Improved dependability for dynamically reconfigurable hardware

Restoration of the reliability index via replication and error correction

J. M. Martins Ferreira [jmf@fe.up.pt]
FEUP / DEEC
Rua Roberto Frias
P-4200-465 Porto

HIBU
Frogsvei 41
N-3603 Kongsberg

Manuel G. Gericota [mgg@dee.isep.ipp.pt]
ISEP / DEE
Rua Dr. António Bernardino de Almeida
P-4200-072 Porto

[this presentation is available online at **http://www.fe.up.pt/~jmf/dak-2004.ppt**]

DAK-forum 2004

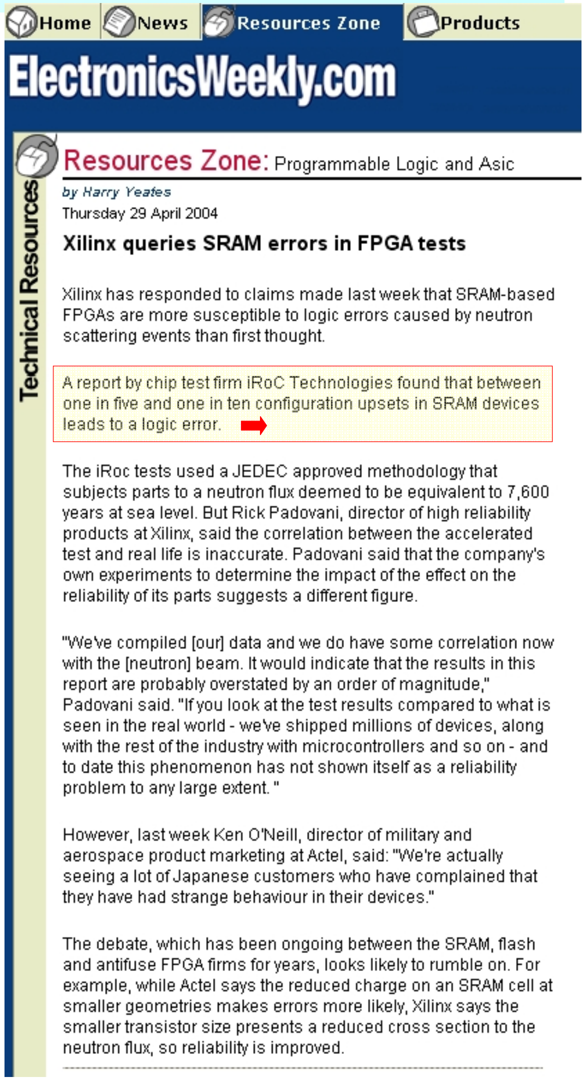
Trondheim, Norway
October 19-21 2004 (1 / 26)

Outline of the presentation

- Introduction and motivation
- Causes of failure
- Concurrent fault detection
- Fault detection latency and fault tolerance
- Fault masking and fault correction
- Research directions
- Conclusion

Introduction and motivation

- Dynamically reconfigurable FPGAs:
  - – Production tests cannot guarantee fault-free operation
  - – Application areas include mission-critical systems
  - – The cost / benefit of spatial redundancy is different from static implementations



Resources Zone: Programmable Logic and Asic

by Harry Yeates

Thursday 29 April 2004

Xilinx queries SRAM errors in FPGA tests

Xilinx has responded to claims made last week that SRAM-based FPGAs are more susceptible to logic errors caused by neutron scattering events than first thought.

A report by chip test firm iRoC Technologies found that between one in five and one in ten configuration upsets in SRAM devices leads to a logic error. ➡

