

PROGRAMMABLE IMPLEMENTATIONS OF MIMO-OFDM DETECTORS: DESIGN, BENCHMARKING AND COMPARISON

*Janne Janhunen, *Teemu Pitkänen, †Olli Silvén, Markku Juntti*

Centre for Wireless Communications, University of Oulu

*Department of Computer Systems, Tampere University of Technology

†Computer Science and Engineering Laboratory, University of Oulu

ABSTRACT

Programmable MIMO-OFDM detector design, benchmarking and comparison of implementations have been considered in this paper. We emphasize the significance of co-optimizing the algorithm, software and hardware together in order to reach demanding energy, latency and area restrictions introduced in current standards. We compare energy consumption of the detection algorithms based on the theoretical complexities in function of signal-to-noise ratio. Applying co-optimizing we show how a carefully designed programmable architecture can achieve the 3G long term evolution (LTE) detection rate requirements with a reasonable energy consumption.

1. INTRODUCTION

Wireless communication systems have experienced tremendous development during the past two decades. Multiple-input multiple-output (MIMO) antenna systems combined with an orthogonal frequency division multiplexing (OFDM) have been introduced to wireless communication systems to respond to capacity and transmission reliability requirements. Due to the plurality of wireless communication standards, extreme flexibility is required from an ideal terminal. Thus, there is an open market for software defined radios (SDR).

We compare programmable implementations of the MIMO detectors with each other and to application-specific integrated circuit (ASIC) implementations. In a programmable design, we emphasize the significance of co-optimizing the algorithm, software and hardware together in order to reach strict energy, latency and area restrictions. In this paper, we concentrate on few different detector algorithms. In a programmable system design, the selection of the algorithm is more flexible than in the ASIC design because the programmable platform can be designed to support in addition to changing parameters also several different algorithms. In a software implementation, one key decision is whether to use a high or low level language, which often causes a tradeoff between implementation effort and resource utilization efficiency. The selection of the number arithmetic has significant impact on both software and hardware design. In general, a floating-point compilation is more efficient than a corresponding fixed-point compilation due to applied intrinsics and shift operations in the fixed-point programming. The effect of the number arithmetic to a hardware complexity is discussed later in the paper.

The term goodput [1] provides a solid basis for a systematic complexity-performance tradeoff for detectors in the evolving next generation cellular systems. Traditionally, the communication system performance is characterized by the frame error rate (FER), which can be used to determine system performance in terms of throughput. The transmission throughput is defined to be equal to

the nominal information transmission rate of information bits times $(1 - \text{FER})$. On the other hand, the hardware sets limits to detection rate of information bits. The goodput is a measure, which combines both the detection reliability and hardware limitation, i.e.,

$$\text{goodput} = \min\{\text{throughput}, \text{detection rate}\}. \quad (1)$$

In order to express the hardware complexity, we have applied a gate equivalent (GE), which is defined as a technology-independent measure corresponding a two-input NAND gate in CMOS technology.

The rest of the paper is organized as follows. Section 2 summarizes implementation aspects of the processor design. Section 3 presents the LTE requirements and discusses how the targets are reached in this study. Section 4 compares results to other programmable implementation and application-specific integrated circuit designs presented in the literature. Finally, in Section 5 we conclude the contributions of the study.

2. IMPLEMENTATION ASPECTS

The study of programmable implementations is motivated by the reduced system development time and the possibility to reuse existing platforms instead of doing expensive ASIC design. Another significant motivation for the large systems is the possibility to reduce high leakage power typical to modern CMOS technologies by reusing the same hardware. In addition, a programmable communication system can adapt to a changing channel condition, and thus, save energy per decoded bit. Hence, during a good channel realization a less complex detector can provide higher goodput with less energy, whereas in worse channels more sophisticated and complex detectors are required in order to enable sufficient goodput. Such solutions are not necessary reasonable in ASIC systems, in which the systems have traditionally designed based on the worst case scenario. Obviously, a hardware implementation can be reconfigurable, but in such designs the amount of control logic starts rapidly to increase decreasing at the same time the power efficiency of the circuit. The advantage of the transport triggered architecture (TTA) as a programmable architecture is its low control overhead and modularity from minimal size processors to highly parallel processors [2]. We designed a processor and programmed an SSFE [3] detector algorithm to give a concrete example how a co-optimized design of the programmable processor can achieve a high performance.

