

三操作数的前导 1 预测算法纠错编码模块的设计与实现

王京京¹, 富 坤¹, 程婷婷¹, 魏思捷¹, 耿跃华²

(1 河北工业大学 计算机科学与软件学院, 天津 300401; 2 河北工业大学 电气工程学院, 天津 300130)

摘 要: 针对三操作数预测算法中的预测误差, 提出了纠错方案。依据前导 1 预测的流程以及预编码规则, 设计了纠错模块的预编码规则, 完成了其整体结构设计, 采用硬件描述语言 VHDL 进行编程, 并通过 Quartus II 进行仿真验证, 得到了相应的 RTL 级电路, 最后对仿真结果进行了分析。仿真结果表明, 设计完成的算法结构能够有效地完成前导 1 预测纠错的功能。

关键词: 前导 1 预测算法; 纠错编码模块; FPGA; VHDL; Quartus II

中图分类号: TP302.2

文献标识码: A

文章编号: 1000-7180(2016)04-0010-06

Algorithmic Design and Implementation of A Three-Operand Leading-one Prediction Error-Correcting Code Module

WANG Jing-jing¹, FU Kun¹, CHENG Ting-ting¹, WEI Si-jie¹, GENG Yao-hua²

(1 School of Computer Science and Engineering, Hebei University of Technology, Tianjin 400401, China;

2 School of Electrical Engineering, Hebei University of Technology, Tianjin 300401, China)

Abstract: In this paper, we propose an error correction scheme based on the prediction error of the three operation number prediction algorithm. According to the process of leading-one prediction and pre-encoding rules, designing a error-correcting module pre-encoding, and completing the overall structure of the error-correcting module, using hardware description language VHDL to program some function module to optimize the design process, and use Quartus II tools to perform comprehensive validation, received the RTL-level hardware circuits, and last the results are analyzed and verified. Simulation results show that the designed algorithm structure can effectively complete the Leading-one prediction correcting function.

Key words: leading-one prediction; error-correcting code module; FPGA; VerilogHDL; Quartus II

1 引言

前导 1 预测算法是对浮点加法运算后的结果进行首 1 位置预测的一种算法, 其在整体的性能上为关键路径, 但是前导 1 预测的位置是有 1 位误差存在的, 因此, 需要设计一个模块来判断预测结果的正确性——前导 1 预测纠错模块。本文参考了两位操作数预判纠错模块的相关文献^[1], 设计出了三个操作数^[2]预判纠错模块的预编码规则, 并根据表达式完成了硬件的实现, 实现了该模块的整体结构, 最后

通过 Quartus II 进行仿真验证, 以证明此方案是否能真正实现预判误差纠错的功能。

图 1 所示为前导 1 预测的结构^[3-5], 其主要包含两个部分, 一部分做基本的前导 1 预测, 另一部分负责对前导 1 位置的纠错。首先在预测部分, 通过预编码规则生成一串“0”, “1”序列, 然后前导 1 检测单元计算出“0”, “1”序列中第一个“1”的所在位置, 并通过纠正树来产生的结果来判断前导 1 位置是否需要纠正。

收稿日期: 2015-07-30; **修回日期:** 2015-09-01

基金项目: 国家自然科学基金(31100711); 河北省高等学校科学技术研究青年基金(20111122); 河北省交通运输厅科学技术项目计划(Y2011087)

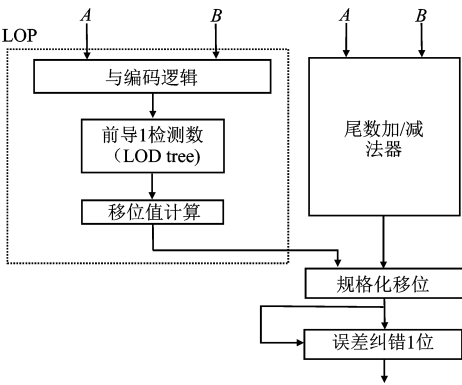


图 1 前导 1 预测的结构图

2 三操作数的前导 1 预测算法编码

根据前导 1 预测算法编码规则^[6-9],对 $F=A+B-C$ 的结果的正负分别采取两种不同的编码方式,即采用正编码和负编码两种情况,根据符号的结果后再进行选择,通过预编码部分可得到结果向量 F ,然后利用编码树电路来对这个向量 F 的首 1 位置进行检测和编码,从而得到了这个首 1 位置的二进制编码。编码规则如表 1、表 2 所示。

