

# A Self-Checking Cell Logic Block for Fault Tolerant FPGAs

S. Pontarelli, G.C. Cardarilli, A. Leandri, M. Ottavi, M. Re, A. Salsano  
{ottavi,pontarelli,salsano}@ing.uniroma2.it  
{marco.re, g.cardarilli}@ieee.org  
Alexleandri@Katamail.com

Department of Electronic Engineering University of Rome "Tor Vergata", Italy

Via di Tor Vergata 110  
00133-Rome-ITALY

## ABSTRACT

*This paper proposes a self-checking Cell Logic Block (CLB) that can be used as building block for on-line testable FPGAs. The proposed cell consists, basically, of a 4 input Look-Up-Table (LUT) and a D Flip-Flop. The CLB is designed using pass-transistor-based multiplexers, either to select the output of the 4-input LUT, or to select signals from other CLBs. The proposed CLB architecture is characterized by a simple circuit to detect incorrect logic voltage levels due to stuck-close and stuck-open faults and by a sensor to test anomalous dissipated currents. In this way, the proposed CLB allows on-line detection of any single transistor fault.*

## 1. INTRODUCTION

Currently FPGAs are widely used for rapid prototyping and for the realization of low cost complex systems. Moreover, their reprogrammability feature can be extremely useful in case of faults. The modular structure of the FPGA allows reprogramming the device replacing the faulty block with a spare one, once a fault is detected.

This feature assures a high degree of fault tolerance, also in extremely hostile environments such as space or radioactive ones.

Commercial FPGAs have facilities that allow fully testing before the programming phase. Implementation of the off-line testing during the normal working is possible as well using these facilities.

However, due to the pervasive use of electronic devices in critical applications, there is a growing interest for systems with on-line testing capabilities. On-line test is the only way for detecting transient faults, as well as the dynamic or the intermittent ones [12]

Permanent and transient faults can be detected on-line only using complex testing techniques ([1],[2],[3]). In principle, it is possible to detect the presence of a fault during normal operation monitoring suitable coded

outputs. These methods detect a fault in a group of CLBs programmed to realize a self-checking circuit [11], but a diagnosis routine must be used to exactly localise the faulty block.

The main drawbacks of these techniques are the necessity to implement complex self-checking circuits, and the downtime caused by the diagnosis routine.

An on-line testable FPGA formed by self-checking CLB can be used to avoid these problems. In literature self-checking CLB based on two-rail multiplexer have been proposed ([4],[5],[6]).

In this paper we propose a new self-checking CLB based on the detection of faulty intermediate voltages or anomalous currents.

The paper is organized as follows. In Section 2 the architecture of the CLB is described, while in Section 3 the differences between normal and faulty behaviors are analyzed in detail. In Section 4 the performance in terms of area overhead and speed are presented and compared, while the conclusions are drawn in Section 5.

