Novel Detector Implementations Achieving 3G LTE Downlink and Uplink Requirements

Tuomo Hänninen and Janne Janhunen and Markku Juntti
Centre for Wireless Communications
University of Oulu, Finland
{tuomo.hanninen, janne.janhunen, markku.juntti}@ee.oulu.fi

***Abstract*—We summarize our recent state-of-the-art programmable and reconfigurable detector implementations targeting 3G LTE downlink and uplink requirements. The downlink transmission is based on the orthogonal frequency division multiplexing (OFDM), whereas the uplink transmission uses the single-carrier frequency-division multiple access (SC-FDMA). The downlink implementation is based on the programmable transport triggered architecture (TTA), which provides a flexible and energy efficient architecture template. In TTA implementation, the LTE detection rate requirements up to 20 MHz bandwidth and 4×4 antenna system with 64-QAM, are achieved by using 1–6 programmable cores in parallel. Each core runs at 277 MHz clock frequency and consumes 55.5–64.0 mW depending on the detector configuration. The downlink detector is based on the selective spanning with fast enumeration (SSFE) algorithm. The uplink field-programmable gate array (FPGA) implementation is targeted for 4×4 antenna system and 64-QAM achieving a detection rate requirement for 20 MHz bandwidth. Two different different architectures are implemented. The first one achieves the detection rate requirement with a single processing block running at 231 MHz and the latter one with 4 blocks in parallel, each running at 247 MHz. The used FPGA board is Xilinx Virtex-6 and the implementation has been carried out using Xilinx Vivado high level synthesis (HLS) tool. The implemented detector is based on the *K*-best algorithm.**

I. Introduction

Wireless communication systems have experienced tremendous development during the last two decades. The 4th generation (4G) wireless communication systems and networks will encounter several implementation challenges due to ever increasing capacity and flexibility requirements, whereas the energy budget cannot have similar increase.

In this paper, we summarize our recent programmable and reconfigurable detector implementations both for uplink [1] and downlink [2]. Because of the different energy consumption limitations and hardware resources available in a base station (BS) and in a mobile device (MD), the detector implementations differ significantly beginning already from the algorithm design. Thus, we emphasize the importance of co-optimizing the algorithm, software and hardware together. In a mobile device, a programmability provides extra value if the computing resources can be utilized for radio, but also for other applications during the time there are resources available. Because the energy budget is more limited in MD than in BS, programmability can be exploited by changing detection algorithm based on the current channel condition. For BS and particularly for FPGA implementation we chose an algorithm, which performs well in all channel conditions.

In the uplink, we apply a high-end field programmable gate array (FPGA) to implement a *K*-best list sphere detector (LSD) [3]. We have used a high level synthesis (HLS) tool to implement the algorithm on the Virtex-6 FPGA. Implementation results show that FPGA is a very suitable implementation platform for modern multiple-input multiple-output (MIMO) receiver algorithms and HLS tools are capable of producing competitive FPGA implementation results reducing significantly the design time.

In the downlink implementation, we apply a programmable transport triggered architecture (TTA) to provide an energy efficient processor design for detection. We programmed the processor to execute a selective spanning with fast enumeration (SSFE) [4] detection for a 2×2 antenna system with 16 quadrature amplitude modulation (QAM) and a 4×4 antenna system with 64-QAM, which are assumed to be the most common setup in the near future.

II. System Model

The MIMO antenna system combined with the orthogonal frequency division multiplexing (OFDM) is a promising wireless access scheme and has been included in many wireless standards and proposals, such as the long term evolution (LTE) downlink and its advanced version LTE-A [5]. The 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 the transmit power. Herein we assume the data transmission increase provided by the spatial multiplexing (SM).

