Software Parallel Implementation of a DS-CDMA Multiuser Detector

Luís Carlos Gonçalves*, Rui Escadas Martins*, António Brito Ferrari*
*Instituto de Engenharia Electrónica e Telemática de Aveiro, Universidade de Aveiro,
3810-193 Aveiro, Portugal,
Emails: luisgo@ua.pt, rmm@ua.pt, ferrari@ua.pt

***Abstract*—In this article the complexity and runtime performance of two Multiuser Detectors for Direct Sequence-Code Division Multiple Access were evaluated in two hardware platforms in order of a possible deployment in base stations. The detectors are based on the Frequency Shift Canceller concatenated with a Parallel Interference Canceller. This detector is very scalable which shadows the complexity penalty of a serial implementation in which other detectors have advantage. Implementations for the Time Division-Code Division Multiple Access deployed in China, in two software platforms one in OpenMP and other in CUDA were done. It is demonstrated that a realtime implementation is possible with a General Propose Graphics Processor Unit. This is part of a broader project aiming to take advantage of present parallel hardware to bring improved Multiuser technology to the present and future Base Stations, mainly in Universal Mobile Telecommunications System-Time Division Duplex.**

I. Introduction

Time Division-Code Division Multiple Access (Universal Mobile Telecommunications System-Time Division Duplex (UMTS-TDD), 1.28MChips/s) has been deployed in the People's Republic of China since 2007. The hardware of the Base Stations (BS) are actually upgraded several times during a decade to meet the technology advances. Better Multiuser Detection (MUD) could be a solution to increase the spectral efficiency and wireless network coverage in the uplink of the existent stations or the new ones, as well to increase energy efficiency in Mobile Stations (MS). Also, it can decrease the number of diversity antennas in the base station, decreasing costs in hardware and increasing energy efficiency. Depending on the age and provider of those BS the upgrade can be done through a board connected to a backplane or connected through optic fiber to a standalone card or computer.

MUD algorithms could be deployed in MS Receivers and BS Receivers from the UMTS-TDD standard in all chip rates: 1.28 MChips/s, 3.84 MChips/s and 7.68 MChips/s. In this work is studied only a possible implementation in the BS, in uplink. MUD application to the uplink is transparent to the MS and the actual specifications of the UMTS-TDD had taken it in account.

At uplink the signal received at BS passed through different (transmission) channels. Here, MUD have a natural utilization, but nowadays is limited to simple detectors as the Parallel Interference Canceller (PIC). MUD have the goal of cancelling the others user's signals (Multiuser Access Interference (MAI)) to recover the user of interest. MUD is used in the receiver and act over the sampled spread signal at baseband. The use of MUD includes some single user detector functionality because it needs to deal with the channel distortion of the portion of the received signal related to the user of interest. The MUD detectors like the Minimum Mean Squared Error Detector (MMSE) and the Frequency Shift Canceller (FSC) can be integrated in a RAKE [1] (composed structures) and can be concatenated with a PIC or a Serial Interference Canceller (SIC) to improve even more its performance. The concatenation with a SIC is more appropriate for downlink because of different receiving signal's power and the PIC for uplink because of analogous receiving signal's power. In [2]–[4] such composed structures with multiuser, single user processing and spatial processing using configurations including the FSC concatenated with a PIC were studied.

The MMSE detector [5]–[7] implies the inversion of a large diagonal matrix typically with LsUxLsU size (Ls is the number of symbols in a slot and U the number of users). This is in an ideal case, as typically the upsampling and channel length must also be taken into account. It is expected, that much more memory resources are needed for the implementation of such algorithm. Also the MMSE Algorithm is not so scalable as the FSC one which has decoupled user processing and multiple small matrix inversions.

Also, despite Iterative Multiuser Detection being claimed as a less complex solution [8], the solution is found in an iterative way and so its implementation must be sequential (serial). Also, it can not converge to the right solution.

With the appearance of many parallel hardware processors and parallel languages or extensions for them [9], many algorithms could be implemented in software (versus Very Large Scale Integration) that were previously too complex. As far is the knowledge of the authors, this work is innovative in the sense that is the first time anyone tries to implement DS-CDMA MUD in parallel architectures as Graphical Processor Units (GPUs) to aim a practical use.

