Journal of Semiconductor Engineering Vol. 2, Issue 3, December, 2021
https://doi.org/10.22895/jse.2021.0006
Received 27 Aug 2021, Revised 27 Sep 2021, Accepted 4 Oct 2021 ISSN(On-line) 2733-6719

On the Hardware Complexity of Tree Expansion in MIMO Detection

Byeong Yong Kong[1], Youngjoo Lee[2], and Hoyoung Yoo[3]

[1] Division of Electrical, Electronic, and Control Engineering, Kongju National University, Cheonan, 31080, South Korea
[2] Department of Electrical Engineering, Pohang University of Science and Technology, Pohang, 37673, South Korea
[3] Department of Electronics Engineering, Chungnam National University, Daejeon, 34134, South Korea

Corresponding Author: Byeong Yong Kong (bykong@kongju.ac.kr)

Funding Information: This research was supported by the National Research Foundation of Korea under Grant 2020R1G1A1102620 funded by the Ministry of Science and ICT, Government of South Korea.

ABSTRACT

This paper analyzes the tree expansion for multiple-input multiple-output (MIMO) detection in the viewpoint of hardware implementation. The tree expansion is to calculate path metrics of child nodes performed in every visit to a node while traversing the detection tree. Accordingly, the tree-expansion unit (TEU), which is responsible for such a task, has been an essential component in a MIMO detector. Despite the paramount importance, the analyses on the TEUs in the literature are not thorough enough. Accordingly, we further investigate the hardware complexity of the TEUs to suggest a guideline for selection. In this paper, we focus on a pair of major ways to implement the TEU: 1) a full parallel realization; 2) a transformation of the formulae followed by common subexpression elimination (CSE). For a logical comparison, the numbers of multipliers and adders are first enumerated. To evaluate them in a more practical manner, the TEUs are implemented in a 65-nm CMOS process, and their propagation delays, gate counts, and power consumptions were measured explicitly. Considering the target specification of a MIMO system and the implementation results comprehensively, one can choose which architecture to adopt in realizing a detector.

KEY WORDS

Multiple-input multiple-output (MIMO), sphere decoding, tree expansion, very large scale integration (VLSI), wireless communications.

1. INTRODUCTION

Multiple-input multiple-output (MIMO) has been an indispensable technology in contemporary wireless communication systems as a means of enhancing the spectral efficiency and the diversity [1]. Along with the advancement of the technology, the number of antennas and the order of modulation in a MIMO system have been increased significantly, necessitating efficient hardware architectures for feasible implementation.

To acquire the maximum-likelihood (ML) estimate of the transmitted signals, the MIMO detection procedure is usually transformed to a tree-search problem, and is solved by the corresponding tree-search algorithm based on the depth-first [2]–[4], the breadth-first [5]–[14], or the metric-first [15]–[17] strategy. Regardless of the strategies, the tree expansion must be performed in every visit to a node while traversing the tree so that path metrics of child nodes can be evaluated. Accordingly, the tree-expansion unit (TEU), which is responsible for such a task, has been an essential component in a MIMO detector [18].

Despite the paramount importance, the mathematical formulae representing the tree expansion are rather simple. As a result, it received relatively little attention and had been implemented in a straightforward manner [2]–[17]. In an attempt to mitigate the hardware complexity, the equations were approximated to the l^1 and l^∞ norms [4], [5], [19], but they may cause severe deterioration of the error-rate performance. In [20], a mathematically equivalent transformation of the tree-

expansion formulae was proposed to diminish the operators by maximizing shareable subexpressions.

Although [20] paved the way to the low-complexity implementation of the TEU without resorting to the performance-degrading approximations, the analyses in [20] are not thorough enough. Accordingly, we further investigate the hardware complexity of TEU architectures to suggest a guideline for selection. In this paper, we focus on a pair of major ways to implement the TEU: 1) a fully parallel (FP) realization; 2) a combination of the formulae transformation and common subexpression elimination (CSE) [20]. For a logical comparison, the numbers of multipliers and adders are first enumerated. To evaluate them in a more practical manner, the TEUs are implemented in a 65-nm CMOS process, and their propagation delays, gate counts, and power consumptions were measured explicitly. Considering the target specification of a MIMO system and the implementation results comprehensively, one can choose which architecture to adopt. More specifically, when a high throughput is of the highest priority, we may parallelize the structure fully. If a low-power consumption is preferred, on the other hand, we may share subexpressions while sacrificing the throughput to a certain extent.

