# A 1.1-Gb/s 115-pJ/bit Configurable MIMO Detector Using 0.13- $\mu$m CMOS Technology

Liang Liu, *Student Member, IEEE*, Fan Ye, *Student Member, IEEE*, Xiaojing Ma, Tong Zhang, *Senior Member, IEEE*, and Junyan Ren, *Member, IEEE*

***Abstract*—This brief presents an efficient and configurable multiple-input–multiple-output (MIMO) signal detector design solution and its high-speed IC implementation. This detector can support 2 × 2/3 × 3/4 × 4 MIMO and quadratic phase-shift keying/16-state quadratic amplitude modulation (QAM)/64-state QAM modulation configurations. The detection algorithm employs an early-pruned technique that can reduce up to 46% node extensions in the K-Best sphere decoder while maintaining an almost maximum-likelihood performance. A parallel multistage folded very large scale integration architecture is accordingly developed that can achieve high detection throughput and configurability. To further improve the IC implementation efficiency, this detector also uses a candidate-sharing structure for partial Euclidean distance calculation and a two-stage sorter for survivor node selection. A test chip has been fabricated using 0.13-$\mu$m single-poly- and eight-metal (1P8M) CMOS technology with a core area of 3.9 mm$^2$. Operating at 1.2-V supply with 137.5-MHz clock, the chip achieves 1.1-Gb/s throughput and consumes 115 pJ per bit, representing 40% more energy efficient than state of the art in the open literature.**

***Index Terms*—Detector, early-pruned K-Best, multiple-input–multiple-output (MIMO), very large scale integration (VLSI).**

## I. Introduction

MULTIPLE-input–multiple-output (MIMO) wireless communication [1] has been adopted in many communication standards such as IEEE 802.11n and IEEE 802.16e/m and is being seriously considered for next-generation gigabit-per-second wireless communications, such as the long-term evolution (LTE) project, the wireless gigabit with advanced multimedia (WIGWAM) project, and the international mobile telecommunications advanced (IMT-A) system [2], [3]. Because of its practical importance, MIMO detector VLSI design has recently received much attention [4]–[9]. Nevertheless, most prior work can only support a single configuration with fixed antenna and constellation size and did not demonstrate a beyond-gigabit-per-second detection throughput.

This work aims to develop a highly configurable MIMO detector design solution that can achieve beyond-gigabit-per-second detection throughput for up to 4 × 4 MIMO with 64-QAM modulation. Moreover, in order to maintain a performance close to maximum-likelihood (ML) detection, we focus on the use of nonlinear breadth-first MIMO detection (also known as K-Best detection). To meet the aforementioned aggressive objective, it is necessary to cohesively consider detection algorithm and very large scale integration (VLSI) architecture design. At the algorithm level, we develop an early-pruned scheme to reduce the computational complexity of K-Best sphere detection. The key is to apply an extension number constraint to prune those relatively less reliable nodes before the partial Euclidean distance (PED) calculation, which could reduce the detection complexity at almost no detection performance degradation. At the detector VLSI architecture level, we develop a parallel multistage folded (PMF) architecture that can seamlessly support various MIMO configurations in terms of antenna number and modulation order. Meanwhile, we develop two techniques to further improve detector VLSI architecture efficiency, including candidate-sharing structure for increasing parallel detection hardware utilization and two-stage sorter for reducing sorting complexity. To evaluate the proposed design solution, we design and fabricate a MIMO signal detector at 130-$\mu$m node. It can support 2 × 2/3 × 3/4 × 4 MIMO and quadratic phase-shift keying (QPSK)/16-state quadratic amplitude modulation (16-QAM)/64-state quadratic amplitude modulation (64-QAM) configurations and only occupies 3.9 mm$^2$. Measurement results successfully demonstrate a 1.1-Gb/s detection throughput and 115-pJ/bit energy consumption for 4 × 4 MIMO with 64-QAM modulation while maintaining an almost-ML detection performance. To the best of our knowledge, this is the highest throughput and most energy efficient among all the MIMO detectors reported in the open literature.

Manuscript received February 24, 2010; revised April 30, 2010; accepted June 17, 2010. Date of publication August 30, 2010; date of current version September 15, 2010. This work was supported in part by China Science and Technology Major Project 2009ZX03007-007 and in part by Shanghai IC Design Special Project 077062005. This paper was recommended by Associate Editor C.-Y. Lee.

