



**HAL**  
open science

## Flexible and Distributed Real-Time Control on a 4G Telecom MPSoC

Camille Jalier, Didier Lattard, Gilles Sassatelli, Pascal Benoit, Lionel Torres

► **To cite this version:**

Camille Jalier, Didier Lattard, Gilles Sassatelli, Pascal Benoit, Lionel Torres. Flexible and Distributed Real-Time Control on a 4G Telecom MPSoC. ISCAS'10: IEEE International Symposium on Circuits and Systems, Paris, France. pp.3961-3964. lirmm-00498529

**HAL Id: lirmm-00498529**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00498529>**

Submitted on 7 Jul 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Flexible and Distributed Real-Time Control on a 4G Telecom MPSoC

Camille Jalier and Didier Lattard  
CEA, LETI, MINATEC  
F38054 Grenoble, France  
{firstname.lastname}@cea.fr

Gilles Sassatelli, Pascal Benoit and Lionel Torres  
University of Montpellier II, LIRMM  
UMR 5506, France  
{lastname}@lirmm.fr

**Abstract**—Applications like 4G baseband modem require single-chip implementation to meet the integration and power consumption requirements. These applications demand a high computing performance with real-time constraints, low-power consumption and low cost. With the rapid evolution of telecom standards and the increasing demand for multi-standard products, the need for flexible baseband solutions is growing. The concept of Multi-Processor System-on-Chip (MPSoC) is well adapted to enable hardware reuse between products and between multiple wireless standards in the same device.

Based on the experience of two heterogeneous Software Defined Radio (SDR) telecom chipsets, this paper presents a distributed control architecture for the homoGENEous Processor arraY (GENEPY) platform for 4G applications. This MPSoC platform is built with telecom baseband processors interconnected with a Network-on-Chip. The control is performed by a MIPS processor embedded in each baseband processor. This control processor can locally reconfigure and schedule the applications with real-time telecom constraints.

## I. INTRODUCTION

The Third Generation Partnership Project (3GPP) has defined the Long Term Evolution (LTE) for 4G radio access. Emerging fourth generation cellular standards like LTE [1] require intensive modem signal processing. This standard involves high data rate, low latency, and relies on OFDMA/MIMO techniques with adaptive modulation. The baseband architecture requires dynamic reconfigurations due to user resource allocation, a high computational demand and low power requirements under real-time constraints. The demodulation stage of a LTE mobile handset is characterized by a 10 GOPS workload with a budget around 200mW [2]. One major challenge therefore relies on devising an architecture that meets the flexibility, power and area requirements.

Traditional telecom chipset are designed with dedicated hardwired solutions which are cost-ineffective for multi-standard mobile handset. For more flexibility, MPSoCs (Multiprocessor System-on-Chip) [3] with multiple programmable processors as system components have been introduced in the telecom field. MPSoCs are well suited for systems with concurrent algorithms like telecom applications. The implementation of such algorithms on two heterogeneous SDR platforms [4] [5] has proven the efficiency of MPSoCs to provide a valuable solution. In the context of SDR platforms, the flexibility and reconfigurability is the key challenge. Homogeneous MPSoCs, which are based on the replication of

identical units, can better provide flexibility, fault tolerance and scalability.

In this paper, we present the homogeneous GENEPY platform which has distributed control processors for solving programmability/scheduling problems. This innovative approach in the context of hard real-time and high complexity is evaluated in terms of performance and power consumption on a modem LTE application.

The rest of this paper is organized as follows. The homogeneous GENEPY platform with a distributed control is discussed in section II. In section III, we present the reconfiguration and scheduling constraints of a LTE demodulation application as a case study. In section IV, we discuss the performance and power consumption results of this distributed solution.

## II. MPSoC ARCHITECTURE

### A. Related Works

Previous works have proposed architectures for LTE modem implementation. Multi-core architectures like Picochip [6], Infineon's MuSIC [7] or Sandbridge's SB3011 platform [8] argue high computational performance and high flexibility. They are homogeneous or heterogeneous DSP-centered and accelerator-assisted MPSoCs. Those solutions proposes a centralized control, reconfiguration and scheduling is performed by one unit. It is not scalable and complex control strategies (pipeline, task migration) are very hard to implement and reconfiguration is often very slow.

