

3rd International Conference on Through-life Engineering Services

Fault tolerant quadded logic cell structure with built-in adaptive time redundancy

Philipp Schiefer*, Richard McWilliam, Alan Purvis

Durham University, South Road, Durham, DH1 3LE, United Kingdom

* Corresponding Philipp Schiefer. Tel.: +44 191 334 2418; fax: +44 191 334 2408. E-mail address: philipp.schiefer@durham.ac.uk

Abstract

This paper describes research carried out using a quadded logic cell (QLC) structure with the purpose of creating a fault tolerant strategy for stuck-at faults. In order to create the tolerant built-in behaviour, the basic logic elements must have resilience against transistor level stuck-at failures. To achieve this, we add fine-grained redundancy to the transistor structure of the individual logic gates. In our research NAND gates which are been used throughout the QLC design. Simulation data shows that the chosen enhanced NAND gate structure can cope with single random stuck-at fault and if not indicates it through a distinct current indication. The QLC design contains four individual logic units which can be configured to perform four different types of two input logic functions. The QLC contains an interconnection structure that links three logic units to form a logic structure with four inputs and one output. This fixed internal structure revolves clockwise in four steps in a “round-robin” time redundancy scheme to create a set number of results. Through a majority voting a combined overall output result gets generated. Individual comparison of each clock cycle result against the voted result reveals the cycle and logic unit combination in which the faulty result has been generated. In this case alteration of the individual logic unit configuration has been used to generate another set of results for pattern mapping to identify the single logic unit within the QLC. After identification a self-initiated logic unit replacement with a spare unit happens. An additional detection method based on power rail grading of the individual logic units is devised to enable built-in classification of the stuck-at fault occurring within the unit and subsequently to trigger self-repair. These features are intended to be self-coordinated without requiring outside influence, making this resulting design capable of autonomous self-healing under specific failure conditions.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of the Programme Chair of the 3rd International Through-life Engineering Conference

Keywords: Time Redundancy; Fault Tolerant; Self-healing; Stuck-at Faults

1. Introduction

Right from the start of the development of integrated circuits the area of fault handling within the manufacturing process and in-service application has been an important area. Every new circuit design requires increasingly sophisticated manufacturing test methodology due to ever-increasing chip complexity. Due to the number of individual elements within a single chip, the required test time and specialized equipment needed also increases with every development step and

significantly impacts the component price. In regards to test time, mass production limits the available time for production test during which the manufacturer must make a decision about the health of each chip and whether it is repairable in the event of manufacturing defects being present. Finding the right balance between test depth and complexity is ongoing research work in this area. Fault tolerance by design is one possible way to alleviate test time requirements. According to [1] the stuck-at fault at gate level and transistor level is among the most prevalent fault condition in device failure modes of

integrated circuits. The research leading this paper is based on the principle of creating a fault tolerant time redundant matrix structure, which can cope with faults by applying autonomous self-healing to maintain a functional logic structure.

Previous research work done in [2] outlined the idea of altering logic with the help of time redundancy instead of hardware redundancy with the focus on finding stuck-at faults. Our ongoing work is based on a matrix structure built out of equal logic units and that will be designed solely out of NAND gates. The structure of each matrix cell contains four concentric logic units arranged in a round-robin time dependant rotating schedule. The concept of round-robin is used in the area of data communication to create fault tolerant communication systems. Our research is focused on the utilization of a fixed number of units at a given time frame and alters this configuration for the next rotating schedule. With this we achieve a 3 out of 4 redundancy concept with one spare unit. This fixed structure operates on the principle that three out of four logic units can be tested for faults and decommission or replacement action carried out when a fault has been detected. In our design the overall matrix structure offers spare logic units between each matrix cell for replacing neighboring faulty logic. The whole cell is designed out of stuck-at fault resilient (SAFR) NAND gates, whose design has been altered to cope with single stuck-at high (SAH) or stuck-at low (SAL) faults at transistor level. For the case of a stuck-at fault which cannot be fixed by the fault tolerant NAND gate, a clear indication of this will trigger the self-healing by directly replacing the faulty logic unit.

