Parallel Implementation Strategies for MIMO ID-BICM Systems

M. Simarro[1], C. Ramiro[2], F.J. Martínez-Zaldívar[1], A.M. Vidal[2], A. González[1], G. Piñero[1], V.M. García[2]

[1] Instituto de Telecomunicaciones y Aplicaciones Multimedia
Universitat Politècnica de València
8G Building- Access D- Camino de Vera s/n- 46022 Valencia (Spain)

[2] Departamento de Sistemas y Computación (DSIC)
Universitat Politècnica de València
Camino de Vera s/n- 46022 Valencia (Spain)

Corresponding author: fjmartin@dcom.upv.es

Abstract

One of the current techniques proposed for multiple transmit and receive antennas wireless communication systems is the use of error control coding and iterative detection and decoding at the receiver. These sophisticated techniques produce a significant increase of the computational cost and require large computational power. The use of modern computer facilities as multi-core and multi-GPU (Graphics Processing Unit) processors can decrease the computational time required, representing a promising solution for the receiver implementation in these systems.

In this paper we explain how iterative receivers can improve the performance of suboptimal detectors. We also introduce a novel parallel receiver scheme based on a hybrid computing model where CPUs and GPUs work together to accelerate the detection and decoding steps; this design comes to exploit the features of the GPU NVIDIA Kepler architecture respect to the previous one in order to optimize the communication system performance.

Keywords: MIMO, ID-BICM, Multi-core/ Many-core, LDPC, Turbo Decoding.

1. Introduction

Multiple-input multiple-output (MIMO) communications systems make use of multiple transmit and receive antennas to improve data rate by increasing the channel capacity and spectral efficiency compared to single input-single output (SISO) systems. It is clear that MIMO has had a huge impact on wireless communication and has been adopted by many modern wireless standards, such as IEEE 802.11n [1] or 3GPP LTE [2]. However, the advantages provided by these systems are achieved at the expense of a higher cost of computation, making essential the implementations of MIMO systems with lower computational cost.

On the other hand, error control coding is necessary to improve reliability of practical systems. Therefore, a good combination of detection and decoding MIMO schemes has drawn attention in recent years. With the publication of the landmark paper on trellis-coded modulation (TCM) of Ungerboek [3] modulation and coding were combined into a single entity to obtain better performance in digital communications. Subsequently with the introduction of BICM, scheme proposed by Zehavi [4], coding and modulation are optimized separately. His scheme increases the time diversity of coded modulation and therefore improves the performance of TCM over Rayleigh fading channels; however over nonfading Gaussian channels, degradation occurs [5].

In [6] Li and Ritcey applied the turbo decoding principle to BICM, known as BICM with iterative decoding (ID-BICM). In this scheme, extrinsic information is calculated for each bit after the first pass, interleaved and then fed back to the demodulator as a priori information on the channel received symbols. ID-BICM significantly outperforms the TCM over both with less computational complexity [5].

This scheme has received attention in wireless communications due to its power and bandwidth efficiency. Indeed, these advantages have motivated the extension of ID-BICM to MIMO systems [7][8]. Clearly, the study of MIMO

We provide an efficient parallel implementation of the MIMO ID-BICM receiver that takes advantage of current multi-core and many-core processors, as multi-GPUs.

ID-BICM systems is important in the nowadays investigation for the current research on decoding, however the implementation difficulties that arise when using these techniques are complex and thus it is necessary to develop efficient algorithms capable of running on high performance computers, such as multi-core systems and GPUs.

The number of scientific contributions and research projects related to the use of GPUs as general purpose computers (GPGPUs) and multi-core processors has significantly increased. The simultaneous use of these two architectures allows fully exploiting the resources of the machine and reducing the response time of high computational cost problems. This approach is currently used, for example, in numerical linear algebra libraries as MAGMA [9] or CULA [10]. GPUs are powerful tools and represent a quantitative leap in the development of high performance hardware. Last year NVIDIA launched its new Kepler architecture designed to maximize computational performance with superior power efficiency, by offering new features to optimize and increase parallel workload execution and therefore offering much higher processing power than the previous Fermi architecture. The new architecture presents relevant improvements that make hybrid computing easier, and very appropriate for a wider set of problems.

This paper aims to provide an efficient parallelization scheme of the receiver implementation in MIMO ID-BICM systems using current computers as multi-core and multi-GPU systems. The proposed scheme tries to take advantage of the new innovations offered by Kepler architecture. Below, we describe two of the most important features of the new architecture used in our paper (see [11] for a detailed description):

Hyper-Q: Enables multiple CPU cores to launch work on a single GPU simultaneously, thereby dramatically increasing GPU utilization and reducing CPU idle times. While the Fermi architecture could support 16 concurrent kernels launches, all of them ending up in the same hardware work queue that put them in a semi-serial state, the new Kepler offers a 32 work queue hardware managed by CUDA Work Distributor (CWD). The main advantage of the CWD is that it can support running kernels from different CUDA streams or even from multiple threads within a process but because they are independent, operations in one stream will not delay the execution of the rest of the streams.

Dynamic parallelism: It allows the GPU to self-schedule and self-generate new workloads based on programmer parameters without the need of intervention from the CPU. Any kernel can launch another kernel and can create the necessary streams, events, and dependencies needed to process additional work without the need for host CPU interaction. This feature should allow a larger variety of parallel workloads to be converted to GPU processing including those with nested loops or basic required serial control tasks.

