A VLSI Architecture of the Soft-Output Sphere Decoder for MIMO Systems

Zhan Guo and Peter Nilsson

Department of Electroscience, Lund University, SE-22100 Lund, Sweden

Abstract—Sphere decoders can approach the performance of Maximum-likelihood (ML) decoders for MIMO systems with lower complexity. In this paper, a VLSI architecture of the modified K-best Schnorr-Euchner (MKSE) sphere decoder is proposed for soft-output MIMO decoding. The MKSE decoder can achieve near-ML performance with higher decoding throughput and lower computational complexity. The proposed VLSI architecture is implemented for a \(4\times4\) 16-QAM MIMO system in a 0.13-\(\mu\)m CMOS technology. The implemented soft-output MKSE chip can achieve a decoding throughput of more than 100 Mb/s with a 0.56 mm\(^2\) core area and 97 K gates.

I. INTRODUCTION

MIMO systems are shown to be capable of achieving an extraordinary spectral efficiency near Shannon bound [1]. The optimal Maximum-likelihood (ML) decoders are too complicated to be feasible for the MIMO systems with a higher modulation constellation and a large number of antennas [2]. Lattice decoders are thus proposed to achieve near-ML performance with reasonable complexity [2] [3].

An early VLSI implementation of the sphere decoding algorithm is presented in [4] for a \(4\times4\) 16-QAM MIMO system, in which a breadth-first search method is employed and coined as K-best SD algorithm. The drawback with the VLSI architecture of [4] is that the decoding throughput is limited to 10 Mb/s at a 100 MHz clock frequency. We proposed a K-best SE (KSE) algorithm in [5], which can achieve a much higher decoding throughput and lower complexity than the K-best SD.

Moreover, the sphere decoders can be extended to support soft decision detection for coded MIMO systems. The most well-known soft-output sphere decoders for MIMO systems are list sphere decoder (LSD) of [6] and list sequential sphere decoder (LISS) of [7]. The disadvantages of the LSD and the LISS lie in higher complexity and lower decoding throughput, since both have to maintain a large stack to calculate the soft-outputs.

A modified KSE (MKSE) algorithm is proposed in [8] for soft-output MIMO decoding. The MKSE decoder can achieve higher decoding throughput and lower computational complexity compared to the conventional soft-output sphere decoders. In this paper, we present a VLSI implementation of the MKSE decoder in 0.13-\(\mu\)m CMOS.

II. MODIFIED K-BEST SE ALGORITHM WITH SOFT-OUTPUT

We assume a MIMO system as described in [6]. As shown in Fig. 1, the information bits \(\mathbf{u}\) is encoded and interleaved to become the coded bits \(\mathbf{c}\), which is the input to the constellation mapper. The soft-output MIMO detector takes channel observations \(\mathbf{x}\), and calculates extrinsic information \(L_e(\mathbf{c})\) for each of the coded bits per symbol vector. Then \(L_e(\mathbf{c})\) is de-interleaved to become the \textit{a priori} input \(L_a(\mathbf{c}')\) to the outer soft-input/soft-output (SISO) decoder, which makes decisions \(\hat{\mathbf{u}}\) about the information bits by \textit{a posteriori} information \(L(\mathbf{u})\).

The MKSE algorithm proposed in [8] is outlined as below:

1) At the root node, initialize one path with metric zero.
2) Extend each survivor path, retained from the previous iteration, to several contender paths, and update the accumulated metric for each path.
3) Sort the contender paths according to their accumulated metric.
4) Select the \(K\) best paths, and discard the other paths.
5) From the \(k\)-th iteration, move the discarded paths to the candidates list with is used to calculate the soft-outputs.
6) If the iteration arrives at the end node, stop the algorithm. Otherwise, go to step 2).

Compared to the KSE decoder [5], the modification due to the MKSE lies in step 5) which can be done with a minor increase in complexity. Only some paths moving should be taken into account in the hardware implementation. The MKSE has the same decoding throughput as the KSE with the same \(K\), since both retain the equal number of survivor paths until the end node.
Both the KSE and the MKSE have advantages in hardware implementations compared to the LSD and the LISS. The KSE/MKSE is a single-direction search algorithm, i.e., the calculation proceeds in forward direction only. Consequently, the KSE/MKSE is easily implemented in a parallel and pipelined fashion achieving fixed and higher throughput. Both the LSD and the LISS are two-direction search algorithms, i.e., the calculation proceeds forward and backward. Their decoding throughput is variable, since it depends on the maximally possible search time. The variable decoding throughput may be an overhead to a practical system.

