

前导数字并行纠错单元的设计与仿真

魏思捷^{1,2}, 富 坤^{1,2}, 袁玉倩^{1,2}, 耿跃华³

(1 河北工业大学 计算机科学与软件学院, 天津 300401;

2 河北省大数据计算重点实验室, 天津 300401;

3 河北工业大学 电气工程学院, 天津 300130)

摘 要: 对前导数字预测算法的误差修正逻辑进行分析改进和设计实现, 重点对该误差修正纠错模块的逻辑设计进行了分析证明, 依据设计的逻辑表达式对其电路进行了设计, 同时采用硬件描述语言 VerilogHDL 编程, 结果使用 Quartus II 进行仿真验证, 使用性能分析软件对提出的纠错逻辑方案进行验证, 可以看出本纠错单元的电路在电路面积和功耗上都有明显的改善。

关键词: 前导数字预测; 误差修正逻辑; VerilogHDL; Quartus II

中图分类号: TP3

文献标识码: A

文章编号: 1000-7180(2017)05-0067-06

Parallel Correction Unit of Leading Digital's Design and Simulation

WEI Si-jie^{1,2}, FU Kun^{1,2}, YUAN Yu-qian^{1,2}, GENG Yue-hua³

(1 School of Computer Science & Engineering, Hebei University of Technology, Tianjin 300401, China;

2 Hebei Province Key Laboratory of Big Data calculation, Tianjin 300401, China;

3 School of Electrical Engineering, Hebei University of Technology, Tianjin 300130, China)

Abstract: In this paper, we put forward a novel design of the error correction logic. At the same time, we also design the circuit and programed it with Quartus II. At the end of the article we also give the verification and simulation results. The results of the performance analysis tools show that the error correction circuit we put forward in this article improved significantly in the circuit area and the power.

Key words: leading-one prediction; error correction logic; verilogHDL; quartus II

1 引言

前导数字的预测算法广泛应用于浮点运算的加减运算中, 以达到提高浮点运算结果规格化移位的效率, 有可能会产生一位误差. 针对这一误差的产生, 误差修正逻辑的电路设计应运而生. 前导数字的预测算法分为两部分: 前导数字预测单元(Leading Zero/ One Prediction)和前导数字检测单元(Leading Zero/One Detection). 前导数字预测单元是将进行运算的两个尾数表示成编码串的形式, 前导数字检测单元对前导数字的位置进行检测, 即首 1 出现在哪一位上. 在这之后再使用错误纠正逻辑对其

预测的结果进行纠错移位.

2 错误修正逻辑研究现状

错误纠正逻辑在不准确的前导数字预测单元中是非常重要的部分, 其纠错过程处于关键路径上, 在以往的文献中对该纠错过程有许多研究. 文献[1]中提出了一种判定树纠错方法, 文献[2-4]是基于该种判定树纠错方法做出的改进. 这几种纠错方法均使用补偿移位纠错, 纠错结构如图 1 所示. 该移位器处于关键路径上, 这样会大大增加有效延时. 图 1(b)所示的结构图中, 错误检测逻辑是使用了加法器中进位信号来决定移位的位数的. 首先, 加法器的

移位位数是使用前导数字预测器的移位位数在粗移位中进行移位的. 错误检测单元使用进位选择单元来检测前导位的进位, 同时检测前导数字预测器中的输出的错误, 而这一位误差的纠正使用的是精移位器. 文献[1]提出的这种纠错电路位于关键路径上, 由于需要进位信号, 所以进位必须被加法器计算出来, 所以会限制加法器的优化实现. 文献[5-6]描述的是另一种方法, 其结构如图 1(c)所示, 该种纠错方法和加法器以及前导数字预测器是并行进行的. 不精确的前导数字预测器的预测结果先进入粗移位器进行移位, 错误检测逻辑的输出结果进入精细移位器进行移位.

