# SEVENTH FRAMEWORK PROGRAMME
## THEME ICT-2007 3.4
Computing Systems

### Final publishable summary report

<table>
<thead>
<tr>
<th>Project details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project acronym:</td>
</tr>
<tr>
<td>Project title:</td>
</tr>
<tr>
<td>Project website:</td>
</tr>
<tr>
<td>Project no.:</td>
</tr>
<tr>
<td>Start date of project:</td>
</tr>
<tr>
<td>Duration:</td>
</tr>
<tr>
<td>Coordinating participant:</td>
</tr>
<tr>
<td>Coordinating person:</td>
</tr>
<tr>
<td>E-mail:</td>
</tr>
<tr>
<td>Phone:</td>
</tr>
<tr>
<td>Fax:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s):</td>
</tr>
<tr>
<td>Version:</td>
</tr>
<tr>
<td>Status:</td>
</tr>
<tr>
<td>Dissemination level:</td>
</tr>
<tr>
<td>Filename:</td>
</tr>
<tr>
<td>Last saved:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dissemination level</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PU</strong></td>
</tr>
<tr>
<td><strong>PP</strong></td>
</tr>
<tr>
<td><strong>RE</strong></td>
</tr>
<tr>
<td><strong>CO</strong></td>
</tr>
</tbody>
</table>

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)
## Document history

<table>
<thead>
<tr>
<th>Release</th>
<th>based on</th>
<th>Date</th>
<th>Author</th>
<th>Remarks / Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1v00_RS</td>
<td>0v08_RS</td>
<td>23-06-2011</td>
<td>Kim Sunesen</td>
<td>Finalized</td>
</tr>
</tbody>
</table>
PROJECT FINAL REPORT

Final publishable summary report

Grant Agreement number: 215881

Project acronym: CRISP

Project title: Cutting edge Reconfigurable ICs for Stream Processing

Funding Scheme: Small or medium-scale focused research project (STREP)

Period covered: from 1-1-2008 to 30-04-2011

Name, title and organization of the scientific representative of the project’s coordinator:

Dr. ir. Paul M. Heysters
CEO
Recore Systems BV

Tel: +31 53 4753 000
Fax: +31 53 4753 009

Project e-mail: info@crisp-project.eu

Project website address: www.crisp-project.eu
### Table of Contents

1. Executive project summary ........................................................................................................... 5
2. Summary description of project context and objectives .............................................................. 7
3. Description of the main S&T results/foregrounds ......................................................................... 9
   3.1 General Stream Processor (GSP) ............................................................................................. 9
   3.1.1 General Purpose processor Device (GPD) ........................................................................ 10
   3.1.2 Reconfigurable Fabric Device (RFD) .............................................................................. 11
   3.1.3 General Stream Processor Demonstrator ......................................................................... 12
   3.2 Run-Time Resource Management .......................................................................................... 14
   3.2.1 The CRISP resource manager .......................................................................................... 15
   3.2.2 Exploitation of run-time reconfiguration ......................................................................... 16
   3.3 Dependability ......................................................................................................................... 17
   3.3.1 Dependability approach ..................................................................................................... 18
   3.3.2 On-chip dependability infrastructure ............................................................................... 19
   3.4 Streaming Applications ........................................................................................................... 20
   3.4.1 Digital Beamforming ......................................................................................................... 20
   3.4.2 Global Navigation Satellite System Application ............................................................... 21
   3.5 A holistic and pragmatic approach ......................................................................................... 22
4. Potential impact and main dissemination activities and exploitation of results ......................... 23
   4.1 Potential impact ....................................................................................................................... 23
   4.2 Main dissemination activities ................................................................................................. 25
   4.3 Exploitation and use of results ............................................................................................... 27
5. Address of project public website and relevant contact details .................................................. 28
   5.1 Project website ....................................................................................................................... 28
   5.2 Project consortium overview ............................................................................................... 28
   5.3 Contact information ............................................................................................................... 29
   5.4 Project logo ........................................................................................................................... 30
1 Executive project summary

(see next page)
Executive project summary

In the Cutting edge Reconfigurable ICs for Stream Processing (CRISP) project, a single highly scalable reconfigurable system concept is developed that can be used for a wide range of streaming applications; from low-cost consumer applications to very demanding specialty applications. Under the coordination of Dr. ir. Paul M. Heysters, Recore Systems, the project is performed by an adept consortium consisting of Recore Systems, University of Twente, Atmel Automotive, Thales Netherlands, Tampere University of Technology and NXP Semiconductors. The total project budget is €4.4M and the consortium is supported by the Seventh Framework Programme of the European Union under grant agreement no. 215881 with €2.8M. The project started January 2008 and ran for 40 months till April 2011.

CRISP takes a pragmatic and holistic approach; from hardware manufacturing to software and application developments. CRISP addresses flexibility, programmability and dependability of reconfigurable many-cores for streaming applications. The project is organized in four themes:

1. **Streaming applications** - Targeted applications range from low-end consumer electronics and automotive applications to demanding high-end medical and defense applications.
2. **General Stream Processor (GSP)** - The GSP is a dynamically reconfigurable many-core for streaming applications in low-power embedded systems.
3. **Run-time resource management** – The flexibility and utilization of a many-core can be dramatically improved by dynamically allocating computation and communication resources at run-time.
4. **Dependability** – Improved dependability and yield of deep-submicron chips with new fault detection and localization techniques and (dynamic) hardware fault circumvention.

The main result of the CRISP project is the design, proof-of-concept implementation, and demonstration of tomorrow’s General Stream Processor including:

- A radically scalable many-core architecture demonstrated in silicon.
- Run-time resource management dynamically reconfiguring cores and network-on-chip.
- On-chip dependability infrastructure for hardware test, diagnose, and repair.
- Prototype GSP with 45 Xentium® DSP cores.

The General Stream Processor is demonstrated on both low-cost satellite navigation and high-performance digital beamforming applications.

