

# Best-Choice Topology: An Optimized Array-based Maximum Finder

Marina Prvan<sup>1</sup>, Julije Ožegović<sup>2</sup>, Ivan Sočo<sup>3</sup>, Duje Čoko<sup>4</sup>

Department of Electronics and Computing, Faculty of Electrical Engineering  
Mechanical Engineering and Naval Architecture (FESB), Split, Croatia

**Abstract**—Extracting maximum from an unsorted set of binary elements is important in many signal processing applications. Since quite few maximum-finder implementations are found in the recent literature, in this paper we provide an update on the current topic. Generally, maximum-finders are considered array-based, with parallel bit-by-bit comparison of the elements, or more efficient tree-based structures, with the hierarchical maximum extraction. In this paper, we concentrate on array-based topologies only, since our goal is to propose a new maximum-finder design called Best-Choice Topology (BCT), which is an optimized version of the standard Array Topology (AT). The usual bit-by-bit parallel comparison is applied for extracting the maximum and its one-of-N address. Boolean expressions are derived for BCT logical design and the minimum-finder equivalent. Functionality of the proposed architecture and the reference designs is verified with Xilinx ISE Design Suite 14.5. Synthesis is done on Application Specific Integrated Circuit (ASIC) TSMC 65nm technology. The conclusion of the paper is two-fold. First, we confirm the timing efficiency of BCT compared to AT. Next, we show that BCT is more efficient than the recent maximum-finder design called Maximum Magnitude Generator (MaxMG) and it has a great potential to be used for real-time signal processing applications.

**Keywords**—Array topology; best-choice topology; maximum finder; maximum magnitude generator

## I. INTRODUCTION

The design of an efficient circuit for finding one or multiple maximum and/or minimum values in an unsorted set of binary numbers is very important. It is used by many applications [1-6] and depending on the purpose, circuits can be designed to produce value and/or the address of the winner element. For example, there are applications which require both maximum value and address [1, 7], while some others require only the fast computation of the maximum element [3, 8]. For example, finding minimum among distances to the cluster heads is crucial in clustering procedure or the neural network design [2, 5]. Extracting minimal distance to the stored database images or finding maximum among different image vectors is important in object recognition and classification [9]. The commonly used rank order filter requires fast computation of the central maximum, median or minimum [10].

Extracting one or multiple winner elements in a set can be solved by sorting the whole set and selecting the first or last few values [11, 12]. This approach is common in rank order filter algorithm, with complete or partial sorting used to extract the median. However, even though many efficient

sorting algorithms are designed in hardware with optimized implementations [9, 13], sorting the whole set for extracting a single maximum/minimum value is not feasible when efficient throughput and latency are required. This raises the demand for the design of the dedicated hardware solutions.

Very few papers on the maximum-finder topologies have been published in recent years. This paper gives an update on the current topic and a solution of the maximum finding problem is proposed by a new logical design named Best-Choice Topology (BCT). It is an optimization of the standard Array Topology (AT), designed as a parallel combinational circuit that generates the value and one-of-N binary address of the maximum among N binary numbers. Minimum finder BCT can be easily derived with the minor change of the Boolean operations. The functionality of BCT and its reference designs is verified and simulated by using Verilog Hardware Definition Language (HDL). Synthesis is done on Application Specific Integrated Circuit (ASIC) TSMC 65 nm technology, by using Cadence Genus Synthesis Solution tool [14]. We show that BCT is faster than AT, especially for lower size of the binary input codewords. Also, it offers greater efficiency in timing, area and power than the recent Maximum Magnitude Generator (MaxMG) design.

The paper is organized as follows. Section II gives a summary of the existing topologies, with AT functionality explained in Section III. Next, in Section IV, a new maximum finder BCT design is proposed with optimization compared to AT. The general equivalent maximum and minimum finder BCT models are given. Section V and Section VI provide simulation and synthesis results. The BCT performance efficiency is evaluated and compared to referent designs. Limitations of the study are discussed in Section VII. Section VIII concludes the paper, followed by the references used.

## II. RELATED WORK

A literature survey is provided in [15], with circuits characterized as array-based or tree-based, depending on their logical design. The former ones are simpler, with bitwise comparisons of all candidates done in parallel, while the latter are more efficient in terms of latency. Authors propose the Array Based Topology (ABT) and some other designs using ABT as a basic building block. ABT is a compromise between performing a parallel comparison of all pairs of k-bit input data elements, and a tree-based hierarchical approach for the maximum extraction. They compare the proposed architecture with the simple Traditional Binary Tree (TBT) solution with

comparator and multiplexer blocks, as well as its more efficient Parallel Binary Tree (PBT) variants like Ripple Carry (RCT) and Carry Select Topology (CST).

Recently, authors in [16] proposed an ABT-based architecture that is more efficient in timing but detrimental to area and power. On the other hand, an efficient maximum-finder topology called Maximum Magnitude Generator (MaxMG) is implemented in [17]. The design is combinational and array-based, so it works as a filter comparing all input data bits in parallel. It consists of three main levels of logic gates, as shown in Fig. 1. The first level extracts the MSB of the maximum element by using OR gates. The second level uses XOR gates for comparing MSB of all inputs with the previous level result MSB bit. Input data elements having the same MSB as the result could be potential maximums.

High signal generated from the second level is sent to the third level where XOR output is multiplied with each of the input MSBs. This level is a filter, providing a signal if the current element is still a potential maximum or it is excluded from the competition. Namely, AND output is transferred to the next level's AND for the next maximum element's bit calculation. If the output of the AND gate is low, the number is excluded from the maximum element competition. In case of the all input data elements having the MBS low, next XOR gate will have a high output, so that AND gate decide which number is chosen.

Authors in [18] propose an optimized maximum-finder circuit that is based on the Leading Zero Count Topology (LCT). The idea is to encode each k-bit binary input element so that a bitwise OR operation can be performed between the coded values. The count of the leading zeros in the result gives the position or index of the winner element. There are some other tree-like topologies in the literature, designed for the first two maximum or minimum values extraction, or calculating the kth best value in an unsorted list of elements, where the position k can be arbitrary [19-21].

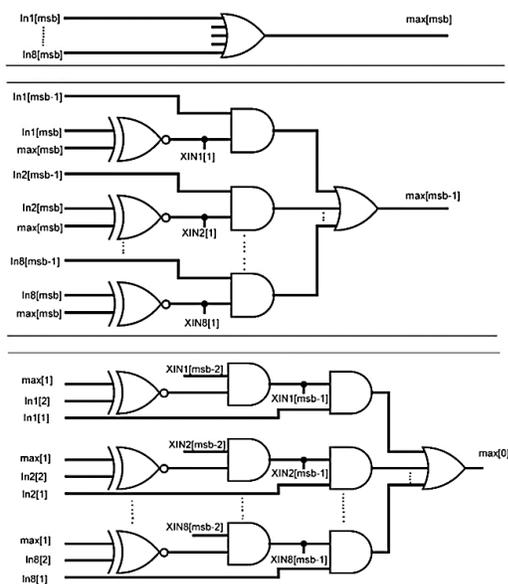


Fig. 1. Generalized MaxMG Topology [17].