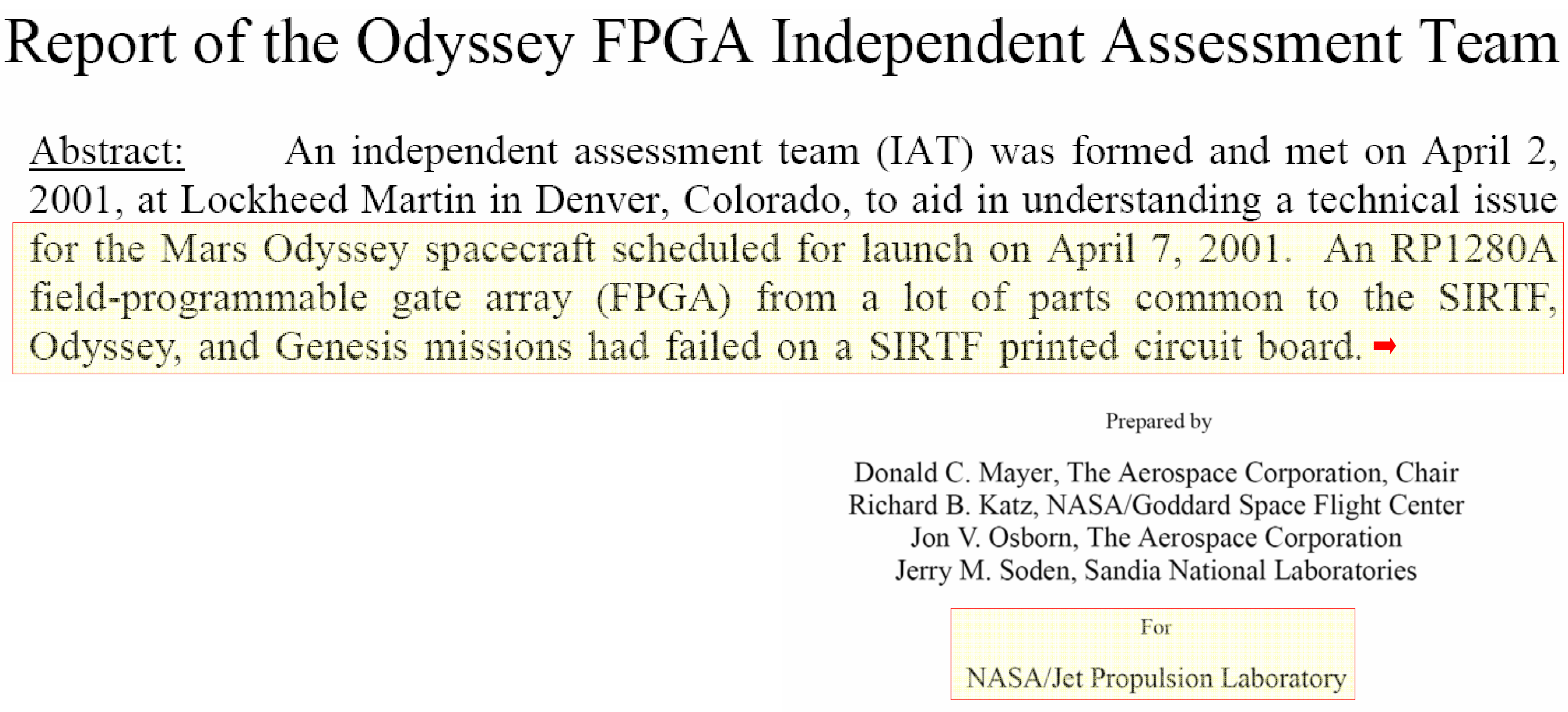
The iRoc tests used a JEDEC approved methodology that subjects parts to a neutron flux deemed to be equivalent to 7,600 years at sea level. But Rick Padovani, director of high reliability products at Xilinx, said the correlation between the accelerated test and real life is inaccurate. Padovani said that the company's own experiments to determine the impact of the effect on the reliability of its parts suggests a different figure.

"We've compiled [our] data and we do have some correlation now with the [neutron] beam. It would indicate that the results in this report are probably overstated by an order of magnitude," Padovani said. "If you look at the test results compared to what is seen in the real world - we've shipped millions of devices, along with the rest of the industry with microcontrollers and so on - and to date this phenomenon has not shown itself as a reliability problem to any large extent. "

However, last week Ken O'Neill, director of military and aerospace product marketing at Actel, said: "We're actually seeing a lot of Japanese customers who have complained that they have had strange behaviour in their devices."

The debate, which has been ongoing between the SRAM, flash and antifuse FPGA firms for years, looks likely to rumble on. For example, while Actel says the reduced charge on an SRAM cell at smaller geometries makes errors more likely, Xilinx says the smaller transistor size presents a reduced cross section to the neutron flux, so reliability is improved.

Introduction and motivation



Report of the Odyssey FPGA Independent Assessment Team

Abstract: An independent assessment team (IAT) was formed and met on April 2, 2001, at Lockheed Martin in Denver, Colorado, to aid in understanding a technical issue for the Mars Odyssey spacecraft scheduled for launch on April 7, 2001. An RP1280A field-programmable gate array (FPGA) from a lot of parts common to the SIRTF, Odyssey, and Genesis missions had failed on a SIRTF printed circuit board. ➡

Prepared by

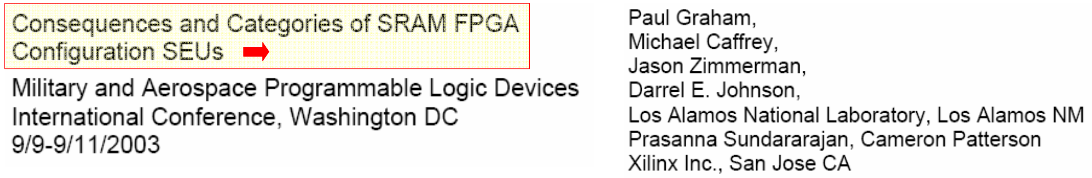
Donald C. Mayer, The Aerospace Corporation, Chair
Richard B. Katz, NASA/Goddard Space Flight Center
Jon V. Osborn, The Aerospace Corporation
Jerry M. Soden, Sandia National Laboratories

For

NASA/Jet Propulsion Laboratory

Causes of failure

- Post-production failure modes may be permanent or temporary — examples:
  - Electromigration phenomena may lead to permanent physical damage
  - Single-event upsets (SEUs) may cause permanent malfunction if not mitigated (modification of SRAM contents changes design and data information)
  - See



Consequences and Categories of SRAM FPGA Configuration SEUs ➡

Military and Aerospace Programmable Logic Devices International Conference, Washington DC
9/9-9/11/2003

Paul Graham,
Michael Caffrey,
Jason Zimmerman,
Darrel E. Johnson,
Los Alamos National Laboratory, Los Alamos NM
Prasanna Sundararajan, Cameron Patterson
Xilinx Inc., San Jose CA