The rest of this paper is organized as follows. The fundamentals of MIMO detection and tree expansion are reviewed in Section 2. Section 3 elucidates the two TEU architectures. In Section 4, the hardware complexities of the TEUs are evaluated in a 65-nm CMOS process. Concluding remarks are made in Section 5.

2. MIMO DETECTION AND TREE EXPANSION

Figure 1 illustrates a MIMO system equipped with N_T transmit antennas and N_R receive antennas, which can be modeled as

$$\mathbf{r} = \mathbf{C}\mathbf{t} + \mathbf{a}, \tag{1}$$

where $\mathbf{r}$ is the $N_R \times 1$ received symbol vector, $\mathbf{C}$ is the $N_R \times N_T$ Rayleigh fading channel matrix, $\mathbf{t}$ is the $N_T \times 1$ transmitted symbol vector, and $\mathbf{a}$ is the $N_R \times 1$ additive white Gaussian noise vector. For the sake of simplicity in processing, the complex-valued model in (1) is frequently transformed to the real-valued counterpart by applying the real-value decomposition (RVD) [21]. Then, it is reformulated as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} = \begin{bmatrix} \mathrm{Re}(\mathbf{r}) \\ \mathrm{Im}(\mathbf{r}) \end{bmatrix} = \begin{bmatrix} \mathrm{Re}(\mathbf{C}) & -\mathrm{Im}(\mathbf{C}) \\ \mathrm{Im}(\mathbf{C}) & \mathrm{Re}(\mathbf{C}) \end{bmatrix} \begin{bmatrix} \mathrm{Re}(\mathbf{t}) \\ \mathrm{Im}(\mathbf{t}) \end{bmatrix} + \begin{bmatrix} \mathrm{Re}(\mathbf{a}) \\ \mathrm{Im}(\mathbf{a}) \end{bmatrix} \tag{2}$$

where Re(·) and Im(·) extract the real and the imaginary components, respectively. The dimensions of $\mathbf{y}$, $\mathbf{H}$, $\mathbf{x}$, and $\mathbf{n}$ are $2N_R \times 1$, $2N_R \times 2N_T$, $2N_T \times 1$, and $2N_R \times 1$, respectively.

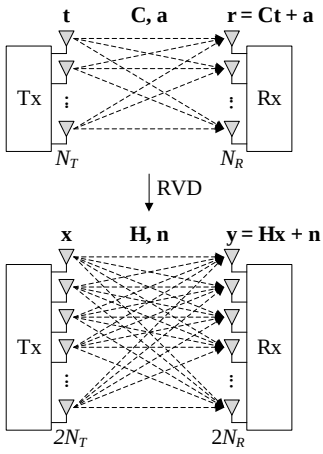

Figure 1. A complex-valued $N_T \times N_R$ MIMO communication system and a real-valued $2N_T \times 2N_R$ system equivalent to each other.

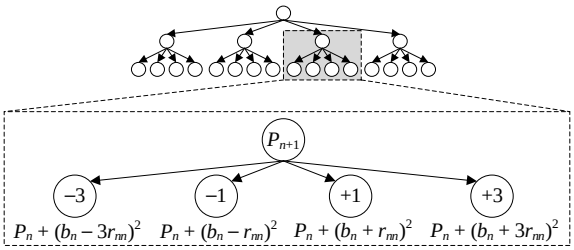

Figure 2. Tree expansion for 16-QAM.