The rest of the paper is organized as follows. Section 2 introduces the main concepts of MIMO ID-BICM; two channel decoders are reviewed and a scheme of iterative receiver is proposed. In section 3 we show how to parallelize the receiver for MIMO ID-BICM communication systems on multi-core and many-core architectures. Finally, main conclusions are reported in section 4.

2. General Scheme MIMO ID-BICM

In this section we recall the model of MIMO ID-BICM systems, which is composed of n_T transmit antennas and n_R receive antennas. A block diagram of this scheme is shown in Fig. 1.

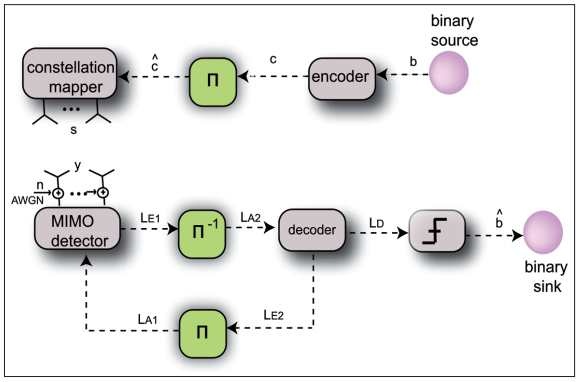

Figure 1. *MIMO ID-BICM block diagram.*

In the transmitter, the information bits $\boldsymbol{b}$ are encoded using an error-correction code, such as turbo codes or LDPC (Low Density Parity Check Codes). The coded bits are then passed through an interleaver $\boldsymbol{\pi}$ and mapped to symbols which are split into n_T transmit antennas. Thereby a complex signal $\boldsymbol{s}=(s_1, \ldots, s_{n_T})^T$ is sent, where each vector component is a symbol s_i taken from a constellation Ω of size $|\Omega|=M$. The baseband equivalent model for the received vector over a Gaussian channel is given by

$$\mathbf{y} = \mathbf{Hs}+\mathbf{n} \tag{1}$$

Where $\mathbf{H}$ is the channel matrix formed by $n_R \times n_T$ complex value entries characterized as zero-mean and unit-variance complex independent Gaussian random variables. The transmitted signal $\boldsymbol{s}$ is perturbed with additive white Gaussian noise $\boldsymbol{n}$.

At the receiver, the detector uses the received vector $\mathbf{y}$ and the known channel matrix $\mathbf{H}$ to calculate the reliability information of transmitted coded bits in form of real valued log-likelihood ratios (LLRs).

$$L_{j,b} = \ln \frac{P(x_{j,b}=1/\boldsymbol{y})}{P(x_{j,b}=0/\boldsymbol{y})} \tag{2}$$

in this system model $x_{j,b}$ denotes the *b-th* bit in symbol *sj*. Using the max-log approximation [10], the LLR can be calculated as:

$$L_{j,b} = \frac{1}{\sigma^2}\left[min_{s \in \chi_{j,b}^{(0)}} \{ \|\boldsymbol{y}-\mathbf{H}\boldsymbol{s}\|^2 - \log P\{\boldsymbol{s}\} \} - min_{s \in \chi_{j,b}^{(1)}} \{ \|\boldsymbol{y}-\mathbf{H}\boldsymbol{s}\|^2 - \log P\{\boldsymbol{s}\} \} \right] \quad (3)$$

where $\chi_{j,b}^{(c)}$ denotes the set of symbol vectors for which the *b-th* bit in symbol s_j equals c. The term -log P{$\boldsymbol{s}$} is iteratively calculated using the a *priori* information provided by the decoder (in the first iteration, this information is zero). Further details about the reliability information provided by the detector can be found in [10]. Next, these values are de-interleaved and used by the channel decoder to calculate their own log-likelihood ratios. The extrinsic information about each coded bit provided by the decoder is fed back as a priori information for the detector in the next iteration of the receiver.

The use of soft detection in MIMO BICM systems can substantially improve their performance with respect to the use of hard detection. Soft-output requires the computation of the LLR for every bit. The calculation of these values suffers from an exponential increase in the complexity with the number of antennas and the constellation size; thus the exact calculation of the LLRs has a very high computational cost. In response to this, numerous alternatives of suboptimal demodulators have been proposed in the literature [12, 13, 14]. A common solution is to calculate approximated LLRs.

One way to reduce the complexity is to perform the calculation of (3) using a list of candidates to obtain an approximation of the LLRs, thus:

$$\hat{L}_{j,b} = \frac{1}{\sigma^2}\left[min_{\boldsymbol{s} \in \mathcal{L} \cap \chi_{j,b}^{(0)}} \{ \|\boldsymbol{y}-\mathbf{H}\boldsymbol{s}\|^2 - \log P\{\boldsymbol{s}\} \} - min_{\boldsymbol{s} \in \mathcal{L} \cap \chi_{j,b}^{(1)}} \{ \|\boldsymbol{y}-\mathbf{H}\boldsymbol{s}\|^2 - \log P\{\boldsymbol{s}\} \} \right] \quad (4)$$

where $\mathcal{L}$ is the candidate list, being a reduced set of all possible transmitted vectors. The size of the candidate list $|\mathcal{L}|$ offers a trade-off between performance and complexity. Next section will show one of the procedures that have been implemented to obtain the list.