Fault detection

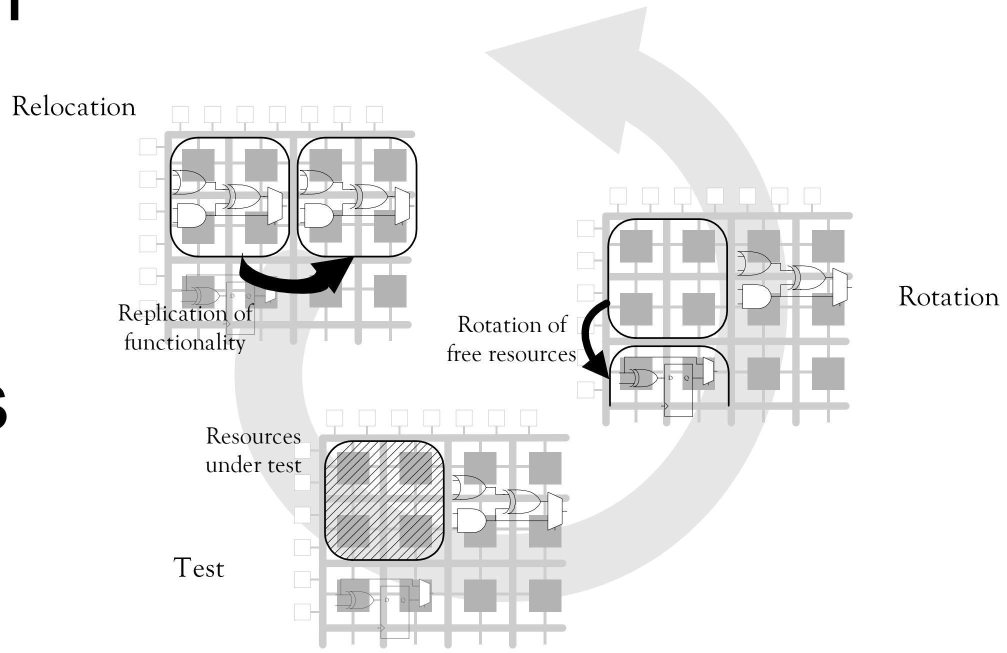
- Dynamic reconfiguration enables concurrent fault detection
  - Modifications in the configuration memory may be tested by *scrubbing*
  - Structural faults that emerge on the field may be detected by *release-to-test* strategies

Fault detection: Scrubbing

- Errors in the on-chip configuration memory may be detected by partial readback (and corrected by partial reconfiguration)
- Scrubbing prevents “design” errors that might lead to functional failure
- Data stored in flip-flop registers is not writable via the configuration memory, so scrubbing does not correct “data” errors

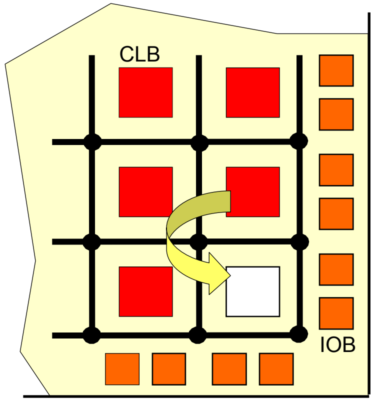
Fault detection: Release-to-test

- The basic idea underlying release-to-test strategies consists of *non-intrusively* replicating a given functional block in another area, and to make the original resources available for test

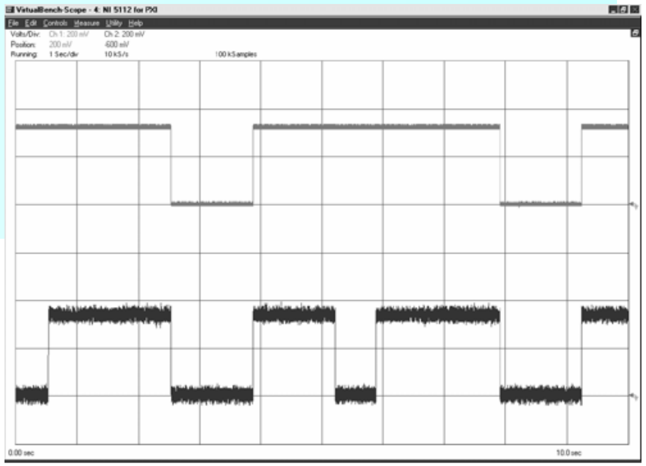


Replication of active resources

- Concurrent fault detection based on release-to-test approaches must provide functional *and* state replication
- Replication at CLB-level
  - – Facilitates state transfer and requires a minimal amount of spare resources
  - – The relative position of the replicated CLB and its replica has an impact on propagation delay