The objective of MIMO detection is to identify $\mathbf{x}$ from $\mathbf{y}$ and $\mathbf{H}$ as precisely as possible. The optimal estimate without *a priori* statistics, i.e., the ML solution, is denoted as $\mathbf{x}_{\mathrm{ML}}$. It can be obtained by solving

$$\mathbf{x}_{\mathrm{ML}} = \arg\min_{\mathbf{z} \in O^{2N_T}} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|^2, \tag{3}$$

where $\mathbf{O}$ is a set of alphabets in real-valued constellation of a modulation scheme such as phase shift keying and quadrature amplitude modulation (QAM). In case of 16-QAM, $\mathbf{O}$ ={–3, –1, +1, +3}. The QR decomposition to $\mathbf{H}$ modifies (3) as

$$\mathbf{x}_{\mathrm{ML}} = \arg\min_{\mathbf{z} \in O^{2N_T}} \|\mathbf{y} - \mathbf{Q}\mathbf{R}\mathbf{z}\|^2 = \arg\min_{\mathbf{z} \in O^{2N_T}} \|\hat{\mathbf{y}} - \mathbf{R}\mathbf{z}\|^2 \tag{4}$$

where $\mathbf{Q}$ is an orthogonal and unitary matrix, and $\mathbf{R}$ is an upper triangular matrix. $\hat{\mathbf{y}}$ is $\mathbf{Q}^H \times \mathbf{y}$ where $\mathbf{Q}^H$ is the Hermitian transpose of $\mathbf{Q}$. The objective function in (4) is the l^2-norm or the Euclidean distance, which can be recursively calculated as

$$P_n = P_{n+1} + (b_n - r_{nn} z_n)^2, \tag{5}$$

where P_n is the partial Euclidean distance at the nth layer. In (5),

$$b_n = \hat{y}_n - \sum_{i=n+1}^{N} r_{ni} z_i, \tag{6}$$

where $\hat{y}_n$ is the nth component of $\hat{\mathbf{y}}$, r_{ij} is the element of $\mathbf{R}$ at the ith row and the jth column, and z_i is the ith component of $\mathbf{z}$. Starting from the root node associated with $P_{2N+1} = 0$, (5) is recursively accumulated until it reaches the final cost P_1 of a leaf node.

In the description above, the tree expansion is to calculate $(b_n - r_{nn}z_n)^2$ for every $z_n \in \mathbf{O}$. Figure 2 exemplifies the expansion for 16-QAM. Each parent node spawns $|\mathbf{O}| = 4$ child nodes, where $|\mathbf{O}|$ is the cardinality or the number of elements in $\mathbf{O}$. For every $z_n \in \mathbf{O} = \{-3, -1, +1, +3\}$, $(b_n - r_{nn}z_n)^2$ is evaluated to be added to P_{n+1}.

3. TREE EXPANSION ARCHITECTURES

We now delve into the two types of TEUs available in the literature mentioned in Section 1: 1) the FP TEU and 2) the CSE TEU [20]. The left-hand side (LHS) of Figure 3 enumerates all the formulae associated with the tree expansion for 256-QAM. Figure 4(a) implements the LHS of Figure 3 in a FP manner. The minus sign at an input of an adder indicates that the input is negated before taking summation. Besides, the multiples of r_{nn}, which depend only on $\mathbf{H}$, are not calculated on-the-fly within the TEU, but are provided by the channel estimator as they change occasionally. If only r_{nn} is given, its multiples can be synthesized by solving the multiple constant multiplication (MCM) problem based on the shift-and-add approach [22], [23]. In computing $(b_n - r_{nn}z_n)^2$ for one $z_n \in \mathbf{O}$, one adder and one multiplier are employed. Due to the full parallelization, in total, $|\mathbf{O}|$ adders and $|\mathbf{O}|$ multipliers are required to build an entire TEU for $|\mathbf{O}|^2$-QAM.

