Research Article

A Power-Efficient Soft-Output Detector for Spatial-Multiplexing MIMO Communications

Hsiao-Chi Wang, Tung-Lin Liu, Yuan-Wei Wu, and Hsi-Pin Ma

Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan

Correspondence should be addressed to Hsi-Pin Ma, hp@ee.nthu.edu.tw

Received 15 June 2011; Revised 14 September 2011; Accepted 17 November 2011

Academic Editor: Zhiyuan Yan

Copyright © 2012 Hsiao-Chi Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

VLSI implementation of a configurable power-efficient MIMO detector is proposed to support 4×4 spatial multiplexing and modulation from QPSK to 64-QAM. A novel tree search algorithm is proposed to enable the detector to provide soft outputs and to be implemented in parallel and pipelined hardware architecture. The frame error rate (FER) of the detector approaches the quasi-optimal sphere decoder, with 0.5-dB degradation. Moreover, the proposed detector can operate at the optimal voltage under different configurations and detect/recover timing error at run time by a novel adaptive voltage scaling technique with double sampling circuitry. The proposed detector, using TSMC 0.18 μm single-poly six-metal CMOS process with a core area of $1.17 \times 1.17 \text{ mm}^2$, provides fixed throughput of 45 Mbps in 64-QAM configuration, 120 Mbps in 16-QAM configuration, and 60 Mbps in QPSK configuration. The normalized power efficiency of the design for 64-QAM and 16-QAM configurations is 1.56 Mbps/mW and 2.53 Mbps/mW, respectively. Compared with the conservative margin-based design, the proposed design achieves a 48.8% power saving.

1. Introduction

Multiple-input multiple-output (MIMO) techniques in combination with high constellation orders have been identified as a promising approach to high spectral efficiency systems. Prominent detection in spatial multiplexing is essential for system performance. Maximum likelihood (ML) detection is the optimal method in spatial multiplexing systems. Sphere detection (SD) methods [1–3] compute the ML solution by taking into consideration only the lattice points inside the sphere with a given radius. Because a soft-in-soft-out error-correction-code (ECC) decoder [4] can have better error-correction performance than a usual hard ECC decoder, a soft-output MIMO detector, which cooperates with ECC better, is needed for coded communication system. The soft-output MIMO detector algorithms, such as list sphere decoder (LSD) [5] and soft single tree search sphere decoder (STS-SD) [6], have quasi-optimal performance with ECC. However, LSDs have huge complexity when list is extended, and STS-SD has variable throughput when channel condition is changed. The high computational intensity

and the variable throughput characteristic of the iterative methods prevent current practical implementations from conforming the requirements for actual chip area, latency, and power consumption.

Fixed-complexity sphere decoder (FSD) algorithm [7] is another practical solution to MIMO detection. The FSD is similar to SD but with different search criteria. FSD performs a fixed tree search by visiting all nodes in the top level and just visiting one node in other levels to simplify the tree search in SD. Therefore, the complexity of FSD can be lower than SD. The FSD algorithm presents a quasi-ML performance and fixed complexity in an uncoded system. However, FSD is not compatible with powerful soft-input ECC decoders and needs some modifications. The work in [8] provides a good reference. In this paper, we reduced the visiting nodes by another way of node visiting distribution and found the local minima for soft-output under the tradeoff between performance and complexity using a modified FSD scheme. Furthermore, a novel method to simplify the SE enumeration [9] is proposed with the FSD. Therefore, the proposed FSD does not need to sort all points of the constellation.

To achieve more power saving, a full-pipelined parallel architecture is proposed according to the modification of the algorithm. The parallel design and adaptive voltage scaling technique can provide a power-efficient ASIC solution.

In this paper, a robust and power-efficient solution for spatial-multiplexing MIMO detection is proposed. The proposed soft MIMO detector can provide LLR output to ECC decoder and provide a turbo-MIMO solution. The proposed MIMO detector has the following features.

(a) *Provision of High Order Modulations.* The proposed MIMO detector supports multiple modulation configurations, including QPSK, 16-QAM, and 64-QAM modulation.