L. Liu, F. Ye, X. Ma, and J. Ren are with the State Key Laboratory of ASIC and Systems, Fudan University, Shanghai 201203, China.

T. Zhang is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2010.2058494

## II. Background

Considering a spatially multiplexed $N \times N$ MIMO system, the equivalent baseband system model after real-value decomposition [10] is expressed as

$$r = Hs + n \tag{1}$$

where $s$ is the $2N \times 1$ transmitted vector, in which each component is taken independently from a set of real constellation points $(\Omega)$, $r$ is the $2N \times 1$ received vector, $n$ is the vector of independent identically distributed zero-mean Gaussian noise samples, and $H$ is the $2N \times 2N$ channel matrix. Usually, QR decomposition [10] is applied to the channel matrix: $H = QR$, where $Q$ is unitary and $R$ is upper triangular. Multiplied by the conjugation of $Q$, the system model is rewritten as

$$y = Rs + \nu \tag{2}$$

where $y = Q^H r$, and $v = Q^H n$ is a noise vector with the same statistics as $n$. The objective of MIMO detection is to recover the original $s$, given $r$ and $H$. The ML estimation of $s$ can be formulated as solving the closest-point-search problem

$$\hat{s}_{\mathrm{ML}} = \arg \min_{s \in \Omega^{2N}} |y - Rs|^2. \tag{3}$$

Because of its inherent computational parallelism, feedforward-only search operation, and almost-ML performance, the K-Best algorithm is often used for high-throughput MIMO detector implementations [10]. It transfers the closest-point-search problem in ML detection to a tree-search problem by rewriting (3) in a progressive way

$$\hat{s} = \arg \min_{s \in \Omega^{2N}} \sum_{i=1}^{2N} \left| y_i - \sum_{j=1}^{2N} R_{ij} s_j \right|^2 \tag{4}$$

and carries out breadth-first tree search by

$$\begin{aligned} T_i &= T_{i+1} + inc_i, \\ inc_i &= \left| y_i - \sum_{j=i+1}^{2N} R_{ij} s_j - R_{ii} s_i \right|^2 \\ &= |P_{i+1} - R_{ii} s_i|^2 \end{aligned} \tag{5}$$

where $T_i$ is the PED corresponding to the node at the $i$th layer. Starting from the top ($2N$th) layer of the tree to the bottom (1st layer), the K-Best detector traverses the tree in a breadth-first way and exploits all the child nodes of $K$ survived father nodes in each layer to select $K$ best candidates associated with $K$ smallest PEDs to be extended next. Since only $K$ survivors lead to the calculation of the next layer, most of the nodes are unnecessarily exploited in brute-force K-Best detection. Therefore, a sphere constraint is typically applied to reduce the unnecessary visits to some nodes [6], i.e., given a radius constraint $r^2$, we only consider those nodes that could possibly lead to a PED less than $r^2$.

## III. Proposed Early-Pruned K-Best Detection

### A. Algorithm Description

Although the use of sphere constraint in K-Best detection can reduce computations to a certain extent, there are still a large amount of wasted calculations induced by visiting too many nodes, particularly in high-order modulations. Because the K-Best sphere decoder places a global constraint $r^2$ on the overall PEDs, this tends to be too loose for the accumulative PEDs in upper layers, where only a few terms are contributed. As a result, branches more likely outside the radius will not be trimmed until the search reaches lower layers, which incurs PED calculation wastes. Moreover, because $r^2$ is a constraint for the PED, a node dissatisfying the constraint will not be pruned until its PED is completely calculated and compared with the radius. Therefore, the reduction of PED calculations in the K-Best sphere decoder is still relatively limited.

To further reduce the computational complexity, we propose an early-pruned technique with a simple basic idea: by replacing the global radius constraint $r^2$ with a set of extension number constraints (i.e., for each $k$th father node at the $i$th layer, we simply put an individual extension number constraint $L_{i,k}$ to each node. At the $i$th layer, the point with the smallest PED corresponding to the $k$th father node is given by

$$\tilde{s}_i^k = \left( y_i - \sum_{j=i+1}^{2N} R_{ij} s_j^k \right) \Big/ R_{ii}. \tag{6}$$

