Abstract

The discussions around the channel coding theory were intense in the last decades, but even more interest around this topic was added once the turbo codes were found by Berrou, Glavieux, and Thitimajshima.

At the beginning of their existence, after proving the obtained decoding performances, the turbo codes were introduced in different standards as recommendations, while convolutional codes were still mandatory. The reason behind this decision was especially the high complexity of turbo decoder implementation. But the turbo codes became more attractive once the supports for digital processing, like Digital Signal Processor (DSP) or Field Programmable Gate Array (FPGA), were extended more and more in terms of processing capacity. Nowadays the chips include dedicated hardware accelerators for different types of turbo decoders, but this approach makes them standard dependent.

The Third-Generation Partnership Project (3GPP) is an organization, which adopted early these advanced coding techniques. Turbo codes were standardized from the first version of Universal Mobile Telecommunications System (UMTS) technology, in 1999. The next UMTS releases (after High Speed Packet Access was introduced) added support for new and interesting features, while turbo coding remained still unchanged. Some modifications were introduced by the Long Term Evolution (LTE) standard, not significant as volume, but important as concept. While keeping exactly the same coding structure as in UMTS, 3GPP proposed for LTE a new interleaver scheme.

Various UMTS dedicated turbo decoding schemes were presented in the literature. Due to the new LTE/ LTE-A interleaver, the decoding performances are improved compared with the ones corresponding to UMTS standard. Moreover, the new LTE interleaver provides support for the parallelization of the decoding process inside the algorithm, taking advantage on the main principle introduced by turbo decoding, i.e., the usage of extrinsic values from one turbo iteration to another. Parallelization is required in order to obtain high data rates, especially when diversity techniques are used.

There are many parallel decoding architectures proposed in the literature in the last years. The obtained results are evaluated on 2 axes. The first one is the decoding performances degradation introduced by the parallel method compared with the serial decoding scheme and the second one is the amount of resources needed for such parallel architecture implementation. A first set of parallel architectures contains the following idea. Starting from the classical method of implementing the Maximum A Posteriori (MAP) algorithm, i.e., going to trellis once to compute the Forward State Metrics (FSM) and then twice to compute the Backward State Metrics (BSM) and also the Log Likelihood Ratios (LLR), several solutions to reduce the decoding latency of $2K$ clock periods per semi-iteration, where K is the data block length, are introduced. The first one reduces the decoding time to half (only K) by starting simultaneously the BSM and FSM computation. After computing half of these values, 2 LLR blocks start working in parallel, the interleaver block being also doubled. Another proposed scheme eliminates the need for the second

interleaver but increases the decoding time with $K/2$ compared with the previous one, a total decoding latency of $3K/2$ clock periods being obtained.

A second set of parallel architectures takes advantage of the Quadratic Permutation Polynomial (QPP) interleaver algebraic-geometric properties. Here efficient hardware implementations of the QPP interleaver are proposed, but the parallelization factor N represents also the number of used interleavers in the proposed architectures.

A third approach consists in using a folded memory to store simultaneously all the values needed for parallel processing. This solution is efficient in terms of power consumption, a comparison with other methods being available in the literature. But for this kind of implementation the main challenge is to correctly distribute the data to each decoding unit once a memory location containing all N values was read. More precisely, the N decoding units working in parallel are writing their data in a concatenated order to the same location, but when the interleaved reading is taking place, these values are not going in the same order to the same decoding unit, but instead they should be redistributed. To solve this, an architecture based on 2 Batcher sorting networks is proposed. But also in this approach, N interleavers are needed to generate all the interleaved addresses that input the master network.

In this thesis, we introduce also a folded memory based approach, but the main difference comparing with the already existent solutions described above is that our proposed solution uses only one interleaver. Additionally, with an even-odd merge sorting unit, the parallel architecture remains close to the serial one, only the Soft Input Soft Output (SISO) decoding unit being instantiated N times. The block memories numbers and dimensions are unchanged between the two block schemes. In terms of decoding performances, with the cost of a small overhead added, the performances of the serial and parallel decoding architectures are kept similar.