表 1 $F \geq 0$ 首 1 位置预测规则

$F = A + B - C$	首 1 的位置	$f_i f_{i+1}$
$0^k 2(s)$	k 或 $k+1$	$0_i 2_{i+1}$
$0^k 12(s)$	k 或 $k+1$	$1_{i+1} 2_{i+2}$
$0^k 11^m(-1,0,2)(s)$	k 或 $k+1$	$1_{i+1} 1_{i+2}$
$0^k 10(s)$	$k+1$ 或 $k+2$	$1_i 0_{i+1}$
$0^k 1(-1)^m 2(s)$	$k+m$ 或 $k+m+1$	$(-1)_{i+1} 2_{i+2}$
$0^k 1(-1)^m 1(s)$	$k+m$ 或 $k+m+1$	$(-1)_{i+1} 1_{i+2}$
$0^k 1(-1)^m 0(s)$	$k+m+1$ 或 $k+m+2$	$(-1)_i 0_{i+1}$

表 2 $F < 0$ 首 1 位置预测规则

$H = C A - B$	首 1 的位置	$f_i f_{i+1}$
$0^k 11(s)$	$k+1$ 或 $k+2$	$(-1)_i (-1)_{i+1}$
$0^k 101(s)$	$k+1$ 或 $k+2$	$(-1)_i 0_{i+1} (-1)_{i+2}$
$0^k 100(s)$	$k+1$ 或 $k+2$	$(-1)_i 0_{i+1} 0_{i+2}$
$0^k 10(-1)^m(s)$	$k+2$ 或 $k+3$	$0_i 1_{i+1}$
$0^k 10(-2)^m(0,1)(s)$	$k+m+1$ 或 $k+m+2$	$0_i 2_{i+1} 0_{i+2}$ $0_i 2_{i+1} (-1)_{i+2}$ $2_i 2_{i+1} 0_{i+2}$ $2_i 2_{i+1} (-1)_{i+2}$
$0^k 10(-2)^m(-1)^n(s)$	$k+m+2$ 或 $k+m+3$	$0_{i-1} 2_i 1_{i+1}$ $2_{i-1} 2_i 1_{i+1}$
$0^k 1(-1)^m 1(s)$	$k+m+1$ 或 $k+m+2$	$1_i (-1)_{i+1}$
$0^k 1(-1)^m 01(s)$	$k+m+1$ 或 $k+m+2$	$1_i 0_{i+1} (-1)_{i+2}$
$0^k 1(-1)^m 00(s)$	$k+m+1$ 或 $k+m+2$	$1_i 0_{i+1} 0_{i+2}$
$0^k 1(-1)^m 0(-1)^n(s)$	$k+m+2$ 或 $k+m+3$	$0_i 1_{i+1}$
$0^k 1(-1)^m 0(-2)^n(s)$	同第 5、6 行	同第 5、6 行
$0^k 1(-1)^m(-2)(s)$	以上所有情况	以上所有情况
$0^k 1(-1)(-2)^m(s)$	以上所有情况	以上所有情况

接下来引入几个中间变量 N, Z, O, T 来分别表示 F_i 中的 $-1, 0, 1, 2$, 即可得到逻辑表达式:

$$f_i(\text{pos}) = Z_i T_{i+1} + (O_{i+1} + N_{i+1})(O_{i+2} + T_{i+2}) + (O_i + N_i) Z_{i+1}$$
$$f_i(\text{neg}) = (N_i + O_i)(N_{i+1} + Z_{i+1} N_{i+2} + Z_{i+1} Z_{i+2}) + (Z_i + T_i) T_{i+1} (Z_{i+2} + N_{i+2}) + (Z_{i-1} + T_{i-1}) T_i O_{i+1} + Z_i O_{i+1}$$

