

# FPGA 运算单元单粒子错误的 Berger/余数联合结果校验法

邢克飞, 杨 俊, 王跃科

(国防科学技术大学机电工程与自动化学院, 长沙 410073)

**摘 要:** 针对空间环境中 FPGA 局部功能运行的重要运算模块受到单粒子效应的威胁, 提出了 Berger/余数联合结果校验法的加固设计方法。通过分析指出, 由于 Berger 结果校验所需的进位链在 FPGA 中不易获取, 加法、减法、乘法运算的正确性利用余数校验法检测, 其它运算结构利用 Berger 结果校验法检测。给出了联合结果校验法的思路和逻辑实现方法, 并分析了该方法对 FPGA 资源和速度性能的影响。单粒子故障仿真注入实验结果表明: 对于 VirtexII 系列 FPGA 运算单元的单粒子效应故障, Berger/余数联合结果校验法的单粒子效应故障检测概率达到 99.8%, 引入的资源增加为 10~40 个 SLICES, 运算速度降至 50~100MHz。

**关键词:** FPGA 运算单元; Berger 结果校验; 余数结果校验; 单粒子效应

**中图分类号:** V520.6; TN409

**文献标识码:** A

**文章编号:** 1000-1328(2011)09-2045-06

**DOI:** 10.3873/j.issn.1000-1328.2011.09.025

## A Berger and Remainder Combined Result Checking Approach for Errors in FPGA's Operation Units Induced by Single Event Effect

XING Ke-fei, YANG Jun, WANG Yue-ke

(Mechatronics and Automation School, National University of Defense Technology, Changsha 410073, China)

**Abstract:** According to the soft error of FPGA's important operation module in space application, a novel approach named Berger/remainder combined result checking method is presented. Because the carry chain of arithmetic operation is difficult to obtain in FPGA logic design, the correctness of addition, subtraction and multiplication are achieved by the remainder result checking method, and other operation modules are examined by the Berger result checking method. The logic implementation of Berger/remainder combined result checking method is given as a case of Virtex II FPGA, the resource increment and performance degradation of the method is analyzed. A single event effect fault injection experiment indicates that the error detection rate is about 98% for Virtex II series FPGAs with resources increment of 10~40 SLICES and operation speed is dropped down to 50~100MHz.

**Key words:** Operation unit; Berger result checking; Remainder result checking; Single event effect

### 0 引 言

基于 CMOS 工艺的 SRAM 型 FPGA (Static Random-Access Memory based Field Programmable Gate Array) 对单粒子效应敏感性限制了它在太空中的应用, SRAM 型 FPGA 单粒子效应及其加固设计

问题受到了高度重视, 目前 NASA/JPL、欧空局、俄罗斯、巴西都有专门的机构研究 Xilinx 高密度 FPGA 抗单粒子效应设计问题<sup>[1-3]</sup>。

FPGA 在进行信号处理(如傅立叶变换、梳状滤波)时需要进行算术运算、逻辑运算和移位旋转运算。定点乘法运算由 FPGA 内部集成的硬件乘法器

实现,加法、与/或/非等逻辑运算是由内部的 CLB (Configurable Logic Block) 实现,这些部件在受到高能粒子影响后,运算结果可能出现瞬时异常,甚至长时间错误。针对此类问题通常有两类解决方法:通过刷新对 FPGA 配置存储器 SEE 的纠错;通过冗余(如 TMR: Triple Modular Redundancy, DMR: Dual Modular Redundancy)设计加强数据位的容错性能<sup>[3-4]</sup>。这两类方法要么需要通过系统设计对 FPGA 配置存储器进行操作,提高了系统设计的复杂度,要么需要引入大量的冗余逻辑,提高了资源使用量,降低了速度性能。Anghel L 等人在文献[5]中对不同类型加法器电路的 TMR 和 DMR 设计给出

