COMPARISON OF THE SOFTWARE DEFINED RADIO IMPLEMENTATIONS OF THE K-BEST LIST SPHERE DETECTION

*Janne Janhunen, *Olli Silvén, Markku Juntti*

Centre for Wireless Communications/*Information Processing Laboratory, University of Oulu
Pentti Kaiteran katu 1, P. O. Box 4500, 90014 Finland
fax: + (358) 8 5532845, email: {janne.janhunen, olli.silven, markku.juntti}@ee.oulu.fi

ABSTRACT

This paper introduces four programmable processor platforms and a multiple-input multiple-output (MIMO) receiver based on the *K*-best list sphere detection algorithm. The four platforms can be considered as current state-of-the-art software defined radio (SDR) devices for wireless communication. The platforms can be categorized as a digital signal processor (DSP), SDR processor, application-specific processor (ASP) and application-specific instruction-set processor (ASIP). The DSP is a popular very long instruction word (VLIW) device (TMS320C6455), the SDR processor employs multi-threading and multiple cores (SB3500 core processor), the ASP is based on transport triggered architecture (TTA), while the ASIP is the SDR processor enhanced with a special instruction-set extension for sorting.

1. INTRODUCTION

A cognitive radio (CR) paradigm for a wireless communications has been thought as a goal, to which a software defined radio (SDR) should evolve. The fundamental idea behind SDR and CR concepts is to support adaptivity, reconfigurability and programmability. Requiring also high performance from the platform, the two concepts set stringent requirements and challenges for the future processor development.

In the third generation partnership project (3GPP) long term evolution (LTE-A) targets [1], 1 Gbps is planned to be transmitted through wireless channel and the requirements for the higher data transmission are constantly increasing. A high-rate wireless communication needs power efficient solutions to process the increasing amounts of data with limited hardware and low power consumption. As the wireless communication standards are continuously evolving, and have several configurations, flexibility is desired from the terminal devices that are expected to be compatible with multiple standards and adaptive to changing operating environment. Consequently, programmable implementations and programmable processors that reach the performance levels required by the wireless communications standards are of great interest. Much of the challenge for the software and hardware implementations is in achieving the necessary computing speed within the power consumption budget of the future mobile devices that also run other applications.

The multiple-input multiple-output (MIMO) antenna transmission is an upcoming technique targeted at wireless communications. A MIMO system can create multiple parallel independent data streams between the transmit and receive antennas and can increase the transmission rate without increasing the spectrum requirement or transmit power. The MIMO antenna system combined with the orthogonal frequency division multiplexing (OFDM) has been included in many wireless standards, such as IEEE 802.11 wireless local area network (WLAN), WiMAX, 3G LTE and LTE-A. The multipath environment causes MIMO channel to be frequency-selective and OFDM can transform such a channel into a set of parallel frequency-flat MIMO channels.

The list sphere detector (LSD) [2] is a variant of the sphere detector that can be used with forward error control (FEC) coding to approximate the soft decision maximum *a posteriori* probability (MAP) detector. Specifically, the so called breadth-first *K*-best LSD is of interest for implementation, because it provides a straightforward implementation and a fixed throughput [3].

In this paper, we introduce four programmable processor based approaches for implementing a *K*-best LSD algorithm to understand their potential and differences in this computationally demanding application. The compared platform are an industry standard digital signal processor (DSP), a software defined radio processor, an application-specific processor (ASP) that employs an application specified data path, and an application-specific instruction-set processor (ASIP) that is actually the used SDR with a special instruction-set extension (ISE) and a respective function unit. We emphasize the required symbol rate in 3G LTE systems, because it sets a strict parallel processing requirements. Table 1 gathers up the platforms and points out their key differences. The effective issue width is a measure of parallelism in architecture, showing that the processors are close to each other in that respect. Typical instruction latency of the processor is presented in clock cycles (cc). Interested readers are referred to read [4] for more detailed description of the *K*-best LSD algorithm and programmable processor comparison. In addition, this paper presents coarse power dissipation values for the DSP and SDR.

