研究与开发

Benes网络容错交换方法及其应用

秦梦远[1,2]，刘宏伟[1]，郝沁汾[1]
（1. 中国科学院计算技术研究所，北京 100095；
2. 中国科学院大学，北京 100049）

摘　要：Benes网络能够实现高交换基数下较低的构建成本和理论高吞吐量，但现有Benes网络对应的路由求解算法不保证网络内部开关单元失效时的可重排无阻塞特性。提出一种非平衡Benes网络，其在特定条件下拥有和Benes网络相同的可重排无阻塞特性。提出一种由Benes网络裁剪获得非平衡Benes网络的方法，可以此法屏蔽Benes网络中发生故障的开关单元，实现网络的容错交换。当用于处理开关阵列生产良率问题时，该方案可重排无阻塞交换规模比传统容错方案平均提升56.05%，最高提升93.75%；当用于处理开关阵列的高可靠容错交换时，在容许最多3个开关单元出现故障前提下，比传统容错方案交换规模提升12.5%至21.9%。提出针对非平衡Benes网络的快速路由求解算法，并使用FPGA验证，验证结果表明，该求解算法不会成为交换系统的性能瓶颈。基于裁剪法，研究同样实现了Benes网络的可控局部重构，使其支持像Crossbar网络那样以局部重构为主的使用方式。

关键词：Benes网络；无阻塞网络；容错；路由求解；硬件加速器
中图分类号：TP302.2；TN256
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2026111

Fault-tolerant switching method of Benes network and its applications

Qin Mengyuan[1,2], Liu Hongwei[1], Hao Qinfen[1]
1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100095, China
2. University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: Benes network can achieve low construction costs and high throughput on high-radix-switching scenarios. However, existing route-resolving algorithms corresponding to Benes network do not guarantee rearrangeable non-blocking (RNB) switching when suffering failure of some of its internal switch units. An unbalanced Benes network was proposed and its RNB switching trait could be guaranteed in certain conditions. A trimming method to convert a Benes network to unbalanced Benes network was proposed, through which failed switching units could be blocked,

收稿日期：2025−07−24；**修回日期：**2025−11−03
通信作者：郝沁汾，haoqinfen@ict.ac.cn
基金项目：国家重点研发计划项目（No.2022YFB4401501）；江苏省重点研发计划项目（No.BE2023006-4）
Foundation Items: The National Key Research and Development Program of China(No.2022YFB4401501), The Key Research and Development Program of Jiangsu Province(No.BE2023006-4)

and fault-tolerant switching was achieved. When solving yield rate problems of the switch array, the trimming method demonstrated advantages over the conventional method on RNB switching radix, with 56.05% in average and 93.75% in maximum. When solving high reliability switching problems of the switch array, the trimming method had 12.5% to 21.9% higher switching radix when tolerating maximum 3 faulty switch units. Moreover, a fast route-resolving method for unbalanced Benes network was proposed and verified via field programmable gate array (FPGA), and the result shows that it doesn't become the bottleneck of the system. Furthermore, through the trimming method, controllable partial reconfiguration of Benes network can be achieved, so Crossbar-like switching form based on partial reconfiguration is also supported by Benes network.

Key words: Benes network, non-blocking network, fault-tolerance, route-resolving, hardware accelerator

0 引言

Benes网络[1]是一种紧凑的Clos类网络[2]，能够通过重构内部开关节点的状态，实现可重排无阻塞特性，且信号完成交换所需经过的开关节点数为相同互连规模下最少。Benes网络可重排无阻塞特性使其被广泛应用于信号交换[3-5]、密码学领域[6]和片上网络[7]。有关于AI加速器的研究也将Benes网络用于稀疏矩阵的压缩与使用[8]。

Benes网络的路由求解算法相比Clos类网络时间复杂度更低且易于被硬件加速。现有Clos类网络路由求解算法的研究成果表明，其时间复杂度普遍在$O(N\sqrt{N})$量级[9]，最好的时间复杂度也仅接近$O(\log N)$[10]。相比之下，Benes网络的单线程时间复杂度固定为$O(N\log N)$[11]，易于并行计算，已有相关研究的并行时间复杂度为$O(\log N)$至$O(N)$[12-13]，且硬件加速器普遍结构较为简单，单次重构求解耗时在数十至数百纳秒之间[14-17]。

