International Journal of Research

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 05, May 2015
Available at http://internationaljournalofresearch.org

Hardware Modeling of Sorting Mechanism for Finding First Maxima/Minima Values from a Set Greater Than 2

[1] Kante Raj Kumar & [2]Dr.V.Thrimurthulu.
[1] II M.Tech VLSI SD Student, CR Engineering College, Tirupati, Chittoor(Dist) A.P, India,
[2] Professor, Head of ECE Dept., CR Engg., College, Tirupati, Chittoor(Dist)A.P, India,
EMAIL: [1]rk.kante@gmail.com, [2] vtmurthy.v@gmail.com

Abstract:

with this document, we have proposed a complete VLSI model of sorting mechanism to find first maxima/minima values using (BWA) Bit Wise And architecture.In this effort 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 with an added advantage of high level scalability depends onanalyzing input data from the most significant bit to the least significant one, the proposed scheme along with parallel structure has been modelled successfully on Xilinx Virtex -5 FPGA with a simple verilog RTL description.

Key Words:

Sorting; Bit Wise and Operation; verilog; RTL description and Xilinx Virtex -5 Fpga

Introduction:

Sorting is a deep-rooted problem in computer science and is a mean of operation in quite a lot of applications. Along with hardware implementation of sorting networks has been identified as well in the history. On the other side, VLSI architectures for partial sorting, which can also be derived from assortment networks (AN), are part of different algorithms in the communication field. Partial sorting is employed, for instance, in and for the decoding of turbo and binary Low-Density-Parity-Check (LDPC) codes, in for maximum-likelihood decoding of arithmetic codes and in for K-best MIMO detectors, non-binary LDPC decoders and turbo product codes respectively. Designs for finding the first two minimum values are used in binary LDPC decoder architectures to implement minterm Sum approximations and recently they have also been proposed for the case of non-binary LDPC decoders. However, very a small number of works, probe the general problem of implementing parallel architectures for finding the first two maximum/minimum values with a logical approach. Similarly, architectures for finding the first W which is greater than maximum/minimum values in a set of *M* elements are of now a days interest. It has been observed by a review of literature survey that the sorting mechanism implemented with a variety of applications such as comparator based, selection based, partial based and radix based. Radix sorting algorithms depends and works on the bit-wise scrutiny of the data to besorted and can be absolute to selection and partial sorting problems. Unfortunately all these algorithms are not up to the bottle neck related to the sorting job whose data size is greater than 2. The proposed design was suggested with Radix based sorting along with parallel architecture can overcome afore mentioned problem. The anticipated elucidation, referred to as Bit-Wise-And(BWA) architecture, works by analyzing the set ofcandidates beginning from the Most-Significant-Bit (MSB) to the Least-Significant-Bit (LSB).

The problem statement can be identified as described below, let for a given set of data D(k) = {d0,….., dk−1} prepared of kelements, now if we wish to notice the first X number of maximum/minimum values, namely x(k) = {x(k)0; x(k)1,….x(k) l, x(k)X−1} where x(X)0 =max(X(X)), x(X)1 = max(X(X) \ {x(X)0}), . . . , x(X)l=max(X(X) \Vl−1k=0{x(X)s}), . . . , x(X)X−1 = max(X(X) \VW−2k=0{x(X)l}). Similarly the same statement can be applied for minimum number of data sorting procedure.

Design development:

Earlier Bit Wise And operation works with the analogy of Radix sorting which depends on the analysis of x(k) values bit by bit from the MSB to the LSB along with the assumption that the values in x(k) are all positive. And for negative or non-positive data the assumptions will be acceptable with two's compliment notations preserving the order relation, by flipping the MSB bit and entire operation woks bit by bit, where as the proposed architecture relays on array of data operation works in parallel.
The proposed architecture also overcomes the problem of zero-row condition.

