e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 6, Issue 2, February 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.54

International Journal Of Multidisciplinary Research In Science, Engineering and Technology (IJMRSET)

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602001 |

Self-Checking PIPO Sorter with Odd-Even Trapezoid

Husayn Abo showfa[1], Osama A.Alhashi[2], Alhade A.Algitta[3], Abobakar B.Zargoun[4], Ali H. Maamar[5]

Lecturer Assistant, Dept. of Computer Engineering, College of Electronics Technology, Bani Walid, Libya[1]

Lecturer Assistant, Dept. of Computer Engineering, College of Electronics Technology, Bani Walid, Libya[2]

Associate Professor, Dept. of Control Engineering, College of Electronics Technology, Bani Walid, Libya[3]

Assistant Professor, Dept. of Control Engineering, College of Electronics Technology, Bani Walid, Libya[4]

Professor, Dept. of Computer Engineering, College of Electronics Technology, Bani Walid, Libya[5]

ABSTRACT: Many applications in real-time systems need a high performance and special purpose architecture for parallel sorting on a huge amount of input data. Sorting arrays which consist of a number of identical processing elements with regular interconnections are good candidates for real-time applications. Using these arrays has become attractive mainly due to the availability of VLSI [1].

Today, many of the applications in which VLSI/ULSI technology is used are deemed to be safety critical. Unfortunately, as the scale of integration has increased, the occurrence of temporary (transient and intermittent) errors is becoming a dominant failure mode in digital circuit (VLSI/ULSI) [2]. The characteristics of these faults necessitate the use of a test strategy, which continuously monitors the operation of the circuit and compare it with some known reference. This approach is usually referred to as Concurrent Error Detection (CED) approach [3]. In this paper we propose to design a self-checking odd-even trapezoid sorter using Berger code.

KEYWORDS: Berger Code, odd-even trapezoid algorithm, Self-Checking.

I.INTRODUCTION

Sorting is the process of reordering a sequence taken as input and producing one that is ordered according to an attribute. Parallel sorting is the process of using multiple processing units to collectively sort an unordered sequence [4]. Sorting networks which consist of a number of identical processing elements with regular interconnections are good candidates for real-time applications. Use these arrays has become attractive mainly due to the availability of VLSI/ULSI [1].

Unfortunately, as the scale of integration has increased so also has the occurrence of intermittent faults. The detection of intermittent faults requires the use of concurrent Error Detection (CED) techniques.[5]

CED is the process of detection errors at the same time as the system is performing its normal operation and that can be achieved through the use of redundancy [2]. Redundancy is the use of extra resources beyond the requirements of the unchecked system. There are three types of redundancy, namely; hardware redundancy, time redundancy, and information redundancy [3].

Information redundancy is an extra bit (check bits or check symbol) added to the informationbits (data word) to form a codeword[3]. Information redundancy (coding techniques) has been identified as a viable mechanism for implementing concurrent error detection (CED) in VLSI/ULSI circuits[2]. Amongst all of the separable codes used in CED schemes, Berger code [3] is the least redundant separable code capable of detecting all unidirectional errors. The construction of the code, and its error detection capabilities are discussed below together with a design of a self checking odd-even trapezoid sorter using Berger code.

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602001 |

II. BERGER CODE

Berger code[6] is an optimal separable code which can detect all unidirectional errors. A Berger code word of length n bits has I information bits and k check bits:
where [k=log 2 (I+1)], n=I+k.
Berger codes are useful for encoding the information bits in digital systems because:
1-They are separable codes. No extra decoders are required to extract the information bits, when needed for processing, from the code word.
2- They detect all unidirectional errors; these are most likely to occur in digital systems.
3- They are optimal, in terms of the number of check bits required for I information bits, amongall the separable codes that detect unidirectional errors [2].
The codeword of the Berger code is formed by appending the check bits to the information bits, the check bits of the code is the binary representation of the number of O's (or the complement of the number of 1's) in the information bits.