但Benes网络内部完全没有冗余，这导致任何开关单元的故障都会使经过其中的链路无法工作，且没有备用链路，进而导致其当前规模下的可重排无阻塞特性被破坏。为规避此类故障，提高开关阵列的良品率、降低生产成本，传统的容错思路集中于使用冗余的开关构建备用链路，如双路网络互相备份[18]、冗余链路[19-20]、扩张型Benes网络[21-23]等，当侦测到主链路存在故障导致阻塞时，会尝试切换到备用链路进行通信以避免阻塞。此时，通常会有至少一半的开关节点作为冗余节点用于提供备用链路，资源浪费严重，且无法给出支持的最大可重排无阻塞交换规模上限。文献[24]讨论了Benes网络在开关出现特定故障状态时的容错能力，但仅限于开关无法完成状态切换，仅能保持“平行”或“交叉”状态的特殊故障，而没有讨论开关完全故障无法传输信号这种普遍故障，实用价值较为有限。

本文提出一种非平衡Benes网络，并证明了其具有可重排无阻塞特性。还提出一种将Benes网络转化为非平衡Benes网络的方法，可根据当前Benes网络故障开关的数量和位置实现动态的容错，并提出对应非平衡Benes网络的路由求解算法。基于非平衡Benes网络的Benes网络容错路由能力较传统双路冗余方案有显著的开关利用率与交换基数优势，其可重排无阻塞交换的特性也确保了生产过程中对不良品进行降级标定后仍能提供一致的使用体验，从而在提升容错使用体验的同时，显著降低容错带来的冗余开销。使用现场可编程门阵列（field-programmable gate array，FPGA）验证该算法的性能，结果表明其与性能最好的Benes网络求解加速器路由求解耗时相当，不会成为系统的瓶颈。经研究，针对非平衡Benes网络的容错方法亦可用于锁定Benes网络内特定链路，实现类似Crossbar网络的可控局部重构效果。本文的贡献如下：

（1）提出非平衡 Benes 网络，证明其具有可重排无阻塞交换特性；

（2）提出并用 FPGA 验证了非平衡 Benes 网络的高性能硬件加速求解方法；

（3）基于非平衡 Benes 网络实现了 Benes 网络的高效容错路由，利用 Benes 网络容错路由方法实现了 Benes 网络的可控局部重构。

1　非平衡 Benes 网络

1.1　网络定义与构建方法

Benes 网络由 Clos 网络经特化构建，分为 2×2 边缘开关单元与上下两路子网，2*N*×2*N* Benes 网络的构建方式如图 1 所示。由 Clos 网络的性质可知，网络的可重排无阻塞特性由其内部的上下两路子网保证，在 2 张子网均具备可重排无阻塞特性时，网络本身也具备可重排无阻塞特性。显而易见，构成 Benes 网络的 2 张子网，其交换规模一致，传入本级 Benes 网络的信号会被均匀地分成 2 组，分别进入上下两路子网完成交换，故也可称其为“平衡 Benes 网络”。

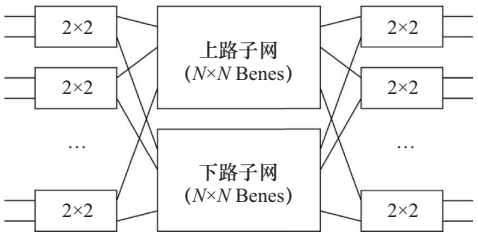


图 1　2*N*×2*N* Benes 网络的构建方式

若在 Benes 网络构建时，上路子网与下路子网的交换规模不一致，选择边缘开关将两张子网对应的输入输出端口尽可能多地互连，并直接引出交换规模较大的子网中剩余未配对的输入输出端口而不经过边缘开关，则可得到非平衡 Benes 网络，一种(2*N*+1)×(2*N*+1)非平衡网络构建方式如图 2 所示。

非平衡 Benes 网络同样遵循 Clos 构建方式，其子网可以是平衡或非平衡的 Benes 网络。由此可见，由于非平衡 Benes 网络不再限制上下路子网的互连规模必须一致，故任意互连规模的非平衡 Benes 网络均可被构建。

