

Efficient Sorting Mechanism for Finding First W Maximum/Minimum Values By Using B.W.A Architecture

K.Raj Kumar¹, Dr.V.Thirimurthulu²

¹II M.Tech VLSI SD Student, CR Engineering College, Tirupati, Chittoor(Dist) A.P, India,

²Professor, Head of ECE Dept., CR Engineering College, Tirupati, Chittoor(Dist) A.P, India,

¹rk.kante@gmail.com, ²vtmurthy.v@gmail.com

Abstract

Applications like non-binary LDPC decoders k-best MIMO detectors & turbo product codes requires VLSI architectures for finding first $w(w>2)$ maxima/minima for implementation. A parallel radix-sort based VLSI architecture is proposed for finding the first w maxima/minima. A B.W.A architecture is explained, which uses a small logic circuits. This B.W.A architecture takes the input by analyzing from M.S.B to L.S.B. A high level of scalability is a key feature in the B.W.A architecture which enables by taking large range of both w & size of input data in a large spectrum of applications. When comparing to the other solutions in literature the practical results shows that the B.W.A architecture achieves less area *i.e.* less than 50%, while implementing on a high-speed 90nm CMOS standard cell technology. Among all cases which are considered shows that B.W.A gives lowest area-delay product.

Index terms: B.W.A architecture, LDPC decoder, selection networks, MIMO detectors

I. INTRODUCTION:

In computer science a crucial role in many applications and a well established defect is sorting. In previous works [3] it has been observed that sorting networks in hardware implementation are done very well. In communication field VLSI architectures and selection networks [5], are a part of various algorithms. For decoding of turbo and binary Low-density- Parity-check(LDPC) codes [1] and [7], for maximum-likelihood decoding of arithmetic codes in [6], for k-best MIMO detectors, non-binary LDPC decoders and turbo product codes in [11], [13] respectively, a partial sorting is observed. To implement min-sum approximations [6], [7] in binary LDPC decoder architectures [8], [10] uses circuits for finding the first two maximum values. Later they have also been used for non-binary LDPC decoders [2]. When coming to systematic approach Some works like [9], [12] gives a general problem of implementing parallel architecture for finding the first two maximum values. VLSI implementations of a) K-best MIMO decoders [14], [17], b) non-binary LDPC decoder [15] c) turbo product codes [16], are designed by architectures for finding the first $w > 2$ maximum/minimum values in a set of elements with $W \leq M=2$. No paper in the literature presents a general analysis for the case $W > 2$.

A comparator based SN is proposed by taking the concept form [5] for sorting networks, and also in [3]. As well as a radix sorting is used which is discussed in [3], and form other approaches. A bitwise analysis of data sorting is observed in radix

sorting and can prolongs to selection and partial sorting problems. An efficient sorting mechanism for finding first w maximum/minimum values from set of M values is proposed in this paper. B.W.A architecture is the proposed solution which works by taking input from M.S.B to L.S.B.

The rest of the paper is structured as follows. Section II describes the problem formulation and the proposed architecture. An initial version of B.W.A architecture is presented especially in section II and then a completed version is explained. Performance results and comparisons are discussed in section III. Section IV is ended with conclusion.

II. PROBLEM FORMULATION AND PROPOSED ARCHITECTURE:

As per [12], we can express the issue of finding the first W maximum/minimum values as takes after. Given a set $X^{(M)} = \{x_0, \dots, x_{M-1}\}$ made of M components, we need to discover the first W maximum/minimum values, specifically $y^{(M)} = \{y_0^{(M)}, y_1^{(M)}, \dots, y_q^{(M)}, \dots, y_{w-1}^{(M)}\}$ where $y_0^{(M)} = \max(X^{(M)})$, $y_1^{(M)} = \max(X^{(M)} \setminus \{y_0^{(M)}\})$, ..., $y_q^{(M)} = \max(X^{(M)} \setminus \bigcup_{k=0}^{q-1} \{y_k^{(M)}\})$, ..., $y_{w-1}^{(M)} = \max(X^{(M)} \setminus \bigcup_{k=0}^{w-2} \{y_k^{(M)}\})$ (likewise substituting max with min). For the purpose of straight forwardness in the accompanying we will examine just the max case.

A. Initial BWA architecture description

Radix sorting depends on the investigation of $X^{(M)}$ values a bit by bit at a time from the MSB to the LSB. In the accompanying, for the purpose of

effortlessness, we will expect that the qualities in $X^{(M)}$ are non-negative. It is significant, that 2's complement values can be clearly utilized also. Without a doubt, any set of N-bit 2's complement values can be changed over in non-negative values, protecting the request connection, by flipping the MSB.

