



## (12) 发明专利

(10) 授权公告号 CN 101699390 B

(45) 授权公告日 2012. 01. 04

(21) 申请号 200910218505. 4

(22) 申请日 2009. 10. 23

(73) 专利权人 西安交通大学

地址 710049 陕西省西安市咸宁西路 28 号

(72) 发明人 邵志标 李凌浩 王丽

(74) 专利代理机构 西安通大专利代理有限责任  
公司 61200

代理人 徐文权

(51) Int. Cl.

G06F 7/575(2006. 01)

审查员 吴媛媛

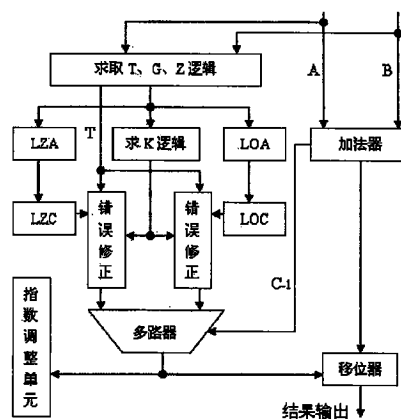
权利要求书 2 页 说明书 8 页 附图 1 页

### (54) 发明名称

用于浮点加法器的自纠错前导 0/1 预测单元

### (57) 摘要

用于浮点加法器的自纠错前导 0/1 预测方法,采用多输入逻辑门以及并行计算相结合的方法实现了输出结果为最终正确结果,不必依赖加法器结果来进行修正;采用了并行计算,如操作数位宽增加,不会影响关键路径长度。能够在计算浮点加法的同时,对计算结果进行规格化处理所需要进行的移位次数以及指数调整信息作出同步的预测,且预测结果不依赖加法器输出而只由预测单元产生,预测结果为无需进一步修正的正确值,预测单元的关键路径不会因为操作数的位宽的加长而变长。



1. 用于浮点加法器的自纠错前导 0/1 预测方法,其特征在于:

1) 采用将前导 0 预测单元和前导 1 预测单元分立的办法,则分别有下列两式来计算数据指示位标记  $f_i$ :

$$f_i = \overline{T_i} \cdot \overline{Z_{i+1}}, i \geq 0 \quad (15)$$

$$f_i = \overline{T_i} \cdot \overline{G_{i+1}}, i \geq 0 \quad (16)$$

其中,式 (15) 为 LZA(Leading Zero Anticipation 的缩写,表示前导 0 预测) 中预测字段求取式,式 (16) 为 LOA(Leading One Anticipation 的缩写,表示前导 1 预测) 中预测字段求取式,其中若  $T_i = 1$ ,表示在不考虑低位进位的情况下,两个二进制操作数  $i$  位上的两个二进制数的和为 1,且不产生向高位的进位,即  $i$  位上两个二进制操作数的值为 1、0 或者 0、1;若  $Z_i = 1$ ,表示在不考虑低位进位的情况下,两个二进制操作数  $i$  位上的两个数的和为 0,且不产生向高位的进位,即  $i$  位上两个二进制操作数的值均为 0;若  $G_i = 1$ ,表示在不考虑低位进位的情况下,两个二进制操作数  $i$  位上的两个数的和为 0,且产生向高位的进位,即  $i$  位上两个二进制操作数的值均为 1,因此在  $i$  位上,两个二进制操作数值的情况只能使  $T_i$ 、 $Z_i$ 、 $G_i$  三个参数中的 1 个参数的值为 1,且至少使一个参数的值为 1,两个分立的 LZA 和 LOA 得到的预测结果,利用来自加法器的和的最高位进位来进行选择:如果最高位产生进位,说明和为正,则需对前导 0 进行预测,选择 LZA 的预测结果;如果最高位不产生进位,说明和为负,则需对前导 1 进行预测,选择 LOA 的预测结果;

2) 高速前导 0、1 预测纠错单元

由于在  $i$  位上,两个二进制操作数值的情况只能让  $T_i$ 、 $Z_i$ 、 $G_i$  三个参数中的 1 个参数的值为 1,且至少使一个参数的值为 1,因此对于该位置用值为 1 的参数标记该位特征——即若  $i$  位上  $T_i$  的值为 1,则此时  $Z_i$ 、 $G_i$  的值均为 0,因此在预测字段的  $i$  位上写 T,所有值为 1 的参数在其相应的位置上构成预测字段,无论对于 LZA 还是 LOA,只有当预测字段中,  $T*GZ*$  或  $T*ZG*$  段后面紧随着 T,才需要对输出的  $n-1$  进行修正,这里  $T*$  表示 T 后面有  $m$  个 T,其中  $m \geq 0$ ;  $Z*$  表示 Z 后面有  $m$  个 Z,其中  $m \geq 0$ ;  $G*$  表示 G 后面有  $m$  个 G,其中  $m \geq 0$ ,为了判断是否需要对输出进行修正,需要对预测字段中的第  $n$  位作出判断,利用 LZC(Leading Zero Count 的缩写,表示前导 0 计数) 和 LOC(Leading One Count 的缩写,表示前导 1 计数) 的输出  $L_i$  对 T 信号进行处理,并通过处理结果进行判断,有:

$$F_i = \sum_{j=0}^i f_j \quad (21)$$

$$L_i = \overline{F_{i-1}} \cdot f_i \quad (22)$$

其中,对于前导 0 预测,有:

$$f_j = \overline{T_j} \cdot \overline{Z_{j+1}}, j \geq 0$$

对于前导 1 预测有:

$$f_j = \overline{T_j} \cdot \overline{G_{j+1}}, j \geq 0$$

$F_i$  为  $f_j$  从第 0 位到第  $i$  位的按位或操作;