### III. ARRAY TOPOLOGY (AT)

BCT proposed in this paper is based on the optimized version of the Array Topology (AT) [15, 22], namely, AT is based on a filtering approach where all candidates are processed in parallel from the most to the least significant bit (MSB to LSB). Progressively, the number of candidates in the calculation of the maximum element is reduced by using the enable signal. This one disables the candidate that loses the chance to become a maximum when it has lost on a specific bit. As shown in Fig. 2, the basic building block of the topology is one AT-cell for each of the k candidate bits.

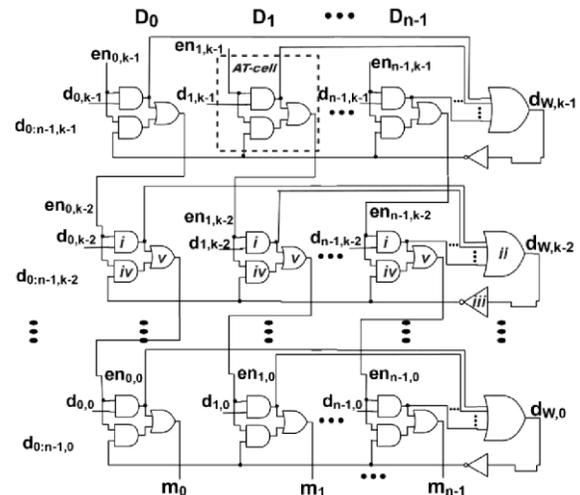


Fig. 2. Array Topology [15].

AT uses the 2-input AND gate (i) to select input data bits from n numbers that are enabled, and the signals are sent to the n-input OR gate (ii) providing the corresponding winner bit. Inverted winner data bit (iii) and 2-input AND gate (iv) ensure that even if all bits were zero, the result is still transferred to the next AT row, or the enable signals would all be low. Next, 2-input OR gate (v) generate the enable signal. Low/high output is generated depending on input signals, enabling/disabling the current input bit. The advantage is that we can get the winner element just after the first level if only one candidate has a high MSB bit, as AT will pass only the bits of that candidate. Besides the winner element, AT gives its corresponding address. It is generated by the enable signal propagated through the bits of each candidate. Value of the address bit is high if the candidate was enabled for all bits, and it is low otherwise.

### IV. PROPOSED BEST-CHOICE TOPOLOGY (BCT)

#### A. Main BCT Concept

The solution for solving maximum-finder problem in the proposed BCT is defined again as a filtering approach. A bit-by-bit comparison of all candidates in parallel from MSB to LSB is applied, progressively reducing the number of candidates for the winner element. BCT functionality can be approximated with the self-organizing binary network that stabilizes when the maximum is found and settled on the result bus. This self-stabilizing concept is based on the combinational type of digital circuit, with binary signal

comparison results obtained. The solution is accomplished by using only feedback coming from the lowest bit-level comparisons.

For example, if some of the N input elements receive negative feedback on a bit-level comparison, they exclude themselves from the winner competition, which is basically the calculation of the maximum element. On the other hand, if some element is high enough to become the winner (maximum), a positive feedback is generated, and system will advance in the maximum finding goal. Therefore, BCT logic can be summarized in two important aspects:

- The comparison is done bit-by-bit in parallel from MSB to LSB and extracting the winner element. This concept is like AT, but BCT uses an improved self-exclusion technique for the bit of the input element as well as fast propagation of the winner decision towards LSB.
- Feedback is implemented by a result (OR) bus where the winner bits are generated. It is important that this bus is unique, independent of the size of the input dataset. This means that there is no additional decision logic (such as in binary trees) but the OR bus realization. In ASIC technology this is the real OR-gates or wired-OR bus implemented with open-collector logic. Thus, additional delay remains only in the tree of OR gates of BCT structure.

Since the main winner competition is done on the lowest bit level, each of the input elements gives its high bit on the bus as if it was a winner and withdraws its high bit together with all the lower significant bits when it has lost the battle for the winner element. Thus, overall behavior of the BCT system is caused by the lowest bit-level components; every input element is entering the winner competition with its bits and it concludes if it is strong enough to be the winner. If it loses on the specific bit, it excludes itself from the further competition.

**B. AT Optimization in BCT**

BCT comparison starts from the basic Single Bit (SB) blocks working in parallel. SB comparator module consists of two AND gates, one OR gate and one inverter, as shown in Fig. 3. Inputs to the SB block are denoted as signals d (data bit) and r (result bit), because r is a bit on the result bus (the bit of the current maximum) and d is a current input data bit (the bit of the current candidate).

Comparison SB block output is marked wi and it becomes the enable signal t for the next bit. Signal t will be high and current input element is enabled while  $d \geq r$ . When  $d < r$ , it means that candidate has lost on a specific bit and t becomes low (Fig. 4). This disables the candidate from the further competition since output signal b will be zero for the current data bit and all further bits up to LSB.

BCT filtering concept is similar but more efficient than AT implementation, namely, in AT SB block, the next bit enable signal cannot be calculated before calculating previously the bit of the current maximum. This will cause one additional gate delay before calculating the winner bit, and shorter critical path delay as presented in Fig. 3. However, in BCT,

enable signal for the next bit can be calculated immediately from the data bit, which means that we can have the maximum calculation started as soon as input data becomes available. This gives an enhancement of faster calculating the result element bits in BCT with respect to the traditional AT. BCT circuit area is foreseen to be the same as AT, since there is the same number of logic gates at the lowest level of SB comparators.

**C. BCT Maximum Finder Example**

BCT network stabilizes when winner is extracted on result bus. This means that there will be several result bus configurations that will change during the winner competition.

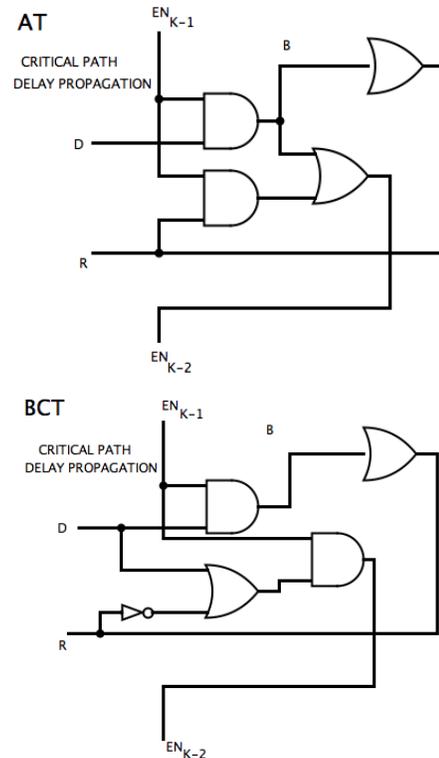


Fig. 3. The AT and BCT Single Bit (SB) Block.

| t | d | r | b | wi |
|---|---|---|---|----|
| 0 | 0 | 0 | 0 | 0  |
| 0 | 0 | 1 | 0 | 0  |
| 0 | 1 | 0 | 0 | 0  |
| 0 | 1 | 1 | 0 | 0  |
| 1 | 0 | 0 | 0 | 1  |
| 1 | 0 | 1 | 0 | 0  |
| 1 | 1 | 0 | 1 | 1  |
| 1 | 1 | 1 | 1 | 1  |

| dr | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| t  | 00 | 01 | 11 | 10 |
| 0  |    |    | 1  | 1  |
| 1  |    |    |    |    |