了资源性能影响分析,结果见表 1,可以看出:TMR 或 DMR 使得目标设计的资源增加 100% ~ 200%,速度性能(即最大工作速率)下降了 15% 左右。

文献[6-8]给出一类针对处理器运算单元的检错设计方法,主要是通过校验码来判断处理器运算结果的正确性,文献[9]基于该方法在 Altera FLEX10K 设计中进行了尝试性研究。本文根据航天工程设计实践,将此设计思路用在 FPGA 运算单元的单粒子效应故障检测中,并以 Xilinx SRAM 型 FPGA 为对象进行了实现和应用,论文最后对此法的性能进行了分析。

表 1 TMR 和 DMR 设计的资源使用情况<sup>[5]</sup>

Table 1 The resources used for TMR and DMR design<sup>[5]</sup>

测试电路	TMR		DMC		测试电路	TMR		DMC	
	资源增加/%	性能降低/%	资源增加/%	性能降低/%		资源增加/%	性能降低/%	资源增加/%	性能降低/%
bk8 × 8	194.35	18.33	90.88	15.00	ks16 × 16	195.78	22.00	93.27	17.30
cla8 × 8	194.31	17.74	90.81	14.51	skl16 × 16	195.29	22.00	92.49	12.85
hc8 × 8	194.48	19.64	91.09	16.07	bk32 × 32	194.90	9.48	91.81	7.75
ks8 × 8	195.09	25.00	92.07	20.45	cla32 × 32	194.67	10.57	91.43	8.65
skl8 × 8	194.67	23.91	91.39	19.56	hc32 × 32	195.55	14.10	92.84	11.53
bk16 × 16	194.64	13.09	91.44	10.46	ks32 × 32	196.37	17.17	94.16	14.15
cla16 × 16	194.43	15.27	91.11	12.50	skl32 × 32	195.69	11.22	93.07	9.18
hc16 × 16	195.03	15.27	92.07	12.93					

### 1 运算单元的 Berger/余数联合结果校验

常用的结果校验法有 Berger 校验和余数校验两种,结果校验的实现思路框图如图 1 所示。

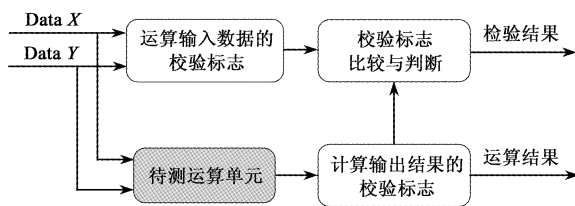


图 1 运算单元结果校验的实现框图

Fig. 1 A frame structure for result checking method of arithmetic & logic unit

结果校验法的输入来自两个部分:第一部分是运算输入数据的校验标志,比如二进制数据“0”或“1”的个数;第二部分是运算输出结果的校验标志,比如进位与否的统计信息。结果校验法通过计算输入数据的校验标志和输出结果的校验标志,将运算结果与预期结果进行比较。条件符合则表明运算过

程正确,否则运算过程有误,需采取其他处理。

#### 1.1 Berger 结果校验法

基于文献[6-9]的算法思路,以二进制加法运算为例,给出 Berger 校验数的推导。现有两个  $n$  位的二进制数:

$$X = (x_{n-1}, x_{n-2}, \dots, x_0),$$

$$Y = (y_{n-1}, y_{n-2}, \dots, y_0)$$

设这两个数进行加法运算的结果为  $S = (s_{n-1}, s_{n-2}, \dots, s_0)$ , 运算过程中的进位链序列为  $C = (c_{n-1}, c_{n-2}, \dots, c_0)$ , 则有

$$x_i + y_i + c_{i-1} = (s_i + c_i) + c_i = s_i + 2c_i \quad (1)$$

其中:  $x_i, y_i, s_i, c_i \in \{0, 1\}$  是第  $i$  个加法操作对应的位。如果令  $N(\cdot)$  表示取二进制数所有位中的 0 的个数,  $c_{in}, c_{out} = c_{n-1}$  分别表示输入和输出的进位,那么:

$$N(X) + N(Y) + c_{in} = N(S) + c_{out} + N(C) \quad (2)$$

这样得出二进制加法运算 Berger 校验数的表达式:

$$S_c = X_c + Y_c - c_{in} - C_c + c_{out} \quad (3)$$

其中:  $S_c, X_c, Y_c, C_c$  与  $N(S), N(X), N(Y), N(C)$  相对应,表示  $S, X, Y, C$  中 0 的个数。

