A Parallel Radix-Sort-Based VLSI Architecture for Finding the First W Maximum/Minimum Values

A parallel radix-sort-based VLSI architecture for finding the first W maximum/minimum values

Guoping Xiao, Maurizio Martina, *Member IEEE*, Guido Masera, *Senior Member IEEE*, Gianluca Piccinini

***Abstract*—VLSI architectures for finding the first W ($W > 2$) maximum (or minimum) values are required in the implementation of several applications such as non-binary LDPC decoders, K-best MIMO detectors and turbo product codes. In this work a parallel radix-sort-based VLSI architecture for finding the first W maximum (or minimum) values is proposed. The described architecture, named Bit-Wise-And (BWA) architecture, relies on analyzing input data from the most significant bit to the least significant one, with very simple logic circuits. One key feature in the BWA architecture is its high level of scalability which enables the adoption of this solution in a large spectrum of applications, corresponding to large ranges for both W and the size of the input data set. Experimental results, achieved implementing the proposed architecture on a high speed 90 nm CMOS standard-cell technology, show that BWA architecture requires significantly less area than other solutions available in the literature, i.e. less than or about 50% in all the considered cases. Moreover, the BWA architecture exhibits the lowest area-delay product among almost all considered cases.**

***Index Terms*—maximum values generator, Non-binary LDPC decoder, MIMO decoder, turbo-product-codes decoder, sorting, selection network.**

I. Introduction

Sorting is a well-established problem in computer science [1] and is a key operation in several applications. Besides, hardware implementation of sorting networks has been addressed as well in the past [1]. On the other hand, VLSI architectures for partial sorting, which can also be derived from selection networks (SN) [2], are part of different algorithms in the communication field. Partial sorting is employed, for example, in [3], [4] and [5], [6] for the decoding of turbo and binary Low-Density-Parity-Check (LDPC) codes, in [7] for maximum-likelihood decoding of arithmetic codes and in [8]–[10] for K-best MIMO detectors, non-binary LDPC decoders and turbo product codes respectively. Circuits for finding the first two minimum values, are used in binary LDPC decoder architectures [11]–[13] to implement $\min$-*sum* approximations [6], [14] and recently they have also been proposed for the case of non-binary LDPC decoders [15]. However, very few works, e.g. [16], [17], investigate the general problem of implementing parallel architectures for finding the first two maximum/minimum values with a systematic approach. Similarly, architectures for finding the first $W > 2$ maximum/minimum values in a set of M elements, with $W \leq M/2$, are designed in VLSI implementations of i) K-best MIMO detectors [18], [19], ii) non-binary LDPC decoders [20], [21], iii) turbo product codes [22]. Unfortunately, to the best of our knowledge, no papers in the open literature present a general analysis for the case $W > 2$.

Stemming from the work described in [2] for sorting networks, in [1] a comparator-based SN is proposed. However, as argued in [1], other approaches, such as the one referred to as radix sorting, can be used as well. Radix sorting algorithms rely on the bit-wise analysis of the data to be sorted and can be extended to selection and partial sorting problems. This paper proposes a parallel VLSI architecture relying on the radix sorting approach for finding the first $W > 2$ maximum/minimum values in a set of M values. Namely, the proposed solution, referred to as Bit-Wise-And (BWA) architecture, works by analyzing the M candidates from the Most-Significant-Bit (MSB) to the Least-Significant-Bit (LSB).

The rest of this paper is structured as follows. The problem formulation and the proposed architecture are described in Section II. In particular, in Section II.A an initial version of the BWA architecture is presented. Then, the complete version is developed in Section II.B. Section III deals with performance results and comparisons. Finally, in Section IV conclusions are drawn.