2.1. Algorithm

Fig. 1 presents an example how the energy consumption of the different detectors change based on the channel quality. The energy consumption in the algorithms is based on the number of ex-

ected operations and their energy consumption weights. The applied energy models for the operations are presented in [4]. The detector parameters have been fixed based on the signal-to-noise ratio (SNR) such that the detection throughput stays constant. The original LORD [5] algorithm with a constant computational complexity is not feasible in energy wise when a 4×4 antenna system and a high order modulation is assumed. The SSFE algorithm has lower energy consumption than the K -best [6] algorithm. However, taking into account the log-likelihood ratio (LLR) computation complexity, the SSFE algorithm is not feasible until the level update vector is decreased to $\mathbf{m} = [1, 1, 1, 1, 2, 2, 2, 3]$ or $\mathbf{m} = [1, 1, 1, 1, 1, 2, 2, 3]$ near 29 dB. The latter one leads to a list size of 12 elements, which provides a low complexity implementation. The energy consumption of the detectors and the bounds, in which the detector is reasonable to change depends on the channel correlation level and system parameters such as the number of antennas and modulation. A linear minimum mean square error (LMMSE) equalizer has not been considered due to poor performance in a correlated channel.

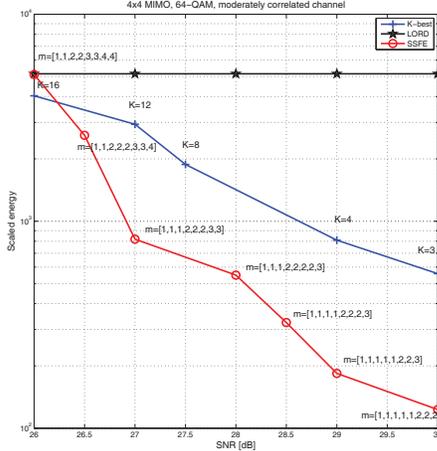


Fig. 1. The energy consumption of the detectors with different parameters. The throughput is constant.

The low complexity detectors such as LMMSE and SSFE provide a high goodput during a good channel realization. Then again, the more complex K -best detector achieves a lower decoding rate due to limited computational resources, but enables still a better goodput than the LMMSE or SSFE in highly correlated channels.

2.2. Hardware

We designed a programmable processor based on the transport triggered architecture [2]. The instruction-set of the architecture includes typical function units (FUs) and special slicer FUs in order to accelerate the SSFE detection. We present two processors, the first one supporting a 12-bit floating-point arithmetic and the second one supporting a 16-bit fixed-point arithmetic. The both arithmetics provide the same bit error rate for the detection. A register-transfer level (RTL) synthesis has been done for the processors with a low-power 130 nm CMOS technology. The corresponding high-throughput technology would increase the clock frequency 1.5–2-fold, but at the same time the power consumption would have a relatively higher increase. The number of parallel function units and their area complexities have been summarized in Table 1. LSU refers to load/store unit between memory and core and RF is an abbrevi-

ation for register file. The processor architecture is presented more detailed in [7].

Table 1. FUs included in the processors and GEs per FU

| FU (latency in clock cycles) | # of FUs | 12-bit FP (200 MHz) | 16-bit FX (200 MHz) |
|------------------------------|----------|---------------------|---------------------|
| ADD/SUB (1 cc) | 8 | 1260 | 520 |
| SLICER (1 cc) | 6 | 500 | 600 |
| MUL (2 cc) | 9 | 930 | 1450 |
| LSU (3/1 cc) | 2 | 380 | 410 |
| RF (1 cc) | 8 | 1190 | 1420 |

The processor complexities have been summarized in Table 2. The results show that the cores synthesized with 200 MHz are almost equal in size. Thus, there is no significant difference in silicon complexity caused by the number arithmetic when optimized word lengths are applied in the design. However, the fixed-point processor achieves a higher clock frequency due to critical path of the single cycle floating-point adder. When approaching the technology limit, the floating-point arithmetic requires larger buffers in the design, which rapidly increases the core size. In general, the total area of 12-bit floating-point and 16-bit fixed-point processors are relative small, which enables feasible multi-core system.