CLB replication

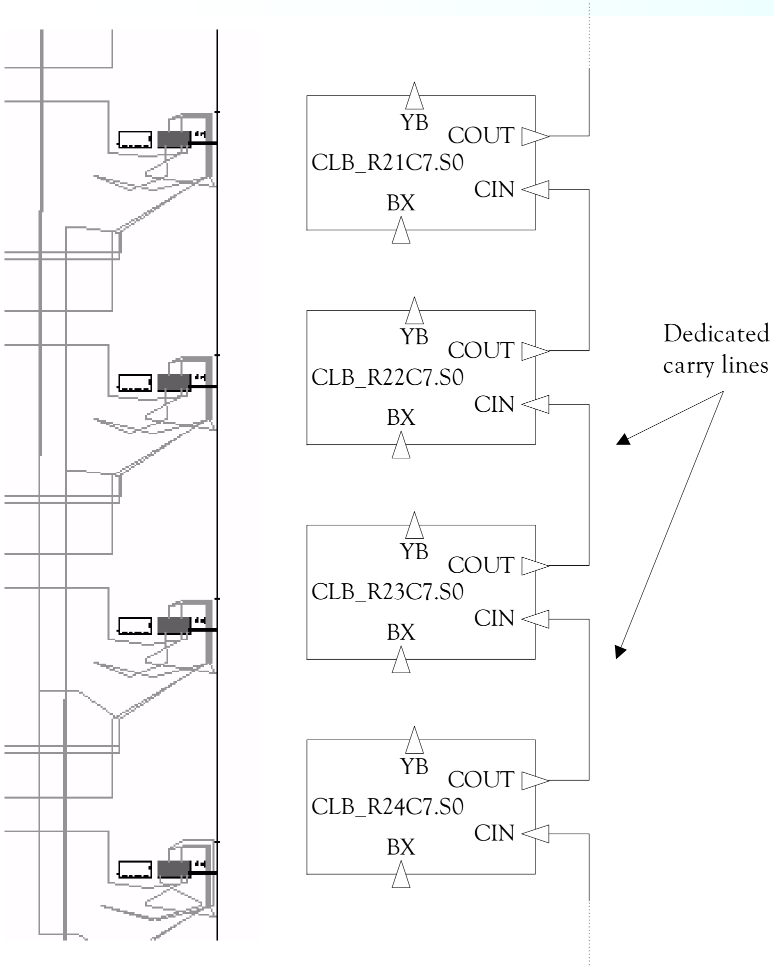


Volts/Div: Ch 1: 200 mV Ch 2: 200 mV
Position: 200 mV -600 mV
Running: 1 Sec/div 10 kS/s 100 kSamples

(paralleling CLB outputs is not a problem)

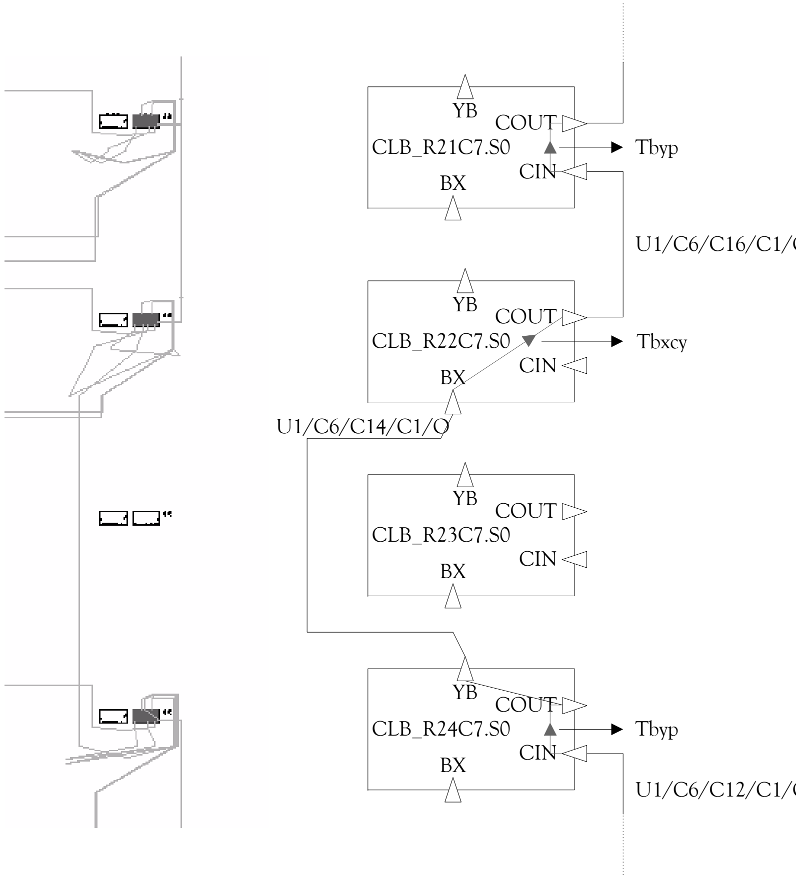
- Replicating the functional configuration of a CLB is done with minimal overhead
- In free-running clock circuits, placing the inputs of the two CLBs in parallel ensures common state acquisition
- Gated-clock circuits need an auxiliary block to provide state transfer