对应于加法/减法/乘法、与/或/异或、移位、旋转运算,Berger 校验相应的校验数列于表 2。

表 2 ALU 对应的 Berger 校验数  
Table 2 Berger checking number of ALU

ALU 运算类别	Berger 校验数
加法(ADD)	$S_c = X_c + Y_c - C_c - c_{in} + c_{out}$
减法(SUB)	$S_c = X_c - Y_c - C_c - c_{in} + c_{out} + n$
乘法(MUL)	$S_c = nX_c + nY_c + X_c \cdot Y_c - C'c$
与(AND)	$S_c = X_c + Y_c - (X \text{ or } Y)c$
或(OR)	$S_c = X_c + Y_c - (X \text{ and } Y)c$
异或(XOR)	$S_c = X_c + Y_c - 2(X \text{ and } Y)c + n$
移位(SHR/SHL)	$S_c = X_c - c_{in} + c_{out}$
旋转(ROR/ROL)	$S_c = X_c$

表 2 中乘法运算校验数中的  $C'$  由下式给出:

$$C' = n^2 - n - N(C') \quad (4)$$

$N(C')$  为向量乘法中所有内部进位之和。下面给出一个加法 Berger 校验的算例,如图 2 所示。

ALU Operation	Berger Check Symbol
(ADD)	Calculation
$C \quad 101111$	$\Rightarrow X_c \quad C_c = 1$
$X \quad 101011$	$\Rightarrow Y_c \quad C_{in} = 0$
$Y \quad + 101101$	$\Rightarrow S_c$
$S \quad 0111000$	$S_c = X_c + Y_c - C_c - C_{in} + C_{out}$ $= 2+2-1-0+1=4$

图 2 加法 Berger 校验的算例

Fig. 2 A Berger checking case for adder

对于二进制数  $X = 101011, Y = 101101$ ,则  $S = X + Y = 011000$ ,进位  $C = 101111$ 。S, X, Y, C 中 0 的个数  $X_c = 2, Y_c = 2, S_c = 4, C_c = 2$ ,此外  $c_{in} = 0, c_{out} = 1$ 。于是  $X_c + Y_c - C_c - c_{in} + c_{out} = 2 + 2 - 1 - 0 + 1 = 4 = S_c$ , 校验结果表明计算过程正确。

### 1.2 余数校验法

余数校验法用在算术运算操作中,如加法、减法和乘法等。余数校验法相对 Berger 校验法使用的资源较少。

数字逻辑中一般进行 16 进制运算,对任意 16 进制数据的算术运算:

$$X = x_{n-1}x_{n-2}, \dots, x_0, \quad Y = y_{n-1}y_{n-2}, \dots, y_0$$

$$S = X \oplus Y = s_{m-1}s_{m-2}, \dots, s_0,$$

$$\oplus \in \{ADD, SUB, MUL\} \quad (5)$$

其中  $n$  为  $X$  和  $Y$  的 16 进制位数, $m$  为  $S$  的 16 进制位数(含溢出进位)。一般为了方便计算取常数  $C = (16 - 1) = 15$  作为除数,分别对  $X$  和  $Y$  取模运算得:

$$R_x = X \bmod C$$

$$= \{ \text{sum}(x_{n-1}, x_{n-2}, \dots, x_0) \} \bmod C,$$

$$R_y = Y \bmod C$$

$$= \{ \text{sum}(y_{n-1}, y_{n-2}, \dots, y_0) \} \bmod C \quad (6)$$

那么,输入的校验数为:

$$R_{C_{in}} = \{R_x \oplus R_y\} \bmod C \quad (7)$$

算术运算的结果  $S$  同样也按位取模  $C$  运算:

$$R_{C_{out}} = S \bmod C$$

$$= \{ \text{sum}(s_{m-1}, s_{m-2}, \dots, s_0) \} \bmod C \quad (8)$$

正确的算术运算应该满足  $R_{C_{in}} = R_{C_{out}}$ 。图 3 给出一个减法余数校验的算例。

ALU Operation	Reminder Check Symbol
(SUB)	Calculation
$X \quad AD9857$	$\Rightarrow R_x = \text{sum}(A,D,9,8,5,7) \bmod 15=7$
$Y \quad - AD6620$	$\Rightarrow R_y = \text{sum}(A,D,6,6,2,0) \bmod 15=7$
$S \quad 003237$	$\Rightarrow R_{C_{in}} = (7-7) \bmod 15=0$
	$R_{C_{out}} = \text{sum}(0+0+3+2+3+7) \bmod 15=0$

图 3 减法余数校验的算例

Fig. 3 A reminder checking case for subtraction

十六进制数  $X = AD9857, Y = AD6620$ ,则  $S = X - Y = 003237$ 。

$$R_x = \text{sum}(A, D, 9, 8, 5, 7) \bmod 15 = 52 \bmod 15 = 7,$$

$$R_y = \text{sum}(A, D, 6, 6, 2, 0) \bmod 15 = 37 \bmod 15 = 7,$$

$$R_{C_{in}} = \text{sum}(R_x - R_y) \bmod 15 = 0 \bmod 15 = 0,$$

$$R_{C_{out}} = \text{sum}(3, 2, 3, 7) \bmod 15 = 15 \bmod 15 = 0,$$

$$R_{C_{out}} = R_{C_{in}}, \text{表明计算过程正确。}$$

### 2 结果校验的 FPGA 实现

Berger 结果校验法在进行算术运算校验时需要用到运算中间的进位链序列  $C$ , 在 FPGA 设计中由于进位链不易访问,因此算术运算的 Berger 结果校验法并不适合在 FPGA 中实现。相反,余数校验法进行算术运算结果校验的时候就容易得多。因此, FPGA 运算单元结果校验过程可采用 Berger 校验法

和余数校验法两者的结合,即利用 Berger 校验法检测表 2 中除了加法、减法、乘法以外运算结果的正确性,利用余数校验法检测加法、减法、乘法运算结果的正确性。

采用结果校验后的运算单元在资源利用和速度性能上都将受到影响。表 3 给出了常用运算单元在采用检错设计前后资源增加和速度降低的比例。测试用例选择了信号处理中常用的 16 进制有符号数运算,FPGA 为 XC2V40-4,数据取自 ISE8.2.03i 在默认设置情况下的综合结果。

移位和旋转运算在 FPGA 中应用较少,其资源增量和速度降低比例与位操作相差不大,本文不做专门讨论。从表中可以看出运算单元采用了结果校验检错设计之后,增加了运算流水线长度,运算速度降低较为明显,并且越是高速运算其速度降低幅度越大,最后都降低至 50~100MHz 左右;资源使用增加 10~40 个 SLICES,增加量并不大,并且,资源增

量是绝对值,不随代码规模变化,以 XC2V40 为例,只增加了 15% 左右的 CLB 资源,如果对于 XC2V1000,则只有 0.8% 左右的增量。

与 TMR 或 DMR 相比,联合结果校验法的资源增加很少,但是对速度性能的影响较大。其主要原因是运算单元的初始运算速度较高,均在 100MHz 以上。对于带宽在 20MHz 以内的信号,文献[10]显示,星载信号处理平台数据运算的速度一般在 80MHz 以内,结果校验的速度可以直接满足要求;对于带宽大于 20MHz 的高速信号可以通过并行处理来降低速度的压力,比如将 80MHz 带宽的信号用四个运算模块并行工作,那么,每个模块的承受的信号带宽就可以等效于 20MHz;同时,结果校验主要用于为数不多的关键或重要运算模块中,因此,运算单元的结果校验可以在不影响正常的信号处理功能的条件下,辅助检测单粒子效应引起的错误。