Table 2. Processor complexities represented in GEs

| Processor | GEs |
|---------------------|--------|
| 12-bit FP (200 MHz) | 65 550 |
| 16-bit FX (200 MHz) | 65 630 |
| 12-bit FP (217 MHz) | 70 810 |
| 16-bit FX (277 MHz) | 70 730 |

We summarize energy dissipations for the processors in Table 3. The global operating voltage for the processors is 1.5 V. The energy dissipation analysis takes into account the execution latency and provides a literature comparison between implementations for consumed energy per received bit. In real-valued 16-QAM and 64-QAM systems, the symbols are represented with four and six bits, respectively. Thus, in a 2×2 antenna system with 16-QAM eight bits and in a 4×4 antenna system with 64-QAM 24 bits are received per symbol vector. The energy dissipation is defined as,

$$E = Pt, \quad (2)$$

where P is power and t is the latency of the algorithm execution.

The 12-bit floating-point processor (200 MHz) consumes 36.80 mW in the 2×2 antenna system and 43.10 mW in the 4×4 antenna system. 17 MHz addition to the clock frequency in the floating-point processor increases the processor power consumption approximately 18 percents. Partly, this is caused by a technology limitation, which requires larger buffers in the interconnection network when clock timing is very close to the limit. Respectively, the 16-bit fixed-point processor (277 MHz) consumes 55.50 mW in the 2×2 antenna system and 64.00 mW in the 4×4 antenna system. For the fixed-point processor, the higher clock frequency is easier to justify since the increased power consumption is in line with the performance gain. The consumed energies per received bit are between 0.89–1.44 nJ in all processors and detector configurations. In addition, the energy per operation (op) is approximately 18 pJ/op, which is

very low for a programmable architecture, cf. [8]. A programmable implementation of the K -best algorithm has been proposed in [9].

Table 3. Processor energy dissipations

| Processor | 12-bit FP | 16-bit FX | 12-bit FP | 16-bit FX |
|---------------------|-----------|-----------|-----------|-----------|
| | 200 MHz | 200 MHz | 217 MHz | 277 MHz |
| 2×2 system | | | | |
| Total energy (nJ) | 8.28 | 8.60 | 8.61 | 11.49 |
| Energy (nJ/bit) | 1.04 | 1.08 | 1.08 | 1.44 |
| 4×4 system | | | | |
| Total energy (nJ) | 21.33 | 21.58 | 21.33 | 22.81 |
| Energy (nJ/bit) | 0.89 | 0.90 | 0.89 | 0.95 |

2.3. Software

Applying high level languages support a fast development cycle of new applications. However, compilation from a high level language can cause a significant tradeoff between the performance and programming effort. In [10], we programmed a similar SSFE algorithm with C language, and in this work, we have been able to reduce the latency up to 40 percent mostly by applying assembly programming. On the other hand, the effort of programming in assembly is much higher and the code legacy is worse. In the previous work, we noticed that compiling fixed-point programs cause significant overhead due to scaling of fixed-point values. In general, intrinsics are another reason for overhead in fixed-point compilations.

We assembly programmed each processor to execute an SSFE detector in the 2×2 antenna system with the 16-QAM and the 4×4 antenna system with the 64-QAM. The level update vectors $\mathbf{m} = [1, 2, 2, 3]$ and $\mathbf{m} = [1, 1, 1, 1, 1, 2, 2, 2]$ have been applied. The processor executes a 2×2 SSFE algorithm, i.e., decodes a symbol vector in 45 clock cycles. By pipelining symbol vector detections, 5 clock cycles more can be gained. With 200 MHz clock frequency without pipelining it corresponds a decoding rate of 35.5 Mbps, with 217 MHz 38.5 Mbps and with 277 MHz 49.2 Mbps. By pipelining the symbol vector detection, the corresponding results are 40 Mbps, 43.4 Mbps and 55.4 Mbps. The detection of the symbol vector in the 4×4 antenna system takes 99 clock cycles. The 200 MHz processor achieves a decoding rate of 48.5 Mbps, 217 MHz processor 52.6 Mbps and 277 MHz processor 67.1 Mbps. By pipelining the symbol vector detection, the corresponding results are 51.0 Mbps, 55.4 Mbps and 70.4 Mbps.

3. REACHING LTE TARGETS

We simulated the SSFE detector in a correlated, moderately correlated and uncorrelated channels with parameters compliant to the 3GPP vehicular A (3GPP-VA) specifications defined by International Telecommunication Union (ITU). In long term evolution (LTE), an adaptive transmission is part of the standard. Thus, when spatial multiplexing is assumed, more antennas and higher modulation order can be utilized during a good channel realization than in a highly correlated channel. We make an assumption that the SSFE detector with $\mathbf{m} = [1, 2, 2, 3]$ (a 2×2 antenna system with a 16-QAM) is used in a highly correlated channel. The SSFE detector with $\mathbf{m} = [1, 1, 1, 1, 1, 2, 2, 2]$ (a 4×4 antenna system with a 64-QAM) is used when the channel realization is good.