While Figure 4(a) is straightforward and works perfectly, it incorporates $|\mathbf{O}|$ multipliers, which are costly. To maximize shareable subexpressions, [20] manipulates the equations as formulated in the RHS of Figure 3, where $(p << q)$ denotes left-shifting p by q bits. In other words, $(p << q) = p \times 2^q$. The RHS is mathematically equivalent to the LHS, and, therefore, does not degrade the error-rate performance at all [20]. Note that the RHS formulae have $(b_n + r_{nn})^2$ in common. By sharing it for all $z_n \in \mathbf{O}$, we can eliminate most of the multipliers in Figure 4(a). The hardware architecture realizing the RHS is depicted in Figure 4(b), where the minus sign and the number at an input of an adder represent the negation and the left-shift amount applied to the input before summation, respectively. It is worth noting that the shift operation by a constant amount does not incur any hardware overhead, as it can be easily realized by hard wiring. The multiples of r_{nn}^2 are provided by the channel estimator, as those of r_{nn} are so in Figure 4(a). A set of an adder and a multiplier colored red in Figure 4(b) solely take care of the terms colored the same in Figure 3. In a similar manner, $b_n r_{nn}$ colored blue in Figures 3 and 4(b) are shared among

$$
\begin{aligned}
(b_n + r_{nn})^2 &= (b_n + r_{nn})^2 \\
(b_n - r_{nn})^2 &= (b_n + r_{nn})^2 - (b_n r_{nn} << 2) \\
(b_n + 3r_{nn})^2 &= (b_n + r_{nn})^2 + (r_{nn}^2 << 3) + (b_n r_{nn} << 2) \\
(b_n - 3r_{nn})^2 &= (b_n + r_{nn})^2 + (r_{nn}^2 << 3) - (b_n r_{nn} << 3) \\
(b_n + 5r_{nn})^2 &= (b_n + r_{nn})^2 + (3r_{nn}^2 << 3) + (b_n r_{nn} << 3) \\
(b_n - 5r_{nn})^2 &= (b_n + r_{nn})^2 + (3r_{nn}^2 << 3) - (3b_n r_{nn} << 2) \\
(b_n + 7r_{nn})^2 &= (b_n + r_{nn})^2 + (3r_{nn}^2 << 4) + (3b_n r_{nn} << 2) \\
(b_n - 7r_{nn})^2 &= (b_n + r_{nn})^2 + (3r_{nn}^2 << 4) - (b_n r_{nn} << 4) \\
(b_n + 9r_{nn})^2 &= (b_n + r_{nn})^2 + (5r_{nn}^2 << 4) + (b_n r_{nn} << 4) \\
(b_n - 9r_{nn})^2 &= (b_n + r_{nn})^2 + (5r_{nn}^2 << 4) - (5b_n r_{nn} << 2) \\
(b_n + 11r_{nn})^2 &= (b_n + r_{nn})^2 + (15r_{nn}^2 << 3) + (5b_n r_{nn} << 2) \\
(b_n - 11r_{nn})^2 &= (b_n + r_{nn})^2 + (15r_{nn}^2 << 3) - (3b_n r_{nn} << 3) \\
(b_n + 13r_{nn})^2 &= (b_n + r_{nn})^2 + (21r_{nn}^2 << 3) + (3b_n r_{nn} << 3) \\
(b_n - 13r_{nn})^2 &= (b_n + r_{nn})^2 + (21r_{nn}^2 << 3) - (7b_n r_{nn} << 2) \\
(b_n + 15r_{nn})^2 &= (b_n + r_{nn})^2 + (7r_{nn}^2 << 5) + (7b_n r_{nn} << 2) \\
(b_n - 15r_{nn})^2 &= (b_n + r_{nn})^2 + (7r_{nn}^2 << 5) - (b_n r_{nn} << 5)
\end{aligned}
$$

Figure 3. Manipulation of the tree-expansion formulae for 256-QAM and common subexpressions therein.

the formulae to minimize the multipliers. The remaining subexpressions colored green are shared as well to remove adders as much as possible. Unlike the multiples of r_{nn}^2, those of $b_n r_{nn}$ cannot be precomputed by the external channel estimator due to b_n. Therefore, they are synthesized and shared within the TEU.