The Section II describes the DS-CDMA detector considered in this work. In Section III the implemented algorithms are presented and a complexity/performance analysis is made. And finally in Section IV, the main conclusions are stated.

II. Detector Description

The Multiuser Detector is constituted by the configuration of Figure 1 (Pre-FSC) [4, Fig. 6.4, Fig. 6.13] or configuration of Figure 2 (Post-FSC) [4, Fig. 6.3, Fig. 6.12] concatenated with a Hard-PIC. The Pre-FSC and Post-FSC are RAKE structures with the a FSC integrated in, with *Pre* and *Post* reflecting the relative position of the FSC to the Maximum Ratio-Shift(delay)-Combining (see Figure 2). The Figure 3 shows these detectors with the FSC concatenated with the PIC. The PIC have the same configuration of the detector Pre-FSC, first the cancelation

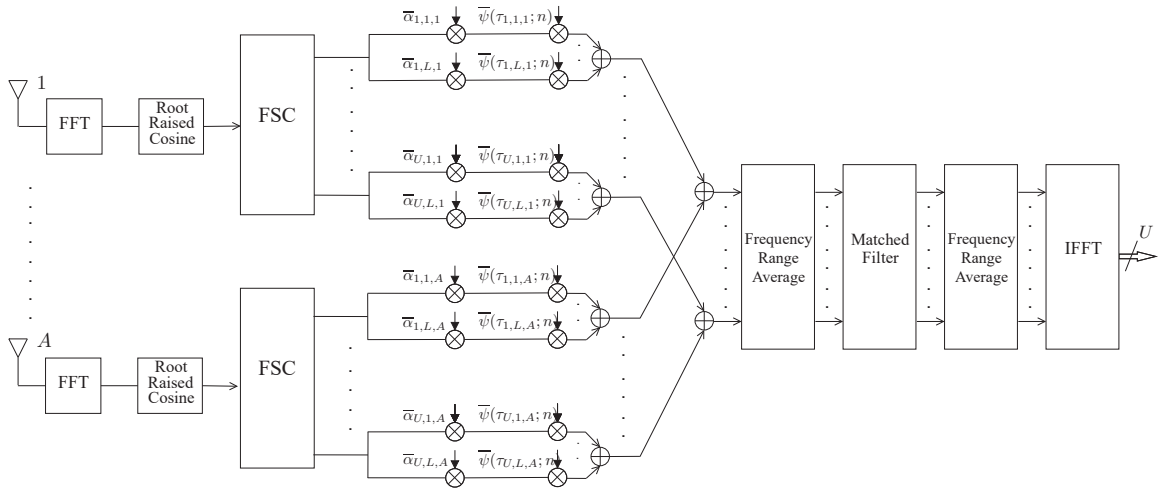

Figure 1. Configuration named *Pre*-FSC of the Detector with FSC

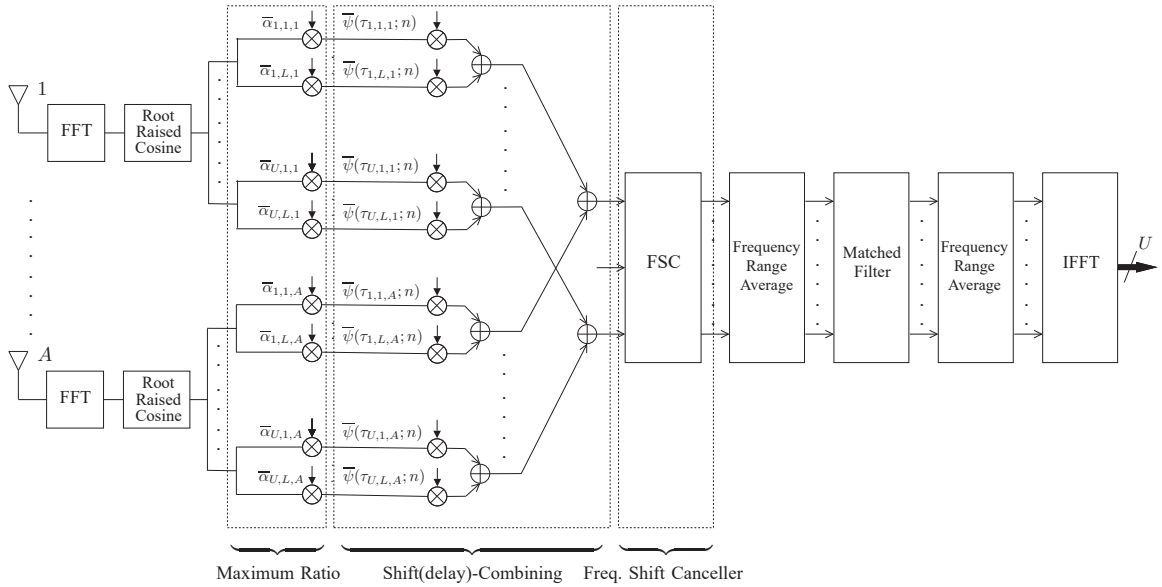

Figure 2. Configuration named *Post*-FSC of the Detector with FSC

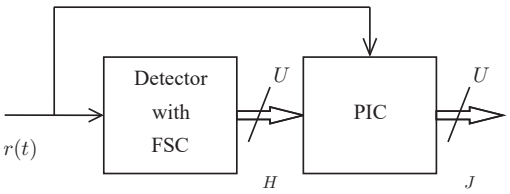

Figure 3. FSC Detector concatenated with PIC

and then the Maximum Ratio-Shift(delay)-Combining. The performance in terms of BER versus Eb/N0 of the proposed Multiuser Detectors was already evaluated in previous work. The configuration of Pre-FSC corresponds to the one with best performance evaluated in [2]–[4] for a multi-antenna scenario. The performance