Fig. 2 illustrates the achievable goodputs for a 5 MHz bandwidth in a two transmit antenna system with a 16-QAM. The required detection rate is 34 Mbps, which is reached by each core of the imple-

mented TTA processors. At high SNR with the 4/5 code rate, the maximum achievable goodput is 27 Mbps. However, when the SNR is between 10–20 dB, a lower code rate enables a better goodput.

Fig. 3 illustrates that a feasible SSFE goodput for the 4×4 antenna system requires a good channel. Thus, we assume that spatial multiplexing with four transmit antennas and a 64-QAM is only used in the uncorrelated or moderately correlated channel. The required detection rate for a 10 MHz bandwidth in the LTE system is 204 Mbps. Thus, four 200MHz, or three 277 MHz cores are required to enable a sufficient decoding rate. Over 160 Mbps goodput is achieved only in the uncorrelated channel realization and with the very high SNR. The half code rate is suitable in the uncorrelated channel when the SNR is between 13–25. In the moderately correlated channel, only the half code rate enables reasonable goodputs at the high end of the feasible SNR range.

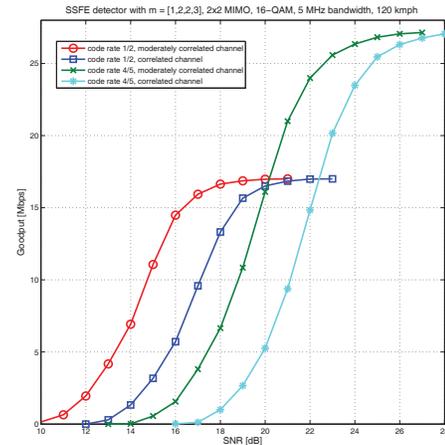


Fig. 2. Goodput results for a 2×2 antenna system with a 16-QAM in the correlated and moderately correlated channels.

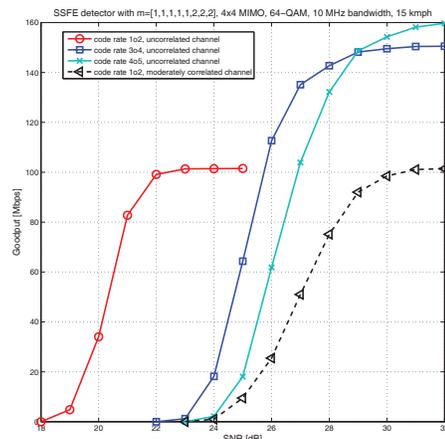


Fig. 3. Goodput results for a 4×4 antenna system with a 64-QAM in the uncorrelated and moderately correlated channels.

4. COMPARISON

We compare application-specific integrated circuit implementations of the K -best and SSFE algorithms to our TTA implementation. The required hardware decoding rate for a 2×2 antenna system with a

Table 4. Energy dissipations comparison

| | [1] | [11] | 16-bit FX (21 cores)[9] | 12-bit FP (4 cores) | 16-bit FX (3 cores) |
|-------------------------------|--------------------|--------------|-------------------------|---------------------|---------------------|
| Platform | ASIC | ASIC | TTA | TTA | TTA |
| Detector | K -best, $K = 8$ | SSFE | K -best | SSFE | SSFE |
| Antenna configuration | 2×2 | 2×2 | 2×2 | 2×2 | 2×2 |
| Modulation | 16-QAM | 16-QAM | 64-QAM | 16-QAM | 16-QAM |
| Clk. freq. (MHz) | 140 | 35 | 250 | 200 | 277 |
| Decoding rate (Mbps) | 140 | 210 | 142 | 142 | 147 |
| Area (kGE) | 110 (180 nm) | 66 (180 nm) | 496 (130 nm) | 262 (130 nm) | 212 (130 nm) |
| Power (mW) | 120 | 23 | 433 | 147 | 167 |
| Energy (nJ/bit) | 0.86 | 0.25 | 3.06 | 1.04 | 1.44 |
| Scaled energy 130 nm (nJ/bit) | 0.57 | 0.16 | 3.06 | 1.04 | 1.44 |

16-QAM and 20 MHz bandwidth is 140 Mbps. The required detection rate can be achieved by four 200 MHz or three 277 MHz TTA processors.