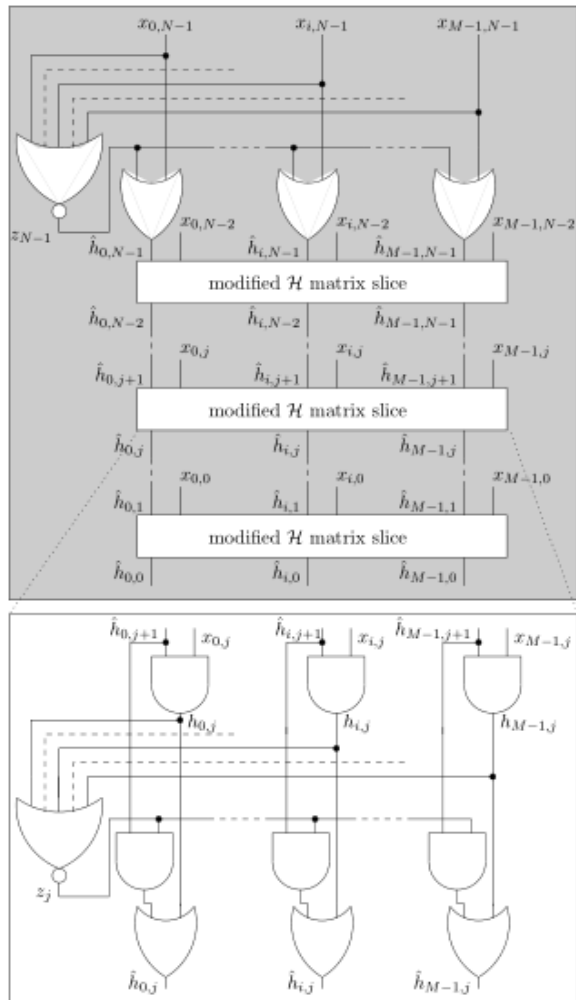


Figure 1: BWA architecture: modified H matrix.

Let us assume the two non-negative binary values are as follows

$$x_a = \{x_{a,N-1}, x_{a,N-2}, \dots, x_{a,1}, x_{a,0}\} \text{ and}$$

$$x_b = \{x_{b,N-1}, x_{b,N-2}, \dots, x_{b,1}, x_{b,0}\}$$

where $x_{a,j}$ and $x_{b,j}$ are the j -th bit of x_a and x_b respectively. By expecting the first $(N-j-1)$ -th MSBs of x_a and x_b have the same value, we can easily acquire the relationship between x_a and x_b by taking into the account bitwise analysis $x_a > x_b$ if $x_{a,j} = '1'$ and $x_{b,j} = '0'$ and viceversa.

The proposed BWA depends on operating recursively the logic-and operation between the near by bits of every x_i from MSB to LSB. Let $h_i = \{h_{i,N-2}, \dots, h_{i,0}\}$ be the group of bitwise logic and operation on x_i where $h_{i,N-2}$

$$z = x_{i,N-2} \wedge x_{i,N-2} \text{ and}$$

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

For $j = N-3, \dots, 0$ with \wedge showing the logic-and operation. A chance of MSB of all x_i values is '1' and all the x_i are monotonic successions of bits that just moves from '1' to '0' is permitted as in the four x_i values of example 1, then, estimating the substance of h_i for $i = 0, \dots, M-1$ from the LSB to MSB permits to discover the first W maximum values.

$$\begin{aligned} x_0 &= \{1 \ 1 \ 1 \ 1\} \\ x_1 &= \{1 \ 1 \ 0 \ 0\} \\ x_2 &= \{1 \ 0 \ 0 \ 0\} \\ x_3 &= \{1 \ 1 \ 1 \ 0\} \end{aligned} \quad \mathcal{H} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

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 H be the $(N-1) \times M$ matrix whose columns are h_i . Only in $x^{(M)}$ contains distinct elements: i) moving from MSB - row to the LSB-row of all \mathcal{H} matrix rows are different up to a certain j , then for $j' < j$ all the rows are zero ($j = 0$ in Example 1), ii) later the first non-zero row, one additional appears along a column while moving from MSB to LSB row. The columns of the first W non-zero rows are the positions of the first W maximum values while moving from SLB-row to MSB row as a consequence. For effective operation BWA requires for the modifications to be done due to the repeated elements may exist in $x^{(M)}$ and x_i is not a monotonic sequence.

