

MODELING ADAPTIVE CODED MODULATION IN REAL TIME PARTIALLY RECONFIGURABLE MOBILE TERMINALS

*L. Conde-Canencia, Y. Eustache**

*J-C Prévotet, Y. Oliva**

Université Européenne de Bretagne - UBS
Lab-STICC CNRS UMR 319 - BP 92116
F-56321 Lorient Cedex - France

Laboratoire IETR-INSA, UMR CNRS 6164
20, Avenue des Buttes de Cosmes, CS 70839
F35708 Rennes Cedex 7 - France

ABSTRACT

This paper considers the design of partially reconfigurable WiMAX 802.16m mobile terminals that use Adaptive Coding and Modulation techniques to enhance the Quality Of Service. Non-binary LDPC codes (with three different coding rates) and three different modulation schemes are considered. As this kind of design requires a huge amount of time and resources to lead to an operational platform, our approach is based on a methodology that describes the complete platform at high level of abstraction. This approach considerably reduces the design time, as it helps the designer in making good choices prior to the final implementation.

Index Terms— Adaptive Coded Modulation, WiMAX 802.16m, real time, partial and dynamic reconfiguration, FPGA, SoC

1. INTRODUCTION

Nowadays, the need for spectrally efficient transmission on mobile and wireless channels is prevalent because our society demands reliable and fast wireless communications and the radio spectrum grows more and more crowded as a consequence of this. High-order modulation schemes and powerful error-correcting codes are then associated to transmit more bits per Hz bandwidth and maximize performance. However, as the radio signal often propagates in a hostile varying environment, the transmission scheme should be able to adapt itself to the state of the channel in real-time. A promising scheme is then Adaptive Coded Modulation (ACM), where the transmitter switches between signal constellations (of varying size) and code rates at discrete time instants. In other words, at a given time, the transmitter chooses symbols from the biggest constellation meeting the Frame Error Rate (FER) requirements and thus ensures maximum spectral efficiency for the given acceptable FER.

From the implementation point of view, ACM techniques claim for reconfigurable architectures able to dynamically and partially configurate. In this context, this work considers

reconfigurable FPGA devices, which have become complex enough to compete with the most efficient circuits (such as ASICs) in various fields of applications (control, signal and image processing, communications, etc). System on Programmable Chips (SoPC) constitute the natural extension of FPGAs and are composed of one microprocessor (at least) and several high-performance hardware accelerators or IPs destined to perform complex operations in real-time. The complexity of such devices has become so important that it often becomes necessary to implement a kernel to manage the software functions running on the processor(s) as well as the hardware accelerators and the reconfigurable area within the SoPC.

In this context, this paper proposes to use the OverSoC methodology [1] to accelerate the design of the ACM mobile terminal and implement it on a reconfigurable SoC, managed by a real-time kernel. OverSoC provides a model at high level of abstraction and allows to efficiently simulate the complex system. The model describes the entire platform according to specific attributes. We consider a Wireless Metropolitan Area Network (MAN) terminal adapted to the WiMAX 802.16m standard specifications. The terminal performs image processing tasks as well as demodulation and channel decoding. This kind of system could be embedded in a wireless security camera or considered for video conference applications.

Regarding channel (or error-correcting) coding, this work considers Non Binary (NB) Low-Density Parity-Check (LDPC) codes [2], which outperform convolutional turbo-codes (CTC) and binary LDPC because they retain the benefits of steep waterfall region for short codewords (typical of CTC) and low error flow (typical of binary LDPC). A NB-LDPC code is defined by an ultra-sparse matrix, characterised by specific parameters such as the frame length (N) and the code rate (R). In the ACM context, the system switches from one code rate to another depending on the channel state. Note that an advantage of NB codes is that, since they are inherently built on high-order fields, it is possible to identify a closer connection between NB-LDPC and high-order modulation schemes: channel decoding is

*Thanks to GDR-ISIS (France) for funding.

performed at symbol-level, which automatically improves performance compared to binary codes [3] [4]. Moreover, a better performance of the q -ary receiver processing has been observed in MIMO systems [5], [6] and [7].

The paper is organised as follows: Section 2 describes the principles of the OverSoC methodology in terms of platform modeling and description. Section 3 details the design and implementation of the mobile terminal considering the WiMAX 802.16m requirements, based on the methodology. The obtained results are presented in Section 4. Finally, Section 5 draws conclusions and identifies future work.