The proposed early-pruned K-Best algorithm only exploits $L_{i,k}$ constellation points nearest to $\tilde{s}_i^k$. Using the well-known zigzag enumeration technology [10], these $L_{i,k}$ best points can be conveniently found. Different from the PED-constraint-based early-pruning technique in [11], this proposed early-pruned scheme prunes less reliable paths before PED calculations and thus can save a large number of computations. The constraint $L_{i,k}$ is determined according to the transmission probability of each nodes. In K-Best detection, a more reliable father node under a relatively better channel condition tends to generate more reliable child nodes [12]; thus, fewer paths need to be extended. Furthermore, because the $K$ survived father nodes are selected by sorting, their PEDs increase with the index $k$, which means that the reliability of a father node decreases with $k$. Based on the preceding discussions, we propose to derive the constraint $L_{i,k}$ using the channel coefficient $R_{ii}$ and the father node index $k$

$$\lfloor \beta(\Omega - k/\Omega)/R_{ii} \rfloor \tag{7}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, $\Omega$ is the constellation size, and $\beta$ is a parameter that can be adjusted to tune the tradeoff between detection complexity and performance. A larger value of $\beta$ means a looser constraint and results in a lower probability that the ML node is pruned, and thus, a better performance will be achieved. Meanwhile, more paths are expanded, and the complexity accordingly increases. Two boundary cases should be considered when calculating $L_{i,k}$ using (7): If the calculated $L_{i,k}$ is less than 0, we simply set $L_{i,k}$ as 0, which means that no new path is expanded and the branch is trimmed; If $L_{i,k}$ is larger than $\Omega$, $L_{i,k}$ should be saturated as $\Omega$, which means that all the corresponding child nodes should be exploited. By introducing the constraint $L_{i,k}$, this early-pruned scheme trims less reliable branches at earlier stages of the search process, leading to a more efficient detection than the conventional K-Best sphere decoder.

It should be pointed out that $L_{i,k}$ is determined before tree search, and its calculation is a frame-based operation when a quasi-stationary channel is assumed. In summary, the proposed detection algorithm can be expressed by seven steps.

1) Calculate $L_{i,k}$ according to (7), and set $T_{2N+1} = 0$.
2) Set $i = 2N$.
3) For the $k$th father node in the $i$th layer, exploit its $L_{i,k}$ best child nodes.
4) Calculate the PEDs of these $L_i = L_{i,1} + L_{i,2} + \cdots + L_{i,K}$ nodes according to (5).
5) If $L_i > K$, select the $K$ best PEDs by sorting; else, sort these $L_i$ PEDs in their ascending order.
6) Set $i = i - 1$. If $i \neq 1$, go back to step 3.
7) Select the node with the smallest PED, and output its corresponding path as the final detection result.

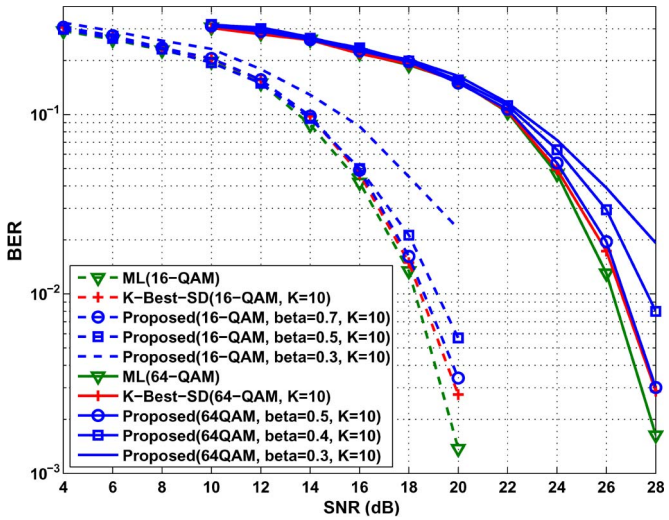


Fig. 1. Simulated BER performance and comparison with K-Best sphere detection and ML detection.

