Adaptivity and Reconfigurability in Wireless Communications

Gerard K. Rauwerda, Gerard J.M. Smit
University of Twente
faculty of Electrical Engineering, Mathematics & Computer Science
P.O. Box 217 – 7500 AE Enschede, the Netherlands
g.k.rauwerda,g.j.m.smit@utwente.nl

Abstract— **A key issue of future wireless communication systems is that they have to be adaptive. In the** *Adaptive Wireless Networking (AWGN)* **project we aim at the implementation of adaptive wireless communication systems in a heterogeneous reconfigurable System-on-a-Chip (HRSoC). We introduce our methodologies for analyzing and mapping DSP functionality in dynamically reconfigurable heterogeneous hardware. A possible implementation of a multi-mode communication system in the MONTIUM architecture is discussed. Suggestions for future activities in the** *Adaptive Wireless Networking* **project are also given.**

Keywords— **Reconfigurable heterogeneous hardware, Wireless communication algorithms, Software defined radio, SoC, MONTIUM**

I. INTRODUCTION

In the *Adaptive Wireless Networking (AWGN)* project we aim at the implementation of adaptive wireless communication systems in a heterogeneous reconfigurable System-on-a-Chip (HRSoC). One of the tasks in the project is to map the algorithms found in these communication systems on a platform of heterogeneous dynamically reconfigurable architectures. The platform is heterogeneous in this sense that processing is performed in general purpose processors (GPPs), bit-level reconfigurable hardware (i.e. FPGA) or in word-level reconfigurable hardware (i.e. MONTIUM). It is expected that performance and power gains will be achieved by applying dynamically reconfigurable heterogeneous architectures [1].

A key issue of future wireless communication systems is that they have to be adaptive. These systems have to adapt to changing environmental conditions (e.g. more or less users in a cell or varying noise figures due to reflections or user movements) as well as to changing user demands (bandwidth, traffic patterns and QoS). Furthermore, these architectures have to be extremely efficient as these are used in battery-operated terminals and cost effective as they are used in consumer products. When the system can adapt – at run-time – to the environment, significant savings in computational costs can be obtained [2]. Although energy-efficiency is a major issue in terminals as they draw their energy from small batteries, energy consumption is also an issue in base stations from a technical (costly cooling of chips and power supplies) point of view.

Different means of adaptivity can be applied to a wireless communication system. In systems one can distinguish between short- and long-term configurations. In a multi-standard communication system, for instance, one can switch between multiple communication standards (long-term), additionally these communication standards can adapt to the dynamic environment or dynamic user demands (short-term). One should distinguish between long-term reconfigurability and short-term reconfigurability, which refers to the period in which a configuration does not change.

In this paper we will give an introduction to one of the activities in the *Adaptive Wireless Networking* project. Reseach related to our work will be briefly described in section II. The target architecture of future mobile communication systems will be presented in section III. In section IV we introduce the different methodologies of implementation and analysis, while mapping DSP algorithms onto the target architecture. In section V some basics about the HiperLAN/2 and Bluetooth communication system will be discussed. These systems are currently under investigation in our project. The obtained results of a first experiment while implementing these communication systems in the target architecture, are shown in section VI. Some conclusions will be presented in section VII and recommendations for future work will be presented in section VIII.

II. RELATED WORK

So far most algorithmic level research on reconfigurability in UMTS, as for example in the *MuMoR* [3] and *Fitness* [4] projects, has focussed on multi-mode reconfigurability to enable Software Defined Radios (SDRs) supporting multiple communication system standards. The *EASY* project [5] aims at developing a power/cost efficient System-on-Chip (SoC) implementation of the HiperLAN/2 standard. In the *Adaptive Wireless Networking*

(AWGN) project, however, we would like to use reconfigurability to allow the communications system to adapt to changing environmental conditions. Therefore we would like to study how techniques that are used in UMTS, such as variable spreading factors, RAKE-receiving, adaptive modulation and coding, MISO and MIMO, can be made adaptive to environmental conditions and how these techniques can make use of the reconfigurability of a heterogeneous reconfigurable architecture.