II. Problem formulation and proposed architecture

According to [17], we can state the problem of finding the first W maximum/minimum values as follows. Given a set $\mathcal{X}^{(M)} = \{x_0, \ldots, x_{M-1}\}$ made of M elements, we want to find the first W maximum/minimum values, namely $\mathbf{y}^{(M)} = \{y_0^{(M)}, y_1^{(M)}, \ldots, y_q^{(M)}, \ldots, y_{W-1}^{(M)}\}$ where $y_0^{(M)} = \max(\mathcal{X}^{(M)})$, $y_1^{(M)} = \max(\mathcal{X}^{(M)} \setminus \{y_0^{(M)}\})$, $\ldots$, $y_q^{(M)} = \max(\mathcal{X}^{(M)} \setminus \bigcup_{k=0}^{q-1}\{y_k^{(M)}\})$, $\ldots$, $y_{W-1}^{(M)} = \max(\mathcal{X}^{(M)} \setminus \bigcup_{k=0}^{W-2}\{y_k^{(M)}\})$ (similarly substituting $\max$ with $\min$). For the sake of simplicity in the following we will discuss only the $\max$ case.

A. Initial BWA architecture description

Radix sorting relies on the analysis of $\mathcal{X}^{(M)}$ values bit by bit from the MSB to the LSB. In the following, for the sake of simplicity, we will assume that the values in $\mathcal{X}^{(M)}$ are non-negative. It is worth noting, that 2's complement values can be straightforwardly used as well. Indeed, any set of N-bit 2's complement values can be converted in non-negative values, preserving the order relation, by flipping

The authors are with the Electronics and Telecommunications Department - Politecnico di Torino - Italy. This work is partially supported by the NEWCOM# NoE.

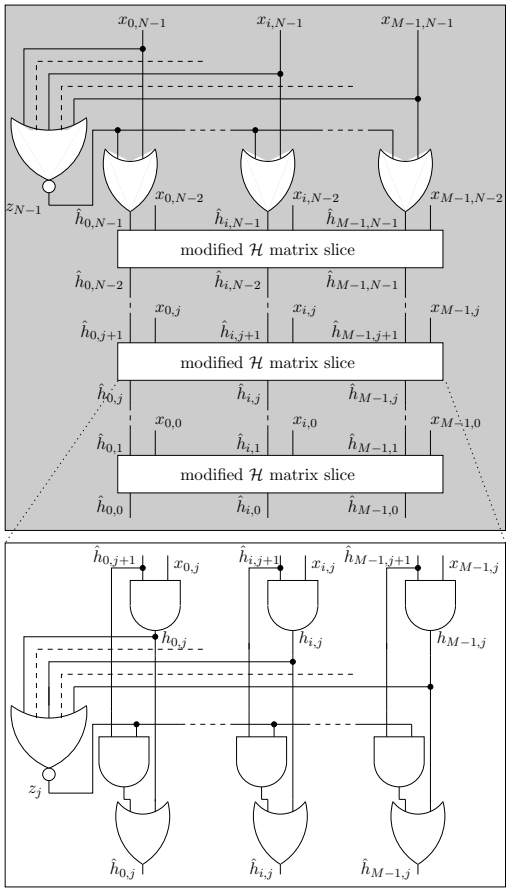

Figure 1: BWA architecture: modified $\mathcal{H}$ matrix.

the MSB. Thus, let $x_a = \{x_{a,N-1}x_{a,N-2}\dots x_{a,1}x_{a,0}\}$ and $x_b = \{x_{b,N-1}x_{b,N-2}\dots x_{b,1}x_{b,0}\}$ be two N-bit non-negative binary values, where $x_{a,j}$ and $x_{b,j}$ represent the j-th bit of x_a and x_b respectively. Assuming the first $(N-j-1)$-th MSBs of x_a and x_b have the same value, we can easily obtain the relationship between x_a and x_b based on bit-wise analysis: $x_a > x_b$ if $x_{a,j}$ = '1' and $x_{b,j}$ = '0', and viceversa.

The proposed BWA relies on performing recursively the logic-and operation between adjacent bits of each x_i value from the MSB to the LSB. Let $h_i = \{h_{i,N-2}\dots h_{i,0}\}$ be the array of bit-wise logic-and operations on x_i, where $h_{i,N-2} = x_{i,N-1} \wedge x_{i,N-2}$ and

$$h_{i,j} = h_{i,j+1} \wedge x_{i,j}, \tag{1}$$

for $j = N-3,\dots,0$ with $\wedge$ representing the logic-and operation. If the MSB of all the x_i values is '1' and all the x_i are *monotonic* sequences of bits, that is only a transition from '1' to '0' is allowed as in the four x_i values of Example 1, then, analyzing the content of h_i for $i = 0,\dots M-1$ from the LSB to the MSB allows to find the first W maximum values.