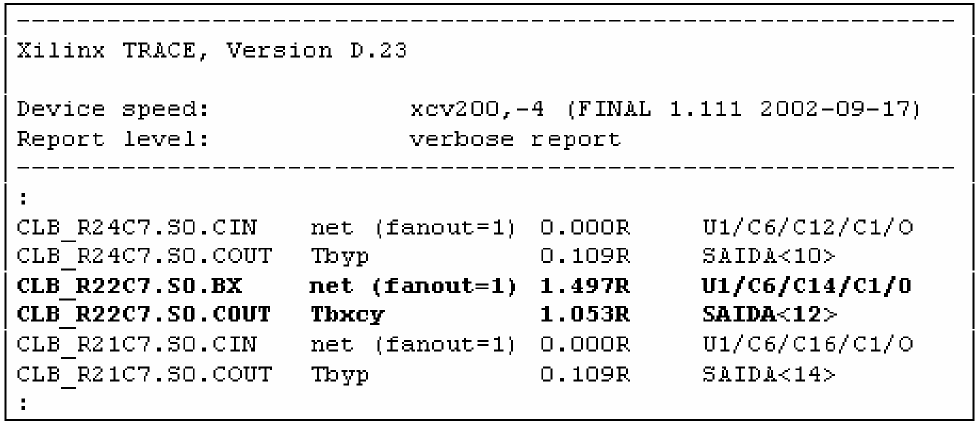
Example: Replicate and release-to-test in a 24-bit binary counter

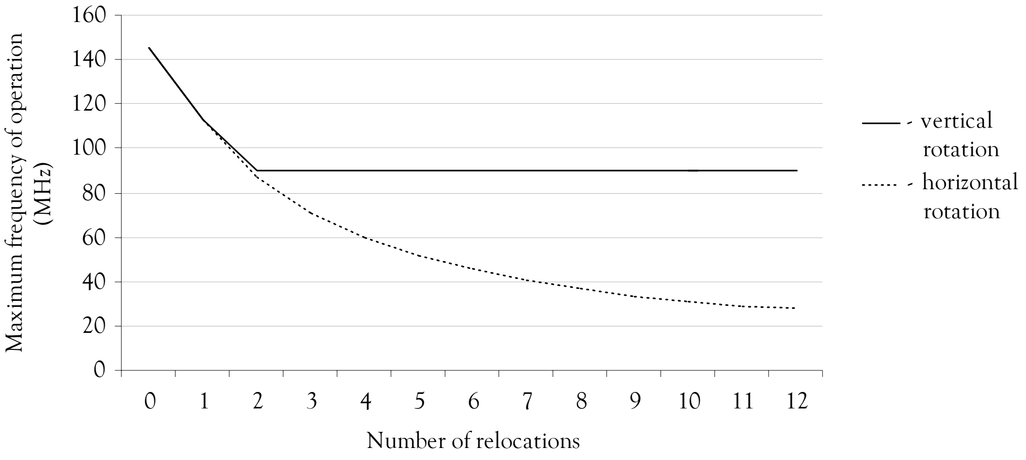


```
library IEEE;
use IEEE.std_logic_1164.all;

entity contador is
  port (
    CLK: in STD_LOGIC;
    Saida: out INTEGER range 0 to 16777215
  );
end contador;

architecture contador_arch of contador is
signal A: INTEGER range 0 to 16777215;
begin
  process(CLK, A)
  begin
    if (CLK='1' and CLK'event) then
      A <= A+1;
    end if;
    Saida<=A;
  end process;
end contador_arch;
```

Example: Replicate and release-to-test in a 24-bit binary counter




```
------------------------------------------------------------------
Xilinx TRACE, Version D.23

Device speed:              xcv200,-4 (FINAL 1.111 2002-09-17)
Report level:              verbose report
------------------------------------------------------------------
:
CLB_R24C7.S0.CIN    net (fanout=1) 0.000R     U1/C6/C12/C1/O
CLB_R24C7.S0.COUT   Tbyp           0.109R     SAIDA<10>
CLB_R22C7.S0.BX     net (fanout=1) 1.497R     U1/C6/C14/C1/O
CLB_R22C7.S0.COUT   Tbxcy          1.053R     SAIDA<12>
CLB_R21C7.S0.CIN    net (fanout=1) 0.000R     U1/C6/C16/C1/O
CLB_R21C7.S0.COUT   Tbyp           0.109R     SAIDA<14>
:
```



Validation: ITC'99 benchmarks

Circuit			Logic		Carry logic	
Reference	Primary inputs	Primary outputs	Number of gates	Number of flip-flops	Lines	Segments
B01	2+2	2	47	5	0	0
B02	1+2	1	29	4	0	0
B03	4+2	4	150	30	0	0
B04	11+2	8	606	66	4	14
B05	1+2	36	977	34	4	16
B06	2+2	6	61	9	0	0
B07	1+2	8	422	49	2	6
B08	9+2	4	168	21	0	0
B09	1+2	1	160	28	0	0
B10	11+2	6	190	17	0	0
B11	7+2	6	484	31	1	4
B12	5+2	6	1037	121	0	0
B13	10+2	10	343	53	1	4
B14	32+2	54	4787	245	11	150

