Mitigating FPGA Interconnect Soft Errors by In-Place LUT Inversion

Naifeng Jing[1], Ju-Yueh Lee[2], Weifeng He[1], Zhigang Mao[1], Lei He[2]
1 School of Microelectronics, Shanghai Jiao Tong University, Shanghai, China
2 Electrical Engineering Department, University of California, Los Angeles, US

Abstract — **Modern SRAM-based FPGAs (Field Programmable Gate Arrays) use multiplexer-based unidirectional routing, and SRAM configuration cells in these multiplexers contribute to the majority of soft errors in FPGAs. In this paper, we formulate an In-Placed inVersion (IPV) on LUT (Look-Up Table) logic polarities to reduce the Soft Error Rate (SER) at chip level, and reveal a locality and NP-Hardness of the IPV problem. We then develop an exact algorithm based on the binary integer linear programming (ILP) and also a heuristic based on the simulated annealing (SA), both enabled by the locality. We report the results for 10 largest MCNC combinational benchmarks synthesized by ABC and then placed and routed by VPR. The results show that IPV obtains close to 4x chip level SER reduction on average and SA is highly effective by obtaining the same SER reduction as ILP does. In addition, IPV does not change placement and routing, and does not affect design closure. To the best of our knowledge, it is the first in-depth study on SER reduction for modern multiplexer-based FPGA routing by in-placed logic re-synthesis.**

Keywords - SRAM-based FPGA; Soft Errors; SER; Routing; Interconnect; Fault Mitigation; Logic Re-synthesis; Logic Polarity

I. INTRODUCTION

Modern SRAM-based FPGAs use SRAM cells to configure logic and interconnects, and their number can be up to 160 million in XilinxVirtex-6 [1]. These FPGAs suffer from single event upset (SEU) induced soft errors, and their resilience against SEU decreases with technology scaling. Therefore, reducing the soft error rate (SER) for SRAM-based FPGAs has gained growing significance. Classic Triple Module Redundancy (TMR) employs circuit redundancy both in LUT and interconnect but with high overhead in area, power and performance. Recent logic re-synthesis techniques, such as ROSE [2], IPR [3], IPD [4] and R2 [5] reduce SER in LUTs by leveraging different logic masking techniques. Targeting on low or no cost in area, power, performance and design closure, they are preferred by non-mission-critical applications, e.g. networkings and communications, which in fact are the primary applications of FPGA. However, these techniques do not explicitly consider the interconnect SER and thus chip level SER reduction could be limited due to the interconnect dominance in FPGA.

Modern FPGA has shifted to the multiplexer-based (MUX-based) unidirectional routing architecture [6][7], where the fault mechanism is different from conventional bidirectional routing as in the previous studies [8][9]. In this paper, considering MUX-based unidirectional routing, we formulate an In-Place inVersion (IPV) of LUT logic polarities to reduce interconnect SER, and reveal a locality and NP-Hardness of the IPV problem. We then develop an exact algorithm based on the binary Integer Linear Programming (ILP) and a heuristic based on the Simulated Annealing (SA), both enabled by the locality. We report the results for 10 largest MCNC benchmark circuits mapped by ABC [10] and then placed and routed by VPR [11]. The results show that IPV can obtain nearly 4x reduction on chip level SER. In addition, SA approach is highly effective by obtaining the same quality of results comparing to exact ILP solutions, but 100x faster in runtime. Our IPV does not change placement and routing, and thus has no impact on design closure. To the best of our knowledge, it is the first in-depth study on SER reduction for modern MUX-based FPGA routing by the in-placed logic re-synthesis.

The paper is organized as follows. Section II introduces the behavior and evaluation of the soft error on the unidirectional routing architecture. Then, we formulate the IPV problem in section III and present IPV properties and algorithms in section IV. Section V shows the experimental results and section VI concludes this paper.

II. PRELIMINARIES

A. FPGA Architecture and Interconnect Fault

An FPGA architecture is mainly defined by Configurable Logic Blocks (CLBs) and routing architectures as illustrated in Fig. 1. Interconnects are critical since they contribute a large portion of total FPGA area and total configuration bits. In our concerned unidirectional routing architecture, both the inter-CLB routing (including connection boxes and switch boxes) and intra-CLB routing employ directional MUXes to route signals. Each MUX is typically configured by several encoded configuration bits (CRAM bits), which contribute to the majority of the CRAM bits in FPGA. For example, we observe that interconnects contribute to nearly 80% of the CRAM bits for the 10 largest MCNC benchmarks when they are synthesized to the minimum FPGA dimensions with 6-input LUTs [12].