Example 1:

$$\begin{array}{l} x_0 = \{1\ 1\ 1\ 1\} \\ x_1 = \{1\ 1\ 0\ 0\} \\ x_2 = \{1\ 0\ 0\ 0\} \\ x_3 = \{1\ 1\ 1\ 0\} \end{array} \qquad \mathcal{H} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

As an example, $h_{i,0}$ = '1' if and only if $x_{i,j}$ = '1' for every $j = 0,\dots,N-1$, namely $x_i = 2^N - 1$. Let $\mathcal{H}$ be the $(N-1)\times M$ matrix whose columns are h_i. If $\mathcal{X}^{(M)}$ contains only distinct elements: i) moving from the MSB-row to the LSB-row all rows of $\mathcal{H}$ are different up to a certain j, then for $j' < j$ all the rows are zero ($j = 0$ in Example 1), ii) when moving from the LSB-row to the MSB-row, after the first non-zero row, one additional '1' appears along a column. As a consequence, moving from the LSB-row to the MSB-row of $\mathcal{H}$, the columns of the first W non-zero rows are the positions of the first W maximum values. Since in general x_i is not a monotonic sequence and repeated elements can exist in $\mathcal{X}^{(M)}$, modifications to effectively employ the BWA technique are required.

B. Completed BWA architecture

As highlighted in Section II-A, the initial BWA principle can be employed on data that are monotonic sequences of bits whose MSB is '1'. If the data in $\mathcal{X}^{(M)}$ do not meet this requirement, the architecture does not work correctly. As an example, the case $x_{i,j}$ = '0' for a certain j and for every $i = 0,\dots,M-1$ causes $h_{i,j'}$ = '0' for every $j' \le j$. In this case, the architecture can not distinguish among different x_i. A similar problem arises if two or more x_i values are non-monotonic sequences of bits. Thus, we add some gates to handle these cases, referred to as *zero-row conditions*. To this purpose we modify (1) as $h_{i,j} = \hat{h}_{i,j+1} \wedge x_{i,j}$ where

$$\hat{h}_{i,j} = \begin{cases} z_{N-1} \vee x_{i,N-1} & \text{if } j = N-1 \\ (z_j \wedge \hat{h}_{i,j+1}) \vee h_{i,j} & \text{if } 0 \le j < N-1 \end{cases}, \tag{2}$$

$\vee$ is the logic-or operation and

$$z_j = \begin{cases} \text{not}\left(\bigvee_{i=0}^{M-1} x_{i,N-1}\right) & \text{if } j = N-1 \\ \text{not}\left(\bigvee_{i=0}^{M-1} h_{i,j}\right) & \text{if } 0 \le j < N-1 \end{cases} \tag{3}$$

detects a zero-row condition. Follows an example.

Example 2:

$$\begin{array}{ll} x_0 = \{0\ 1\ 1\ 1\} & z_3 = 1 \\ x_1 = \{0\ 1\ 0\ 1\} & z_2 = 0 \\ x_2 = \{0\ 0\ 0\ 0\} & z_1 = 0 \\ x_3 = \{0\ 1\ 1\ 0\} & z_0 = 0 \end{array} \qquad \hat{\mathcal{H}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Example 2 shows a simple case, where the modified $\mathcal{H}$ matrix ($\hat{\mathcal{H}}$), that is an $N \times M$ matrix, is given. Indeed, as explained in the next paragraphs, the maximum values are selected checking $\hat{h}_{i,0}$ values. Handling zero-rows leads to the slice-architecture depicted in light gray in Fig. 1, where each slice corresponds to one row of $\hat{\mathcal{H}}$. The bottom part of Fig. 1 shows the circuit to implement (2) and (3), where $\hat{h}_{i,j}$ acts as $h_{i,j}$, but, if a zero-row condition occurs, then $\hat{h}_{i,j} = \hat{h}_{i,j+1}$. As it can be observed in the modified $\mathcal{H}$ matrix, the proposed structure ensures $\hat{h}_{i,0}$ = '1' for at least one value of $i = 0,\dots M-1$. Thus, the selection of the maximum values in the proposed architecture is performed checking $\hat{h}_{i,0}$ values. Let $\mathcal{I}$ be the set of indices $i = 0,\dots,M-1$ such that $\hat{h}_{i,0}$ = '1'. If $\mathcal{I} = \{k\}$, i.e. it contains only one element, then $y_0 = x_k$. Otherwise, $\mathcal{X}^{(M)}$ contains more instances of the