TABLE I
AVERAGE NUMBER OF EXTENDED NODES (4 × 4, 16-QAM)

| SNR | 14 | 16 | 18 | 20 |
|---|---|---|---|---|
| K-Best sphere detection | 145 | 133 | 118 | 100 |
| Proposed scheme with $\beta$=0.7 | 68 | 68 | 68 | 68 |

TABLE II
AVERAGE NUMBER OF EXTENDED NODES (4 × 4, 64-QAM)

| SNR | 22 | 24 | 26 | 28 |
|---|---|---|---|---|
| K-Best sphere detection | 344 | 289 | 240 | 190 |
| Proposed scheme with $\beta$=0.5 | 103 | 103 | 103 | 103 |

### B. Simulation Results

The proposed early-pruned K-Best algorithm is compared with the K-Best sphere decoder in terms of uncoded bit-error-rate (BER) performance and computational complexity measured by the average number of visited nodes in each tree search. The simulation setup is based on 16-QAM and 64-QAM transmission over a 4 × 4 Rayleigh flat-fading MIMO channel with additive white Gaussian noise. For each simulation, 1000 packets, each containing 1024 bytes of information bits, were transmitted, and each package transition spanned one realization of the MIMO channel. Fig. 1 compares the BER of the proposed algorithm with different $\beta$'s to the K-Best sphere detection and ML detection. The radius constraint $r^2$ in the K-Best sphere detection is set in the same way as in [6]. As shown in Fig. 1, as the value of $\beta$ increases, the detector can achieve a better detection performance. When $\beta$ is equal to 0.7 and 0.5, respectively, this proposed detection algorithm can achieve the performance very similar to that of K-Best sphere detection and ML detection for 16-QAM and 64-QAM.

Tables I and II show the average number of nodes visited when using this proposed algorithm and conventional K-Best sphere detection. Because the radius $r^2$ is determined by the channel noise statistics, the K-Best sphere detection has a signal-noise-ratio-dependent variable computational complexity. The computational complexity of this proposed algorithm, on the other side, is constant, and at least 32% and 46% node extensions are reduced for 16-QAM and 64-QAM, respectively.

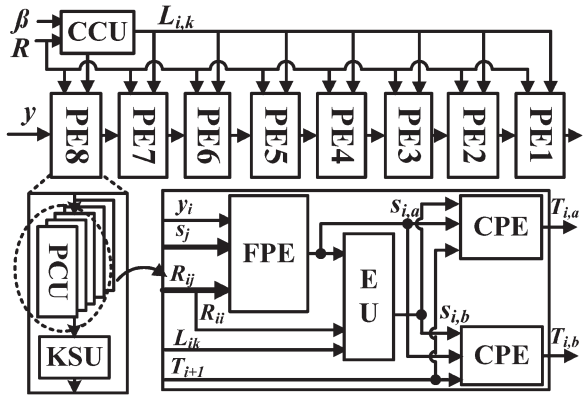


Fig. 2. Proposed parallel multistage folded architecture for implementing early-pruned K-Best detection.

## IV. VLSI ARCHITECTURE DESIGN

### A. Proposed PMF Architecture

To effectively exploit the inherent parallelism of breadth-first search and meanwhile maintain a high configurability, we develop a PMF MIMO detector architecture to implement the previously presented detection algorithm, as shown in Fig. 2. The constraint calculation unit receives channel information $R$ and parameter $\beta$ as inputs and calculates $L_{i,k}$ according to (7). The detector contains eight stages of process elements (PEs), corresponding to the eight layers of the searching tree in the case of 4 × 4 MIMO configuration. Each stage of PE includes $K$ parallel PED calculation units (PCUs) dealing with $K$ father nodes simultaneously to improve the throughput and a K-Best select unit selecting $K$ smallest PEDs. The PCU consists of a father node PE (FPE), an enumeration unit (EU), and two folded child node PEs (CPEs). The FPE computes $P_{i+1}$ in (5), which is common to all the child nodes extended from the same father node. The EU computes $\tilde{s}_i^k$ in (6) and enumerates the real-value constellation points in a zigzag manner [10]. When the $L_{i,k}$ best nodes have been found, the enumeration stops to save power. The CPEs calculate the two PEDs each time and finish the computation of all the $L_{i,k}$ PEDs in a time-multiplexed manner.

This proposed MIMO detector architecture can efficiently realize various configurations in terms of antenna number and modulation order. The antenna configuration is enabled by the multistage architecture, which can be configured to different MIMO modes by activating different PE stages. For example, we configure the detector to a 2 × 2 MIMO mode by closing the PE4-PE1 and generating the detection result from PE5. The folded architecture offers the ability to support modulation configurations. For example, the child node PEs calculate the PEDs by reusing hardware resources recursively. The maximum reuse time, which is defined as folding factor $N_m$, is decided by the modulation order and is set to 1, 2, and 3 for QPSK, 16-QAM, and 64-QAM, respectively. The PE is closed when finishing the calculation of the $L_{i,k}$ PEDs. Generally, $L_{i,k}$ is much smaller than the constellation size; thus, a great amount of power can be saved.

