and Architecture (CAPA)*, 2009, to appear.
- [2] Nallatech Ltd, "DIMETalk 3 Product Brief," 2007.
- [3] Hunt Engineering Ltd. Homepage. [Online]. Available: <http://www.hunteng.co.uk>
- [4] S. Shukla, N. W. Bergmann, and J. Becker, "APEX - A Coarse-Grained Reconfigurable Overlay for FPGAs," in *Proceedings of the IFIP VLSI SoC*, 2005.
- [5] Thomson. Scream 4K/2K Resolution-Independent Grain Reducer. Thomson Grass Valley. [Online]. Available: <http://www.thomsongrassvalley.com>
- [6] DaVinci. Homepage. [Online]. Available: <http://geniusofdavinci.com>
- [7] A. Kumar, A. Hansson, J. Huisken, and H. Corporaal, "An FPGA Design Flow for Reconfigurable Network-Based Multi-Processor Systems on Chip," in *Proc. Design, Automation & Test in Europe Conference & Exhibition DATE '07*, 16–20 April 2007, pp. 1–6.
- [8] K.-B. Lee, T.-C. Lin, and C.-W. Jen, "An Efficient Quality-Aware Memory Controller for Multimedia Platform SoC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 620–633, May 2005.
- [9] S. Whitty and R. Ernst, "A Bandwidth Optimized SDRAM Controller for the MORPHEUS Reconfigurable Architecture," in *Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, April 2008.
- [10] D. Blythe, "Rise of the Graphics Processor," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 761–778, May 2008.
- [11] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Mauerer, and D. Shippy, "Introduction to the Cell multiprocessor," in *IBM Journal of Research and Development*, 2005.
- [12] V. Baumgarte, G. Ehlers, F. May, A. Nüchel, M. Vorbach, and M. Weinhardt, "PACT XPP—A Self-Reconfigurable Data Processing Architecture," in *The Journal of Supercomputing*, 2004.
- [13] F. Campi, A. Deledda, M. Pizzotti, L. Ciccarelli, P. Rolandi, C. Mucci, A. Lodi, A. Vitkovski, and L. Vanzolini, "A dynamically adaptive DSP for heterogeneous reconfigurable platforms," in *Proceedings of the Conference on Design, Automation and Test in Europe*. ACM Press New York, NY, USA, 2007, pp. 9–14.
- [14] Abound Logic. Homepage. [Online]. Available: <http://www.aboundlogic.com/index.html>
- [15] E. M. Panainte, K. Bertels, and S. Vassiliadis, "The Molen Compiler for Reconfigurable Processors," *ACM Transactions in Embedded Computing Systems (TECS)*, February 2007.
- [16] F. Thoma, M. Kühnle, P. Bonnot, E. M. Panainte, K. Bertels, S. Goller, A. Schneider, S. Guyetant, E. Schüler, K. D. Müller-Glaser, and J. Becker, "MORPHEUS: Heterogeneous Reconfigurable Computing," in *Proceedings of 17th International Conference on Field Programmable Logic and Applications (FPL07)*, August 2007.
- [17] G. de Haan, P. Biezen, H. Huijgen, and O. A. Ojo, "True motion estimation with 3d recursive search block matching," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 3, pp. 368–379, October 1993.
- [18] S. Eichner, G. Scheller, U. Wessely, H. Rückert, and R. Hedtke, "Motion compensated spatial-temporal reduction of film grain noise in the wavelet domain," in *SMPTe Technical Conference, New York*, 2005.
- [19] S. Heithecker, A. do Carmo Lucas, and R. Ernst, "A High-End Real-Time Digital Film Processing Reconfigurable Platform," *EURASIP Journal on Embedded Systems, Special Issue on Dynamically Reconfigurable Architectures*, vol. 2007, pp. Article ID 85 318, 15 Pages, 2007.
- [20] G. Kahn, "The semantics of a simple language for parallel programming," in *Information Processing*, 1974.