B. Complete BWA architecture:

As explained in section II-A, the initial BWA principle can be operated on the data that are monotonic sequences of bits whose MSB is '1'. The architecture does not work properly if the data in $x^{(M)}$ do not meet the requirement. 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 $j' \leq j$. while taking this case into account the architecture can not compare between different x_i . If two or more x_i values are non-monotonic sequences then the same problem will arise. To overcome the problem we add some gates which is referred as a *zero row conditions*. To this one we modify as the following equation

$$\hat{h}_{i,j} = \begin{cases} h_{i,j+1} \wedge x_{i,j} & \text{if } j = N-1 \\ (z_j \wedge \hat{h}_{i,j+1}) \vee h_{i,j} & \text{if } 0 \leq j \leq N-1 \end{cases} \tag{3}$$

V is the logic-or operation and

$$z_{i,j} = \begin{cases} \text{not} (\bigvee_{i=0}^{M-1} x_{i,N-1}) & \text{if } j=N-1 \\ \text{not} (\bigvee_{i=0}^{M-1} h_{i,j}) & \text{if } 0 \leq j \leq N-1 \end{cases} \quad (4)$$

detects a zero-row condition. Follows an example.

$$\begin{matrix} x_0 = \{0 & 1 & 1 & 1\} & z_3 = 1 \\ x_1 = \{0 & 1 & 0 & 1\} & z_2 = 0 \\ x_2 = \{0 & 0 & 0 & 0\} & z_1 = 0 \\ x_3 = \{0 & 1 & 1 & 0\} & z_0 = 0 \end{matrix} \quad \hat{\mathcal{H}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

The above example shows the simple case where the modified \mathcal{H} matrix ($\hat{\mathcal{H}}$), where an $N \times M$ matrix is given. The maximum values are checked by checking $\hat{h}_{i,0}$ values which is explained in following paragraphs.

The figure-1 shows the slice architecture depicted as light gray which leads to handle zero-row, where each slice corresponds to one row of ($\hat{\mathcal{H}}$). To implement (3) and (4) the circuit are shown in the bottom part of the figure -1, where $\hat{h}_{i,j}$ acts as $h_{i,j}$, but when a zero-row condition occurs, then $\hat{h}_{i,j} = h_{i,j+1}$. As it can be seen in the modified \mathcal{H} matrix, the proposed structure ensures $\hat{h}_{i,0} = 1$ for at least one value of $i=0, \dots, M-1$. By this the selection of maximum values in the proposed architecture is performed by 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 $x^{(M)}$ contains more instances of the maximum value. If \mathcal{I} contains W elements, then the first W maximum values are the elements $\{x_i : i \in \mathcal{I}\} \subset \mathcal{X}^{(M)}$.

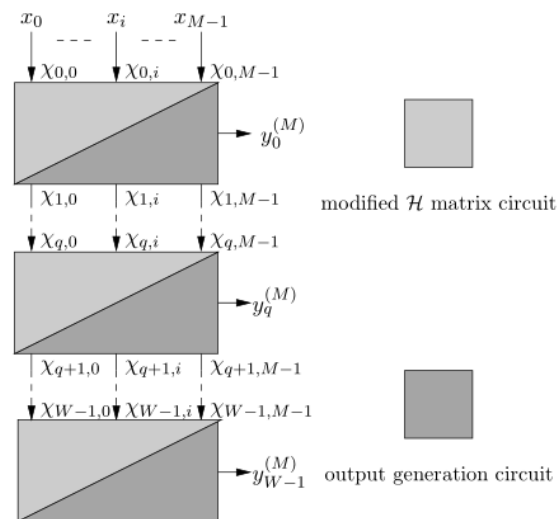


Figure 2: BWA architecture: cascade of W stages.

A new search is required if \mathcal{I} contains less than W elements. To produce a new set of elements $\mathcal{X}' = x'_0, \dots, x'_{M-1}$ where the maximum value is replaced by zero, by simplifying the selection we use a circuit discussed above as output generation circuit based on $\hat{h}_{i,0}$ values, is able to find maximum from M elements. In each stage shown in fig.2 contains the modified \mathcal{H} (light gray part) having one instance of the circuit and one instance of output generation circuit (dark gray part). The maximum value of q -th input set occurs only when the q -th stage finds $y_q^{(M)}$. For the case $q=0$ the operation is performed by means of output generation circuit which is shown as dark gray part in fig.3. A slice is an output generation circuit relies on $M-1$ blocks contains $M \times N$ selection blocks and N combiners where each made of an M -input logic or, and $g_i = \text{not}(\tau_{i-1}) \wedge \hat{h}_{i,0}$ for $i=1, \dots, M-1$ and $g_0 = \hat{h}_{0,0}$ are the M selection signals used in the selection blocks to replace t by zero or to forward x_i to next slice.