2. Time dependent round-robin logic structure

The most frequently used fault tolerant concept in electrical systems is the approach of triple modular redundancy [3, 4]. This concept goes back on the seminal ideas of Von Neumann on n-type redundancy [5], in which a number three defines the minimal system requirement for a fault tolerant system. The hardware requirement for this system comprises a 200% overhead not including the voter. As a result the concept of time redundancy has been developed in which the same hardware is reused three times to generate three individual results created at different times. Through this concept a single event upset (SEU) that affects one result can be detected and eliminated with the help of a majority voting of the results [2, 4]. For non-permanent faults this concept has been proven within different applications [6]. In case of permanent hardware faults, the used hardware has to be altered for creating three results out of non-identical hardware otherwise three faulty results would be generated and passed as a majority result. Our research work focus on the idea of time redundancy for creating a number of results this creates additional information for fault detection. This concept varies to other redundancy concepts in this way that it does not use N-type parallel hardware or the same hardware gets used several times. Instead, our concept is built around the idea of hardware alteration per individual time cycle where the hardware structure is a cluster structure, or quadded

logic cell (QLC), made out of four individual equal logic units similar to the approach of [7] where a set of tiles was used. This tile structure builds the foundation of hardware reconfiguration on demand and is used in different concepts to restore logic functionality after a fault has been detected. The block diagram of the QLC is presented in Figure 1a. The approach utilizes three tiles to create the required logic functionality and a spare, which can be reconfigured to replace a faulty tile. This happens on a static approach. Our research introduces a round-robin scheme wherein one round-robin interval is made out of four reconfigurations of this cluster element and the associated configuration structure of tile elements in accordance to round-robin clock is presented in Figure 1b. A reconfiguration assembles three of the logic elements by replacing one of the previous units with the spare tile logic unit cell, hence creating a new configuration. By cycling through four different configurations four results are created via different combinations of hardware. For fault-free conditions each result should be identical.

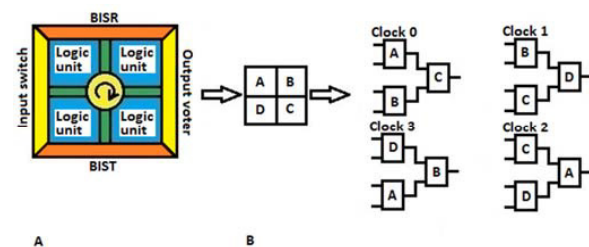


Fig. 1 (a) Block diagram of QLC, (b) configuration of logic units in conjunction to round-robin clock

This concept will be extended in the future by incorporating both functional built-in self-repair (BISR) and built-in self-test (BIST) as seen in the QLC block diagram presented in Figure 1a.

3. Functionality of logic unit

The design of the logic unit of the QLC is based on the ideas from [8, 9] where a basic reconfigurable block was proposed. This concept is built around the idea that a switching matrix between inputs and output is introduced in order to selectively switch required logic gate into the circuit. With the help of the switching matrix, a single functional gate is selected out of a number of possible gates. With this concept the functionality of the logic gate between fixed pins can be guaranteed with appropriate switching. Our research concept adds the additional feature of selecting gates of different logic function, hence offering programmable functionality in addition to redundancy. The resulting logic unit contains the four elemental logic functions: AND, NAND, OR and NOR (Figure 2).

