中山大学学报（自然科学版）
第31卷 第3期 ACTA SCIENTIARUM NATURALIUM Vol.31 №3
1992年 UNIVERSITATIS SUNYATSENI 1992

# 二元树结构的容错分析

姜丽帆
（计算机科学系）

**摘 要** 分析几种二元树结构的容错方法，给出了容错的重构规则.

**关键词** 二元树，并行处理，容错，重构

## 1 引 言

二元树结构是一种使用十分普遍的并行处理结构，互连方式简单而且规则，便于VLSI设计制造。为了在集成度增大时提高二元树结构的产量和可靠性，必须考虑它的容错问题。

在$N=2^k-1$个处理器与$k$层完全二元树的$N$个结点之间建立一一对应关系，完全二元树中有边相连的结点所对应的处理器之间连接一条数据传输线。这样建立起来的结构我们称为$N$个处理器的二元树结构。把处理器简称为结点，数据传输线简称为连接线。采用VLSI技术，能够集成在单一芯片中的结点数越来越多，结点和连接线失效更加难以避免。如果在一些结点和连接线失效时芯片仍旧可以使用，芯片生产率将会大大提高，芯片的可靠性也可有所增加。因为结点具有许多处理功能，而连接线只用于传输数据，故结点一般比连接线复杂得多，失效概率也大得多。又因为连接线的失效可以看成是所连接点之一失效，故我们以下只考虑结点失效的情况。

二元树结构的容错问题已有过许多研究[1～3]。容错方法一般都是采用引入冗余结点和连接线的方式。通过冗余结点和连接线将整个二元树重新构造（也称为重构）来避开失效结点。重构之后的二元树结构若与原结构相同称为非降级重构，若与原结构不同则称为降级重构。

过去所获得的结果基本上属于降级重构，而本文着重讨论非降级重构。并提出若干种重构方法。其中重点讨论在应用中最具有实际意义的单结点失效的情形，并给出了具体的计算规则。

## 2 单结点失效时的容错方法

文[1]中证明了单结点失效时降级重构的容错方法。这里提出一种非降级重构方

本文1991年12月20日收到

案。为了在单个结点失效时实现非降级重构，我们必须至少加一个冗余结点。

我们采用如下方法来构造一棵单结点失效时的容错二元树结构：首先引入一个冗余结点，并把它与根结点及其儿子相连。然后，把所有结点与其孙子结点利用一条冗余边相连（图1）。把除冗余结点之外的所有结点这样连接：若一个结点的两个子树都不包

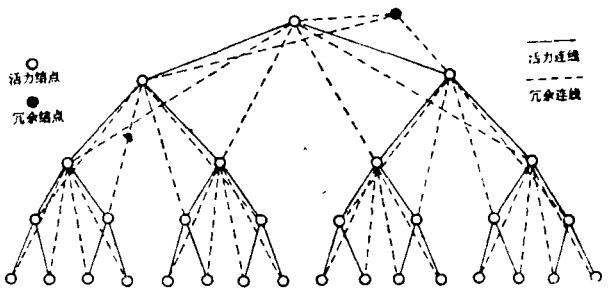


图1 $N=2^5-1$个结点的单结点出错的容错二元树结构

Fig.1 Faulf—tolerance in binary tree architecture with one failed node for $N=2^5-1$ of nodes

含失效结点，则按原有方式连接（即通过活动线连接）；若其中一个儿子是失效结点，则这个结点应连至该失效儿子的两个儿子；若该结点的两个儿子皆有效而有一子树中的一个结点失效，则该结点连到该失效子树的根结点和该根结点的有效儿子结点（在此假定'根结点的两个儿子结点中有一个失效'，否则总可把"该结点的两个儿子皆有效"的假设推移到下层直至' '所述情况出现为止）。若根失效，则冗余结点与根结点的两个儿子相连；若根有效而一个子树有失效结点，则冗余结点与根结点及其无失效结点的子树的根相连；失效结点不与任何结点相连。与此同时，断开该失效结点与其两个子结点和父结点的活动连线；断开该失效结点的兄弟结点与其父结点的活动连线，断开其父结点的兄弟结点与其祖父结点的活动连线。依次类推直到最高层。这就是该容错二元树结构的重构方法。如果采用H-树的构造方法，该结构易于VLSI实现。N个结点的上述结构的冗余边数等于$N$。由于单个处理器失效概率大于多个处理器同时失效概率，因此，这种结构是十分有用的。