Single-carrier frequency-division multiple access (SC-FDMA) has been chosen for the LTE uplink due to its reduced peak-to-average power ratio, which reduces the transmitter power amplifier cost. For the LTE-A uplink the combination of the MIMO and SC-FDMA has been considered in order to increase the transmission rate. This requires more advanced receivers for combating the inter-antenna interference (IAI) and the inter-symbol interference (ISI).

We assume a MIMO transmission architecture with M transmit and N receive antennas, where $M \leq N$. The transmitter in downlink case utilizes a layered space-time architecture with vertical encoding in a 2×2 and horizontal encoding in a 4×4 antenna system. In horizontal encoding, the

signals are separately coded into multiple codewords before spatial multiplexing. The encoded data bits are interleaved and modulated to symbols with quadrature amplitude modulation. A bit-interleaved coded modulation (BICM) is applied.

A single carrier MIMO transmission assumes vertically encoded data stream. The encoded data stream is interleaved and modulated into symbols. After the parallel-to-serial conversion, a cyclic prefix (CP) is added. At the receiver, a discrete fourier transform (DFT) is performed and the symbols from the allocated carriers are selected. After the frequency domain equalization, the symbols are transformed into time domain and the detector is used to calculate the bit log-likelihood ratios (LLR) for the decoder.

The received signal vector on sth subcarrier in a real-valued system 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, $\mathbf{y}_s \in \mathbb{R}^{2N}$, $\mathbf{x}_s \in \mathcal{A}$ denotes the transmitted symbol vector, $\mathcal{A} \in \mathbb{R}$ is the symbol alphabet, $\mathbf{n}_s \in \mathbb{R}^{2N}$ is a noise vector containing complex Gaussian fading coefficients and $\mathbf{H}_s \in \mathbb{R}^{2N\times 2M}$ denotes the channel matrix. The entries of $\mathbf{x}_s$ are chosen independently of each other from a QAM constellation. A soft detection is applied at the receiver. The detected symbol bits are deinterleaved and fed to the decoder. In this study, we concentrate on the soft detection part that usually dominates the computational complexity at the receiver.

III. Error rate performance

The performance of the different receiver algorithms are compared with computer simulations. Two different sphere detectors are compared to the MMSE equalizer with soft demodulation. For the *K*-best algorithm two different list sizes and for the SSFE algorithm different node spanning vectors are used. The used channel model is Vehicular A. The simulations are performed with a single-user MIMO case where both the transmitter and the receiver has two or four antennas. The channel is assumed to have a high or moderate correlation.

Fig. 1 presents a downlink comparison for the *K*-best and the SSFE algorithms. The *K*-best algorithm outperforms the SSFE algorithm, but requires a sorting operation, which is an expensive operation. Thus, less complex SSFE algorithm is chosen for the downlink implementation.

The performances of the different receivers in a correlated Vehicular A channel in uplink case are presented in Fig. 2. The simulation results show the frame error rate (FER) performance of the single user MIMO case and it can be seen that the 2-stage receiver with a time domain sphere detector using the K-best algorithm outperforms the traditional MMSE equalizer with soft demodulation. The K-best algorithm with list size 8 is chosen for the uplink implementation, because it has lower complexity but the performance is still close to the K-best algorithm with list size 16.

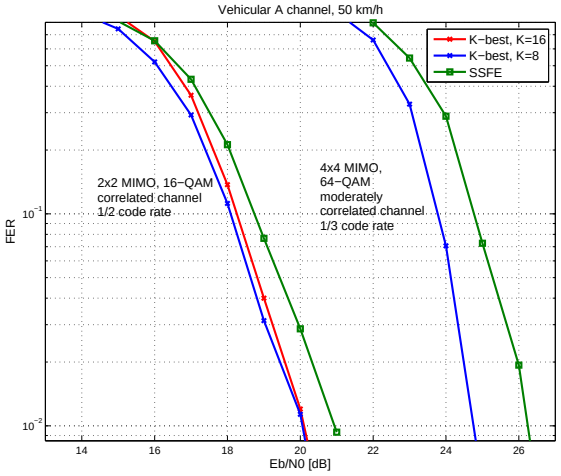

Fig. 1. Detection performance comparison in a vehicular A channel for 2×2 and 4×4 antenna setups in the downlink.

