



# MORPHEUS

---

## **D7.2: Demonstration of selected test case subsets on the final MORPHEUS SoC hardware platform/simulator (advanced draft)**

CONTRACT NO	MORPHEUS IST 027342			
TYPE OF DOCUMENT	Report			
DATE	16/09/2009			
ABSTRACT	This document provides a description of the scenarios that will be demonstrated per case study at the end of the MORPHEUS project.			
AUTHORS	F. Ieromimon, K. Potamianos, N. Voros (ICOM – Leader)			
CONTRIBUTORS	U. Heinkel, J. Knaeblein, A. Schneider (Alcatel-Lucent), G. Gaillat., C. Batarriere (TOSA), H. Sahlbach (TUBS), W. Putzke-Röming (DTO)			
WORKPACKAGE	WP7			
CONFIDENTIALITY LEVEL	Public			
FILING CODE	MORPHEUS-ICOM-D7.2-R0.3.doc			
DOCUMENT HISTORY				
<u>Release</u>	<u>Date</u>	<u>Reason of change</u>	<u>Status</u>	<u>Distribution</u>
R0.1	09/07/2009	Initial ToC	On-going	All WP5 members
R0.2	27/08/2009	All partner contributions integrated	On-going	ARTTIC, Project Co-ordinator

## Table of Contents

1. INTRODUCTION .....	1
2. EXECUTIVE SUMMARY .....	1
3. WIRELESS TELECOMMUNICATIONS .....	2
3.1. DEMONSTRATOR OVERVIEW .....	2
3.2. DEMONSTRATED FEATURES .....	3
3.3. TOOLS USED .....	3
3.4. EVALUATION.....	3
4. NETWORK ROUTING SYSTEMS .....	4
4.1. DEMONSTRATOR OVERVIEW .....	4
4.2. TOOLS USED .....	6
4.3. EVALUATION.....	6
5. SYSTEMS FOR INTELLIGENT CAMERAS.....	6
5.1. DEMONSTRATOR OVERVIEW .....	6
5.2. DEMONSTRATED FEATURES .....	7
5.3. TOOLS USED .....	7
5.4. EVALUATION.....	8
5.5. CONCLUSION .....	9
6. HIGH DEFINITION VIDEO/DIGITAL FILM TECHNOLOGY .....	9
6.1. DEMONSTRATOR OVERVIEW .....	9
6.2. DEMONSTRATED FEATURES .....	10
6.3. TOOLS USED .....	10
6.4. EVALUATION.....	11
7. CONCLUDING SECTION .....	11
8. PUBLISHABLE ABSTRACT .....	12

## Table of figures

Figure 1: ICOM demonstrator system overview .....	2
Figure 2: Screen shot of a simulation run for the QAM Demapper, running a QAM64 modulation scheme.....	3
Figure 3: Tools developed with C# and the .Net 2.0 framework.....	5
Figure 4: Example of Video data without colour failure .....	6
Figure 5: Demonstrator without Ethernet connection.....	6
Figure 6: Motion detection Algorithm overview .....	7
Figure 7: MORPHEUS board (left) with extension board and Ethernet adapter.....	9

## 1. Introduction

---

The purpose of this deliverable is to provide a brief description of the four case studies that will be demonstrated in the context of WP7.

In section 3, the demonstrator of INTRACOM is described. The system demonstrated is part of the 802.16j PHY layer.

Section 4, deals with the demonstration of the application test case chosen by TOSA to assess the MORPHEUS toolset and simulator. After a brief presentation of the demonstration and demonstrator the application synthesis through the MORPHEUS toolset is described. Some suggestions and limitations about the toolset conclude the TOSA section.

In Section 5, the Demonstrator "Network routing systems" by TUC/ALU shows the In-Service-Reconfiguration of a network device. A hardware failure is detected in a network node. Reconfiguration data is submitted through the network and filtered out at the damaged node. When the reconfiguration stream is complete the node reconfigures itself and now acts without the design error. The hardware failure is visualized using a video application. In the first (erroneous) case the colour values are inverted while after reconfiguration the video data is transmitted properly.

Finally, in Section 6 the demonstrator of DTO/TUBS is presented. It consists of the film grain noise reduction application, which is now mapped to the MORPHEUS chip and board. The application parts, which are motion estimation, motion compensation and a discrete wavelet transformation, have been redesigned and re-implemented to match with the heterogeneous concept of the MORPHEUS platform.

## 2. Executive summary

---

This deliverable presents a brief description of the scenarios demonstrated for proving the added value of the MORPHEUS technology.