$b = t \&\& d$

| r | d | $\neg r \vee d$ |
|---|---|-----------------|
| 0 | 0 | 1               |
| 0 | 1 | 1               |
| 1 | 0 | 0               |
| 1 | 1 | 1               |

| dr | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| t  | 00 | 01 | 11 | 10 |
| 0  |    |    | 1  | 1  |
| 1  |    | 1  |    |    |

$w = (t \&\& \neg r) \vee (t \&\& d)$   
 $w = t \&\& (\neg r \vee d)$

Fig. 4. Boolean Expressions for the BCT SB Block.

|                                              |
|----------------------------------------------|
| result bus = {0000000}                       |
| $x_0 = \{00010101\}_{(2)} = 21_{(16)}$       |
| $x_1 = \{00101010\}_{(2)} = 42_{(16)}$       |
| $x_2 = \{00110101\}_{(2)} = 53_{(16)}$       |
| $x_3 = \{01000100\}_{(2)} = 44_{(16)}$       |
| $x_4 = \{10001000\}_{(2)} = 88_{(16)}$       |
| $x_5 = \{10000111\}_{(2)} = 87_{(16)}$       |
| $x_6 = \{11010110\}_{(2)} = D6_{(16)}$       |
| $x_7 = \{01100110\}_{(2)} = 66_{(16)}$       |
| result bus = {0000000}                       |
| bits <sub>x0</sub> = {00010101}              |
| bits <sub>x1</sub> = {00101010}              |
| bits <sub>x2</sub> = {00110101}              |
| bits <sub>x3</sub> = {01000100}              |
| bits <sub>x4</sub> = {10001000}              |
| bits <sub>x5</sub> = {10000111}              |
| bits <sub>x6</sub> = {11010110}              |
| bits <sub>x7</sub> = {01100110}              |
| result bus = {11111111} = FF <sub>(16)</sub> |
| bits <sub>x0</sub> = {00000000}              |
| bits <sub>x1</sub> = {00000000}              |
| bits <sub>x2</sub> = {00000000}              |
| bits <sub>x3</sub> = {00000000}              |
| bits <sub>x4</sub> = {10000000}              |
| bits <sub>x5</sub> = {10000000}              |
| bits <sub>x6</sub> = {11000000}              |
| bits <sub>x7</sub> = {00000000}              |
| result bus = {11000000} = C0 <sub>(16)</sub> |
| bits <sub>x4</sub> = {10000000}              |
| bits <sub>x5</sub> = {10000000}              |
| bits <sub>x6</sub> = {11000110}              |
| result bus = {11010110} = D6 <sub>(16)</sub> |
| address indicator = {01000000}               |

Fig. 5. BCT Maximum Extraction Example.

Example is given in Fig. 5. Initially, result bus is set to zero, so all input elements are winner candidates. This initial condition will reset the result bus and trigger the winner competition. It will put low bit of the result to high, enabling the enable signal generation. When the k-bit results of the SB comparisons are calculated for each of the input elements, array of k n-input OR gates is used for bitwise extraction of the temporary maximum bits on the result bus. This temporary result bus configuration forces each of the elements to compete by giving its high bits to the bus. Result is calculated when all candidates have lost the competition but the winner element that is larger or equal to the temporary result bus configuration. This configuration is generated by doing OR operation between each element's k-bit result bits. These are higher or equal to the winner, so that all maximum bits are extracted correctly after the OR operation. One-of-n binary address of the maximum is generated from the enable bits, where address bits are set low for all candidates but the winner position.

#### D. Generalized Maximum-Finder BCT Logic

Maximum or winner element is denoted as:

$$y_{max} = y_{k-1} y_{k-2} y_1 y_0 \quad (1)$$

It is computed in the following manner. As a first step, a parallel execution of n circuits with k slices produce n codewords which represent bitwise comparisons. Each

codeword corresponds to the data k-bit number that is taken as input, as well as the enable signal  $w_i$  which will enable or disable the current number from the further competition. We refer to these code words as  $bdata = b_{k-1} b_{k-2} \dots b_1 b_0$  where  $b_i$  in each slice is calculated as:

$$b_i = d_i \wedge t_i, i=0, 1, 2, \dots, k \quad (2)$$

While data element is greater or equal to the winner element, its resulting bit  $b_i$  will be high, and it will be put to low as soon as data element losses on a specific bit in the winner competition. Data bit is presented as  $d_i$  and the  $i$ th bit comparison element enable signal is marked as  $t_i$ . Based on the annulment law in the Boolean algebra, as soon as enable signal becomes low, the result bit  $b_i$  will be put to low as well, causing the data element to be disabled from the further winner competition. Enable or disable signal  $w_i$  from the previous bit is transmitted to the next one in the ripple way:

$$t_{k-1} = t_{k-1} \wedge 1 = t_{k-1}$$

$$t_{k-2} = w_{k-1} \wedge t_{k-1} = w_{k-1} \wedge t_{k-1}$$

$$t_{k-3} = w_{k-2} \wedge w_{k-1} \wedge t_{k-1} = w_{k-2} \wedge t_{k-2}$$

$$t_0 = w_1 \wedge t_1 \quad (3)$$

Signal is propagating from MSB to LSB in the data element, enabling the calculation of the  $bdata$  codeword for each number competing for the winner position. One can see in the equations above that lower significance bit enable signals  $t_{k-2}, t_{k-3}, \dots, t_0$  depend on the win bits  $w_{k-1}, w_{k-2}, \dots, w_1$  where each win bit is enabling or disabling the next bit in the data element winner competition. This win signal tells if the current element is still in the competition or it has lost on the specific bit. Therefore, signal  $t_i$  depends on  $w_{i+1}$  and once the data element loses the battle in the winning network, enable signals  $t_i, t_{i-1}, \dots, t_1$  will become zero, automatically annulling the corresponding  $bdata$  bits  $b_i, b_{i-1}, \dots, b_1$ . Win bits  $w_i$  are:

$$w_i = t_i \wedge (\neg r_i \vee d_i) \quad (4)$$

In the equation (4), data bit  $d_i$  of the current element is competing with the result bit  $r_i$  of the winner element that is currently on the result bus. Result bus will obtain several configuration changes until the winner element is settled on the result bus and the winning network stabilizes in the search for the maximum element. For n elements competing in the network, result is put to zero at the very beginning. This reset of the result bus will start the competition and destabilize the network. Due to this initial zero configuration, each of the input data elements will be potential winner as they are all greater than the current result.

After the bitwise competition for n k-bit elements in parallel, winner is extracted on the result bus. It is done based on the  $bdata$  codewords of each data element as:

$$rk-1 = bdata_0(k-1) \vee \dots \vee bdata_{(n-1)}(k-1)$$

$$rk-2 = bdata_0(k-2) \vee \dots \vee bdata_{(n-1)}(k-2)$$

$$r1 = bdata_01 \vee bdata_11 \vee \dots \vee bdata_{(n-1)}1$$

$$r0 = bdata_00 \vee bdata_10 \vee \dots \vee bdata_{(n-1)}0 \quad (5)$$

E. Minimum-Finder BCT

Minimum finder BCT (Fig. 6) is derived with a minor change of Boolean expressions in Fig. 4. Unlike previously calculating enable signal  $w_i$ , we invert the data bit to obtain enable signal for minimum instead of the maximum:

$$w_i = t_i \wedge (r_i \vee \neg d_i) \tag{6}$$

The minimum-finder BCT is based on the same concept as maximum-finder circuit. The exact difference is in the winner element extraction. Naturally, since the same logic gates are used, the bit  $d$  of the current element must be inverted, unlike the bit of the result  $r$  in the maximum-finder BCT variant.