The MAGALI platform [5] is a heterogeneous NoC-based MPSoC for Mobile Terminal in 65nm low-power CMOS. This solution supports OFDMA/MIMO standards with a reduced power consumption. This approach is based on a centralized control processor combined with hardwired local controllers at unit level. The local controller has a limited flexibility to explore advanced scheduling as task mapping, data-dependent reconfiguration, etc.

### B. A Homogeneous Processor Array - GENEPY

The homogeneous GENEPY platform is a MPSoC based on the replication of a telecom baseband processor, called a SMEP unit. This core integrates a Smart Memory Engine (SME) for fast data manipulation and a processing cluster with two DSPs. The SME handles four logical buffers mapped on

a same 32KB local memory (RAM data). The buffers (size, location) are dynamically configurable to fit applicative needs and are managed as circular buffers for data-flow operations. Data manipulation on the four buffers are performed by four attached Read Processes (RP). A Read Process executes microinstructions to read data from the buffer and therefore generates read addresses, writes data to a specific target and handles synchronizations between Read Processes. The write target of RPs can be either the Network Interface (NI) to access other SMEP units, another buffer in the local memory, or one of the two DSPs in the processing cluster. Communication interconnect inside the SMEP can handle 6 parallel 32-bits transfers at 400MHz, i.e. a 77GBits/s bandwidth. The interconnect is configured at each clock cycle based on RPs requests.

In the processing cluster, each DSP reads incoming data from an input FIFO, the intermediate processing values are stored in a local memory and the results are written into an output FIFO. The datapath has been optimized to perform intensive computing on a data-flow with a minimal power budget. Each VLIW DSP can perform 4 parallel 16-bits multiplications at 400MHz, i.e. 3.2 GMAC/s for the processing cluster.

This elementary unit of the processor array is highly programmable. The computing is DSP-based to provide the software flexibility. The SME block with micro-programmable Read Processes enables completely software-defined data manipulation. The unit reconfiguration and scheduling is performed by a local Control Processor. The SMEP architecture, presented in figure 1.

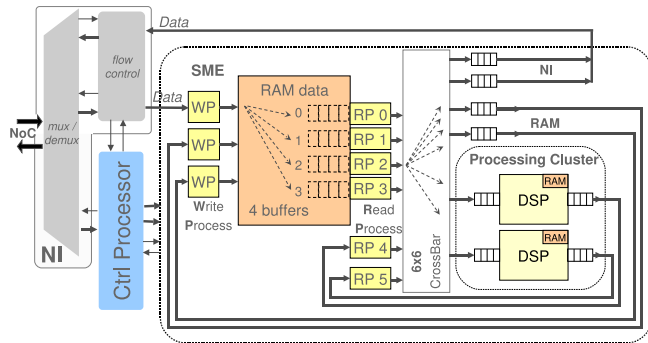


Fig. 1. Elementary unit (SMEP) of the homogeneous processor array

The implementation details on the SME and the DSP are not in the scope of this paper.

We have designed the homogeneous processor array GENEPEY with SMEP units interconnected by an asynchronous Network-on-Chip [9] using Network Interfaces (NI) (figure 2). The GENEPEY platform is characterized by a fully distributed computation and control.

### C. The Distributed Control Processors

In a telecom platform, the scheduling is often quite complex due to dynamic modulation schemes. To support the LTE standard and even future standards, it is valuable to have

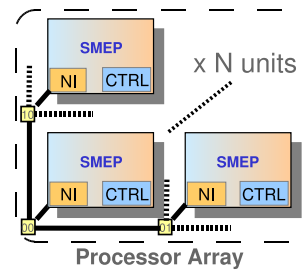


Fig. 2. Homogeneous Processor Array GENEPEY

more flexibility especially in a homogeneous approach to explore techniques such as load balancing, dynamic remapping or fault-tolerance. The Control Processor is a 32-bit MIPS processor which manages dynamic reconfigurations, real-time scheduling, synchronizations. The CPU has several extensions to improve its efficiency(figure 3):

- Input/Output extension to speed-up communication between Control blocks (packetization/depacketization)
- Timer extension to handle real-time constraints.
- Configuration handler to improve reconfiguration speed.