CRISP results are actively and successfully communicated to a wider audience and to the scientific community through brochures, project website, presentations, publications, exhibition and tutorial activities, press interviews, and press releases. CRISP addresses the imminent challenges and opportunities arising with the continued miniaturization of IC devices for low-power embedded streaming application. Streaming applications have very high market potentials in a competitive market driven by innovation and dominated by short time-to-market and short product lifespan. Reconfigurable many-cores help cope with the uncertainties and increase the life span of the hardware platform. Europe is a major player in the embedded arena today and reconfigurable many-core technology is an excellent opportunity for Europe to accomplish world leadership tomorrow.
2 Summary description of project context and objectives

The CRISP project researches the design of more robust and easy-to-use embedded processors that can adapt to virtually any stream processing application. Already today streaming applications are omnipresent and typical examples include wireless communications baseband processing, multimedia processing, sensor processing, medical image processing, and intelligent antenna signal processing.

In the consumer market, people use ever more digital processing in streaming applications. For example, consider cell phones, personal multimedia players, photo and video cameras and navigation systems. This trend will definitely continue and will drive the demand for more processing power in embedded systems. Especially upcoming multimedia and digital communications applications require massive digital stream processing. New standards provide better quality by using advanced algorithms, which are extremely demanding in terms of processing power. For instance, the computational requirements for wireless communications already exceed the performance of the most powerful state-of-the-art microprocessors. Innovations in streaming applications and improvements in digital processing performance go hand-in-hand. Streaming applications thus have very high market potentials as product differentiators in a competitive market that is driven by constant innovation and dominated by short time-to-market and short product lifespan.

The ambition of the CRISP project is to devise a scalable and dependable reconfigurable many-core platform solution that will drive future technological innovations in the consumer, automotive, medical and defense markets. These innovations are enabled by smaller, cheaper, more reliable and efficient electronic components and systems. The envisioned platform solution includes a massively parallel processing architecture in combination with innovative run-time resource and dependability management. There are many challenges in developing large multi-core platform solutions. CRISP addresses the fundamental underlying problems and the concerns driving this project are indirectly attributable to the miniaturizations of semiconductor technology and translate into three essential questions:

- **How can the intrinsic processing potential of a large multi-core architecture be exploited optimally for a wide range of streaming applications?**
- **How can multi-core systems be programmed efficiently?**
- **How can large multi-core integrated circuits using deep submicron semiconductor processes be made reliable and self-repairing?**

The CRISP project uses a holistic and pragmatic applied research approach to find answers to these questions. CRISP thus addresses optimal utilization, efficient programming and dependability of reconfigurable many-cores for streaming applications. The main objective of CRISP is to create a General Stream Processor for tomorrow’s streaming applications and demonstrate it on both low-cost consumer electronics and high performance applications. The project is organized around four central themes:
Streaming applications - Targeted applications range from low-end consumer electronics and automotive applications to demanding high-end medical and defense applications.

General Stream Processor (GSP) - The GSP is a dynamically reconfigurable many-core for streaming applications in low-power embedded systems.

Run-time resource management - The flexibility and utilization of a many-core can be dramatically improved by dynamically allocating computation and communication resources at run-time.

Dependability - Improved dependability and yield of deep-submicron chips with new fault detection and localization techniques and (dynamic) hardware fault circumvention.

CRISP objectives thus include concrete hardware manufacturing as well as software development of tools, systems, and applications. The CRISP approach demonstrates the synergy of the four themes and in order to achieve the main objective, each of the four project themes have concrete objectives including

| Streaming applications | Use realistic streaming applications for evaluating project results.  
Gain experience with application development on reconfigurable many-core systems.  
Improve the design methodology for developing streaming applications. |
| General Stream Processor (GSP) | Create scalable reconfigurable many-core architecture.  
Integrate dependability infrastructural IP.  
Manufacture reconfigurable many-core IC.  
Demonstrate scalability of many-core architecture.  
Develop software development tools and APIs to program the GSP. |
| Run-time resource management | Develop novel algorithms and software techniques for dynamic run-time mapping.  
Improve resource utilization.  
Demonstrate feasibility of run-time mapping for real-time applications.  
Improve programmability and reconfigurability of scalable many-cores.  
Show adaptability to the presence of faulty cores and communication links in hardware platform. |
| Dependability | Increase dependability of reconfigurable many-cores to sustain a minimum of required quality of service (QoS), while operating.  
Develop dependability infrastructural IP with self-diagnosis capabilities to detect and locate hardware faults in many-cores.  
Develop self-repairing concepts for reconfigurable many-cores using run-time resource management.  
Make testing of deep-submicron many-core chips simpler and less expensive. |

CRISP is a multi-disciplinary endeavor. To achieve the project objectives, highly specialized knowledge is required about applications, front-end and back-end hardware design, tools development, and system software development. An important objective of the CRISP project is to bring together a team capable of executing all aspects.
Furthermore, it is an important objective of the CRISP Consortium to disseminate project results to the scientific community as well as a boarder audience. The project includes concrete dissemination objectives including the creation and maintenance of a project website, printing of an attractive project results brochure, writing of press releases, organization of project tutorial and exhibitions.

Finally, the CRISP Consortium aims to prepare the further exploitation of project results beyond the life of the project including medium- and long-term commercial exploitation and short-term R&D exploitation.

3 Description of the main S&T results/foregrounds

The CRISP project delivers a number of tangible results in hardware and software solutions that combines to deliver the main project result which is the design and proof-of-concept implementation and demonstration of tomorrow’s General Stream Processor (GSP). In the following, we highlight the key results organized around the project themes: streaming applications, general stream processor, run-time mapping, and dependability.

We first present the General Stream Processor; the hardware architecture template and the manufactured prototype General Stream Processor with 45 Xentium® DSP cores. Second, we describe the developed run-time resource management software that is used to dynamically reconfigure the cores and the on-chip and inter-chip network of the GSP. Third, we present the developed on-chip dependability infrastructure for hardware test, diagnose, and repair. Finally, we describe the developed streaming application demonstrators for low-cost satellite navigation and high-performance digital beamforming applications.