As shown from Fig. 1, when one of the bits flips its value due to a soft error on a MUX, an erroneous input signal (dotted) is then selected. If this erroneous signal from the faulty MUX reaches the primary outputs of the chip, a functional failure at the chip level occurs. Note that this fault mechanism is not a bridging fault as studied for bidirectional routing in [8][9].

B. Fault Rate Evaluation

We evaluate the failure rate for each CRAM bit under the single fault assumption[1] by SER_b, which is defined as follows.

[1] Simultaneous multiple SEUs seldom happen in commercial FPGAs as reported in [13].

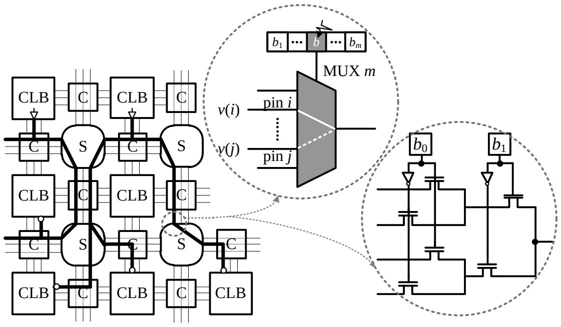

Figure 1. FPGA unidirectional routing and multiplxer structure.

Definition 1: *For a circuit* **C** *with n primary inputs, the soft error rate of a* CRAM *bit b, denoted as* SER$_b$ *is the probability of functional failures on the circuit due to the* SEU *on bit b.*

$$\mathrm{SER}_b = \Pr(\mathbf{C}_b(x) \neq \mathbf{C}_{\bar{b}}(x) \mid b \xrightarrow{\mathrm{SEU}} \bar{b}) \tag{1}$$

where $x \in (0,1)^n$ is the exhaustive set of input vectors and $\mathbf{C}_b(x)$ is the circuit outputs under x, and $\mathbf{C}_{\bar{b}}(x)$ is the circuit outputs when b is flipped due to SEU. The SER$_b$ can be obtained by exhausting 2^n input vectors, which is very time-consuming. In practice, it can be approximated by a Monte Carlo based fault simulation of as many as K times, which can provide a good accuracy as studied in [14].

In general, the metric of SER$_b$ applies to any circuit element as long as it is configured by CRAM bits. For example, we introduce SER$_\mathbf{R}$ in (2), as the total routing SER to evaluate the sensitivity of functional failure to all the CRAM bits in routing elements denoted by **R**. We also quantify the circuit fault rate by all the CRAM bits in various elements in circuit **C** as in (3), which is referred as chip level SER in this paper.

$$\mathrm{SER}_\mathbf{R} = \sum\nolimits_{b\in\mathbf{R}} \mathrm{SER}_b \tag{2}$$

$$\mathrm{SER} = \mathrm{SER}_\mathbf{C} = \sum\nolimits_{b\in\mathbf{C}} \mathrm{SER}_b \tag{3}$$

III. Problem Formulation

A. Motivating Example of Fault Masking

As illustrated in Fig. 1, when MUX m has one of its CRAM bits b flipped due to SEU, the output is driven by net j instead of the desired net i. If j carries a different logic value $v(j)$ from $v(i)$, a fault is injected onto the inputs of the immediate fan-outs of m. However, if $v(j)$ equals to $v(i)$, no fault is injected even if SEU happens. That is, the fault can be instantly masked at m. In addition, SER$_b$ also depends on the observability don't care of MUX m, $obv(m)$, which indicates if the fault can be masked by logic during its propagation to circuit outputs as defined in [3]. As a result, SER$_b$ can be given by $(v(i)\oplus v(j))\bullet obv(m)$.

Note that logic polarity can be independently determined for each input and the output of an LUT in FPGA (see one example in Fig. 2). This technique is called logic polarity inversion and has been used to optimize timing [15] and power [16]. Here, an example in Fig. 3 shows how the logic polarity inversion helps to reduce SER on a single LUT. Given the observability for the MUX and the two values on pin i and pin j involved, one can see that the SER of bit b_k can be reduced from 0.5 to 0 by inverting the logic polarity on net j.