In summary, in MIMO ID-BICM the soft detector estimates the bits that are mapped into the transmit vector and provides information about how reliable these estimates are. This supplementary information is exploited by a channel decoder to achieve better decoding performance. Moreover, the performance is further improved using iterative receiver structures where the detector and the decoder exchange information during certain number of iterations (N_{iter}). This parameter provides a trade-off between complexity and performance.

2.1 Channel coding

Practical wireless communication schemes utilize channel codes such as convolutional, Turbo or Low-Density Parity-Check (LDPC) codes. The purpose of channel coding is to reduce the bit error rate. In our implementation the specific channel coding considered has been the Turbo Codes (TC). Currently, our purpose is to use the channel encoder LDPC codes together with the idea of list decoding. Next, we provide a brief description of LDPC and TC.

2.1.1 LDPC codes

LDPC codes allow communicating effectively very close to the Shannon capacity limit of the channel and at a low complexity in a wide range of communications systems. Today LDPC decoders are strong competitors to turbo codes and are even used in many standards, such as DVB-S2 [15], WiMax [16] or WiFi [17].

LDPC codes are linear block channel codes with sparse (matrix with few ones in comparison to the number of zeros) parity-check matrix **H**. A convenient way to describe an LDPC code is in terms of its factor graph (Tanner graph). Tanner introduced an effective graphical representation for LDPC codes which helps to describe the decoding algorithm. The graph is a bipartite graph meaning that the nodes are separated into two sets. On one side we can find variable nodes c_j, and on the other side the graph has nodes called check nodes f_i. The two types of nodes are connected if the element h_{ij} of **H** is equal to 1. In Fig. 2 an example of Tanner graph is shown associated with the check equation and the parity check matrix of equation (5), where the rows are the check nodes and the columns represent the variable nodes.

$$\boldsymbol{H} = \begin{array}{c} \begin{matrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{matrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{matrix} \quad (5)$$

$$f_0 = c_1 + c_3 + c_4 + c_7$$
$$f_1 = c_0 + c_1 + c_2 + c_5$$
$$f_2 = c_2 + c_5 + c_6 + c_7$$
$$f_3 = c_0 + c_3 + c_4 + c_6$$

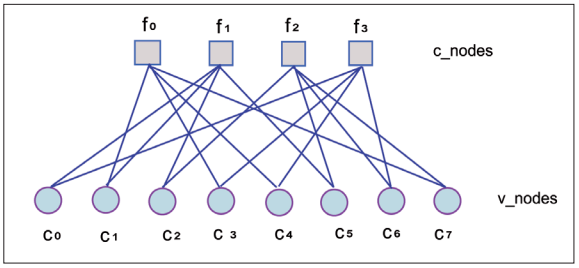

Figure 2. *Tanner graph of a linear block code parity-check matrix H.*

These codes achieve a remarkable performance with iterative decoding based on belief propagation; decoding details can be found in [18]. Furthermore, this idea of it-

In MIMO ID-BICM the soft detector estimates the bits that are mapped into the transmit vector and provides information about how reliable these estimates are. This extra information is exploited by iterative channel decoders to improve its error performance.

erative decoding can be combined with the list decoding concept introduced by Elias [19]. In this type of decoding the idea is to generate a list of codewords containing the closest codewords to the received vector opposed to single decoding that returns a single codeword.

2.1.2 Turbo Codes

One of the most used channel codes are convolutional codes, with the decoding strategy based on the Viterbi algorithm. The advantages of convolutional codes are used in turbo codes [20] which are the specific channel coding scheme considered in this work.

The structure used in turbo encoders is shown in Fig. 3. The turbo encoder uses two Recursive Systematic Convolutional (RSC) codes to encode the same input bits, with an interleaver Π between the encoders.

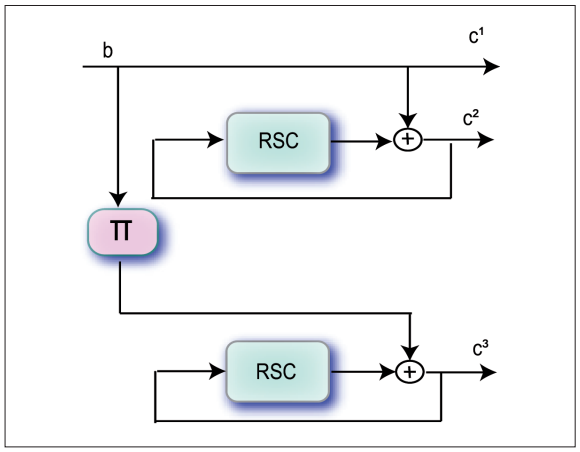

Figure 3. *Turbo encoder with rate 1/3.*

The first encoder operates on the input bits, represented by $\boldsymbol{b}$, while the second encoder operates on the input bits which have been permuted by the interleaver. The output then is represented by $\boldsymbol{c}=(\boldsymbol{c}^{(1)},\boldsymbol{c}^{(2)},\boldsymbol{c}^{(3)})$. If the input frame $\boldsymbol{b}$ is of length k and the output frame $\boldsymbol{c}$ of length n, then the encoder rate is $R=k/n$.