### B. Techniques to Further Improve Implementation Efficiency

We develop two techniques that can further improve the MIMO detector VLSI architecture efficiency. The first

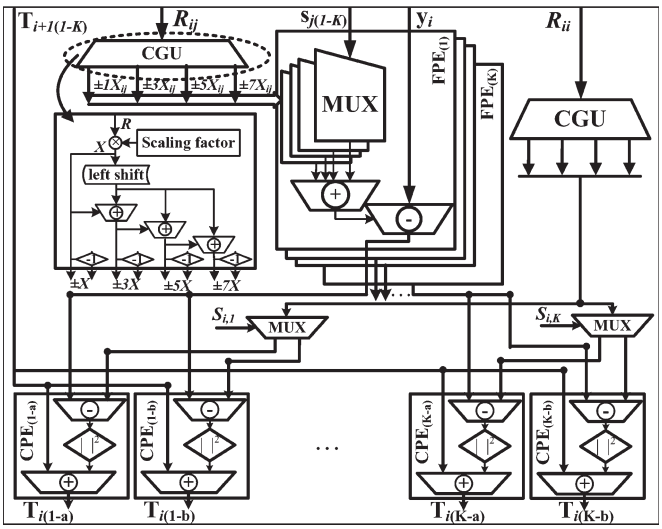


Fig. 3. Candidate-sharing structure for partial Euclidean distance calculation.

technique aims to share candidate PED calculation to reduce the corresponding computational complexity, and the second one intends to reduce the sorting complexity through a two-stage sorter. These two techniques are described as follows:

*1) Candidate-Sharing PCU:* PED calculation is the most complex process in MIMO detection. A straightforward realization of the PED calculation tends to incur a large silicon overhead. For example, one FPE at the $i$th stage needs $2N-i$ multipliers and $2N-i$ adders to finish the $P_{i+1}$ calculation in (5). Moreover, $K$ such FPEs need to be duplicated to calculate $K$ father nodes simultaneously. One of the most effective methods to save silicon area in a parallel design is resource sharing. Since the real-value QPSK/QAM point is selected from a finite set of odd integers and then multiplied with a normalization factor, e.g., $s \in [\pm 7, \pm 5, \pm 3, \pm 1] \times 1/\sqrt{42}$ for 64-QAM, we propose a candidate-sharing structure to improve PED calculation efficiency, where $R_{ij}s_j$ in (5) is not calculated separately in each FPE. Instead, a candidate generation unit (CGU) is used to generate all the possible values of $R_{ij}s_j$. $K$ FPEs share a single CGU. Each FPE finds the right candidate with a multiplexer and finishes the $P_{i+1}$ calculation. Similarly, the results of $R_{ii}s_i$ in (5) is also generated by a CGU and shared by all the $2K$ CPEs. With the proposed candidate-sharing technology, considerable hardware resources are saved. In addition, taking the property that $s$ is taken from a known finite constellation, the computation of $Rs$ can be pushed back to the preprocessing stage, which is performed significantly less often and hence can further reduce the power. The architecture of the resulted PCU is shown in Fig. 3.

*2) Two-Stage Sorter:* In K-Best detector, sort plays a critical role in throughput and area/power efficiency. A distributed sorter (DS) [6], which divides the elements to be sorted into a number of groups and then sorts each group separately by local sorters, has been developed to reduce the sorting complexity and latency. However, it is an approximate sorter, and error happens when the elements to be sorted are unevenly distributed. To exploit the benefits of the DS while preventing its deficiencies and to support different modulation configurations, we use a two-stage sorter as shown in Fig. 4.

The first stage consists of a $2K$-size folded bubble sorter (BS) and a data buffer. Exploiting the folded architecture of

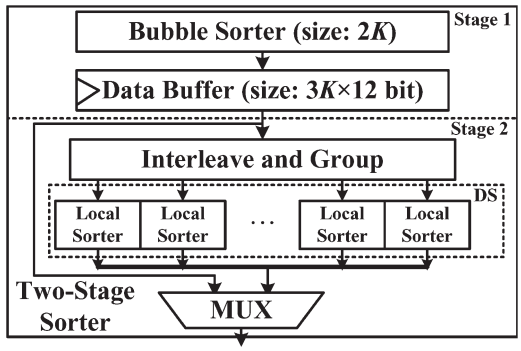


Fig. 4. Block diagram of the two-stage sorter.