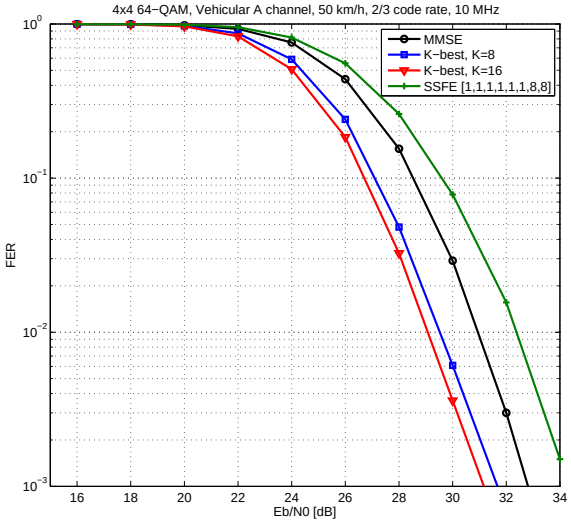

Fig. 2. Detection performance comparison in a correlated vehicular A channel for a 4×4 antenna setup in the uplink.

IV. Downlink implementation on TTA

Transport triggered architecture is an architecture template, in which the function units are triggered by data transports [6]. This is contrary behavior to conventional operation triggered architectures. In general, over ten parallel function units (FU) in very long instruction word (VLIW) processor causes the register file and bypass logic to dominate the silicon area. The drawback is removed in TTA by giving the data path control to the software, which reduces the hardware complexity. It is an ideal architecture template to compare different arithmetic implementations due to its low control overhead that is approximately on par with a finite state machine controlled hardware accelerator. Block diagram of the TTA accelerator is shown in Fig. 3.

A. Function Units

Multipliers have two clock cycle latency to enable a shorter critical path and reduce silicon area. Thus, the processor architecture achieves higher clock frequency. The multipliers support pipelined execution, i.e., new data can be loaded to input ports in every clock cycle. To avoid extra shift operations in the fixed-point processor, the result of the fixed-point multiplication is scaled inside the FU.

The adder FUs include both addition and subtraction operations which are executed in a single clock cycle. A single cycle adder based on the fixed-point arithmetic is not on the critical path of the design.

The slicer is a simple single cycle FU and is the only special function unit in the processor design. The slicing operation executes comparisons between a variable and constant values defined by the the modulation order. The comparisons with constants is more efficiently to execute with hardware than software due to software execution would cause expensive branches. The slicer unit has two inputs: the first input defines how many symbol candidates the unit will output and the second input is the value to be sliced.

TTA itself has a very low utilization of register files due to an efficient interconnection network (bypass network). In general, using registers is more expensive than using memory in terms of silicon area. However, the small number of input and result elements and a strict latency requirement favor to use mainly registers instead of memory. The processor architecture includes eight register files with eight register slots each. In addition, there is a Boolean ($2 \times 1 -$ bit) register file.

To support memory access, the processor architecture includes two LSUs (load/store unit). The LSU can read and write memory. The memory can be read in three clock cycles and write in a single clock cycle. The LSU is triggered with memory address.

B. Results

Area complexity for the TTA implementation is reported in gate equivalents, which is a technology-independent measure corresponding a two-input NAND gate in CMOS technology.

We synthesized the processor design with a low-power 130 nm CMOS technology. We programmed two efficient assembly schedules for the SSFE algorithm. The first one is for a 2×2 antenna system with a 16-QAM and the second one is for a 4×4 antenna system with a 64-QAM. The maximum clock frequency for the processor is 277 MHz.