INTRACOM demonstrates the basic toolset use, using as testing vehicle part of the 802.16j PHY layer. TUC/ALU presents a demonstrator for "Network routing systems" which shows the In-Service-Reconfiguration of a network device. To demonstrate the capacities of the MORPHEUS toolset and MORPHEUS simulator TOSA chose to implement a motion detection algorithm dedicated for the domain of intelligent surveillance systems. This demonstration tests the MORPHEUS toolset and the MORPHEUS chip simulator. The concept of an integrated toolset demonstrates high gain in productivity and re-use with a few knowledge of the hardware. The toolset has limitations due to its youth but it is a very powerful tool. From simulator side PiCoGA and ARM are used for this demonstration and the simulator fulfils his purpose with some possible improvements to optimize the mapping of the application. Combined with the possible use of multiple HREs MORPHEUS achieved to be a high performance easy to use heterogeneous platform.

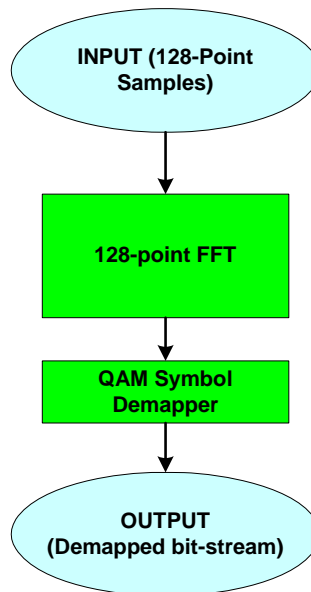
Finally, DTO/TUBS's demonstrator presents the implementation of the film grain noise reduction application on the MORPHEUS demonstration chip, which is operated in the MORPHEUS board. The original application was adapted and re-implemented to fit with the heterogeneous MORPHEUS architecture. Although the originally proposed real-time requirement for high resolution digital cinema images was not met, the MORPHEUS architecture proved to be suitable for complex image processing tasks.

### 3. Wireless telecommunications

---

#### 3.1. Demonstrator overview

The system targeted by ICOM is a reduced subset of the original application specification, calling for a part of the 802.16j PHY layer. Specifically a 128-point FFT and a QAM demapper capable of supporting QPSK, QAM16 and QAM64 modulation schemes were mapped onto the MORPHEUS platform.



**Figure 1: ICOM demonstrator system overview**

The system illustrated above was first coded in 'C' and tested with small datasets consisting of one FFT symbol (i.e. 128 complex sample points). The 'C' code was compiled and run on a PC/Linux system, to provide a reference point for the MORPHEUS implementation.

The 'C' code was subsequently modified, so that the parts targeted for the MORPHEUS DREAM HRE were compatible with the reduced 'C' syntax expected by the SPEAR tool. We discovered that even the reduced demonstrator code did not fit in the DREAM HRE. Therefore, a sequence of use scenarios were captured, synthesized and compiled with the tool-chain and simulated. The use scenarios cover the QAM decoder under all modulation schemes supported (QPSK, QAM16 and QAM64) and the 128-point FFT. Data-generating and check functions placed at the borders of the constructed scenarios were employed to ensure that the simulator results, such as those appearing on the screen-shot below, were consistent.



The initialization, configuration and data-access overhead can be a significant fraction of total execution time, but only when dynamic reconfiguration is required. In these cases, it must be possible to "hide" the time taken for reconfiguration must be able to "hide" in the background of normal function execution. However, for statically configured datapaths that are employed for processing and control of high-rate data streams, the configuration time is an insignificant fraction of system up-time, provided that DMA transfers between DEBs connecting the system components can execute in the background of system processing. In short, we conclude that MORPHEUS is a platform capable of supporting both high-performance statically configured datapaths, and dynamically configurable systems, provided that the associated timing constraints are met by careful system design.

Some work towards user friendliness, especially with the SPEAR tool, will result in a substantially increased tool usability. Specifically, the 'C' source parser needs to support flow-control expressions.