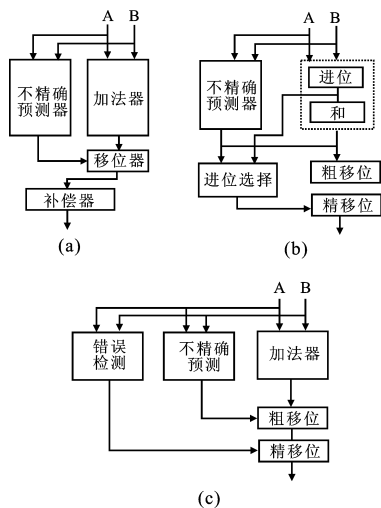


图 1 纠错方法结构图

3 前导数字预测的预编码逻辑

对浮点运算中的两个操作数 A, B 的尾数部分进行预编码是为了将两位尾数使用两态变量进行编码, 方便将其使用字母变量进行表示, 实现数字信号的设计. 对其进行编码实现对于前导数字的预测.

将 A 和 B 两个尾数分别表示为: $A = a_{n-1}a_{n-2}\cdots a_1a_0$, $B = b_{n-1}b_{n-2}\cdots b_1b_0$.

将 $C = c_{n-1}c_{n-2}\cdots c_1c_0$ 定义为

$$c_i = \begin{cases} b_i & \text{如果为加法} \\ \bar{b}_i & \text{如果为减法} \end{cases}$$

在预编码模块中, 分别定义进位传播函数 T, 进位产生函数 G, 进位消除函数 Z, 在这里 t_i, g_i, z_i 分别代表 T, G, Z 的第 i 位. 其表达式如下:

$$t_i = a_i \oplus c_i, g_i = a_i c_i, z_i = a_i c_i.$$

前导数字的进位传播函数 T 表示的是该位不产生进位, 如果有进位传播到此位置就向高位传播, 进位产生函数 G 就是产生进位位的标志, 进位消除函

数 Z 就是如果有进位传播到这一位上, 那么在此位置不再向高位传播.

产生前导 0 的时候, 会产生模式串 $t * g z *$ 和 $z * z$, 这里的 $*$ 仍然代表 0 或者任意串. 同样的, 在产生前导 1 的时候, 会产生模式串 $t * z g *$ 和 $g * g$. 指示函数使用 F 表示后有:

$$f_i = t_{i+2}(z_{i+1}z_i + g_{i+1}g_i) + t_{i+2}(g_{i+1}z_i + z_{i+1}g_i).$$

对于 f_{n-2} 和 f_{n-1} 的计算:

$$t_{n+1} = t_n = t_{n-1}, z_n = z_{n-1}, g_n = g_{n-1}.$$

由于前导数字的预测逻辑是为了寻找前导 1 或者前导 0, 所以这种编码逻辑对于前导数字预测的计算不管是加法或者减法来说都是很合适的, 与符号的正负无关. 不管结果的输出符号是正还是负, 预测串中均包含前导 0. 通过计算 LZD 输出的偏移量来计算预测串中前导 0 的个数.

前导数字的检测模块是用来检测前导数字预测器中产生的前导数字的个数的. 该前导数字检测模块对前导数字预测模块产生的标志向量 F 进行编码, 即将其编码为位串的形式, 然后使用粗粒度移位进行规格化移位. 首先, 预测串必须被转换成 one-hot(独热码)的形式, 即除了前导数字的显著位之外的其他位均置为 0, 例如: 串 00001101 的 one-hot 表示就是 00001000. 为了表示方便, 定义一个变量 $S = s_{n-1}, s_{n-2}, \cdots, s_1 s_0$.