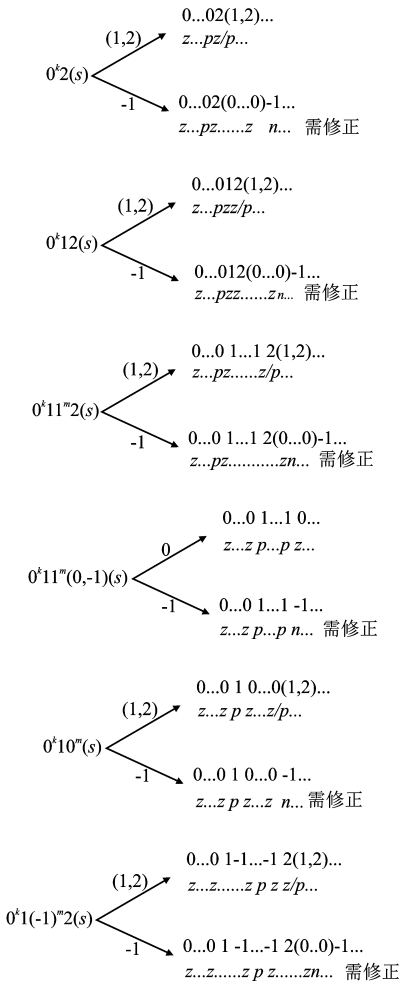
3 前导 1 预测纠错模块预编码的设计与实现

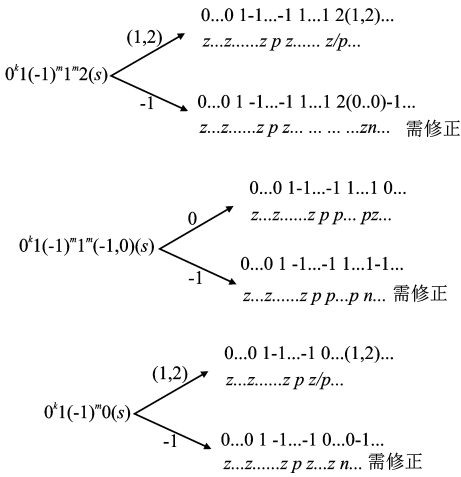
由以上的表可以看出,预编码规则会产生 1 位误差,本文依据三操作数的预编码逻辑规则^[2],设计了纠错模块预编码规则及硬件实现方案,完成了相应各子模块的设计,并通过 Quartus II 进行仿真。由仿真结果分析可得到,设计的预编码实现方案可以有效的完成功能。

3.1 前导 1 纠错模块预编码规则

对于前导 1 的预测,关键是要找到所有会产生 1 位误差的情况,在这里分情况一一列举出来。

对 $F \geq 0$ 的情况进行分析:





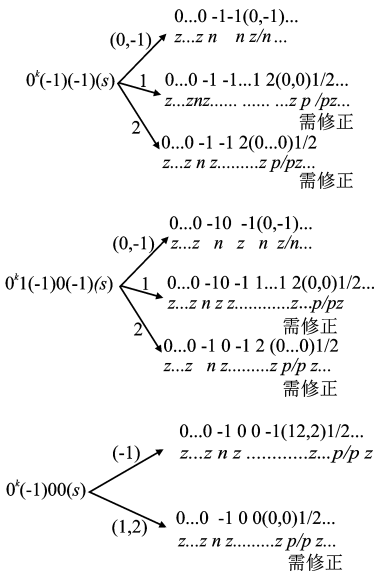
由以上形式可以看出,分别将(-1)、0、1、2,用中间变量 N 、 Z 、 O 、 T 来表示,通过表达式运算转化成新的数串 G_p ,其中 n 、 z 、 p 分别表示(-1)、0、1,由此总结归纳出表达式如下:

$$p_i = Z_i T_{i+1} + O_{i+1} T_{i+2} + O_{i+1} O_{i+2} + O_i Z_{i+1} + (N_{i+1} + O_{i+1}) N_i Z_{i+1} + (N_i + O_i) N_{i+1} O_{i+2} + (N_i + O_i) N_{i+1} T_{i+2}$$

$$n_i = Z_{i-1} N_i \overline{T_{i+1}} + T_{i-1} N_i \overline{T_{i+1}} \quad z_i = \overline{p_i} + n_i$$