At the receiver the general structure of an iterative turbo decoder is shown in Fig. 4. Two decoders are linked by interleavers in a structure similar to the encoder. In our case the decoder receives as input the LLRs of the received sequence.

These LLRs are the soft output of the previous detector and provide information to the decoder about the reliability of each bit in the sequence. In the first iteration, the first decoder takes only the values from the detector and produces a soft output. The soft output from the first encoder is then used as additional information for the second decoder, which uses this information with the detector output to calculate its own estimate of the data bits. In the second iteration the first decoder decodes again, but now with additional information about the value of the input bits provided by the output of the second decoder in the first iteration. This cycle is repeated until certain condition is reached, achieving an improvement in the Bit Error Rate at each new iteration.

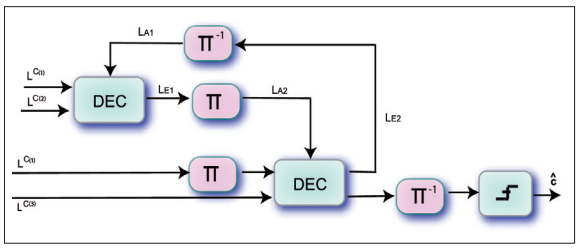

Figure 4. *Turbo decoder scheme.*

2.2. The proposed generation of candidates

It has been discussed previously that, due to the computational complexity in the calculation of the exact LLRs, one of the alternatives is the use of a list with possible candidates to calculate an estimate of these values. The candidate list can be generated using several techniques such as list sphere decoder, lattice reduction or bit flipping.

As we can see in Fig. 5 and it is detailed below in our implementation, we have calculated the values in (3) by bit flipping technique.

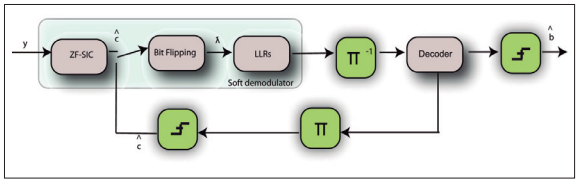

Figure 5. *Receptor scheme implemented.*

In this scheme the Zero-Forcing (ZF) detector is implemented. Hard output MIMO detectors are alternatives to soft demodulators that provide decisions for the coded bits but they do not supply reliability information. ZF is a linear detector which provides sub-optimal performance but offers significant computational complexity reduction with tolerable performance degradation. The performance of this algorithm can be improved by using nonlinear techniques as detection with successive interference cancellation (SIC). The idea of our implementation is to use this simple detector with low complexity and to improve the performance of the whole system by means of a feedback stage.

In the first iteration a ZF-SIC [21] provides an estimate of the symbols vector that was broadcast $\hat{s}_{ZF\_SIC}$. From this estimate, the bit-flipping module generates an initial candidate list and the LLRs values are calculated from this list using (4). The candidate list is stored for subsequent iterations and the soft outputs are passed to the decoder. In the following iterations the bit-flipping mod-

ule takes the solution provided by the detector in the previous step as its starting point. The new list calculated is added to the initial list and the LLRs values are calculated now from all candidates.

As discussed before, one way of generating the candidate list is to flip some bits of the hard solution. The bit flipping module in Fig. 5 takes the output of ZF-SIC demodulator as a starting point in the first iteration whereas for the following iterations it takes the output of the decoder. In this manner, additional $log_2 M$ branches are considered at each level adding new candidates as it is shown in Fig. 6. The figure represents an algorithm for a 4 x 4 MIMO system with QPSK symbols, where the bits of the initial hard solution are $\hat{s}$=[10 11 01 00].

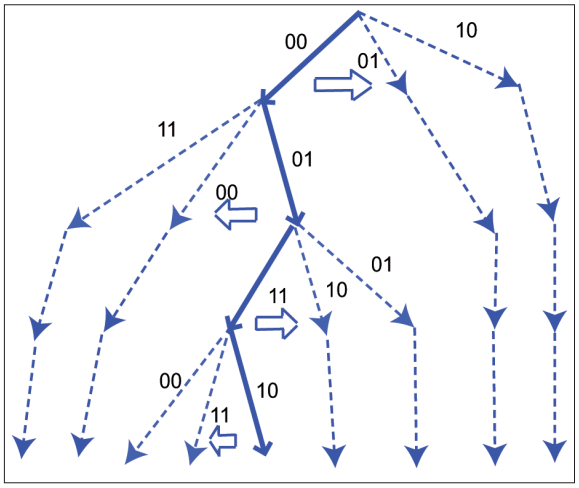

Figure 6. *Generating the candidate list.*

Each of these additional branches has one bit flipped with respect to its original hard symbol. Thus additional branches are obtained by flipping each bit and after that, these new branches are completed using SIC path until the bottom is reached. This operation is repeated for each level and finally the candidate list is obtained by adding the calculated paths to the initial hard solution.
In Fig. 7 we see the performance in terms of bit error rate (BER) of the proposed scheme compared to the same scheme without feedback and to the ZF-SIC hard detector. Results obtained from simulation using a 4 x 4 MIMO scheme and 16-QAM modulation have been depicted. The channel code is a turbo code with rate 1/3 and a block length of 10 000 bits.