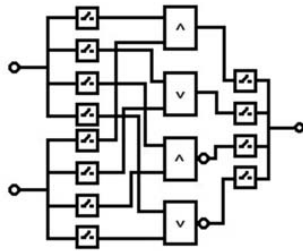


Fig. 2: Block diagram of logic unit

4. Fault pattern matching in QLC

The time redundant round-robin approach produces four individual results. The best case scenario, each result is produced through the use of different logic functionality performed with altered logic gates. The round-robin method ensures that for each clock cycle a 66% logic unit commonality is retained. Hence, by using different logic functions in each logic unit, a faulty gate within a single logic unit will only affect one result at one round-robin clock cycle. Simple pattern matching can be employed to locate the faulty logic unit. Alternatively fault tracing back to a single logic unit is still possible by using analytical methods. To demonstrate the effect of a single gate fault within a logic unit and the technique of pattern matching, an XOR gate is created within a QLC. The XOR logic design is presented in Figure 3. In Figure 4a the mapping of the different round-robin intervals are defined and Figure 4b the trust table of the XOR logic function for a certain input stimulus is shown along with the result. For this example, the logic function within logic unit C (AND function) has a stuck-at low output fault. This is reflected in the output data. After the first clock cycle a majority voter is formed, the QLC identifies the presence of a fault within clock0 cycle and alters configuration by switching the first two configuration information bits of the logic units. Through this a variation of the logic function within 66% of the logic unit creates functional diversity, which can be used for pattern matching. The round-robin interval also produces a fault at the next clock0 cycle. By superimposing this information common attribute is the AND configuration. So in this example the AND gate with the stuck-at low output fault has been identified and can now be disabled or replaced by a spare logic unit.

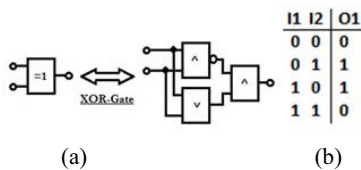
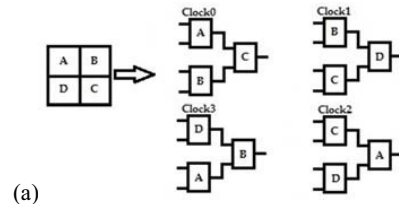


Fig. 3: (a) XOR gate and simulation within QLC logic units; (b) trust table of XOR function



	Logic unit				Data		
	A	B	C	D	I1	I2	O1
Clock0	NAND	OR	AND		0	1	0
Clock1		NAND	OR	AND	0	1	1
Clock2	AND		NAND	OR	0	1	1
Clock3	OR	AND		NAND	0	1	1
Clock0	OR	NAND	AND		0	1	0
Clock1		OR	NAND	AND	0	1	1
Clock2	AND		OR	NAND	0	1	1
Clock3	NAND	AND		OR	0	1	1

(b)

Fig. 4: (a) QLC logic unit configuration for logic units in accordance to round-robin interval; (b) Associated trust table with faulty logic unit for mapped XOR function

5. Self-healing of faulty logic unit of QLC

The QLC is embedded into a matrix structure in which two spare QLS logic units have been located to replace a faulty unit in close proximity. Figure 5 shows a block diagram of a single QLC of a matrix with spare logic units. This shared resource may only be utilized by one of the adjacent QLCs. The integrated mechanism for replacing a faulty logic unit in close proximity is done by additional control logic located between these two spare logic units. The functionality of this control circuit is designed in such a way that all control and data lines are switched over to the spare unit. Once allocated, a spare QLC is locked to prevent its use by the opposite neighbor via a reserved signal.

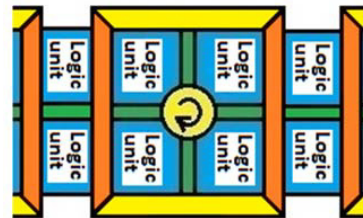


Fig. 5: Single QLC with neighboring spare logic units

6. Fault tolerant NAND gate

A common fault condition in any logic structure is the stuck-at fault [1]. They are able to cause errors in the output, or else can lay dormant in some cases such the output is still valid for all input combinations. Only through a certain sequence of input stimulus within the manufacturing test will the presence of faults be revealed. Extended manufacturing testing is expensive and requires deep knowledge of the hardware and system functionality to cover all possible fault

condition in regards to stuck-at faults. Our research is focused on creating a logic gate structure that is robust against single stuck-at faults and in the case of not been able to mask a certain faults, issues a defined built-in response signal to indicate the incidences of the fault.