ITC’99 benchmarks: Δf and size

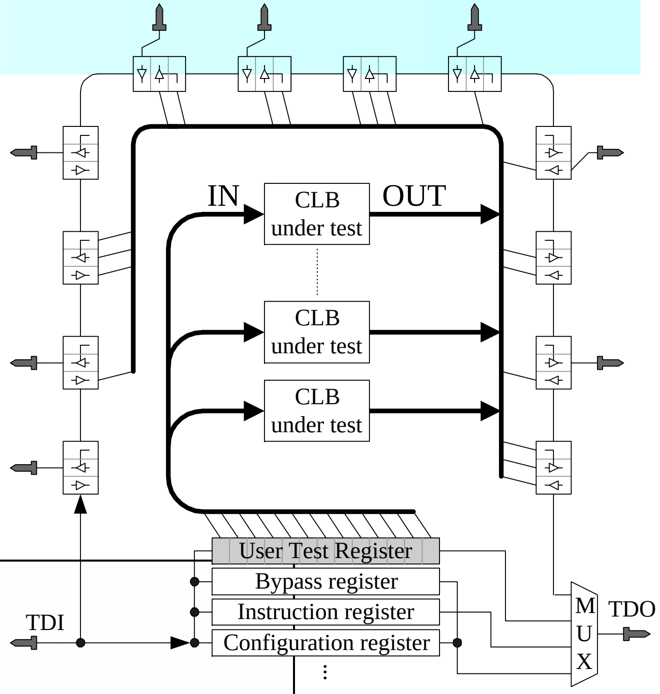
Circuit reference	Maximum frequency deviation (%)		Total size of the reconfiguration files (bytes)		Ratio between the total size of the reconf. files by CLB (%) (horizontal>vertical)
	Vertical	Horizontal	Vertical	Horizontal	
B01	-5,5	0,0	48 350	56 102	16,0
B02	0,0	0,0	7 016	10 623	51,4
B03	-1,9	-4,9	120 705	138 484	14,7
B04	-6,1	-29,3	548 595	665 419	21,3
B05	-17,3	-36,9	1 130 985	1 286 031	13,7
B06	-2,7	0,0	45 291	53 503	18,1
B07	-23,6	-37,8	354 367	425 214	20,0
B08	-5,8	-5,8	150 093	178 339	18,8
B09	-1,8	-4,9	112 107	129 855	15,8
B10	-7,5	-7,6	195 571	245 455	25,5
B11	-10,5	-36,0	500 261	614 093	22,8
B12	0,0	-1,2	1 275 804	1 631 953	27,9
B13	-4,3	-42,8	258 827	332 954	28,6
B14	-13,5	-47,8	5 195 444	6 070 485	16,8

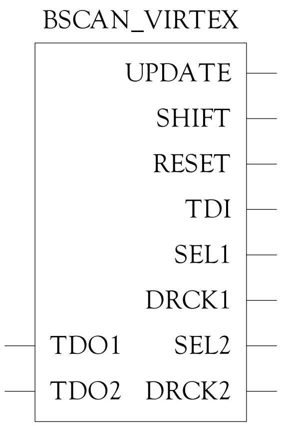
ITC’99 benchmarks: size per CLB

Circuit reference	Number of occupied CLBs	Mean size of the reconfiguration files by CLB (bytes)		Ratio between the mean size value of the reconf. files by CLB (%) (horizontal>vertical)
		Vertical	Horizontal	
B01	6	8 058	9 350	16,03
B02	1	7 016	10 623	51,41
B03	11	10 973	12 589	14,73
B04	54	10 159	12 322	21,30
B05	103	10 980	12 485	13,71
B06	5	9 058	10 700	18,13
B07	31	11 431	13 716	19,99
B08	17	8 829	10 490	18,82
B09	12	9 342	10 821	15,83
B10	20	9 778	12 272	25,51
B11	39	12 827	15 745	22,75
B12	119	10 721	13 713	27,92
B13	37	6 995	8 998	28,64
B14	333	15 601	18 229	16,84

Structural fault detection in CLBs

- Test vector application / response capturing is carried out via the 1149.1 boundary-scan interface





```
Device utilization summary:

  Number of errors:                 0

  Number of 4 input LUTs:           0 out of 4704       0%
  Number of SLICEs:                14 out of 2352       1%
  Number of Slice Flip Flops:      26 out of  4704      1%
  Number of BSCANs:                 1 out of 1        100%


Total equivalent gate count for design:  256
```

Fault detection latency

Partial reconfiguration file size and reconfiguration time for each step in the replication of synchronous circuits with clock enable

Replication using the auxiliary relocation block

	Number of bytes	TCK = 20 MHz Time (ms)
Copy of the internal logic functionality and place of the input signals in parallel	11 289	9,705
BY_C=1∧CC=1	441	0,379
CC=0	277	0,238
BY_C=0	277	0,238
Connect of the clock enable inputs of both CLBs	2 145	1,844
Disconnect of all the auxiliary relocation circuit signals	2 217	1,906
Place of the CLB outputs in parallel	4 129	3,550
Disconnect of the original CLB outputs	1 333	1,146
Disconnect of the original CLB inputs and test configuration	18 392	15,813
Total	40 500	34,820

Partial reconfiguration file size and reconfiguration time for each step in the replication of synchronous circuits with free-running clock and of combinational circuits

Replication without auxiliary relocation block

	Number of bytes	TCK = 20 MHz Time (ms)
Copy of the internal logic functionality and place of the input signals in parallel	12 163	10,457
Place of the CLB outputs in parallel	3 993	3,433
Disconnect of the original CLB outputs	1 073	0,923
Disconnect of the original CLB inputs and test configuration	18 392	15,813
Total	35 621	30,625

Fault detection latency

Partial reconfiguration file size and reconfiguration time of the test configurations