下面给出对一般情形按上述重构方法重构时用以编程的规则。设二元树结构共有$n$层，则第$\alpha$层的结点自左至右可用符号$a_{\alpha\beta_\alpha}$表示，其中$\alpha=1,2,\cdots,n$，$\beta_\alpha=1,2,\cdots 2^{\alpha-1}$。假定结点$a_{ki}$($k\epsilon(1,2,\cdots,N)$，$i\epsilon(1,2,\cdots 2^{k-1})$)失效。则重构规律可表示如下：

1）建立递推关系

$$\begin{cases} j_0=i \\ j_\mu=\begin{cases} \dfrac{j_{\mu-1}}{2} & \text{当}j_{\mu-1}\text{是偶数} \\ \dfrac{j_{\mu-1}+1}{2} & \text{当}j_{\mu-1}\text{是奇数} \end{cases} \end{cases} \qquad \mu=1,2,\cdots,k-1 \qquad (1)$$

2）建立冗余连接规律

$$a_{k-1,j_1}\cdots\cdots\begin{cases} a_{k+1,2i-1} \\ a_{k+1,2i} \end{cases} \quad \text{这里}(k+1)\leqslant n \qquad (2)$$

$$a_{k-\mu,j\mu}\cdots\cdots\begin{cases}a_{k-\mu+2,j_{\mu-2}+1} & \text{当}j_{\mu-2}\text{是奇数}\\ a_{k-\mu+2,j_{\mu-2}-1} & \text{当}j_{\mu-2}\text{是偶数}\end{cases}\qquad \mu=2,\cdots,k-1 \tag{3}$$

$$a_0\cdots\cdots\begin{cases}a_{1,1}\\ a_{2,i_2+1} & \text{当}j_2\text{为奇数}\\ a_{2,i_2+1} & \text{当}j_2\text{为偶数}\end{cases} \tag{4}$$

其中，……表示用冗余线连接，$a_0$为冗余结点。

3）建立活动线断开规律

$$a_{k,j}\nrightarrow\begin{cases}a_{k+1\ 2i-1}\\ \\ a_{k+1\ 2i}\end{cases}\qquad \text{这里}(k+1)\leqslant n \tag{5}$$

$$a_{k-1,j_1}\nrightarrow\begin{cases}a_{k1i}\\ a_{k,i+1} & \text{当}i\text{为奇数}\\ a_{k,i-1} & \text{当}i\text{为偶数}\end{cases} \tag{6}$$

$$a_{k-\mu,j\mu}\nrightarrow\begin{cases}a_{k-\mu+1,i_{\mu-1}-1} & \text{当}j_{\mu-1}\text{为偶数}\\ a_{k-\mu+1,j_{\mu-1}-1} & \text{当}j_{\mu-1}\text{为奇数}\end{cases}\qquad \mu=2,\cdots\cdots,k-1 \tag{7}$$

其中$\nrightarrow$表示活动线断开。

从上述的规则可见：①当$a_{k,i}$结点失效时，需连接$k+2$条冗余线，断开$k+2$条活动线。即连接及断开线的数目仅与失效结点所在层的层数有关。②按上述重构规则进行重构后仍保持原来的树结构。事实上，由2）及3）知道，第$k$层与以下各层的状态除了用$a_{k-1,j_1}$代替了原来的$a_{k,i}$之外（见(2)、(5)、(6)三式）一律保持原来的状态，也就是说第$k$层及其以下各层的结构与原结构一样。

对$k-1$层及其以上各层的结构来说，由(3)及(7)式知道，$a_{k-2,j_2}$代替了原来的$a_{k-1,j_1}$，并与第$k$层的两个结点相连（其中一个由(3)式决定，用冗余线相连，另一个由(7)式决定，用活动线相连），类似地，$a_{k-3,j_3}$代替了原来$a_{k-2,j_2}$，…，$a_{1,1}$代替了原来的$a_{2,j_{k-2}}$，且皆与下一层的两个结点相连，另由(4)式知道冗余结点$a_0$代替了$a_{1,1}$，且与下一层的两个结点用冗余线相连。与此同时，(3)、(4)式所没有涉及的结点都保持原来的连接结构。

综上所述，重构之后，从层数、结点数及连接规律上仍保持与原来的树结构一样。

## 3　多个处理器失效时的容错方法

当每一层最多只出现一个失效结点时，〔1〕给出了一种容错结构，它利用每层结点的平移法来实现每层的容错（见图2）。

这一二元树结构是按如下方法构造的：每层在右部加入一个冗余结点。每个结点都与其左部相邻结点的两个儿子用两条冗余边相连，与其右部相邻结点的左儿子用一条冗余边相连（若右部相邻结点是冗余结点，则与该冗余结点的子冗余结点相连）。每一冗