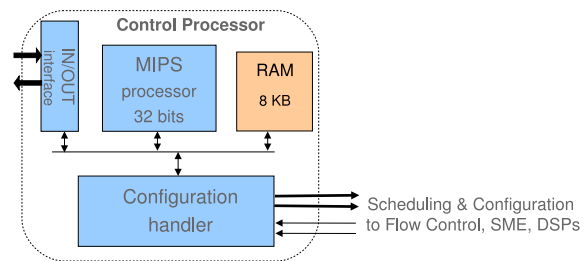


Fig. 3. The MIPS Control Processor

The Control Processor manages the NI, the SME and the processing cluster. The MIPS processor, with a RISC instruction set, executes the scheduling and configuration of the SMEP unit. The MIPS processor can send and receive NoC packets, requests configuration transfers from a configuration server and access RAM and registers of its unit. To minimize to re-configuration overhead, the control processor can reconfigure a block during a processing by using shadow configuration registers. A block can switch from a configuration to another in a single cycle.

Each control processor has to manage locally the SMEP unit but they can exchange information through the NoC to ensure a coherent control over the platform. The MIPS processor offers a flexible solution to support advanced control strategies.

### D. Design Results

The silicon area is extracted after logic synthesis with 65nm low-power CMOS technology at 400MHz. All designs include test mechanisms like scan chains and memory BIST.

As shown in table I, data manipulation with the SME block and data processing with the cluster of 2 DSPs contribute for 85 % of the silicon area. The Control Processor has a

TABLE I  
SYNTHESIS RESULTS - TECHNO ST65NM - 400MHZ

<i>SMEP</i>	area (mm <sup>2</sup> )	
Router (NoC)	0.159	6.6 %
2 DSPs	0.816	34.2 %
SME	1.216	50.8 %
Control Processor	0.122	5.1 %
NI	0.079	3.3 %
<b>SMEP+ NoC</b>	<b>2.392</b>	<b>100 %</b>

relatively small silicon impact, accounts for only 5% of the SMEP area. The use of a flexible MIPS processor is a good flexibility/silicon impact trade-off.

### III. REAL-TIME AND DISTRIBUTED CONTROL ON A LTE APPLICATION

#### A. Reference LTE application

This study focuses on the downlink part of the LTE standard and more precisely on the demodulation side. Using the terminology defined in [10], data are transmitted in 10ms frames equally divided in 10 sub-frames also called TTIs (Time Transmission Intervals), *i.e.* the TTI equals 1 ms. The system is designed to transmit on 4 antennas and to receive on 2 antennas, which requires a high performance processing, because of the implementation of diversity and spatial multiplexing schemes.

Our reference application is composed of 5 tasks (figure 4):

- 2 Channel Estimation Modules, one for each RX antenna based on Wiener filtering.
- 2 interpolation algorithms of the channel coefficients over the whole bandwidth.
- 1 MIMO MMSE decoder that implements a 4x2 double-Alamouti algorithm.

The modulation scheme depends on the user resource allocation. The application defines five operating modes from a low-quality (QPSK), low data-rate transmission to a high quality (64-QAM), high data-rate transmission. The application is mapped on two SMEP units separating Channel estimation and interpolation on SMEP 0 from MIMO decoding on SMEP 1. To process a TTI, each task needs at least dozens of reconfiguration and scheduling phases depending on the operating mode (QPSK to 64-QAM).

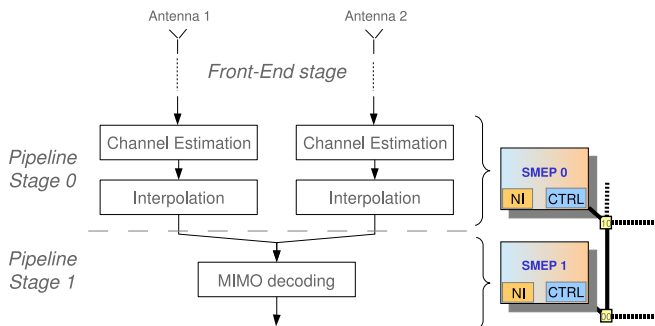


Fig. 4. Mapping of the LTE application

Usually, LTE application are not pipelined to reduced the control complexity. The platform is configured to support only one operating mode at a time and completely reconfigured when a new operating mode is detected. In this work, the application is pipelined to speed-up the execution time. For instance, the SMEP 0 can process Channel Estimation and Interpolation with the operating mode 5 for an incoming TTI, and in parallel the SMEP 1 processes MIMO decoding with the operating mode 1 for the previous TTI.