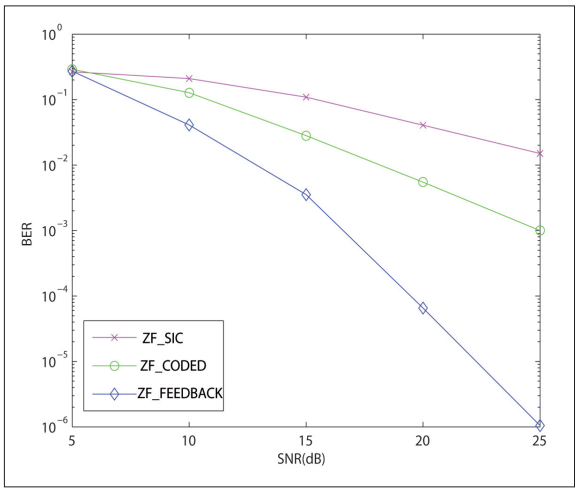

Figure 7. *BER performance for different MIMO receivers.*

It can be clearly observed that the hard detector ZF-SIC is outperformed when channel coding is added to the system. Thus, the curve ZF_{CODED} implements the scheme shown in Fig. 5, but without feedback between decoder and detection. We can see how the suboptimal performance of the hard detector (with a reduced computational cost) is improved due to the feedback when soft information is exchanged between the detector and the decoder.

As a future work, the channel decoder as Iterative List Decoding of LDPC Codes [22] is going to be implemented and the list provided by the decoder will be used for calculating the LLRs values for subsequent iterations.

3. Parallel implementations

Next, we will detail the most important aspects of the proposed parallel scheme, emphasizing the benefits we would get if we use a multi-core processor with two GPUs NVIDIA Kepler as accelerators.

The proposed GPU implementation will be composed of two CUDA kernels: the first one is the detector kernel and the second one is the decoder kernel. Although in Fig. 8 the bit interleaver and de-interleaver appear as independent modules we will assume that the de-interleaver is part of the detector and the interleaver is part of the decoder. These modules must be implemented and optimized in order to improve the performance of the CUDA code, for example using shared and constant memories (faster than global memory) to store work variables in order to accelerate the access to this data. We already have implemented a soft-output fixed sphere decoder and LDPC decoder in CUDA [23, 24] that can be used as alternatives.

In Fig. 8 it can be observed that the CPU is in charge of dividing the flow of symbols stored in a buffer in different codewords. For example if we are using codewords of $Length$= 4320 bits and 16-QAM modulation, the $\frac{4320}{log_2 16}$ = 1080 symbols correspond to the first codeword (CW1), the following 1080 symbols correspond to the second codeword (CW2) and so forth.

The CPU can create as many threads (MPI or OpenMP) as cores it has. Each codeword will be processed by one of these cores (threads), so by using Hyper-Q every thread will create a CUDA stream individually producing a separate connection between CPU and GPU, as shown in Fig. 9. This model offers high benefits due to the fact that multiple processes can share the GPU and increase its occupancy.

For each codeword, CPU copies its symbols in the global memory of the GPU. At this point the process of detect-

Our design exploits the maximum occupancy of the GPU since multiple words can be decoded in parallel through separated queues allocated at each CUDA stream

ing and decoding symbols begins. These algorithms are also implemented to make efficient use of the many threads of the GPU. As mentioned previously these implementations are already optimized using shared memory in order to accelerate the access to this data. However, we cannot use shared memory to store the LLRs calculated at the output of the bit interleaver/de-interleaver, because the shared memory cannot be passed between kernels, since the contents of a shared block are defined within an execution block of threads only.

Following our strategy of parallelization, we will use dynamic parallelism that allows the GPU to generate new work for itself, synchronize on results, and control the scheduling, all without involving the CPU. As shown in Fig. 10 when the detector kernel has finished, new child threads can be launched and spawned by adapting to the work needed in the decoder process. When the decoder kernel is completed a new detector step is executed by the original threads with the new LLRs values, if the number of iterations is smaller than N_{iter} (see section 2). With this new feature, the system CPU can then be freed up for additional tasks, and can use this time to transform LLRs to bits or symbols ($\hat{b}_i=1$ *if* $Lb_i>0$ *and* $\hat{b}_i=0$ *if* $Lb_i<0$).

4. Conclusions

The use of multi-core and many-core architectures for the efficient implementation of MIMO receivers represents a good alternative for the implementation of these systems due to its computational complexity. Therefore, in this