TABLE 1
SUMMARY OF DESIGN CONSIDERATIONS

Sno	Design consideration	Selection
1	Compiler	Xilinix14.4Vivado
2	Programming Language	Verilog
3	Architecture	Parallel BWA
4	FPGA	Virtex -5
5	Interface	USB

Modeling of proposed architecture:

As mentioned earlier, the (BWA) Bit Wise And architecture along with parallel Radix sort having the two methods: applying data from monotonic sequences and implementing proposed logic operation in parallel. The first method is used to implement more data values and the second method is used to unite both having the following principle of operations and to enhance its physical parameters such as speed and utilization power. The falconer cannot hear the falconer

Bit Wise AND Operation:

Computer programming applications, the most popular bit wise operations operates on one or more bit patterns or binary numerals at the level of their entity bits. It is a fast, primitive action directly supported by the processor, and is used to influence values for comparisons and calculations. On simple low-cost processors, typically, bitwise operations are substantially faster than division, quite a lot of times faster than multiplication, and every so often significantly faster than addition. While modern processors typically perform addition and multiplication just as fast as bitwise operations due to their longer instruction pipelines and other architectural design choices, bitwise operations do commonly use less power because of the reduced use of resources.

A bitwise AND takes two equal-quantity of binary representations and evaluates the logical AND operation on each pair of the analogous bits, by multiplying them. Thus, if both bits in the compared position are 1, the bit in the resulting binary representation is 1 (1 × 1 = 1); otherwise, the result is 0 (1 × 0 = 0). For example:

1101 (decimal 13)

AND 1011 (decimal 11)
= 1001 (decimal 9)

Parallel Radix Sort Operation:

Radix sort is one of the linear sorting algorithms for integers. It works with sorting the input numbers on each digit, for each of the digits in the numbers. However, the process adopted by this sort method is somewhat counterintuitive, in the sense that the numbers are sorted on the least-significant digit first, followedby the second-least significant digit and so on till the most significant digit.The time complexity of the algorithm is as follows: Suppose that the n input numbers have maximum k digits. Then the Counting Sort procedure is called a total of k times. Counting Sort is a linear or O(n) algorithm. So the entire Radix Sort procedure takes O(kn) time. If the numbers are of finite size, the algorithm runs in O(n) asymptotic time.Radix Sort (v, k) includes the following two stages of work
1. For i - 1 to k
2. Do use a stable sort to sort Array v on digit k.

The parallelization and its implementation can be done as shown in the following procedure

Parallelization:

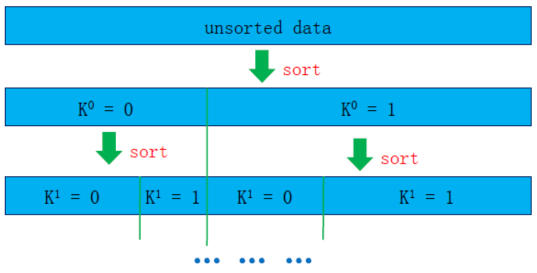


Figure1: Radix sort parallelization

- MSB digit is at first
- There May be Large or Small Pieces of Data
- It is not an easy process to make all the pieces to be sorted fast and efficiently.

Implementation:

The parallelization of Radix sort can be implemented as described below stages

- The most interesting part is the counting
- At first a structure tag is used to count the elements
- Tag is a dyadic array unsigned integer of [K + 1][n]
- K is the number of threads and n is the number of possible values of K^t.
- Parallel stable sort the array on Kt.
- Step 1:Every thread goes through its piece of data, and counts
  Elements with K^t= (possible value 1),
  Elements with K^t= (possible value 2),
  Elements with K^t= (possible value 3),
  ………..

Elements with K^t= (possible value n).

- Step 2:After all the threads finish their counting, get all the numbers and generate some start positionsfor these pieces.

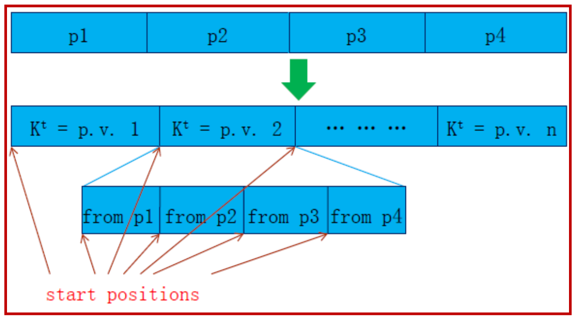


Figure2: Radix sort parallelization

- Step 3:With these start positions, every thread goes through its piece of data again and move the elements to an extra array, and make the extra array sorted on digit K^t.