#### B. Real-Time and Distributed Control

As shown in figure 5, a single task for instance Channel Estimation requires different configurations (configuration ID 0 to 4) and a specific scheduling called a configuration sequence. To process a configuration sequence, the control processor has to send a request to a configuration server to load the right configuration ID in the SMEP unit. Then the control processor can indicate the next configuration ID to the configuration handler. To meet real-time constraints, the Control Processor works in parallel to the application processing. In our example, when the processing blocks executes the configuration 2, the Control Processor prepares the next configuration 3.

The MIMO decoding algorithms depends on information extracted after Channel estimation. So after Channel Estimation, SMEP 0 sends a packet to SMEP 1 with the next operating mode. This packet exchange ensures a coherent pipeline with a dynamic operating mode.

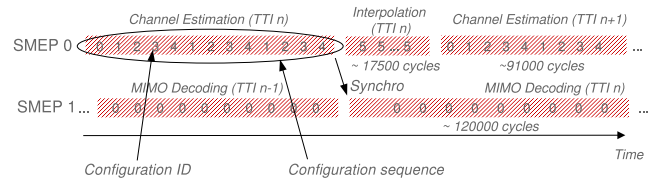


Fig. 5. Pipelined LTE modem with configuration sequences

The MIPS processor handles events by polling a status register. This solution enables a good reactivity of the software compared to interrupt mechanisms. To reduce the power consumption of the polling mechanism, the MIPS processor can disabled its clock by software when its job is finished. Then the clock is automatically enabled in one cycle when a new event is detected. This technique limits the power consumption of the polling mechanism with zero timing penalty. Real-time scheduling is achieved by using fast event detection and no operating system overhead.

To program the management strategy, the programmer can rely on a API and a GCC cross-compiler tool chain.

## IV. PERFORMANCE AND POWER CONSUMPTION RESULTS

#### A. Performance results

Performance results are extracted with a simulation platform. To increase simulation speed, the NoC is modeled in TLM SystemC using post-layout parameters. All SMEP units are modeled at RTL level to provide cycle-accurate results.

TABLE II  
CONTROL PROCESSOR ACTIVITY

	Scheduling	Reconfiguration	Total
Channel Estimation and Interpolation	3 %	2 %	5 %
MIMO Decoding	1%	1%	2 %

The control processor is clocked each time scheduling or reconfiguration is needed. We have measured the control processor activity as shown in table II. The control processor is active less than 6% of the total TTI processing time. The performance is achieved by using hardware I/O interfaces for fast packetization/depacketization and a configuration handler than reconfigure the data-flow very efficiently.

The polling technique on the MIPS processor enable a fast detection of events. Typically, the processor is able to detect a reconfiguration event, then send a request to the configuration server and finally activate the scheduling of that configuration in around 60 clock cycles.

A pipelined application needs a more complex scheduling technique, that implies more MIPS instructions. But the overhead on the processing time is only 0.2% ( $1\mu s$  overhead on  $530\mu s$  TTI processing) . Using a hardware I/O interfaces, the communication between control processor only takes few cycles.

As the scheduling and reconfiguration is processed in parallel to the telecom application, the performance cost of a flexible control is negligible on LTE applications.

### B. Power consumption results

To evaluate the power consumption, the platform has been placed and routed in 65 nm low-power CMOS technology. We have simulated a complete TTI processing with the placed and routed netlist. Table III presents the average power consumption at gate-level of the two SMEP units executing the application. The power consumption of the Control Processor is very low, this is mainly due to a reduced activity over the TTI processing (< 5% activity). The clock signal of the control processing is gated during non-active period to reduce the power consumption. For an unpipelined application the power consumption of the Control Processor is 2.7mW compared to 3.8mW for a pipelined application.