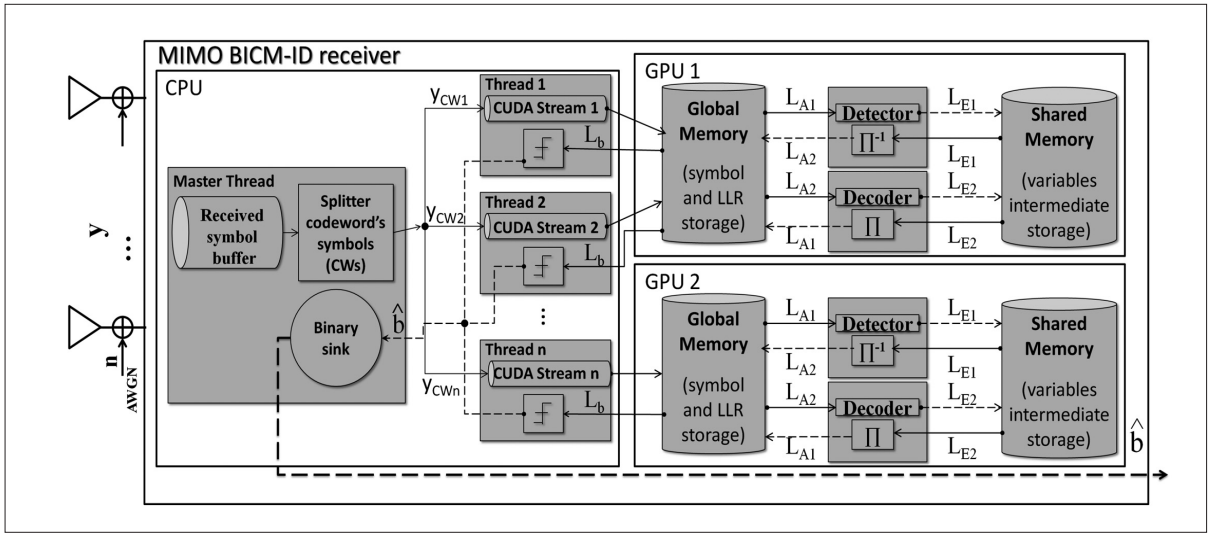

Figure 8. *High-level parallel architecture overview for the considered MIMO ID-BICM receiver.*

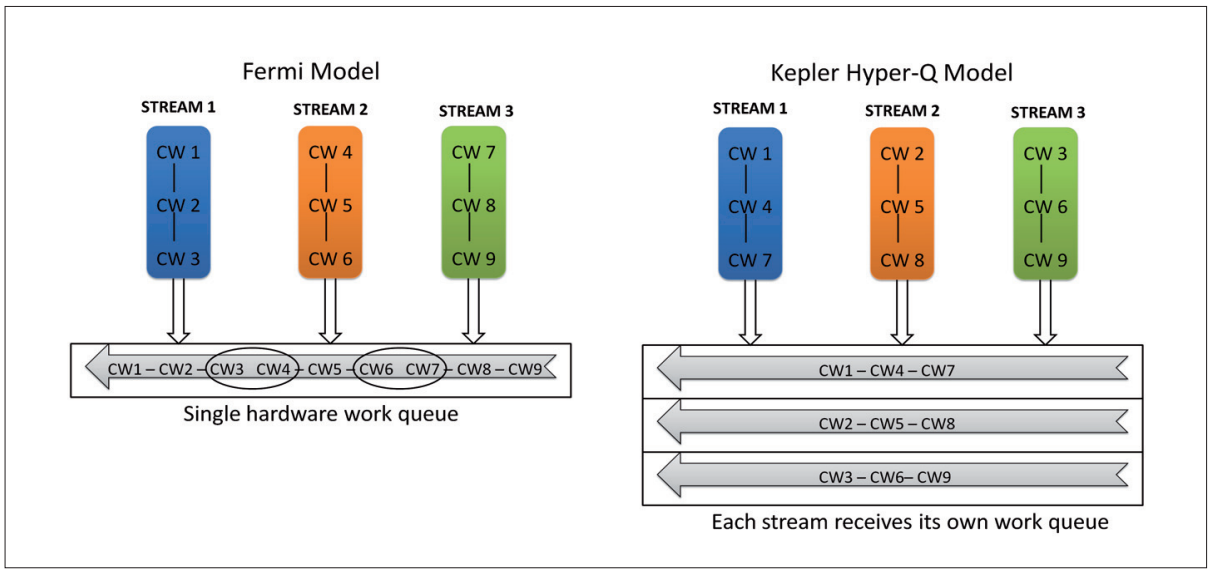

Figure 9. *Hyper-Q working with CUDA Streams: In the Fermi model shown on the left, only codewords (CW3, CW4) and (CW6, CW7) can be decoded concurrently due to intra-stream dependencies caused by the single hardware work queue. The Kepler Hyper-Q model allows all streams to decode simultaneously using separate work queues.*

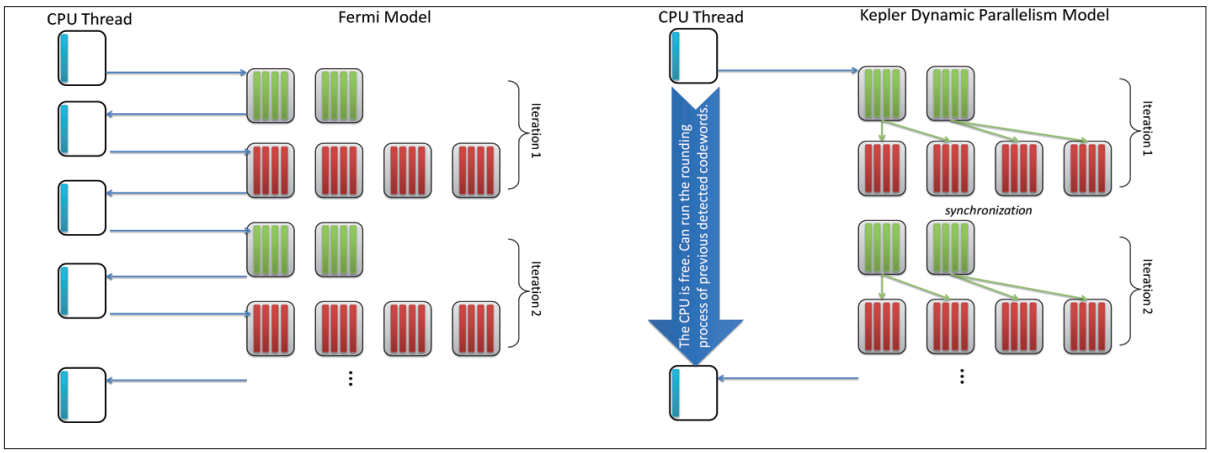

Figure 10. *Dynamic Parallelism allows more parallel code in our application to be launched directly by the GPU onto itself (right side of image), thus the CPU is free to run the rounding process, rather than requiring CPU intervention (left side of image).*