Recently, there have been several announcements of heterogeneous reconfigurable architectures on a single chip [6], [7]. For example, Xilinx delivers a combination of Virtex II FPGA technology and one or more PowerPC 405 – on a single chip –, the Virtex-II Pro. However, conventional reconfigurable processors are bit-level reconfigurable and are far from energy-efficient. At the University of Twente in the *Chameleon* project [8] the first steps have been made to define an energy-efficient heterogeneous reconfigurable System-on-Chip architecture. *Pleiades* at the University of California, Berkeley, is exploring reconfiguration of coarser-grain application-specific building blocks with an emphasis on low-power computations [9]. Chameleon Systems created one of the first practical commercial implementation of coarse-grain reconfigurable technology for data intensive Internet, DSP and other high performance telecommunication applications [10]. Furthermore, Quicksilver's *adaptive computing machine (ACM)* technology is intended for low-power mobile devices [11], whereas PACT [12] proposes an *extreme processor platform (XPP)* based on clusters of coarse-grained processing array elements.

III. Reconfigurable heterogeneous architecture

In the *Chameleon* project [8] at the University of Twente a dynamically reconfigurable heterogeneous System-on-a-Chip (SoC) is being defined. The SoC contains a general purpose processor, a fine-grain reconfigurable part (consisting of FPGA tiles) and a coarse-grain reconfigurable part (see Fig. 1). The latter comprises several Montium processor tiles. The algorithm domain of the Montium comprises 16-bit digital signal processing (DSP) algorithms that contain multiply accumulate (MAC) operations. A Montium-tile is designed to execute highly regular computational intensive DSP kernels [13]. The irregular parts of the algorithms run on the general purpose processor.

Using heterogeneous hardware makes it possible to implement an algorithm at the right granularity: algorithms that operate on bit-level data can be mapped on bit-level reconfigurable hardware and algorithms that operate on word-level data can be mapped on word-level reconfigurable hardware. This way algorithms can be implemented in a relatively energy-efficient manner while still retaining performance and reconfigurability. The SoC yields a combination of performance, flexibility and energy-efficiency.

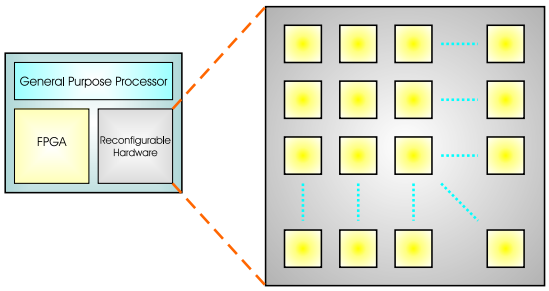

Fig. 1
Chameleon heterogeneous SoC architecture.

IV. Methodology

In order to map the DSP functions onto the heterogeneous reconfigurable architecture, we use two different methodologies:

- Functional simulations
- Behavioural simulations

A. Functional simulation

For functional simulations we use a co-simulation environment with Matlab and ModelSim, as depicted in Fig. 2. Using this environment, one can simulate the (baseband) functionality of a complete communication system in software (i.e. using Matlab) and partly in hardware (i.e. using ModelSim).

In Matlab code all functions of the system under simulation are defined. The software part of the simulator is furthermore responsible for the control mechanisms in the system under simulation. VHDL code describes the functionality implemented in hardware. ModelSim uses this code to perform simulations of the functionality in hardware. The software and hardware counterparts cooperate in one simulation with each other via file transfers.