$$s_i = f_{n-1} + f_{n-2} + \cdots + f_i.$$

序列 S 出现的形式将会是 00...01...11, 就是说对于相邻的两位来说可能出现的情况是 (0,0)(0,1)(1,1) 这三种形式, 但是不会出现 (1,0) 根据这一事实可以知道, 当 $i > j$ 的时候有 $s_i s_j = s_i, s_i + s_j = s_i$, 根据这两种情况可以将由八位二进制序列 F 推出的带有权值的 ($\log_2^n$) 位的 Z 的形式简化为

$$\bar{z}_2 = f_7 + f_6 + f_5 + f_4;$$

$$\bar{z}_1 = f_7 + f_6 + \bar{f}_5 \cdot \bar{f}_4 (f_3 + f_2);$$

$$\bar{z}_0 = f_7 + f_6 \cdot f_5 + \bar{f}_6 (\bar{f}_4 f_3 + \bar{f}_5 \bar{f}_2 f_1).$$

而通过这种方法我们可以得到, 粗粒度移位的规格化移位量 $Z(z_2 z_1 z_0)$ 就是粗粒度移位器的移位数. 但是 Z 只能计算 0 ~ 7 范围之内的前导 0 的个数, 但是对于都是 0 的情况并没有做出说明, 需要定义一个全零标识符 V, 在 $F = 0$ 时 Z 的每位都为 1, 所以 f_0 并没有参加 z 的计算, 只是通过 s_0 计算 V, 所以该全零标识符定义 $\sum f_i (7 < i < 0)$.

4 并行纠错模块的设计

本文所提出的错误检测方法是从前导数字预测器的输入开始的,该前导数字的预测单元包含预测器、前导数字检测和加法器.这样一来,前导数字的预测器的一部分硬件就被重复使用了,这样就可以减少硬件面积和功耗了.

在文献中[4-6]中,相比之前提出的方法来说输出结果的时间较早,只要粗粒子移位转换器的输出在预测结果输出之前输出,就不会受影响.这样,在本文提出的结构中,利用错误检测模块和粗粒度移位器模块的时间差,来减少时延、功耗以及硬件面积.

这种方法是检查在最低显著位 LSB 和前导数字位置之间的 $\bar{z} * gx$ 子串, x 表示任意串.前导数字预测器中产生的 S 串,即在第三部分提到的 S .被当做下面运算的输入: $K = S \cdot Z, J = S \cdot G$. K, J 串分别表示了, $\bar{z}$ 和 g 在前导数字和 LSB 之间是否会产生.串 $M = m_{n-1}, m_{n-2} \cdots m_1, m_0$, 是为了从前导数字位置和 LSB 之间除去消除位中第一次出现的生成位 $R = \bar{M}J$.

所以现在串 R 记录了产生位的出现,这是负责对前导数字位置上的进位的.产生位在 R 中至少出现一次,我们使用如下公式来检查:

$$c = r_{n-1} + r_{n-2} + \cdots + r_1 + r_0.$$

$c = 1$ 表示前导数字的位置上有一个进位.但是错误纠正仍然是由加法器的输出来决定的.当加法器的输出是正数并且 $c = 1$ 时或者加法器的输出是负的并且 $c = 0$ 时,错误的修正是需要的.

$$C = c \oplus \text{SUM}[n-1]$$

在这里 $\text{SUM}[n-1]$ 是表示加法器输出的最高显著位 MSB. C 是一个单独的位数字,如果 C 的值是 1 的时候,表示需要 1 位的纠正,需要向右进行 1 位的精细移位, C 的值为 0 的时候,表示这时候不需要进行位置的错误修正.

本节对纠错单元的逻辑理论进行了改进和设计,只对前导数字之后的数字串进行编码的这种方法,可以准确地将前导数字之后有无进位的情况检测出来,如果有进位产生并且可以传播到前导位置上,那么对纠错的结果进行改正.

5 并行纠错单元的硬件实现

对于纠错逻辑方法在理论上进行设计之后需要对其电路进行硬件设计,才能更加完整清晰地对该

纠错单元进行验证.为了使得验证的过程逻辑结构更加清晰,对本文的纠错逻辑单元的电路设计采用分层的方法,这样在每层输出变量的同时都能对该层输出的变量进行检验.