paper we have proposed a hybrid general scheme of a MIMO BICM iterative receiver. Our design allows the maximum occupancy of the GPU since we can decode multiple words in parallel through separate queues for each CUDA stream. In addition, our receiver model makes use of all the resources available in the system and can simultaneously run code on the CPU and the GPU.

In this paper iterative detection and decoding scheme for MIMO systems with error control coding have also been analyzed. We have proposed feedback using a single detection scheme, showing better performance in terms of BER than the same scheme without feedback. The method iterates the channel decoder and a soft detector is able to find a set of candidates from which the estimated LLRs values can be computed. This scheme can be applied to any type of error control code; in this paper we have explained the two most commonly used types of codes in wireless communication, showing results obtained with turbo code simulations. However, further work using LDPC codes with list decoding is currently being implemented in our simulations.

Acknowledgments

This work has been partially funded by PROMETEO/2009/013 project of Generalitat Valenciana, projects TEC2009-13741 of the Ministerio Español de Ciencia e Innovación, TEC2012-38142-C04 of the Ministerio Español de Economía y Competitividad, and PAID-05-2011 of Universitat Politècnica de València.

References

[1] IEEE Draft Standard; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications; Amendment 4: Enhancements for Higher Throughput, IEEE P802.11n/D3.0 Sep. 2007.

[2] 3rd Generation Partnership Project; Technical Specifications Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 9), 3GPP Organization Partners TS 36.212, Rev 8.3.0, May 2008.

[3] G. Ungerboek, "Channel coding with multilevel/phase signals", IEEE Trans. Inform. Theory, vol IT-28, pp 56-67, Jan.1982.

[4] E. Zehavi, "8-PSK trellis codes for a Rayleigh fading channel", IEEE Trans. Commun. Vol. 36, pp. 1004-1012, Sept. 1988.

[5] X. Li and J. Ritcey, "Bit interleaved coded modulation with iterative decoding using soft feedback", Electron. Lett. vol.34, no. 10, pp. 942-943, May 1998.

[6] G. Caire, G. Taricco and E. Biglieri, "Bit-Interleaved Coded Modulation", IEEE Transactions on Information Theory, Vol. 44, no. 3, May 1998.

[7] A.M. Tonello, "Space-time Bit-inerleaved Coded Modulation with an Iterative Decoding Strategy", Proc. of IEEEVehicular Technology Conference (VTC) Fall 2000, Boston, US, pp. 473-478, September 24-28, 2000.

[8] A.M. Tonello, "MIMO MAP Equalization and Turbo Decoding in Interleaved Space-time Coded Systems", IEEE Trans. On Communications, vol. 51, no. 2, pp. 155-160, February 2003.

[9] S. Tomov, R. Nath, P. Du, J. Dongarra. MAGMA Library, http://icl.cs.utk.edu/magma.

[10] CULA, tools. CULA Reference Manual. www.culatools.com.

[11] Kepler Compute Architecture http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf

[12] B. M. Hochwald and S. Ten Brink, "Achieving Near-Capacity on a Multiple-Antenna Channel". IEEE Transactions on Communications, vol. 51 no. 3, pp. 389-399, March 2003

[13] C. Studer, A. Burg and H. Bölcskei, "Soft-Output Sphere Decoding Algorithms and VLSI Implementation", IEEE Journal on selected areas in Communications, vol. 26, no. 2, pp. 290-300, February 2008.
[14] J. Boutros, N. Gresset, L. Brunel and M. Fossorier, "Soft-Input Soft-Output lattice sphere decoder for linear channels", Proc. IEEE GLOBECOM ´03, vol. 3, pp. 1583-1587, December 2003.
[15] European Telecommunications Standards Institute. Digital video broad-casting (DVB) second generation framing structure, cannel coding and modulation systems for broadcasting interactive services, news gathering and other broadband satellite applications. DRAFTEN 302307 DVBS2-74rl5, 2003.
[16] 802.11-2007 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
[17] IEEE 802.16e, "IEEE 802.16e TGe Working Document, (Draft Standard) –Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licenses Bands, 802.16e/D4", 2004.
[18] Venkatesan Guruswami, "Iterative Decoding of Low-Density Parity Check Codes (A Survey)", CoRR, September 2006.
[19] Elias, "List Decoding for Noisy Channel", Res. Lab Electron. MIT, Cambridge, MA, Techn Rep. 335, 1957.
[20] C. Berrou, "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)", IEEE International Conference on Communications, ICC '93. Vol.2, pp. 1064-1070, May 1993.
[21] P.W. Wolniansky, G.J. Foschini, G.D Golden, R.A Valenzuela, "V-BLAST: An arquitecture for realizing very high data rates over the rich scattering wireless channel". Proc 1998 Int. Symp. On Advanced Radio Technologies. Boulder, Colorado. 9-11 September 1998.
[22] T. Høholt and J. Justesen, "Iterative list decoding of LDPC codes", Proceedings of the 17th international conference on Applied algebra, algebraic algorithms and error-correcting codes, pp. 18-19, 2007.
[23] S. Roger, C. Ramiro, A. González, V. Almenar , A.M Vida, "An Efficient GPU Implementation of Fixed-Complexity Sphere Decoders for MIMO Wireless Systems", Integrated Computer-Aided Engineering, vol. 19, no.4, pp. 341-350, October 2012.
[24] F. J. Martínez-Zaldívar, A. M. Vidal-Maciá, Alberto Gonzalez, Vicenç Almenar, "Tridimensional block multiword LDPC decoding on GPUs", The Journal of Supercomputing, Volume 58, Issue 3, pp 314-322, December 2011.