$F_i$  具有以下特点:

存在一个确定的  $n$ ,使对于任意的  $i < n$  的  $F_i$  为 0,其余  $F_i$  为 1,即有对于  $j < n$  的任

意  $f_j = 0$ , 这里采用“大端顺序”,  $F_i$  为 LZA/LOA 输出的结果字串;

$L_i$  具有以下特点:

对于确定的  $n$ , 有  $L_n = 1$ , 其余  $L_i = 0$ , 为 LZC/LOC 的未被纠错的输出;

求判断使能信号 EN 的方法;

$$en_i = L_i \cdot T_{i+1} \quad (23)$$

$$EN = \sum en_i \quad (24)$$

$T_{i+1} = A_{i+1} \oplus B_{i+1}$ ,  $A$  和  $B$  为两个加数,  $EN$  为  $en_i$  的按位或结果,  $EN$  作为是否需要预测结果进行修正判断的使能信号;

对于确定的  $n$ , 有  $L_n = 1$ , 这时只有  $T_{n+1} = 1$  时, 才需要对结果是否需要修正进行判断, 即  $en_n = 1$  时才可能需要修正,  $EN$  为  $en$  压缩成的单一使能信号;

若  $EN = 0$  无需修正输出结果, 也就无需判断是否修正;

若  $EN = 1$  时, 需要对结果是否修正进行判断, 判断是否需要输出  $n-1$  做修正, 即已知预测字段的第  $n+1$  位为  $T$ , 那么预测字段的第  $n+1$  位是否存在进位信号  $C_n$ , 若  $T_n = 1$ , 则只需判断进位信号  $C_n$  的取值, 为此, 引入一个新的内部信号  $K$ , 将之定义为:

$$K_0 = 0 \quad (25)$$

$$K_i = T_{i-1}G_i, i \geq 1 \quad (26)$$

若  $K_m = 1$ ,  $m > n$ , 且  $K_i \neq 1 (n < i < m)$ , 则只要存在  $T_i \neq 1 (n < i < m)$  便可判断  $C_n$  的取值必为 0, 对字段  $K$  进行如下处理:

对于前导 0 预测部分, 利用 LZC 产生信号  $F$  对字段  $K$  进行处理:

$$F_i = \sum_{j=0}^i f_j \quad (27)$$

$$K'_i = K_i \cdot F_i, i \geq 0 \quad (28)$$

同法对  $T$  信号进行处理:

$$T'_i = \overline{T_i} \cdot F_i, i \geq 0 \quad (29)$$

以上两式是用来对  $K$  和  $\overline{T}$  信号的第 1 位到第  $n$  位进行清零处理, 则  $K'$  即为对  $K$  字段的 0 到  $n$  位清零后的字段, 对  $K'$  进行处理, 将  $K'$  从第  $m$  位以后全部清零, 0 至第  $m-1$  位全部置 1:

$$Q_i = \sum_{j=0}^i K'_j \quad (30)$$

对  $T'$ 、 $Q$  信号进行按位与操作, 即获得含有目标字段, 且其余部分全置 0 的字段:

$$d_i = T'_i \cdot Q_i, i \geq 0 \quad (31)$$

$$D = EN \cdot \sum d_i \quad (32)$$

若  $D = 1$ , 则不会产生进位信号  $C_n$ , 对于前导 0 预测需要进行修正, 即将  $n$  取代  $n-1$  作为预测输出结果; 相应的对于前导 1 预测, 无需进行修正。

## 用于浮点加法器的自纠错前导 0/1 预测单元

### 技术领域

[0001] 本发明涉及一种前导 0/1 预测方法,具体涉及一种用于浮点加法器的自纠错前导 0/1 预测方法。

### 背景技术

[0002] 文献“G. Zhang, W. Hu, Z. Qi, “Parallel error detection for leading zero anticipation”. Journal of Computer Science and Technology, v 21, n 6, p 901-906, November 2006.”提出了一种并行的不依赖加法器进位信号便可实现纠错的前导 0、1 预测单元纠错技术并予以实现。其主要思想是利用“positive tree”和“negative tree”将“positive encoding”与“negative encoding”的结果进行压缩,最后通过或逻辑来输出正确结果。其特有的树状压缩结构使其“encoding”信号必须通过一系列逻辑门才最终到达输出端,而这种分级的树状结构,其关键路径长度由其级数决定,而级数又由操作数位宽决定。对于位宽较大的操作数,此种技术缺乏优势。