(b) *Soft-Output Performance.* With a modified tree search algorithm, the proposed MIMO detector, which provides soft-valued outputs, is compatible with soft-in-soft-out ECC decoders to attain enhanced detection performance.

(c) *Parallel and Pipelined Architecture.* Iterative sphere decoding methods prevent decoders from efficient hardware implementation. The proposed soft-output fixed-complexity sphere decoder (SFSD) retains the advantage of the fixed-complexity sphere decoder (FSD) [7] and therefore can be implemented in parallel and full pipelined designs to increase hardware efficiency.

(d) *Fixed Throughput.* General soft-output sphere decoding solutions only provide variable throughput, which makes imperfect use of hardware resources due to the sequential nature. The SFSD provides fixed throughput and achieves better performance than usual sphere decoders when the throughput is fixed. The maximum throughput is 120 Mbps in the proposed decoder.

(e) *Error-Recovered Adaptive Voltage Scaling.* A novel adaptive voltage scaling method is applied to the detector to reduce power dissipation. With double sampling circuitry, a timing error will be detected and recovered at run time. Therefore, an optimal voltage can be achieved and also keep the processing from functionality violation. With these techniques, a 48.8% saving in power is achieved.

(f) *Configurable, Complexity-Efficient, and Power-Efficient Hardware Implementation.* The configurable ASIC provides fixed throughput of 45 Mbps in 64-QAM configuration, 120 Mbps in 16-QAM configuration, and 60 Mbps in QPSK configuration. The normalized power efficiency is 1.56 Mbps/mW and 2.53 Mbps/mW for 64-QAM and 16-QAM configurations, respectively. The complexity efficient is 1.19 Mbps/K-gate for 16-QAM configuration.

The remainder of the paper is organized as follows. Section 2 reviews the conventional sphere decoding algorithm. The proposed SFSD algorithm with related simulation results is introduced in Section 3. Section 4 presents the logic design. Section 5 reports the hardware implementation. Finally the paper is concluded in Section 6.

2. Conventional Sphere Decoding Algorithm

Let us consider a MIMO system with M transmitting and N receiving antennas ($N \geq M$). M streams of Q bits of data, $x_{j,b}$, $j = 1, 2, \dots, M$, and $b = 1, 2, \dots, Q$, are mapped to an M -dimensional transmitted symbol vector $\mathbf{s} = [s_1 s_2 \dots s_M]^T$, using 2^Q -QAM modulation. The received complex vector, $\mathbf{y}$, is given by

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}, \quad (1)$$

where $\mathbf{H}$ is an $N \times M$ channel matrix, which is assumed to be known in advance, and $\mathbf{n}$ is the complex Gaussian noise vector.

The *a posteriori* log-likelihood ratio (LLR) of the bit $x_{j,b}$, conditioned on the received symbol vector, $\mathbf{y}$, provides a soft-in-soft-out detector information for decision and can be expressed as

$$L(x_{j,b} | \mathbf{y}) = \ln \frac{\Pr[x_{j,b} = +1 | \mathbf{y}]}{\Pr[x_{j,b} = -1 | \mathbf{y}]}. \quad (2)$$

By Bayes' theorem, the max-log approximation, and proof in [5, 6], (2) can be rewritten as

$$L(x_{j,b} | \mathbf{y}) \cong \min_{\mathbf{s} \in \mathcal{S}_{j,b}^{(-1)}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 - \min_{\mathbf{s} \in \mathcal{S}_{j,b}^{(1)}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2, \quad (3)$$

where $\mathcal{S}_{j,b}^{(-1)}$ and $\mathcal{S}_{j,b}^{(1)}$ are search spaces, $\{\mathbf{s} | x_{j,b} = -1\}$ and $\{\mathbf{s} | x_{j,b} = 1\}$, respectively. The maximum-likelihood (ML) minima in (3) is expressed as

$$\mathbf{s}^{\text{ML}} = \arg \min_{\mathbf{s} \in \mathcal{C}^M} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2, \quad (4)$$

where $\mathcal{C}^M$ is the set of constellation symbols in the M -dimensional complex space. Letting $\overline{x_{j,b}^{\text{ML}}}$ be the binary complement of the b th bit in the j th data symbol of $\mathbf{s}^{\text{ML}}$, the other minima in (3) can be expressed as

$$\overline{\mathbf{s}}_{j,b}^{\text{ML}} = \arg \min_{\mathbf{s} \in \mathcal{S}_{j,b}^{(\overline{x_{j,b}^{\text{ML}}})}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2. \quad (5)$$