B. Behavioural simulation

Behavioural simulations are useful in order to analyse the feasibility of implementing the (baseband) functionality of a communication system in a heterogeneous reconfigurable system-on-a-chip (HRSoC). In contrast to functional simulations, in behavioural simulations only the behaviour of the functionality is described on an abstract level. Fig. 3 depicts the idea of our reconfigurable archi-

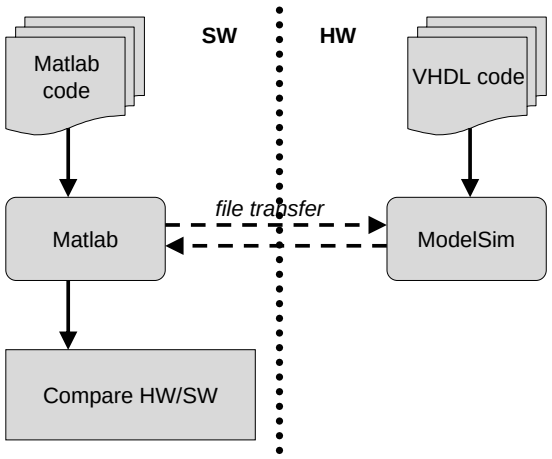

Fig. 2
THE CO-SIMULATION ENVIRONMENT.

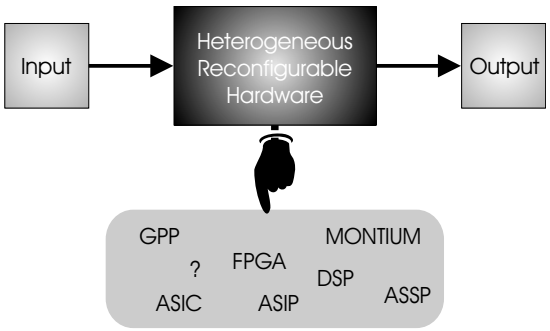

Fig. 3
THE RECONFIGURABLE ARCHITECTURE SIMULATOR.

tecture simulator, which is implemented using OMNeT++ [14] and performs behavioural simulations.

The analysis of our mapping problem has to be performed on a high-level, hence a high-level behavioural model will satisfy our approach. A heterogeneous reconfigurable platform has to be supported, wherein the behaviour of the different kinds of reconfigurable hardware/processors can be specified.
The characteristics are listed below:

- Reconfigurable heterogeneous architecture
  - multiple processor types
    [GPP, FPGA, MONTIUM, DSP]
- Time constraints
  - processing delay [clock-cycles]
  - clock frequency [MHz]
- Configuration constraints
  - full configuration delay [clock-cycles]
  - partial configuration delay [clock-cycles]
- Power constraints
  - energy consumption [nJ]
- Communication constraints
  - processed data [bits]
  - configuration data [bits]

Basically, the heterogeneous reconfigurable architecture is modelled as a 'black box', with some input and some output. Inside this 'black box' the heterogeneous reconfigurable architecture is defined. Our approach is that the contents of the 'black box' can be anything, ranging from GPPs to ASICs or a combination of both reconfigurable and dedicated hardware. These hardware parts are connected to each other by some kind of communication network. The input of the reconfigurable architecture is controlled by an *input manager* and the output is analyzed/controlled by an *output manager*.

The heterogeneous reconfigurable architecture consists of multiple processing entities of different kinds. The number of processing entities, in the simulator called processors, has to be known by the simulator. Furthermore the heterogeneous architecture, which can be a System-on-Chip, has a task list defined. This list, which can be updated dynamically, contains all tasks that the heterogeneous reconfigurable architecture has to perform in consecutive order.

V. APPLICATION

We present the initial results of mapping a set of characteristic algorithms on our dynamically reconfigurable heterogeneous platform. Currently, the physical layers of two communication systems are subject of our research: a rather complex wireless LAN system, the HiperLAN/2 standard (similar to IEEE 802.11a), and a less complex communication system, the Bluetooth standard.