While Figure 4 sketches the TEUs for 256-QAM, the TEUs for 16- and 64-QAMs can be easily derived from the figure. A set of components placed above the upper horizontal dashed line corresponds to the TEU for 16-QAM. A set of those above the lower dashed line is capable of accommodating 64-QAM. As shown, both TEUs scale favorably with the increasing order of modulation.

4. ANALYSES ON HARDWARE COMPLEXITY

This section analyzes the hardware complexity of the TEUs for 16-, 64-, and 256-QAMs, which are the most prevalently employed modulations in practice. In Table 1, the total numbers of operators in a TEU are summarized. Let us assume that b_n and r_{nn} are quantized in W bits. Then, all the adders and multipliers in Figure 4(a) are W-bit adders and W-bit multipliers, respectively. Accordingly, the total number of W-bit adders and that of W-bit multipliers in a FP TEU are equal to $|\mathbf{O}|$. There is no $2W$-bit operator. On the other hand, the CSE TEU includes one W-bit adder and two W-bit multipliers. In addition, since the product of a W-bit multipliers is in $2W$ bits, the remaining adders are all $2W$-bit adders. Therefore, the CSE TEUs for 16-, 64-, and 256-QAMs have 4, 11, and 25 $2W$-bit adders, respectively. Owing to the CSE, the CSE TEUs apparently have fewer multipliers than the FP counterparts. In exchange for such a benefit, they must undertake several $2W$-bit adders, which are more expensive than W-bit adders.

Table 1 also enumerates the numbers of operators in the critical paths, which are colored red in Figure 4. The longest path in the FP TEU always consists of one