TABLE III  
DETAILED POWER CONSUMPTION OF GENEPY TO PROCESS A TTI WITH  
A HIGH DATA-RATE AND HIGH QUALITY OPERATING MODE

Data Processing	113.7 mW	59 %
Data Manipulation	70.8 mW	36.7%
Control	3.8 mW	2.0 %
Network Management	4.4 mW	2.3 %
<b>Total</b>	<b>192.7 mW</b>	<b>100 %</b>

Figure 6 shows a detailed view of the power consumption profile during a TTI processing, with separate contributions for processing (DSP), data reordering (SME) and control (MIPS) for SMEP0.

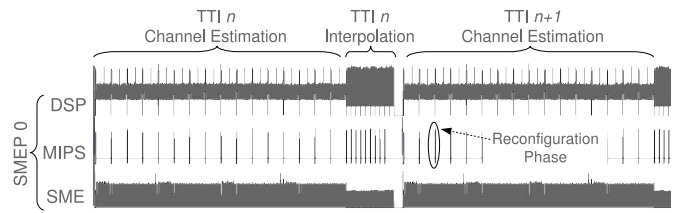


Fig. 6. Power Consumption Profile of a SMEP unit

## V. CONCLUSION

We have presented the GENEPY platform, a low-power homogeneous MPSoC for 4G Mobile Terminals. The major component is the SMEP baseband processor, able to provide data manipulation at 77 GBits/s and computing at 3.2 GMAC/s at a 400MHz operating frequency. Due to separate data handler and data processing blocks, this architecture is efficient and configurable. The control over the platform is fully distributed on MIPS processors: this solution is highly flexible and scalable with a moderate area overhead of 5.1% of the platform silicon area. For a pipelined LTE application, the Control Processor is only active 5% of the processing time with a power consumption of 3.8 mW (2 % of the global power consumption). This flexible and distributed control architecture for a homogeneous platform enables to execute a wide range of control strategies with a minimal Performance/Energy impact.

Our future research efforts comprise the enhancement of the SMEP unit towards a better power management. As the MIPS processor is used only at 5% of its capacity, we will explore its use to support distributed power management algorithms, task migration, fault tolerance, load balancing, etc.

## REFERENCES

- [1] J. Berkmann, C. Carbonelli, F. Dietrich, C. Drewes and W. Xu, "On 3g LTE Terminal Implementation - Standard, Algorithms, Complexities and Challenges," *Wireless Communications and Mobile Computing Conference, IWCMC '08*, pp. 970-975, Aug. 2008
- [2] C.H. van Berkel, "Multi-Core for Mobile Phones," *Design, Automation & Test in Europe Conference (DATE '09)*, pp. 1260-1265, Apr. 2009
- [3] W. Wolf, A.A. Jerraya and G. Martin, "Multiprocessor System-on-Chip (MPSoC) Technology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 10, pp. 1701-1713, Oct. 2008
- [4] D. Lattard *et al.*, "A Reconfigurable Baseband Platform Based on an Asynchronous Network-on-Chip," *Journal of Solid-State Circuits (JSSC)*, vol. 43, pp. 223-235, Jan. 2008
- [5] F. Clermidy, R. Lemaire, X. Popon, D. Kténas and Y. Thonnart, "An Open and Reconfigurable Platform for 4G Telecommunication: Concepts and Application," *Euromicro Conference on Digital System Design (DSD)*, Aug. 2009
- [6] D. Pulley, "Multi-core DSP for Base stations: Large and Small," *Design Automation Conference (ASPDAC'08)*, pp. 389-391, March 2008
- [7] U. Ramacher, "Software-defined radio prospects for multistandard mobile phones," *Computer*, vol. 40, no. 10, pp. 62-69, 2007
- [8] J. Glossner *et al.*, "The sandbridge sb3011 platform," *EURASIP J. Embedded Syst.*, vol. 2007, no. 1, pp. 16-16, 2007
- [9] Y. Thonnart, E. Beigné and P. Vivet, "Design and Implementation of a GALS Adapter for ANoC based Architectures," *IEEE symp. on Asynchronous Circuits and Systems (ASYNC '09)*, pp. 13-22, May 2009
- [10] 3GPP TSG-RAN, 3GPP TS36.211, *Physical Channels and Modulation (Release 8)*, v8.1.0 (2007-11)