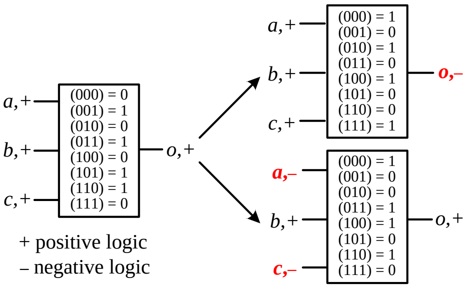

Figure 2. Atomic operations for LUT logic polarity inversion.

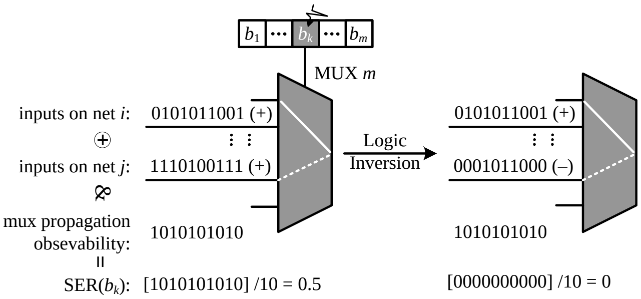

Figure 3. LUT polarity inversion improves fault tolerance on interconnects.

B. Problem Formulation

Logic polarity inversion may lead to conflict among multiple LUTs. An example is illustrated in Fig. 4, where m_1 may require LUT 2 as negative to locally mitigate the fault, while m_2 may require LUT 2 to stay positive. To find an optimal logic polarity assignment for all the LUTs and minimize SER$_\mathbf{R}$, we formulate the In-Place inVersion (IPV) problem as follows.

Formulation 1 (In-Place inVersion Problem): *Given a circuit, assign the logic polarity for each* LUT, *such that the* SER *for all multiplexer-based interconnects is minimized.*

Given the single fault assumption, each CRAM bit connects two LUT nodes, which are called as pseudo fan-in LUT pair and denoted as $L(b_k)$ and $l(b_k)$. For the example in Fig. 4, $L(b_k)$ = LUT 3 is the desired driving LUT and $l(b_k)$ = LUT 1 is the driving LUT selected due to SEU. Our IPV problem tries to reassign the polarities for the pseudo fan-in LUT pairs of each CRAM bit, such that SER$_\mathbf{R}$ can be minimized.

IV. Properties and Algorithms

A. Properties of IPV Problem

Property 1: *Our* IPV *problem is* NP-Hard.

Sketch of Proof: This theorem can be reduced from the binary Max-Sum (labeling) problem which is known to be NP-Hard [17]. We skip the details of the proof due to the limited space.

In IPV problem, when one or multiple LUTs are selected to be inverted for fault masking, SER$_\mathbf{R}$ should be updated accordingly after each inversion. Intuitively, each update needs a new pass of circuit fault simulation that is highly time-consuming. In this paper, we reveal that SER$_\mathbf{R}$ can be analytically updated by a pre-calculation of fault rate on each bit based on theorem 2.

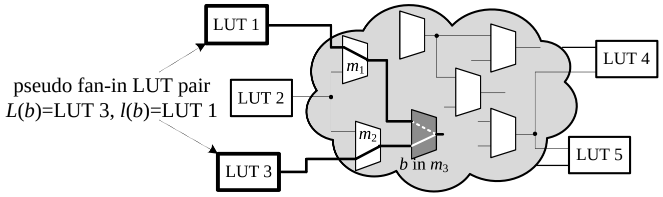

Figure 4. The pseudo fan-in pair of a SEU affected bit.

Property 2 (Locality of Bit SER upon Polarity Inversion): *Under single fault assumption and for given logic network,* SER$_b$ *for a routing* CRAM *bit solely depends on the logic polarities of its pseudo fan-in* LUT *pair, independent of the polarities of the other* LUT*s in the network.*

We also skip the proof due to the limited space here.