The chosen gate type for this work is the NAND gate, which is an extensively in logic circuits. We plan to build the entire QLC logic design in NAND logic at a later stage. Here we perform a comparative assessment of the stuck-at fault resilience for the standard (non-redundant) and enhanced NAND gate designs. The enhanced gate type can be described in terms of redundancy at the transistor level as seen in Figure 6b. Similar research has replaced a single transistor with dual transistor redundancy [10-13] or quadded transistor structure [14-16]. Figure 6a shows a normal NAND gate circuit in comparison to the dual transistor redundant NAND gate presented in Figure 6b.

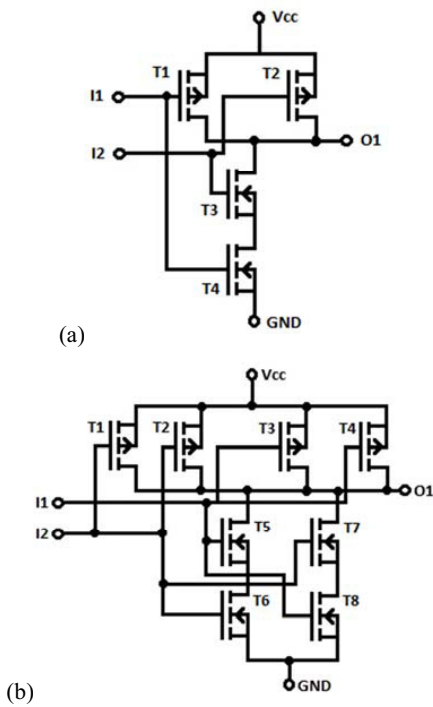


Fig. 6: (a) Normal NAND gate; (b) SAFR NAND gate

The detailed response of the enhanced NAND gate in the event of single stuck-at high or low faults can be seen in Table 1. The Table 1a defines the input sequence applied to the two inputs combined as one number and the Table 1b identifies the affected transistor and the associated condition (H=stuck-at High (SAH); L=stuck-t Low (SAL)) [17]. Table 1c shows the logic results of the simulation in accords to the stimulus (presented in Table 1a). The data shows the occurrence of undefined states (denoted by “x” in the table) that are present the output for certain cases. The overall resulting fault rate is expressed as 25% of all the possible cases resulted in an undefined output status. Table 1d

enumerates the current measurement indicator into the NAND gate circuit results. In this case the value of the current is not relevant and only a deviation from the normal short switching current in regards to a constant current flow is relevant on indicated as a “c” in the table. The data indicates that current flow conditions occur only in certain invalid output states.

I1	I2	I3	I4	T1	T2	T3	T4	O1	O2	O3	O4	C1	C2	C3	C4
00	01	10	11	H				1	1	1	x	0	0	0	c
00	01	10	11		H			1	1	1	x	0	0	0	c
00	01	10	11			H		1	x	1	0	0	c	0	0
00	01	10	11				H	1	1	x	0	0	0	c	0
00	01	10	11	L				1	1	x	0	0	0	0	0
00	01	10	11		L			1	x	1	0	0	0	0	0
00	01	10	11			L		1	1	1	x	0	0	0	0
00	01	10	11				L	1	1	1	x	0	0	0	0

(a) (b) (c) (d)

x = undefined c = current

Table 1: Simulation data of SAH (H) and SAL (L) for a single NAND gate

The stuck-at simulation results of the SAFR NAND in regards to SAH and SAL at individual transistors are presented in Table 2. The input stimulus and the selection of the effected transistor within the SAFR NAND gate are of the same structure as the normal NAND gate presented in Table 1. In conjunction to the normal NAND gate, the SAFR NAND gate only shows four undefined output results within the Table 1c in accordance to the stuck-at fault simulation. This is a fault rate therefore falls to 8.3% in comparison to the 25% for the normal NAND gate. Constant current flow into the SAFR NAND gate is again enumerated in Table 2d where the simulation result data indicates that for every undefined output an associated current flow is present. In contrast, the normal NAND gate shows no clear link between an undefined output and a significant constant current flow during a presents of a stuck-at fault at a certain input stimulus [18]. Furthermore, there are eight cases in the SAFR design where current flow occurs: in four of these cases the output exhibits significant current flow and invalid output; in the remaining four cases there is current flow but the gate still delivers a valid result. These last four cases represent a condition where the gate still delivers correct results and where an intervention would not be essential. This distinction, while seemingly useful, would require additional (and perhaps unnecessary) hardware overhead that would be a contradiction against the desire to create efficient intrinsically fault tolerant hardware.