[0003] 文献“O. Mauro, P. Francesco ;S. Simone, V. Giuseppe, “Analysis and implementation of a novel leading zero anticipation algorithm for floating-point arithmetic units”, IEEE Transactions on Circuits and Systems II: Express Briefs, v 54, n 8, p 685-689, August 2007.”中提出的预测方法,其结果有可能出现一位误差,需要利用加法器最高位输出来进一步纠错。

[0004] 文献“D. Giorgos, G. Kostas, M. Christos, N. Dimitris, “Low-power leading-zero counting and anticipation logic for high-speed floating point units”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, v 16, n 7, p 837-850, July 2008.”中的前导 0/1 预测单元,预测一种结果作为输出,而利用“decision logic”来对结果的正确性进行判断。一旦“decision logic”发现输出结果有误,就必须对输出结果进行 1-2 位的移位以得到正确结果。在预测有误的情况下势必影响输出结果的速度。

[0005] 典型前导 0、1 预测单元如下,前导 0 预测单元是指用来确定数据的高位上的 0 直到出现第一个 1 的那部分数据长度的单元,它包括 LZA (Leading Zero Anticipation: 前导 0 预测) 和 LZC (Leading Zero Counting: 前导 0 计数) 两部分;前导 1 预测单元是指用来确定数据的高位上的 1 直到出现第一个 0 的那部分数据长度的单元,它包括 LOA (Leading One Anticipation: 前导 1 预测) 和 LOC (Leading One Counting: 前导 1 计数) 两部分。

[0006] 对于一个确定的二进制数,若其含有  $k$  位的前导 0,则此数据可以被表示成  $0^k 1x^*$  的形式,其中指数  $k$  代表前导 0 的个数,  $x$  代表 0 或者是 1 的任意数,而  $*$  代表  $x$  之后的 0 位或者任意位的  $x$ 。类似的,对于一个确定的二进制数,若其含有  $k$  位的前导 1,则此数据可以被表示成  $1^k 0x^*$  的形式,其中指数  $k$  代表前导 1 的个数,  $x$  代表 0 或者是 1 的任意数,而  $*$  代表  $x$  之后的 0 位或者任意位的  $x$ 。因此对前导 0 检测包括了对第一个非零数字位的确定,或者说包括了对前导的  $k$  位 0 向第  $k+1$  位上的 1 的转换的确定。类似的前导 1 预测也是对数据第一次从 1 转换为相邻的 0 的位置的确定一样。

[0007] 通常两个数相加可以归为三种情况：两个正数相加，两个负数相加，一个正数和一个负数相加。在浮点加法器中，由于两个操作数的位数需要经过对位移位才能进行运算，所以当两个正数相加或者两个负数相加时，不会产生前导 0 或者前导 1，所以无需对此时结果的前导 0 或者前导 1 进行判断。而只有当两个异号的操作数做加法时，才有可能产生大量的前导 0、1。此所谓有效减法 (effective subtraction)。

[0008] 前导 0 出现在进行有效减法（包括同号两操作数相减和异号两操作数相加）操作时，减法结果为正的时候；前导 1 出现在进行有效减法操作时，减法结果为负的时候。前导 0 预测单元通过对浮点加法器的输入信号 A 和 B，在移位、排列、取反之后的每一位进行逻辑运算，再通过对运算结果进行处理，最终实现预测前导 0 的目的。其所需逻辑运算包括：

$$[0009] \quad T = A \oplus B \quad (1)$$

$$[0010] \quad G = A \cdot B \quad (2)$$

$$[0011] \quad Z = A \cdot B \quad (3)$$

[0012] 当逻辑运算结果的开始段的形式为：T\*GZ\* 时（这里，开始字段每一位上的符号 T、G、Z 分别代表该位上  $T = 1$ 、 $G = 1$  或  $Z = 1$ ；而其中第一个 \* 代表有最高位的 T 后面有任意个个数大于等于 0 的 T 重复出现；第二个 \* 代表任意字段），出现前导 0。假设第一个与 T\*GZ\* 形式不匹配的位出现以前有 n 位，那么 A/B 和中将有 n 个或者 (n-1) 个前导 0 出现，这取决于这个不匹配位（第 n+1 位）的形式：若为 T，结果不确定，要看是在 n+1 位上是否有进位信号，但若为 G，则有 n-1 位前导 0。类似的对于结果以 T\*ZG\* 形式开头的，将会产生前导 1。假设第一个与 T\*ZG\* 形式不匹配的位出现以前有 n 位，那么 A/B 和中将有 n 个或者 (n-1) 个前导 1 出现，这取决于这个不匹配位（第 n+1 位）的形式：若为 T，则需考察 n+1 位上的进位信号；若为 Z，则有 n-1 位前导 1。另外，作为补充，对于非有效减法操作，当结果以 Z\* 开头时，也会出现前导 0。类似的，当结果以 G\* 开头时，会出现前导 1。