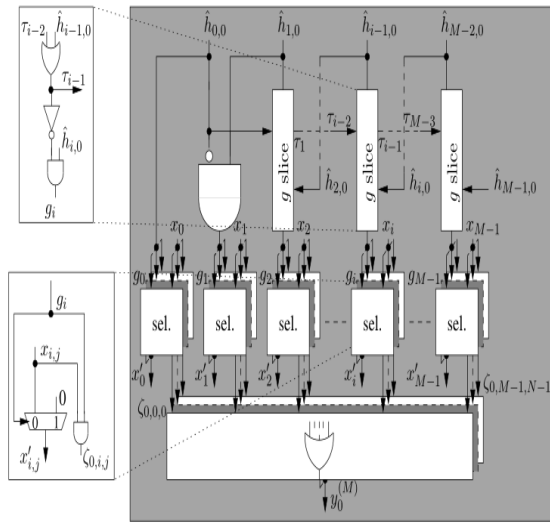


Figure 3: BWA architecture: output generation circuit in the case $q = 0$.

Every g_i signal is implemented as a slice (shown in the top left part of Fig. 3), in which the τ_{i-1} term is obtained as

$$\tau_{i-1} = \bigvee_{u=0}^{i-1} \hat{h}_{u,0} \quad (6)$$

i.e. : The remaining τ with $l = u + 1, \dots, M$ are '1' and in the present stage only x_u is selected only when $\hat{h}_{u,0} = '1'$. To calculate $\zeta_{q,i}$ more accurately, the g_i is exploited with reference to fig.3 by using 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

$$x_{q,i} = \begin{cases} x_i & \text{if } x_i \notin \bigcup_{k=0}^{q-1} \{y_k^M\} \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

and the terms $y_{q,j}^{(M)}$ and $X_{q,i,j}$ represent the j -th bit of $y_q^{(M)}$ and $X_{q,i}$ respectively (see the sel. block in the left part of Fig. 3). As an example, for $q = 0$ and $q = 1$ we have $X_{0,i} = x_i$ and $X_{1,i} = x'_i$ respectively. The q -th maximum is obtained finally by the N -combiners in which each are made of an M -input logic-or shown in the bottom of the fig-3.

$$y_q^{(M)} = \bigvee_{i=0}^{M-1} \zeta_{q,i}, \quad (8)$$

The proposed architecture pipelining enhances the throughput, but leads to an area overhead. As an example, between each of the W stages in Fig. 2, adding a pipeline register (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

In the context of i) K-best MIMO detectors, ii) Non-binary LDPC decoder architectures and iii) Turbo product code architectures with their experimental results are shown and compared with solutions presented in the literature. We re-implement the solutions obtained from [14], [15], [16], [19] as stand-alone units due to several works do not give the complete synthesis results. Moreover, the Partial Bitonic (PB) architecture proposed includes in 19[24]. Finally, the SN derived from [2] and proposed in [1] is discussed in the following paragraphs and included in the comparison as well.

A. SN review:

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

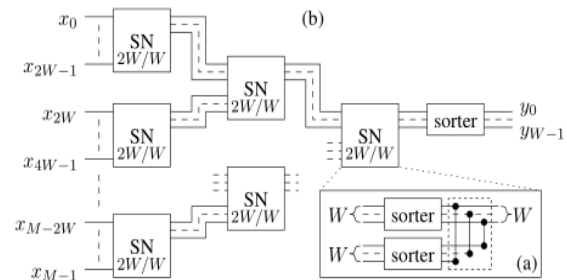


Figure 4: M/W SN general structure.

B. K-best MIMO detectors:

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 in the K-best MIMO detectors detailed in [14], [19]. Moreover, a typical value for the data width is sixteen bits, $N = 16$, according to [14], [19]. So, for real-value detectors we have $M = W \cdot \sqrt{Q}$ z, namely $M = 20$ and $M = 80$ for 16-QAM and 64-QAM respectively. The proposed architectures in [11] and [19], both deals with a 4×4 64-QAM K-

best MIMO detector. In particular, the architecture in [11] relies on the bubble-sort approach whereas in [19] tree-sort is applied.

C. Non-binary LDPC decoder architecture:

The proposed non-binary LDPC decoder architecture in [15] deals with codes in GF(32) and GF(64), i.e. $M = 32$ and $M = 64$ respectively. Moreover, these proposed architectures operates on a reduced number of messages, at least sixteen, i.e. $W = 16$. The bubble check sorter which is proposed in [15] 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 $W\sqrt{W}$.