第一级电路如图 2 所示,由于第一级电路是要得出进位传播函数 T ,进位消除函数 Z 和进位产生函数 G ,这三个函数的产生是本纠错电路的基本形式,在设计之初是想按照加法和减法分开的方式进行输出,仿真的时候就要使用不一样的基本模块,这样就是两个电路,不能有效地减少电路面积.所以在第一级电路的设计上,借鉴了多路选择器的设计方法, s 作为一个选择信号,标志着该纠错电路的尾数进行的是加法运算还是减法运算,在这里规定 $s = 0$ 时表示的是减法, $s = 1$ 时表示的是加法.这样就实现了在加法和减法的时候,分别取不同的 b 值,即逻辑表达中使用的中间变量 c .按照逻辑表达式分别输出 Z, T, G .

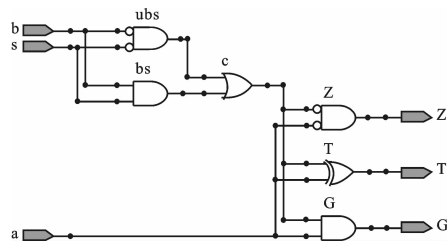


图2 第一级电路

在得出 T, G, Z 之后就要对其结果进行编码,本章的电路设计是以 8 bit 的基础模块为例,进行结果输出,所以使用第一级电路构成 8 bit 位的 T, G, Z 输出模块如图 3 所示.

第三级电路如图 4 所示,是对第二级电路的得出的 T, G, Z 三个变量的结果进行逻辑编码处理,使用一个变量 f 对需要进行纠错的几种模式串一一表示,这种表示方式就是使用模式串的方式对所有的纠错模式进行划分覆盖,而将这几种纠错的情形都放在一起,将其结果进行逻辑或运算,若存在其中任何一种形式的纠错模式串, f 即可将其输出.图 4 表示的就是对两个 8 bit 的尾数存在的纠错模式进行综合检测并将其输出.

使用第三级电路得出的结果,将第三级电路作为一个子模块,使用其输出对得出的结果进行操作.第三级电路得出的信号是 f ,可以看出图 5 中得出的输出结果是 M 和 J ,这里的 M 由于在下一级电路会用到 $\bar{M}$,所以在这里的输出结果中直接输出的是 $\bar{M}$,但是为了方便表示,直接将变量名记为 M .使用