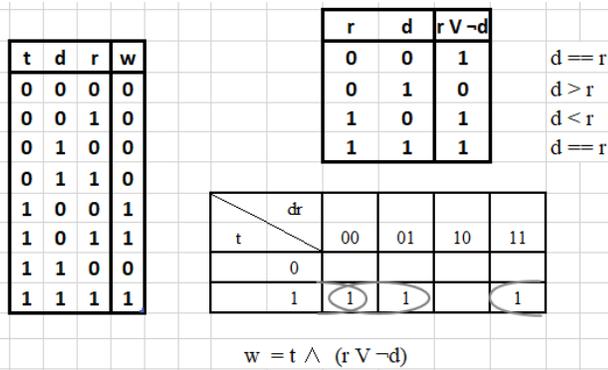


Fig. 6. Minimum Finder BCT Boolean Expressions.

V. VERILOG SIMULATION RESULTS

A. Experimental Setup

The referent designs used as competitors to BCT are array-topologies only, as our goal in the paper is not to compare array-based with tree-based topologies. Goal is to demonstrate the improvement over the state of the art with the proposed design, but only among array-based topologies. Therefore, to obtain a fair comparison, the competitors which are selected are AT and the recently introduced MaxMG. The topologies are implemented in Verilog HDL and simulated within the Xilinx ISE Design Suite 14.5 development environment. Timing and area results are given for the comparison of  $n$   $k$ -bit input values in an unsorted set of elements where  $n=\{8,16,32,64\}$  and  $k=\{8,16,32,64\}$ . Synthesis is done with Cadence Genus synthesis tool on ASIC TSMC 65 nm technology, with constraints adjusted to virtual clock pulse of

20 ns. This working frequency represents the clock period of the integrated circuit used to implement the design and to constrain critical input/output path delay.

B. Xilinx ISE Simulation

The functionality of the topologies is verified by the simulation presented in Fig. 7. These are the results for the winner competition between  $n=8$  input elements, each of them  $k=8$  bits wide (example in Fig. 5). AT has the largest delay in the simulation, since more than 1 ns is needed for the winner extraction. On the other hand, MaxMG and BCT are faster, with 900ps delay. BCT configuration changes are presented on the result bus, where a SB-based competition between elements is done. It is shown in the simulation how BCT asynchronous network is destabilized with the changed input data and stabilized again when the winner is found on the result bus.

VI. ASIC SYNTHESIS RESULTS AND DISCUSSION

A. Area

The concise way of presenting the results is inspired by [15]. The area comparison of BCT with the reference circuits is given in Tables I and II. We present evaluation of the number of logic gates in ASIC, with area estimated in  $\mu m^2$ . In the theoretical calculation based on BCT and AT logical designs, BCT consists of the same number of logic gates as AT in the lowest level of SB comparators. Namely, when calculating number of gates from the digital logic scheme, an 8-bit maximum selector in a set of 8 input elements based on AT consists of:

- 128 2-input AND gates
- 64 2-input OR gates
- 8 8-input OR gates

That is in total 192 2-input gates and 8 8-input OR gates [23], where  $m$ -input logic gates can be implemented as a binary tree of  $m-1$ -unit gates [15]. This is in total  $8*(8-1) = 56$  gates for AT. BCT uses the same number of gates in SB blocks, i.e. 2 2-input AND gates and 1 2-input OR gate for each of the 64 SB blocks. Hence, BCT uses 192 gates for SB parallel comparisons, and with an array of  $n$   $k$ -input OR gates for the result bus implementation, that is again  $8*(8-1) = 56$  gates.

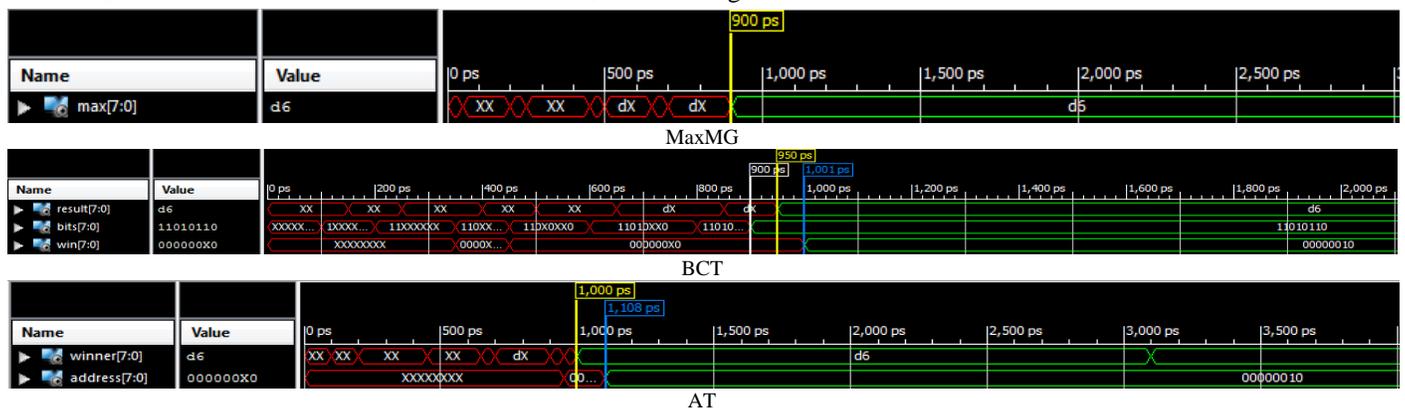


Fig. 7. Simulation Results.

TABLE. I. AREA (# LOGIC GATES)

| k,n   | MaxMG | AT   | BC   |
|-------|-------|------|------|
| 8,8   | 143   | 147  | 110  |
| 8,16  | 292   | 251  | 206  |
| 8,32  | 601   | 524  | 417  |
| 8,64  | 1191  | 1078 | 818  |
| 16,8  | 315   | 281  | 222  |
| 16,16 | 655   | 465  | 414  |
| 16,32 | 1336  | 948  | 841  |
| 16,64 | 2670  | 1868 | 1652 |
| 32,8  | 646   | 429  | 446  |
| 32,16 | 1359  | 834  | 830  |
| 32,32 | 2758  | 1658 | 1689 |
| 32,64 | 5476  | 3254 | 3314 |
| 64,8  | 1315  | 862  | 894  |
| 64,16 | 2771  | 1667 | 1662 |
| 64,32 | 5598  | 3322 | 3385 |
| 64,64 | 11596 | 8394 | 8394 |

TABLE. II. AREA COMPARISON [UM<sup>2</sup>]

| k,n   | MaxMG    | AT       | BC       |
|-------|----------|----------|----------|
| 8,8   | 293,40   | 335,88   | 266,40   |
| 8,16  | 637,56   | 592,92   | 533,88   |
| 8,32  | 1257,12  | 1223,28  | 1058,40  |
| 8,64  | 2913,48  | 2539,80  | 2114,28  |
| 16,8  | 623,52   | 630,00   | 537,12   |
| 16,16 | 1077,84  | 1129,32  | 1077,84  |
| 16,32 | 2138,40  | 2289,60  | 2138,40  |
| 16,64 | 4274,28  | 4572,00  | 4274,28  |
| 32,8  | 1281,24  | 1090,44  | 1078,56  |
| 32,16 | 2785,32  | 2139,84  | 2160,72  |
| 32,32 | 4296,60  | 4289,76  | 4296,60  |
| 32,64 | 8585,64  | 8545,68  | 8585,64  |
| 64,8  | 2599,20  | 2184,84  | 2161,44  |
| 64,16 | 5636,88  | 4284,36  | 4328,28  |
| 64,32 | 8609,76  | 8598,24  | 8609,76  |
| 64,64 | 19436,40 | 19459,80 | 19436,40 |

Similarly, the total number of logic gates for MaxMG implementation is dependent on the number of stages and input data elements, where the number of stages is equal to the number of input data bits. For example, in theoretical calculation from the MaxMG logic scheme, 8 bytes input maximum selector based on MaxMG consists of:

- 4 2-input OR gates in the first stage;
- 4 2-input OR gates, 8 XNOR gates and 8 AND gates in the second stage;
- 4 OR, 8 XNOR and 16 AND gates for each of the remaining 6 stages.