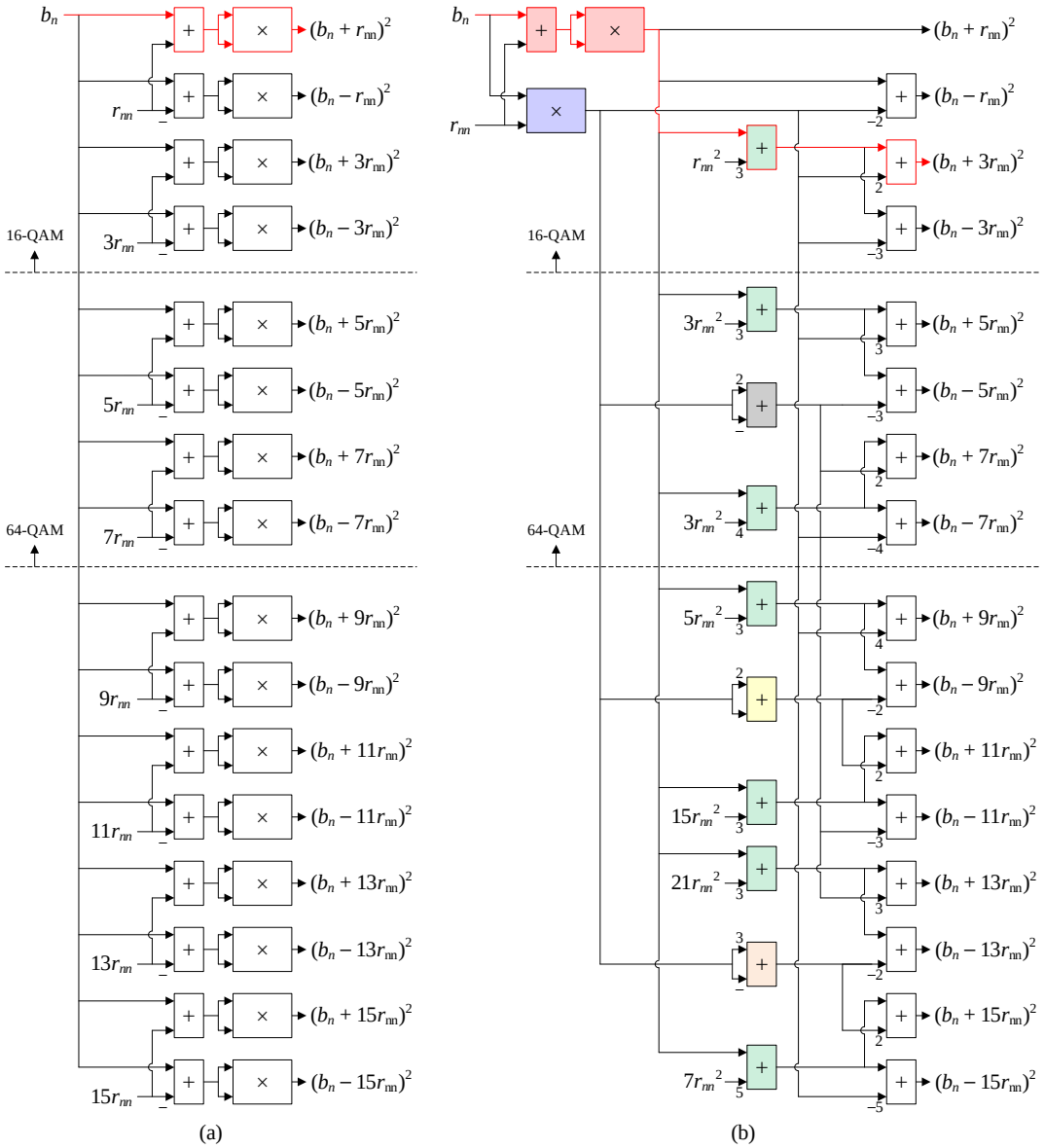

Figure 4. TEUs for 256-QAM. (a) FP implementation. (b) CSE implementation.

adder and one multiplier regardless of the order of modulation. In contrast, a signal in the CSE TEU undergoes one *W*-bit adder, one *W*-bit multiplier, and two 2*W*-bit adders at most, which is longer than in the FP TEU. In short, the CSE decreases the number of operators, but increases the number of stages to propagate when applied to the TEU.

While Table 1 is logically correct, actual latencies in the realized circuitry can be different due to many practical factors of implementation such as fan-in, fan-out, and gate sizing. To evaluate more realistically by taking into account such factors, the TEUs were first described in Verilog hardware description language (HDL), and then were synthesized by Synopsys Design Compiler in a 65-nm CMOS process. The corresponding results are summarized in Table 2. Equivalent gates were counted by regarding a two-input NAND as one. Power consumptions were measured by back-annotating switching activities. For the sake of comprehensive evaluation, the figure of merit is defined to be the product of the equivalent-gate count and the critical-path delay. It is worth noting that the implementation results may vary with the methodologies employed. However, since the synthesizer optimizes logics comprehensively, the variation is usually not significant at the end. The results in Table 2 are obtained from such quasi-optimal designs.

Although the numbers of operators in the critical paths are identical for all constellations in Table 1, the actual path delays in Table 2 are indeed different to each other mainly due to the large fan-out of b_n that feeds all the adders in parallel. The higher the order of constellation is, the higher the parallel factor is, the larger the fan-out is, and the longer the path delay is.

Table 1. Numbers of Operators

Constellation		16-QAM		64-QAM		256-QAM	
TEU Architecture		FP	CSE	FP	CSE	FP	CSE
Total Operators in a TEU	*W*-Bit Multipliers	4	2	8	2	16	2
	W-Bit Adders	4	1	8	1	16	1
	2*W*-Bit Adders	0	4	0	11	0	25
Operators in the Critical Path	*W*-Bit Multipliers	1	1	1	1	1	1
	W-Bit Adders	1	1	1	1	1	1
	2*W*-Bit Adders	0	2	0	2	0	2

Table 2. Implementation Results in a 65-nm CMOS

Constellation	16-QAM		64-QAM		256-QAM	
TEU Architecture	FP	CSE	FP	CSE	FP	CSE
Critical-Path Delay (ns)	1.36 (1.00)	1.53 (1.13)	1.4 (1.00)	1.64 (1.17)	1.44 (1.00)	1.68 (1.17)
Equivalent Gate	10.50k (1.00)	10.94k (1.04)	17.78k (1.00)	17.02k (0.96)	35.18k (1.00)	27.28k (0.78)
Power (mW)	4.11 (1.00)	3.92 (0.95)	6.29 (1.00)	6.23 (0.99)	12.9 (1.00)	10.12 (0.78)
Figure of Merit	14275 (1.00)	16743 (1.17)	24890 (1.00)	27909 (1.12)	50652 (1.00)	45831 (0.90)