Input Seq.				Fault Location Seq.								Output Seq.				Current Seq.			
S1	S2	S3	S4	T1	T2	T3	T4	T5	T6	T7	T8	O1	O2	O3	O4	C1	C2	C3	C4
00	01	10	11	SAH	---	---	---	---	---	---	---	1	1	1	0	0	0	0	C
00	01	10	11	---	SAH	---	---	---	---	---	---	1	1	1	0	0	0	0	C
00	01	10	11	---	---	SAH	---	---	---	---	---	1	1	1	0	0	0	0	C
00	01	10	11	---	---	---	SAH	---	---	---	---	1	1	1	0	0	0	0	C
00	01	10	11	---	---	---	---	SAH	---	---	---	1	x	1	0	0	C	0	0
00	01	10	11	---	---	---	---	---	SAH	---	---	1	x	1	0	0	C	0	0
00	01	10	11	---	---	---	---	---	---	SAH	---	1	1	x	0	0	0	C	0
00	01	10	11	---	---	---	---	---	---	---	SAH	1	1	x	0	0	0	C	0
00	01	10	11	SAL	---	---	---	---	---	---	---	1	1	1	0	0	0	0	0
00	01	10	11	---	SAL	---	---	---	---	---	---	1	1	1	0	0	0	0	0
00	01	10	11	---	---	SAL	---	---	---	---	---	1	1	1	0	0	0	0	0
00	01	10	11	---	---	---	SAL	---	---	---	---	1	1	1	0	0	0	0	0
00	01	10	11	---	---	---	---	SAL	---	---	---	1	1	1	0	0	0	0	0
00	01	10	11	---	---	---	---	---	SAL	---	---	1	1	1	0	0	0	0	0
00	01	10	11	---	---	---	---	---	---	SAL	---	1	1	1	0	0	0	0	0
00	01	10	11	---	---	---	---	---	---	---	SAL	1	1	1	0	0	0	0	0

x = undefined C = current

(a) (b) (c) (d)

Table 2: Simulation data of SAH and SAL for the SAFR design NAND gate

A PSPICE timing diagram of a typical SAFR NAND gate in case of no fault (timing from 0 to 2ms) and a SAH fault (timing from 2ms to 4ms) is presented in Figure 7. The SAH fault is stimulated at transistor T1 of the gate and for this case a valid output result (OUT01) is been generated. This particular stuck-at fault condition manifests only during one particular input stimulus which is also indicated through the current flag generated via the PSpice CURRFLAG (see timing interval 3.5-4ms). This flag could be used to trigger a reconfiguration in regards of switching in a spare logic unit.

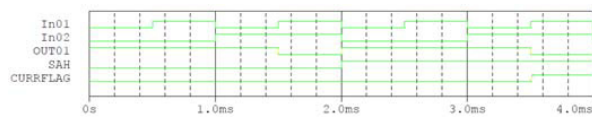


Fig. 7: Timing diagram of SAFR NAND gate in case of SAH fault (0ms to 2ms faultfree; 2ms to 4ms SAH fault at T1)

7. Conclusion

We have presented on-going research in fault-tolerance, detection and repair using the QLC design approach. This strategy depends upon two built-in fault tolerant features. The temporal-based round-robin approach enables detection of a fault occluding with the logic gate with the help of intersection remapping of successive round-robin clock cycles, followed by rearrangement of the logic unit configuration data. At the fine-grained level, the SAFR NAND gate design with built-in current based fault detection is able to trigger an additional flag that could be used to initiate replacement of logic unit by a spare unit. This operation could occur within a single round-robin clock cycle. The next steps for our research are to generate a matrix cluster including spare logic units and to investigate the behavior of this matrix in regards of stuck-at faults for different QLC matrix elements. Further investigation should evaluate (a) the performance and fault tolerant behavior, (b) the effect of fault occurring within the voter logic present within each QLC and

(c) strategies for synchronizing all individual QLCs to a global clock cycle.