3.1 General Stream Processor (GSP)

The General Stream Processor (GSP) is a scalable platform template for performing virtually any streaming application, from low-end consumer applications to high-end applications. The GSP architecture is essentially a heterogeneous many-core system-on-chip (SoC) containing general purpose processor cores (e.g. ARM9™), reconfigurable cores (e.g. Xentium®), and memory tiles. The proposed architecture is designed with the principles of locality-of-reference and concurrency in mind. Hence, different levels of storage are defined in the CRISP many-core architecture; local data memories are incorporated in Xentium processing tiles and distributed on-chip memory is available in the GSP by means of the Smart Memory Tiles. Moreover, concurrent processing of computation kernels in the streaming applications can be performed on the many parallel Xentium processing tiles in the GSP. The Network-on-Chip and the distributed on-chip memories provide the communication infrastructure in the GSP to manage the data streams between the computation kernels in the streaming applications.

Within the scope of the CRISP project, a scalable GSP platform is built and subsequently demonstrated using many multi-core devices that are connected to each other to create a large many-core system-of-chips. The GSP demonstrator is built from two kinds of chips: a General Purpose processor Device (GPD) and Reconfigurable Fabric Devices (RFD). The GSP depicted in Figure 1 combines one GPD and five RFDs to build a many-core with 46 processing cores: one ARM® processor and 45 Xentium DSP cores. The
Xentium DSP core is explained in more detail below. The inter-chip connection is handled by a dedicated chip-to-chip interface that extends the Network-on-Chip (NoC) of the RFDs across multiple chips. The chip-to-chip communication is instrumental in demonstrating scalability as it allows NoC communication to extend seamlessly across chip boundaries.

![Image](image-url)

*Figure 1: Illustration of a GSP demonstrator connecting one GPD chip and five RFD chips in an off-chip network. This GSP connects 45 Xentium DSP cores and one ARM® core in one network*

Results on the General Stream Processor are published in:


### 3.1.1 General Purpose processor Device (GPD)

The GPD is based on a traditional bus-based architecture and is depicted in Figure 2. It is designed to run at a clock frequency of 200 MHz, and contains an ARM926 processor, ROM, SRAM, and a wide range of peripherals. Among the common interfaces of the GPD, special focus has been given to the chip-to-chip (C2C) interface. In the GSP demonstrator (refer to Figure 1) the C2C interface bridges the GPD with the NoC of the RFDs giving the GPD full access to all RFD resources.
3.1.2 Reconfigurable Fabric Device (RFD)

The RFD contains the reconfigurable hardware blocks of the GSP platform. The RFD is a tiled processor comprising a grid of tiles interconnected via a packet-switched NoC. It is a complete SoC with I/O, testing infrastructure, and clock and reset management. In CRISP, a twelve-tile SoC architecture is implemented. The diagram of the twelve-instance RFD is given in Figure 3. The RFD contains the following operational components that can be used to implement streaming DSP applications:
The main communication infrastructure in the RFD is provided by the NoC. All Xentium processing tiles and memory resources are accessible via the NoC. Moreover, the NoC can be used to configure the clock manager of the RFD, and to access scan chains and memory Built-In Self-Test units of the Xentium tiles. The NoC provides means for on-chip scalability of the GSP template. Hence, larger many-core SoCs can be designed by increasing the number of routers in the NoC. In the GSP demonstrator, off-chip scalability is addressed by extending the NoC across the RFD chip boundaries using dedicated MCP interfaces. The MCP provides a transparent off-chip interface to bridge two RFDs or to connect the RFD with the C2C interface of the GPD. The MCP interfaces enable prototyping of large many-core systems-of-chips. Scaling the packet-switched NoC to larger dimensions results generally in increased NoC routing overhead because of the increasing number of hops in the NoC. Therefore, the CRISP project researches hierarchical addressing and NoC routing. The Die Link Interfaces (DLI) provides means for hierarchical addressing to the GSP.

The Xentium® DSP IP Core is a programmable high-performance digital signal processing (DSP) core. The Xentium VLIW\(^1\) architecture provides high-performance and energy-efficiency by optimizing parallel operation at instruction level. All communication with the Xentium tile is performed using memory-mapped I/O; all modules in the Xentium tile are given a dedicated address range in the Xentium memory map. The memory map for each Xentium is defined by a reconfigurable routing table in its network interface. The network interface also implements direct memory access (DMA) reading and writing the tightly coupled memory of the Xentium Tile. The core modules of the Xentium tile are the Xentium core, tightly coupled data memory and a NoC interface. Moreover, the Xentium tile contains additional logic that is used to control scan chains and memory Built-In Self-Test at run-time. The additional logic is part of the Dependability Infrastructural IP, described below where it is referred to as the Xentium Tile Wrapper.

The Smart Memory Tiles (SMT) of the RFD provide shared memory that is accessible through the NoC. Besides being accessible with random access, the SMT has multiple reconfigurable Address Generation Units allowing to configure FIFO functionality or implement elastic buffering.

### 3.1.3 General Stream Processor Demonstrator

The CRISP General Stream Processor demonstrator is realized by manufacturing the GPD and RFD chips and installing them on a PCB that serves as prototype and verification platform. Figure 4 shows an overview of the implementation. The PCB installs one GPD and five RFD chips connected point-to-point to seamlessly extended the RFD network-on-chip across all chips. Besides connecting the GPD and RFD chips to prototype the GSP,
the PCB serves as verification platform with an FPGA and various interfaces including high-speed data links and RF. The verification platform is used both for post-silicon validation of the GSP and for implementation of the beamforming and GNSS streaming applications. The GPD and RFD chips were manufacture in respectively 130nm and 90nm CMOS; and both run at a clock frequency of 200 MHz.