Number of configurations	Number of bytes	TCK = 20 MHz Time (ms)
2nd	3 115	2,678
3rd	623	0,536
4th	634	0,545
5th	613	0,527
6th	512	0,440
Total	5 497	4,726

Shifting time for test vector application

Number of vectors	Length (bits)	Total (bits)	TCK = 20 MHz Application time (ms)
40	13	520	0,066

Shifting time for the test vector responses from a CLB under test

Number of cells of the BS register in a XCV200	Number of vectors	TCK = 20 MHz Shifting time (ms)
1 022	40	4,088

The mean time to test the full CLB matrix is also the worst case **fault detection latency**

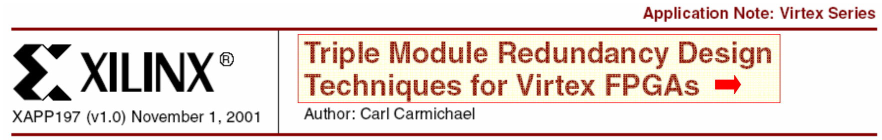
Mean time for the test of a 1 176 CLBs matrix

Occupation type: 25% synchronous + 50% combinational + 25% empty

43 679,188 ms	TCK = 20 MHz
26 472,235 ms	TCK = 33 MHz

Fault detection latency x fault masking

- A fault detection latency higher than 40 s may be acceptable in some applications, but may be a problem in many others
- Fault masking by spatial redundancy may solve the problem until the defective CLB is flagged / soft error is corrected
- See



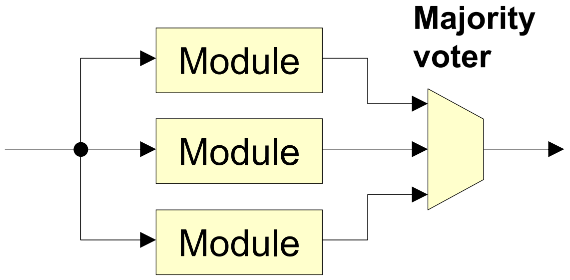
Application Note: Virtex Series

XILINX®

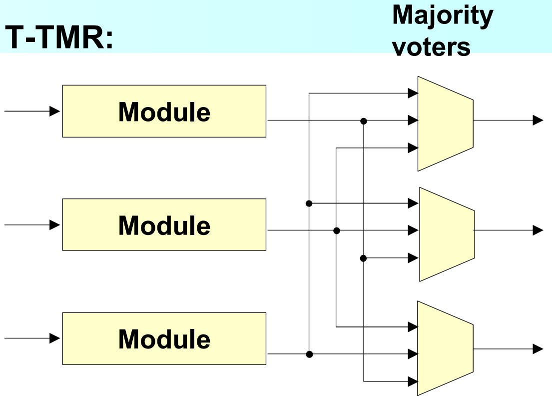
XAPP197 (v1.0) November 1, 2001

Triple Module Redundancy Design Techniques for Virtex FPGAs ➡

Author: Carl Carmichael



Spatial redundancy



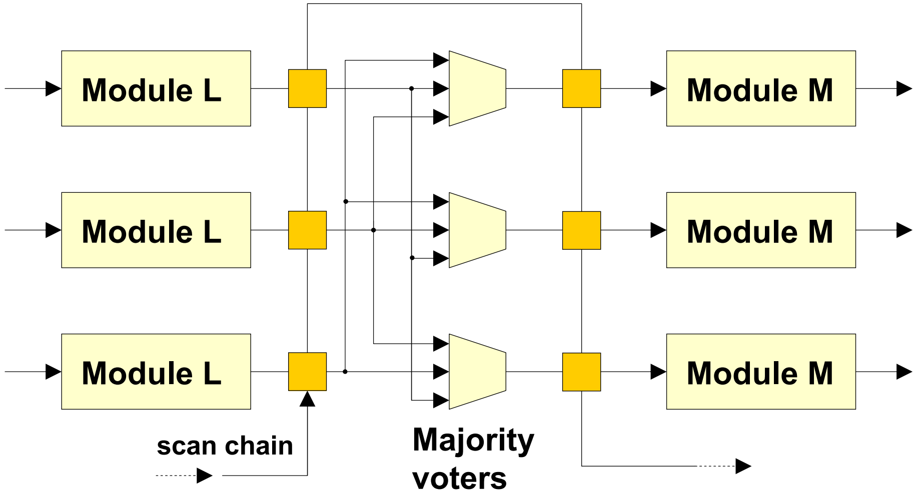
- N-NMR implementations enhance reliability by allowing voter failure
- Earlier NMR implementations were a form of static redundancy, but dynamic reconfiguration brings an added value
  - Just-in-time implementation saves space
  - The reliability index may be restored

Fault detection and correction in N-NMR via replication of CLBs

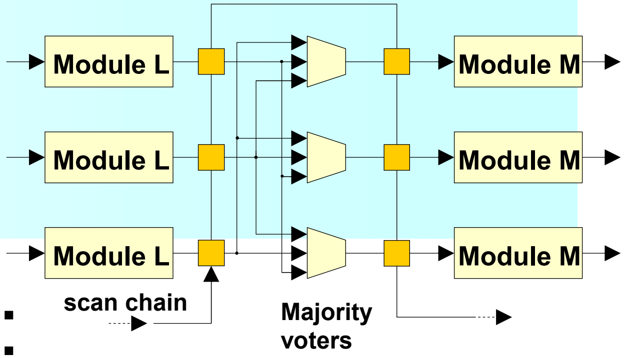
- The CLB testing approach previously described enables the identification of a defective CLB (structural fault)
- Replication will be used to remove the defective CLB from operation (and to reestablish the reliability index)