By means of the QR decomposition of the channel matrix $\mathbf{H} = \mathbf{Q}\mathbf{R}$, (4) and (5) can be reformulated as

$$\mathbf{s}^{\text{ML}} = \arg \min_{\mathbf{s} \in \mathcal{C}^M} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2, \quad (6)$$

$$\overline{\mathbf{s}}_{j,b}^{\text{ML}} = \arg \min_{\mathbf{s} \in \mathcal{S}_{j,b}^{(\overline{x_{j,b}^{\text{ML}}})}} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2, \quad (7)$$

respectively, where $\tilde{\mathbf{y}} = \mathbf{Q}^H \mathbf{y} = \mathbf{R}\mathbf{s} + \tilde{\mathbf{n}}$, $\mathbf{Q}$ is an $N \times M$ matrix with orthogonal unit norm columns, $\mathbf{R}$ is an $M \times M$ upper-triangle matrix, and $(\cdot)^H$ denotes the Hermitian matrix operator. Note that the noise term $\tilde{\mathbf{n}} = \mathbf{Q}^H \mathbf{n}$ keeps the same statistical properties as $\mathbf{Q}$ is a unitary matrix [1].

TABLE 1: Mean and variance of the distribution of the number of the visiting branches for each layer under SD ordering and FSD ordering at SNR = 10 dB in 4×4 MIMO 16-QAM System.

Layer	SD ordering		FSD ordering	
	Mean	Variance	Mean	Variance
4	2.875270419	6.793185738	4.744884884	13.61550639
3	2.096801078	3.001800963	1.627991568	1.339530374
2	1.621236302	1.412882193	1.406256828	0.75118707
1	1.283783145	0.648927997	1.20332918	0.384272589

In addition, sorted QR decomposition (SQRD) [10] is applied, computing the Euclidean distance (ED) $\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2$ in (6) and (7) recursively as

$$d_i(\mathbf{s}^i) = d_{i+1}(\mathbf{s}^{i+1}) + \left| \tilde{y}_i - \sum_{j=i}^M R_{i,j} s_j \right|^2, \quad (8)$$

$$i = M, M-1, \dots, 1,$$

where $d_{M+1}(\mathbf{s}^{M+1}) = 0$, $\mathbf{s}^i = [s_i, s_{i+1}, \dots, s_M]^T$, and $\tilde{y}_i$ and $R_{i,j}$ are, respectively, elements in $\tilde{\mathbf{y}}$ and $\mathbf{R}$. Lattice points with partial Euclidean distance (PED), $d_i(\mathbf{s}^i)$, greater than the square of a given radius r , are invalid. Therefore, the candidate search is confined to lattice points within a sphere to reduce the detection complexity. Nevertheless, the sequential search property and the variable computational complexity prevent the conventional SD from efficient hardware design.

3. Proposed Soft-Output FSD (SFSD)

3.1. Algorithm Description. FSD algorithm [7] is another solution to MIMO detection, which is similar to SD but has two major differences.

- (i) A fixed tree search is performed that FSD visits all nodes in the top level and just visits one node in other levels to simplify the tree search in SD.
- (ii) The channel matrix ordering [11] is modified that the smallest column norm of channel matrix $\mathbf{H}$ is ordered in the first level of the tree.

The FSD algorithm presents a quasi-ML performance and fixed complexity in an uncoded system. However, FSD cannot be compatible with powerful soft-input ECC decoders.