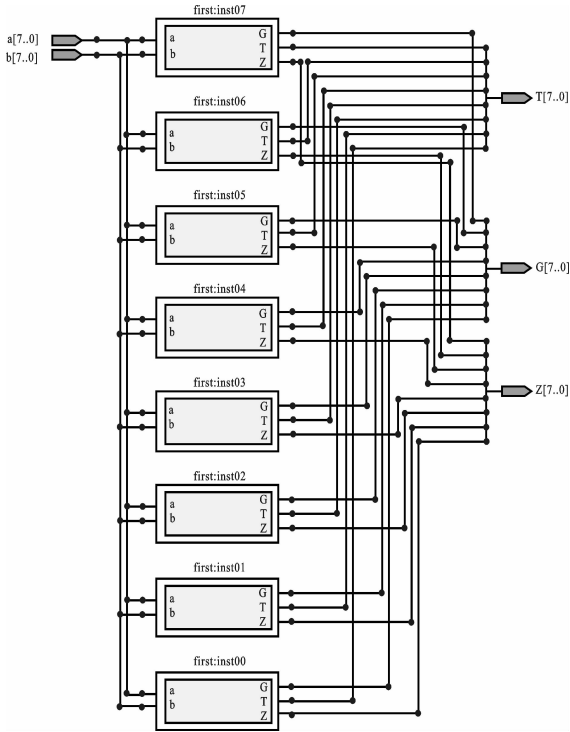


图 3 第二级电路

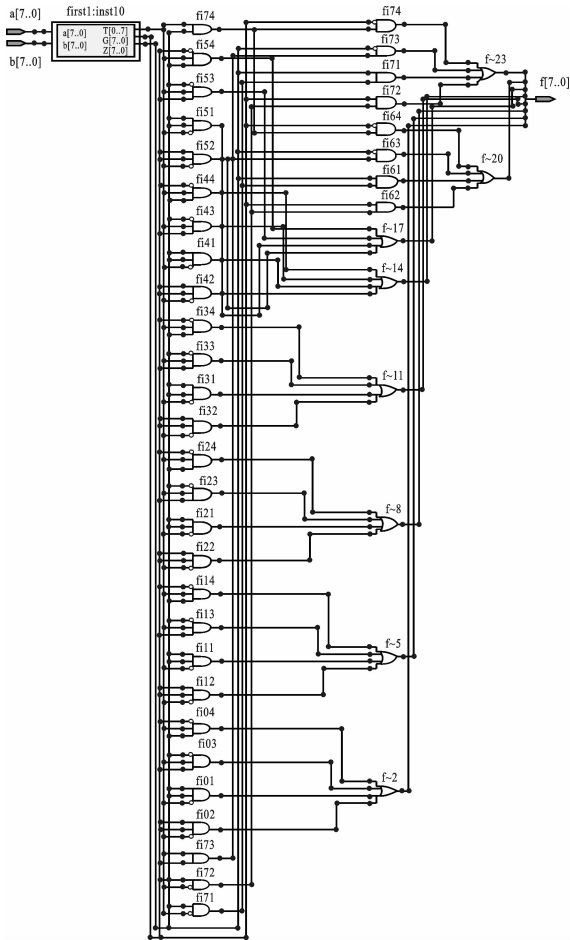


图 4 第三级电路

输出信号 f 将得出的数字串表示成 one-hot 形式 s 直接在电路中作为一个中间变量未输出, K 和 J 分别代表的是 z 和 g 在前导位和最低显著位之间的位置, 变量 M 的出现时为了消除位于前导位和最低显著位之间产生的进位。

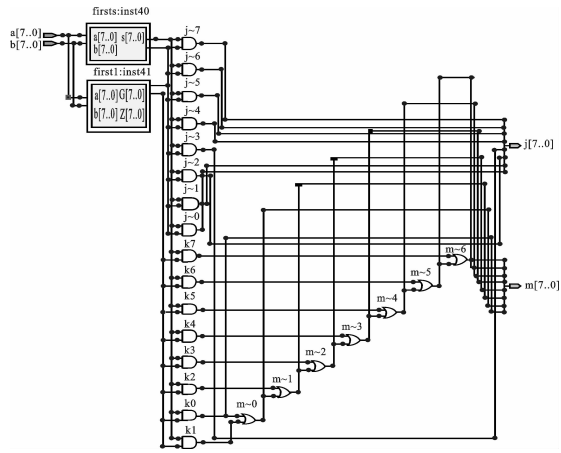


图 5 第四级电路

最后输出的纠错信号是 c , 这其中有一个输出的中间信号 R 在这里没有对 R 进行直接输出是为了节省一级电路, 直接将上一级电路得出的 $\bar{M}$, J 进行逻辑与运算得出 R , 再由 R 得到 c . 这里得出的最终结果 c 表示的情况是, 有实验数据输入的时候, 在前导位和最低显著位之间是否有进位产生. 最终信号输出电路如图 6 所示。

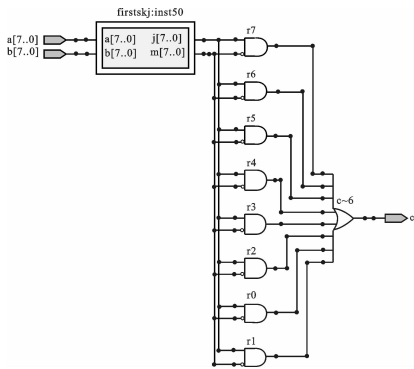
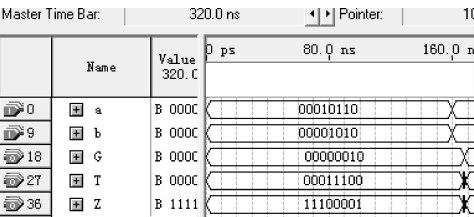