Acknowledgements

This work is supported by the EPSRC Centre for Innovated Manufacturing in Through-life Engineering Services EP/I033246/1.

References

- [1] P. O'Connor and A. Kleyner, *Practical reliability engineering*: John Wiley & Sons, 2011.
- [2] D. A. Reynolds and G. Metzke, "Fault Detection Capabilities of Alternating Logic," *Computers, IEEE Transactions on*, vol. C-27, pp. 1093-1098, 1978.
- [3] M. Niknahad, O. Sander, and J. Becker, "Fine grain fault tolerance- A key to high reliability for FPGAs in space," in *Aerospace Conference, 2012 IEEE*, 2012, pp. 1-10.
- [4] E. Dubrova, "Fault tolerant design: An introduction," *Department of Microelectronics and Information Technology, Royal Institute of Technology, Stockholm, Sweden*, 2008.
- [5] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata studies*, vol. 34, pp. 43-98, 1956.
- [6] K. G. Shin and K. Hagbae, "A time redundancy approach to TMR failures using fault-state likelihoods," *Computers, IEEE Transactions on*, vol. 43, pp. 1151-1162, 1994.
- [7] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "Low overhead fault-tolerant FPGA systems," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 6, pp. 212-221, 1998.
- [8] T. Koal, D. Scheit, and H. T. Vierhaus, "A Concept for Logic Self Repair," in *Digital System Design, Architectures, Methods and Tools, 2009. DSD '09. 12th Euromicro Conference on*, 2009, pp. 621-624.
- [9] T. Koal, D. Scheit, and H. T. Vierhaus, "A scheme of logic self repair including local interconnects," in *Design and Diagnostics of Electronic Circuits & Systems, 2009. DDEC'S '09. 12th International Symposium on*, 2009, pp. 8-11.
- [10] T. Koal, D. Scheit, and H. T. Vierhaus, "Zuverlässige Elektronik-Systeme aus unzuverlässigen Komponenten," pp. 53-58, 2008.
- [11] T. Koal, D. Scheit, Scho, x, M. Izel, and H. T. Vierhaus, "On the Feasibility of Built-In Self Repair for Logic Circuits," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2011 IEEE International Symposium on*, 2011, pp. 316-324.
- [12] P. Beckett, "A Low-Power Reconfigurable Logic Array Based on Double-Gate Transistors," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, pp. 115-123, 2008.
- [13] R. Kothe, H. T. Vierhaus, T. Coym, W. Vermeiren, and B. Straube, "Embedded Self Repair by Transistor and Gate Level Reconfiguration," in *Design and Diagnostics of Electronic Circuits and systems, 2006 IEEE*, 2006, pp. 208-213.
- [14] C. Bolchini, G. Buonanno, D. Sciuto, and R. Stefanelli, "Innovative design of CMOS fault tolerant structures," in *Wafer Scale Integration, 1995. Proceedings., Seventh Annual IEEE International Conference on*, 1995, pp. 267-276.
- [15] A. H. El-Maleh, B. M. Al-Hashimi, A. Melouki, and F. Khan, "Defect-tolerant n²-transistor structure for reliable nanoelectronic designs," *Computers & Digital Techniques, IET*, vol. 3, pp. 570-580, 2009.
- [16] A. H. El-Maleh, A. Al-Yamani, and B. M. Al-Hashimi, "Transistor-Level Defect Tolerant Digital System Design at the Nanoscale," *Research Proposal Submitted to Internal Track Research Grant Programs*, 2007.
- [17] E. J. McCluskey and T. Chao-Wen, "Stuck-fault tests vs. actual defects," in *Test Conference, 2000. Proceedings. International*, 2000, pp. 336-342.
- [18] P. C. Maxwell, R. C. Aitken, K. R. Kollitz, and A. C. Brown, "IDDQ and AC scan: the war against unmodelled defects," in *Test Conference, Proceedings., International*, 1996, pp. 250-258.