For soft-output MIMO detection, the quasi-ML solution in (6) and the local minima in (7) are essential to calculate LLR. Therefore, the proposed FSD is a modification of the FSD by finding the local minima in (7) for soft-output. Accordingly, a soft-output SD requires more branches than the FSD. Monte Carlo simulations for the number of required visiting branches for SD and FSD algorithms were performed. Tables 1 and 2 show the mean and variance of the number of branches for a visited node at each layer in a 4×4 MIMO system using 16-QAM and 64-QAM, respectively. The distribution helps FSD algorithm to fix the number of

TABLE 2: Mean and variance of the distribution of the number of visiting branches for each layer under SD ordering and FSD ordering at SNR = 20 dB in 4×4 MIMO 64-QAM System.

Layer	SD ordering		FSD ordering	
	Mean	Variance	Mean	Variance
4	3.872876347	28.55566101	9.119207119	105.9677423
3	2.23370341	5.910370473	1.603919127	1.855906856
2	1.589407569	1.867583287	1.342484971	0.795757947
1	1.271200344	0.906038796	1.16755946	0.377373481

branches in tree traversal. Both tables show higher mean and variance of visited branches in SD ordering compared with FSD ordering in all layers except layer 4. This can reduce the searching nodes and save computation complexity. By denoting the number of branches at the i th layer as n_i , we can denote the distribution of branch number for 4 layers by $\mathbf{n} = [n_1, n_2, n_3, n_4]^T$. A diagram can show the idea in Figure 1. We can see that means in Tables 1 and 2 are in the descending order from the third level to the first level. Then we can find that the property of n_i is

$$E[n_3] \geq E[n_2] \geq E[n_1]. \quad (9)$$

SFSD may follow the property and can be fixed to $\mathbf{n} = [2, 3, 4, P]^T$, $\mathbf{n} = [3, 3, 3, P]^T$, where P is equal to the number of the constellation points (2^Q) or other distributions which follow (9).

Figure 2 shows FER performance of different SFSD distributions with LLR maximum $L_{\max} = 16$ in 4×4 MIMO 16-QAM system. The simulation distributions $\mathbf{n} = [n_1, n_2, n_3, n_4]^T$ on the figure are selected based on (9) and the mean/variance of the visiting branches in Tables 1 and 2. SD can be considered as the optimal performance here. We can see that the branches of node can affect the performance of SFSD, and more branches can have better performance directly. But too high distribution $\mathbf{n}$ can cause the tree search complexity to increase. Table 3 presents the total visited nodes in different SFSD distributions with 16-QAM modulation. The distribution $[1, 1, 1, 16]^T$ is used in the hard output FSD algorithm. The total visited nodes may represent the computation complexity of SFSD. We can find that the $\mathbf{n} = [1, 2, 2, 16]^T$ SFSD needs the least branch expansion and has only 0.5 db performance degeneration compared to SD. The $\mathbf{n} = [2, 3, 4, 16]^T$ SFSD has better performance which degrades about 0.1 dB but has the highest complexity. It is a tradeoff between performance and complexity. Therefore, the branch distribution $\mathbf{n} = [1, 2, 2, 2^Q]^T$ for tree search has low complexity and is proposed to full-pipeline parallel hardware architecture implementation in the proposed SFSD. The top layer, n_4 , is set to the number of all constellation points.

The usual sphere decoders adopt the SE enumeration [9], which sorts all of the constellation points by the Euclidean distances between the received signal y_i and itself. The point with the smallest distance has the first priority and enumerates others in ascending order of distance. Because proposed SFSD $[1, 2, 2, P]^T$ only needs to enumerate the

FIGURE 1: SFSD: $\mathbf{n} = [1, 2, 2, 4]^T$ in 4×4 MIMO QPSK system. FIGURE 2: Different SFSD distributions and optimal SD in coded 4×4 MIMO 16-QAM system.TABLE 3: Visited nodes in tree search with different fixed branches in 16-QAM modulation 4×4 MIMO system.