- When thread k come to an element withK^t= p.v. e,
  It increase tag [k+1][e].

The process repeats up to the final stage and is as shown in following figure

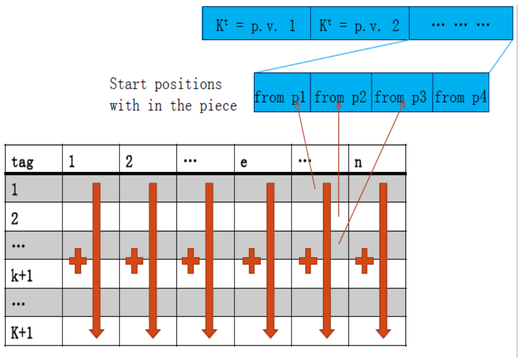


Figure3: Final stage of implementation

The parallel Radix sort evaluation is predicted in the following graph

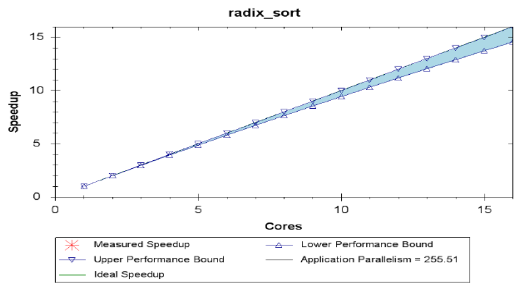


Figure4: Radix Sort evaluation

Front-End Modeling:

This phase of implementation contains the following stages simulation using Xilinx 14.4 Vivado suite, synthesis using Xilinx 14.4 XST and verifying on Virtex – 5 FPGA board.

VLSI Structural design for proposed method:

The intention of parallel Radix Sort based BWA technique is to create high speed VLSI architecture for sorting applications. The combined architecture is shown in the figure5. The proposed scheme is descripted using Verilog HDL with RTL coding style and is simulated and synthesized with Xilinx tools. The design is successfully verified on Virtex -5 FPGA board.

I. Simulation and Synthesis Results

The Verilog RTL Description of the above article is simulated and synthesized using Xilinx14.4 (ISE-Simulator), it is noticed that at some instance of simulation time the set of data is sorted in parallel and various results are shown in the following figures.