Besides, in accordance with Table 1, the CSE TEUs have longer critical-path delays than the FP TEUs. Specifically, the propagation in the critical path of the CSE TEU takes about 15% longer time than that of the FP TEU. Besides, the FP TEU for 16-QAM integrates slightly fewer equivalent gates than the CSE counterpart. As the order of modulation increases, however, the CSE TEU relies on fewer gates, implying that the CSE is getting more effective. In particular, the CSE TEU for 256-QAM has 22% fewer gates. Putting them altogether, we can confirm that the CSE indeed elongates the propagation delay but diminishes the hardware resources. In terms of the power dissipation, the CSE TEU is always more efficient than the FP ones. Since the delay and the gate count are in a trade-off, the figure of merit can serve as a comprehensive measure for evaluating the TEUs. In 16- and 64-QAMs, the FP ones are better, whereas the opposite holds for 256-QAM. The simple and straightforward structure of the FP TEU results in better measures than the CSE TEU for the low-order constellations. As the order of modulation increases, however, a significant number of multipliers are evicted by the CSE, and 256-QAM is the lowest-order constellation in which the effect of CSE becomes dominant enough to make the CSE TEU superior than the FP TEU.

5. CONCLUSION

The two types of the TEUs, i.e., the FP and the CSE architectures, have been analyzed in a 65-nm CMOS technology. For 16- and 64-QAMs, the FP TEU is superior to the CSE counterpart in terms of the critical-path delay multiplied by the equivalent gate count. On the contrary, in the modulation of the highest order considered, i.e., 256-QAM, the CSE TEU outperforms the FP TEU. Such results are originated from the CSE that becomes more effective for higher-order modulations. Grounded on the analyses in Section 4, we may select a TEU architecture that suits the best to a certain specification.

ACKNOWLEDGMENT

The authors would like to thank the IC Design Education Center (IDEC) of South Korea for supporting electronic design automation (EDA) tools.

REFERENCES

[1] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[2] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Atlanta, GA, USA, Mar. 2004, pp. 1620–1625.

[3] J. Jaldén and B. Ottersten, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Atlanta, GA, USA, Mar. 2004, pp. 1620–1625.

[4] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.

[5] P.-Y. Tsai, W.-T. Chen, X.-C. Lin, and M.-Y. Huang, "A 4 × 4 64-QAM reduced-complexity K-best MIMO detector up to 1.5 Gbps," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Paris, France, May 2010, pp. 3953–3956.

[6] T.-H. Kim and I.-C. Park, "Small-area and low-energy *K*-best MIMO detector using relaxed tree expansion and early forwarding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 10, pp. 2753–2761, Oct. 2010.

[7] M. Shabany and P. G. Gulak, "A 675 Mbps, 4 × 4 64-QAM *K*-best MIMO detector in 0.13 um CMOS," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 1, pp. 135–147, Jan. 2012.

[8] T. H. Tran, H. Ochi, and Y. Nagao, "A 2D sorter-based *K*-best algorithm for high order modulation MIMO systems," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Vancouver, BC, Canada, Sep. 2014, pp. 1–5.