B. Locality based SER Calculation

Based on locality theorem, SER$_{\mathbf{R}}$ with possible inversions can be calculated of at most 4x complexity to SER$_{\mathbf{R}}$ as in (4).

$$\mathrm{SER}_b^4 = \begin{cases} \mathrm{SER}_b^{00} & +L(b)\ \&+l(b) \\ \mathrm{SER}_b^{01} & +L(b)\ \&-l(b) \\ \mathrm{SER}_b^{10} & -L(b)\ \&+l(b) \\ \mathrm{SER}_b^{11} & -L(b)\ \&-l(b) \end{cases} \tag{4}$$

where the quadruplicated values of {SER$_b^{00}$, SER$_b^{01}$, SER$_b^{10}$, SER$_b^{11}$} are called SER *quadruplet* of bit *b*. Each quadruplet provides four error rates indicated by the two superscripted numbers, telling if one of its pseudo fan-in LUTs is inverted or not. For abbreviation, we denote it as SER$_b^4$[$P_{L(b)}$, $P_{l(b)}$], where P is a function telling the polarities of LUTs $L(b)$ and $l(b)$, i.e. + or –. Thus, the total routing SER$_{\mathbf{R}}^4$ can be written as

$$\mathrm{SER}_{\mathbf{R}}^4 = \sum_{b\in\mathbf{R}} \mathrm{SER}_b^4[P_{L(b)}, P_{l(b)}] \tag{5}$$

Eq. (5) reveals that SER$_{\mathbf{R}}$ for a given circuit can be updated as an algebraic sum upon each CRAM bit by its SER quadruplet. To get their values, currently we use the fault simulation method that is within 4x complexity as SER$_{\mathbf{R}}$ in (2). In this way, the iterative fault simulation after each reassignment of LUT polarity can be avoided.

C. Binary ILP Based Algorithm

In this section, we use a binary ILP formulation to provide us an insight on the capability of IPV improvement. We take a set of binary variables x_i to denote whether a LUT i is inverted or not, i.e., a positive LUT has its x_i as 1. A binary *inverting quadruplet* {f_b^{00}, f_b^{01}, f_b^{10}, f_b^{11}} is used to denote the polarities of the pseudo fan-in LUTs for each routing bit b. As a result, the binary ILP formulation for our IPV problem is given by

$$\min \quad \mathrm{SER}_{\mathbf{R}}^4 = \sum\nolimits_{b\in\mathbf{R}} \sum\nolimits_{\substack{ij=00,\\01,10,11}} \mathrm{SER}_b^{ij} \cdot f_b^{ij} \tag{6}$$

$$\begin{aligned} \text{s.t.}\quad & f_b^{00} \le 1 - x_s \\ & f_b^{00} \le 1 - x_t \\ & f_b^{00} + 1 \ge 1 - x_s + 1 - x_t \end{aligned} \tag{7}$$

where x_s=$L(b)$, x_t=$l(b)$. This set of constraints models the fact that exactly one value in the quadruplet {SER$_b^{ij}$} should be selected for each bit b using f_b^{ij} as a mask. Other constraints on f_b^{01}, f_b^{10} and f_b^{11} can be similarly written as those in (7).

In our ILP formulation, by forcing corresponding x_i to be 0 in the constraints, it also applies to the situation that some LUT input or output polarities are not invertible.

D. Simulated Annealing Based Algorithm

We also propose a Simulated Annealing (SA) based algorithm to solve IPV efficiently while providing good quality of routing SER reduction compared to ILP. The SA based algorithm starts from the initial circuit with positive logic polarities for all the LUTs. Then, it switches to another LUT polarity assignment by inverting one random LUT polarity at each move. Objective function of the new assignment is evaluated by (5). New reassignment with a better cost is always accepted, and the worse reassignment is accepted conditionally based on the acceptance probability. The annealing starts from a temperature of 0.008, and is updated by a decreasing factor of 1.003. It stops till the minimum temperature of 2.0e-6 is reached.

V. Experimental Results

A. Experimental Settings

In this section, we report experiments for the ten largest MCNC combinational circuits as in [4]. We used parameterized architecture in VPR [11] to characterize different FPGA architectures. Firstly, we perform logic optimization and technology mapping onto 4 and 6-input LUTs by Berkeley ABC [10]. Each mapped circuit is packed by two different CLB architectural settings respectively, i.e. 4-input LUT with cluster size 4 and 6-input LUT with cluster of 8. Then, VPR was used to implement a minimum dimension FPGA array without incurring extra unused CRAM bits that exceeds the actual need of the circuit. We applied Monte Carlo based fault simulation to generate the bit SER quadruplets. Note that our IPV algorithm applies to either combinational or sequential circuits to mitigate interconnect SEU fault as long as the bit SER quadruplets are available. We used Mosek as our ILP solver to seek for the globally optimal assignment of the LUT polarities.