These two communication standards have been selected, because they are already part of ongoing research at the University of Twente [15] and they are different enough to give an indication whether our approach is feasible or not. As a first experiment we have mapped the baseband processing part of the HiperLAN/2 receiver onto MONTIUM-tiles, next we have mapped the baseband functionality of the Bluetooth receiver onto the MONTIUM architecture [16].

A. HiperLAN/2 receiver

The task of the physical layer in HiperLAN/2 is to modulate bits that originate from the data link control layer on the transmitter side and to demodulate them on the receiver side. The transmission format of the physical layer is a burst, which consists of a preamble and a data part.

Orthogonal frequency division multiplexing (OFDM),

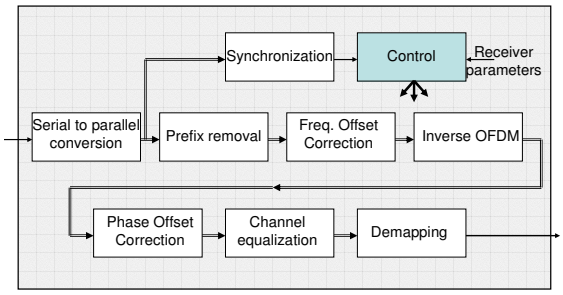

Fig. 4
DEMODULATOR PART OF THE HIPERLAN/2 RECEIVER

which is a special kind of multicarrier modulation, has been used in HiperLAN/2. The modulation technique divides the high data rate information in several parallel bit streams and each of these bit streams modulates a separate subcarrier. 52 subcarriers are being transmitted in parallel per radio channel, which occupies a bandwidth of 20 MHz. However, 4 of these subcarriers are used to transmit pilot tones.

One OFDM symbol is represented by 80 complex input samples (in the time domain), so once per 80 samples the receiver has enough information to demodulate the received samples and output the resulting bits. The sample rate of the input signal is 20 MHz, so the duration of an OFDM symbol is 4 μs.

The receiver not only has to convert the received signal to data bits by performing the inverse of the transmitter, but it also has to try to inverse distortions caused by the radio channel, which is performed in the frequency offset correction, phase offset correction and channel equalization functions.

Fig. 4 depicts the baseband functionality of the HiperLAN/2 receiver. We will describe the most important functions (most computational intensive) that are being subjected to further research, in order to map the functionality on the MONTIUM architecture. We only consider the receiver part of the HiperLAN/2 physical layer, although the transmitter part can be described in a similar way.

In the HiperLAN/2 receiver, information in the time domain has to be translated to the frequency domain. In order to synchronize, prefixes and preambles are detected with matched filters. After synchronization, the inverse OFDM function, which takes care for the time-to-frequency-domain translation, performs a fast Fourier transform (FFT), applied on 64 complex samples. Finally, the hard output bits are determined by comparing the soft output values with values in a lookup table. The size of the lookup table depends on the used modulation scheme (BPSK, QPSK, 16-QAM or 64-QAM).

B. Bluetooth receiver

The task of the physical layer in Bluetooth is to modulate bits that origin from the data layer on the transmitter side and to demodulate them on the receiver side, and vice versa. The Bluetooth system uses packet-based transmission: the information stream is fragmented into packets. In each slot, only a single packet can be sent. All packets have the same format, starting with an access code, followed by a packet header, and ending with the user payload. Single slot, 3-slot and 5-slot packets have been defined.

The channel is a hopping channel with a nominal hop dwell time of 625 μs that corresponds to a single slot. To simplify the implementation, full-duplex communication is achieved by applying time-division duplexing (TDD).

In the Industrial, Scientific and Medical (ISM) band, the signal bandwidth of the Bluetooth system is limited to 1 MHz. For robustness, a binary modulation scheme was chosen. With the mentioned bandwidth restriction, the data rates are limited to about 1 Mbps. Bluetooth uses Gaussian-shaped frequency shift keying (GFSK) modulation with a nominal modulation index of $k = 0.32$. Logical ones are sent as positive frequency deviations, logical zeros as negative frequency deviations.