difference, between the two configurations, for a probability of error of 10^{-3} is only about 1dB for 2 antennas, modulation QPSK, chip rate 3.84MChips/s, spatial diversity, spreading factor of 16 [4]. For one antena the Pre-FSC configuration is the one with less complexity but it scales with the number of antennas. In the Post-FSC configuration despite its lower performance in standalone (without the PIC) the

	i7(Single Thread)	Xeon(Single Thread)
1 ant, 16 users, 1HS, Pre, 36MFPI	14.3ms, 2.5GF	18.6ms, 1.9GF
2 ant, 16 users, 1HS, Pre, 66.6MFPI	25.7ms, 2.6GF	34.1ms, 2.0GF
1 ant, 16 users, 1HS, Post, 77.3MFPI	29.6ms, 2.6GF	40.8ms, 1.9GF
2 ant, 16 users, 1HS, Post, 79.7MFPI	31.0ms, 2.6GF	43.0ms, 1.9GF

Table I
PERFORMANCE DATA FOR SERIAL CODE. IT IS ASSUMED THAT THE COMPLEXITY FOR 14HS IS 14X OF THE 1HS. GF - GIGAFLOPS. HS - HALF SLOTS, MFPI - MILLION FLOATING POINT INSTRUCTIONS

complexity do not scales with the number of antennas and it is more suitable for a multi-antenna scenarios.

The configurations of Figure 1, 2 reduce to the RAKE-2D when the FSC operation is removed. The block frequency average corresponds to a downsampling in time domain. The first downsampling factor is equal to the number of samples per chip, and the second downsampling aims to provide one sample per symbol and is thus equal to maximum spreading factor (MSF=16). So the length of the IFFTs performed at the end of the chain are smaller than