1) Detector Performance: We assembly programmed the processor to execute an SSFE detector for a 2×2 antenna system with a 16-QAM and a 4×4 antenna system with a 64-QAM. The level update vectors $\mathbf{m} = [1, 2, 2, 3]$ and $\mathbf{m} = [1, 1, 1, 1, 1, 2, 2, 2]$ are used. The processor executes the 2×2 SSFE algorithm, i.e. decodes a symbol vector in 45 clock cycles. With 277 MHz clock frequency it corresponds a decoding rate of 49.2 Mbps. The detection of symbol vector in 4×4 antenna system takes 99 clock cycles. The 277 MHz processor achieves a decoding rate of 67.0 Mbps.

TABLE I
PROCESSOR COMPLEXITIES REPRESENTED IN GES AND PER CENTS

Processor (GE, %)	16-bit FX (277 MHz)
Total	70 730 (100)
Arithmetic	24 460 (35)
ICN	19 530 (27)
Inst. decoder	11 260 (16)
Inst. fetch	3 400 (5)
Reg. banks	12 080 (17)

Table II summarizes the number of operations during the algorithm execution. The additions, subtractions and multiplications are dominating operation in the SSFE algorithm. Even though the number of slicing operations is not very high, a single cycle slicer is an important SFU in the processor. As stated before, register usage is low in the TTA due to the efficient bypass network. In this design, registers are used mainly to store symbol candidates and their PEDs and only the 4×4 antenna system application uses some memory in addition to registers.

TABLE II
THE NUMBER OF OPERATIONS DURING THE SSFE ALGORITHM EXECUTION

Operation	# of OPS in 2×2 system	# of OPS in 4×4 system
ADD	54	208
SUB	54	100
MUL	124	313
SLICER	22	47
RF reads	122	367
RF writes	85	135
memory reads	-	21
memory writes	-	27

2) Energy Dissipation: We summarize power and energy dissipations for processor in Table III. For power we have separated the cell internal and the net switching power dissipations. The cell internal power is consumed when a cell input changes, but there is no change in the output. The net switching power is dissipated when charging and discharging the load capacitance at the cell output. The global operating voltage for the processor is 1.5 V.

We prefer energy dissipation analysis, which takes into account the execution latency and provides a literature comparison between implementations for consumed energy per received bit. In 16-QAM and 64-QAM systems, the symbols are represented with four and six bits, respectively. Thus, in the 2×2 antenna system eight bits and in the 4×4 antenna system 24 bits are received per symbol vector. The energy dissipation is defined as,