In the receiver part, the transmitted radio signal is converted back into a binary NRZ signal. The radio signal, which is received, conveys all its information in the frequency deviation of the signal.

The Bluetooth receiver contains an FM-discriminator, a Low Pass Filter (LPF) and a threshold detector. In the FM-discriminator FM-to-AM conversion is performed by multiplying the received signal with its delayed version. After FM-to-AM conversion, the signal is passed through the LPF, which is implemented as a FIR filter, in order to block all high frequencies that occur due to multiplication. Finally, the consecutive bits are detected by a threshold detector.

VI. RESULTS

As a first experiment we have mapped the baseband processing part of the HiperLAN/2 receiver onto MONTIUM-tiles.

In order to perform all receiver processing (excluding the synchronization and control part), 2 or 3 MONTIUM-tiles are required (depending on modulation type), when

we assume the tiles to run at a clock frequency of 100 MHz. We estimated the processing delay in the receiver of one OFDM symbol to be about 11 μs when 64-QAM modulation is applied. Furthermore, the processing delay using BPSK modulation will be about 8 μs. Hence, we have an adaptable HiperLAN/2 receiver, which can be implemented in 2 or 3 MONTIUM-tiles depending on modulation scenario. In this way the baseband processing of one OFDM symbol is pipelined over multiple MONTIUM-tiles, since every 4 μs a new OFDM symbol has to be processed. The actual modulation type used depends on the channel conditions. This initial experiment shows that adapting to the channel conditions is useful for saving baseband processing.

The functionality of the Bluetooth receiver appeared to have a fairly simple signal processing part, which can be implemented in the MONTIUM architecture quite well. The computation delays of the different receiver parts seem to be a few clock cycles. The processing while receiving one bit takes about 130 ns, when a FIR filter with 50 coefficients is applied and the clock frequency of the MONTIUM-tile is 100 MHz.

This experiment showed the flexibility of the MONTIUM. An HiperLAN/2 receiver was implemented, which can adapt to the used modulation scheme, as well as a flexible Bluetooth receiver on the same architecture. Long-term as well as short-term reconfiguration of the MONTIUM has been demonstrated. The implementation of both Bluetooth and HiperLAN/2 baseband processing in the MONTIUM architecture shows the ability of the MONTIUM architecture for long-term configuration. Furthermore, the HiperLAN/2 baseband receiver showed the ability of short-term reconfiguration, while the configuration depends on the used modulation type.

VII. CONCLUSIONS

We presented one of the activities in the *Adaptive Wireless Networking* project. The focus of the activity will be on mapping DSP algorithms on a heterogeneous reconfigurable architecture. Two different methodologies are used to analyse the mapping of DSP functions onto the heterogeneous reconfigurable architecture. Functional simulations are used for simulating the (baseband) functionality of a complete communication system in software and partly in hardware. Behavioural simulations are useful in order to analyse the feasibility of implementing the (baseband) functionality of a communication system in a heterogeneous reconfigurable system-on-a-chip (HRSoC).

For our first experiment we use the first methodology. Because we can use the same (MONTIUM) architecture for a HiperLAN/2 receiver as well as a Bluetooth receiver, the experiment showed that the MONTIUM architecture is quite flexible.

The amount of tiles that is needed depends on the modulation type (BPSK, QPSK, 16-QAM or 64-QAM). The actual modulation type used depends on the channel conditions. For BPSK and QPSK 2 MONTIUM-tiles are needed and for more complex modulation types 3 tiles are needed. The results of this initial experiment show that adapting to the channel conditions (i.e. short-term reconfigurability) is useful for saving baseband processing.

Secondly, the functionality of the Bluetooth receiver appeared to have a fairly simple signal processing part, which can be implemented in one MONTIUM-tile quite well. A multi-standard communication systems has been created by reconfiguration of the same architecture (i.e. long-term reconfigurability) to either HiperLAN/2 or Bluetooth.