图 6 最终信号输出电路

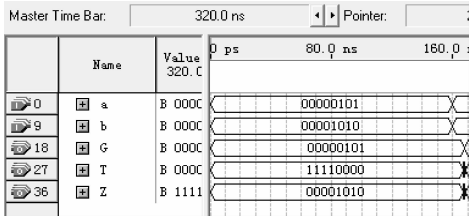
6 仿真结果及性能分析

改进的并行纠错方法是只是用加法器中的一位最高显著位, 不同于进位选择逻辑中需要使用加法器中的进位位. 尽管本章改进的并行纠错电路也需要用到加法器中的输出, 但是它不在关键路径上. 该纠错逻辑是在加法器输出结果之前就输出结果的, 并且它的输出也是在粗移位器输出结果之前输出

的. 下面使用两个例子分别对本电路进行验证,检测其输出的结果是否与理论的输出值一致.

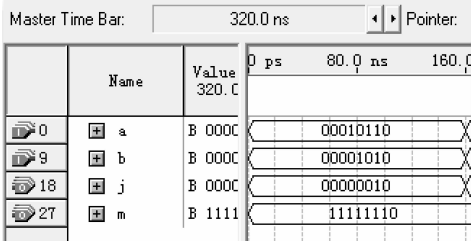


(a)加法结果

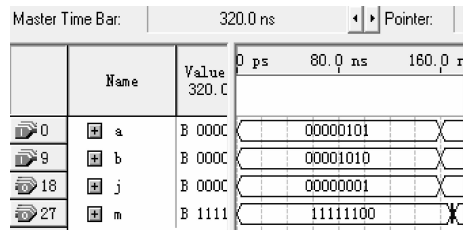


(b)减法结果

图 7 变量 T, G, Z 仿真结果



(a)加法结果



(b)减法结果

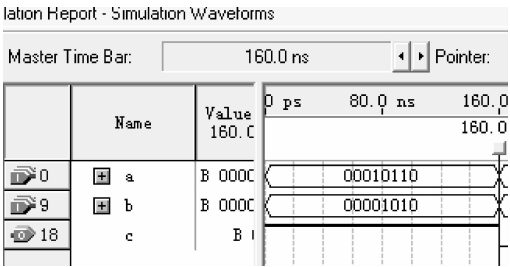
图 8 变量 M, J 仿真结果

如表 1 所示,按照逻辑表达式,将其理论结果计算如下,对比仿真验证得出的结果,均正确. 所以本文设计的电路在在逻辑和硬件实现上正确.

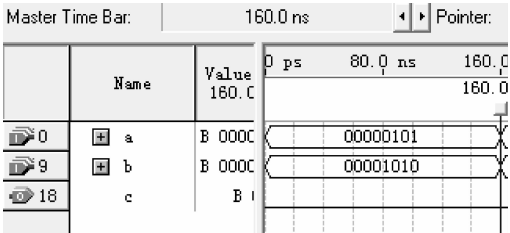
表 1 加法和减法的举例结果

加法		减法	
A	00010110	A	00000101
B	00001010	B	00001010
T	00011100	T	11110000
G	00000010	G	00000101
Z	11100001	Z	00001010
$\bar{M}$	11111110	$\bar{M}$	11111100
J	00000010	J	00000001
C	1	C	0

如表 1 所示的两组测试数据,分别是加法和减法的情况. 列表将各级变量的输出值都给出,方便和各级仿真结果进行一一对比. 图 7~9 分别是加法的情况下产生纠错信号的仿真结果,对比表中的理论值,证明本章设计的电路在逻辑上是正确的.



(a)加法结果



(b)减法结果

图 9 最终输出 C 仿真结果

使用 Design Compiler 对本文的电路进行性能分析,按照本文提出的逻辑设计出的硬件电路和文献[1]中的纠错判定树的方法进行对比性能,比较结果如表 2 所示,相比文献[1]中提出的判定树纠错方法本文提出的改进的并行纠错模块,在电路面积和功耗的减少方面有明显的优势.