	[2, 3, 4, 16]	[2, 2, 2, 16]	[1, 2, 2, 16]	[1, 1, 1, 16]
Nodes	657	241	177	65

two smallest points of constellation, the SE enumeration can be simplified here. Therefore, the proposed SFSD does not need to sort all points. The following is the presentation of simplified enumeration. Figure 3 is an example of the 16-QAM constellation. The red point is the point of received signal y_i . The complex valued constellation can be separated into real part and imaginary part. The nearest point can be found by comparing with the value of the dotted lines in real part and imaginary part, respectively. Then the nearest point which is inside the green cycle is confirmed. The second nearest point of real part and imaginary part can be easily found by comparing with the values of the

FIGURE 3: Proposed simplified enumeration for the proposed SFSD.

gray lines which include the nearest point. The two points 2a and 2b inside the blue cycles are the second nearest points of real part and imaginary part, respectively. Then PED computations between those points and y_i are required. The distances between the nearest point and the received signal point in the real part and imaginary part are denoted as d_{R1} and d_{I1} , respectively. The distances between the second nearest point and the received signal point in the real part and imaginary part are denoted as d_{R2} and d_{I2} , respectively. Then, the second nearest point can be decided by $\min\{(d_{R1} + d_{I2}), (d_{R2} + d_{I1})\}$. Only some comparisons help the SFSD to achieve the SE enumeration. Note that these PED computations of real part and imaginary part are equivalent to the PED computations of two points in the complex valued constellation. Therefore, the simplified SE enumeration just costs two PED computations.

The sequential design for tree search has variable throughput due to the unpredictable number of visited nodes, which causes hardware inefficiency. The early termination (ET) [6] can solve the problem. It confines the frame run time to $N_{\text{frame}} \cdot D_{\text{avg}}$ and lets every frame have the same delay during N_{frame} MIMO detections. The function confines the maximum number of visited nodes to a value in each tree search as

$$D_{\max}(k) = N_{\text{frame}} \cdot D_{\text{avg}} - \sum_{i=1}^{k-1} D(i) - (N_{\text{frame}} - k)M, \quad (10)$$

$$k = 1, 2, \dots, N_{\text{frame}},$$

where $D(i)$ denotes the number of visited nodes for the i th MIMO symbol vector.

3.2. Simulation Results. The simulation environment is Rayleigh flat fading channel with AWGN. The algorithm is

FIGURE 4: FER of coded 4×4 MIMO detection schemes with 16-QAM.

FIGURE 5: Compared with STS-SD [6], the average number of parent nodes visited in coded 4×4 MIMO detection schemes with 16-QAM is reduced by the proposed method.

FIGURE 6: Block diagram of the proposed MIMO detector.

evaluated in terms of the number of visited parent nodes in tree search and frame-error-rate (FER) performance. All simulations are performed in a 4×4 MIMO system with Gray mapping QAM constellation modulation. In the simulations, data are coded in a rate $R = 1/2$ convolutional code with constraint length 7 and a $[133_o, 171_o]$ polynomial generator. A soft-input Viterbi decoder (max-log BCJR algorithm [12]) is employed in the receiver. A frame consists of $N_{\text{frame}} = 64$ MIMO symbols. In other words, a frame consists of randomly interleaved $N_{\text{frame}} \cdot M \cdot Q$ bits after outer encoding.

Figure 4 illustrates the frame-error-rate (FER) performance of the optimal SD, STS-SD [6], and proposed SFSD. At 2%-FER, only a 0.5-dB loss is observed between the SFSD and the optimal SD. The performance degradation is even smaller between the SFSD and the STS-SD.

Recursive computation of the PED as (8) for tree search leads to considerable computational complexity and latency. Hence, the number of visited parent nodes in tree search is an indicator for searching complexity and latency. Figure 5 shows the comparison of the average number of visited parent nodes between the STS-SD algorithm [6] and the proposed SFSD. It can be seen that the proposed method can reduce 20% of the average number of visited parent nodes. Accordingly, the low complexity advantage of the SFSD over the STS-SD can be observed.