B. Comparision between the ILP and SA appraoches

Table I first presents size statistics of the benchmark circuits. Then, it compares fault rate reductions in terms of SER ratios before and after applying IPV by ILP and SA approaches for all routing MUXes. From the table, one can see that both ILP and SA approaches significantly reduce SER. For example, for 4-input LUT with cluster size 4, the interconnect SER is reduced by 1.2x to 17.2x with an average around 6x. For 6-input LUT with cluster size 8, the SER is reduced about 5.4x on average. In addition, by considering the CRAM bit percentage, the SER can be reduced by 3.97x and 3.67x on average at the chip level for the 4-input and 6-input LUT, respectively.

The SER reduction for 6-input LUT and 8 LUTs per cluster is slightly smaller, because larger LUT and cluster sizes have fewer interconnects and fewer MUXes that limit the room for improvement. While it is natural for different circuits to obtain different improvements, "des" has the lowest and much smaller SER reduction. This is because the values of SER quadruplet in

TABLE I. INTERCONNECT SER REDUCTION FOR 10 BENCHMARK CIRCUITS

Circuit	**LUT size *k*=4, Cluster size *N*=4**							**LUT size *k*=6, Cluster size *N*=8**						
	#LUT	**Int. SER Reduce**		**Int. CRAM bit%**	**Chip SER Reduce**	**Runtime**		**#LUT**	**Int. SER Reduce**		**Int. CRAM bit%**	**Chip SER Reduce**	**Runtime**	
		ILP	**SA**			**ILP**	**SA**		**ILP**	**SA**			**ILP**	**SA**
ex5p	622	2.51	2.51	93.02	2.27	4131.4	35.5	458	1.81	1.90	77.95	1.78	36000*	25.6
alu4	744	2.04	2.05	91.61	1.87	36000*	41.3	524	1.96	1.99	72.37	1.82	36000*	27.2
misex3	773	3.05	3.05	91.61	2.57	4830.0	44.9	530	2.98	2.98	73.39	2.50	3845.6	29.2
apex4	821	4.79	4.79	93.57	3.83	2990.7	58.1	618	4.91	4.91	79.70	3.86	5876.8	43.0
apex2	1014	3.91	3.91	92.76	2.94	584.8	64.8	729	3.75	3.75	76.93	2.82	295.2	51.3
Seq	1084	3.32	3.32	92.76	2.75	2115.4	78.5	782	3.40	3.40	77.61	2.78	7284.4	57.9
ex1010	1120	7.26	7.26	93.18	4.72	4132.3	70.4	682	7.46	7.46	78.87	4.94	5899.2	55.8
Des	1750	1.16	1.17	88.22	1.16	36000*	71.6	1056	1.07	1.09	60.82	1.09	36000*	34.5
Spla	2229	17.22	17.22	94.74	8.95	4602.6	183.8	1524	14.05	14.05	82.32	7.77	811.5	141.5
Pdc	2304	14.60	14.60	94.54	8.59	3159.5	206.3	1609	12.51	12.51	82.71	7.31	5474.1	153.8
Avg.	–	5.99	5.99	92.60	3.97	–	–	–	5.39	5.40	76.27	3.67	–	–

"des" are high and close to each other, which limits the design freedom that can be leveraged by IPV.

Table I finally reports runtimes. This runtime excludes the fault simulation time for SER quadruplets, which is relatively small comparing to the runtime consumed by ILP. From the table, one can see that ILP is able to solve most of circuits optimally, while SA can obtain the same SER reductions as ILP but runs almost 100x faster. For the other circuits as marked where a timeout of 10 hours is applied to the ILP solver like "des", SA obtains slightly higher SER reductions. These results show that the SA approach is highly effective and efficient for IPV problem. It is worthwhile to point out that circuits like "des" are not the largest circuits in experiments, so the efficiency of ILP depends on both circuit size and structure. When ILP and SA obtain the same SER reductions, LUT inversions in their solutions are often not the same and ILP inverts fewer LUTs in general. This implies that there are multiple "optimal" solutions from the point of view of SER reduction.