[0013] 在浮点数的规格化操作中，也会出现对前导 0 的确定的操作。此时，可以看成一个浮点数与一个数字 0 做加减运算，结果以 Z\* 开头。因此，如果一个前导 0 预测单元同时用于减法操作和浮点数的规格化操作，则必须分别对应两种操作，对前导 0 预测结果进行区别。对于减法操作结果以 T\* 开头才能产生前导 0，对于浮点数规格化操作，其结果应该以 Z\* 开头才能产生前导 0。但是对于浮点数的规格化操作，其低位不会向高位产生进位而使预测结果产生误差，故其预测结果无需另做修正。

[0014] (1) 通常情况下前导 0、1 位置的确定

[0015] 对于任意一位的数据是否为所需要确定的第一位的先导数（从数据的高位算起，前导 0 之后的第一个 1 或者先导 1 之后的第一个 0），都可以通过对它和与它左右相邻的三位数据的逻辑计算中确定。由此定义数据指示位标记  $f_i$ ，这里采用高位端起始的顺序：

$$[0016] \quad f_0 = \overline{T}_0 T_1 \quad (4)$$

$$[0017] \quad f_i = T_{i-1} (G_i \overline{Z}_{i+1} + Z_i \overline{G}_{i+1}) + \overline{T}_{i-1} (Z_i \overline{Z}_{i+1} + G_i \overline{G}_{i+1}) \quad (5)$$

[0018] 当第 i 位的  $f_i$  值为 1，并且没有更高位的值为 1 时，则所求的先导数位置便处在第 i 位或者第 i+1 位上。

[0019] 如果前导 0、1 预测分别实现，那么预测单元只需对 i 位和 i+1 位进行计算。对于前导 0 预测单元，有：

$$[0020] \quad f_i^{zeros} = T_i \oplus \overline{Z}_{i+1}, i \geq 0 \quad (6)$$

[0021] 当第  $i$  位的  $f_i$  值为 1, 并且没有更高位的值为 1 时, 则所求的前导 0 后的第一个 1 便出现在第  $i$  位或者第  $i+1$  位上。

[0022] 类似的对于前导 1 的预测, 有:

$$[0023] \quad f_i^{ones} = T_i \oplus \overline{G_{i+1}}, i \geq 0 \quad (7)$$

[0024] 同样, 当第  $i$  位的  $f_i$  值为 1, 并且没有更高位的  $f_i$  的值为 1 时, 则所求的前导 1 后的第一个 0 便出现在第  $i$  位或者第  $i+1$  位上。由此, 如果将前导 0、1 预测单元分开, 那么对于预测算法要简化许多。对于只预测前导 0, 可将求取  $f_i$  的公式简化为:

$$[0025] \quad f_i = \overline{T_i} \cdot \overline{Z_{i+1}}, i \geq 0 \quad (8)$$

[0026] 同理, 对于只考虑预测前导 1, 可将公式变为:

$$[0027] \quad f_i = \overline{T_i} \cdot \overline{G_{i+1}}, i \geq 0 \quad (9)$$

[0028] (2) 前导 0 个数的编码统计

[0029] 有两种基本的统计前导 0 个数的方法, 都是通过对  $f_i$  组成的字段进行处理, 而确定前导 0 个数。其一就是直接对第一个 1 前面 0 的个数进行处理, 另外一个就是使用分级树进行处理。在直接建立单调字段的方法中, 应首先找出第一个 1 所在的位置, 通过如下关系式建立单调段:

$$[0030] \quad F_i = \sum_{j=0}^i f_j \quad (10)$$

$$[0031] \quad L_i = \overline{F_{i-1}} \cdot f_i \quad (11)$$

[0032] 一旦字段建立, 则  $i$  与  $i+1$  的位置便可得出。

[0033] (3) 前导 0、1 预测中的纠错

[0034] 前述的前导 0 预测单元是不精确的: 计数时可能产生一位的误差。典型的前导 0、1 预测单元在进行移位时候对此误差进行消除。一个较快的方法是在进行规格化移位时进行纠错。即在精确移位时对结果进行调整, 这就要求移位器的高位有判断是否存在前导 0 的能力。另外还有一种更快的电路来实现此功能, 它使用了预测控制信号。如果 LZA 预测需要移动  $n$  位, 但是检查到第  $n$  位为 0, 则产生预测控制信号, 使最终移  $n+1$  位, 否则, 则最终移  $n$  位即可。

[0035] 文献中提出了错误纠正同样可以通过对预测位的进位信号来实现。对于前导位  $i$ , 错误指示可表示为:

$$[0036] \quad e_i = L_i T_{i+1} (C_{i+1} \oplus (T_i + (\overline{A_i \oplus T_{i-1}}))) \quad (12)$$

[0037]  $C_{i+1}$  是第  $i+1$  位的进位信号;  $A_i$  是加数位。则全局错误判断信号:

$$[0038] \quad e = \sum_i e_i \quad (13)$$

[0039] 还有文献介绍了另外一种等价的错误指示信号:

$$[0040] \quad e_i = L_i (T_{i-1} \oplus A_i \oplus \overline{C_i}) \quad (14)$$

[0041] 其中  $C_i$  是第  $i$  位的进位信号。可见上述两种方法均需要借助加法器进位信号来产生修正信号。

## 发明内容

[0042] 本发明的目的在于克服上述现有技术的缺点, 提供了一种能够在计算浮点加法的

同时,对计算结果进行规格化处理所需要进行的移位次数以及指数调整信息作出同步的预测,且预测结果不依赖加法器输出而只由预测单元产生,预测结果为无需进一步修正的正确值,预测单元的关键路径不会因为操作数的位宽的加长而变长的用于浮点加法器的自纠错前导 0/1 预测单元。