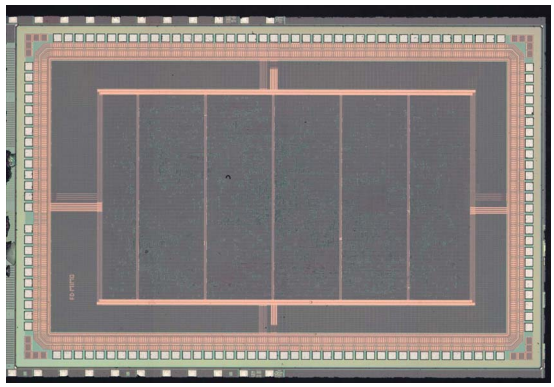

Fig. 5. Die micrograph.

TABLE III
AREA PARTITION BY FUNCTION BLOCK

| Function Block | FPE | CPE | EU | Sorter | Other Logic |
|---|---|---|---|---|---|
| Area | 14.2% | 20.1% | 15.8% | 40.6% | 9.3% |

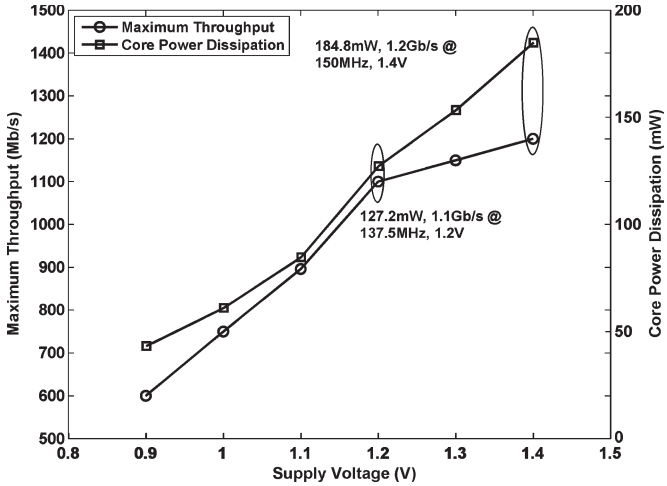


Fig. 6. Measurement of the core power dissipation and the maximum throughput versus supply voltage.

the CPE, the BS also reuses the hardware iteratively with a same folding factor $(N_m)$ as that of the CPE. Selecting $K$ temporary winners from the outputs of the CPEs per iteration, the BS generates a total of $N_m$ sets of $K$ survivors. The data buffer stores these survivors and sends them to the second stage each $N_m$ cycles. The outputs of the BS are partially sorted in two dimensions. The BS sorts the PEDs within a set in

TABLE IV
COMPARISON OF THIS CHIP TO PREVIOUS MIMO DETECTOR

| | Ref. [6] | Ref. [7] | Ref. [8] | Ref. [9] | This Work |
|---|---|---|---|---|---|
| Antenna Size | 4×4 | 4×4 | 2×2 ∼ 6×4 | 2×2 | 2×2 ∼ 4×4 |
| Modulation | 64-QAM | 64-QAM | QPSK∼ 64-QAM | QPSK∼64-QAM | QPSK∼64-QAM |
| Performance (close-to-ML) | Yes | Yes | No | Yes | Yes |
| Process Technology ($\mu$m) | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 |
| Gate Count ($K$) | 2950 | 114 | 205 | 55 | 491 |
| Throughput (Mb/s) | 100 | 675 | 114 | 146 | 1100 |
| Power Efficiency (pJ/bit) | 8470 | 200 | 263 | N/A | 115 |

one dimension. The average PED values among sets are sorted in their ascending orders due to the zigzag enumeration. The folded BS offers two advantages: 1) By eliminating the nodes with PEDs larger than the $K$th best at each iteration, the BS greatly reduces the number of elements to be sorted in the second stage. 2) The partially sorted output of the BS further simplifies the task of the second stage from sorting random data to merging ordered lists. In the second stage, a DS is used to choose the final $K$ best results among the temporary winners generated by the first stage. To tackle the uneven distribution problem, an interleave-and-group block (IGB) is used. Taking the second advantage of the BS, the IGB evens the PED value among groups in a way that the smaller (larger) PEDs in the former outputted sets are grouped with the larger (smaller) PEDs in the later sets.