Table 1: Compared platforms

	Inst.	Effective issue width	Inst. latency
DSP (TMS320C6455)	VLIW	8	1 (cc)
SDR (SB3500)	compound	12	4 (cc)
ASIP (SB3500+sorter)	extended compound	12	4 (cc)
ASP (TTA)	move	10	1 (cc)

The rest of the paper is organized as follows. Section 2 presents the MIMO data detection problem and introduces the 3G LTE specification requirements, in which this work is

based on. The implementations and results are documented in Section 3. Section 4 discusses about power dissipation issues on software defined radios. Section 5 highlights the most important pros and cons of the programmable implementations. Section 6 concludes the paper.

2. MIMO-OFDM DETECTION PROBLEM

A MIMO–OFDM based multi-antenna system is assumed with M transmit and N receive antennas. The received signal vector on sth subcarrier can be presented as

$$\mathbf{y}_s = \mathbf{H}_s\mathbf{x}_s + \mathbf{n}_s, \quad s = 1,2...,S, \tag{1}$$

where S is the number of subcarriers, $\mathbf{y}_s \in \mathbb{C}^N$, $\mathbf{x}_s \in \mathbb{C}^M$ is the transmitted symbol vector and $\mathbf{n}_s \in \mathbb{C}^N$ is the noise vector. The symbol $\mathbf{H}_s \in \mathbb{C}^{N\times M}$ denotes the channel matrix. The entries of $\mathbf{x}_s$ are chosen independently of each other from a quadrature amplitude modulation (QAM) constellation.

The ML detector minimizes the Euclidean distance between the received signal $\mathbf{y}$ and the lattice points $\mathbf{Hx}$ and selects the lattice point that minimizes the Euclidean distance to the received vector $\mathbf{y}$, i.e.,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathscr{A}^M} \| \mathbf{y} - \mathbf{Hx} \|^2, \tag{2}$$

where $\mathscr{A}$ is the symbol alphabet and $\| \cdot \|^2$ denotes the L_2 norm of a vector. The exhaustive search can be used to solve the ML detection problem. However, it becomes computationally impractical as the number of transmit antennas is increased or a higher order modulation method is used. The sphere detection algorithm solves the ML approximation (2) by limiting the search to the constellation points that lie inside an M-dimensional hyper-sphere [5]. The number of visited nodes in the search tree can be reduced by limiting the search to inside a sphere with radius d using the sphere constraint $d^2 \geq \| \mathbf{y} - \mathbf{Hx} \|^2$.

The description of K-best algorithm is bypassed to save space, but we refer to [6] to get more details on the algorithm. However, it can be briefly stated that the K-best algorithm belongs to the group of tree search algorithms, in which the computational complexity greatly depends on the number of transmit antennas, modulation method and the list size K used in the algorithm. The design parameter K indicates how many partial Euclidean distance (PED) alternatives and corresponding symbols are stored on each level of the tree, and expanded on the next level. Thus, the chosen K has a significant impact on the implementation complexity but also to the frame error rate (FER) performance.

2.1 Implementation Requirements

The real time requirements are based on the 3G LTE specification. A single OFDM symbol on five MHz bandwidth consists of 512 (300 used) individual symbol vectors, each mapped to a single OFDM subcarrier. Seven OFDM symbols should be processed in 0.5 ms, and, thus, symbol duration of 71.42 μs is required. In other words, 300 symbol vectors should be processed in 71.42 μs.

The list size and fixed-point implementation word lengths were decided based on the computer simulation results. Figure 1 presents FER results of several detectors: the K-best LSD with three different list sizes, the maximum likelihood detector, the maximum a posteriori detector and the linear MMSE detector. A rather demanding high correlation typical urban channel is assumed in the simulations. A soft-output K-best detector outperforms the linear MMSE detector and the hard decision ML detector. Furthermore, the K-best detector provides a sufficient decoded performance and an implementable computational complexity for a MIMO–OFDM system. The figure indicates that the list size K=16 provides a good trade-off between the performance and the computational complexity. A smaller list size reduces the complexity at the prize of degraded decoding performance.

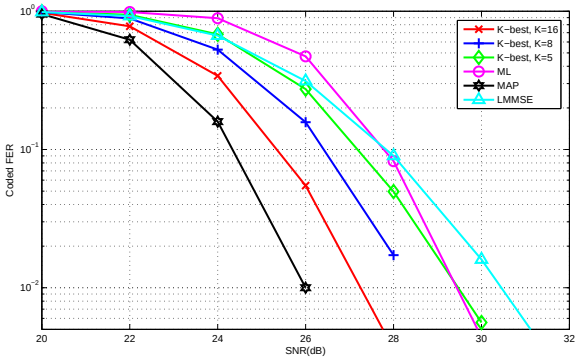

Figure 1: FER comparison for real-valued 2×2 64-QAM system in typical urban channel with 16-bit fixed-point arithmetic.