2. PLATFORM MODELING AND DESCRIPTION

The platform presented in this paper was modeled according to the OverSoC methodology [1], which relies on an iterative process. The designer starts specifying the entire platform model (application + kernel + hardware execution platform) by using specific attributes and constraints (such as timing, logic resources, scheduling ...). These are specified using a dedicated graphical tool that then generates a SystemC code that can be executed to simulate the model. The simulation outcomes several metrics to check if all the constraints were met and if there was no error in the functional execution. If there were functional errors or non met constraints, a log file indicates the nature of the problem (for example, insufficient number of logic resources to execute the hardware accelerators or application tasks that cannot be scheduled with the retained scheduling policy). The designer then identifies the attributes responsible for the malfunctioning and modifies them. Finally, the previous steps are performed in an iterative way until a satisfactory solution is found.

The platform description is the major point in the exploration process. The designer has to specify the entire model which is composed of three distinct parts: the application, the kernel and the architecture, which are described as follows:

Application Model. The application is described as a task graph. At the first level of description, a task is specified by its name, nature (hardware, software or heterogeneous), execution time on the target and two priority lists with its successors and predecessors in the task graph. An heterogeneous task can be implemented in both hardware and software. If the task is executed in a reconfigurable area two additional attributes are specified: the number of the required logic resources and the configuration time.

Kernel Model. The role of the kernel is to manage and schedule the different tasks of the application. To be specific, three main services are considered and modeled:

1. The **task management** service to create and destroy tasks dynamically.

MCS	Modulation	Code rate
1, 2, 3, 4	QPSK	1/2, 2/3, 3/4, 5/6
5, 6, 7, 8	16-QAM	1/2, 2/3, 3/4, 5/6
9, 10, 11, 12	64-QAM	1/2, 2/3, 3/4, 5/6

Table 1. Modulation and Coding Schemes (MCS) in the WiMAX 802.16m standard

2. The **scheduling service** that sequences the tasks in real time following the specific policy provided by the designer.
3. The **decision service** to decide whether an heterogeneous task has to be implemented in software or hardware depending on specific constraints compatible with the real-time processing.

Architecture Model The components of the platform are described at high level of abstraction. For the processor, only the frequency of operation is necessary to assess the duration of a processing cycle. At this level, the reconfigurable area is seen as a set of logic available resources.

3. MODELING THE WIMAX 802.16M MOBILE TERMINAL

3.1. ACM techniques in WiMAX 802.16m

ACM techniques were introduced in Mobile WiMAX to enhance coverage and capacity in mobile applications. The principle of ACM techniques is to choose the Modulation and Coding Scheme (MCS) that maximises the system throughput while guaranteeing an acceptable error rate. The WiMAX 802.16m standard supports 12 MCSs, each one characterized by a modulation (of order M) and a coding rate (R). These MCSs are detailed in Table 1, which is to be read as: MCS1 corresponds to a QPSK modulation ($M = 4$) and $R = 1/2$ or MCS7 corresponds to a 16-QAM ($M = 16$) and $R = 3/4$.

The spectral efficiency (η) of a digital communication system is defined as the maximum throughput (in bit/s) divided by the bandwidth (in Hz). It is calculated as $\eta = R \times \log_2 M$. For example, for MCS1 $\eta = 1/2$ bits/s/Hz and for MCS7 $\eta = 3$ bits/s/Hz. Note that maximizing η results in maximizing the system throughput. In [8], the authors present simulation results of the NB-LDPC receiver obtained with a dedicated System Level Simulator (SLS) that follows the specifications of the WiMAX 802.16m standard [9] [10]. Figure 1 shows the spectral efficiency achieved using ACM (i.e. the MCS that provides the highest throughput subject to a maximum FER is selected) as a function of the Signal-to-Noise ratio (SNR) of the channel. Two different channels are considered in Figure 1: the Additive White Gaussian Noise (AWGN) channel and the ITU Pedestrian B (at 3 km/h) using

a bandwidth equal to 20 MHz. Note that the curves in Figure 1 contain the information for both choices. In this figure, DAVINCI stands for the NB-LDPC decoder (i.e. the one considered in our work) and CTC stands for Convolutional Turbo Codes (considered for comparison purposes).