That is in total 4 gates in the first stage, 36 gates in second stage and  $44 \cdot 6 = 264$  gates in total for the remaining MaxMG stages. This gives in total 304 gates for MaxMG, which is 22% larger than BCT in theoretical calculation. We consider in our calculation that XNOR is accomplished as XOR with an inverter, i.e. a single XOR gate area approximation. In the area comparison summarized in Fig. 8, we can see that the above theoretical calculations are verified. The synthesis result for BCT is almost the same as AT with small improvement factor. Also, BCT is more efficient than MaxMG, with the average improvement around 10%.

B. Propagation Delay

Timing efficiency is measured from the moment all input data is obtained, which is 3000 ps after the first data read-out. When the last signal is transferred to the final block in the design, the total propagation delay of the synthesized circuit is obtained. The results are presented in Table III.

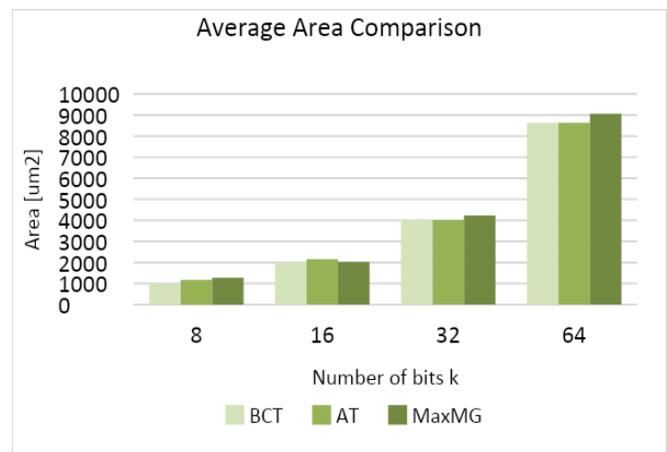


Fig. 8. Area Comparison.

TABLE. III. CRITICAL DATA PATH DELAY [NS]

| k,n   | MaxMG  | AT     | BC     |
|-------|--------|--------|--------|
| 8,8   | 9,926  | 5,687  | 4,442  |
| 8,16  | 10,317 | 6,026  | 5,566  |
| 8,32  | 11,592 | 6,975  | 5,423  |
| 8,64  | 11,672 | 7,381  | 5,508  |
| 16,8  | 18,331 | 7,536  | 5,670  |
| 16,16 | 18,695 | 7,033  | 6,492  |
| 16,32 | 19,504 | 7,697  | 6,349  |
| 16,64 | 19,734 | 7,542  | 6,439  |
| 32,8  | 34,903 | 8,336  | 8,126  |
| 32,16 | 35,368 | 8,758  | 8,587  |
| 32,32 | 35,951 | 8,886  | 8,507  |
| 32,64 | 36,019 | 8,914  | 8,668  |
| 64,8  | 68,107 | 13,144 | 13,038 |
| 64,16 | 68,314 | 13,610 | 13,389 |
| 64,32 | 69,546 | 13,769 | 13,355 |
| 64,64 | 71,521 | 15,576 | 15,548 |

It is shown that BCT outperforms referent designs in timing. Since BCT is the optimized version of AT, it provides more efficient calculation of the enable signal, and faster extraction of the winner on the result bus. There is no ripple effect in BCT which is present in AT, causing a poor timing result, namely, bits having the same significant values must wait on the output signal from other bits.

Even though each AT-cell in AT row can be executed in parallel, every AT-cell in the column is dependent of the previous one at the higher level. This means that lower-level AT-cell in the same column cannot begin its work before the enable signal for that AT-cell is calculated in former AT-cell. Also, enable cannot be calculated before the winner data bit is generated, and all the winner bits cannot be calculated until all enable bits settle down [15]. Since in AT circuit next bit enable signal cannot be calculated before calculating the bit of the current maximum, this will cause one additional gate delay before calculating the winner bit. However, in BCT, enable signal for the next bit can be calculated immediately from the data bit, which means that we can have the maximum element calculation started as soon as the input data becomes available. The comparison between BCT and AT is visualized in Fig. 9.

Fig. 10 shows the average propagation delay trendlines when BCT is compared with AT and MaxMG. We can see that BCT optimization is more present for smaller number of bits in the input data. In that case, BCT is faster in extracting the maximum from the dataset for around 20 % decrease in propagation delay. For larger number of bits,  $k > 32$ , the average difference between BCT and AT is negligible (around 2%). This is caused by the enable signal propagation when larger number of input bits is compared in the input.

BCT is more efficient than MaxMG (Fig. 10(b)), since it is a parallel structure comparing all input bits at once in the winner competition. Maximum element can be settled on the result bus by using smaller number of logic gates causing on smaller delay on logic since BCT is a smaller circuit. BCT does not require  $k$  calculation stages processing in parallel for  $k$ -bit maximum value generation with a single bit carry propagation to the next stage. Therefore, BCT will calculate the maximum with at most  $(n/8k) + 1$  result bus configuration changes, where  $n$  is number of elements and  $k$  is number of data bits.