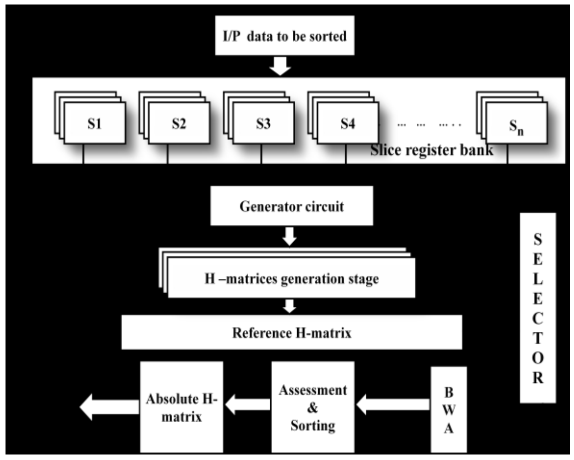


Figure5: Proposed Architecture

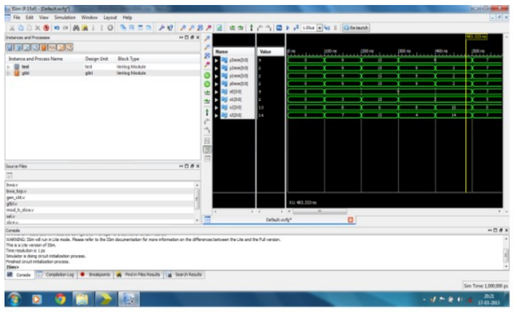


Figure6. Simulation output

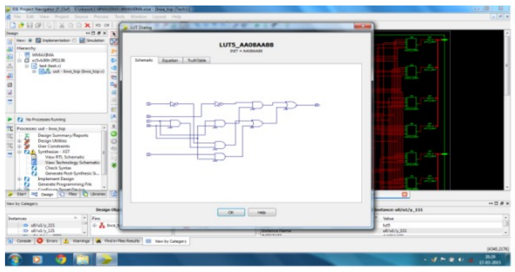


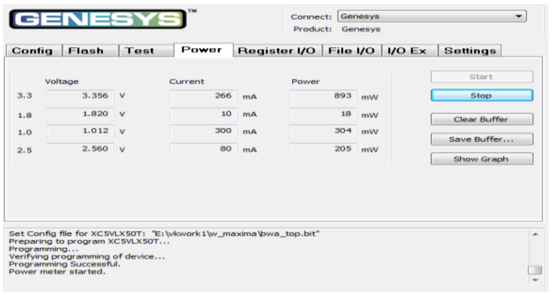
Figure8. Synthesized output

The Synthesized report is summarized in the following Table.

TABLE 2
SUMMARY OF SYNTHESIS REPORT

Sno	Parameter	Quantification
1	Target Device	xc5vlx50t-2-ff1136
2	Slice Logic utilization	Less than 5%
3	Slice logic distribution	100%
4	IO utilization	6%
5	Specific feature factor	30%
6	Total logic delay	11.035 nS
7	Total offset delay	11.035 nS
8	Total path delay	6.945 nS
9	Real time compilation	7Sec
10	Total memory usage	282028 Kb

Results of on Xilinx Virtex-5 FPGA board verification



	Voltage		Current		Power	
3.3	3.356	V	266	mA	893	mW
1.8	1.820	V	10	mA	18	mW
1.0	1.012	V	300	mA	304	mW
2.5	2.560	V	80	mA	205	mW

```
Set Config file for XC5VLX50T:  "E:\vkwork1\w_maxima\bwa_top.bit"
Preparing to program XC5VLX50T...
Programming...
Verifying programming of device...
Programming Successful.
Power meter started.
```

Fig 9: power analysis on Xilinx Virtex-5 FPGA board

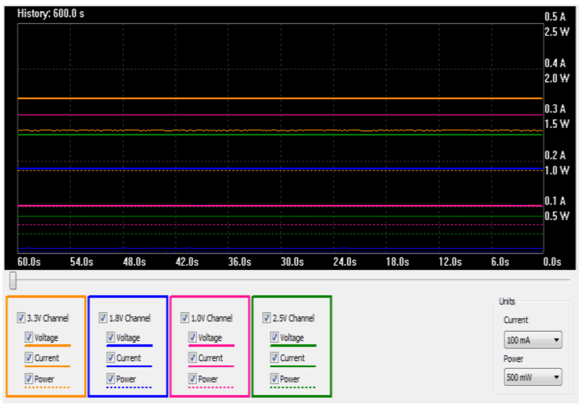


Fig10: power utilization graph

Conclusion: This paper analyzes the VLSI hardware modeling of parallel Radix Sorting using BWA architecture, the modified architecture relies on bit wise illusion of input data depends on finite number of multistage where each stage is designed with simple unit, the proposed architecture is descripted using verilog HDL and successfully verified on Xilinx Virtex – 5 FPGA.

References:

[1]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*, GianlucaPiccinini

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

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

[4] S. Papaharalabos, P. T. Mathiopoulos, G. Masera, and M. Martina,“Novel non-recursive max* operator with reduced implementation complexityfor turbo

decoding," *IET Communications*, vol. 6, no. 7, pp.702–707, Jul 2012.

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

[6] F. Guilloud, E. Boutillon, and J. L. Danger, "_-min decoding algorithmof regular and irregular LDPC codes," in *International Symposium onTurbo Codes and Related Topics*, 2003, pp. 451–454.

[7] 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.

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

[9] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-bestsphere decoding for MIMO detection," *IEEE Journal on Selected Areasin Communications*, vol. 24, no. 3, pp. 491–503, Mar 2006.

AUTHORS

KANTE RAJ KUMAR received his BTech degree in Electronics & Comm. 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, Tirupathi (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.