On the other hand, a parallel architecture relying of a matrix structure is implemented in the present work followed by the original description in [15]. It is essential to point out that, the data in each row of the matrix are supposed to be in order which is discussed in [15], a pre-sorting circuit has been added in re-implementation of the bubble-check architecture. For the check node unit of a non-binary LDPC decoder a reduced complexity sorter is proposed in [20] 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.

D. Turbo product code architecture:

In the Chase-Pyndiah algorithm [13] a selection of the least reliable bits is necessary. As an example, in [16] 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 [16]. Since the sorter architecture in [16] is optimized for the case $M = 32, W = 3$ it cannot be straightforwardly extended to other cases.

E. Comparisons :

The BWA architecture, as well as the reference architectures in [3], [14], [15], [16], [18] are all described in verilog, simulated and synthesized using Xilinx ISE 14.1 development platform, then placed and routed (P&R) FPGA Genesys Virtex-5 development board. 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.. 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 ($P=A \cdot C$), the proposed BWA architecture is comparable with [16] and is better than most of the other compared solutions.

Table 1: Post P&R results: area (A), critical path delay (C) and area-delay-product ($P=A \cdot C$).

	K-best MIMO detectors			Non-binary LDPC decoder architectures			Turbo product code architectures		
	$M=20, W=5, N=10$ A [mm ²], C [ns], P [mm ² ·ns]	$M=80, W=10, N=16$ A [mm ²], C [ns], P [mm ² ·ns]	$M=16, W=16, N=5$ A [mm ²], C [ns], P [mm ² ·ns]	$M=64, W=16, N=5$ A [mm ²], C [ns], P [mm ² ·ns]	$M=32, W=3, N=5$ A [mm ²], C [ns], P [mm ² ·ns]	$M=64, W=4, N=5$ A [mm ²], C [ns], P [mm ² ·ns]			
SN [11]	34643 0.36 10.9	254077 10.91 42.78	28912 0.68 23.36	65968 2.59 39.27	14597 0.09 6.24	32123 0.35 10.61			
PB [24]	43266 0.47 18.68 ^(a)	305753 11.64 93171	38106 0.16 40.98	55267 0.58 10.34	19007 0.12 6.35	52178 0.54 10.20			
[18]	33484 ^(a) 0.66 ^(a)	93171 3.82	-	-	-	-			
[23]	62211 ^(a) 0.94 ^(a)	406108 15.13 ^(a) 3.82	18.42	-	-	-			
[20]	-	-	29733 ^(a) 1.13 ^(a)	31960 ^(a) 47157 ^(a) 40.60 ^(a) 1.93 ^(a)	-	-			
[22]	-	-	-	-	8105 0.02 2.41	-			
BWA	9185 ^(a) 0.16	1692 1.42	29880 47.51	6153 0.22	3629 0.40	9793 0.59			
					4562 0.03	6.72 8510 10.28			

^(a) Obtained by extending the architecture to the test case. ^(b) Obtained by adding a pre-sorting circuit.

Table 1

IV. Conclusion

In this work a parallel radix-sort-based VLSI architecture for finding the first W ($W > 2$) maximum/minimum values 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 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] 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.
- [2] 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.
- [3] D. E. Knuth, The Art of Computer Programming. Addison-Wesley, 1998, vol. 3 - Sorting and Searching.
- [4] 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
- [5] V. E. Alekseyev, "Sorting algorithms with minimum memory," Kiber-netica, 5, pp. 99–103, 1969.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] C. Leroux, C. Jégo, 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.
- [17] 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.
- [18] 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>.
- [19] 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.
- [20] 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.

AUTHORS:



K. Raj Kumar received his B.Tech degree in Electronics & Communication Engineering from Kandula Obula Reddy Memorial engineering college, Kadapa (A.P), India, in the year 2013. Currently pursuing his M.Tech degree in VLSI System Design at Chadalawada Ramanamma Engineering College, Tirupati(A.P), India.. His area of research includes low power VLSI design.



Dr.V.Thrimurthulu M.E., Ph.D., MIETE., MISTE Professor & Head of ECE Dept. He received his Graduation in Electronics & Communication Engineering AMIETE in 1994 from Institute of Electronics & Telecommunication Engineering, New Delhi, Post Graduation in Engineering M.E specialization in Microwaves and Radar Engineering in the year Feb, 2003, from University College of Engineering, Osmania University, Hyderabad., and his Doctorate in philosophy Ph.D from central University, in the year 2012. He has done his research work on Ad-Hoc Networks. He has published 20+ papers in national & International Journals.