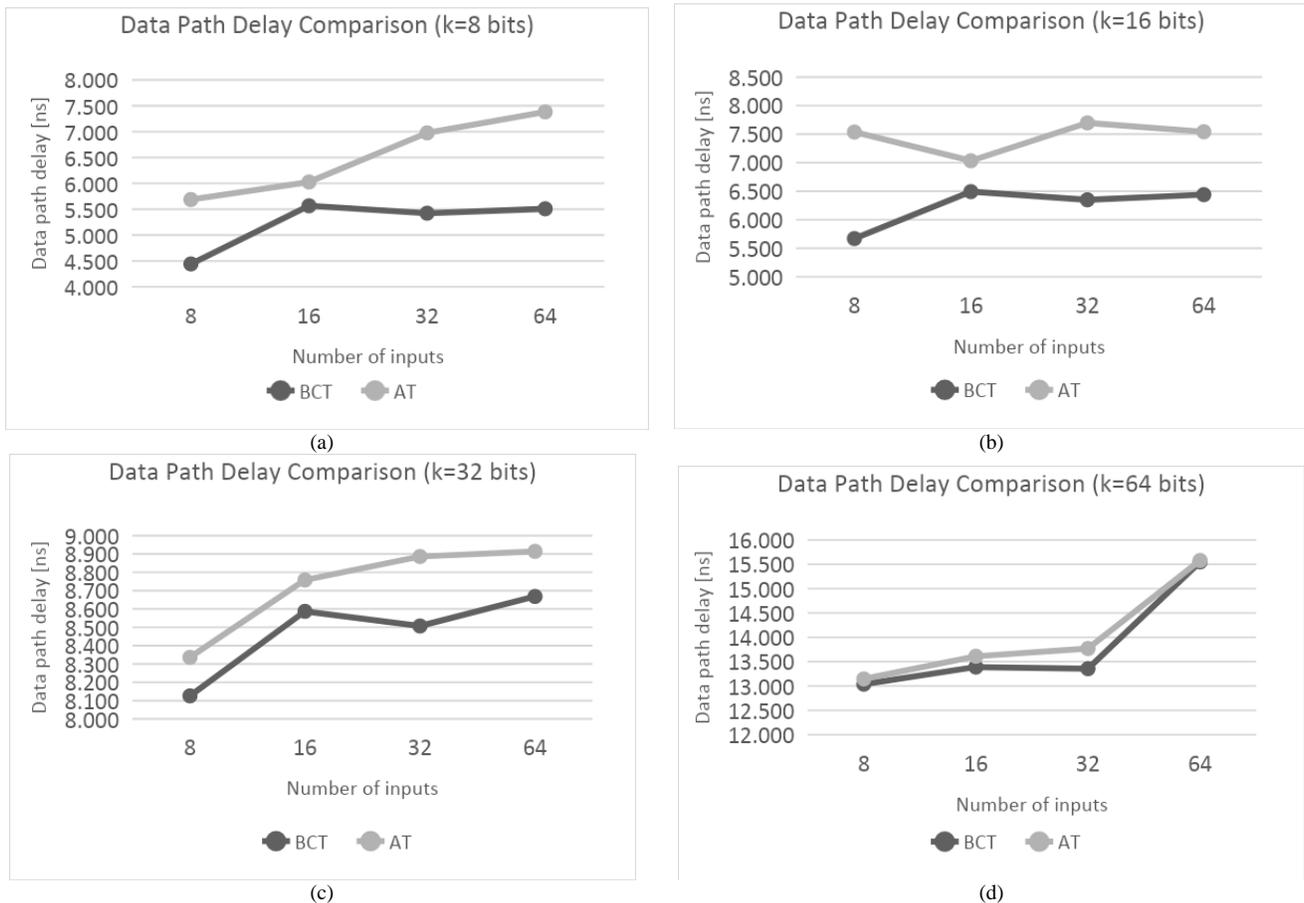


Fig. 9. Propagation Delay Comparison [ns].

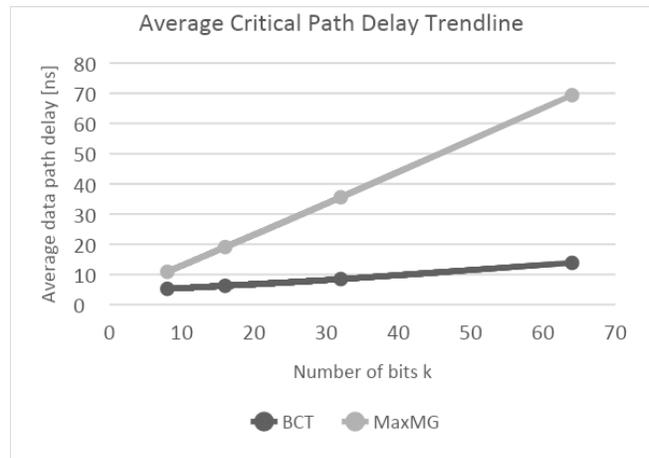
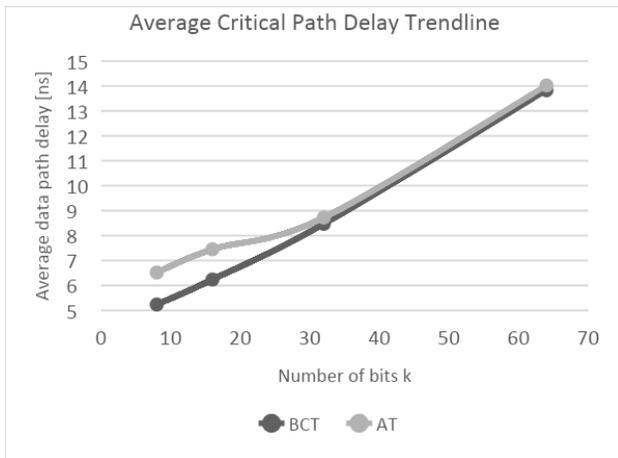


Fig. 10. Average Propagation Delay Comparison. (a) BCT and AT, (b) BCT and MaxMG

C. Power Consumption

Dynamic power is energy that a digital circuit uses for its work. It represents the power consumed when the design runs, being calculated as the sum of the power supplied on logic and for input/output signals.

Table IV summarizes the results on the dynamic power consumption for ASIC circuit. The average power consumption is almost the same for BCT and AT as expected. MaxMG is considered one of the most energy efficient topologies in [17] outperforming tree-based and ABT-based circuits. However, in our experiment, MaxMG is almost three times slower than AT and BCT. Due to the larger number of logic gates and area, it has the highest switching activities causing the largest dynamic power consumption and power consumed on logic, especially with larger number of bits in the input code word. The average power trendlines for BCT and referent designs are given in Fig. 11 and visualizing the power comparison result.

D. Area and Power Delay Product (ADP and PDP)

Power-Delay-Product (PDP) and Area-Delay-Product (ADP) are commonly used to determine quality of a synthesized digital circuit [24-26]. PDP is a product of the average power consumption and gate delay. It estimates energy used per operation and estimates the average energy consumed per switching event [27]. ADP expresses trade-off between circuit area and average delay, as the main goal in the design of a digital circuit is to minimize area per logic gate.

It is shown in Tables V and VI that BCT is the best circuit in terms of standard measures for the quality of the logical design, namely, BCT outperforms the referent architectures, offering the minimized ADP and PDP metrics. It is straightforward since BCT has the smallest time and area complexity compared to MaxMG and lower delay and lower or equal area compared to AT when synthesized on ASIC TSMC 65 nm technology. Also, BCT is slightly more power efficient than AT in 50% of the measurement cases in our experiment. This provides an improvement when PDP is calculated, considering the lower propagation delay.