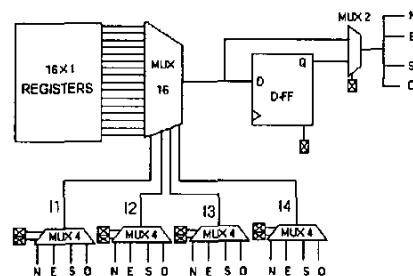


Fig. 1: Proposed CLB Architecture

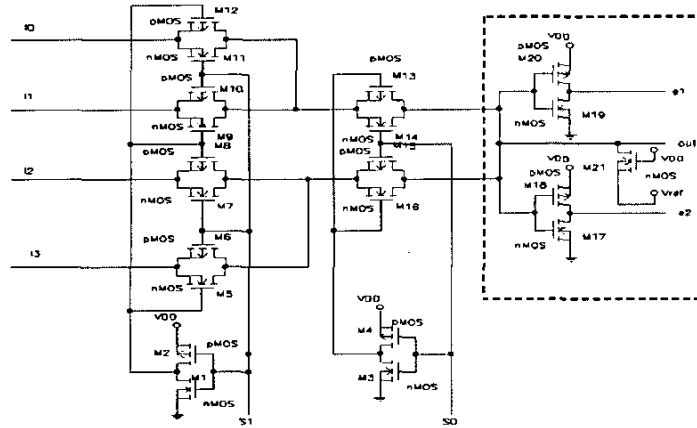


Fig. 2: Pass-transistor scheme of a 4-input multiplexer

## 2. SELF-CHECKING CELL LOGIC BLOCK ARCHITECTURE

The proposed CLB architecture is shown in Fig.1: The CLB is composed by a 16 X 1 SRAM storing the programming bits of the LUT and by a 16 X 1 multiplexer to select the output implementing the function  $F(I_1, I_2, I_3, I_4)$ . The output of MUX16 (combinatorial function) can be directly connected to the interconnect network, or can be stored in the D-Flip-Flop. The 2 input multiplexer MUX 2 selects which output is given as the CLB output. The 4-input multiplexers MUX4 are used to select the interconnections between the various CLBs. The programming bits used as select bits of the multiplexer are stored in another SRAM. Since routing is not in the scope of this paper, in Fig.1 we have just introduced the minimum resources for covering this aspect. Modifying the number of the multiplexers or the number of select bits, major routing resources can be supported. The resources of this CLB are of two types: the memory elements (D-FF and SRAM) and multiplexers. In this work we propose two different modalities to check the presence of faults of those components. For the memory elements we use a Built In Current Sensor (BICS) similar to that proposed in [7]. The Gnd and the Vcc nodes of the memory elements are connected to the nodes of the BICS sensor to detect anomalous dissipated current. Although various BICS sensors have been proposed in the literature (e.g. [8], [9]), we use the conservative scheme proposed in [7] because it uses a limited number of transistors. For the multiplexers in the CLB we use the scheme described in paper [10]. Fig. 2 shows how to check a 4 input multiplexer by adding two inverters (M17/M18 and M19/M20) to its output (out). The transistors used in the inverter are characterized by different aspect ratios to achieve different voltage

thresholds. In the following, we call  $V_{T1}$  and  $V_{T2}$  the two different thresholds. If the out node is at an intermediate voltage the outputs  $e1$  and  $e2$  present different voltage levels values, otherwise  $e1$  and  $e2$  are at the same level.

$V_{out}$	$e1$	$e2$
$GND < V_{out} < V_{T1}$	$V_{DD}$	$V_{DD}$
$V_{T1} < V_{out} < V_{T2}$	$V_{DD}$	$GND$
$V_{T2} < V_{out} < V_{DD}$	$GND$	$GND$

Tab. 1: Output of inverters M17/M18 and M19/M20

Table 1 shows the levels of the  $e1$  and  $e2$  outputs with respect to the out voltage. In the presence of faults, this simple circuit produces intermediate voltages allowing fault detection [10]. We extend this circuit using an additional transistor (nMOS M21 in Fig.2) with  $V_G = V_{DD}$  and  $V_S = V_{ref}$  being  $V_{T1} < V_{ref} < V_{T2}$ . If a stuck open fault occurs, this transistor pushes the output of the multiplexer toward  $V_{ref}$ . Finally, the self-checking cell uses an error controller to evaluate the presence of faults. The input of the error controller gets the outputs  $e1$  and  $e2$  of the multiplexer, and the  $e1$  and  $e2$  of the BICS, while the output is two-rail coded. In absence of faults, the outputs  $e1$  and  $e2$  can assume the values 11 or 00, so the input configuration to the error controller is characterized always by a even parity. If an error occurs the multiplexer output pair shall assume the values 01 or 10, so the input word presents an odd parity. A parity checker can be used to implement the error controller. In our implementation we used a xor tree to sum modulo 2 the  $e1$  inputs and another xor tree to sum modulo 2 the  $e2$  inputs. The outputs of the parity checker are 00 or 11, if no error occurs, 01 or 10 otherwise. It is easy to demonstrate that this implementation is totally self-checking.

### 3. SELF-CHECKING CELL LOGIC BLOCK BEHAVIOR

The behavior of the Self-Checking Cell in absence of faults is quite simple. The programming bits select the routing configurations and set the combinatorial or sequential behavior of the cell. The selected inputs I1-I4 allow to select an output of the 16 X 1 SRAM realizing the function  $F(I1,I2,I3,I4)$ . In the following, we describe the behavior in presence of faults.

The BICS sensor detects the faults in the memory elements (the D-FF and the 16 X 1 SRAM where the programming bits are stored). So an error signal is given to the error control circuit. The faults in the multiplexer can be divided in stuck-close and stuck-open transistor faults.

Let us suppose that a stuck-close fault occurs in a pass-transistor of the multiplexer. The fault gives erroneous outputs under the following conditions:

1. the input line that drives the faulty transistor is not selected
2. the faulty transistor form a path from its input line to the selected ones
3. the value of the two input line are different.