VIII. FUTURE WORK

In further research, adaptive algorithms for WCDMA communication systems will be considered. An interesting case will be systems that adapt to changing environmental conditions as well as to changing user demands (QoS). A control system that is able to adapt the WCDMA receiver in order to minimize the energy consumption, while satisfying the quality constraints at run-time is presented in [2]. These adaptations should be done at run-time to deal with the continuously changing external environment.

Power consumption is another important issue, since energy-efficiency is a major issue in mobile terminals. At the moment, the energy consumption of the MONTIUM architecture is estimated at 50 mW per tile in .12 technology, running at 100 MHz, with a size of 2.7 mm^2 per tile. Comparison of the algorithm mappings as compared to other types of architectures should be done in terms of performance and power consumption.

Finally, scheduling of tasks on the reconfigurable heterogeneous architecture can be implemented in different ways: Multiple tiles can perform different functions instantaneously, while the data in the tile is changing dynamically or the data stays in one tile, while the tile is reconfigured dynamically. Simulations have to show the advantages and disadvantages of these scheduling strategies, therefore the behavioural simulator is developed.

The two simulation methodologies – functional and behavioural–, used for implementation of HiperLAN/2 and Bluetooth, can be used for implementation and analysis while mapping WCDMA DSP algorithms onto the heterogeneous reconfigurable System-on-a-Chip.

ACKNOWLEDGEMENT

This research is supported by the Freeband Knowledge Impulse programme, a joint initiative of the Dutch Ministry of Economic Affairs, knowledge institutions and industry.

REFERENCES

[1] Jan Rabaey. Silicon Architectures for Wireless Systems. Tutorial, Hotchips 2001, August 2001.

[2] Lodewijk T. Smit, Gerard J.M. Smit, and Johann L. Hurink. Energy-efficient Wireless Communication for Mobile Multimedia Terminals. In *Proceedings of The International Conference On Advances in Mobile Multimedia*, pages 115 – 124, Jakarta, Indonesia, September 2003.

[3] MuMoR project. `http://www.mumor.org`.

[4] FITNESS project. `http://www.ist-fitness.org`.

[5] EASY project. `http://easy.intranet.gr`.

[6] Xilinx. Virtex-II Pro Platform FPGAs. `http://www.xilinx.com/virtex2pro/`.

[7] Altera. Excalibur Embedded Processor Solutions. `http://www.altera.com/products/devices/excalibur/exc-index.html`.

[8] Chameleon project – Reconfigurable computing in handheld multimedia computers. `http://chameleon.ctit.utwente.nl`.

[9] Pleiades project - Ultra-Low-Power Reconfigurable Computing. `http://bwrc.eecs.berkeley.edu/Research/Configurable_Architectures/`.

[10] Chameleon Systems. Reconfigurable Communications Processor. `http://www.chameleonsystems.com`.

[11] QuickSilver Technologies. `http://www.quicksilvertech.com`.

[12] PACT XPP Technologies . `http://www.pactcorp.com`.

[13] Paul M. Heysters, G. J. M. Smit, and E. Molenkamp. A Flexible and Energy-Efficient Coarse-Grained Reconfigurable Architecture for Mobile Systems. *Journal of Supercomputing*, 2003. *to appear*.

[14] Andras Varga. OMNeT++ – Object-Oriented Discrete Event Simulator. `http://www.omnetpp.org`, 2003.

[15] SDR project - A Bluetooth-HiperLAN/2 SDR receiver. `http://www.sas.el.utwente.nl/home/SDR/`.

[16] G. K. Rauwerda, G. J. M. Smit, L. F. W. van Hoesel, and P. M. Heysters. Mapping Wireless Communication Algorithms to a Reconfigurable Architecture. In *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'03)*, pages 242 – 251, Las Vegas, Nevada, USA, June 2003.