Fig. 2 shows comparative simulation results with an outer rate 1/2 convolutional code with memory 2, 4×4 16-QAM system, the MKSE should retain the discarded paths from the 4-th stage to the last stage. In our implementation, the soft-output module calculates the soft values first based on the discarded paths retained from the 4-th stage. The final soft outputs are calculated just after the $K_M = 20$ paths are obtained at the last stage. In this way, the MKSE does not need to keep up to 80 paths until the last stage, since the transferred data in the soft-output module is the soft values (path metrics) instead of the retained paths. The implementation complexity of the soft-output module is thus reduced sufficiently. Moreover, the decoding period and latency of the hard-output module are not affected by the soft-output module. Consequently, the MKSE has the same decoding throughput as the KSE.

Fig. 4 shows the block diagram of the soft-output module. It consists of five soft-value processing elements (SPE) and a soft-value calculation (SVC) unit. The SPE units accept the discarded paths from the 4-th PE to the 8-th PE, and calculates bit metrics for bit 1 to bit 8. Then the 5-th SPE accepts 15 discarded paths in series from the 4-th PE, and calculates bit metrics for bit 9 to bit 12. The bit metric wordlength is determined to be 24 bits according to simulations. The higher 12 bits represents the bit metric $BM^{(1)}$ when $bit = 1$, while the lower 12 bits represents the bit metric $BM^{(0)}$ when $bit = 0$. All the 16 bit metrics corresponding to a symbol vector are obtained after the calculation in the 8-th SPE. The SVC unit simply subtracts $BM^{(0)}$ from $BM^{(1)}$ for each bit metric, and output the final soft values. The SVC unit consists two subtractors in parallel to calculate the soft outputs. Hence, the calculation period of the 8-th SPE combined with the SVC unit is within 30 clock cycles, as shown in Fig. 3.

Note that the bit metrics for bit 9 to bit 12 calculated in the 5-th SPE are estimation values based on the ZF path augmenting, since the symbol corresponding to bit 9 to bit 12 is not detected until the 6-th stage. The ZF path augmenting is also used in the 7-th SPE, since the symbol corresponding to bit 13 to bit 16 is not detected until the 8-th stage. Therefore, the 5-th and 7-th SPE need to estimate the corresponding symbol by rounding the extra z input.

III. VLSI ARCHITECTURE

The VLSI architecture proposed in [5] for the KSE algorithm is easily extended to support the soft-output MKSE algorithm, with only a soft-output module appended. Fig. 3 shows the overall architecture of the MKSE decoder for a 4×4 16-QAM MIMO system. When the soft-output module is disabled, the architecture is simply a hard-output KSE decoder as described in [5]. Otherwise, the architecture becomes a soft-output MKSE decoder.

The function of the soft-output module is to exploit the discarded paths from the hard-output module. According to the fixed-point simulation results for the 4×4 16-QAM MIMO system, the MKSE should retain the discarded paths from the 4-th stage to the last stage. In our implementation, the soft-output module calculates the soft values first based on the discarded paths retained from the 4-th stage. The final soft outputs are calculated just after the $K_M = 20$ paths are obtained at the last stage. In this way, the MKSE does not need to keep up to 80 paths until the last stage, since the transferred data in the soft-output module is the soft values (path metrics) instead of the retained paths. The implementation complexity of the soft-output module is thus reduced sufficiently. Moreover, the decoding period and latency of the hard-output module are not affected by the soft-output module. Consequently, the MKSE has the same decoding throughput as the KSE.