$$E = Pt, \tag{2}$$

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

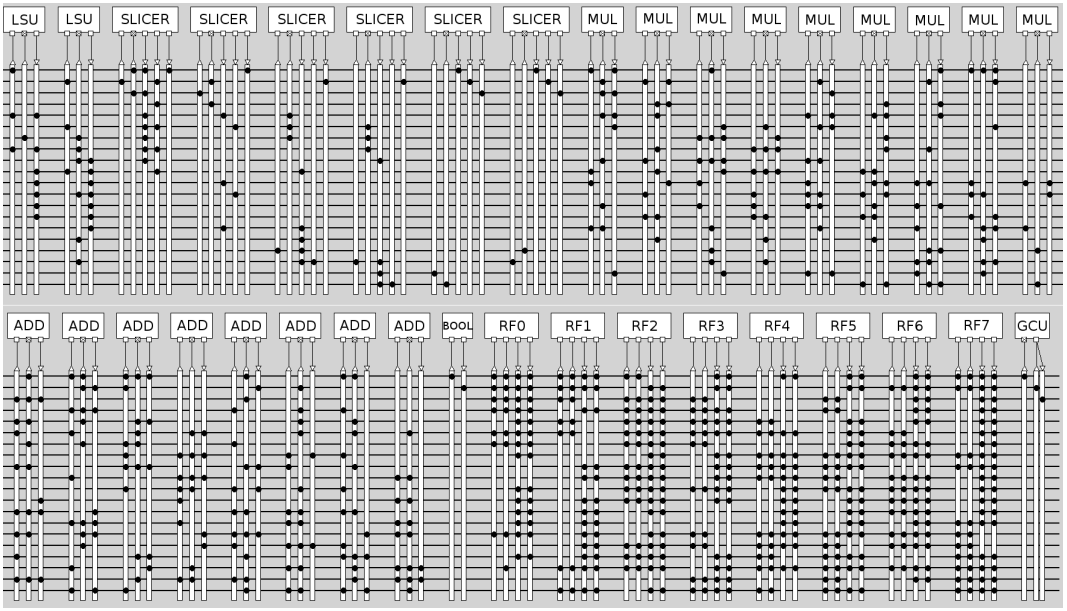

Fig. 3. Block diagram of the TTA accelerator.

TABLE III
PROCESSOR POWER AND ENERGY DISSIPATIONS

Processor 4×4 antenna system	16-bit FX (277 MHz)
Total dynamic P (mW)	64.00
Cell internal P (mW)	32.70
Net switching P (mW)	31.20
Total energy (nJ)	22.81
Energy (nJ/bit)	0.95

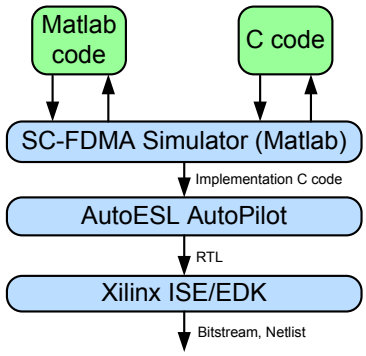

Fig. 4. Tool flow

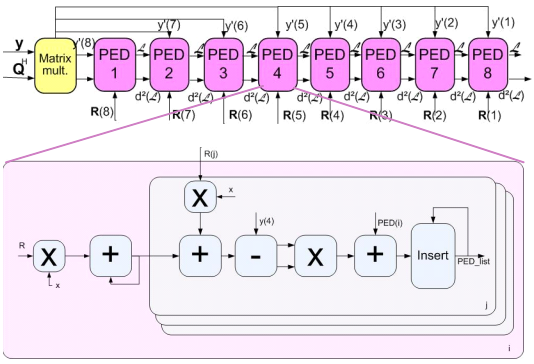

Fig. 5. Block diagram of the Architecture I.

V. UPLINK IMPLEMENTATION ON FPGA

The uplink design is implemented using Xilinx Vivado HLS tool. The tool is used for converting the C code to the register-transfer language (RTL). The HLS tool gives a new abstraction level and hides some of the design complexity. Xilinx ISE/EDK tools are used for synthesis and place & route. All the throughput and complexity results are taken after the place & route. The implementation tool flow can be seen in Fig. 4.

A. Architecture I

Architecture I follows the original idea of the K-best tree search algorithm. Eight partial euclidean distances (PEDs) are calculated on the first level. On levels between 2–8, eight more distances are calculated resulting in 64 PEDs. These 64 PEDs need to be sorted resulting in 8 surviving PEDs. Levels 2–8 need a sorter. Sorting N samples requires N operations, and thus, level including sorter can not be scheduled in less than 64 clock cycles (pipeline initiation interval is larger than or equals 64 clock cycles). The macroarchitecture shown in Fig. 5 is designed for the Architecture I. The PED 1 block calculates eight distances and does not require sorting. PED 2–8 blocks calculate 64 distances and include an insertion sorter.