![Figure 4: Overview of the hardware of the General Stream Processor demonstrator](image)

The Xentium cores of the RFD were implemented as Hardmacros. Figure 5 shows a photo of the RFD die with the nine Xentium Hardmacros, the two Memory Tiles (in the middle), and clock and power blocks (in the lower left corner). The logic of the NoC is laid out between the processing and memory tiles. The dependability manager is also laid out as glue logic with most of the logic in the lower left as indicated on the photo.

Hardware without software is of little use and the CRISP project also developed and ported the software and tools needed to run applications on the GSP. The GPD runs Linux with the necessary drivers for accessing the RFD network-on-chip and other IOs. The GPD can be programmed in C using an ARM tool-chain. The Xentium DSP cores are operated as bare-metal machines and programmed in C or assembly using the Xentium tool-chain. To ease the programming of the GSP platform, a C-API\(^4\) is developed the platform. The API provides high-level access to all platform resources. On top of this API, advanced system software is running to support the dynamic detection of platform components and automatic (re)configuring of the platform used by the run-time resource management and dependability software.

\(^4\) Application Programming Interface
3.2 **Run-Time Resource Management**

In contrast to conventional application development, applications are mapped on the GSP hardware at run-time instead of at design-time. In the CRISP project, a run-time resource manager is devised and implemented that automatically determines on which cores tasks are to be executed and which parts of the communication infrastructure are to be used, given optimization criteria. Both spatial and temporal criteria are taken into account by the resource manager. Fully integrated with the GSP hardware, the resource manager software computes resource assignments and (re)configures the platform resources. Run-time resource management is key when developing adaptive systems for streaming application that must combine flexibility with real-time latency and throughput requirements. Results from the CRISP project are published in:

3.2.1 The CRISP resource manager

Figure 6 illustrates run-time resource management. The resource manager receives queries for resources from applications and based on the current resource availabilities it attempts to compute a resource assignment. If successful, the application is admitted to execute and the available platform resources are updated. If not successful, the application is rejected as its resource demands could not be met. Note that the resource manager updates dynamic information about applications and platforms. It has thus no prior information about the applications to be handled and it can handle (run-time) changes to the platform.

![Diagram of run-time resource management](image)

**Figure 6: Illustration of run-time resource management**

The GSP platform incorporates a distributed memory model, and besides the efficiency arguments, the Xentiums are not designed to run any middleware. Therefore, the GPD manages the resources of the connected RFDs in a centralized manner. Each application requests the amount and type of resources required for its execution. The resource manager determines whether these requests can be fulfilled. The resource manager shields the tasks that are already present on the platform from the potential interference with newly started applications. The resource management enforces the following policies:

- **admission control** – an application is only allowed to start if the system can allocate, upon request, the resources required to meet its performance constraints;
- **guaranteed resource provisions** – the access of a running task to its allocated resources cannot be denied by any other task.
If an application cannot be added to the system without a violation of this policy, then the resources for the application will not be allocated. Hence, in that case the application is refrained from execution on the system by the run-time resource manager.

Besides the book keeping, the resource manager has to solve a complex resource allocation problem within a short time. The resource allocation problem concerns two dimensions: the spatial and temporal dimension. Explicitly considering the (relative) location of resources within a platform may avoid inefficient resource allocations, where efficiency may be measured in the amount of resources being allocated, or in energy consumption of the platform. Orthogonal to the spatial dimension, we consider the temporal dimension. This does impose some uncertainty, as we assume that we do not know in advance which applications are started or stopped at a certain time. Therefore, optimal solutions cannot be guaranteed because it depends on future events. The resource management algorithms thus resort to finding feasible solutions, while optimizing towards secondary objectives, such as minimal energy consumption or highest performance. Additionally, in the case that multiple applications may be executed simultaneously on the platform, a reasonable platform state has to be maintained that allows for more than one resource request to be fulfilled. Given these properties and the complexity, heuristics are used to device and implement fast algorithms. Graph representations are used for applications and platforms to formulate the resource allocation as a mathematical constraint optimization problem.

### 3.2.2 Exploitation of run-time reconfiguration

The GSP hardware platform is scalable and the resource manager scales with it to ensure programming efficiency. The configuring of the platform includes the assignment of tasks to specific Xentium cores, and configuring the routing tables of the NoC with routing information. When a programmer performs these steps (manually) at design-time, an assumption must be made on the availability of resources. This results in potential conflicts, when resources are in use by other applications, or when resources are unavailable due to hardware faults. A more extensive design-time analysis can only consider a limited number of use-cases, and event- or user-triggered actions are difficult to anticipate on. Any change in the application or in the platform that is not captured by the design-time analysis, results in incorrect or unpredictable behavior.

Even in application specific platforms, where less flexibility is required, resource allocation at run-time can have additional advantages in fault tolerance. For example, when faults are detected by a postproduction test, otherwise disabled spare parts may be used to replace the malfunctioning components; this is called static redundancy. A chip then may still be usable if the number and type of faults are non-critical. This approach is used in the manufacturing process of e.g. the Cell processor, where not all eight Synergistic Processing Elements are required, resulting in a higher production yield. Such scenarios only work when the application mapping to the architecture can deal with these deficiencies. On the other hand, safety critical systems often use online fault detection mechanisms. With redundant resources available, the system may continue its operation if a detected fault can be isolated. Thus, to provide support for fault tolerance, a run-time resource manager should be in place to account for the free, allocated and faulty resources in the system.
Larger systems, such as heterogeneous computing clusters, use a distributed memory model to overcome the high communication overhead of shared memory systems. Such systems often use middleware to present the system in a homogeneous manner. However, distributed memory multi-processor systems are in general heterogeneous, and it may be quite expensive in terms of performance and energy consumption to expose them as homogeneous systems. Keeping the property of homogeneity may even be considered impossible, due to asymmetric resource allocation for multiple applications simultaneously running in the system. Considering parallel programming, the most complex platform is a multi-user, heterogeneous cluster made up of processors of different architectures, interconnected via a heterogeneous communication network.