表 3 运算单元结果校验法的资源和性能变化情况

Table 3 Influence to resources and performance by result checking approach for operation unit

类别		原始参数	校验设计参数	参数对比	
$S = A \pm B$ 8bits 加法/减法 余数校验	资源使用	SLICES	13	24	增加 11
		FFs	24	29	增加 5
		4-LUTs	8	31	增加 23
	速度性能 (MHz)	122.865	100.776	减小 18%	
$S = A \times B$ 8bits 乘法 余数校验	资源使用	SLICES	9	27	增加 18
		FFs	16	33	增加 17
		4-LUTs	0	38	增加 38
	速度性能 (MHz)	100.492	53.926	减小 46.3%	
乘法运算使用了 FPGA 的硬件乘法器 (MULT18 × 18s)					
$S = A \& B / S = A   B$ 8bits 与/或 Berger 校验	资源使用	SLICES	14	46	增加 32
		FFs	23	53	增加 30
		4-LUTs	8	75	增加 67
	速度性能 (MHz)	229.410	78.370	减小 66%	
$S = A \wedge B$ 8bits 异或 Berger 校验	资源使用	SLICES	14	50	增加 36
		FFs	24	53	增加 29
		4-LUTs	8	69	增加 61
	速度性能 (MHz)	229.410	73.169	减小 68%	

### 3 运算单元单粒子效应检测性能分析

运算单元 Berger/余数联合校验法的检错性能可以利用故障注入实验进行分析。该故障注入系统有三个主要功能:按照故障模型和故障生成方法产生实验需要的故障矢量集  $F$ ;利用专门的接口控制器把故障注入到被测对象的指定位置;收集被测对象的故障响应信息并进行分析。故障注入试验过程

如图 4 所示。

算术逻辑运算单元故障矢量集的生成:以 Xilinx FPGA 为例,算术逻辑运算是通过可配置位配置 Slice 的逻辑状态完成的,因此,可以通过修改相关运算模块对应的配置位来完成故障模拟。不同的故障位置就对应一组表现形式不同的故障矢量集(配置存储器文件集合)。

故障注入接口控制器:由一片 FPGA 实现,它按

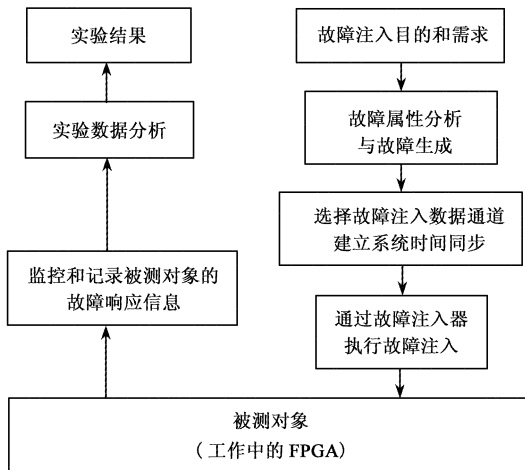


图 4 故障注入试验流程

Fig. 4 Flow chart of fault injection experiment

照注入信息协议要求,在某一确定时刻,将计算机生成的故障(配置存储器文件)在不中断工作负载的情况下,注入到被测 FPGA 的指定位置中,同时将故障监控信息采集、传输到计算机中。

结果分析:由专用计算机软件实现,该软件通过收集被测 FPGA 对某一故障矢量的响应信息,完成该次试验的分析,包括故障检测情况、检测概率计算等。

本节仍以表 2 中的测试用例为故障注入对象,在一个 FPGA 设计中分别包含了 8bits 的 16 进制加法、减法、乘法等算术运算和与、或、异或等逻辑运算。利用图 5 所示的故障注入系统,在故障生成计算机中随机生成算术逻辑运算故障矢量集  $F$ ,并根据单粒子效应的伴随特性,通过配置存储器的相应配置位的翻转获得运算单元的故障模型,最后通过故障注入接口控制器动态注入到被测 FPGA 中。