B. Architecture II

An alternative design is also implemented. To be able to schedule a real-valued 4×4 K-best tree search in less than 64 cycles, a totally different structure is needed. We decided to implement [7] as an alternative architecture. This distributed

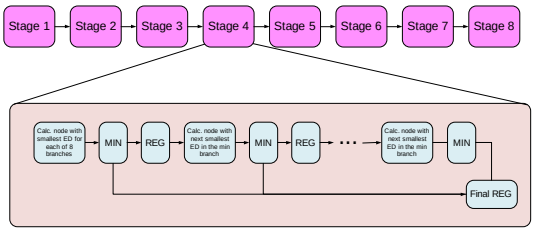

Fig. 6. Block diagram of the Architecture II.

K-best scheme does not require sorting and enables to find the K best candidates in just K clock cycles. Fig. 6 shows the designed macroarchitecture for the Architecture II.

C. Implementation results

The implementation of Architecture I processes a symbol vector in 64 clock cycles and achieves a 247 MHz clock frequency. This means that a single detector block achieves 93 Mbps detection rate. However, several blocks can be used in parallel. The target detection rate 347 Mbps is achieved applying four parallel processing blocks, which achieves a detection rate of 372 Mbps. Architecture II processes a symbol vector in 16 clock cycles and achieves a 231 MHz clock frequency. A single processing block based on the Architecture II detector achieves 347 Mbps. Architectures I and II are compared in Table IV. Four parallel Architecture I blocks achieve the target throughput with less resources than Architecture II.

TABLE IV
K-BEST ($K = 8$) AREA RESULTS

	Four $II = 64$ blocks	One $II = 16$ block
LUT	34476	69383
FF	50044	97676
DSP48	216	228
BRAM	28	287
Max frequency (MHz)	247	231
Detection rate (Mbps)	372	347

VI. Conclusions

We summarized our recent detector implementations achieving 3G LTE downlink and uplink requirements. Different platforms are used for downlink and uplink implementations because of different flexibility requirements and hardware resources available in BS and MD. The downlink implementation is based on the SSFE algorithm. The algorithm is implemented on a programmable TTA, which provides a flexible and energy efficient architecture template. In TTA implementation, the LTE throughput requirements up to 20 MHz bandwidth and 4×4 antenna system with 64-QAM, are achieved by using 1–6 programmable cores in parallel. Each core runs at 277 MHz clock frequency and consumes 55.5–64.0 mW depending on the detector configuration. The uplink implementation is based on the more complex K-best algorithm. The algorithm is implemented on a high-end Xilinx Virtex-6 FPGA using Xilinx Vivado high level synthesis tool. Two different architectures are implemented. The first one achieves the throughput requirement with a single processing block running at 231 MHz and the latter one with 4 blocks in parallel, each running at 247 MHz.

Acknowledgment

The authors would like to thank Xilinx Inc., especially J. Noguera, B. Ozgul and J. Keskinen, for the possibility to evaluate Xilinx Vivado High Level Synthesis tool and for their co-operation. The authors would also like to thank J. Ketonen for her efforts related to this paper.

References

[1] J. Ketonen, J. Karjalainen, M. Juntti, and T. Hanninen, "Mimo detection in single carrier systems," Proceedings of the European Signal Processing Conference, 2011.

[2] J. Janhunen, T. Pitkänen, O. Silvén, and M. Juntti, "Fixed- and floating-point arithmetic processor comparison for MIMO-OFDM detector," *IEEE J. Select. Topics Signal Processing*, vol. PP, no. 99, p. 1, Dec. 2011.

[3] 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.

[4] 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 *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May19–23 2008, pp. 737–741.

[5] 3rd Generation Partnership Project (3GPP), "http://www.3gpp.org," 9.6.2011.

[6] P. Jääskeläinen, V. Guzma, A. Cilio, and J. Takala, "Codesign toolset for application-specific instruction-set processors," in *Proc. SPIE Multimedia on Mobile Devices*, Jan. 29–30 2007, pp. 65 070X–1 – 65 070X–11.

[7] M. Shabany, K. Su, and P. Gulak, "A pipelined scalable high-throughput implementation of a near-ml k-best complex lattice decoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 31 2008-april 4 2008, pp. 3173 –3176.