4. Logic Design

The proposed MIMO detector, shown in Figure 6, consists of a preprocessing and an SFSD block. The preprocessing block performs sorted QR decomposition (SQRD) and FSD ordering. The matrix $\mathbf{R}$ is decomposed by SQRD and consists of real valued diagonal elements, R_r , and complex valued off-diagonal elements, R_c . The computation of $\tilde{\mathbf{y}} = \mathbf{Q}^H \cdot \mathbf{y}$ is also performed in the block. FSD ordering chooses the column with the second smallest column norm to process in each iteration of the SQRD algorithm instead of the smallest column norm. The smallest column norm of channel matrix is ordered in the top level of tree; the other columns of channel are ordered to be decreased from right to left. In other words, the second top level has the greatest column norm, and the leaf level has the second smallest column norm. The SFSD block performs the proposed SFSD algorithm and adopts proper parallel design for high throughput and full pipeline design for high clock rate. Figure 7 shows the detailed block diagram of the proposed SFSD. The SFSD block consists of several PED modules (PED4, PED3, PED2, and PED1), a list administration unit (LAU), and an LLR module. The PED4, PED3, PED2, and PED1 modules calculate PEDs for layer 4, 3, 2, and 1, respectively. The LAU module compares and sorts the Euclidean distances (EDs) which are accumulated by the PED modules. After 2^Q times (2^Q is the constellation size) of sequential update checking, the LAU module outputs the calculated local minima λ for each layer and the ML solution $\mathbf{x}^{\text{ML}}$ (Figure 7). LLR computation module substrates the smallest distance value λ_k^{ML} from the local minima distance values λ_k which corresponds to the local minima with binary complement of the k th bit of $\mathbf{x}^{\text{ML}}$ and computes each LLR

FIGURE 7: Architecture of the proposed SFSD.

FIGURE 8: Double sampling circuitry.

FIGURE 9: Timing diagram of the error detection and recovery.

value. The order of LLR values is not the same as the order of the original input data because SQRD preprocessing reorders the matrix $\mathbf{R}$. Hence, the order recovery is also implemented in the LLR computation module.

4.1. Adaptive Voltage Scaling with Double Sampling. Dynamic voltage scaling (DVS) [13] is a common technique to adjust system voltage and reduce power consumption. Some methods have been proposed for pessimistic designs that require large safety margins [14]. Instead of using guard range from the safety margin, Razor [15, 16] proposed an in situ timing error detection and correction mechanism, which can overcome variation from the PVT and significantly reduce the power consumption. In Razor-based DVS, each flip-flop in the critical path is augmented with a shadow latch, which is triggered by a delayed clock. By comparing the data sampled by the main flip-flop and the shadow latch,

a timing error can be detected and the corrupt value will be restored by the correct value in the shadow latch.

An adaptive voltage scaling with double sampling scheme is applied to the proposed SFSD. As shown in Figure 8, the main flop is clocked by clk and the delayed flop is clocked by clk_del , which is delayed with respect to clock clk . The delay between the clock edges of clk and clk_del is designed such that the correct value is obtained at the next clock edge of clk_del , even in the presence of unpredictability in the wire delay. Therefore, the delayed flop is designed to operate in an error-free manner. The XOR gate is used to compare the data captured by the main flop and delayed flop. When the outputs of the two flops differ, the signal errq is active to indicate a timing error and turn the main flop to the error mode. Afterward, the correct value captured by the delayed flop will be sent by the main flop in the next cycle, causing a one-cycle penalty for error recovery. Figure 9 shows the timing diagram of the error detection and recovery.

At each pipeline stage, an error control circuit presented in Figure 10 is added for generating suitable control signals when an error is detected. Let the number of bit-lines in the pipeline be w lines. The XOR outputs (errq signals) generated at all the w bit lines at each pipeline stage of the pipeline are ORed together and fed as an input to the error control circuit. For the Error-OR-tree, the error signals from all pipeline flip-flops are OR together to generate a unified error signal. If the design has many double sampling flip-flop, the fan-in of the OR gate can be very large, requiring OR signal to be pipelined. For error stabilization, in some cases, it is possible for error signal to become metastable. So we add two flip-flops at the global error output to overcome this problem.