This habilitation thesis summarizes the main contributions of the author in the field of turbo codes implementation for latest wireless communication systems, covering the time-frame following his Ph.D dissertation. Chapter 1 presents, from a general point of view, the convolutional turbo codes, including the turbo coding principles, the types of constituent encoder and interleavers. Chapter 2 continues the discussion, introducing the WiMAX turbo codes particularities, explaining in addition the trellis diagram and the puncturing procedure. Chapter 3 repeats the particularization, but this time for LTE/ LTE-A communication systems. In Chapters 4 and 5 the turbo decoding procedure is explained for WiMAX, respectively for LTE/ LTE-A systems. The SISO units are described, together with the corresponding theoretical latency and decoding rate. Chapter 6 introduces the proposed solutions for the FPGA implementation of the 2 decoding structures, for WiMAX and LTE/ LTE-A. Moreover, for LTE/ LTE-A systems, parallelization methods are developed by the authors, using the properties of the QPP interleaver, for which an efficient implementation scheme is also proposed. The Chapter presents ate the end throughput and speed results obtained when targeting a XC5VFX70T chip on Xilinx ML507 board. Chapter 7 provides simulation curves indicating the influence of the block dimension, number of iterations, and digital modulation type on the decoding performance. Also, there are compared the results obtained when using serial decoding, parallel decoding, and parallel decoding with overlap. The Conclusions section provides the main conclusions of this work, indicating also the research directions the authors take into consideration for this topic, and outlines several aspects related to the evolution and development of the author's academic career.

Rezumat

Discuţiile în jurul teoriei codării canalului radio au fost intense în ultimele decenii, iar un interes crescut a fost adăugat acestui subiect cu introducerea codurilor turbo de către Berrou, Glavieux, şi Thitimajshima.

La începutul existenţei lor, după demonstrarea performanţelor de decodare obţinute, codurile turbo au fost introduse în diferite standarde ca recomandări, în timp ce codurile convoluţionale încă erau obligatorii. Motivul din spatele acestei decizii a fost în special complexitatea ridicată a implementării decodorului turbo. Dar codurile turbo au devenit din ce în ce mai atractive pe masură ce suportul pentru procesarea digitală, precum procesoarele digitale de semnal (Digital Signal Processor - DSP) sau ariile de logică programabilă (Field Programmable Gate Array – FPGA) au avut o extindere continuă din punct de vedere al capacităţii de procesare. În prezent aceste dispozitive includ acceleratoare hardware dedicate pentru diferite tipuri de decodoare turbo, dar această abordare le face dependente de un anumit standard.

Third-Generation Partnership Project (3GPP) este o organizaţie de standardizare care a adoptat timpuriu aceste tehnici avansate de codare. Codurile turbo au fost standardizate din prima versiune a tehnologiei Universal Mobile Telecommunications System (UMTS), în 1999. Următoarele versiuni alea UMTS (după introducerea High Speed Packet Access) au adăugat suport pentru funcţionalităţi noi şi interesante, dar codurile turbo au rămas în continuare neschimbate. Unele modificări au fost introduse de standardul Long Term Evolution (LTE), nu semnificative ca pondere, dar foarte importante ca idee conceptuală. Păstrând aceeaşi schemă de codare ca în UMTS, 3GPP propune în LTE o nouă schemă de intreţesere.

Diverse arhitecturi de decodare turbo dedicate standardului UMTS sunt prezentate în literatura de specialitate. Datorită noului bloc de întreţesere din LTE/ LTE-Advanced, performanţele de decodare sunt îmbunătăţite comparativ cu cele ale standardului UMTS. În plus, noul bloc de întreţesere LTE furnizează suport pentru paralelizarea procesului de decodare în interiorul algoritmului, beneficiind de principiul fundamental introdus de decodarea turbo şi anume folosirea valorilor extrinseci de la o iteraţie turbo la alta. Paralelizarea este necesară pentru obţinerea de rate ridicate de decodare, mai ales în contextul utilizării tehnicilor de diversitate.

Există multe scheme de decodare paralelă propuse în literatura de specialitate în ultimii ani. Rezultatele obţinute sunt evaluate pe două direcţii. Prima se refera la degradarea performanţelor de decodare provocată de utilizarea unei metode paralele de decodare comparativ cu schema de decodare serială, iar a doua vizează cantitatea de resurse necesare implementării unei astfel de arhitecturi paralele. Un prim set de arhitecturi paralele se bazează pe ideea următoare. Plecând de la metoda clasică a implementării algoritmului Maximum A Posteriori (MAP), şi anume parcurgând diagrama de stări (trellis) într-un sens pentru a calcula metricile Forward State Metrics

(FSM), respectiv în sens opus pentru a calcula valorile Backward State Metrics (BSM) şi rapoartele logaritmice de probabilitate Log Likelihood Ratios (LLR), sunt introduse mai multe soluţii de reducere a latenţei de decodare de $2K$ perioade de ceas per semi-iteraţie, unde K este lungimea blocului de date. Prima metoda înjumătăţeşte timpul de decodare (doar K perioade de ceas) prin calcularea simultană a BSM şi FSM. După calcularea a jumătate din aceste valori, două blocuri LLR pornesc calculul în paralel, necesitând în acelaşi timp şi două blocuri de întreţesere. O altă schemă propusă elimină nevoia utilizării unui al doilea bloc de întreţesere, dar creşte timpul de decodare cu $K/2$ în comparaţie cu soluţia anterioară, obţinându-se în final o latenţă a decodării de $3K/2$ perioade de ceas per semi-iteraţie.