The simulator takes into account the effect of log-likelihood ratio (LLR) clipping [7] with threshold $L_{\max} = 8$. The LSD output list is used to calculate the approximation of the probability LLR of each transmitted bit. By limiting the dynamic range of the LLR, the required list size can be decreased reducing the computational complexity at the same time.

A 16-bit fixed-point arithmetic (6-bit integer, 10-bit fraction) was chosen leading to very minor performance degradation compared to the floating point performance. The performance loss with 12-bit (5-bit integer, 7-bit fraction) fixed-point word is already significant. This indicates that no benefit can be achieved with shorter word length processor, but a standard 16-bit word length is very satisfactory.

3. IMPLEMENTATIONS OF LSD ALGORITHM

Fair comparison between implementations of an algorithm on different platforms requires certain parameters to be fixed. On the application side, such parameters are the number of antennas, modulation method, word lengths, list size and system model. On the platform side, the implementations should exploit the full potential of the resources and tools of each architecture. Table 2 lists the key parameters of the application and implementations. The use of standard 16-bit word length does not favor any particular programmable processor in our comparisons, so the differences arising can be traced to architectural features. Since the focus of this study is on SDR platforms, the programmability is an important feature of the compared processors.

3.1 VLIW DSP implementation

Texas Instruments TMS320C6455 is a high performance embedded processor, which has been used as a benchmark

Table 2: Implementation properties

Signal model	real-valued
Antenna configuration	2×2
Arithmetic	16-bit (6-bit int., 10-bit frac.)
Modulation method	64-QAM
Detector	K-best LSD
List size (K)	8 and 16

for the DSP implementation of the K-best LSD algorithm. C6455 is a VLIW processor, which operates on 1200 MHz clock frequency. The processor provides for a performance of 22.5 Mbps (list size K=16) for the PED calculation and control. However, with a software sorter and a decreased list size the computation of a single symbol vector takes 8109 clock cycles (K=8), and, thus, the processor provides for a detection rate less than 1.8 Mbps. The reason for high latency is that the processor does not offer special functions to support the sorting.

The slow sorter is a significant drag on the algorithm execution, because the sorter is called after a newly computed PED. The software sorter turned out to be inefficient in earlier studies [8], but even the improved and simplified software sorter does not bring a solution to the sorting problem. In [9], hardware accelerated co-processor is implemented for sorting besides the uniprocessor. A similar idea of co-processor is required for the DSP implementation to meet the real time requirements. C6455 results are included in Table 3 without including the latency caused by the sorter.

Table 3: PED calculation and control code on DSP with a list size $K = 16$

Level	# of PEDs	# of Cycles
1st	8	63
2nd	64	130
3rd	128	214
4th	128	231
tot.	328	638

The DSP is running on 1200 MHz clock frequency. The processor does not support a parallel processing of the symbol vectors, and, thus, symbol vectors must be processed in serial. To meet the symbol rate required in 3G LTE system, a single symbol vector must be processed in 0.23 μs, which means 285 clock cycles. If symbol rate requirements (285 cc) are compared to achieved DSP results (638 cc), it can be calculated that even the PED calculation and control should be over two times faster. Furthermore, the processor requires a co-processor to do sorting in parallel with PED calculation.

3.2 SDR Implementation

The SB3500 is a multi-core device that features include compound instructions, single input multiple data (SIMD) vectorization units and hardware support for multiple threads. The SB3500 architecture has three cores each having four hardware threads. All the hardware threads can operate simultaneously, and, thus, multiple concurrent program execution is supported. This feature provides an advantage in implementations, where parallelism is required, e.g., processing multiple parallel subcarriers in the MIMO–OFDM receiver implementation. All threads can simultaneously execute instructions, but only one thread may issue an instruction on a cycle boundary [10].