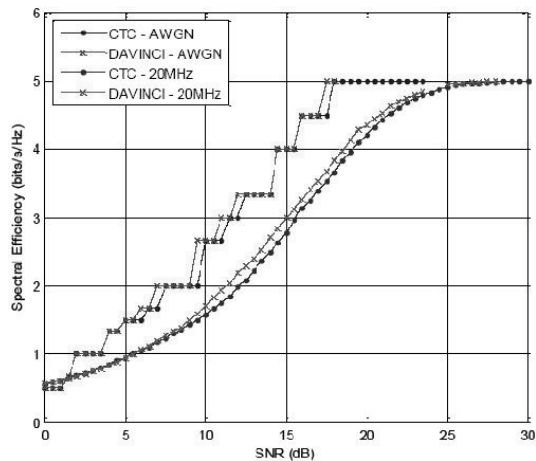


Fig. 1. Spectral efficiency as a function of the SNR with ACM in the AWGN and ITU-R Pedestrian channels (20 MHz)

3.2. Implementation aspects

For a real scenario, we consider a mobile wireless terminal based on the WiMAX802.16m standard that includes a camera and performs image processing algorithms together with demodulation and channel decoding. The transmitted information is divided into frames of length $N = 192$ symbols. As we consider an NB-LDPC over GF(64), each symbol corresponds to 6 bits. The NB-LDPC decoder is based on the L-Bubble-Check EMS algorithm, which presents an optimized complexity/performance trade-off [11].

To apply the principles of ACM techniques, the channel decoder and the demodulator are implemented within a dynamic reconfigurable area or block of the FPGA. Moreover, the FPGA could be designed so that several MCS (i.e. decoder-demodulator blocks) are implemented at the same time but in such a way that two configurations are never active simultaneously (as this would lead to a waste of resources). On the other hand, at each reconfiguration, it would be necessary to download the configuration file into the FPGA, which requires a significant amount of time. In order to reduce the configuration time, whenever a configuration is active, the other has the opportunity to be downloaded independently in another part of the FPGA.

The tasks related to the image processing are acquisition, average, subtract, threshold, erosion, dilatation, reconstruction, labelling, wrapping, computation of the center of gravity and display. These are all implemented in hardware with the

exception of acquisition, computation of the center of gravity and display, which are executed on the processor. The hardware image processing tasks correspond to static hardware accelerator blocks and are obviously not reconfigurable.

Two *sensor* tasks are considered: the *speed_sensor* task, which periodically determines the mobile speed, and the *channel_sensor* task, whose role is to identify the channel properties. The first one is performed in software and provides an estimation of the speed of the mobile terminal.

The *channel_sensor* task, also performed in software, provides an estimation on the channel propagation conditions. In the WiMAX 802.16m standard, this information is included in the Channel Quality Indicator (CQI), which is utilized to provide channel-state information from the user terminals to the base station scheduler. In our study, we only consider the SNR as the channel parameter that determines the MCS to be used in order to maximize the spectral efficiency of the transmission. Figure 1 shows the spectral efficiency that can be obtained while guaranteeing a Frame Error Rate lower than 10^{-2} . The value of the spectral efficiency corresponds to a MCS configuration.

Then, in our simplified model, the MCS choice is performed in two steps: the *speed_sensor* information provides the channel model (AWGN or Pedestrian) and the channel estimation (SNR parameter) delivers the best MCS configuration.

Each time the *channel_sensor* task is executed, a new configuration may be loaded into the FPGA while the received information continues to be processed in the previous configuration. In the following, the simulation of our model is going to evaluate the correctness of such implementation. The target is a Virtex5 FPGA comprising a PowerPC processor at 400 MHz. This processor executes the real-time kernel as well as the software functions.

3.3. Platform description for the wireless mobile terminal

3.3.1. Application Model

In order to feed the model, all software or hardware tasks were implemented and the different execution times were measured on the Virtex5 FPGA target. The configuration time of the decoder and demodulator were also determined.

An application .xml description file was generated taking into account synchronization and communication between tasks. Table 2 summarizes the information about tasks while Figure 2 describes the application tasks' graph.