4.2. Modified Cell-Based Flow. The timing paths of each submodule are analyzed after top-down synthesis. The critical paths of the target design are in PED1 and PED2 modules. Therefore, the double sampling flip-flops are inserted into the critical paths and ORed all the error signals on each double sampling pipeline stage. Figure 11 shows the block diagram of the SFSD after double sampling pipeline insertion.

5. Hardware Implementation

The proposed DVS soft output MIMO detector, using TSMC $0.18\ \mu\text{m}$ single-poly six-metal (1P6M) CMOS technology, is

FIGURE 10: Pipeline stage with double sampling techniques.

FIGURE 11: Block diagram of the SFSD modified to support DVS.

TABLE 4: ASIC Summary of The Proposed MIMO Detector.

ASIC specification		
Technology	TSMC 0.18 μm 1P6M CMOS	
Supply voltage	1.8 V	
Power consumption	64-QAM	195.8 mW at 120 MHz
	16-QAM	145.5 mW at 120 MHz
Maximum throughput	64-QAM	45 Mbps at 120 MHz
	16-QAM	120 Mbps
	QPSK	60 Mbps
Maximum clock frequency	120 MHz	
Core area	$1.16928 \times 1.17414 \text{ (mm}^2\text{)}$	

shown in Figure 12. Table 4 lists the ASIC specification. All the power and throughput information in Table 4 operate at 1.8 V and 120 MHz.

This design surpasses the previous researches [6, 17] for its high power efficiency. A comparison (with the proposed postlayout simulation results) is given in Table 5, where the

FIGURE 12: Layout.

power efficiency is normalized to mitigate the impact of different process factors and throughputs using

Normalized power efficiency

$$= \left(\text{Power} \times \left(\frac{1.8}{V_{DD}} \right)^2 \times \frac{0.18}{\text{Process}} \times \frac{1}{\text{Throughput}} \right)^{-1}. \quad (11)$$

While normalizing power efficiency using (11), the nominal V_{DD} of the proposed design is 1.8 V, instead of the scaled voltage, to show the power-saving effect of the applied voltage scaling technique.

By scaling down the supply voltage, the power consumption of the design can be reduced, as shown in Figure 13.

TABLE 5: Comparison of MIMO detector implementations.

	STS-SD [6]		MKSE [17]		SOSD [18]		This work
Fixed throughput	No	Yes	Yes	No	Yes		
Technology	0.25 μm	0.13 μm	0.35 μm	0.13 μm	0.18 μm		
Max. frequency	71 MHz	100 MHz	200 MHz	198 MHz	120 MHz ($V_{DD} = 1.8 \text{ V}$)		
Gate count	56.8 K	97 K	91 K	350 K	100.6 K		
Max. throughput (16-QAM)	28.4 Mbps (SNR = 14 dB)	106.6 Mbps	53.3 Mbps	287.9 Mbps	120 Mbps		
Complexity efficiency (16-QAM)	—	—	0.59 Mbps/K-gate	0.82 Mbps/K-gate	1.19 Mbps/K-gate		
Normalized power efficiency (16-QAM)	—	—	0.20 Mbps/mW	1.88 Mbps/mW	2.53 Mbps/mW ($V_{DD} = 1.08 \text{ V}$) (Freq. = 100 MHz)

FIGURE 13: Power reduction with the proposed DVS at 100 MHz.

The lowest supply voltage for error-free operations at 100-MHz clock rate for 64-QAM and 16-QAM modulation is 1.44 V and 1.26 V, leading to 34.7% and 49.22% power reduction, respectively. When voltage scales to 1.26 V for 64-QAM modulation or 1.08 V for 16-QAM modulation, timing error occurs on the critical path, which is pipelined by double sampling circuitry. The duplicate circuitry (delayed flip-flop) is triggered by the error signal and recovers the critical path from the timing error. While further scaling down the supply voltage, some subcritical paths may suffer from timing error and cannot be recovered. When the supply voltage decreases to 1.08 V for 64-QAM modulation, the functionality of the design fails due to incorrect data transferred by subcritical paths. With the proposed adaptive voltage scaling method with double sampling, the supply voltage can scale down to 1.08 V and 1.26 V, and the power reduction approaches 48.8% and 61.25%, for 16-QAM and 64-QAM modulation, respectively. Figure 14 shows the maximum clock rate and the current of the design in 16-QAM modulation. The power consumption ratio of each module is shown in Figure 15.