故障注入试验中用到的测试用例如下:

用例 1:  $y_1 = A + B; y_2 = y_1 \text{ or } C; y_3 = y_2 \text{ and } D;$

$y_4 = y_3 \text{ xor } E; y_5 = y_4 - F; y = y_5 \times G;$

用例 2:  $y_1 = A + B; y_2 = y_1 - C; y_3 = y_2 \text{ and } D;$

$y_4 = y_3 \text{ or } E; y_5 = y_4 \text{ xor } F; y = y_5 \times G;$

用例 3:  $y_1 = A \text{ and } B; y_2 = C \text{ or } D; y_3 = E \text{ xor } F;$

$y_4 = y_1 + y_2; y_5 = y_4 - y_3; y = y_5 \times G;$

用例 4:  $y_1 = A + B; y_2 = C - D; y_3 = y_1 \text{ xor } y_2;$

$y_4 = y_3 \text{ and } E; y_5 = y_4 \text{ or } F; y = y_5 \times G;$

用例 5:  $y_1 = A \text{ or } B; y_2 = y_1 \text{ and } C; y_3 = D + E;$

$y_4 = y_2 - y_3; y_5 = y_4 \text{ xor } F; y = y_5 \times G;$

其中,  $A、B、C、D、E、F、G$  为 8bits 输入信号,  $y$  为

16bits 输出信号。

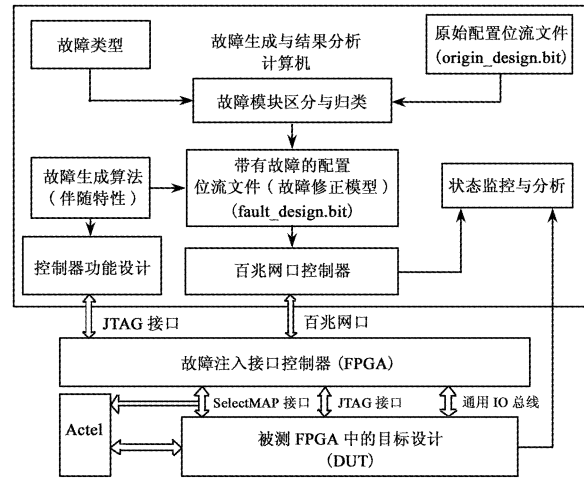


图 5 SRAM 型 FPGA SEE 故障注入系统结构

Fig. 5 System frame of single event effect fault injection experiment for SRAM-based FPGA

对测试用例进行故障注入试验后,得到 Berger/余数联合校验法的故障检测性能见表 4。从表 4 中可以看出,对于运算单元的 SEE 故障, Berger/余数校验法的单粒子效应故障检测概率达到 99.8%。对每个测试用例共注入 60,000 个运算单元故障,没有检测到的故障约为 100 个左右,这些故障一部分是故障检测算法漏掉的运算单元瞬时的故障(故障持续时间太短),另一部分是由于故障注入修正模型在形成运算单元故障的时候具有一定的偏差概率,也就是说“故障”注入并没有真正的形成“故障”引起的,因此实际的空间应用中,运算单元的 Berger/余数校验法故障检测概率将会更高。与 TMR/DMR 等运算单元单粒子效应检测方法相比,资源使用并未明显提高,但检错性能却没有下降。

表 4 运算单元故障检测性能

Table 4 Error detection performance for operation unit

测试用例	资源使用	试验次数	每次故障注入数量	平均故障检测数量	故障检测率
1	154	100	600	598.7	99.8%
2	156	100	600	599.1	99.9%
3	160	100	600	599.0	99.8%
4	161	100	600	598.6	99.8%
5	164	100	600	599.0	99.8%

#### 4 结论

本文从空间应用背景下 FPGA 局部功能运行的重要运算模块易受到单粒子效应威胁角度出发,研究

了 Berger/余数联合结果校验的加固设计方法。通过对两种结构校验方法的阐述比较,指出由于 Berger 结果校验所需的进位链不易获取,在 FPGA 设计中加法、减法、乘法运算的正确性利用余数校验法检测,其它运算结构利用 Berger 结果校验法检测。