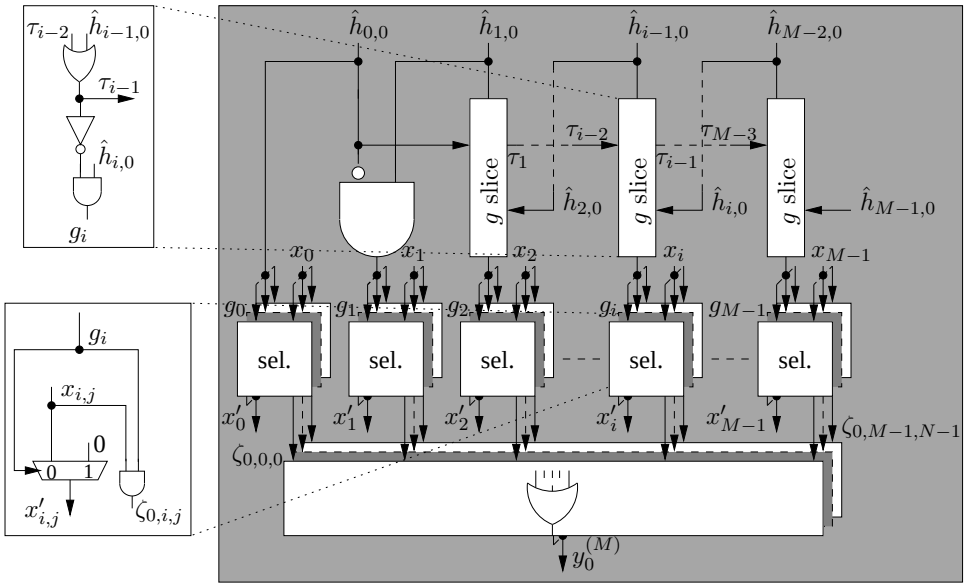

Figure 3: BWA architecture: output generation circuit in the case $q = 0$.

maximum value. If $\mathcal{I}$ contains W elements, then the first W maximum values are the elements $\{x_i : i \in \mathcal{I}\} \subset \mathcal{X}^{(M)}$. If $\mathcal{I}$ contains less than W elements a new search is required.

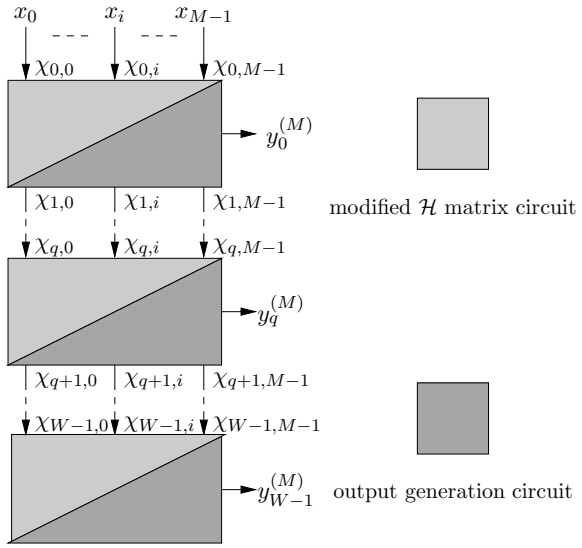

Figure 2: BWA architecture: cascade of W stages.