The list sizes were set to $K = 8$ and $K = 16$. In 64-QAM with a real-valued signal model there are $\sqrt{64} = 8$ QAM symbols, which have a straightforward impact on an efficient vectorization on SB3500 processor. The short data type (16-bit) is used for both PED and the corresponding symbol identifier. Vectorization is disabled if conditional statements or function calls are inside the loop. Our implementation includes four search levels due to two transmit antennas, 64-QAM and real-valued signal model. A single C programmed function was implemented for each level.

The PED computation and control is efficiently implemented on SDR as can be seen in the next section. However, when the software sorting function is included to the data flow, the vectorized PED calculation is broken, mostly because of the sorter function calls. Table 4 summarizes the latencies for levels and sorting. There are less sorter calls than calculated PEDs, because the radius d is limited to discard the most unlikely paths in the search tree. The radius is selected so that there is no impact on the decoding performance.

Table 4: SDR latencies including software sorter with a list size $K = 8$

Level	# of PEDs	# of Cycles
1st	8	55
2nd	64	1396
3rd	64	1485
4th	64	1372
tot. PED	200	4308
Sorter	(114)	2016
total		6324

3.3 ASIP Implementation

ASIP is an implementation, where SB3500 core processor is enhanced with an instruction-set extension and a respective function unit. The SB3500 core hides four processor cycles into a single thread cycle. Thus, a fairly high latency instruction-set extension sorter could be implemented [11]. For example, an ISE sorter with the latency of a single thread cycle could be practical to implement without violating the data dependency principles of the architecture.

Table 5 presents the latencies of the distance calculations in all four levels, when list sizes $K = 16$ and $K = 8$ are used. Totally, 328 PEDs are calculated in 674 cycles. The PED calculation with a single hardware thread, hence, reaches a performance of 2.67 Mbps. If all the 12 hardware threads of the ASIP are allocated for the PED calculation, detection rate of 32.0 Mbps is achieved. Respectively, a decoding rate of 3.4 Mbps is achieved on a pure SDR platform. The decrease of the throughput is almost a 10-fold.

Based on the information from the manufacturer, the SB3500 instruction-set can be expanded [11]. Most of the

Table 5: PED calculation and control code on ASIP with list sizes $K = 16$ and $K = 8$

Level	PED($K = 16$)	Cycles	PED($K = 8$)	Cycles
1st	8	40	8	39
2nd	64	129	64	130
3rd	128	243	64	133
4th	128	262	64	142
tot.	328	674	200	444

instructions have four pipeline stages. It is sometimes suitable to introduce multiple new operations to implement a particular algorithm and avoid breaking the four stage pipeline of the processor. The multithreaded pipeline hides latencies and allows fairly complex extensions to be designed, such as a sorter. Because of the strict real time requirements of the K-best algorithm, a single thread cycle sorter, which can be used in parallel with the PED calculation, is required.

If we assume the ASIP, which has 12 hardware threads each running on 150 MHz clock frequency, we can roughly calculate the real time requirements for the list sphere detector. 12 symbol vectors can be processed in parallel and 25 sequential calculations are required, which means that a single thread can spend approximately 2.85 μs to calculate one symbol vector. With 150 MHz clock frequency, this means 427 cycles per symbol vector. Comparing the calculated and achieved results in the ASIP implementation, a required symbol rate is not far from the achieved result.

3.4 ASP Implementation

The same parameters were chosen for the ASP implementation [12] to guarantee the latency and throughput comparison between the other platforms. TTA resembles a VLIW architecture, in which the operation latencies are visible to the programmer. The function units are connected with transport paths and sockets [13]. In the transport triggered architecture, the operations are consequences of data transports, and the processor designer has the freedom to build the optimal data transmission by adding enough paths between the logic and memory.

The current ASP implementation [12] achieves the decoding rate of 7.6 Mbps with hardware complexity of 25 kGE (gate equivalents) including a single cycle hardware sorter. The processor operates on 280 MHz processor clock frequency. The starting point for the study has been a small footprint processor so that multiple cores can concurrently process several parallel subcarriers. A single symbol vector is processed in 441 clock cycles. When the list size is decreased to $K = 8$, a single symbol vector is processed in 261 clock cycles, which leads to a decoding rate of 12.8 Mbps. 4-fold parallelism in processor would process a single symbol vector in 96 clock cycles ($K = 8$), which leads to 35 Mbps decoding rate .