3.3.2. Kernel Model

Based on the the application requirements, a simple kernel based on MicroC OSII [12] has been modeled. The kernel provides the basic mechanisms that are required to manage tasks (creation, destruction, etc.) and synchronization mechanisms (mutexes, semaphores, etc.). Finally, message queues

Task's name	Type	Execution time (ms)
speed_sensor	software	0.2
channel_sensor	software	0.3
decoding_i	hardware	0.172 to 0.355
demodulation_i	hardware	0.003 to 0.012
acquisition	software	0.5
average, subtract, threshold	hardware	4.74
erosion, dilatation	hardware	0.421
reconstruction	hardware	9.53
labelling	hardware	7.52
wrapping	hardware	8.48
center of gravity	software	0.02
display	software	40

Table 2. Tasks' attributes

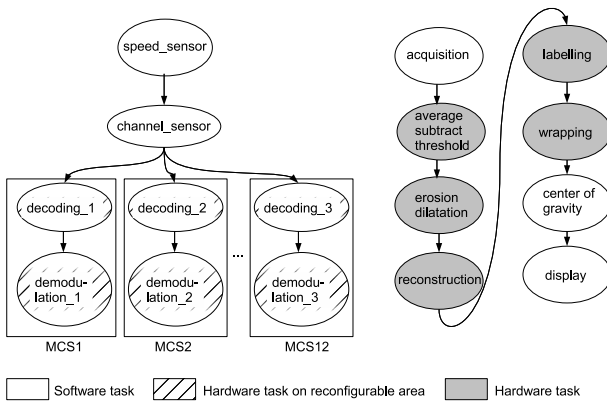


Fig. 2. Tasks' graph of the application

are also needed in order to transfer data from one task to another. Initially, the task's content is purely virtual and consists of an infinite loop using the *wait()* primitive of SystemC. This primitive can model the execution time of hardware and software tasks. Moreover, the execution time of each kernel service has been measured on the target and provided to the model.

3.3.3. Hardware Platform Model

The hardware platform model consists of a general processor and a dynamically reconfigurable block of an FPGA. Since the hardware image processing tasks consist of static hardware blocks, they have not been included in this model; only the software image processing tasks have been represented as a single continuous task. At this level of description, the processor is simply a black box and the reconfigurable area is represented by a number of logic cells that can accommodate the different dynamic hardware blocks. The attributes of this platform have also been determined (processor speed, memory resources and block size of logic resources).

4. RESULTS

The entire model has been provided to the OveRSoC tool in order to generate the SystemC executable file. Several simulations have been performed and have demonstrated the feasibility of the approach.

In order to depict the use of the exploration methodology, we propose to present two simulation scenarios for the proposed platform. The first scenario consists in specifying attributes that lead to a bad functioning of the complete system. After analyzing the obtained metrics, the user will modify the platform attributes in order to get a satisfactory outcome from the simulator.

In the first scenario (scenario 1), the reconfigurable hardware dimensions are set to approximately match the size of one couple of decoding-demodulation blocks. It is also assumed that the channel conditions are known and that the system starts with a default configuration : *decoding_1_HW* and *demodulation_1_HW* (see Figure 3). In this simulation, the kernel starts by executing the *speed_sensor_SW* task; the *channel_sensor* task execution follows and new channel conditions are detected. Before continuing with the *img_processing_SW* task, the new configuration signal is asserted. Following the kernel indications, the reconfiguration service tries to set up the new configuration blocks before stopping the old ones, as specified in the application requirements. Note that the demodulation block is the only block that is configured (due to a smaller number of logic cells compared to the decoding block). Therefore, we can see that the *decoding_1_HW* task continues to execute while the demodulation process is carried out by the *demodulation_2_HW* task, which constitutes a system dysfunction. The same problem is observed at 8000ns.

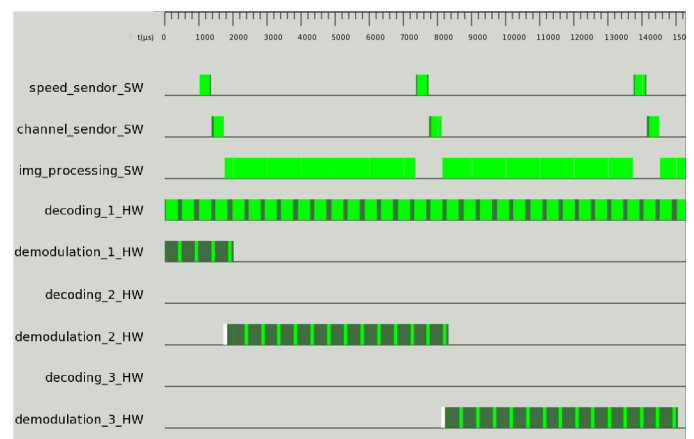


Fig. 3. Gantt chart for scenario 1. In white color, the configuring time for HW tasks. In lighter color, the executing time. In darker color, the synchronizing time

According to the methodology, the designer is then forced to modify specific attributes and to perform several simula-

tions until an appropriate outcome is reached. After several iterations, it is found that one viable solution consists in modifying the size of the reconfigurable area so that two demodulators and two decoders may be implemented at the same time.