Algorithm 1 Multiuser Detector Code Description of Pre-FSC+PIC Configuration

Load *data* (Burst, Channel)
Generate *Signatures Waveforms* in Discrete Fourier domain without Channel impairment
Generate Fast Fourier Transform (FFT) of Root Raised Cosine(RRC) and Raised Cosine
Start statistics, Start counting time
Filter with Root Raised Cosine the input Burst (one for each antenna), It is kept a discrete time domain and a discrete frequency domain copy, BURSTANTn (discrete time domain)
Generate the Noise Power Density at the input of the FSCs (from the estimate of the noise power, number of symbols per user in a half slot and the RRC filter)
Parallel begin nusers
  Generate *Signatures Waveforms* in Discrete Fourier domain with Channel Impairment for the user correspondent to the thread and each antenna
  Barrier
  for nantennas **do**
    Frequency Shift Canceller
    Matching Filter to user channel at the antenna
    Accumulate
  end for
  Downsampling (correspondent operation in Discrete Fourier domain) by the upsampling factor
  Matching Filter to the Spreading Code
  Downsampling (correspondent operation in Discrete Fourier domain) by the Maximum Spreading Factor of the system
  Inverse Fast Fourier Transform
  Symbol Demodulation
  Reconstruction of the user signal with Channel impairment from the bits for each antenna, USERiANTn
  Barrier
  Sum of the all users reconstructed signals for each antenna and with channel impairment, SUMANTn (operation divided by the threads equally). Each thread sum, one subset of the samples, trough the users at each antenna.
  Barrier
  for nantennas **do**
    Cancellation (BURSTANTn-SUMANTn+USERiANTn)
    Matching Filter (discrete time domain operation) to the antenna user channel
    Accumulate
  end for
  Downsampling by the upsampling factor
  Correlation (equivalent to Matching Filter) to the Spreading Code
  Downsampling by the Maximum Spreading Factor of the system
  Symbol Demodulation
Parallel end
Stop statistics, Stop counting time

Algorithm 2 Multiuser Detector Code Description of Post-FSC+PIC Configuration. Only shown the diferences from Algorithm 1

...
...
Parallel begin nusers
  Generate *Signatures Waveforms* in Discrete Fourier domain with Channel Impairment (the channel includes Maximum Ratio Combining before FSC) correspondent to the thread (nusers signatures each thread)
  for nantennas **do**
    Matching Filter to user channel at the antenna
    Accumulate
  end for
  Frequency Shift Canceller
  ...
  ...
Parallel end
...

	i7 (2 Cores)	Xeon (6 Cores)	GeForce 740M (with i7)	K40 (with Xeon)
1 ant, 2 taps, 1HS, Pre, 36MFPI	7.3ms, 4.9GF(2x)	3.0ms, 12.0GF(6.3x)	5.6ms, 6.4GF	2.4ms, 15.0GF
2 ant, 2 taps, 1HS, Pre, 66.6MFPI	13.2ms, 5.0GF(1.9x)	5.1ms, 13.1GF(6.6x)	7.0ms, 9.5GF	2.9ms, 23.0GF
1 ant, 2 taps, 1HS, Post, 77.3MFPI	18ms, 4.3GF(1.7x)	6.5ms, 11.9GF(6.3x)	6.2ms, 12.5GF	2.5ms, 30.9GF
2 ant, 2 taps, 1HS, Post, 79.7MFPI	19ms, 4.2GF(1.6x)	6.8ms, 11.7GF(6.2x)	6.5ms, 12.3GF	2.6ms, 30.7GF
1 ant, 2 taps, 14HS, Pre			48.5ms, 10.4GF	6.1ms, 82.6GF
2 ant, 2 taps, 14HS, Pre			89.3ms, 10.4GF	11.3ms, 82.5GF
1 ant, 2 taps, 14HS, Post			58.2ms, 18.6GF	7.6ms, 142.4GF
2 ant, 2 taps, 14HS, Post			61.5ms, 18.1GF	8.1ms, 137.8GF
2 ant, 2 taps, 14HS, Post, UP=16, 91.0x14MFPI			74.9ms,17.0GF	11.2ms, 113.8GF

Table II
PERFORMANCE DATA. THE K40 GPGPU IS AT BASE FREQUENCY OF 745MHZ. IT IS ASSUMED THAT THE COMPLEXITY FOR 14HS IS 14X OF THE 1HS. IT IS ASSUMED ALSO THAT THE TIME FOR OPENMP FOR 14HS IS 14X OF THE 1HS. GF - GIGAFLOPS. HS - HALF SLOTS, MFPI - MILLION FLOATING POINT INSTRUCTIONS. UP - UPSAMPLE, IF OMITTED IS 4