Un al doilea set de arhitecturi paralele foloseşte proprietăţile algebrice şi geometrice ale blocului de întreţesere Quadratic Permutation Polynomial (QPP) din LTE. Aici sunt propuse implementări hardware eficiente ale blocului QPP, însă factorul de paralelizare N reprezintă în acelaşi timp şi numărul de blocuri de intreţesere necesare în schemă.

O a treia abordare constă în folosirea unei memorii pliate pentru a stoca simultan toate valorile necesare procesării paralele. Această soluţie este eficientă din punct de vedere al consumului de putere, iar o comparaţie cu alte metode este disponibilă în literatura de specialitate. Preocuparea principală pentru acest gen de implementare este însă distribuţia corectă a datelor către fiecare unitate de decodare atunci când este citită o locaţie de memorie conţinând cele N valori corespunzătoare. Mai exact, cele N unităţi de decodare care lucrează în paralel işi scriu datele într-o manieră concatenată la aceeaşi locaţie de memorie, dar atunci când se realizează citirea întreţesută aceste valori nu mai merg în aceeaşi ordine la aceleşi unităţi de decodare, ci ar trebui redistribuite. Pentru a rezolva această problemă a fost propusă o arhitectură cu două reţele de sortare Batcher. Dar şi în această abordare este nevoie de N blocuri de întreţesere pentru a genera toate adresele întreţesute care alimentează reţeaua principală.

În această teză de abilitare este propusă tot o metodă bazată pe o memorie pliată, însă diferenţa fundamentală comparativ cu soluţiile existente descrise mai sus este că arhitectura de faţă necesită utilizarea unui singul bloc de întreţesere, independent de ordinul de paralelizare N. În plus, folosind o unitate de sortare de tipul par-impar, arhitectura paralelă rămâne apropiată de cea serială, doar unitatea de decodare Soft Input Soft Output (SISO) fiind instanţiată de N ori. Numărul şi dimensiunile blocurilor de memorie rămân neschimbate între cele două scheme. Iar din punct de vedere al performanţelor decodării, rezultatele obţinute pentru soluţia paralelă sunt similare soluţiei seriale, preţul plătit fiind o suprapunere limitată la împărţirea blocului iniţial de date în cele N blocuri de date folosite în paralelizare.

Această teză de abilitare include deci contribuţiile principale ale autorului aduse domeniului teoriei şi implementării codurilor turbo în ultimele standarde de comunicaţie fără fir, acoperind perioada ulterioară prezentării tezei de doctorat. Capitolul 1 prezintă, într-o manieră generală, codurile turbo convoluţionale, incluzând principiile codării turbo, tipurile de codoare constituente şi exemple de blocuri de întreţesere. Capitolul 2 continuă descrierea, introducând particularităţile codurilor turbo din sistemele WiMAX şi prezentând suplimentar diagrama de stări aferentă unui cod şi blocurile de reducere a ratei native de codare (puncturing). Capitolul 3 reia particularităţile codurilor, de această dată pentru sistemele de comunicaţii LTE/ LTE-A, tratând din nou problematica diagramei de stări şi a blocului de întreţesere. În Capitolul 4 şi Capitolul 5 sunt

explicate procedurile de decodare aferente codurilor sistemelor WiMAX, respectiv LTE/ LTE-A. Sunt descrise aici unităţile SISO, incluzându-se informaţii despre latenţele de decodare teoretice şi ratele de decodare aferente. Capitolul 6 prezintă soluţiile propuse de către autor pentru implementarea pe FPGA a celor doua structuri de decodare WiMAX, respectiv LTE/ LTE-A. În plus, pentru sistemele LTE/ LTE-A, autorul dezvoltă metode de paralelizare a decodării, folosindu-se de proprietăţile blocului de întreţesere QPP, pentru care este de asemenea propusă o soluţie eficientă de implementare. Capitolul include la final rezultate de rată de decodare şi viteză de procesare obţinute pentru o implementare pe cipul XC5VFX70T de pe placa Xilinx ML507. Capitulul 7 furnizează rezultate ale simulărilor care descriu influenţa lungimii blocului de date, a numărului de iteraţii, a modulaţiei digitale folosite asupra performanţelor de decodare. De asemenea, sunt comparate rezultatele decodării obţinute în cazul serial, paralel şi paralel cu suprapunere. Ultima secţiune a tezei de abilitare conţine concluziile acestui studi, indicând în acelşi timp şi direcţiile de cercerate pe care autorul le va urma în viitor, dar şi diferite aspecte legate de evoluţia şi dezvoltarea carierei academice a autorului.