[0043] 为达到上述目的,本发明采用的技术方案是:

[0044] 1) 采用将前导 0 预测单元和前导 1 预测单元分立的办法,则分别有下列两式来计算数据指示位标记  $f_i$ :

$$[0045] \quad f_i = \bar{T}_i \cdot \bar{Z}_{i+1}, i \geq 0 \quad (15)$$

$$[0046] \quad f_i = \bar{T}_i \cdot \bar{G}_{i+1}, i \geq 0 \quad (16)$$

[0047] 其中,式 (15) 为 LZA 中预测字段求取式,式 (16) 为 LOA 中预测字段求取式,两个分立的 LZA 和 LOA 得到的预测结果,利用来自加法器的和的最高位进位来进行选择:如果最高位产生进位,说明和为正,则需对前导 0 进行预测,选择 LZA 的预测结果;如果最高位不产生进位,说明和为负,则需对前导 1 进行预测,选择 LOA 的预测结果;

[0048] 2) 高速前导 0、1 预测纠错单元

[0049] 无论对于 LAZ 还是 LOA,只有当预测字段中,  $T*GZ*$  或  $T*ZG*$  段后面紧随着 T,才需要对输出的  $n-1$  进行修正,为了判断是否需要对输出进行修正,需要对预测字段中的第  $n$  位作出判断,利用 LZC 和 LOC 的输出  $L_i$  对 T 信号进行处理,并通过处理结果进行判断,有:

$$[0050] \quad F_i = \sum_{j=0}^i f_j \quad (21)$$

$$[0051] \quad L_i = \bar{F}_{i+1} \cdot f_i \quad (22)$$

[0052] 其中,对于前导 0 预测,有:

$$[0053] \quad f_j = \bar{T}_j \cdot \bar{Z}_{j+1}, j \geq 0$$

[0054] 对于前导 1 预测有:

$$[0055] \quad f_j = \bar{T}_j \cdot \bar{G}_{j+1}, j \geq 0$$

[0056]  $F_i$  为  $f_j$  从第 0 位到第  $i$  位的按位或操作;

[0057]  $F_i$  具有以下特点:

[0058] 存在一个确定的  $n$ ,使对于任意的  $i < n$  的  $F_i$  为 0,其余  $F_i$  为 1,即有对于  $j < n$  的任意  $f_j = 0$ ,这里采用“大端顺序”, $F_i$  为 LZA/LOA 输出的结果字符串;

[0059]  $L_i$  具有以下特点:

[0060] 对于确定的  $n$ ,有  $L_n = 1$ ,其余  $L_i = 0$ ,为 LZC/LOC 的未被纠错的输出;

[0061] 求判断使能信号 EN 的方法;

$$[0062] \quad en_i = L_i \cdot T_{i+1} \quad (23)$$

$$[0063] \quad EN = \sum en_i \quad (24)$$

[0064]  $T_{i+1} = A_{i+1} \oplus B_{i+1}$ , A 和 B 为两个加数,EN 为  $en_i$  的按位或结果,EN 作为是否需要对预测结果进行修正判断的使能信号;

[0065] 对于确定的  $n$ ,有  $L_n = 1$ ,这时只有  $T_{n+1} = 1$  时,才需要对结果是否需要修正进行判断,即  $en_n = 1$  时才可能需要修正,EN 为  $en$  压缩成的单一使能信号;

[0066] 若  $EN = 0$  无需修正输出结果,也就无需判断是否修正;

[0067] 若  $EN = 1$  时,需要对结果是否修正进行判断,判断是否需要对输出  $n-1$  做修正,即

已知预测字段的第  $n+1$  位为  $T$ , 那么预测字段的第  $n+1$  位是否存在进位信号  $C_n$ , 若  $T_n = 1$ , 则只需判断进位信号  $C_n$  的取值, 为此, 引入一个新的内部信号  $K$ , 将之定义为:

$$[0068] \quad K_0 = 0 \quad (25)$$

$$[0069] \quad K_i = T_{i-1}G_i, i \geq 1 \quad (26)$$

[0070] 若  $K_m = 1$ ,  $m > n$ , 且  $K_i \neq 1 (n < i < m)$ , 则只要存在  $T_i \neq 1 (n < i < m)$  便可判断  $C_n$  的取值必为 0, 对字段  $K$  进行如下处理:

[0071] 对于前导 0 预测部分, 利用 LZC 产生信号  $F$  对字段  $K$  进行处理:

$$[0072] \quad F_i = \sum_{j=0}^i f_j \quad (27)$$

$$[0073] \quad K'_i = K_i \cdot F_i, i \geq 0 \quad (28)$$

[0074] 同法对  $T$  信号进行处理:

$$[0075] \quad T'_i = \bar{T}_i \cdot F_i, i \geq 0 \quad (29)$$

[0076] 以上两式是用来对  $K$  和  $\bar{T}$  信号的第 1 位到第  $n$  位进行清零处理, 则  $K'$  即为对  $K$  字段的 0 到  $n$  位清零后的字段, 对  $K'$  进行处理, 将  $K'$  从第  $m$  位以后全部清零, 0 至第  $m-1$  位全部置 1:

$$[0077] \quad Q_i = \sum_{j=0}^i K'_j \quad (30)$$

[0078] 对  $T'$ 、 $Q$  信号进行按位与操作, 即获得含有目标字段, 且其余部分全置 0 的字段:

$$[0079] \quad d_i = T'_i \cdot Q_i, i \geq 0 \quad (31)$$

$$[0080] \quad D = EN \cdot \sum d_i \quad (32)$$

[0081] 若  $D = 1$ , 则不会产生进位信号  $C_n$ , 对于前导 0 预测需要进行修正, 即将  $n$  取代  $n-1$  作为预测输出结果; 相应的对于前导 1 预测, 无需进行修正。

[0082] 本发明采用多输入逻辑门以及并行计算相结合的方法实现了输出结果为最终正确结果, 不必依赖加法器结果来进行修正; 采用了并行计算, 如操作数位宽增加, 不会影响关键路径长度。

## 附图说明

[0083] 图 1 是本发明的整体结构图;

[0084] 图 2 是前导 0 纠错逻辑结构图。

## 具体实施方式

[0085] 下面结合附图对本发明作进一步详细说明。

[0086] 参见图 1, 用于浮点加法器的自纠错前导 0/1 预测单元, 对于两个特定数的加法操作, 可通过对其进行按位处理得出  $T$ 、 $G$ 、 $Z$  信号, 再对  $T$ 、 $G$ 、 $Z$  三个信号进行处理便可实现前导 0、1 预测。由于采用前导 0、1 预测单元相结合的办法, 必然使硬件结构复杂, 关键路径变长, 不利于提高速度。所以本发明的设计中采用将前导 0 预测单元和前导 1 预测单元分立的办法, 则分别有下列两式来计算  $f_i$ :

$$[0087] \quad f_i = \bar{T}_i \cdot \bar{Z}_{i+1}, i \geq 0 \quad (15)$$

$$[0088] \quad f_i = \bar{T}_i \cdot \bar{G}_{i+1}, i \geq 0 \quad (16)$$



[0089] 其中,式 (15) 为 LZA 中预测字段求取式,式 (16) 为 LOA 中预测字段求取式。两个分立的 LZA 和 LOA 得到的预测结果,可以利用来自加法器的和的最高位进位来进行选择:如果最高位产生进位,说明和为正,则需对前导 0 进行预测,选择 LZA 的预测结果;如果最高位不产生进位,说明和为负,则需对前导 1 进行预测,选择 LOA 的预测结果。

[0090] (1) 预测误差产生原因分析

[0091] 首先分析一下 LZA 产生误差的原因。对于 LZA 的预测,若结果中存在前导 0,则预测字段开始的形式必然为  $T*GZ^*$ 。假设  $T*GZ^*$  有  $n$  位,且  $C_i$  为第  $i$  位的进位信号。当  $C_{n-1} = 1$  时,应该将加法运算结果左移  $n-1$  位,当  $C_{n-1} = 0$  时,应该将加法运算结果左移  $n$  位。下面分析  $C_{n-1}$  信号的取值。可见,紧接着开始字段只有如下两种情况:

$$[0092] \quad T*GZ^* \begin{cases} GX^* \\ TX^* \end{cases}$$

[0093] 若  $T*GZ^*$  后为  $G$ ,则必然向前一位 ( $n-1$  位) 产生进位;若  $T*GZ^*$  后为  $T$ ,则要看其后是否有进位信号传递过来,和  $T$  位本身求和向前一位 ( $n-1$  位) 产生进位。由此,得出最终移位信号的取值应条件:

[0094] 当最终移位信号取  $n$  时应满足:

$$[0095] \quad T_n \cdot \bar{C}_n = 1 \quad (17)$$

[0096] 当最终移位信号取  $n-1$  时应满足:

$$[0097] \quad G_n + T_n \cdot C_n = 1 \quad (18)$$

[0098] 对于 LOA 的预测,若结果中存在前导 1,则预测字段开始的形式必然为  $T*ZG^*$ 。同样假设  $T*ZG^*$  有  $n$  位,且  $C_i$  为第  $i$  位的进位信号。当  $C_{n-1} = 1$  时,应该将加法运算结果左移  $n$  位,当  $C_{n-1} = 0$  时,应该将加法运算结果左移  $n-1$  位。接下来仍然分析  $C_{n-1}$  信号的取值。不难看出,紧接着开始字段只有如下所示的两种情况:

$$[0099] \quad T*ZG^* \begin{cases} ZX^* \\ TX^* \end{cases}$$

[0100] 若  $T*ZG^*$  后为  $Z$ ,则必然不会向前一位 ( $n-1$  位) 产生进位;若  $T*ZG^*$  后为  $T$ ,则要看其后是否有进位信号传递过来,和  $T$  位本身求和向前一位 ( $n-1$  位) 产生进位。由此,得出最终移位信号的取值应条件:

[0101] 当最终移位信号取  $n$  时应满足:

$$[0102] \quad T_n \cdot C_n = 1 \quad (19)$$

[0103] 当最终移位信号取  $n-1$  时应满足:

$$[0104] \quad Z_n + T_n \cdot \bar{C}_n = 1 \quad (20)$$

[0105] 综上所述,在随机情况下,当 LZA 中预测字段的第  $n+1$  位取  $G_n$ 、 $T_n$  概率相同,且  $C_n$  取值 1、0 的概率也相同时,移位信号取  $n-1$  的概率为 75%,类似的对于 LOA,预测字段的第  $n+1$  位取  $Z_n$ 、 $T_n$  概率相同,且  $C_n$  取值 1、0 的概率也相同时,移位信号取  $n-1$  的概率为 75%,所以这里选择  $n-1$  作为预测单元的输出,并通过  $T_n$ 、 $C_n$  对预测结果进行修正。

[0106] 由于此种修正方案需要利用加法器进位信号  $C_n$  对预测结果进行修正,使用过程中

必须让前导 0、1 预测单元等待加法器输出结果,不利于速度提高,于是下面本发明提出一种在前导 0、1 预测单元内部对修正信号加以判断的方案。

[0107] (2) 高速前导 0、1 预测纠错单元的提出

[0108] 由上,无论对于 LAZ 还是 LOA,只有当预测字段中,  $T*GZ*$  或  $T*ZG*$  段后面紧随着 T,才需要对输出的  $n-1$  进行修正。则为了判断是否需要对输出进行修正,需要对预测字段中的第  $n$  位作出判断。利用 LZC 和 LOC 的输出  $L_i$  对 T 信号进行处理,并通过处理结果进行判断,有:

$$[0109] \quad F_i = \sum_{j=0}^i f_j \quad (21)$$

$$[0110] \quad L_i = \bar{F}_{i-1} \cdot f_i \quad (22)$$

[0111] 求判断使能信号 EN 的方法;

$$[0112] \quad en_i = L_i \cdot T_{i+1} \quad (23)$$

$$[0113] \quad EN = \sum en_i \quad (24)$$

[0114] 当  $EN = 1$  时,需要对结果是否修正进行判断;当  $EN = 0$  是无需修正输出结果,也就无需判断是否修正。

[0115] 若需要判断是否需要对输出  $n-1$  做修正,即已知预测字段的第  $n+1$  位为 T,关键在于预测字段的第  $n+1$  位是否存在进位信号  $C_n$ 。若  $T_n = 1$ ,则只需判断进位信号  $C_n$  的取值。为此,本发明引入一个新的内部信号 K,将之定义为:

$$[0116] \quad K_0 = 0 \quad (25)$$

$$[0117] \quad K_i = T_{i-1}G_i, i \geq 1 \quad (26)$$

[0118] 若  $K_m = 1$ ,  $m > n$ , 且  $K_i \neq 1 (n < i < m)$ , 则只要存在  $T_i \neq 1 (n < i < m)$  便可判断  $C_n$  的取值必为 0。可对字段 K 进行如下处理:

[0119] 对于前导 0 预测部分,利用 LZC 产生信号 F 对字段 K 进行处理:

$$[0120] \quad F_i = \sum_{j=0}^i f_j \quad (27)$$

$$[0121] \quad K'_i = K_i \cdot F_i, i \geq 0 \quad (28)$$

[0122] 同法对 T 信号进行处理:

$$[0123] \quad T'_i = \bar{T}_i \cdot F_i, i \geq 0 \quad (29)$$

[0124] 以上两式是用来对 K 和  $\bar{T}$  信号的第 1 位到第  $n$  位进行清零处理。则  $K'$  即为对 K 字段的 0 到  $n$  位清零后的字段。接下来对  $K'$  进行处理,将  $K'$  从第  $m$  位以后全部清零,0 至第  $m-1$  位全部置 1:

$$[0125] \quad Q_i = \sum_{j=0}^i K'_j \quad (30)$$

[0126] 如此,对  $T'$ 、Q 信号进行按位与操作,即获得含有目标字段,且其余部分全置 0 的字段:

$$[0127] \quad d_i = T'_i \cdot Q_i, i \geq 0 \quad (31)$$

$$[0128] \quad D = EN \cdot \sum d_i \quad (32)$$

[0129] 若  $D = 1$ ,则不会产生进位信号  $C_n$ ,对于前导 0 预测需要进行修正;相应的对于前导 1 预测,无需进行修正。

[0130] 本发明设计的自纠错前导 0、1 预测单元整体结构如附图 1 所示。在前导 0、1 预测单元工作时,首先通过“求取 T、G、Z 逻辑”求取被操作数的 T、G、Z 的值,并将之分别送入 LZA、LOA、求 K 逻辑以及错误修正单元。再由错误修正单元将 T、K 以及 LZC 或 LOC 产生的结果信号 L、F 进行处理,得到纠错信息对 LZC 或 LOC 的输出结果进行修正,以得到正确的移位信息。通过加法器进位信号选择相应的前导 0 或前导 1 预测结果,将之送入指数调整单元和移位器对浮点加法器结果进行规格化处理。