The two-stage sorter can support different modulation orders. For QPSK, the $2K$-size BS is sufficient to carry out strict sorting; thus, the second stage can be simply bypassed. The only difference in sorting 16-QAM and 64-QAM signals is the number of PEDs to be sorted in each group of the DS. Thus, by adjusting the number of active comparators in the local sorter, the two-stage sorter can be configured to 16-QAM and 64-QAM modes.

## V. CHIP IMPLEMENTATION AND COMPARISON

The previously presented configurable early-pruned K-Best MIMO detector has been fabricated in a 0.13-$\mu$m 1P8M CMOS technology. Fig. 5 shows the chip micrograph. The chip is 2.5 mm × 3.8 mm silicon with a core area of 3.9 mm$^2$. The total area can be partitioned by function blocks, as shown in Table III. Fig. 6 shows the maximum throughput and core power consumptions (measured by probing the core current) with different supply voltages. At the normal 1.2-V core power supply, the detector can work at 137.5-MHz clock rate to support a 1.1-Gb/s throughput. The measured energy consumption is 115 pJ/bit. The detector achieves a peak detection throughput of 1.2 Gb/s at a 1.4-V voltage supply while consuming a core power of 185 mW. The detection throughput in Fig. 6 is given by

$$\text{Throughput} = f_c \times \frac{\log_2 \Omega \times 2N}{N_m} \tag{8}$$

where $\Omega$ is the real constellation size, $N$ is the antenna size, $f_c$ is the clock frequency, and $N_m$ is the folding factor.

Table IV summarizes the major characteristics of this chip and presents a comparison with other reported MIMO detectors. In terms of detection throughput, our design is outstanding, compared with other detectors, and it is for the first time to achieve a beyond-gigabit-per-second throughput. This detector also achieves the lowest energy consumption, which is 40% less than the best previous result [7]. The hardware cost of our implementation is larger than previous $4 \times 4$ 64-QAM detectors in [7] and [8], which is reasonable, considering the fact that this chip has high configurability and the highest detection throughput.

## VI. CONCLUSION

This brief has presented the design of a MIMO detector chip. At the algorithm level, the proposed early-pruned K-Best detection reduces the complexity significantly and remains a close-to-ML performance. At the architecture level, a PMF architecture has been proposed for high-throughput and flexible configurability. Candidate-shared PCU and two-stage sorter are developed to further reduce the hardware silicon cost. Measurement results have shown that the proposed detector outperforms all the other reported chips in terms of throughput and energy consumption.

## REFERENCES

[1] B. M. Hochwald and S. T. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, May 2003.
[2] N. Jayant, "Special issue on gigabit wireless," *Proc. IEEE*, vol. 92, no. 2, pp. 195–197, Feb. 2004.
[3] M. Simon, S. Winter, U. Klepser, and R. Weigel, "An OFDM MIMO transmitter for wireless gigabit with advanced multimedia (WIGWAM)," in *Proc. Eur. Conf. Wireless Technol.*, Oct. 2007, pp. 134–137.
[4] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-Best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.
[5] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.
[6] S. Chen, T. Zhang, and Y. Xin, "Relaxed K-Best MIMO signal detector design and VLSI implementation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 3, pp. 328–337, Mar. 2007.
[7] M. Shabany and P. G. Gulak, "A 0.13 $\mu$m CMOS 655 Mb/s $4 \times 4$ 64-QAM K-Best MIMO detector," in *Proc. IEEE ISSCC*, Feb. 2009, pp. 256–257/257a.
[8] C.-J. Huang, C.-W. Yu, and H.-P. Ma, "A power-efficient configurable low-complexity MIMO detector," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 2, pp. 485–496, Feb. 2009.
[9] C. Barbara, M. Guido, and V. Emanuele, "Decoding the golden code: A VLSI design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 156–160, Jan. 2009.
[10] M. Wenk, M. Zellweger, A. Burg, N. Felber, and W. Fichtner, "K-Best MIMO detection VLSI architectures achieving up to 424 Mbps," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 1151–1154.
[11] Q. Li and Z. Wang, "Early-pruning k-Best sphere decoder for MIMO systems," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 1994, pp. 40–44.
[12] W. Zhao and G. B. Giannakis, "Reduced complexity closest point decoding algorithms for random lattices," *IEEE Trans. Wireless Commun.*, vol. 5, no. 1, pp. 101–111, Jan. 2006.