As the ASP has 280 MHz clock rate, a single cycle instruction takes 3.57ns. According to this assumptions, there are 19,997 clock cycles time to process 300 symbol vectors, which means that each symbol vector has to be processed in 66 clock cycles. Thus, at least seven ASP processors are required to achieve the symbol rate requirements. This would lead to an area complexity of 175 kGE, which is still considerably low for a programmable processor.

The implementation includes general-purpose function units, which makes it more interesting compared to the previous ASP implementation [14]. With a limited number of sources, the implementation is simple and the gate count is low. The more precise function unit description is given in [12].

4. POWER DISSIPATION

The software defined radio introduces new challenges for processor architecture designers. The platform should provide for supercomputer performance, while the power dissipation should be suitable for mobile device. In general, the maximum power dissipation for mobile device is limited to 3W, since consumption over the limit requires cooling for the device. In mobile devices the typical idle time is long and the power dissipation should be minimized in that state. However, the peak power dissipations might rise very high in applications, in which the computing complexity is dense.

Texas Instruments is providing a power dissipation estimate for C6455 processor on typical activity [15]. Based on the documentation, the typical activity of C6455 assumes 60 percent of CPU usage and 50 percent DDR2 usage on 250 MHz. The total power dissipation is 2.30W (internal logic 1.76W and IO 0.54W), when 1.25V core voltage and 1200 MHz CPU frequency is assumed. Thus, C6455 processor is more suitable for base station use rather than mobile devices.

The typical power dissipation for a single 500 MHz SBX core is 100 mW [16]. Thus, it can be estimated that SB3500 platform with three 500 MHz cores consumes approximately 300 mW and the platform with 600 MHz cores a bit more. Although, the battery of SB3500 software defined radio (e.g. 1200 mA) will not last more than four hours with a continuous typical operating dissipation, it is much better than the corresponding half an hour operating time with the C6455 processor.

5. DISCUSSION

Table 6 summarizes the results for four different programmable processor. The table presents the achieved and required latency in clock cycles for a single symbol vector calculation. The last column presents the required speed-up for the current implementations in order to achieve the symbol rate specified in 3G LTE requirements. The inefficient software sorter degrade the performance of the DSP and SDR implementations. However, with a hardware accelerated sorter, the ASIP and ASP implementations achieve a very promising results. The ASP includes already a single cycle sorter. Thus, it can be assumed that a single thread cycle (four core cycles) hardware sorter is possible to implement for the ASIP. Furthermore, the small cap between the achieved and required latency in the ASIP implementation is most probably attainable by improving the C compilation.

Although the compared platforms belong to the category of programmable processors, the actual programming of particular processor differs significantly. Because TTA based ASP has visible function unit latencies, the programmer has the responsibility to schedule the instructions such that the result is read on time. This leads to the situation, where the program has to be rewritten if the latency increases even in a single function unit. In a conventional processor, where the function unit latency is invisible to the programmer, there

Table 6: Implementation summary and requirements for real time performance for a single symbol vector calculation

	Achieved latency (cc)	Required latency (cc)	Required speed-up
DSP	8109 ($K = 8$)	286	28.4
SDR	6324 ($K = 8$)	427	14.8
ASIP	674 ($K = 16$)	427	1.6
ASP	441 ($K = 16$)	66	6.7

is a hardware support to lock up the processor in case the result is read too early. Also programming in TTA assembly the scheduling of the data transports might become laborious, specifically with complex algorithms. The ASP processor is programmable for smaller list sizes and lower-order QAM constellations for the *K*-best algorithm, but the general-purpose function units can be programmed also for other tasks.

Since the SDR concept assumes adaptivity and programmability, functionally fixed hardware implementations are in general out of interest. Thus, the programmable processors are more interesting also from the cognitive radio perspective. While current SDR processors can be programmed to execute all kinds of algorithms required in signal processing, the fundamental challenge is to achieve a sufficient performance and a low power dissipation. In addition, the processor comparison based on the million operations per second (MOPS) or million multiply accumulate per second (MMACS) is not valid alone, since for instance a fast memory access is critical in many applications.

6. CONCLUSIONS

We implemented the *K*-best LSD algorithm on four programmable platforms having eye on the software defined radio concept, which is a current trend in the wireless communication research. The implemented algorithm requires a high computing power and an efficient sorting operation, which gives an interesting benchmark results from the perspective of a software defined radio. The 3G LTE standard defines very high symbol rate requirements, which are challenging to achieve with programmable processors. The symbol rate requirements lead to the fact that platform has to support multiple subcarrier processing in parallel.