Biographies

M. Ángeles Simarro Haro was born in Albacete, Spain in 1988. She received the Telecommunication Engineering degree from the Universitat Politècnica de València in September 2011 and the MSc. Degree in Telecommunication Technologies in July 2012. Currently, she is working on her research toward the PhD degree at UPV. Her research interest is focused on soft demodulation, decoding and iterative receivers in MIMO systems.

Carla Ramiro Sánchez was born in Valencia, Spain, in 1984. She received the Engineer Degree in Computer Science and Technical Engineer Degree in Telematics in 2009, both from the Universitat de València, and the MSc. degree in Parallel and Distributed Computing in 2010, from the Universitat Politècnica de València, Spain. She is working as assistant researcher in the Department of Information Systems and Computation at the Universitat Politécnica de València. Her research focuses on parallelization of signal processing problems on the different cores of a CPU and GPU.

Francisco José Martínez Zaldívar was born in Paiporta, Spain, in 1966. He received the Computer Science and Ph.D. degrees from the Universitat Politècnica de València, Spain, in 1990 and 2007 respectively. He is currently Associate Professor at the Departamento de Comunicaciones, Universitat Politècnica de València. His current research interests include parallel computing in signal processing

Antonio M. Vidal receives his M.S. degree in Physics from the "Universidad de Valencia", Spain, in 1972, and his Ph.D. degree in Computer Science from the "Universitat Politècnica de València", Spain, in 1990. Since 1992 he has been in the Universitat Politècnica de València Spain, where he is currently a full professor in the Department of Computer Science. He is the coordinator of the project "High Performance Computing on Current Architectures for Problems of Multiple Signal Processing", currently developed by INCO2 Group and financed by the Generalitat Valenciana, in the frame of PROMETEO Program for research groups of excellence. His main areas of interest include parallel computing with applications in numerical linear algebra and signal processing.

Alberto González was born in Valencia, Spain, in 1968. He received the Ingeniero de Telecomunicación degree from the Universidad Politecnica de Catalonia, Spain in 1992, and Ph.D degree from de Universitat Politècnica de València (UPV), Spain in 1997. His dissertation was on adaptive filtering for active control applications. From January 1995, he visited the Institute of Sound and Vibration Research, University of Southampton, UK, where he was involved in research on digital signal processing for active control. He is currently heading the Audio and Communications Signal Processing Research Group (www.gtac.upv.es) that belongs to the Institute of Telecommunications and Multimedia Applications (i-TEAM, www.iteam.es). Dr. Gonzalez serves as Professor in digital signal processing and communications at UPV where he heads the Telecommunications Faculty since 2012. He has published more than 70 papers in journals and conferences on Signal Processing and Applied Acoustics. His current research interests include fast adaptive filtering algorithms and multichannel signal processing for communications, 3D sound reproduction and MIMO wireless systems.

Gema Piñero (Ms. Telecommunications Engineering, Universidad Politécnica de Madrid '90; Ph.D. Engineering, Universitat Politècnica de València '97) was born in Madrid (Spain) in 1965. She is currently an Associate Professor at the Universitat Politècnica de València. Since 1990 she has been involved in different research projects including array signal processing, mobile communications and sound quality in the Audio and Communications Signal Processing (GTAC) group, www.gtac.upv.es, of the Institute of Telecommunications and Multimedia Applications (iTEAM) of Valencia. In 1999 she leaded several research projects on sound quality evaluation for the automotive industry. Since 2001 she has leaded or participated in several projects on WCDMA and 4G wireless communications supported by public and private funding (Spanish Government, Regional Government, Telefónica, NVIDIA). She has also published more than 50 contributions in journals and conferences on signal processing algorithms for sound and communications applications. Her current research interests include coordinated systems and spatial division multi-user techniques in wireless communications, and adaptive algorithms for distributed sensors in audio applications.

Víctor M. García obtained a degree in Mathematics and Computer Science (Universidad Complutense, Madrid) in 1991, later an MSc degree in Industrial Mathematics (University of Strathclyde, Glasgow) in 1992 and a Ph. D. degree in Mathematics (Universitat Politècnica de València) in 1998. He is a T.U. (senior lecturer) in the Universitat Politècnica de València, and his areas of interest are Numerical Computing, parallel numerical methods and applications.