One important aspect of resource allocation is the maintenance of precise and consistent state information of the GSP platform. Therefore, the resource management system software needs to know which cores and links in the SoC are available to map tasks. By using feedback from dependability software on faulty components, the resource manager knows about broken links and erroneous cores in the GSP.

### 3.3 Dependability

The dependability of large scale multi-processor systems-on-chip is becoming an important concern, especially when the SoCs are used in mission-critical applications. Dependability is the extent to which a system can be relied upon to perform its intended functions under defined operational and environmental conditions at a given instant of time or given interval. In the CRISP project dedicated hardware and software is developed to implement and demonstrate embedded dependability to make systems self-testing and self-repairing. The CRISP approach improves reliability, availability, and maintainability of embedded systems. The CRISP General Stream Processor implements an on-chip dependability infrastructure that allows testing for faulty cores and interconnect components while the GSP is operational. In the CRISP project, a dependability manager and a Xentium tile processor wrapper are devised and implemented on the RFD chip. To use the hardware, dependability software is developed to test the network-on-chip and to control the use of dependability testing. Moreover, dependability features and information are integrated with the run-time resource manager to demonstrate self-repair and graceful degradation by circumventing faulty hardware. Results on dependability have been published in:


3.3.1 Dependability approach

Manufacture testing is a conventional method to check the correctness of chips using structural tests and built-in self-tests. Manufacture testing uses dedicated Design-For-Test (DFT) logic and is done once before chips are shipped and used. The testing detects permanent hardware faults and blue-prints each successfully tested chip as “correct” independent of any application. The CRISP dependability approach lifts and reuses DFT logic to perform structural tests and built-in self-tests of cores at run-time, that is, while the chip is in use and possibly running user applications. The testing allows to detect and localize cores with permanent hardware faults. Dedicated dependability software running on the GPD is developed to control the testing which is done either on request or periodically. The testing can be done in the back-ground while user applications are running. The dependability software also implements run-time testing of the NoC to ensure reliable communication.

In combination with run-time resource management, run-time dependability testing allows for fault-tolerance or self-repair when hardware faults can be circumvented. In case spare resources are available, the resource manager can remap the application to use these. In case, the platform is either seriously compromised, or its available resources are near depletion, applications may not be allowed to execute. For robustness reasons, even more flexibility may thus be added to an application. By providing multiple quality-of-service levels the probability may increase that an application is allowed to start, albeit in a reduced form. Other scenarios exist where it is preferred to stop secondary applications to free resources to remap compromised primary applications. The CRISP approach to
self-testing and self-repairing thus allows the General Stream Processor to gracefully degrade to prolong its life time while spare resources are gradually depleted and/or quality-of-service is gradually reduced.

### 3.3.2 On-chip dependability infrastructure

To permit testing of tiles at run-time this method requires additional on-chip infrastructural IP that is designed and integrated in the RFD architecture at design-time. Figure 7 shows the most important components of the dependability architecture: the Dependability Manager (DM), the Xentium tile wrappers around the Xentium tiles, and the GPD executing the Xentium tile dependability software. During dependability testing, the NoC is used as a test access mechanism to transport test stimuli to and test responses from the wrappers of the tiles-under-test. The NoC makes no difference between the transportation of functional data and dependability test data.

![Diagram of GSP dependability architecture for the Xentium tiles](image)

**Figure 7: GSP dependability architecture for the Xentium tiles**

The Dependability Manager (DM) shown in Figure 7 consists of a test pattern generator (TPG), a test responses evaluator (TRE) and a finite state machine (FSM). The FSM controls the DM and communicates with the dependability software running on the GPD. Deterministic test patterns for the Xentium tile(s) have been generated at design-time. The TPG in the DM reproduces these deterministic test patterns at run-time using a linear-feedback shift register combined with a reseeding technique. The silicon area of the DM is less than two percent of the total area of the RFD. The Xentium Tile Wrapper allows switching the operating mode of the Xentium tile between functional mode and dependability test mode via commands issued by the DM over the NoC. In functional mode, the wrapper is transparent. In dependability test mode, the wrapper controls the
input and output of the tile. Test data from the DM received via the NoC is forwarded to the test logic of the tile and test results from the tile are sent to the DM via the NoC. Since the NoC makes no difference between the transportation of functional data and dependability test data, the total bandwidth of the NoC has to be shared between the application’s functional data and the dependability infrastructure’s test data. Dynamically pausing and resuming of the dependability test is therefore supported and used to ensure that the required NoC bandwidth is always available for the running applications. This prioritization of functional data over dependability test data on the NoC ensures that the dependability tests do not impact the performance of the user application(s).

### 3.4 Streaming Applications

In CRISP, concrete streaming applications are worked out for digital beamforming and Global Navigation Satellite System (GNSS) reception. Digital beamforming is chosen as a typical example of a high-end application with high throughput and high computing demands. Digital beamforming is widely used in defense and space applications and is also becoming increasingly popular in consumer electronics applications such as femto-cell and wireless access gateways. Satellite navigation systems have become omnipresent in e.g. cars, smart phones, and wrist watches. GNSS is chosen as a typical example of a low-end consumer application. The ambition of the CRISP platform is to scale from low-end to high-end applications using the same reconfigurable multi-core System-on-Chip template. Results on streaming applications are published in:


#### 3.4.1 Digital Beamforming

Digital beamforming is used in an increasing range of products like sonar systems, radar systems, radio astronomy telescope systems, and base stations for wireless telecommunications. In the CRISP project, the digital beamforming application is derived from the radar field where requirements are demanding in terms of both data throughput and processing power. For instance, a system with sixty-four antenna receive channels typically has input rates of tens of Giga bytes per second and requires several Giga Multiply Accumulate (MAC) operations per channel.
Figure 8: Functional architecture of a digital beamformer system

Figure 8 depicts the functional architecture of a beamformer system. In the analog part, signals are received from multiple antennas and converted into digital signals. In the digital part, antenna processing is applied to each signal for calibration or equalization and beamforming combines signals into beams that are further processed. The CRISP beamforming demonstrator uses 39 of the 45 Xentium DSP cores of the GSP.