通过对以上几种情况的分析,可以归纳总结出需要纠正的 G_p 几种形式: $z^k p^m n(s)$ 、 $z^k p p^m z(s)$ 、 $z^k p p^m n(s)$,当有任意一种出现时,就需要进行对预测的位置进行加 1 修正。

对 $F < 0$ 的情况进行分析:



同理分析,剩下的几种情况在此不再赘述,其需要修正的情况与以上出现的情况一致。

由以上形式可以看出,分别将(-1)、0、1、2,用中间变量 N 、 Z 、 O 、 T 来表示,通过表达式运算转化

成新的数串 G_n ,其中 n 、 z 、 p 分别表示(-1)、0、1,由此总结归纳出表达式如下:

$$n_i = (N_i + (N_{i-1} + O_{i-1})O_i)(N_{i+1} + Z_{i+1}N_{i+2} + Z_{i+1}Z_{i+2}) + (N_{i-1}Z_i + (Z_{i-1} + T_{i-1})T_i)T_{i+1}(Z_{i+2} + N_{i+2}) + (N_{i-2}Z_{i-1} + (Z_{i-2} + T_{i-2})T_{i-1})T_iO_{i+1} + (N_{i-1} + O_{i-1})Z_iO_{i+1}$$

$$p_i = (O_{i-2}T_{i-1} + (Z_{i-2} + T_{i-2})Z_{i-1})O_i + (O_{i-1} + N_{i-1})T_iT_{i+1} + (Z_{i-1} + T_{i-1})Z_iT_{i+1}$$

$$z_i = \overline{p_i} + n_i$$

通过对以上几种情况的分析,可以归纳总结出需要纠正的 G_n 几种形式: $z^k n z^m p(s)$,当这种形式出现时,就需要进行对预测的位置进行加 1 修正。

3.2 预编码规则的硬件实现方案

上节总结出了由 $F \geq 0$ 、 $F < 0$ 转化成 G_p 、 G_n 的表达式,根据表达式运算后即可得到用 n 、 p 、 z 来表示的 G_p 和 G_n . 本节通过 VHDL 语言在 Quartus II 里设计电路来实现预编码结果的输出. 当输出的 G_n 与理论上应该输出的 G_n 一致时,则说明设计的方案是切实可行的。

首先,设计第一级预编码子模块,即中间变量 Z 、 O 、 T 、 N 的输出功能模块,如图 2 所示。

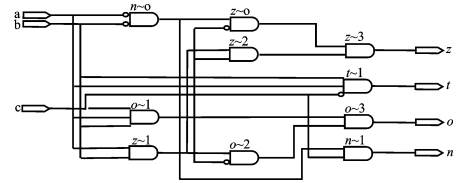


图 2 第一级(RTL 结构图)

然后,在第一级预编码子模块的基础上设计第二级预编码子模块,在这里需要将第一级模块加载到第二级当中,根据上一小节总结出的表达式设计出第 i 位 G_i 输出功能子模块的组合逻辑电路,如图 3、图 4 所示为经 Quartus II 综合后的 RTL 结构图。

在前两级的基础上设计第三级功能子模块,同理也需要将第二级和第一级加载到第三级当中. 在第三级中会得到 8 位输出,即可得到 8 位的 g 输出串,如图 5、图 6 所示。

4 前导 1 预测纠错模块的功能验证

在上一节中通过实现预编码模块得到 8 位的向量 G ,接下来验证硬件电路得到的向量 G 的正确性。

4.1 实例列举

这两种情况分别列举个例来进行验证.