## **4. Network routing systems**

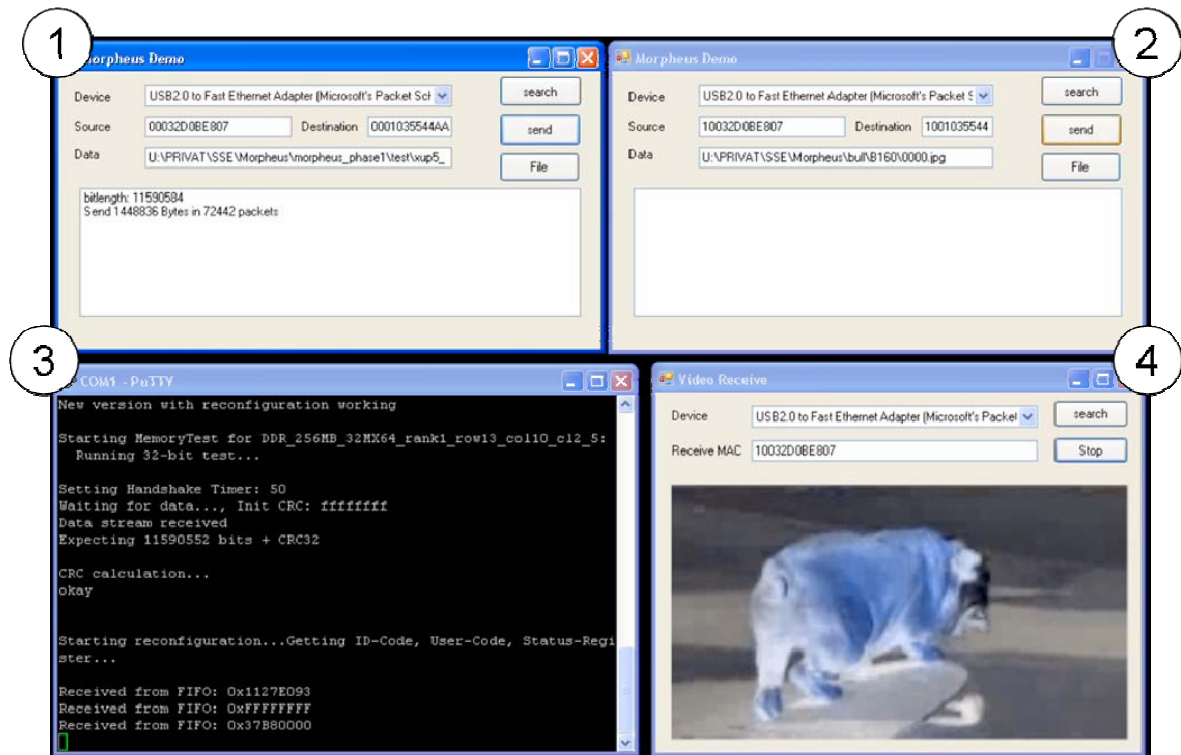
---

### **4.1. Demonstrator overview**

The MORPHEUS demonstration of ALU/TUC shows a new reconfiguration technology for System-on-Chip in a telecommunication network. Design parts of deployed devices are updated based on information distributed to the System-on-Chip with in-band Ethernet packet stream while the network equipment stays in service.

During MORPHEUS demonstration video data will be transmitted from PC to the board via Ethernet interface. At the beginning the ABLogic eFPGA is loaded with an erroneous configuration that changes the colour of the video data and send them back to the PC. The received video data is shown on the PC, so the failure will be visible for the user. After the reconfiguration data was transmitted using the same Ethernet interface to the board, the eFPGA will be reconfigured with a configuration without this failure, so the returned video data keeps its original colour.

Two Windows tool were developed with C# and the .Net 2.0 framework, which control the boards and show the functions. The first one called "MORPHEUS Demo" sends arbitrary data wrapped in Ethernet packages to the board. If the tool recognizes the data as video or reconfiguration data it transmits them in a special format that the receiver understands. The second tool called "Video Receive" shows the returned video data.

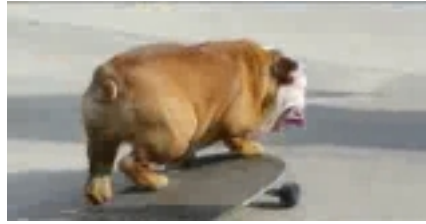


**Figure 3: Tools developed with C# and the .Net 2.0 framework**

The tools are shown on the picture above. Tool 1 and 2 are instances of "MORPHEUS Demo" while tool 4 is an instance of "Video Receive". The RS232 output of the MORPHEUS board is displayed in tool 3, which is an arbitrary terminal program. This RS232 interface is used for debugging issues to display the internal behaviour of the MORPHEUS chip and does not affect any networking functionality.

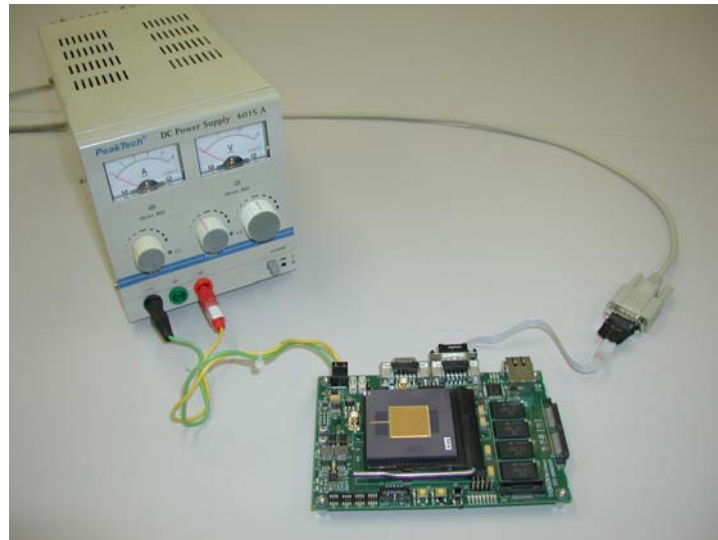
The following steps are performed to run the demonstration:

- Start the MORPHEUS board with the erroneous configuration. Start the four tools presented above on the host PC.
- Push the "search" button in the tools 1, 2 and 4 and select the networking device connected with the board (crossover cable) using the drop down menu.
- Set up the source and destination addresses in the three tools. The source address of the "MORPHEUS Demo" (tool 2) instance that transmit the video data must equal the "Receive MAC" of the "Video Receive" tool. The "MORPHEUS Demo" instance that transmit the reconfiguration data must be configured with the source address "00778899AABB" and the destination address "123456789ABC", because the ABLogic eFPGA filters the data out of packages only with these addresses.
- Start the "Video Receive" tool by pushing the "Receive" button. Then go to tool 2 that transmit the video data, push the "File" button and select the first picture of the video stream. When you click the "send" button the wrong coloured video should immediately start in the "Video Receive" tool.
- Go to the "MORPHEUS Demo" instance that shall transmit the reconfiguration data, push the "File" button and select the bitfile with the correct eFPGA configuration. Click the send button to start the transmission.
- After the transmission the terminal program shows some messages about the successful receiving of the appropriate data and the calculation of the CRC. While the reconfiguration processes, the video stops in the "Video Receive" tool because no data will be returned in this time. The video data without colour failures (see Figure 4 below) is received again, when reconfiguration is completed.



**Figure 4: Example of Video data without colour failure**

The demonstrator (without Ethernet connection) is shown in the following picture.



**Figure 5: Demonstrator without Ethernet connection**

## 4.2. Tools used

Several tools were used for the development of the MORPHEUS demonstrator. First of all SpecScribe, the follower tool of SpecEdit, developed by Alcatel Lucent and TU Chemnitz, was used for a formal specification of parts of the design. The mentioned tool used for the project is now able to generate synthesizable VHDL-code for algorithmic descriptions.

FlexEOS was used to place and route the design for the eFPGA. The basic netlist was generated using Mentor Precision. For compiling the final design and programming on MORPHEUS board the ARM compiler from ST was used. The implementation details are part of the Deliverable 5.7.

## 4.3. Evaluation

The application “network routing systems” shows the capability of the MORPHEUS chip and board to be integrated in a network environment. The functionality of the chip and board are proven as well as the in-service-reconfiguration via Ethernet. The reconfiguration packet filter works properly so the network can be used to transmit user data and reconfiguration data in a mixed mode saving the costs for manual distribution of updates by service technicians.

# 5. Systems for intelligent cameras

---

## 5.1. Demonstrator overview

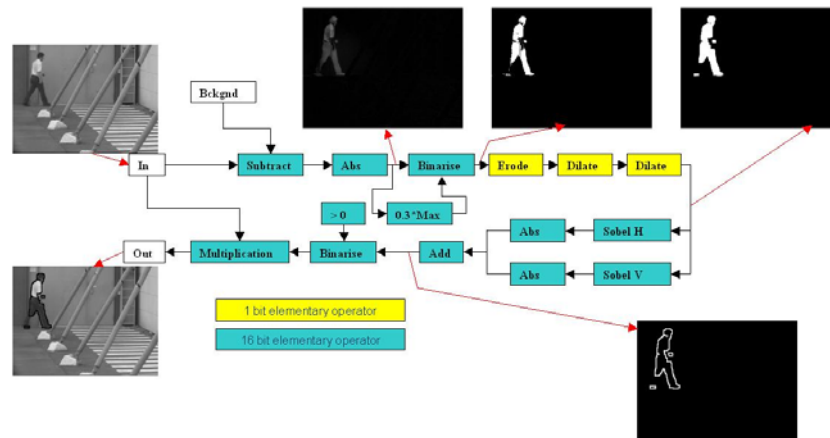
This section provides a brief presentation of the TOSA phase 2 demonstrators for MORPHEUS. This demonstrator aims at implementing a motion detection algorithm dedicated for the domain of

intelligent surveillance systems. While the first phase of MORPHEUS consisted in implementation of the algorithm on a custom platform emulating the MORPHEUS concept, the second phase consists in implementation of the algorithm with the MORPHEUS toolset and the MORPHEUS simulator. Demonstrated features will be presented in section 5.2, then tools used will be presented in sections 5.3 and finally evaluation will be presented in section 5.4.