To simplify the selection we use a circuit referred to as *output generation circuit* that, based on $\hat{h}_{i,0}$ values, is able to find the maximum among M elements and to produce a new set of M elements $\mathcal{X}' = x'_0, \ldots, x'_{M-1}$ where the maximum value is replaced by zero. Thus, the complete architecture, shown in Fig. 2, is made of W stages, where each stage contains one instance of the circuit to produce the modified $\mathcal{H}$ (light gray part) and one instance of the output generation circuit (dark gray part). As a consequence, the q-th stage finds $y_q^{(M)}$, that corresponds to the maximum value of the q-th input set. This operation is accomplished by the means of the output generation circuit shown in dark gray in Fig. 3 for the case $q = 0$. The output generation circuit relies on $M - 1$ blocks referred to as g slice, $M \times N$ selection blocks and N combiners each made of an M-input logic-or, where M selection signals $g_i = \mathrm{not}(\tau_{i-1}) \wedge \hat{h}_{i,0}$ for $i = 1, \ldots, M-1$ and $g_0 = \hat{h}_{0,0}$ are used in the selection blocks to forward x_i to the next slice or to replace it by zero. Each g_i signal is implemented as a slice (see the top left part of Fig. 3), where the τ_{i-1} term is obtained as

$$\tau_{i-1} = \bigvee_{u=0}^{i-1} \hat{h}_{u,0}, \tag{4}$$

that is: when $\hat{h}_{u,0} =$ '1', then the remaining τ_l with $l = u+1, \ldots, M-1$ are '1' and so in the current stage only x_u is selected. More precisely, with reference to Fig. 3, g_i is exploited in the selection blocks to compute $\zeta_{q,i}$, one of the M candidates, where only one $\zeta_{q,i} \neq 0$. Each bit of $\zeta_{q,i}$ is computed as $\zeta_{q,i,j} = g_i \wedge \chi_{q,i,j}$ where

$$\chi_{q,i} = \begin{cases} x_i & \text{if } x_i \notin \bigcup_{k=0}^{q-1} \{y_k^{(M)}\} \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

and the terms $y_{q,j}^{(M)}$ and $\chi_{q,i,j}$ represent the j-th bit of $y_q^{(M)}$ and $\chi_{q,i}$ respectively (see the sel. block in the left part of Fig. 3). As an example, for $q = 0$ and $q = 1$ we have $\chi_{0,i} = x_i$ and $\chi_{1,i} = x'_i$ respectively. Finally, the q-th maximum is obtained:

$$y_q^{(M)} = \bigvee_{i=0}^{M-1} \zeta_{q,i}, \tag{6}$$

corresponding to the N combiners each made of an M-input logic-or in the bottom part of Fig. 3. Pipelining the proposed architecture improves the throughput, but leads to an area overhead. As an example, adding one pipeline register between each of the W stages in Fig. 2, (i.e. $W - 1$ pipeline registers), implies adding $W - 1 - q$ registers to each $y_q^{(M)}$, to increase the throughput by about W times.

III. Experimental results and comparisons

Experimental results obtained in the context of i) K-best MIMO detectors, ii) Non-binary LDPC decoder architectures and iii) Turbo product code architectures are shown and compared with solutions presented in the literature. Since several works do not give complete synthesis results, we re-implement the solutions presented in [18], [20], [22], [23] as stand-alone units. Moreover, we include the Partial Bitonic (PB) architecture proposed in [24]. Finally, the SN derived from [2] and proposed in [1] is summarized in the following paragraphs and included in the comparison as well.

A. SN review

The SN described in [1] is a special case of sorting network that moves the W largest values out of $M = 2W$ inputs to the first W output lines ($2W/W$ SN). It relies on two W-element sorters and a $2W$-element pruned-merger, depicted as two solid-line boxes and one dashed-line box respectively in Fig. 4 (a). In this work the sorters are implemented as even-odd Butcher sorting networks [1] and the pruned-merger is made of W comparators to select the W largest values. As argued by [1] this network can be extended to the case $M = n \cdot W$ by using $n - 1$ instances of the $2W/W$ SN. Unfortunately, the W maximum values obtained with this solution are not sorted. Thus, for a fair comparison with the proposed BWA architecture the SN is connected to a further W-element sorter. The general block scheme of the SN is shown in Fig. 4 (b).