Fig. 2 shows comparative simulation results with an outer rate 1/2 convolutional code with memory 2. The parameters of the MKSE simulated are $K=5$ and $k=4$, and hence the MKSE has 80 candidate paths for the soft-outputs generation [8]. It is clear from Fig. 2 that the MKSE ($K=5$) outperforms the KSE ($K=5$) in performance by about 1 dB at BER=$10^{-4}$. This shows that the performance of the KSE with larger $K$ can be achieved by the MKSE with much smaller $K$. Both the decoding throughput and the implementation complexity can thus be improved significantly in the MKSE due to smaller $K$. Moreover, note that the MKSE has the same performance as the LSD with a list of 80 candidates, although the complexity of the MKSE is much lower than that of the LSD with the same size of candidate paths [8].
Fig. 5 shows the structure of the 5-th SPE. It consists of three symbol metric calculation (SMC) units. Similarly, the 4-th SPE consists of two SMC units, the 7-th SPE consists of four SMC units and so on. Each SMC unit consists of four bit metric calculation (BMC) units and one DEMOD unit. The DEMOD unit, implemented in a 4-to-4 combinational decoder, demodulates the detected symbol corresponding to four bits. Each output bit of the DEMOD unit is an input to one of four BMC units in the SMC unit. The BMC unit accepts the path metrics (PM) in series and the loaded metrics (LM) from the previous stage of SPE, and calculates the corresponding bit metrics. For the unit SMC3 in Fig. 5, the DEMOD unit accepts the symbol estimated by rounding the \( z \) input. Moreover, The LM inputs of SMC3 are fixed to the maximum value corresponding to the chosen wordlength, since the metrics for bit 9 to bit 12 are not available from the 4-th SPE.

Fig. 6 shows the structure of the BMC unit, which mainly consists of a comparator, two registers and three MUXs. It loads the LM input at the first clock cycles, then accepts the path metrics and performs the comparison in series. After all the path metrics are compared, the bit metrics \( BM^{(0)} \) and \( BM^{(1)} \) are hold at the outputs of the two registers.

Compared to the hard-output KSE, the additional resources to the soft-output MKSE are 16 4-to-4 combinational decoders, 64 comparators, 64 BM registers and 2 subtractors used in the SVC unit. However, wordlength requirements are relaxed in the MKSE compared to the hard-output KSE. The MKSE tries to find a range of better points and calculate the soft values, while the hard-output KSE tries to search for the best point corresponding to the hard-output. What the MKSE focuses is on the diversity of the searched points, not on the absolute accuracy. That is also why a soft-output MIMO detector always prefers finding as many points as possible. Our fixed-point simulation results prove the statement above.

IV. IMPLEMENTATION RESULTS

The proposed VLSI architectures are modeled in Verilog HDL, synthesized using Synopsys Design Compiler, and routed using Cadence SoC Encounter. The RTL and gate level netlists are all verified against the same test vectors generated from the MATLAB fixed-point model. The post-layout timing is verified using Synopsys PrimeTime with net and cell delays back annotated in SDF format.

The soft-output MKSE decoder is to be sent for fabrication in a 0.13-\( \mu \)m CMOS technology. Fig. 7 shows the layout of the chip. The chip core area is 0.75×0.75 mm\(^2\) with 97 K gates.
The post-layout timing simulation shows that the chip can be operated at a maximal clock frequency of 200 MHz. The maximal decoding throughput of the MKSE is thus expected to be more than 100 Mb/s, and the corresponding decoding latency is 1.2 µs. The implemented MKSE chip can make its soft-output module work in sleeping mode in order to reduce its power consumption. This flexibility is useful when the hard-output MIMO decoding could meet the system requirements.

Table I shows the silicon complexity comparison between the hard-output KSE [5] and the soft-output MKSE. The equivalent gates number is defined as the total core area divided by the area of a drive-1 NAND gate. Note that the core equivalent gates number of the MKSE is only 6.5 percent higher than that of the hard-output KSE due to relaxed wordlength requirements. This penalty is worth since the MKSE supports soft-outputs.

For soft-output 4×4 16-QAM MIMO decoding, there has been one silicon implementation estimation published [9] whose complexity is estimated in a 0.18-µm CMOS process. To enable a comparison, we scaled the MKSE decoder from 0.13-µm to 0.18-µm [10]. Table II shows the silicon complexity comparison between [9] and the MKSE. The comparison results show that the MKSE can achieve higher decoding throughput with lower complexity. It should be noted that the performance of [9] is better than that of the MKSE (K=5), since [9] uses the LSD (K=256) algorithm, as shown in Fig. 2.

V. CONCLUSION

As a modification to [5], a VLSI architecture is proposed for the MKSE algorithm supporting soft-output MIMO decoding. The implementation results show that it is feasible to achieve the near-optimal performance and high decoding throughput for 4×4 16-QAM MIMO detection using the proposed algorithm and VLSI architecture with reasonable complexity.

REFERENCES