For algorithm details and implementation considerations see document *D5. : Evaluation Report of phase 1*, *D5.5: Phase 2 specification of application test case subsets* and *D5.7: Implementation and evaluation of test cases applications on MORPHEUS platform*. This document's aim is to give an overview of demonstrator and demonstration.

## 5.2. Demonstrated features

The application used by TOSA is the motion detection algorithm (see figure below) already described in the documents *D5.5: Phase 2 specification of application test case subsets* and *D5.7: Implementation and evaluation of test cases applications on MORPHEUS platform*. This algorithm receives images of a standing camera as input it outputs the same images with moving targets boundaries underlined in black. This simple algorithm presents all the challenging problems of implementation on an embedded low consumption platform. The needs of this algorithm in terms of processing power, memory bandwidth or reconfiguration rate are important and even if the diagram is very simple there are different ways to implement it. *D5.4 : Evaluation Report of phase 1* deals with these various possibilities of implementation.



**Figure 6: Motion detection Algorithm overview**

The main difference with phase1 motion detection is that the background image is not accumulated as it was in phase 1 but initialised with a background reference. This change is due to the nature of processing. Phase 1 demonstrator processed video flow whereas phase 2 demonstrator processes only single images. Another important difference is that only 80\*80 pixels tiles are processed instead of 768\*576 pixels images because of time necessary to process a small amount of data on the simulator. Among the different HREs available in the MORPHEUS chip only the PiCoGA is used for all the operators except maximum value extraction which was processed by the ARM in a first time and replaced by a hard coded threshold during simulation on MORPHEUS simulator. The whole Algorithm control is done by the ARM.

## 5.3. Tools used

The tool used to achieve the demonstration is the MORPHEUS toolset. TRT and ARCES have intensively tested this algorithm with different tools on different HREs. Here the aim is to use the standard toolset without tuning manually any parameter of the application synthesis. The standard toolset even if it is limited on some points allows automatic generation of the application This point is very interesting to enable high gains in productivity. The application is implemented as a succession of elementary operators to enable re-use. These elementary operators are:

- Subtract of two images

- Absolute value of one image
- Maximum value of an image (Processed by the ARM)
- Binarisation of an image against a threshold
- Erosion of an image
- Dilatation of an image
- Sobel filtering horizontal and vertical
- Addition of two images
- Multiplication of two images

Thanks to the integrated toolset the synthesis of our application is very simple. Operators are described in C, which is the first manual input. SPEAR is then used to automatically build the interface of each operator described in C code. This step provides the modules to run by the HREs. The operators could then be stored into an image processing library to enable re-use. Second manual input is the sequence of operators that composes the algorithm which could be the only manual input if the operators required already exist in the image processing library. The sequence of operators is synthesised in an ARM application to control the HREs. The application is then ready to be simulated.

#### 5.4. Evaluation

All the following considerations are available at writing time and are susceptible to change with the toolset updates.

The whole demonstration can't be simulated in a single run. At writing time the toolset doesn't support too complex applications so operator blocks must be executed independently to achieve the whole algorithm. Each operator block is simulated taking output of previous operator block for its input.

The operator blocks run independently are:

- Sub abs binarise
- Erode dilate dilate
- Convolution\_H Convolution\_V abs abs add
- Binarise
- Multiply

Maximum value is not used and is replaced by a hard coded threshold. All the operators are run by PiCoGA. We have to take care that conditional instructions are forbidden for PiCoGA code synthesis.

Only tiles are processed and measures of metrics for WP5 are extrapolated to have figures on an image. Each step result is compared with SystemC software simulation to validate the processing. Extended results are presented in *D5.7: Implementation and evaluation of test cases applications on MORPHEUS platform*.

To fully cope with our need, it would be very useful to have a performance analysis tool that could return measures on the implementation like ARM load rate, memory bandwidth use, internal memory use, load of each HRE... All these data could accelerate the optimisation work by seeing where to put the optimization effort to be efficient. At writing time only number of cycles run by simulator is available. We saw in *D5.4: Evaluation Report of phase 1* that processing power isn't a tuneable parameter to match the implementation with the hardware whereas memory bandwidth and reconfiguration rate are tuneable to fit with the hardware capacities. Getting these parameters would be very useful. This optimization work requires the capacity to modulate the operator blocks running and so to modify the organization of the application as we want. This requirement is not met for the moment because our operator blocks couldn't be composed of too much operators to run properly.