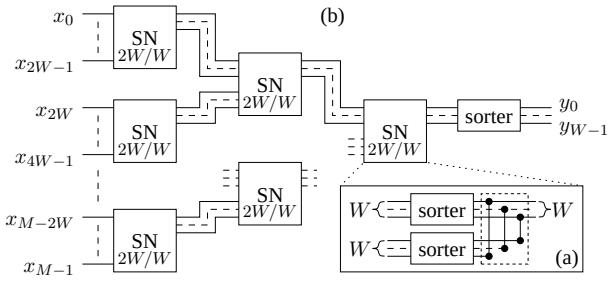

Figure 4: M/W SN general structure.

B. K-best MIMO detectors

In the K-best MIMO detectors detailed in [18], [23], [25]–[27], we observe that for 16-QAM and 64-QAM modulations ($Q = 16$ and $Q = 64$) at least 5-best and 10-best nodes ($W = 5$ and $W = 10$) are required respectively. Moreover, according to [18], [23], [26], [27] a typical value for the data width is sixteen bits, $N = 16$. So, for real-value detectors we have $M = W \cdot \sqrt{Q}$, namely $M = 20$ and $M = 80$ for 16-QAM and 64-QAM respectively. Both the architectures proposed in [18] and [23] deal with a 4×4 64-QAM K-best MIMO detector. In particular, [18] relies on the bubble-sort approach whereas in [23] tree-sort is applied.

C. Non-binary LDPC decoder architectures

The non-binary LDPC decoder architectures proposed in [20], [21] deal with codes in GF(32) and GF(64), that is $M = 32$ and $M = 64$ respectively. Moreover, they operate on a reduced number of messages, at least sixteen, that is $W = 16$ and we fix the data width to five bits ($N = 5$) as in [21].

The bubble check sorter proposed in [20] relies on a simplified extended min-sum (EMS) algorithm for check node processing that reduces the EMS original complexity from the order of W^2 to the order of $W\sqrt{W}$. In [28] it is implemented in several sequential rounds. On the other hand, in this current work, following the original description in [20] we implemented it as parallel architecture relying of a matrix structure. It is worth pointing out that, since the data in each row of the matrix described in [20] are supposed to be in order, our re-implementation of the bubble-check architecture has been equipped with a pre-sorting circuit. In [21], a reduced complexity sorter for the check node unit of a non-binary LDPC decoder is proposed. However, such an architecture relies on d_c rounds, where the M inputs are sliced and analyzed sequentially round by round. Since in this current work we deal with parallel sorting only, the architecture proposed in [21] is not considered in the comparison.

D. Turbo product code architectures

In the Chase-Pyndiah algorithm [10] a selection of the least reliable bits is necessary. As an example, in [22] a parallel implementation of turbo product codes that requires parallel partial sorting is addressed. Thus, in this section results for $M = 32, 64$ and $W = 3, 4$ are presented. The data width is five bits, $N = 5$, as in [22]. Since the sorter architecture in [22] is optimized for the case $M = 32, W = 3$ it can not be straightforwardly extended to other cases.

E. Comparisons

The BWA architecture, as well as the reference architectures in [1], [18], [20], [22]–[24] are all described in VHDL, simulated with ModelSim, synthesized using Synopsys Design Compiler, then placed and routed (P&R) using Cadence SoC Encounter on a 90 nm CMOS standard-cell technology, where a 2-input nand gate occupies 2.8 μm^2 [29]. Thanks to its scalability the BWA architecture can be easily adapted to the whole range of M, W and N values of the three considered applications. In Table I area (A) and critical path delay (C) for each architecture are compared. As it can be observed, the proposed BWA architecture features the lowest complexity among the solutions compared in Table I. The area of the BWA architecture is indeed less than half the area of the other solutions and about half the complexity of [22]. Besides, the critical path delay of the BWA architecture is almost comparable with that of other implementations. Finally, if we compute the area-delay-product (P=A·C), the proposed BWA architecture is comparable with [22] and is better than most of the other compared solutions. Further experiments adding pipelining have shown proportional throughput increase and area overhead for each pipeline stage of BWA architectures is always less than 35%.

IV. Conclusions

In this work a parallel radix-sort-based VLSI architecture for finding the first W ($W > 2$) maximum/minimum values

Table I: Post P&R results: area (A), critical path delay (C) and area-delay-product (P=A·C) .