F 分为两种情况,分别为 $F \geq 0$ 和 $F < 0$. 对于

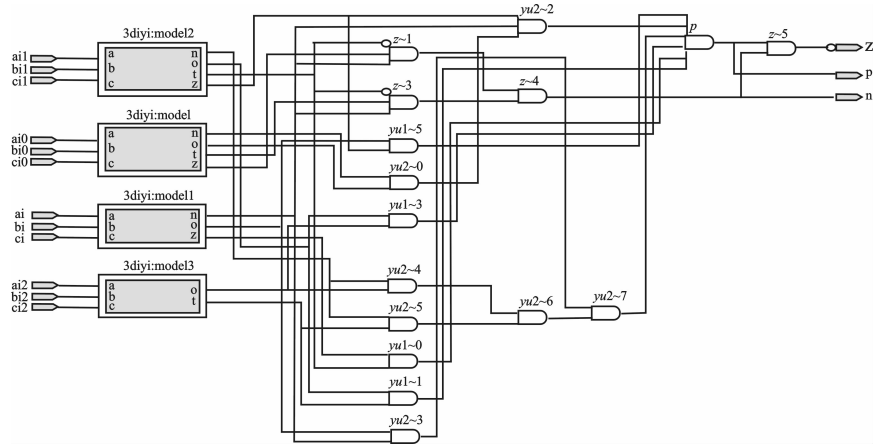


图 3 第二级 $F \geq 0$ (RTL 结构图)

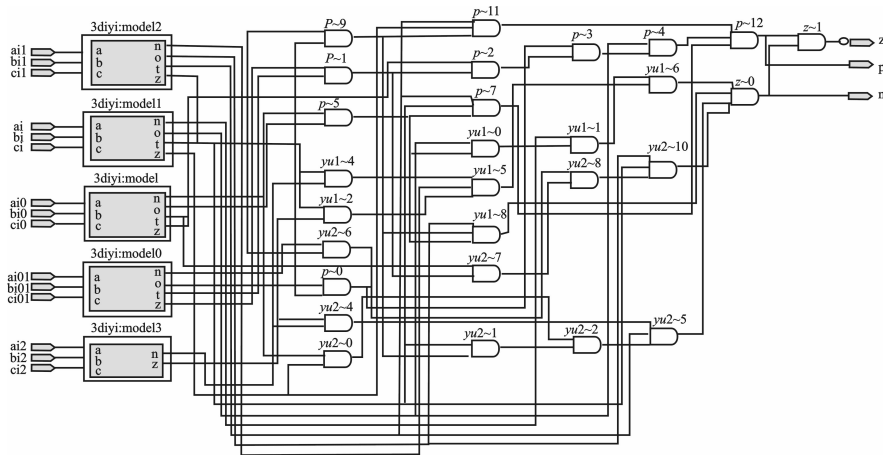


图 4 第二级 $F < 0$ (RTL 结构图)

对 $F \geq 0$ 几种需要修正的情况进行分析:

$0^k 2(s)$ 中需要修正的情况:

A	0	0	1	0	0	0	0	0
B	0	0	1	0	0	0	0	0
C	0	0	0	1	1	1	0	0
F_p	Z	Z	T	N	N	N	Z	Z
G_p	z	p	z	n	z	p	z	z

$0^k 12(s)$ 中需要修正的情况:

A	0	0	0	1	0	0	0	0
B	0	0	1	1	0	0	0	0
C	0	0	0	0	0	0	1	0
F_p	Z	Z	O	T	Z	Z	N	Z
G_p	z	p	z	z	z	z	n	z

$0^k 11^m(0, -1)(s)$ 中需要修正的情况:

A	0	0	1	1	0	0	0	0
B	0	0	0	0	1	1	0	0
C	0	0	0	0	0	0	0	1
F_p	Z	Z	O	O	O	O	Z	N
G_p	z	p	p	p	z	p	z	n

对 $F < 0$ 几种需要修正的情况进行分析:

$0^k(-1)(-1)1(s)$ 中需要修正的情况:

A	0	0	0	1	1	1	0	0
B	0	0	0	0	0	1	0	1
C	0	1	1	0	0	0	0	0
F_p	Z	N	N	O	O	T	Z	O
G_n	z	n	z	z	z	z	z	p