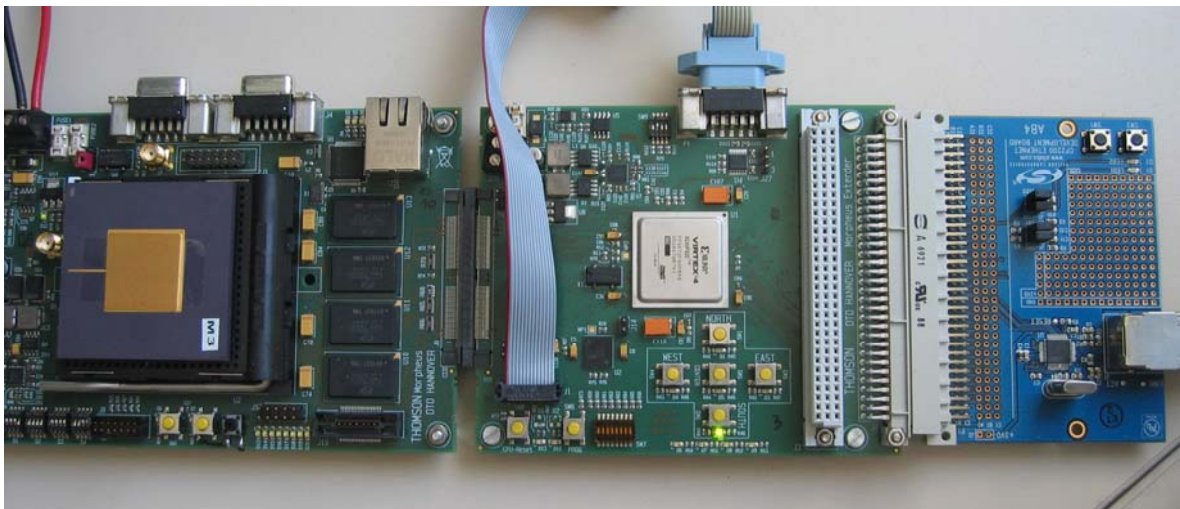
## 5.5. Conclusion

We can conclude that the application is built thanks to the toolset and is running on the MORPHEUS simulator. The application is temporary split to be run on the simulator because the simulator is not yet finished. This point is very important because balancing reconfigurations and memory loadings and downloadings is a powerful lever in image processing implementations. Among the HREs available we only use PiCoGA and so we use only a fraction of the MORPHEUS chip. We have also to keep in mind that no optimisation driven by the application has been done, We have tried to be the most generic possible in operator writing. The advantage of an integrated toolset for generation of an application is huge. The drawbacks are simulation time, youth and some lacks like a performance analysis tool which could be very useful to see where to put the optimization effort to have an important gain on the implementation.

## 6. High definition video/digital film technology

### 6.1. Demonstrator overview

The film grain noise reduction application is demonstrated on hardware. Therefore, the core component of the demonstrator is the MORPHEUS board, which is equipped with the MORPHEUS chip. The MORPHEUS board is a printed circuit board, which was designed by DTO during the course of the project and contains all peripheral components for the MORPHEUS chip. This includes the external SRAM memory, an Ethernet connector, two serial interfaces, some debug ports, a connector for an external clock and the connector for a power supply. In order to ease the image data transport to and from the MORPHEUS chip, an additional extension board was designed by DTO. This board connects to a daughter board interface on the MORPHEUS board and is equipped with a Xilinx Virtex 4 FPGA. The board is able to read and write the SRAM memory and connects to the controlling host PC via an Ethernet interface. The complete setup is depicted in the figure below.



**Figure 7: MORPHEUS board (left) with extension board and Ethernet adapter**

In this setup the instructions for the MORPHEUS chip and the image data is fed to the MORPHEUS board by the PC and the image processing is triggered. As soon as the processing results are available, the de-noised images can be fetched and displayed by the host PC. The DPX file format was kept as the image data format to allow a direct comparison with the reference implementation. QCIF (176x144 pixels) was chosen as the target resolution to reach an acceptable processing speed for the demonstration. Due to the existing memory bottleneck of the MORPHEUS chip implementation and the sequential execution of the single processing steps (for details see D5.6 and D5.7), the result images will not be presented as a video stream with 24 frames per second. Instead the results will be shown image-wise to allow a direct comparison of the noised and de-noised images.

## 6.2. Demonstrated features

The application demonstrated consists of the same image processing steps and algorithms as the application presented at the last review (D7.1). The application starts with a bidirectional block-based motion estimation, followed by motion compensation and a three dimensional discrete wavelet transformation. In wavelet space the noise reduction is performed and the application is completed by an inverse discrete wavelet transformation.

All application parts except the motion estimation have been implemented with the same characteristics as the reference implementation. Due to the hardware resource limitations of the MORPHEUS chip, the motion estimation part had to be modified. The block size for motion estimation was decreased to 4x4 pixels to mitigate the influence of the limited computational resources on the processing speed. As the image size was scaled as well, this decreased block size does not have major consequences on the quality of the processing results. All other application parts are implemented with the same functionality as in the phase one implementation so a direct comparison of both implementations is feasible.