	K40
2 ant, 2 taps, 14HS, UP=8, 69.5x14MFPI	8.0ms, 121.6GF
2 ant, 3 taps, 14HS, UP=8, 100.8x14MFPI	9.7ms, 145.5GF
2 ant, 2 taps, 14HS, UP=16, 91.0x14MFPI	9.8ms, 130.0GF

Table III
PERFORMANCE DATA FOR THE CONFIGURATION POST-FSC+PIC WITH MORE TAPS AND/OR ANTENNAS AND UPSAMPLE WITH THE HIGHER CLOCK OF K40 (875MHZ). GF - GIGAFLOPS. HS - HALF SLOTS, UP - UPSAMPLE, MFPI - MILLION FLOATING POINT INSTRUCTIONS.

FFT performed at the beginning (1 to MSF x Upsample).

In a case of a standalone implementation, the Frequency Shift Algorithm is well adapted to single user processing (different from jointly detection in MMSE which all users must be detected at same time) because the processing is decoupled from the other users even if it needs to know which users are active. The number of points that a slot has support in frequency domain through a FFT is independent of the upsampling and channel length. For those reasons the algorithm purposed in [2]–[4] does not need large matrices inversions (18x18, 1.28 Mchips, 3.84Mchips/s) with upsampling and channel length.

The proposed implementation is for 1.28MChips/s but it is easily configurable for 3.84MChips/s. Despite the fact, in this work, it is treated the case of 16 users of spreading factor (SF) of 16, the detector supports the mix of other lower spreading factors. For example, one user of spreading factor 4 is treated as 4 users (16/SF=4) of spreading factor of 16 [2]–[4], in both the FSC and PIC. With that, both FSC and PIC are not limited with InterPath Interference (IPI) because for lower spreading factors each symbol are detected independently of neighbors symbols. Also the detector allows a mix of QPSK, 8PSK and 16-QAM modulations.

This detector is valid for Beamforming and for Spatial Diversity if it is given the correspondent channel to do the processing. The frequency offset impairment between the carrier in the transmitter and the reference carrier in the receiver can be compensated after the FSC[1] because each user spread spectrum signal remains cyclostationary with that offset. Because the midamble interval, the bits of each side of the slot must be recovered separately. In the case of joining the sides parts, each user signal looses the cyclostationary.

III. IMPLEMENTATION, COMPLEXITY RESULTS AND PERFORMANCE DISCUSSION

Both configurations were implemented in serial code, OpenMP and CUDA in a computer with a i7-3537U CPU (two cores) and with Geforce 740M GPU (Compute Capability (cc) 3.0), and a computer with Dual-CPU Intel(R) Xeon(R) E5-2640 2.50GHz (6 cores per CPU) with a K40 General Propose GPU (GPGPU) (cc 3.5). In the computer with i7 CPU was used Visual Studio and in computer with the Xeon CPUs was used Eclipse with *gcc*. The CPUs and GPUs are connected through a PCI2 bus.

The serial code was used to measure the complexity in Millions Single Precision Floating Point Operations with PAPI [2] and to take the reference runtime in both machines.

The receiver Pre-FSC+PIC takes about 36 Millions floating point (single precision) operations to process half slot in 1.28MChips/s with 16 users, spreading factor of 16, 1 antenna, 2 taps per user, QPSK modulation and upsampling of four. The block FSC alone takes 76% of those operations not taking in account operations that are done outside the block and that are only needed by it (see Algorithm 1, which is a parallel one for OpenMP). A version in single precision was compared against a double precision one and it was found that the former implementation had less than 1% relative error in the symbols (before quantization in bits) recovered by the FSC and the PIC. Then single precision is enough for the computations.

The receiver Post-FSC+PIC takes about 77.3 Millions floating point (single precision) with the same scenario. The significant increase in complexity is due to the added processing requirements in the generation of the signature waveforms (see Algorithm 2).

First Column of Tables I, II shows the complexity of each configuration with an added scenario of two antennas.

Table I shows the runtime and the performance for a serial code implementation running in one single core and single thread.