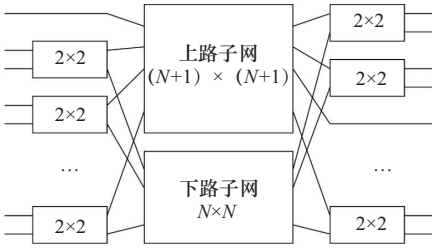


图 2　一种(2*N*+1)×(2*N*+1)非平衡网络构建方式

1.2　可重排无阻塞特性的证明

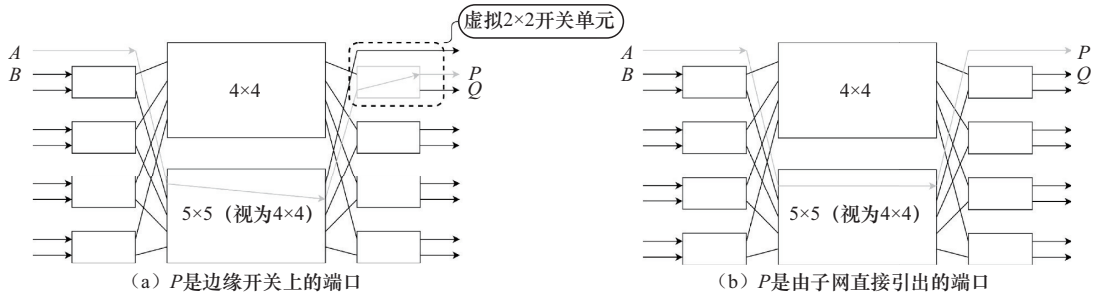
非平衡 Benes 网络具有可重排无阻塞交换特性的充分条件为：

（1）构成非平衡 Benes 网络的 2 个子网均为无阻塞交换网络（无阻塞、可重排无阻塞、严格无阻塞均可）；

（2）上下路子网交换规模差值≤1 且由子网直接引出的输入输出端口最多只能有 1 个。

证明如下：

选择输入侧由子网直接引出的端口 *A* 及任意路由终点 *P*，构建链路 *A*→*P*。显然，非平衡 Benes 网络内，任意一个由全部输入输出端口构成的非冲突全局路由请求中，一定包含该链路 *A*→*P* 对应的请求。由于由子网直接引出的输入输出端口最多只能有 1 个，故要么 *P* 位于边缘开关上，而全部的边缘开关均可路由至任意子网；要么 *P* 由规模更大的子网直接引出。注意到 *A* 同样由规模更大的子网引出，故 *A* 与 *P* 均可连接到同一个子网中，且该子网拥有无阻塞交换特性，因此链路 *A*→*P* 一定可以被构建。构建完链路 *A*→*P* 后，若剩余路由请求总能被满足，则说明非平衡 Benes 网络拥有可重排无阻塞特性。人为屏蔽已构建的路由 *A*→*P*，按 *A* 与 *P* 对应端口特点的不同，可分 2 种情况讨论。差值≤1 时可完成可重排无阻塞交换的实例如图 3 所示，其中图 3（a）表示 *P* 是边缘开关上的端



（a）*P*是边缘开关上的端口　　（b）*P*是由子网直接引出的端口

图3　差值≤1时可完成可重排无阻塞交换的实例

口，图3（b）表示*P*是由子网直接引出的端口。

（1）情况1：输出侧端口*P*是边缘开关上的端口，如图3（a）所示。此时屏蔽链路*A*→*P*后，输出侧将存在2个输出端口不在同一个边缘开关内。将其视为在一个不能变更开关状态的2×2虚拟开关单元中，即可得到1张Benes网络。由以往的求解算法[11-14]可知，任意Benes网络在外层求解时，依其迭代初始边缘开关选择的开关状态不同，外层求解存在至少2个可行解。对于图3（a）构建的网络，选择上述虚拟开关单元作为迭代起始点，由于其开关状态无法变更，故可行解数量退化为1个，但仍能保证有解，即依然能够实现可重排无阻塞交换。

（2）情况2：*P*也是由交换规模较大的子网直接引出的。则屏蔽链路*A*→*P*后，如图3（b）所示，2张子网的可用互连规模将相同，网络的剩余部分可视为一张Benes网络，且拥有Benes网络的全部性质，因此网络必然能够实现剩余的可重排无阻塞交换。