表 2 性能分析比较

性能指标	输出时间/ ps	电路面积/ μm^2	功耗/ nW
判定树纠错	716	121	1 767
改进的并行纠错	1 240	69	1 013

7 结束语

本文提出的是一种新的前导数字预测的纠错设计方案,本纠错设计方案不基于加法器的进位信号,不同于以往的设计. 相比并行的纠错判定树方法,本纠错逻辑设计方案中,使用加减选择信号,相比分别预测前导 0 和前导 1 的纠错单元,将前导数字(0 和 1)的预测集成为一个模块,使电路面积节省了一半. 此外可以通过复用部分硬件,在不增加额外延时的前提下减少了电路面积,降低了功耗,通过以上方

法,提高了前导数字纠错单元的运算性能和工作效率.这样就可以提升整个浮点运算单元的规格化移位的效率,并且可对浮点加法器的运算速度改进起到很大的改进作用.

参考文献:

- [1] Javier D Bruguera, Tomas Lang. Leading-one prediction with concurrent position correction[J]. IEEE Transactions Computers, 1999, 48(10):1083-1097.
- [2] Suzuki H, Morinaka H, Makino H, et al. Leading-zero anticipatory logic for high-speed floating point addition[J]. IEEE Journal of Solid State Circuits, 1996, 31(8): 1157-1164.
- [3] Kershaw R, Bays L, Freyman R, et al. A programmable digital signal processor with 32-bit floating-point arithmetic[C]//IEEE Solid Circuits Conference, Digest of Papers, IEEE, 1985.
- [4] Hays W, Kershaw R, Bays L, et al. A 32-bit VLSI

digital signal processor[J]. IEEE Journal of Solid State Circuits, 1985, SC-20(5):998-1004.

- [5] Zhang Ge, Qi Zichu, Hu Weiwu. Anovel design of leading zero anticipation circuit with parallel error detection[C]//IEEE International Symposium on Circuits and Systems. IEEE, 2005.
- [6] Yao Tao, Gao Deyuan. A novel concurrent error detection circuit for leading zero anticipator[C]// Proc. of 2nd International Conference on Computer Engineering and Technology (ICCET), 2010. IEEE, 2010.

作者简介:

魏思捷 女, (1992-), 硕士研究生. 研究方向为计算机智能控制、电子设计自动化.

富坤(通讯作者) 女, (1979-), 博士, 副教授. 研究方向为可重构计算、计算机系统结构.

E-mail: fukun@hebut.edu.cn.

(上接第 66 页)

行进行了初步探索,分析了 SkyEye 的内部架构和 VxWorks 启动原理,并选择 BootRom 仿真作为系统仿真的第一步,BootRom 的汇编代码是对 VxWorks 启动过程的表述,也是逐步完成研究的重要部分.通过改写 SkyEye 的装载模块,实现了对 BIN 文件的装载兼容;以 LDFSR 指令的设计为例介绍了通用指令设计方式.

本文旨在为完整的 VxWorks 操作系统的仿真运行提供研究基础.

参考文献:

- [1] 王宇舟,王二超. Leon2 微处理器 IP 核原理及应用[J]. 单片机与嵌入式系统应用, 2005, 6(8): 41-45.
- [2] 王雷,樊晓桢,王党辉. 龙芯 3A 平台 Vxworks 移植的研究和实现[J]. 微电子学与计算机, 2012, 29(2): 86-90.
- [3] 张忠,樊留群. VxWorks 在 S3C2410 上的 BSP 设计[J]. 微型电脑应用, 2005, 21(10): 22-25.

作者简介:

赵愚 男, (1992-), 硕士研究生. 研究方向为计算机应用技术. E-mail: zhaoyu1504@qq.com.

席隆 男, (1965-), 正高级工程师. 研究方向为空间综合电子学.