	K-best MIMO detectors		Non-binary LDPC decoder architectures		Turbo product code architectures	
	M=20, W=5, N=16 A [μm^2] C [ns] P [mm^2 · ns]	M=80, W=10, N=16 A [μm^2] C [ns] P [mm^2 · ns]	M=32, W=16, N=5 A [μm^2] C [ns] P [mm^2 · ns]	M=64, W=16, N=5 A [μm^2] C [ns] P [mm^2 · ns]	M=32, W=3, N=5 A [μm^2] C [ns] P [mm^2 · ns]	M=64, W=4, N=5 A [μm^2] C [ns] P [mm^2 · ns]
SN [1]	34043 10.6 0.36	254977 42.78 10.91	28932 23.36 0.68	65988 39.27 2.59	14597 6.24 0.09	33213 10.61 0.35
PB [24]	43286 10.8 0.47	305753 38.06 11.64	22068 7.31 0.16	55267 10.54 0.58	19507 6.35 0.12	52778 10.20 0.54
[18]	33484[a] 19.68[a] 0.66[a]	93171 40.98 3.82	- - -	- - -	- - -	- - -
[23]	62221[a] 15.13[a] 0.94[a]	406108 45.35 18.42	- - -	- - -	- - -	- - -
[20]	- - -	- - -	29735[b] 37.96[b] 1.13[b]	47157[b] 40.86[b] 1.93[b]	- - -	- - -
[22]	- - -	- - -	- - -	- - -	8105 2.41 0.02	- - -
BWA	9345 16.92 0.16	29880 47.51 1.42	6153 36.29 0.22	9793 40.59 0.40	4562 6.72 0.03	8510 10.28 0.09

[a] Obtained by extending the architecture to the test case. [b] Obtained by adding a pre-sorting circuit.

is presented. The proposed solution, that relies on bit-wise analysis of the input data, is based on a W-stage architecture, where each stage is made of simple logic circuits referred to as modified $\mathcal{H}$ circuit and output generation circuit respectively. Obtained results show that BWA architectures feature lower area than other solutions proposed in the literature and competitive area-delay-product.

REFERENCES

[1] D. E. Knuth, *The Art of Computer Programming*. Addison-Wesley, 1998, vol. 3 - Sorting and Searching.

[2] V. E. Alekseyev, "Sorting algorithms with minimum memory," *Kibernetica, 5*, pp. 99–103, 1969.

[3] S. Papaharalabos, P. T. Mathiopoulos, G. Masera, and M. Martina, "Novel non-recursive max* operator with reduced implementation complexity for turbo decoding," *IET Communications*, vol. 6, no. 7, pp. 702–707, Jul 2012.

[4] M. Martina, S. Papaharalabos, P. T. Mathiopoulos, and G. Masera, "Simplified Log-MAP algorithm for very low-complexity turbo decoder hardware architectures," *IEEE Trans. on Instrumentation and Measurement*, vol. 63, no. 3, pp. 531–537, Mar 2014.

[5] F. Guilloud, E. Boutillon, and J. L. Danger, "λ-min decoding algorithm of regular and irregular LDPC codes," in *International Symposium on Turbo Codes and Related Topics*, 2003, pp. 451–454.

[6] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X. Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. on Communications*, vol. 53, no. 8, pp. 1288–1299, Aug 2005.

[7] S. Zezza, S. Nooshabadi, and G. Masera, "A 2.63 Mbit/s VLSI implementation of SISO arithmetic decoders for high performance joint source channel codes," *IEEE Trans. on Circuits and Systems I*, vol. 60, no. 4, pp. 951–964, Apr 2013.

[8] 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, Mar 2006.

[9] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. on Communications*, vol. 55, no. 4, pp. 633–643, Apr 2007.

[10] R. M. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Trans. on Communications*, vol. 46, no. 8, pp. 1003–1010, Aug 1998.

[11] K. Gunnam, G. Choi, and M. Yeary, "A parallel VLSI architecture for layered decoding for array LDPC codes," in *IEEE International Conference on VLSI Design*, 2007, pp. 738–743.

[12] D. Oh and K. K. Parhi, "Min-sum decoder architectures with reduced word length for LDPC codes," *IEEE Trans. on Circuits and Systems I*, vol. 57, no. 1, pp. 105–115, Jan 2010.