OpenMP was used to parallelize the code in the Windows and Linux platforms. In Algorithm 1 is presented the description of Multiuser Detector Code for the configuration Pre-FSC+PIC and in Algorithm 2 is shown the differences from Algorithm 1 for the configuration Post-FSC+PIC. Because the processing for recovering the users in the detector with FSC are (almost) independent it was created a thread for each user. Then, it was created 16 threads. Algorithms 1 and 2 have as references Figures 1 and 2 respectively for the detector with FSC. The PIC as already said have an identical structure to the Pre-FSC but the equivalent operations between the two are made in the time domain instead. As can be seen by Algorithm 1, 2 inside the parallel section of the code there are barriers[3] instructions to synchronize the data between threads. The first barrier, only in Algorithm 1, guarantees that the Channel Impaired Signatures Waveforms of each user (thread) needed by all threads (users) are generated all when needed. The second barrier, in both Algorithms, guarantees that the reconstruction of each spread signal (in each thread) from bits detected from the detector with FSC is completed when needed. This operation is part of the PIC detector. The third and last barrier guarantees the generation of the sum of all reconstruct users when needed.

Each time that the FSC is called 32 (18x18) matrix inversions for 1.28MChips/s are made. For 16 users, 1 antenna, there are 16x32 inversions. For the Pre-FSC+PIC configuration the number of matrix inversions scales linearly with the number of antennas but this number remains constant in the configuration Post-FSC+PIC.

The second and third Column of Table II shows the performances with OpenMP. It is also shown the scalability

[1]The carrier frequency offset can be compensated just after the IFFTs with minor symbol degradation in relation to the case of a chip level compensation. The receiver reference carrier must be the same for all the antennas.

[2]*http://icl.cs.utk.edu/papi/*

[3]Each thread runs the same code with different data. The OpenMP barrier instruction stops each thread in that point until all threads reach it. Then all threads resume.

in relation to the serial code[4]. In the case of Dual-Xeon machine, it was found that the best runtime achieved was with affinity configuration of a single CPU.

The target time to recover both half slots is about $1.4ms$ considering one single carrier with half slots used for the uplink. This target was not achieved in this work but it was adopted an equivalent one that minimizes the latency, principally in the GPUs. Then instead of processing each half slot at the time, it is processed 14 half slots at once in a maximum time of $10ms$[5].

The algorithms were also implemented using CUDA. The implementation was made with 21 kernels for the Pre-FSC+PIC, and 26 kernels for the Post-FSC+PIC, all called only one time, doing the processing for the 16 users times 14 half slots (or only one) at once. It was made only one copy from host (CPU) to device (GPU) with the initial data and one copy from the device to host with the final bits. The data remains all time in GPU external memory between host to device kernels calls. There was not needed any synchronization (like the barriers in OpenMP) between the kernels. That was because there is no CPU code between kernels, and consecutive kernels in the same CUDA stream are serialized. See fourth and fifth Column of Table II for runtime and performance. The runtime include the processing time needed in a realtime implementation like the data transfer between the CPU and GPU and back. The runtime in Pre-FSC+PIC, 1 antenna, is mainly due to the FSC kernel (about 90%). The arrays from FSC kernel are memorized in the GPU main memory. Correspondent cache memory to this main memory is important because the FSC kernel have many no coalescent accesses. It needs more than the maximum 48K primary cache of each Streaming Multiprocessor (SMX) of cache storage, making the FSC kernel memory bounded. This performance was the best performance achieved and it was obtained with a single thread program calling the kernels. The Post-FSC+PIC, despite more complex than Pre-FSC+PIC for 2 antennas, has better performance because, have only half of the FSC runs and so less cache constrains.

Table III shows the performance for the Post-FSC+PIC with the increase of number of taps per user and antenna, and/or the increase of the number of antennas, with upsample of 8 and 16[6] and a higher K40 clock.

IV. Conclusions

The complexity of a Multiuser Detector for uplink was evaluated to investigate the possibility of its deployment in UMTS-TDD Base Stations. The main contributor to execution time in CUDA was lack of cache capacity. A significant performance improvement is expected when new hardware versions incorporating larger caches become available, like the K80 GPGPU. Also a PCI3 connecting CPU to GPU could improve the runtime.