3.4.2 Global Navigation Satellite System Application

Satellite navigation applications of today range from cheap, single-frequency receivers embedded in mobile phones to expensive, multi-frequency, centimeter-accurate scientific receivers. In the CRISP project, the GNSS application is specified and designed to support the existing U.S. based NAVSTAR Global Positioning System (GPS) and the future Galileo (European satellite navigation system) signals transmitted in the L1 frequency band.

A GNSS receiver has three main blocks i) a radio front-end for analogue signal processing (shown as Radio in Figure 9), ii) a digital baseband processing part for navigation data decoding and signal time-of-arrival measurements (shown as Digital Baseband Processing in Figure 9), and iii) navigation calculus to determine position, velocity and time (PVT) (shown as Navigation Task in Figure 9). The digital baseband processing tasks are: acquisition (search for satellites) and tracking. The GNSS application is demonstrated on the GSP using the DSP cores of one RFD. The satellite signals are received using a commercial off-the-shelf radio front-end installed on the GSP verification board.

Figure 9: Illustration of GNSS application tasks.
While most of the current state-of-the-art GNSS receivers are using dedicated hardware for digital signal processing (ASIC technology), the trend is going towards Software Defined Radio motivated by the ongoing modernization efforts in satellite navigation systems. The European Galileo system together with systems from Russia (GLONASS) and China (Compass) are emerging to compete with U.S. based GPS, which is also undergoing a modernization program. The specifications of new systems are still evolving and also the algorithm development in the field of multi-system reception is active. Thus, future receivers should have high computational power and flexibility for updates. A scalable platform with multiple reconfigurable processing tiles meets both of these requirements.

3.5 A holistic and pragmatic approach

The holistic and pragmatic approach of CRISP includes hardware design and manufacturing as well as development of tools, applications, and system software. This approach makes it possible to advance research into fundamental trade-offs in the intersection of hardware and software where reconfigurable computing resides. The scalable NoC-based many-core General Stream Processor (GSP) architecture developed in CRISP delivers sufficient and scalable stream processing performance for a wide range of applications in both high- and low-end markets. The GSP is equipped with a sophisticated run-time resource management system that can dynamically change the allocation of application tasks to processing cores and data transfers to communication channels. The run-time resource management turns the GSP into a flexible and efficiently programmable coarse-grained reconfigurable computing platform. Moreover, CRISP combines the run-time management with dedicated dependability hardware and software to create a self-repairing dependable GSP. The approach is a novel way to exploit the inherent redundancy of NoC-based many-cores and to address tomorrow’s predicted issues with accelerated degradation of ICs as processing geometries continue to shrink.
4 Potential impact and main dissemination activities and exploitation of results

4.1 Potential impact

It is only a matter of time before, for the class of streaming applications, conventional computing architectures will be replaced by reconfigurable multi-core computing platforms. Today, streaming applications are already omnipresent; and societal and technological innovations continue to widen the demand for novel and improved applications. The CRISP demonstrators are representative for both low- and high-end applications. Satellite navigation systems have become omnipresent in e.g. cars, smart phones, and wrist watches, which makes navigation a practical proof of concept application for the low-end consumer market. Digital beamforming is a typical example of a high-end application with challenging throughput and computing demands. Digital beamforming is used in an increasing range of products like base stations for wireless telecommunications, radar systems, sonar systems, and radio astronomy telescope systems. Streaming applications are expected to create a huge drive and momentum for reconfigurable platform chips. The CRISP project anticipates on this expected architectural shift by researching streaming applications, reconfigurable IP cores, interconnect technologies, run-time tools and dependability issues.

European excellence in forthcoming streaming applications

Europe is a major player in the embedded arena today and the holistic approach of CRISP is necessary for European companies to achieve world leading positions in computing solutions and products for streaming applications.

The reconfigurable technology researched and developed within CRISP is promising and directly relevant for the next generation of Smart Antenna and Active Electronic Steering Antenna (AESA) based sensor systems. For such applications, run-time mapping and real-time reconfiguration is considered to be a must in the future. Today’s solutions for high performance streaming application combining big conventional FPGAs and DSPs will not deliver the cost effectiveness – in terms of power and space efficiency – and flexibility needed for tomorrow’s applications. Novel solutions are required; enabling short time-to-market and reduced costs of ownership, for increasingly complex products and systems. Reconfigurable many-core platforms are expected to be key to lowering the costs of ownership and to achieve the software defined antenna functions that will be required for developing the next generation of Smart Antenna and AESA based sensor systems. Tomorrow’s mission critical streaming application must deliver dependable quality-of-service while running on less dependable hardware. The CRISP project anticipates on this development and lets application engineers experiment with techniques that combine on-chip dependability self-diagnostics and dynamic reconfiguration to make applications self-repairing by circumventing hardware faults.

In order to sustain or accomplish world-leadership in mission critical high performance streaming applications, it is essential to master application development on reconfigurable many-core platforms exploiting the flexibility, massive parallelism, and dependability testing.
The Global Navigation Satellite System (GNSS) application demonstrated in the CRISP project is used to showcase the latest research on multi-core GNSS receivers. The developed technology also has a lot of potential in other streaming application areas. It is foreseen that related multi-core applications will emerge in the areas of antenna arrays in cellular base-stations (i.e. Multiple-Input Multiple-Output (MIMO) antenna systems), sonar signal beamforming, multimedia streaming (video and graphics, multi-channel audio), Software Defined Radio implementations of multi-standard cellular transceivers, and multi-modal user interfaces (with e.g. speech and/or motion recognition and speech synthesis).

**Increased market share of inexpensive generic platforms**

CRISP aims at the important domain of streaming applications, which is expected to grow faster than other application domains with the adoption of new standards. The CRISP project delivers a novel reconfigurable multi-core computing platform. These types of platforms are urgently needed, since hardwired logic (i.e. ASIC) is getting prohibitively expensive for an ever wider range of products. Currently, many products make use of fine-grained reconfigurable FPGAs. Their usage, however, results in severe overheads unacceptable for most mass market applications. These overheads are related to device area, power consumption, and unit costs.