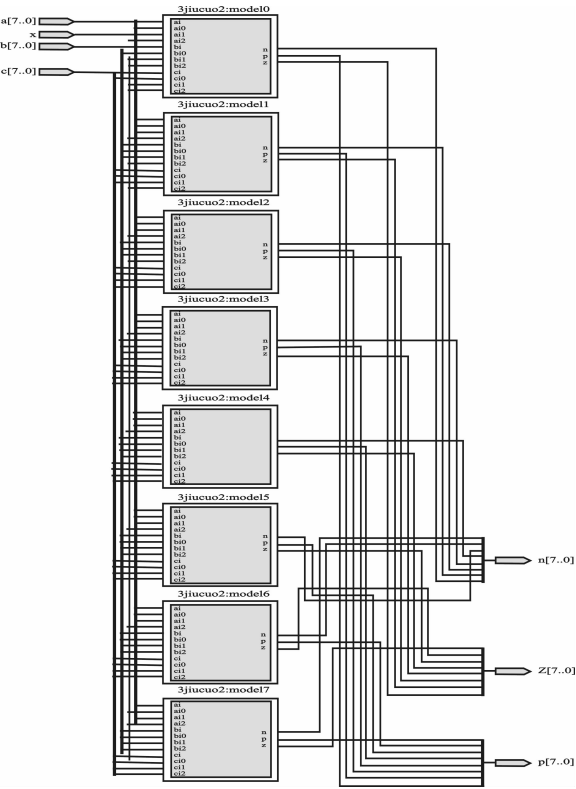


图 5 第三级 $F \geq 0$ (RTL 结构图)

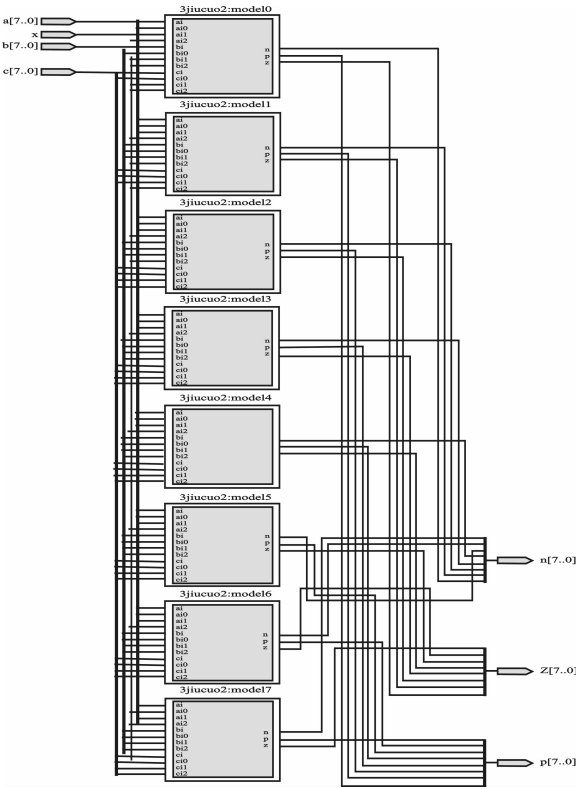


图 6 第三级 $F < 0$ (RTL 结构图)

$0^k(-1)(-1)2(s)$ 中需要修正的情况:

A 0 0 0 0 1 0 1 0
B 0 0 0 0 1 0 1 0
C 0 0 1 1 0 0 0 0
 F_p Z Z N N T Z T Z
 G_n z z n z z p z z

$0^k(-1)0(-1)(s)$ 中需要修正的情况:

A 0 0 0 0 1 1 0 0
B 0 0 0 0 0 1 0 1
C 0 1 0 1 0 0 0 0
 F_p Z N Z N O T Z O
 G_n z n z z z z z p

4.2 仿真实验

以下均是 Quartus II 时序仿真的结果,在仿真结果中当 z 为 1 时,表示该位为 z ; z 为 0 时,表示该位不为 z . p, n 与 z 的原理相同. 经过和理论结果的对照,证明仿真的结果是完全正确的. 在 $F \geq 0$ 时,以上例子的输出结果为 $z^k p z^m n(s)$ 、 $z^k p p^m z(s)$ 、 $z^k p p^m n(s)$ 其中的一种形式,表示需要 1 位纠正,如图 7 所示. 在 $F < 0$ 时,以上例子的输出结果为 $z^k n z^m p(s)$ 的形式,表示需要 1 位纠正,如图 8 所示.