The ASIP implementation achieves a promising throughput and the symbol rate of the current implementation is very close to the 3G LTE specifications. Furthermore, the C compilation can be most probably improved with hand coded assembly, such that the symbol rate requirements are achieved. The ASP implementation benefits from the design freedom and achieves promising throughput results with a low area complexity. To achieve the required symbol rate seven low complexity ASPs should process in parallel. The drawback of the assembly programmed ASP implementation based on the TTA is that the implementation phase is rather laborious and resembles more an application-specific integrated circuit (ASIC) designing than programming a DSP.

REFERENCES

[1] 3rd Generation Partnership Project (3GPP), "http://www.3gpp.org."

[2] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[3] K. Wong, C. Tsui, R. K. Cheng, and W. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, Scottsdale, AZ, May 26-29 2002, pp. 273–276.

[4] J. Janhunen, O. O. Silvén, and M. Juntti, "Programmable Processor Implementations of *K*-Best List Sphere Detector for MIMO Receiver," *Signal Processing, Elsevier, submitted*, Feb. 2009.

[5] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum–likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[6] M. Myllylä, P. Silvola, M. Juntti, and J. R. Cavallaro, "Comparison of two novel list sphere detector algorithms for MIMO-OFDM systems," in *Proc. IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Helsinki, Finland, Sep. 11-14 2006, pp. 12–16.

[7] M. Myllylä, J. Antikainen, M. Juntti, and J. Cavallaro, "The effect of LLR clipping to the complexity of list sphere detector algorithms," in *Proc. Annual Asilomar Conf. Signals, Syst., Comp.*, Pacific Grove, USA, Nov. 4-7 2007, pp. 1559–1563.

[8] J. Janhunen, O. Silvén, M. Myllylä, and M. Juntti, "A DSP implementation of a *K*-best list sphere detector algorithm," in *Proc. Finnish Sign. Proc. Symp.*, Oulu, Finland, Aug. 30 2007, p. 6.

[9] M. Bednara, O. Beyer, J. Teich, and R. Wanka, "Tradeoff analysis and architecture design of a hybrid Hardware/Software sorter," in *Proc. IEEE Application-Specific Systems, Architectures, and Processors*, vol. 1, Paderborn, Germany, Jul. 10–12 2000, pp. 299–308.

[10] J. Glossner, M. Moudgill, D. Iancu, G. Nacer, S. Jinturkar, S. Stanley, M. Samori, T. Raja, and M. Schulte, "The Sandbridge Sandblaster convergence platform," 2005, http://www.sandbridgetech.com/documents/sandbridge_white_paper_2005.pdf.

[11] S. Mamidi, E. Blem, M. Schulte, J. Glossner, D. Iancu, A. Iancu, M. Moudgill, and S. Jinturkar, "Instruction set extensions for software defined radio on a multithreaded processor," in *Proc. Compilers, Architecture and Synthesis for Embedded Systems*, San Francisco, USA, 2005, pp. 266–273.

[12] J. Antikainen, P. Salmela, O. Silvén, M. Juntti, J. Takala, and M. Myllylä, "Fine-grained application-specific instruction set processor design for the *K*-best list sphere detector algorithm," in *International Symposium on Systems, Architectures, Modeling and Simulation (SAMOS)*, Samos, Greece, Jul. 21–24 2008, pp. 108–115.

[13] H. Corporaal, *Microprocessor Architectures: from VLIW to TTA*. John Wiley & Sons Ltd., 1998.

[14] J. Antikainen, P. Salmela, O. Silvén, M. Juntti, J. Takala, and M. Myllylä, "Transport triggered architecture implementation of list sphere detector," in *Proc. Finnish Sign. Proc. Symp.*, Oulu, Finland, Aug. 30 2007.

[15] G. Martinez, "TMS320C6455/C6454 Power Consumption Summary," Texas Instruments, Dallas, Texas, Tech. Rept. SPRAAE8B, 2007.

[16] Sandbridge Technologies, "Sandbridge Announces Certified BDTI Communications Benchmark (OFDM) Results for its Sandblaster SB3500," Sandbridge Tecnologies, Tech. Rep., 2009.