In particular, embedded low-power systems increasingly profit from the integration of signal processing capabilities of new energy-efficient multi-core architectures with added flexibility. By combining techniques from the application domain and computer engineering, embedded systems can improve the production of solutions both concerning costs and time:

<table>
<thead>
<tr>
<th>Reduction of costs</th>
<th>Reuse of a single design template will reduce development and verification costs and increase design and software development productivity. Moreover, reconfigurable many-core platform chips can be used for a large variety of applications by reconfiguring the same chip after fabrication;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduction of time to market</td>
<td>The development time for new solutions can be drastically reduced by replacing application-specific platforms with energy-efficient general stream processor platforms using reconfigurable many-core technology.</td>
</tr>
</tbody>
</table>

Moreover, the ability to reconfigure a system and to use this capability to make systems dependable is extending product lifetime. Because users can reconfigure their purchased product to increase service levels, hardware can be used for a longer period of time. Furthermore, built-in dependability techniques allow for graceful degradation. Because of these aspects, products will have a longer life cycle, become cheaper, and be less of a burden to the environment.

Dependable and reconfigurable multi-core systems will serve various application domains ranging from consumer electronics to mission critical space applications. For consumer electronics shorter time-to-market and cost reductions are key differentiators to win the competition. For high-end mission critical applications, for instance in space, extending lifetime under harsh conditions, and easy upgrading are important aspects when designing dependable and reconfigurable systems. Both orthogonal application
domains focus on improving dependability in combination with reconfigurable multi-core hardware, however, motivated by partly different grounds.

Mastering new powerful reconfigurable computing platforms
CRISP motivations and goals remain highly relevant. The recent HiPEAC Vision roadmap identifies these opportunities and recommends to strengthen research and development of reconfigurable architectures and self-adaptive systems. The CRISP project delivers a novel coarse-grained reconfigurable many-core technology to the mass markets. The emerging reconfigurable technology provides excellent opportunities for Europe to accomplish world leadership.

4.2 Main dissemination activities
The CRISP consortium considers dissemination of the project and results to be an important activity, which is imperative for fruitful commercial and scientific exploitation of the project results. CRISP motivations, plans, and results are therefore actively and successfully disseminated throughout the project both to a focused scientific and industrial community and to a wider audience. Besides scientific publications and presentations, the project is thus promoted via the project website, brochures, presentations, publications, exhibition and tutorial activities, press interviews, and press releases. In the following, we list the main concrete dissemination activities. A more complete list of dissemination activities is provided separately.

Project website: www.crisp-project.eu
The CRISP website presents an overview of the project to the public. It contains information on the background, the consortium and the technological issues addressed. The website is regularly updated throughout the project including lists of dissemination events, news and publications. The website is a constantly world-wide open window to CRISP.

Press releases and press interviews
The CRISP consortium sent out three press releases. The first, in connection with the start of the project in 2008. The second, in connection with the first live demonstration of the General Stream Processor at the Design, Automation and Test in Europe (DATE) Exhibition in 2011. And the third in connection with the final review in June 2011. The released press was widely picked-up in the electronic media and covered among others by EETimes, Bits&Chips, Engadget, Elector, TG Daily, Gizmag, and Futura Sciences. The CRISP consortium has also attracted a number of press interviews leading to articles in technology magazines including Automatisering Gids (NL), Bits&Chips (NL), and New Electronics (UK).
Project results brochure
The General Stream Processor (technology) is applicable by many industries for many applications. The technology has the potential to bring many innovations to consumer products and is tangible, due to the developed prototype chips. For these reasons, it is expected that companies, universities, media and individuals will have great interest in the General Stream Processor. The eight page CRISP results brochure gives an overview of the CRISP project, its results and technology and the CRISP consortium. The brochure is handed out at events (exhibitions, conferences, etc.) and provided to individuals (i.e. in response to information requests via the project website). Besides marketing, the brochure contributes to the (commercial) exploitation and utilization of the project results. The brochures has already been distributed for instance in connection with the 2011 Design, Automation and Test in Europe (DATE) Conference and Exhibition in Grenoble, France.

Project exhibition
The CRISP project was presented at the DATE 2009, the SoC 2010, and the DATE 2011 Exhibitions. The Design, Automation and Test in Europe (DATE) Conference and Exhibition is a unique European event bringing together researchers, users and vendors as well as specialists in the design, test and manufacturing of electronic circuits and systems. DATE is a yearly event attracting more than 2000 attendees. The International Symposium on System-on-Chip (SoC) 2010 is an annual event held in Tampere, Finland, attracting around a hundred researchers and industrial specialists. At DATE 2009 in Nice, France, the CRISP motivations, plans, and specifications were promoted using flyers and the CRISP project poster. At SoC 2010, preliminary results were presented using flyers and posters. At DATE 2011 in Grenoble, France, the first live demonstrations of the General Stream Processor hardware and software were given and project results were promoted using the project results brochure and poster. The exhibition activities were successful and confirmed the great interest in the CRISP topics and results.
Project tutorial
In collaboration with the GETA graduate school, the CRISP Consortium organized the SoC 2010 Tutorial on Dependable Hardware and Software for Embedded Multicore Computing Platforms. The tutorial was held in connection with the International Symposium on System-on-Chip in Tampere, Finland. The Graduate School in Electronics, Telecommunications and Automation (GETA) is a post graduate programme offered jointly by five universities: Aalto University School of Science and Technology, Tampere University of Technology and the Universities of Oulu, Turku and Jyväskylä. The full-day tutorial attracted around forty attendees from industry and academia to learn from a collection of international industrial and academic lectures from Recore Systems (NL), Freescale Semiconductor (IL), Altreonic (BE), University of Twente (NL), and Tampere University of Technology (FI).