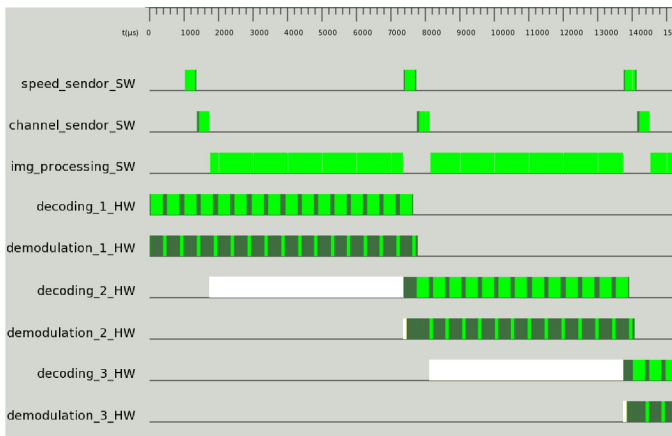


Fig. 4. Gantt chart for scenario 2

Figure 4 shows the model simulation with a larger dynamic reconfigurable area (scenario 2). Now, the blocks from configuration 2 are successfully configured into the FPGA while configuration 1 is still working. In this scenario, the incoming data are processed by the correct tasks, which means that the system is capable of switching between configurations without losing any information. At the time of the next new configuration request (near to 8000ns), the configuration service takes advantage of the inactive state of the configuration 1 blocks, to replace them with the configuration 3 blocks. This way, the user may determine the adequate dimensions of the dynamic reconfigurable area to guarantee the correct functioning of the system with an optimal number of resources. Note that in the case that the configuration 1 is requested instead of the configuration 3, the configuring time would not be necessary.

5. CONCLUSION

In this paper, we have considered a high-level methodology to model partial and dynamic reconfiguration on an ACM mobile wireless terminal based on the WiMAX 802.16m standard. This approach allows to rapidly explore architectural choices to validate the implementation of the real-time application. High-level simulation results have demonstrated the feasibility of the approach. The outlook of this work will consist in implementing the application on the target platform and test the effectiveness of the model. We also plan to improve the decision aspects and to include low power constraints in the model.

6. REFERENCES

- [1] B. Miramond, E. Huck, and F. et al. Verdier, "Oversoc : a framework for the exploration of rtos for rsoc platforms," *International Journal on Reconfigurable Computing*, vol. 2009, no. 450607, pp. 1–18, dec 2009.
- [2] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over GF(q)," *IEEE Communications Letters*, vol. 2, no. 6, pp. 159–166, june 1998.
- [3] D. Sridhara and T.E. Fuja, "Low density parity check codes defined over groupes and rings," in *Proc. Inf. Theory Workshop*, Oct. 2002.
- [4] D. Declercq, M. Colas, and G. Gelle, "Regular $gf(2^q)$ -ldpc coded modulations for higher order qam-awgn channel," in *Proc. ISITA*. Parma, Italy, Oct. 2004.
- [5] X. Jiand, Y. Yan, X. Xia, and M.H. Lee, "Application of non-binary ldpc codes based on euclidean geometries to mimo systems," in *Int. Conference on wireless communications and signal processing*. Nanjing, China, Nov. 2009.
- [6] S. Pfletschinger and D. Declercq, "Getting closer to mimo capacity with non-binary codes and spatial multiplexing," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, dec. 2010, pp. 1–5.
- [7] F. Guo and L. Hanzo, "Low-complexity non-binary ldpc and modulation schemes communicatins over mimo channels," in *IEEE Vehicular Technology Conference (VTC'2004)*. Los Angeles, USA, Sept. 2004.
- [8] A. Mourad and I. Gutierrez, "Davinci report: System level evaluation, issue 2," <http://www.ict-davinci-codes.eu/project/deliverables/D232.pdf>.
- [9] IEEE 802.16m, "Evaluation methodology document (emd)," 15 January 2009.
- [10] IEEE 802.16m, "System description document (sdd)," 10 April 2009.
- [11] E. Boutillon and L. Conde-Canencia, "Simplified check node processing in nonbinary ldpc decoders," in *Proc. Int. Symp. Turbo Codes*, Brest, France, Sept. 2010.
- [12] ,," <http://www.micrium.com>.