[9] T. H. Tran, H. Ochi, and Y. Nagao, "A 4 × 4 multiplier-divider-less *K*-best MIMO decoder up to 2.7 Gbps," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Melbourne, VIC, Australia, Jun. 2014, pp. 1696–1699.
[10] C.-F. Liao, J.-Y. Wang, and Y.-H. Huang, "A 3.1 Gb/s 8 × 8 sorting reduced *K*-best detector with lattice reduction and QR decomposition," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 12, pp. 2675–2688, Dec. 2014.
[11] 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. (ISCAS)*, Island of Kos, Greece, May 2006, pp. 1151–1154.
[12] M.-T. Shiue, S.-S. Long, C.-K. Jao, and S.-K. Lin, "Design and implementation of power-efficient *K*-best MIMO detector for configurable antennas," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 11, pp. 2418–2422, Nov. 2014.
[13] 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.
[14] B. Y. Kong and I.-C. Park, "Improved sorting architecture for *K*-best MIMO detection," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 9, pp. 1042–1046, Sep. 2017.
[15] W. Xu, Y. Wang, Z. Zhou, and J. Wang, "A computationally efficient exact ML sphere deocder," in *Proc. Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dallas, TX, USA, Nov.–Dec. 2004, pp. 2594–2598.
[16] B. Y. Kong and I.-C. Park, "Fast detection for spatial modulation MIMO based on cost estimation," *Electron. Lett.*, vol. 52, no. 8, pp. 671–673, Apr. 2016.
[17] T.-H. Kim and I.-C. Park, "High-throughput and area-efficient MIMO symbol detection based on modified Dijkstra's search," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1756–1766, Jul. 2010.
[18] B. Y. Kong and I.-C. Park, "Interference cancellation architecture for pipelined parallel MIMO detectors," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Bordeaux, France, Dec. 2018, pp. 77–80.
[19] B. Y. Kong and I.-C. Park, "Adaptive metric calculation for improving detection capability of MIMO detectors," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Dresden, Germany, Jun. 2013, pp. 1–5.
[20] B. Y. Kong and I.-C. Park, "Hardware-efficient tree expansion for MIMO symbol detection," *Electron. Lett.*, vol. 49, no. 3, pp. 226–228, Jan. 2013.
[21] L. Azzam and E. Ayanoglu, "Reduced complexity sphere decoding for square QAM via new lattice representation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Nov. 2007, pp. 4242–4246.
[22] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 10, pp. 677–688, Oct. 1996.
[23] B. Y. Kong and I.-C. Park, "FIR filter synthesis based on interleaved processing of coefficient generation and multiplier-block synthesis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 31, no. 8, pp. 1169–1179, Aug. 2012.

AUTHOR BIOGRAPHIES

Byeong Yong Kong received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2011, 2013, and 2017, respectively.

Since 2019, he has been an Assistant Professor with the Division of Electrical, Electronic, and Control Engineering, Kongju National University (KNU), Cheonan, South Korea. Prior to joining KNU, from 2017 to 2018, he was a Senior Researcher with the Agency for Defense Development (ADD), Daejeon, where he was involved in the research of guided missile systems. From 2018 to 2019, he was a Postdoctoral Researcher and a Research Assistant Professor with KAIST. His current research interests include algorithms and very-large-scale integration (VLSI) architectures for wireless communications and digital signal processing.

Dr. Kong was a recipient of the 1st Place Award at the Altera FPGA Design Contest in 2015.

Youngjoo Lee received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2008, 2010, and 2014, respectively.

Since 2017, he has been with the Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, South Korea, where he is currently an Associate Professor. Prior to joining POSTECH, from 2014 to 2015, he was with the Interuniversity Microelectronics Center (IMEC), Leuven, Belgium, where he was involved in the research of reconfigurable SoC platforms for software-defined radio systems. From 2015 to 2017, he was with the Faculty of the Department of Electronic Engineering, Kwangwoon University, Seoul, South Korea. His current research interests include algorithms and architectures for embedded processors, intelligent mobile multimedia systems, advanced error-correcting codes, and next-generation communication systems.

Hoyoung Yoo received the B.S. degree in electrical and electronics engineering from Yonsei University, Seoul, South Korea, in 2010, and the M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012 and 2016, respectively.

Since 2016, he has been with the Department of Electronics Engineering, Chungnam National University (CNU), Daejeon, where he is currently an Associate Professor. Prior to joining CNU, in 2016, he was with Samsung Electronics, Hwasung, South Korea, where he was involved in the research of nonbinary LDPC decoders for NAND flash memories. His current research interests include algorithms and architectures for error-correcting codes, FPGA reverse engineering, GNSS communication, and 5G communication systems.