FIGURE 14: Maximum clock rate and current versus supply voltage.

FIGURE 15: The double sampling module for the proposed DVS consumes only 0.41% power of the detector.

By using the error monitoring circuit, timing error information can be reported and conveyed to the voltage controller at run time to control the voltage regulator to power up/down the system. If the supply voltage is inadequate to support required timing of critical paths, the power controller will send a decision bit to indirectly control the voltage source which is provided by a DC-DC converter. The power management of the platform is completed through these mechanisms. The double sampling scheme shows well-behaved function, provides a solution to timing error detection and recovery, and accomplishes a robust DVS platform.

6. Conclusion

A power-efficient SFSD for MIMO detection is presented in the paper. The configurable SFSD with a novel tree search algorithm achieves soft-output decoding with fixed throughput of 120 Mbps in 16-QAM modulation. The FER of the detector approaches the optimal sphere decoder, with 0.5-dB degradation. A novel adaptive voltage scaling method with double sampling circuitry provides error detection/recovery and a 48.8% power saving.

Acknowledgments

The authors would like to thank the National Chip Implementation Center (CIC) of the National Applied Research Laboratories, Taiwan, for EDA tool support, as well as fabrication and measurement of the proposed chip. They would also like to thank the anonymous reviewers and editor for their valuable suggestions, which helped them improve this paper. This research is supported in part by the National Science Council, Taiwan, under Grants NSC-99-2221-E-007-050, NSC-99-2220-E-007-024, NSC-98-2219-E-007-003, supported in part by NTHU/ITRI joint research center, and supported in part by MediaTek-NTHU Joint Lab.

References

- [1] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, 2003.
- [2] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, no. 5, pp. 463–471, 1985.
- [3] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI Implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, 2005.
- [4] J. Hagenauer and P. Hoeher, "Viterbi algorithm with soft-decision outputs and its applications," in *Proceedings of the IEEE Global Telecommunications Conference & Exhibition (GLOBECOM '89)*, pp. 1680–1686, November 1989.
- [5] B. M. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, 2003.
- [6] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, 2008.
- [7] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, Article ID 4543065, pp. 2131–2142, 2008.
- [8] L. G. Barbero, T. Ratnarajah, and C. Cowan, "A low-complexity soft-MIMO detector based on the fixed-complexity sphere decoder," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 2669–2672, April 2008.
- [9] C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Mathematical Programming, Series B*, vol. 66, no. 2, pp. 181–199, 1994.
- [10] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, and K. D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, 2001.
- [11] C. Hess, M. Wenk, A. Burg et al., "Reduced-complexity mimo detector with close-to ml error rate performance," in *Proceedings of the 17th Great Lakes Symposium on VLSI (GLSVLSI '07)*, pp. 200–203, March 2007.
- [12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.
- [13] M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, "Scaling, power, and the future of CMOS," in *Proceedings of the IEEE International Electron Devices Meeting (IEDM '05)*, pp. 9–15, December 2005.
- [14] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "Dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, 2000.
- [15] D. Ernst, N. S. Kim, S. Das et al., "Razor: a low-power pipeline based on circuit-level timing speculation," in *Proceedings of the 36th Annual International Symposium on Microarchitecture*, pp. 7–18, 2003.
- [16] S. Das, D. Roberts, S. Lee et al., "A self-tuning DVS processor using delay-error detection and correction," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, 2006.
- [17] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best Sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, 2006.
- [18] C. H. Liao, T. P. Wang, and T. D. Chiueh, "A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, Article ID 5405138, pp. 411–421, 2010.