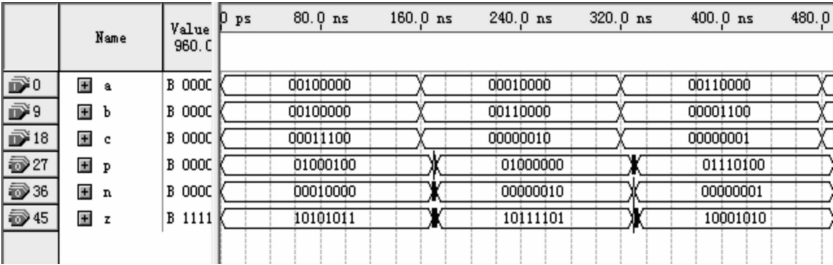


图 7 $F \geq 0$ 实验数据的仿真结果

	Name	Value	ps	80.0 ns	160.0 ns	240.0 ns	320.0 ns	400.0 ns	480.0
		960.0							
0	a	B 000C		00011100	X	00001010	X	00001100	X
9	b	B 000C		00000101	X	00001010	X	00000101	X
18	c	B 000C		01100000	X	00110000	X	01010000	X
27	n	B 001C		01000000	X	00100000	X	01000000	X
36	p	B 000C		00000001	X	00000100	X	00000001	X
45	z	B 1101		10111110	X	11011011	X	10111110	X

图 8 $F<0$ 实验数据的仿真结果

5 结束语

本文完成了三操作数前导 1 预测算法纠错模块预编码和预编码的硬件结构设计,通过编写可综合的代码、设计编码逻辑规则,生成子模块从而实现了三操作数的前导 1 预测算法纠错模块的整体结构.使用 VHDL 语言作为设计输入,仿真数据均在 Quartus ii 下进行综合得到.仿真结果表明,设计的三操作数前导 1 预测算法纠错模块可以快捷正确地完成其功能,达到了预期的目标.

参考文献:

[1] Javuer D Bruguera, Tomas Lang. Leading-one prediction with concurrent position correction[J]. IEEE Transactions on Computers, 1999, 48(10): 1083-1097.

[2] 梅小露. 浮点乘加部件中三操作数前导 1 预测算法的设计[J]. 微电子学与计算机, 2005, 22(12): 41-45.

[3] 李星. 前导 1 预测算法的设计与实现[J]. 计算机科学, 2013, 40(4): 31-33.

[4] Vojin G Oklobdzija. An algorithmic and novel design of a leading zero detector circuit ; Comparison with

logic synthesis[J]. IEEE Transactions on VLSI Systems, 1994, 2(1): 124-128.

[5] 张峰. 128 位浮点乘加融合部件的研究与实现[D]. 长沙:国防科学技术大学, 2007.

[6] 罗淑贞, 富坤, 高艳, 等. 基于 FPGA 的三操作数前导 1 预测算法的设计与性能分析[J]. 微电子学与计算机, 2015, 32(5): 16-20.

[7] Ji Rong, Ling Zhiqiang. Comments on "leading-one prediction with concurrent position correction"[J]. IEEE Transactions on Computers, 2009, 58(12): 1726-1727.

[8] P M Farmwald. On the design of high performance digital arithmetic units[D]. Palo Alto: Stanford University, 1981.

[9] N T Quach. Reducing the latency of floating-point arithmetic operations[D]. Palo Alto: Stanford University, 1994: 1-80.

作者简介:

王京京 女, (1990-), 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, 讲师. 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 9 页)

[5] Du G, Li M, Lu Z, et al. An analytical model for worst-case reorder buffer size of multi-path minimal routing NoCs [C] // Networks-on-Chip (NoCS), 2014 Eighth IEEE/ACM International Symposium on. Ferrara: IEEE, 2014: 49-56.

作者简介:

杜高明 男, (1977-), 博士, 副研究员. 研究方向为多核集成电路体系结构、二维/三维片上网络集成体系结构. E-mail: dugaoming@ hfut. edu. cn.