FPGA 实现资源与性能测试表明:运算单元采用了结果校验检错设计之后,增加了运算流水线长度,运算速度和资源使用量受到影响,但资源增量是个绝对值,不随代码规模变化。故障注入实验研究结果表明,算术运算和逻辑运算的 Berger/余数联合校验方法对于运算单元的 SEE 故障,检测概率超过 99%,与 TMR/DMR 等运算单元单粒子效应检测方法相比,具有明显优势,非常适合在资源受限的空间仪器设备中采用。

该方法已经在星载信号处理平台的单粒子效应防护设计中得到成功应用。

### 参 考 文 献

- [ 1 ] Henry B G, Insoo J, Allan J, et al. Analysis of single-event upset rates on the Clementine and Cassini solid - state recorders [J]. *Journal of Spacecraft and Rockets*, 2010, 47(1): 169 - 176.
- [ 2 ] Chumakov A I, Vasil'ev A L, Kozlov A A, et al. Single-event-effect prediction for ICs in a space environment [J]. *Russian Microelectronics*, 2010, 39(2): 74 - 78.
- [ 3 ] Fernanda L K, Luigi C, Ricardo R. Fault-tolerance techniques for SRAM-based FPGAs[M]. Springer, Netherlands, 2006.
- [ 4 ] 邢克飞, 杨俊, 王跃科, 等. Xilinx SRAM 型 FPGA 抗辐射设计技术研究[J]. *宇航学报*, 2007, 28(1): 123 - 129. [ Xing Ke-fei, Yang Jun, Wang Yue-ke, et al. Study on the anti-radiation technique for xilinx SRAM-based FPGA[J]. *Journal of Astronautics*, 2007, 28(1): 123 - 129. ]
- [ 5 ] Anghel L, Alexandrescu D, Nicolaidis M. Evaluation of soft error tolerance technique based on time and/or space redundancy [C]. *The 13rd Symposium on Integrated Circuits and Systems Design ( ICSD2000 )*, Manaus, Brazil, September 18 - 24, 2000.
- [ 6 ] Lo J C. An SFS berger check prediction ALU and its application to self checking processor designs [J]. *IEEE Transactions on Computer-Aided Design Of Integrated Circuits and Systems*, 1992, 11(4): 525 - 540.
- [ 7 ] Lo J C. Concurrent error detection in arithmetic and logical operations using Berger codes [C]. *The 9th Symposium on computer arithmetic*, Santa Monica, California, USA, September 6 - 8, 1989.
- [ 8 ] Metra C. Novel berger code checker [C]. *IEEE International Workshop on Defect and Fault tolerance in VLSI systems*, Lafayette, Louisiana, November 13 - 15, 1995.
- [ 9 ] Srinivasan V, Julian W F, William H R, et al. Evaluation of error detection strategies for an FPGA-based self-checking arithmetic and logic unit [C]. *Military and Aerospace Applications of Programmable Devices and Technologies Conference ( MAPLD )*, Washington DC, USA, September 7 - 9, 2005.
- [ 10 ] 邢克飞, 杨俊, 王跃科, 等. 星载软件无线电平台及其单粒子效应研究方法 [J]. *空间科学学报*, 2006. 26(6): 477 - 482. [ Xing Ke-fei, Yang Jun, Wang Yue-ke, et al. Single event simulation and evaluation methodology for spaceborne software radio platform [J]. *Chinese Journal of Space Science*, 2006, 26(6): 477 - 482. ]

作者简介:邢克飞(1979 - ),男,讲师/博士,主要从事空间仪器工程以及超大规模集成电路单粒子效应故障特性与防护技术等方面的教学与科研工作。

通信地址:长沙国防科技大学三院 323 教研室(410073)

电话:(0731)84575313

E-mail:StevenXing@nudt.edu.cn

(编辑:余 未)