[0131] 以前导 0 预测单元相对应的错误修正处理单元为例,对错误纠正单元内部结构加以说明,如附图 2 所示。其中 L、F 信号均来自 LZC 部件,T、K 信号来自各自的产生逻辑。“求和”是指将输入信号利用多输入或门按位求和(即按位进行逻辑“或”操作),以得到 EN 信号和 D 信号。乘积代表逻辑“与”操作,纠错移位逻辑是通过纠错移位信号将原有前导 0 预测但愿输出的  $n-1$  修正为  $n$ ,即将前导 0 预测单元产生的结果字符串左移一位。相应的,对于前导 1 预测单元的纠错单元,D 输入信号取反即可实现对前导 1 预测单元的纠错。

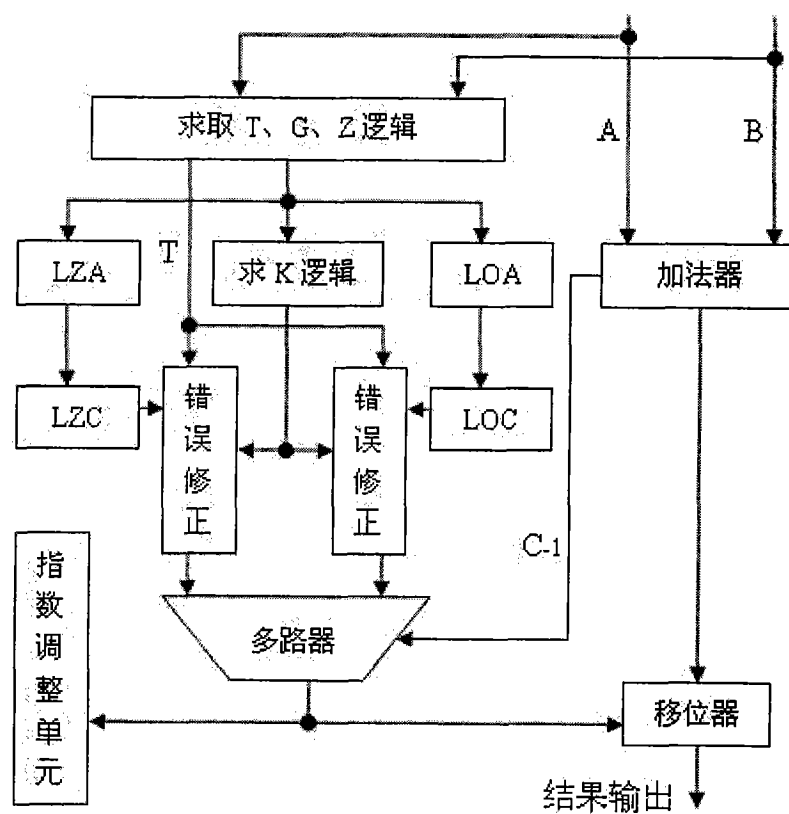


图 1

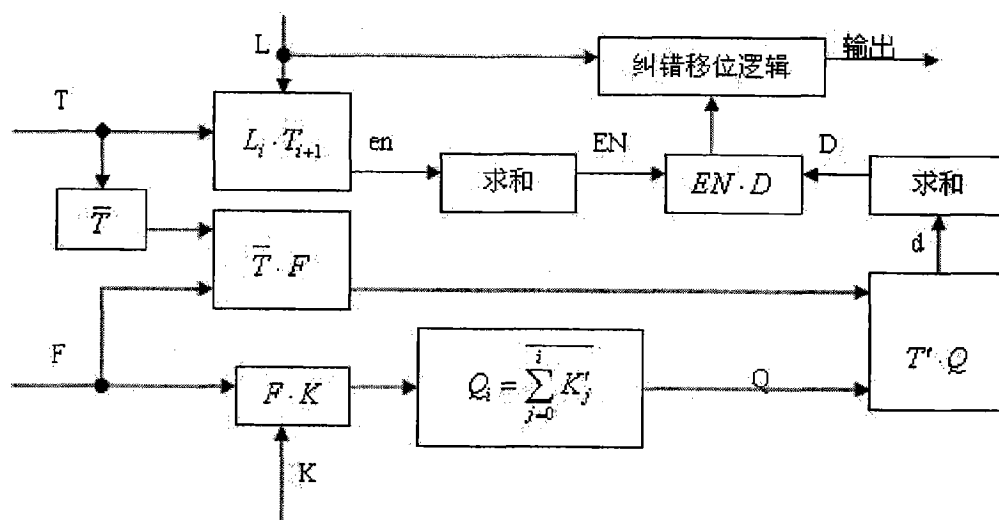


图 2