TABLE IV. POWER CONSUMPTION [NW]

| k,n   | MaxMG     | AT        | BC               |
|-------|-----------|-----------|------------------|
| 8,8   | 5232,62   | 6541,54   | <b>5961,82</b>   |
| 8,16  | 15357,74  | 10120,18  | <b>9993,22</b>   |
| 8,32  | 31692,88  | 19859,73  | <b>18919,74</b>  |
| 8,64  | 77495,90  | 44881,81  | <b>41692,10</b>  |
| 16,8  | 14003,28  | 7959,35   | <b>7754,28</b>   |
| 16,16 | 36124,69  | 21215,25  | <b>20935,51</b>  |
| 16,32 | 69477,02  | 30746,31  | <b>31330,02</b>  |
| 16,64 | 152778,94 | 75189,38  | <b>76165,04</b>  |
| 32,8  | 45535,67  | 13174,44  | <b>13427,08</b>  |
| 32,16 | 91966,01  | 22596,36  | <b>22724,47</b>  |
| 32,32 | 175197,80 | 47303,70  | <b>48112,36</b>  |
| 32,64 | 316724,20 | 109041,50 | <b>109650,40</b> |
| 64,8  | 60469,86  | 21380,60  | <b>21353,39</b>  |
| 64,16 | 140307,13 | 38008,61  | <b>38225,02</b>  |
| 64,32 | 293379,06 | 81205,02  | <b>82651,28</b>  |
| 64,64 | 738234,36 | 167856,56 | <b>167039,52</b> |

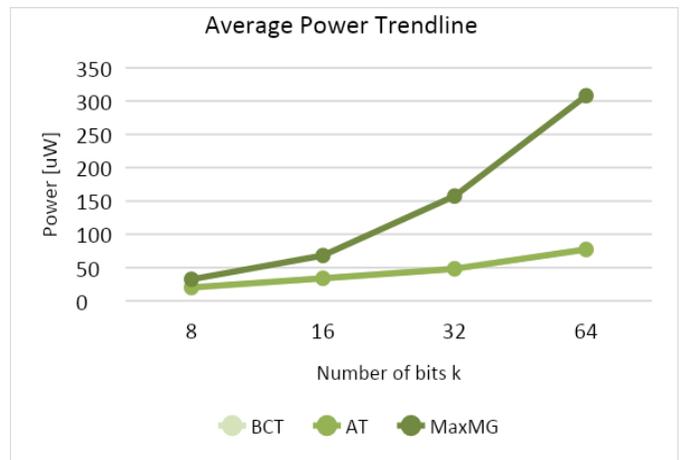


Fig. 11. Power Comparison.

TABLE. V. ADP RESULT [ $\mu\text{m}^2 \cdot \text{US}$ ]

| k,n   | MaxMG   | AT     | BC            |
|-------|---------|--------|---------------|
| 8,8   | 2,91    | 1,91   | <b>1,18</b>   |
| 8,16  | 6,58    | 3,57   | <b>2,97</b>   |
| 8,32  | 14,57   | 8,53   | <b>5,74</b>   |
| 8,64  | 34,01   | 18,75  | <b>11,65</b>  |
| 16,8  | 11,43   | 4,75   | <b>3,05</b>   |
| 16,16 | 20,15   | 7,94   | <b>7,00</b>   |
| 16,32 | 41,71   | 17,62  | <b>13,58</b>  |
| 16,64 | 84,35   | 34,48  | <b>27,52</b>  |
| 32,8  | 44,72   | 9,09   | <b>8,76</b>   |
| 32,16 | 98,51   | 18,74  | <b>18,55</b>  |
| 32,32 | 154,47  | 38,12  | <b>36,55</b>  |
| 32,64 | 309,25  | 76,18  | <b>74,42</b>  |
| 64,8  | 177,02  | 28,72  | <b>28,18</b>  |
| 64,16 | 385,08  | 58,31  | <b>57,95</b>  |
| 64,32 | 598,77  | 118,39 | <b>114,98</b> |
| 64,64 | 1390,11 | 303,11 | <b>302,20</b> |

TABLE. VI. PDP RESULT [ $\mu\text{J}$ ]

| k,n   | MaxMG    | AT      | BC             |
|-------|----------|---------|----------------|
| 8,8   | 51,94    | 37,20   | <b>26,48</b>   |
| 8,16  | 158,45   | 60,98   | <b>55,62</b>   |
| 8,32  | 367,38   | 138,52  | <b>102,60</b>  |
| 8,64  | 904,53   | 331,27  | <b>229,64</b>  |
| 16,8  | 256,69   | 59,98   | <b>43,97</b>   |
| 16,16 | 675,35   | 149,21  | <b>135,91</b>  |
| 16,32 | 1355,08  | 236,65  | <b>198,91</b>  |
| 16,64 | 3014,94  | 567,08  | <b>490,43</b>  |
| 32,8  | 1589,33  | 109,82  | <b>109,11</b>  |
| 32,16 | 3252,65  | 197,90  | <b>195,14</b>  |
| 32,32 | 6298,54  | 420,34  | <b>409,29</b>  |
| 32,64 | 11408,09 | 972,00  | <b>950,45</b>  |
| 64,8  | 4118,42  | 281,03  | <b>278,41</b>  |
| 64,16 | 9584,94  | 517,30  | <b>511,79</b>  |
| 64,32 | 20403,34 | 1118,11 | <b>1103,81</b> |
| 64,64 | 52799,26 | 2614,53 | <b>2597,13</b> |

### VII. LIMITATIONS OF THE STUDY

The motivation for our application is in the Compact Muon Solenoid (CMS) detector design upgrade project [27]. BCT is tested as a selection algorithm applied in the front-end detector readout electronics chain. It offered a great potential to be implemented in front-end part of the concentrator ASIC design for selecting the data sent for further processing in the back-end architecture [28].

Current limitation of the study is that we only consider selection cases where a single maximum extraction is needed on a specific position. However, there exist many applications in which M largest numbers from N inputs must be selected. This is common in high-energy physics experiments and there are some dedicated solutions by using design parameterized

and pipelined sorting units [29, 30]. Also, when several equal maximums are present in the input dataset, current BCT design provides a maximum value and an indicator array with all maximums' positions marked with the high bit. This means that, like in any array-based topology, if multiple equal winners exist in the input dataset, it requires some additional priority logic to filter out the most significant winner position.

### VIII. CONCLUSION

In this paper, we propose a new optimized array-based topology BCT, used to extract maximum or minimum value from an unsorted set of N binary numbers, as well as its one-of-N binary address. BCT is based on the same concept as AT, but it is more efficient in timing, especially for smaller width of the input code word  $k < 32$ . This makes BCT particularly important for applications that usually deal with these input sizes, such as high-speed network packet schedulers or the winner-take-all circuits. From the experimental results in the paper, the conclusion is two-fold. First, it is confirmed that BCT is indeed an optimized AT solution. Second, we show that it outperforms the recent referent design when synthesized on ASIC 65 nm technology. BCT extracts winner among N k-bit numbers with lower area and latency, consuming the smallest dynamic power. This makes BCT the best option among the existing array-based competitors when synthesized on ASIC technology.

### IX. FUTURE WORK

To deal with the current limitations of the study, we will follow several additional future research directions. First, a research is foreseen on using BCT for data selection in the context where M of N elements must be extracted. It would be interesting to vary the (M, N) parameter pairs and to examine their influence on area, delay and power. Next, we will study the resource usage when priority logic is added to BCT, so that only a single maximum position is marked, where the one of the simplest implementations can be encoder-based. Finally, as this paper reveals a great potential of BCT to be used as an efficient maximum finder circuit in some real-time signal processing application, for our future work, we will consider using BCT in the rank order filter design. Due to simplicity of logic and fast extraction of the maximum, it is expected to obtain better results than existing research with MaxMG [31].

### REFERENCES

- [1] Monemi, A., Ooi, C. Y., Palesi, M., & Marsono, M. N. (2017). Ping-pong round robin arbiter. *Microelectronics Journal*, 63, 81-93.
- [2] Azuma, Masaki and Hiroomi Hikawa. "Supervised learning of DPLL based winner-take-all neural network." 2014 IEEE International Conference on Evolvable Systems (2014): 117-124.
- [3] Fan, R., Zhang, Y., Li, B. (2017). Motion Classification-Based Fast Motion Estimation for High-Efficiency Video Coding. *IEEE Trans. on Multimedia*, 19(5), 893-907. doi:10.1109/tmm.2016.2642786.
- [4] Gaitan, V. G., Gaitan, N. C., & Ungurean, I. (2015). CPU Architecture Based on a Hardware Scheduler and Independent Pipeline Registers. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(9), 1661-1674. doi:10.1109/tvlsi.2014.2346542.
- [5] Raghavan, R., & Perera, D. G. (2017). A fast and scalable FPGA-based parallel processing architecture for K-means clustering for big data analysis. 2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM). doi:10.1109/pacrim.2017.8121905.