III. SHIFT REGISTER

A shift register is a digital device used for storage and transfer of data. The data to be stored could be the data appearing at the output of an encoding matrix before they are fed to the main digital system for processing or they might be the data present at the output of a microprocessor before they are fed to the driver circuitry of the output devices [7]
The basic building block in all shift registers is the flip flop, mainly a D-type flip-flop and the storage capacity of the shift register equals the number of flip-flops used.
Based on the method used to load data onto and read data from shift registers, they are classified as serial-in serial-out (SISO) shift registers, serial-in parallel-out (SIPO) shift registers, parallel-in serial-out (PISO) shift registers and parallel-in parallel-out (PIPO) shift registers. In this paper, a PIPO register will be used to design the self-checking odd-even trapezoid sorter because the PIPO register is faster than an another types of shift register.
In this paper the PIPO shift register will be used.

IV.ODD-EVEN TRANSPOSITION ALGORITHM

The odd-even transposition sort is designed for the processor array model in which the processing elements are organized into a one-dimensional mesh.[4]

```
Local a //element to be sorted
t  //element taken from adjacent processor
Begin
for  ifrom 1 to n/2 do
for all pj where 0<=j<=n-1 do
if j<n-1 and odd(j) then //odd-even exchange
Successor(a)=t                // get value from successor
Max(a,t)=successor(a)   // give away larger value
A=max(a,t)                    // keep max value
Endif
If even(j) then
Successor(a)=t                // get value from successor
Max(a,t)=successor(a)   // give away larger value
A=max(a,t)                    // keep max value
Endif
Endfor
Endfor
end
```

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602001 |

V. ODD-EVEN TRANSPOSITION ALGORITHM

PIPO sorter which are matrix arrangement of processing cells can be structured to sort parallel input. Each processor in this array has one registers A, and each processor can access the register of its two neighbors, as shown in figure (1). And each neighbors processor has a control unit. The PIPO sorter performs n/2 iteration and each iteration has two phases. In the first phase called odd exchange, the value of a in every odd numbered processors (except processors n-1)is compared with the value of a stored in the successors processor. The values are exchanged, if necessary, so that lower numbered processors contains the smear values. In second phase, called even exchange, the value of a in every even-numbered processor s compared with the value of a in the successor processor. After n/2 iteration the values must be sorted.

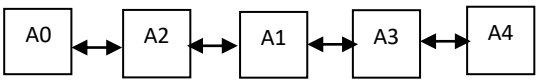

Figure (1) PIPO sorter

We will apply the odd-even transposition sort to design self-checking PIPO sorter. The PIPO sorter consists of array of cells, each cell consists of:

- One register (A) 15-bits of each.
- One register (CSRA) 4-bits of each
- 15-bits comparator circuit.
- Swapping circuit.
- Two Totally Self-checking circuits.
- Control unit between two neighbors.
- Up/down counter (for cell 0 only) used to show the status of the PIPO sorter.
- Buffers

VI. SELF CHECKING HARDWARE

The hardware of the checker circuit depends on the type of the redundancy to be used. In this paper, an information redundancy will be used to design the self-checking PIPO sorter.
Self-checking circuits allow on-line error detection, that means faults can be detected during the normal operation of the circuit. It can detect both transient and permanent faults.
In self-checking PIPO sorter, Two Totally Self-checking circuits are needed, one when data word is to be popped from registers (storage fault), and another after swapping operation (processing fault).
When new data word is pushed into the sorter, its check symbol should be generated and pushed into sorter at same time.

Self-Checking for storage faults

Figure (2) shows the block diagram of the self-checking for register Ai and Ai+1.

To move the operands from the registers A, and Ai+1 to the comparator circuit, the operands can be transferred directly into the input of the comparator circuit (CS) and at the same time a copy of the operands is moved to the CSG circuits, the comparator circuit can start processing the operands at the same time as the checker start to check the operands, if no errors in the operands are detected, then the result obtained from the comparator circuit can be taken as result of error free operands. If the checkers detect an error in the operands, an error signal will be activated to stop the compare circuit from passing the result.
The advantage of this method is that the total processing time is only equal to comparator circuit time, as the checker and processing of the operands is done in parallel.