It can be concluded that the K40 GPGPU is a possible solution for a realtime implementation in a single carrier base station, with upsample of four with the Pre-FSC+PIC for a single antenna and the Post-FSC+PIC for two and three antennas with the three antennas detector limited to three taps per user and antenna. All ran in less than $10ms$ for the 14 half slots recovered in uplink (in basestation). It was achieved about 72x speedup in relation to serial code implementation on the a Xeon CPU with the Post-FSC+PIC, 2 antennas, 2 taps and upsample of 4. A greater upsample factor may be needed for better tap delay, chip and symbol resolution. These greater upsample increases the complexity mainly due to the operations in time domain. Also, it was evaluated the Post-FSC+PIC with upsample of 8 and 16 within the same time constrains.

Processing Power for more carriers could be obtained with multiple GPGPU cards in the same computer.

Also it is expected that Xeon processors with more cores (as many as 18, actually) with OpenMP could meet the runtime constrains in the future.

Future work includes to implement these algorithms on FPGA, with OpenCL, in order to achieve better energy efficiency.

[4]The scalability superior to the number of cores can be explained by the CPU Hyperthreading associated with a large amount of outer cache (L3).

[5]Seven slots allocated to uplink in a frame of $10ms$ and 14 slots.

[6]Must be reported that the Post configuration have different implementations between upsample of 4 and upsample of 8 and 16 in order to achieve better runtime. That is why it can noticed incoherency between complexity in MFPIs.

Acknowledgments

Grant from Fundação para a Ciência e Tecnologia (FCT), Portugal supported by POPH/FSE. Support from project UID/CEC/00127/2013 from FCT. Thanks to Prof Paulo Dias and to IrisLab for providing the K40 GPGPU and the testing server respectively both from University of Aveiro.

References

[1] J. G. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill, Inc, Fifth ed., 2007.

[2] L. Gonçalves and A. Gameiro, "Multi-Sensor Frequency Domain Multiple Access Interference Canceller for DS-CDMA Systems," *European Transactions on Telecommunications, John Wiley & Sons, Ltd*, vol. 18, pp. 263–273, April 2007. http://doi.wiley.com/10.1002/ett.1146.

[3] L. Gonçalves and A. Gameiro, "Erratum: Multi-Sensor Frequency Domain Multiple Access Interference Canceller for DS-CDMA Systems," *European Transactions on Telecommunications, John Wiley & Sons, Ltd*, vol. 19, p. 495, June 2008. http://doi.wiley.com/10.1002/ett.1297.

[4] L. Gonçalves, *Detecção Multiutilizador no Domínio da Frequência para Sistemas DS-CDMA*. Ph. D. Thesis, Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Aveiro, Portugal, 2009. http://luisgo.dyndns.org/docu/Phd_Luis_Goncalves.pdf http://ria.ua.pt/bitstream/10773/2225/3/2010000093.pdf.

[5] A. Klein and P. W. Baier, "Linear Unbiased Data Estimation in Mobile Radio Systems Applying CDMA," *IEEE Journal of Selected Areas in Communications*, September 1993. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=233218.

[6] A. Host-Madsen and K.-S. Cho, "MMSE/PIC multiuser detection for DS/CDMA systems with inter- and intra-cell interference," *IEEE Transactions on Communications*, vol. 47, pp. 291–299, February 1999. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=752135.

[7] M. Vollmer, M. Haardt, and J. Götze, "Comparative Study of Joint-Detection Techniques for TD-CDMA Based Mobile Radio Systems," *IEEE Journal on Selected Areas in Communications*, vol. 19, August 2001. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=942509.

[8] E.-L. Kuan and L. Hanzo, "Burst-by-Burst Adaptive Multiuser Detection CDMA: A Framework for Existing and Future Wireless Standards," *Proceedings of the IEEE*, vol. 91, pp. 278–302, February 2003. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1182063.

[9] M. D. McCool, "Scalable Programming Models for Massively Multicore Processors," *Proceedings of IEEE*, vol. 96, pp. 816–831, May 2008. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4490125.