[13] C. Condo, M. Martina, and G. Masera, "VLSI implementation of a multi-mode turbo/LDPC decoder architecture," *IEEE Trans. on Circuits and Systems I*, vol. 60, no. 6, pp. 1441–1454, Jun 2013.

[14] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. on Communications*, vol. 47, no. 5, pp. 673–680, May 1999.

[15] E. Li, D. Declercq, and K. Gunnam, "Trellis-based extended min-sum algorithm for non-binary LDPC codes and its hardware structure," *IEEE Trans. on Communications*, vol. 61, no. 7, pp. 2600–2611, Jul 2013.

[16] C. L. Wey, M. D. Shieh, and S. Y. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE Trans. on Circuits and Systems I*, vol. 55, no. 11, pp. 3430–3437, Dec 2008.

[17] L. G. Amarù, M. Martina, and G. Masera, "High speed architectures for finding the first two maximum/minimum values," *IEEE Trans. on VLSI*, vol. 20, no. 12, pp. 2342–2346, Dec 2012.

[18] M. Shabany and P. G. Gulak, "A 675 mbps, 4x4 64-QAM K-Best MIMO detector in 0.13 um CMOS," *IEEE Trans. on VLSI systems*, vol. 20, no. 1, pp. 135–147, Jan 2012.

[19] B. Wu and G. Masera, "Efficient VLSI implementation of soft-input soft-output fixed-complexity sphere decoder," *IET Communications*, vol. 6, no. 9, pp. 1111–1118, Sep 2012.

[20] E. Boutillon and L. Conde-Canencia, "Bubble check: a simplified algorithm for elementary check node processing in extended min-sum non-binary LDPC decoders," *IET Electronics Letters*, vol. 46, no. 9, pp. 633–634, Apr 2010.

[21] X. Zhang and F. Cai, "Reduced-complexity decoder architecture for non-binary LDPC codes," *IEEE Trans. on VLSI systems*, vol. 19, no. 7, pp. 1229–1238, Jul 2011.

[22] C. Leroux, C. Jego, P. Adder, M. Jezequel, and D. Gupta, "A highly parallel turbo product code decoder without interleaving resource," in *IEEE Workshop on Signal Processing Systems*, 2008, pp. 1–6.

[23] P. Tsai, W. Chen, X. Lin, and M. Huang, "A 44 64-QAM reduced-complexity K-Best MIMO detector up to 1.5Gbps," in *IEEE International Symposium on Circuits and Systems*, 2010, pp. 3953–3956.

[24] K. Gunnam, G. Choi, W. Wang, and M. Yeary, "Parallel VLSI architecture for layered decoding," Texas A&M University, Tech. Rep., May 2007, available online at http://dropzone.tamu.edu/TechReports.

[25] Y. Sun and J. R. Cavallaro, "Low-complexity and high-performance soft MIMO detecion based on distributed M-algorithm through trellis-diagram," in *IEEE International Conference on Acoustics, Speach and Signal Processing*, 2010, pp. 3398–3401.

[26] D. Patel, V. Smolyakov, M. Shabany, and P. G. Gulak, "VLSI implementation of a WiMAX/LTE compliant low-complexity high-throughput soft-output K-Best MIMO detector," in *IEEE International Symposium on Circuits and Systems*, 2010, pp. 593–596.

[27] T. Kim and I. Park, "Small-area and low-energy K-Best MIMO detector using relaxed tree expansion and early forwarding," *IEEE Trans. on Circuits and Systems I*, vol. 57, no. 10, pp. 2753–2761, Oct 2010.

[28] E. Boutillon, L. Conde-Canencia, and A. Al-Ghouwayel, "Design of a GF(64)-LDPC decoder based on the EMS algorithm," *IEEE Trans. on Circuits and Systems I*, vol. 60, no. 10, pp. 2644–2656, Oct 2013.

[29] A. Pulimeno, M. Graziano, and G. Piccinini, "UDSM trends comparison: From technology roadmap to UltraSparc Niagara2," *IEEE Trans. on VLSI*, vol. 20, no. 7, pp. 1341–1346, Jul 10.1109/TVLSI.2011.2148183.