In contrast to the fully parallel solution on three FPGAs on the intermediate platform, the application is executed sequentially on the MORPHEUS chip. The main reason for this sequential execution of the application parts are the memory requirements: the complete application requires 16 data streams to and from off- and on-chip memory. As the amount of available NoC channels to and from memory is limited to a maximum of six channels (two channels for off-chip memory, four channels for on-chip memory), the application cannot be executed completely in parallel. Especially the amount of off-chip memory ports (ten ports in the reference implementation, two ports for the MORPHEUS chip) enforces a sequential execution. During the demonstration, a short sequence of images will be buffered in the external SRAM. The first processing step will be the RGB2Y conversion of these images executed on the eFPGA, followed by the bidirectional motion estimation and motion compensation of the luminance images on the PACT XPP. In the last step, the discrete wavelet transformation is executed on the DREAM and the de-noised images are stored in SRAM again. Finally, these images are read and displayed by the PC. A more detailed description and analysis of the application mapping can be found in D5.6 and in the MORPHEUS book, chapter 15.

## 6.3. Tools used

During the development of the MORPHEUS implementation various established and new tools were used.

For the ABLogic eFPGA implementation of the RGB2Y conversion Emacs was kept as a VHDL editor, as it offers code completion and provides an advanced VHDL editing mode. The synthesis was conducted by Mentor Graphics Precision, the place and route steps were done by the Flexeos tools provided by ABLogic. For the verification of the application's behaviour, the code was simulated in ModelSim in a small ABLogic wrapper, which was originally developed by TU Chemnitz and that was adapted to the application's needs.

For the development of the GriffyC and RISC code for the DREAM HRE UltraEdit was the editor of choice. The code was compiled by the GriffyC and RISC compilers provided by ST. For behavioural simulation the PicoGA debugger and a SystemC processor simulator by CoWare were used.

For the XPP, the NML and FNC-PAE code was implemented using an IDE provided by PACT called xdseditor. For behavioural simulation, the developed programs were simulated in PACT's SystemC simulator (xsim) and debugged using the corresponding debugger (xdbg).

The ARM code, which is needed to control and connect all application parts, was developed with a common text editor. As a compiler, ARM ADS 1.2 was selected. This comparatively old version delivered the best compilation results for the MORPHEUS chip, newer versions caused serious compatibility problem.

As soon as the correctness of the behaviour of the HRE code was verified in the high-level simulation environments, the code was integrated with the ARM C code and simulated using the MORPHEUS HDL simulator in ModelSim. In the final step, the compiled application was loaded to

the MORPHEUS chip via a TK interface developed by ARCES and the results were verified on silicon hardware.

For collaborative work between DTO and TUBS the version control system CVS was used. For automation tasks and to ease the rather complex build process, several scripts and Makefiles were developed.

#### **6.4. Evaluation**

The film grain noise reduction algorithm was successfully integrated in the MORPHEUS chip and the usability of the heterogeneous MORPHEUS platform for complex image processing applications with heterogeneous algorithm characteristics is successfully demonstrated. Certain parts of the MORPHEUS architecture such as the XPP's 4D DMA address generators or the fast reconfiguration capabilities of the DREAM turned out to be beneficial for the application mapping. On the other hand, other elements such as small on- and off-chip memories complicated the application mapping and lead to multiple design iterations.

With regard to the originally proposed real-time requirements, the demonstrator fails to achieve the required frame rate of 24 frames per second for high definition digital cinema resolutions, which is mainly due to hardware restrictions of the actual MORPHEUS chip. These limitations only apply to the actual instantiation of the MORPHEUS architecture. In general, the architecture is suited for image processing tasks. Results in terms of numbers have been presented in D5.7.

Compared to the phase one implementation, the development effort for the MORPHEUS implementation of the application is considerably reduced to the effort required for the FPGA solution. This can be explained by the availability of predefined and programmable data transport mechanism in the MORPHEUS platform. In the phase one implementation, these elements had to be designed individually for each application part.

Finally, the MORPHEUS chip consumes significantly less electrical power than the FPGA board of the phase one implementation (2W vs. 60W). With regard to these figures, the differences in computational capabilities are more or less acceptable as an array of several MORPHEUS chips with modified characteristics should be able to achieve the same processing performance as the reference board at a much lower power level.

### **7. Concluding section**

---

In the previous sections, we provided a brief overview of the scenarios that will be demonstrated in the context of WP7.