For example, if the M12 transistor is stuck-close the fault is activated when the line I1 is selected, and the value I0 and I1 are 1 and 0, respectively. In this case the voltage value at the output node is:

$$V_{out} \approx R_{M12} / (R_{M10} // R_{M9} + R_{M12}) V_{DD}$$

Where  $R_{Mn}$  is the resistance of the transistor named  $Mn$ . With suitable values of the transistor ratio  $W/L$  the output value is  $V_{T1} < V_{out} < V_{T2}$ , so the output  $e1$  and  $e2$  are different. In Fig. 3 the waveforms in the case of a stuck-close fault in a pass-transistor are shown. The transistor aspect ratio for the nMOS and the pMOS are  $W/L = 2.0\mu m / 0.25\mu m$  and  $4.5\mu m / 0.25\mu m$ , respectively.

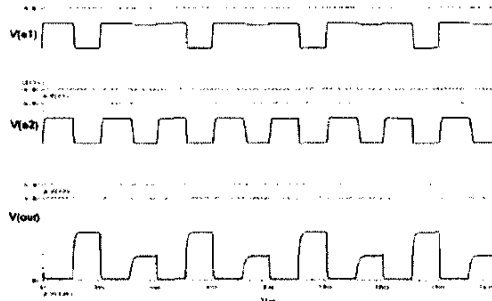


Fig. 3: Output waveforms with a stuck-close transistor

Now, let us suppose a stuck-open fault occurring in a pass-transistor of the multiplexer. The fault is activated when the input line that pushes the faulty transistor is selected. The M21 transistor drives the output node

toward the value  $V_{ref}$ . If  $V_{T1} < V_{ref} < V_{T2}$  the output  $e1$  and  $e2$  are different, so the fault is detected. In Fig. 4 simulation results are shown when  $V_{ref} = V_{DD}/2$ .

If a fault occurs in the inverters M1/M2 or M3/M4 the effects are similar to those described above in case of a stuck-close in a pass transistor. In fact, a fault in the inverter forces the output toward the same value of the input.

So, two transistors of the same kind (p or n) in the multiplexer are on in two different pass-transistors. In this case, if the inputs of the pass-transistors are different,  $V_{out}$  is set at an intermediate voltage value.

Finally, a fault in the inverters M17/M18 and M19/M20 is detected, because it forces the outputs  $e1$ ,  $e2$  toward the value 01 or 10.

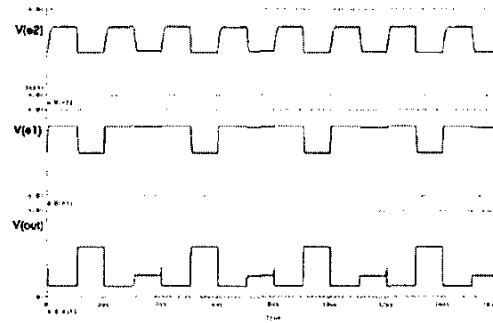


Fig. 4: Output waveforms with a stuck-open transistor

### 4. SIMULATION RESULTS

In this section the performance of our Self-Checking Cell with respect to a cell without fault tolerance capabilities is compared. The cell has been simulated using a  $0.25\mu m$  CMOS technology file. SPICE simulations were carried out to estimate the speed of the proposed solution. In Fig. 5 are shown the output  $V_{out}$  for the Self-Checking Cell (dashed line) and for a standard multiplexer with the same transistor aspect ratio.

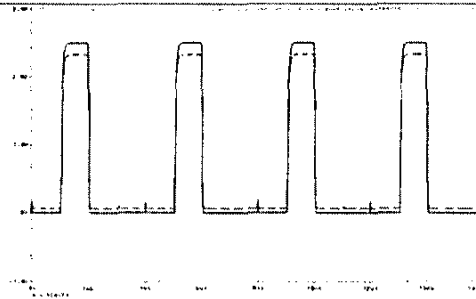


Fig. 5: Output waveforms for the Self-Checking Cell and the normal one

The transistor aspect ratio for the nMOS M21 is set to  $W/L = 0.9\mu\text{m}/0.25\mu\text{m}$ , obtaining a  $V_{\min} = 10\% V_{DD}$  and  $V_{\max} = 90\% V_{DD}$ . The value of  $V_{DD}$  used in the simulation is 2.5V. The simulation results show a good speed behavior of the modified multiplexer with respect to the normal one.

The area overhead is computed considering the number of transistors necessary for the realization of a normal cell and the proposed self-checking cell. We assume that the area of the CLB is proportional to the number of transistors; In fact the various kind of transistors used in the CLB have similar dimensions.