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602001 |

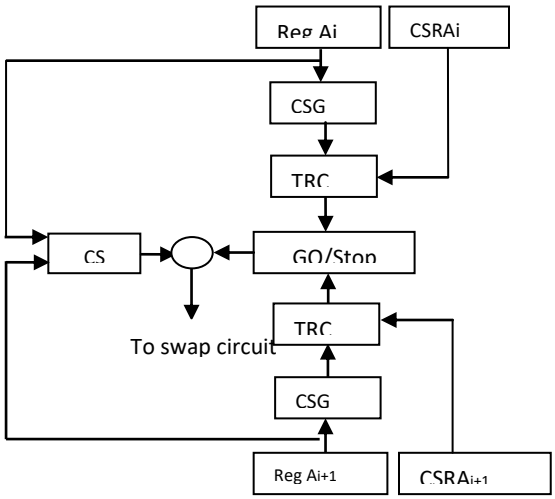

Figure (2) self-checking of storage faults

Self-Checking for processing fault

Figure (3) shows a block diagram of self-checking for processing faults.

After two operands of register A and Ai+1 are compared in comparator circuit, the output of the compare circuit is fed to swap circuit to exchange two operands if operand Ai+1 is greater than operand A, and then two operands are moved to the checker. If the checkers detect an error in the operands, an error signal will be activated to stop the buffer from passing the operand to the register.

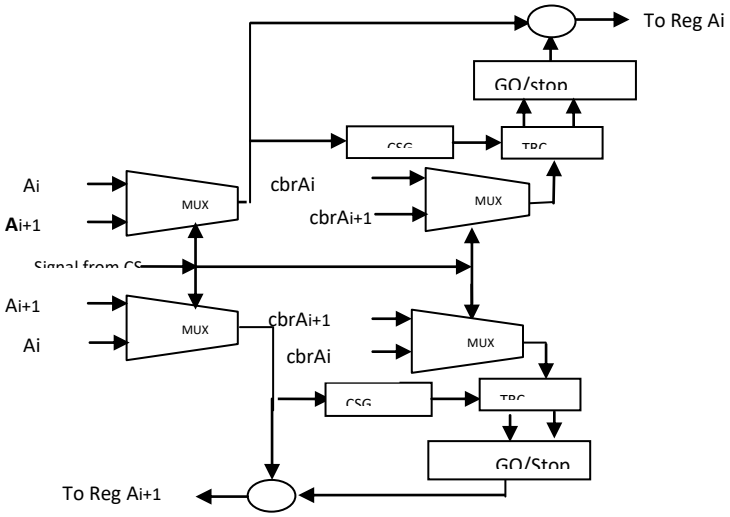

Figure (3) Self-Checking for processing fault

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602001 |

VII. CONTROL UNIT

Each two neighbors cells has its own control unit, as cells pass the information among themselves, and each control unit communicates with its left neighbor as well as its neighbor to its right. There are two types of control units: the main control unit, the state controller for the all cells (CUi).

The main control unit

The block diagram of CU0 is shown in figure (4)

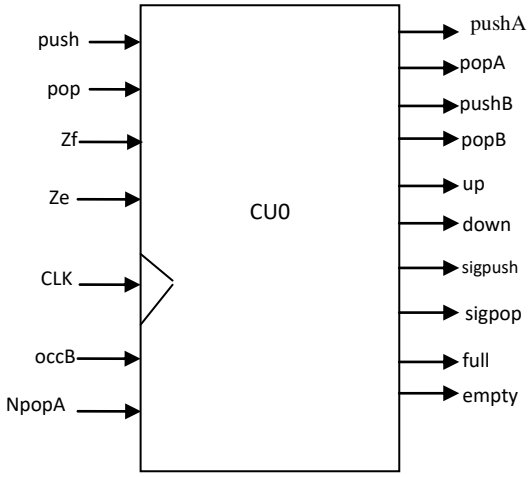

Figure (4) The block diagram of CU0

The function of CU0 is:

- To receive push signal from the host and allow new data words to be pushed into the PIPO sorter.
- To receive sort signal from the host and sends start_sort signal toCU1 and to Up cunter.
- To receive pop signal from the host and allow data words to be popped from the PIPO sorter,

Control unit of the there cells (CUi)

The block diagram of CU1 is shown in figure (5).
The control unit of cell1 receives push and pop signals from cell0 and NpopA from next cell and sends sigpush and sigpop signals to next cell and occB and NpopA signals to cell0.
The function of the control unit for cell1 is:

- receives the codeword from previous cell.
- execute the compare and swap operation with between two neighbors cell and send smallest value to next cell
- send sort signal to next to CUi+1

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602001 |

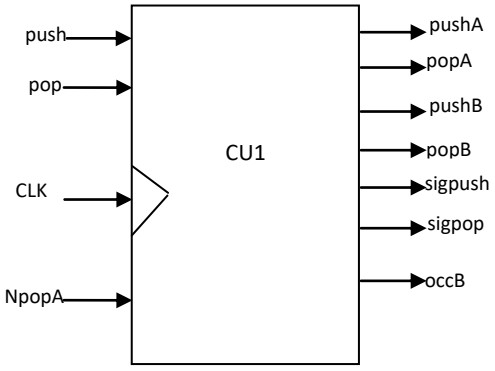

Figure (5) The block diagram of CU1

VIII. CONCLUSIONS

The work in this paper was concerned with the investigation of the effectiveness and the cost of using Berger code in self-checking PIPO sorter. The following remarks are concluded:

1. The penalty of using concurrent error detection could be time (Time redundancy), hardware (Hardware redundancy), or both (Information redundancy). Using coding techniques (Information redundancy) reducing the penalty time and the hardware penalty. In other words, information redundancy is the best choice when the cost of hardware and the time are both important and needed to be reduced.
2. Berger code is an optimal code to detect all unidirectional errors. In Berger code the information codeword ratio decreases while the information word length increases.
3. PIPO sorter provide some technological advances; smaller and regular units provide systems with more efficiency, and simplicity in fabrications.
4. As sorting array based on ODD-EVEN transposition, it has a regular structure and simple interconnection links. Both the regularity and the simplicity of ODD-EVEN transposition sorting array are preserved by the presented fault tolerance technique so that redundancy can fit into the system nicely.
5. All checkers are implemented to be fault secure or totally self-checking, it is well applicable to real-time application which require high throughput as well as high reliability.

REFERENCES

[1] Sheng-Chiech Liang, " High Yield and Reliable Sorting networks for VLSI and WSI Implementation ", The university of Arizona, 1991.
[2] A. H. Abdolhadi, Ali H. Maamar, " Self Checking Register File using Berger Code", 6the WSEAS Int. Conf. on circuits, systems electronics, control and signal processing, Cairo, Egypt, Dec 29-31-2007.
[3] Huda B. Abugharsa, "*self Checkingsystolic LIFO Stack*", , 7th WSEAS Int. Conf. on Instrumentation, Measurement, Circuit and Systems (IMCAS '08), Hangzhou, China, April 6-8- 2008.
[4] Ali H. Maamer, *" Lectures in Introduction to Computing*", 2009.
[5] Mohamed A. Abufalgha, " *Self Checking Control Unit Using Information redundancy*",University of al-Mergib, May 2008.
[6] J.M. Berger, "*A note on Error Detecting Codes for Asymmetric Channels*", Information and Control, vol4, 1961
[7]Anil K. Maini, "Digital Electronics*: Principles, Devices and Applications" ,* 2007 .
[8] Michael J. Quinn, " Parallel Computing Theory and practice ", Second Edition, international Edition 1994.

Impact Factor
7.54

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com