N-NMR plus online error detection

- An internal scan chain capturing the module and voter outputs enables the detection of incoherencies
- Fault detection latency still exists, but fault masking prevents system malfunction



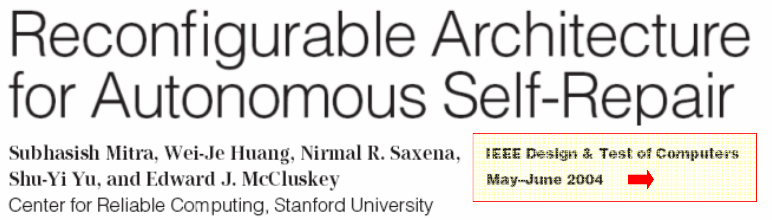
Error correction



- If an incoherency is detected:
  - A scrubbing procedure is launched to read-and-compare the configuration bitstream for the affected area
  - If no error is found, each CLB in the affected module / voter is tested (a defective CLB will be replicated and removed from service)
- Error correction via CLB replication or scrubbing reestablishes the reliability index

Research directions

- One-chip “self-healing” architectures may be achieved via self-reconfiguration (a microprocessor core controls the self-reconfiguration port and scan chains)
- Dual-chip or multi-chip architectures may monitor the error detection circuitry of each other - See



Reconfigurable Architecture for Autonomous Self-Repair

Subhasish Mitra, Wei-Je Huang, Nirmal R. Saxena, Shu-Yi Yu, and Edward J. McCluskey
Center for Reliable Computing, Stanford University

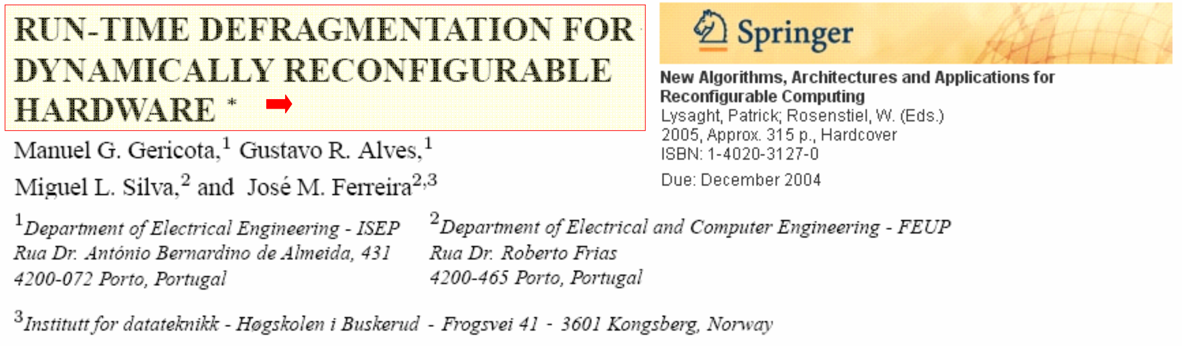
IEEE Design & Test of Computers
May–June 2004 ➡

Conclusion

- The CLB replication and test procedure proposed enables concurrent non-intrusive fault detection, but fault detection latency prevents true fault tolerance
- Combining the proposed fault detection techniques with spatial redundancy enables low overhead fault tolerance for DR-FPGAs (and self-healing for SR-FPGAs)

Conclusion

- Dependability will also be improved by runtime defragmentation of the FPGA logic space (using the proposed CLB replication and test procedure)
- See



RUN-TIME DEFRAGMENTATION FOR DYNAMICALLY RECONFIGURABLE HARDWARE * ➡

Manuel G. Gericota,[1] Gustavo R. Alves,[1]
Miguel L. Silva,[2] and José M. Ferreira[2,3]

[1] *Department of Electrical Engineering - ISEP*
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto, Portugal

[2] *Department of Electrical and Computer Engineering - FEUP*
Rua Dr. Roberto Frias
4200-465 Porto, Portugal

[3] *Institutt for datateknikk - Høgskolen i Buskerud - Frogsvei 41 - 3601 Kongsberg, Norway*

Springer

New Algorithms, Architectures and Applications for Reconfigurable Computing
Lysaght, Patrick; Rosenstiel, W. (Eds.)
2005, Approx. 315 p., Hardcover
ISBN: 1-4020-3127-0
Due: December 2004

Improved dependability for dynamically reconfigurable hardware

Restoration of the reliability index via replication and error correction

J. M. Martins Ferreira [jmf@fe.up.pt]
FEUP / DEEC
Rua Roberto Frias
P-4200-465 Porto

HIBU
Frogsvei 41
N-3603 Kongsberg

Manuel G. Gericota [mgg@dee.isep.ipp.pt]
ISEP / DEE
Rua Dr. António Bernardino de Almeida
P-4200-072 Porto

[this presentation is available online at **http://www.fe.up.pt/~jmf/dak-2004.ppt**]