Table 4 summarizes the comparison. The comparison is limited to the implementations, which are applying the same system design assumptions than used in this work. ASIC implementations [1, 11] are optimized to execute detection with parameters presented in table. Adding more flexibility to an application-specific circuit requires a significant increase of silicon area both in detection and control logic. For instance, detectors supporting 4×4 antenna system are reported to consume up to 209 kGE and 290 mW in [1], and respectively, 254 kGE and 200 mW in [11]. On the other hand, the core of the implemented TTA processor is rather simple and is programmable to execute several configurations of the SSFE detection algorithms, but also other algorithms. The small core enables a rather low complex multi-core system, in which cores can be turned on and off on demand to save energy.

ASIC implementations are synthesized with a 180 nm technology and TTA implementations with a 130 nm technology. To scale energy values we use a factor 1.5 between two successive technologies. The silicon areas are presented in gate equivalents, which is a technology independent measure. The energy consumption per received bit of the multi-core TTA processor is only 1.8–6.5 times higher than in optimized ASIC designs. The result proves that the programmable architecture is capable of achieve both high throughput and low-energy efficiency. Particularly, when the same hardware can be programmed to execute more than one algorithm the energy efficiency is emphasized in terms of reduced leakage power.

5. CONCLUSIONS

We emphasized the significance of co-optimizing the algorithm, software and hardware together in order to reach strict energy, latency and area restrictions. We compared energy consumption of the detection algorithms based on the theoretical complexities in function of signal-to-noise ratio in the channel. The applied method revealed the energy efficient operating points for different detector algorithms. Applying the co-optimizing methods we showed how a carefully designed programmable architecture can achieve the LTE detection rate requirements with a reasonable energy consumption.

6. REFERENCES

- [1] J. Ketonen, M. Juntti, and J. Cavallaro, "Performance-complexity comparison of receivers for a LTE MIMO-OFDM system," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3360–3372, June 2010.
- [2] H. Corporaal, "Design of transport triggered architectures," in *Proceedings of the IEEE International Symposium on Design Automation of High Performance VLSI Systems*, Notre Dame, IN, USA, Mar. 1994, pp. 130–135.
- [3] M. Li, B. Bougart, E. Lopez, and A. Bourdoux, "Selective spanning with fast enumeration: A near maximum-likelihood MIMO detector designed for parallel programmable baseband architectures," in *Proceedings of the IEEE International Conference on Communications*, Beijing, China, May19–23 2008, pp. 737–741.
- [4] J. Pool, A. Lastra, and M. Singh, "Energy-precision tradeoffs in mobile graphics processing units," in *Proceedings of the IEEE International Conference on Computer Design*, Lake Tahoe, CA, USA, Oct.12–15 2008, pp. 60–67.
- [5] M. Siti and M. Fitz, "A novel soft-output layered orthogonal lattice detector for multiple antenna communications," in *Proceedings of the IEEE International Conference on Communications*, June11–15 2006, pp. 1686–1691.
- [6] M. Myllylä, P. Silvola, M. Juntti, and J. R. Cavallaro, "Comparison of two novel list sphere detector algorithms for MIMO-OFDM systems," in *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, Helsinki, Finland, Sept.11–14 2006, pp. 12–16.
- [7] J. Janhunen, T. Pitkänen, O. Silvén, and M. Juntti, "Fixed- and floating-point arithmetic processor comparison for MIMO-OFDM detector," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 8, pp. 1588–1598, Dec. 2011.
- [8] W. Dally, J. Balfour, D. Black-Shaffer, J. Chen, C. Harting, V. Parikh, J. Park, and D. Sheffield, "Efficient embedded computing," *IEEE Communications Magazine*, vol. 41, no. 7, pp. 27–32, July 2008.
- [9] P. Salmela, *Implementations of Baseband Functions for Digital Receivers*, Ph.D. thesis, Tampere University of Technology, Tampere University of Technology, Tampere, Finland, Aug. 2009, [Online]. Available: <http://dSPACE.cc.tut.fi/dpub/bitstream/handle/123456789/6031/salmela.pdf>.
- [10] J. Janhunen, P. Salmela, O. Silvén, and M. Juntti, "Fixed- versus floating-point implementation of MIMO-OFDM detector," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, May22–27 2011, pp. 3276–3279.
- [11] J. Niskanen, J. Janhunen, and M. Juntti, "Selective spanning with fast enumeration detector implementation reaching LTE requirements," in *Proceedings of the European Signal Processing Conference*, Aalborg, Denmark, Aug.23–27 2010, pp. 1379–1383.