余结点与下一层的最右3个结点（包括冗余结点）用冗余边相连。

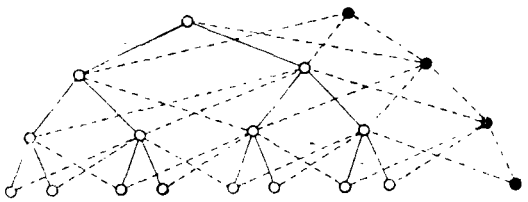

图2 每层可容纳一个失效结点的二元树结构

Fig.2 Binary tree architecture with only one failed node contained at every level

对于这种二元树结构，重构方法如下：从左从下往上往右进行遍历。每个被遍历的结点都将下一层的与之相连的两个结点作为儿子，这两个结点都是无失效的，还没有被上一层作为儿子、位于与正被遍历的结点相连的下一层结点的尽可能左边的两个结点。在构造方法中，每个结点与左部相邻结点的两个儿子增加冗余边是为了在同层左部有结点失效时重构用；每个结点与右边相邻结点的左儿子用冗余边相连是为了下一层的儿子结点或其左部结点有一失效结点时重构用。

根据上述观察，可以给出每一层可容纳$x$个出错结点的容错二元树结构。每层在右边增加$x$个冗余结点。每个非冗余结点都与左邻$x$个结点的儿子用冗余边相连，与其右邻$\left[\frac{x}{2}\right]+1$个结点的儿子用冗余边相连。每层可容纳2个出错结点的容错二元树结构（图3）。

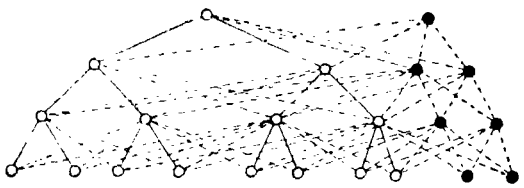

图3 每层可容纳2个出错结点的容错二元树结构

Fig.3 Fault—tolerance in binary tree architecture with two failed nodes contained at every level

如果我们在任何两个兄弟结点之间加入一个冗余结点，并使它与这兄弟结点的父结点及儿子结点用冗余边相连，便可实现任何两个兄弟结点中有一个出错结点时仍可实现不降级重构。$N=2^4-1$个结点的容错二元树结构如图4所示，其实此结构对于具有爷爷的孙子结点中任何四个结点中有两个失效仍旧可以重构。

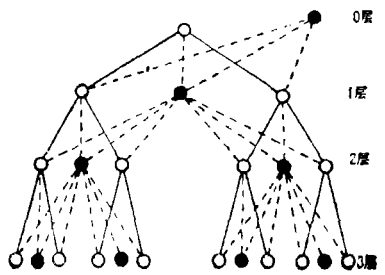


图4 多结点容错二元树结构活动结点数$N=2^4-1$

冗余结点数$=\frac{N}{2}+\frac{1}{2}$

Fig.4 Fault—tolerance in binary tree architecture of multi-node, with $N=2^4-1$ of $\frac{N}{2}+\frac{1}{2}$ of spare nodes

图4中的构造方法与[1]中的相似，但它更便于VLSI设计，特别是H-树的设计。在$H$-树的设计方法中高层结点相隔较远，低层结点才距离近，因此，如果把上述介绍的几何方法混合使用，如高层用单结点容错，中层用单层容错，低层用多结点容错，则可以在减少冗余结点的个数的同时保持容错特性基本不变。

## 参考文献


1 Aghavendra RCS, Avigienis A, Ercegovac M. IEEE T C, 1984, C—33:568～572

2 Gordon D,Koren I, Silberman G M. IEEE Transactions on Computers, 1984, C-33(1): 104～108

3 Hassan A S M, Agarwal V K. IEEE Transactions on Computers, 1986, c—35 (4): 356～361


# Fault-tolerant Analysis for the Binary Tree Configuration

*Jiang Lifan**

**Abstract** Several fault-tolerant methods for the binary tree architecture are proposed in this paper. The corresponding raconfiguration algorithms are given.

**Keywords** binary tree, parallel processing, fault-tolerant, reconfiguration

* Department of Computer Science