The needed resources in terms of memory elements are equal to twenty-six bits to configure the logic block (16 SRAM cell for the LUT, 10 SRAM cell to configure the mux and the D-FF and a D-FF) each one has been implemented by using a 6 transistors cell. The number of transistors for the input multiplexers depends on the number of inputs. It's easy to demonstrate that, if the number of inputs is  $2^n$ , the number of transistors is  $N = 2n + 2^{n+2} - 4$  for the normal multiplexer,  $N = 2n + 2^{n+2} + 1$ , for the Self Checking one.

The error controller can be realized by using two 7 inputs XOR trees. If we realize the XOR as shown in Fig. 6 the number of transistor for the error controller can be easily evaluated. In fact, we have 6 XOR for every XOR tree. The required 12 XOR can be realized by 48 transistors.

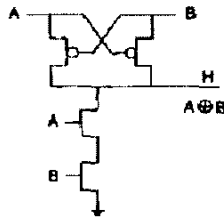


Fig. 6: XOR scheme for the two-rail parity checker

Consequently the total area overhead is 32%. The results are summarized in Tab. 2.

Element	Normal cell			SC Cell		
	#El	#Tr	Tot	#El	#Tr	Tot
2-input mux	1	6	6	1	11	11
4-input mux	4	16	64	4	21	84
16-input mux	1	68	68	1	73	73
Mem. element	27	6	162	27	6	162
BICS	-	-	-	1	18	18
Err. Controller	-	-	-	1	48	48
Tot.	300			396		

Tab. 2: Transistor count for the normal Cell and the Self-checking one

## 5. CONCLUSIONS

We have developed a logic cell with built-in self-checking features. This cell can be used as building block for on-line testable FPGAs. The major feature of the proposed cell is its capability to detect a single fault (stuck-at-0/1 or transistor stuck-open/close) during normal operations. A simple circuit for detection of stuck-open/close faults is proposed; simulation results confirm the effectiveness of this circuit in terms of speed and area overhead.

## 6. REFERENCES

- [1] J.H. Lach, H. Mangione-Smith, and M. Potkonjak, "Low Overhead fault tolerant FPGA systems", IEEE Trans. VLSI Systems, vol.6 no.2, June 1998, pp.212-221.
- [2] S. D'Angelo, C. Metra, G. Sechi, "Transient and Permanent Fault Diagnosis for FPGA-Based TMR Systems", Defect and Fault Tolerance in VLSI Systems, 1999, pp. 330-338
- [3] J.Emmert, C. Stroud, B.Skaggs; M. Abramovici, "Dynamic Fault Tolerance in FPGAs via Partial Reconfiguration", IEEE Symposium on Field-Programmable Custom Computing Machines, 2000.
- [4] P. K. Lala, A. Singh, "Logic Cell Design For On-Line Testable FPGAs", ASICs, 1999. AP-ASIC '99. The First IEEE Asia Pacific Conference, 1999 pp. 351-354
- [5] P. K. Lala, A. Singh, A. Walker, "A CMOS-based Logic Cell for the Implementation of Self-Checking FPGAs" IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 1999.
- [6] P. K. Lala, A. Walker, "An On-Line Reconfigurable FPGA Architecture", IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2000.
- [7] F.Vargas, M.Nicolaidis, "SEU-tolerant SRAM design based on current monitoring", Proceedings of 24th International Symposium on fault tolerant computing, pp. 106-115, June 1994.
- [8] T.Calin, F.Vargas, M.Nicolaidis, "Upset-tolerant CMOS SRAM using current monitoring: prototype and test experiments," IEEE International Test Conference, pp. 45-53, 1995.
- [9] J.Tang, K. Lee, B. Liu, "A practical current sensing technique for IDDQ testing", IEEE Transaction on VLSI, pp. 303-310, June 1995.
- [10] C. Metra M. Favalli P. Olivo B. Riccò, "CMOS Checkers with Testable Bridging and Transistor Stuck-on Faults", IEEE International Test Conference, 1992. pp. 948
- [11] G.C. Cardarilli, A. Malvoni, M. Ottavi, S. Pontarelli, M. Re, A. Salsano, "System-on-Chip Implementation and Fault-Coverage Estimation of a Fault-Tolerant State Machine", 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001) Orlando, USA, July, 2001.
- [12] M. Weyerer, G. Goldemund, "Testability of electronic circuit", Prentice Hall, 1992.