- [6] Rongchun Li, Yong Dou, Dan Zou, Shi Wang, and Ying Zhang. (2015). Efficient graphics processing unit based layered decoders for quasicyclic low-density parity-check codes. *Concurr. Comput. : Pract. Exper.* 27, 1 (January 2015), 29-46. Doi: 10.1002/cpe.3193, <http://dx.doi.org/10.1002/cpe.3193>.
- [7] Dimitrakopoulos, G., Kalligeros, E., Galanopoulos, K., "Merged switch allocation and traversal in network-on-chip switches", *IEEE Trans. Comput.*, 2013, 62, (10), pp. 2001–2012.
- [8] Chakrabarti I, Batta K.N.S., Chatterjee S.K. (2015) Efficient Pixel Truncation Algorithm and Architecture. In: *Motion Estimation for Video Coding. Studies in Computational Intelligence*, vol 590. Springer, Cham, [https://doi.org/10.1007/978-3-319-14376-7\\_6](https://doi.org/10.1007/978-3-319-14376-7_6).
- [9] Hu, C., Ye, M., Du, Y., Lu, X. "Vector projection for face recognition", *Comput. Electr. Eng.*, 2014, 40, (8), pp. 51–65.
- [10] K. Vasanth, E. Sindhu, R. Varatharajan, VLSI architecture for Vasanth sorting to denoise image with minimum comparators, *Microprocessors and Microsystems*, Volume 71, 2019, 102880, ISSN 0141-9331, <https://doi.org/10.1016/j.micpro.2019.102880>.
- [11] Farmahini-Farahani A., Duwe III, H.J., Schulte, M.J., Compton, K. "Modular design of high-throughput, low-latency sorting units", *Computers*, *IEEE Trans.*, vol. 62, no. 7, pp. 1389–1402, 2013.
- [12] Kante, R.K., Thrimurthulu "Efficient Sorting Mechanism for Finding First W Maximum/Minimum Values", *Manager's Journal on Embedded Systems; Nagercoil Vol. 3 (3)*, (Aug-Oct 2014), pp. 39-44.
- [13] Zuluaga, M., Milder, P., & Püschel, M. (2016). Streaming Sorting Networks. *ACM Transactions on Design Automation of Electronic Systems*, 21(4), 1–30. doi:10.1145/2854150.
- [14] Genus Synthesis Solution, Genus Cadence Official, October 2019, [https://www.cadence.com/content/cadence-www/global/en\\_US/home/tools/digital-design-and-signoff/synthesis/genus-synthesis-solution.html](https://www.cadence.com/content/cadence-www/global/en_US/home/tools/digital-design-and-signoff/synthesis/genus-synthesis-solution.html)
- [15] Yuce, B., Ugurdag, H. F., Goren, S., Dundar, G., "Fast and Efficient Circuit Topologies for Finding the Maximum of n k-Bit Numbers", (2014), *Computers*, *IEEE Transactions on.* 63. 1868-1881. 10.1109/TC.2014.2315634.
- [16] Smrithi SV., Agrawal, S. "A fast architecture for maximum/minimum data finder with address from a set of data," 2016 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2016, pp. 1-6. doi: 10.1109/ICCCI.2016.7479988.
- [17] Kathirvel, S., Jangre R., and Ko, S. "Design of a novel energy efficient topology for maximum magnitude generator", in *IET Computers and Digital Techniques*, vol. 10, no. 3, pp. 93-101, 5 2016.
- [18] Huang, X. P., Fan, X. Y., Zhang, S. B., and Zhang, F. (2010, November). An optimized tag sorting circuit in WFQ scheduler based on leading zero counting. In 2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology (pp. 533-535).
- [19] Amaru, L.G., Martina, M., Masera, G. "High speed architectures for finding the first two maximum/minimum values", (2012). In: *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 20 n. 12, pp. 2342-2346. ISSN 1063-8210.
- [20] Biroli, A.D.G. and Wang, J. C. "A fast architecture for finding maximum (or minimum) values in a set", 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, 2014, pp. 7565-7569.
- [21] Xiao, G., Martina, M., Masera, G., and Piccinini, G. "A Parallel Radix-Sort-Based VLSI Architecture for Finding the First W Maximum/Minimum Values", in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 890-894, Nov. 2014.
- [22] Vai, Michael & Moy, Man. (1993). Real-Time Maximum Value Determination on an Easily Testable VLSI Architecture. *Circuits and Systems I: Fundamental Theory and Applications*, *IEEE Transactions on.* 40. 283 - 285. 10.1109/81.224304.
- [23] Vinnakota, B., Rao, V.V. B. "A new circuit for maximum value determination", *IEEE Trans. Circuits Syst. I*, vol. 41, pp. 929-930, Dec. 1994.
- [24] Kiran, N.K. and Alok K. "Leakage Power Reduction and Power Delay Product (PDP) Improvement using Dual Stack Method." *International Journal of Engineering and Technical Research (IJETR)*, ISSN: 2321-0869 (O) 2454-4698 (P), Volume-4, Issue-4, April 2016.
- [25] Meher, P. K. and Park, S. Y. "Area-Delay-Power Efficient Fixed-Point LMS Adaptive Filter With Low Adaptation-Delay," in *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 362-371, Feb. 2014. doi: 10.1109/TVLSI.2013.2239321.
- [26] Praveen, P., Zahid, A. "Variation of Power and Delay in Digital CMOS Circuit Design in DSM Technology", *International Journal of Engineering Trends and Technology (IJETT)*, V45(9),449-453 March 2017. ISSN:2231-5381.
- [27] Bilki B. and the CMS Collaboration (2015): " CMS Forward Calorimeters Phase II Upgrade". *Journal of Physics: 16th International Conference on Calorimetry in High Energy Physics (CALOR 2014)*, Conf. Ser. 587 012014, doi:10.1088/1742-6596/587/1/012014.
- [28] CMS Collaboration: "The Phase-2 Upgrade of the CMS endcap calorimeter Technical Design Report", CERN-LHCC-2017-023, CMS-TDR-019, ISBN 978-92-9083-459-5, 2018.
- [29] Farmahini-Farahani, A., Gregerson, A., Schulte, M., Compton, K. "Modular high-throughput and low-latency sorting units for FPGAs in the Large Hadron Collider." 2011 IEEE 9th Symposium on Application Specific Processors (SASP) (2011): 38-45.
- [30] T. N. Van, V. T. Thien, S. N. Kim, N. P. Ngoc and T. N. Huu, "A high throughput pipelined hardware architecture for tag sorting in packet fair queuing schedulers," 2015 International Conference on Communications, Management and Telecommunications (ComManTel), DaNang, 2015, pp. 41-45. doi: 10.1109/ComManTel.2015.7394257.
- [31] Narasimha-Murthy, P., and Suryakala, S. V. "Rank order filter design using maximum magnitude generator." *Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017 International Conference on. IEEE, 2017.