As far as INTRACOM is concerned, significant experience was gained with the use of the MORPHEUS tools. However, there are many improvements to be made in order to have a toolset that fully supports the designer during the design process. For example, the overall toolset should be made user-friendlier, releasing the designer from the time consuming process of properly interfacing the various toolset parts. Also, the SPEAR parser should be augmented in order to be able to parse simple flow-control language constructs. The aforementioned problems were the main reasons that forced INTRACOM to reduce the size of the targeted application, which is significant smaller compared to the original one.

For TOSA, MORPHEUS Phase 2 demonstration tests the MORPHEUS toolset and the MORPHEUS chip simulator. The concept of an integrated toolset demonstrates high gain in productivity and re-use with a few knowledge of the hardware. The toolset has limitations due to its youth but is a very powerful tool. From the simulator side TOSA used PiCoGA and ARM for this demonstration and the simulator fulfils his purpose with some possible improvements to optimize the mapping of the application. Combined with the possible use of multiple HREs MORPHEUS achieved to be a high performance easy to use heterogeneous platform.

The demonstrator "network routing systems" shows the reconfiguration of the MORPHEUS chip using an Ethernet network. The reconfiguration data stream was distributed in the network and filtered out at the device. This was shown using a video application which is in the first case colour-

inverted and in the second case error-free. An RS232-interface was used for debugging issues and shows the internal values of the chip during operation.

The DTO/TUBS demonstrator presents the implementation of the film grain noise reduction application, which is described in various work package 5 deliverables, on the MORPHEUS demonstration chip that is operated in the MORPHEUS board. Although the originally proposed real-time requirement for high resolution digital cinema images was not met, the MORPHEUS architecture proved to be suitable for complex image processing tasks.

## **8. Publishable Abstract**

---



# MORPHEUS

---

## **D7.2: Demonstration of selected test case subsets on the final MORPHEUS SoC hardware platform/simulator**

CONTRACT NO	MORPHEUS IST 027342
TYPE OF DOCUMENT	Report
DATE	16/09/2009
ABSTRACT	This document provides a description of the scenarios that will be demonstrated per case study at the end of the MORPHEUS project.
AUTHORS	F. Ieromimon, K. Potamianos, N. Voros (ICOM – Leader), CONTRIBUTORS, U. Heinkel, J. Knaeblein, A. Schneider (Alcatel-Lucent), G. Gaillat., C. Batarriere (TOSA), H. Sahlbach,(TUBS), W. Putzke-Röming (DTO)
WORKPACKAGE	WP7
CONFIDENTIALITY LEVEL	Public
FILING CODE	MORPHEUS-ICOM-D7.2-R0.3.doc

## Context

This deliverable is part of the MORPHEUS project which is a European initiative financed under the 6th FP and addresses innovative solutions for embedded computing based on dynamically reconfigurable platform and tools.

MORPHEUS project aims at satisfying embedded systems new demanding requirements in terms of computing performance, cost-efficient development, functional flexibility and sustainability by developing a global solution based on a modular heterogeneous SOC platform providing dynamically reconfigurable computing completed by a software oriented design flow and a consistent toolset.

MORPHEUS is a 3-year project started in 2006 and gathering all the required expertises from several countries: academics, industrials, SMEs.

## Aim of the deliverable

The purpose of this deliverable is to provide a report on the MORPHEUS scenarios that will be demonstrated in the context of WP7. It includes a description of the tools and the context in which they were used.

## Content of the deliverable

To demonstrate the added value of the MORPHEUS technology, this deliverable provides insight in the scenarios that used for demonstrating the capabilities of the MORPHEUS platform and the MORPHEUS toolset. In that context, INTRACOM presents demonstration scenarios for part of the 802.16j PHY layer.

To demonstrate the capacities of the MORPHEUS toolset and MORPHEUS simulator TOSA chose to implement a motion detection algorithm dedicated for the domain of intelligent surveillance systems. This document provides details on the toolset and on the simulation. Evaluation of the whole synthesis chain is done and some improvements are proposed. Main limitations are due to the youth and to the complexity of the toolset and will disappear with the next versions of the toolset.

The Demonstrator "Network routing systems" by TUC/ALU shows the In-Service-Reconfiguration of a network device. A hardware failure is detected in a network node. Reconfiguration data is submitted through the network and filtered out at the damaged node. When the reconfiguration stream is complete the node reconfigures itself and now acts without the design error. The hardware failure is visualized using a video application. In the first (erroneous) case the colour values are inverted while after reconfiguration the video data is transmitted properly.

This deliverable describes the DTO/TUBS demonstrator, which presents the implementation of an advanced film grain noise reduction application on the MORPHEUS demonstration chip operating in the MORPHEUS board. The demonstrator setup consists of the MORPHEUS board itself and an extension board, which is used for transport image data to and from the external memory on the MORPHEUS board.