Project book chapter
The CRISP consortium has written a book chapter giving a comprehensive overview of the whole project. This book chapter introduces CRISP, presents the concepts, and outlines the preliminary results of the project. The chapter appears in Reconfigurable Computing - From FPGAs to Hardware/Software Codesign, Cardoso, João M. P.; Hübner, Michael (Eds.), Springer, 2011, ISBN 978-1-4614-0060-8. This book gives an overview of the latest advancements in European technology for reconfigurable computing.

4.3 Exploitation and use of results
The results and momentum of the CRISP project offer ample basis for further use and exploitation. The CRISP Consortium has gained and disseminated important knowledge about and understanding of tomorrow’s technology for streaming applications, reconfigurable many-cores, and dependability hardware and software. This knowledge will be used directly in the R&D activities of the industrial partners and extended and disseminated by the academic partners through further research and education activities. Furthermore, CRISP delivers tangible results in both hardware and software. Working demonstrators implemented on real hardware provide a tremendous asset in the further commercial, educational, and experimental exploitation of project results. New research projects including STARS\(^5\), NEST\(^6\), and TOETS\(^7\) have already been launched to continue research along the path set out by CRISP and to pick up on CRISP results. The concepts and knowledge from CRISP provides the consortium an excellent position to exploit the anticipated shift to dependable reconfigurable many-core architectures.

---

\(^5\) Sensor Technology Applied in Reconfigurable Systems (STARS); [www.stars-project.nl](http://www.stars-project.nl)

\(^6\) Netherlands Streaming (NEST); [http://www.nest-consortium.nl](http://www.nest-consortium.nl)

\(^7\) Towards One European Test Solution (TOETS); Catrène Project [http://www.catrene.org/](http://www.catrene.org/)
5 Address of project public website and relevant contact details

5.1 Project website

www.crisp-project.eu

5.2 Project consortium overview

http://www.crisp-project.eu

Project coordinator: Dr. Paul M. Heysters
Project manager: Dr. Kim Sunesen
Email: info@crisp-project.eu

Postal address:
Recore Systems BV
P.O. Box 77
7500 AB Enschede
The Netherlands

The Consortium:
Recore Systems (NL)
Reconfigurable computing IP and semiconductors

University of Twente (NL)
Computer Architecture and Embedded Systems group

Atmel (D)
Design and manufacture of advanced semiconductors

Thales (NL)
Communication, sensors, land & naval systems, security

Tampere University of Technology (FI)
DSP and Communications System-on-Chip group

NXP (NL)
Semiconductors, system solutions and software
5.3 **Contact information**

<table>
<thead>
<tr>
<th><strong>Recore Systems</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Paul M. Heysters, Coordinator</td>
<td>Recore Systems BV</td>
</tr>
<tr>
<td>Kim Sunesen, Project Manager</td>
<td>P.O. Box 77</td>
</tr>
<tr>
<td></td>
<td>7500 AB Enschede</td>
</tr>
<tr>
<td></td>
<td>The Netherlands</td>
</tr>
<tr>
<td></td>
<td>tel. +31 53 4753 000</td>
</tr>
<tr>
<td></td>
<td>www: <a href="http://www.recoresystems.com">www.recoresystems.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>University of Twente</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gerard J.M. Smit</td>
<td>Computer Architecture for Embedded Systems</td>
</tr>
<tr>
<td></td>
<td>Dept. of Computer Science</td>
</tr>
<tr>
<td></td>
<td>University of Twente</td>
</tr>
<tr>
<td></td>
<td>P.O. Box 217</td>
</tr>
<tr>
<td></td>
<td>7500 AE Enschede</td>
</tr>
<tr>
<td></td>
<td>The Netherlands</td>
</tr>
<tr>
<td></td>
<td>tel. +31 53 489 3770</td>
</tr>
<tr>
<td></td>
<td>www: caes.ewi.utwente.nl</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Atmel Automotive</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Werner Brugger</td>
<td>Atmel Automotive GmbH</td>
</tr>
<tr>
<td></td>
<td>Lise Meitner Str. 15</td>
</tr>
<tr>
<td></td>
<td>D-89081 Ulm</td>
</tr>
<tr>
<td></td>
<td>Germany</td>
</tr>
<tr>
<td></td>
<td>tel. +49 731 5094 248</td>
</tr>
<tr>
<td></td>
<td>www: <a href="http://www.atmel.com">www.atmel.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Thales Netherlands</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Henk van Zonneveld</td>
<td>Thales Netherlands</td>
</tr>
<tr>
<td></td>
<td>Zuidelijke Havenweg 40</td>
</tr>
<tr>
<td></td>
<td>7554 RR Hengelo</td>
</tr>
<tr>
<td></td>
<td>The Netherlands</td>
</tr>
<tr>
<td></td>
<td>tel. +31 (0)74 248 30 33</td>
</tr>
<tr>
<td></td>
<td>www: <a href="http://www.thalesgroup.com/NLhome">www.thalesgroup.com/NLhome</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Tampere University of Technology</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Jari Nurmi</td>
<td>Tampere University of Technology</td>
</tr>
<tr>
<td></td>
<td>P.O.Box 553</td>
</tr>
<tr>
<td></td>
<td>FI-33101 Tampere</td>
</tr>
<tr>
<td></td>
<td>Finland</td>
</tr>
<tr>
<td></td>
<td>tel. +358 40 506 4460</td>
</tr>
<tr>
<td></td>
<td>www: <a href="http://www.tut.fi">www.tut.fi</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NXP Semiconductors</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bart Vermeulen</td>
<td>NXP Semiconductors</td>
</tr>
<tr>
<td></td>
<td>High Tech Campus 37</td>
</tr>
<tr>
<td></td>
<td>5656 AE Eindhoven</td>
</tr>
<tr>
<td></td>
<td>The Netherlands</td>
</tr>
<tr>
<td></td>
<td>tel. +31 40 27 29804</td>
</tr>
<tr>
<td></td>
<td>www: <a href="http://www.nxp.com">www.nxp.com</a></td>
</tr>
</tbody>
</table>
5.4 Project logo