VI. CONCLUSION AND FUTURE WORK

Targeting the routing multiplexers that are the dominant routing elements in the modern unidirectional FPGA routing architecture, we have identified a new fault masking mechanism and formulated an IPV problem for In-Place inVersion of LUT logic polarities to maximize fault masking for all interconnect multiplexers. We have shown that the problem is NP-Hard and also revealed a locality for IPV. Based on the locality, we have developed two algorithms based on ILP and SA approaches. Experiments have shown that IPV obtains nearly 4x reduction on average for the chip level SER. In addition, SA is effective and efficient because it obtained the same fault reductions as ILP did in most cases, but ran almost 100x faster. Note that IPV does not change placement and routing, therefore it is an in-place optimization. It should have little or no impact on design closure. This will be verified in the future.

While the proposed IPV algorithms and implementations apply to both combinational and sequential circuits, we have not yet developed SER calculation for sequential feedbacks. This will be the focus of our future work. Furthermore, the interaction between IPV and those techniques from literatures such as [2-5] will be studied for further SER mitigation.

REFERENCES

[1] Xilinx, "Virtex-6 Family Overview". Jan-2010.

[2] Yu Hu, Zhe Feng, Lei He, and Rupak Majumdar, "Robust FPGA resynthesis based on fault-tolerant Boolean matching", in *ICCAD*, 2008, pp. 706-713.

[3] Zhe Feng, Yu Hu, Lei He, and Rupak Majumdar, "IPR: in-place reconfiguration for FPGA fault tolerance", in *ICCAD*, 2009, pp. 105-108.

[4] Ju-Yueh Lee, Zhe Feng, and Lei He, "In-Place Decomposition for Robustness in FPGA", in *ICCAD*, 2010.

[5] Manu Jose, Yu Hu, Rupak Majumdar, and Lei He, "Rewiring for robustness", in *47th DAC*, 2010, pp. 469-474.

[6] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and single-driver wires in FPGA interconnect", in *FPT*, 2004, pp. 41-48.

[7] Smith A.M., Constantinides G.A., Wilton S., and Cheung P., "Concurrently optimizing FPGA architecture parameters and transistor sizing: Implications for FPGA design", in *FPT*, 2009, vol. 54-61.

[8] S. Golshan and E. Bozorgzadeh, "Single-event-upset (SEU) awareness in FPGA routing", in *44th DAC*, 2007, pp. 330.

[9] E. S. S. Reddy, V. Chandrasekhar, and M. Sashikanth et al, "Detecting SEU-caused routing errors in SRAM-based FPGAs", in *18th International Conference on VLSI Design*, 2005, pp. 736-741.

[10] *ABC: A system for sequential synthesis and verification*. Berkeley Logic Synthesis and Verification Group.

[11] J. Luu, I. Kuon and P. Jamieson et al., "VPR 5.0: FPGA cad and architecture exploration tools with single-driver routing, heterogeneity and process scaling", in *FPGA*, 2009, pp.133-142.

[12] Naifeng Jing, Ju-Yuen Lee and Zhe Feng et al., "Quantitative SEU fault evaluation for SRAM-based FPGA architectures and synthesis algorithms", *accepted by the 21st International Conference on Field Programmable Logic and Applications*, 2011.

[13] Ken Chapman, "SEU Strategies for Virtex-5 Devices", 2009.

[14] Samuel Luckenbill, Ju-Yueh Lee, Yu Hu, Rupak Majumdar, and Lei He, "RALF: reliability analysis for logic faults: an exact algorithm and its applications", in *DATE*, 2010, pp. 783–788.

[15] Kai Zhu, "Post-route LUT output polarity selection for timing optimization", in *15th International symposium on Field programmable gate arrays*, 2007, pp. 89–96.

[16] J. H. Anderson, F. N. Najm, and T. Tuan, "Active leakage power optimization for FPGAs", in *12th International symposium on Field programmable gate arrays*, 2004, pp. 33–41.

[17] "A Linear Programming Approach to Max-Sum Problem: A Review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1165–1179, Jul. 2007.