若2张子网的交换规模差值≥2，则无法保证任意全局路由请求的可重排无阻塞。差值≥2时无法完成可重排无阻塞交换的实例如图4所示，当差值=2时，输入和输出侧2×2开关单元中必然各有2个单元仅连接其中一个子网，若其对应的2个端口A和B提出输出侧端口*P*和*Q*，*P*和*Q*位于同一个2×2开关中，显而易见*A*和*B*均通过下路子网完成交换，而*P*和*Q*的路由互斥，无法同时路由至下路子网，上述交换请求无法做到可重排无阻塞。

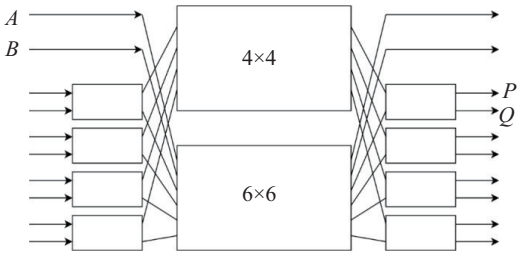


图4　差值≥2时无法完成可重排无阻塞交换的实例

1.3　与Benes网络的关系

$M\times M$非平衡Benes网络可由$N\times N$ Benes网络经过裁剪得到，其中$M<N$且$\lceil \mathrm{lb}\,M\rceil=\mathrm{lb}\,N$。给定$M$与$N$后，选择$k$个边缘开关单元，将其对应输入输出端口标记为“未使用”；同时选择至多1个开关单元，将其中任意一个端口标记为“未使用”，并指定一个开关状态以便将端口的“未使用”标记传递给其中一个子网。过程需要满足$N-M=2k$或$N-M=2k+1$，以确保其满足非平衡Benes网络的可重排无阻塞约束。

将端口的“未使用”状态传递至子网，完成子网的非平衡Benes网络可重排无阻塞约束构建，此时选择的k个边缘开关单元不再是任意指定，而是受到上层网络传递的“未使用”端口影响，推断得出。继续将本级端口的“未使用”状态递归地传递给子网，直至子网自身为2×2开关单元为止。将两路链路均标记为“未使用”的开关单元删除，同时将仅有一路链路标记为“未使用”的开关单元用链路代替，即可完成对应规模非平衡Benes网络的转化。每层递归如算法1所示。

算法 1：端口使用情况传递（输入侧，递归）

输入：输入端口集合 $\boldsymbol{A}$，包含端口标记状态，未使用的会被标记为“×”

输出：子网输入端口集合 $\boldsymbol{B}$

For each A_i in $\boldsymbol{A}$ **do**:

If A_i is marked “×”:

Let A_j=neighbor of A_i

If A_j is marked “×” **do**:

Mark all the two input ports connected to 2×2 switch of A_i “×”

Else **do**:

If already had PENDING port **do**:

Mark all the two input ports connected to 2×2 switch of A_i “×”

Else **do**:

Mark A_i PENDING

End If

End If

End If

End For

Let A_p=port that marks PENDING

Let B_u=port that connected to upper out of the 2×2 switch

Let B_v=neighbor of B_u

If B_v is marked “×” **do**:

Mark A_u “×”

Else **do**:

Let B_l=port that connected to lower out of the 2×2 switch

Mark B_l “×”

End If

Return B=$\{B_i\}$

非平衡 Benes 网络的裁剪法构建方式如图 5 所示，如图 5（a）、图 5（b）和图 5（c）所示，“×”标记的链路为“未使用”，将该标记逐级传递给内层，并按相应规则裁剪受影响的开关单元，即可得到如图 5（d）所示的 5×5 非平衡 Benes 网络，经优化展示后如图 5（e）所示，该网络由 2×2 和 3×3 非平衡 Benes 网络组成。

1.4　基于非平衡 Benes 网络的 Benes 容错路由方法

非平衡 Benes 网络可由 Benes 网络经裁剪得到，故可根据当前 Benes 网络内发生故障的具体开关单元数量和位置，将其裁剪为不同规模的定制非平衡 Benes 网络，使裁剪掉的部分恰好包含全部故障开关单元，即可通过非平衡 Benes 网络的可重排无阻塞特性确保裁剪后的剩余链路仍然能够正常工作。在仅有少量开关故障时，裁剪得到的非平衡 Benes 网络规模总能高于裁剪前互连规模的一半，从而实现了网络的高效容错路由。

开关故障状态分类如图 6 所示，分别称为完全故障与部分故障，对应完全失效与开关在特定状态下不能正常工作。对于前者，认为其完全丧

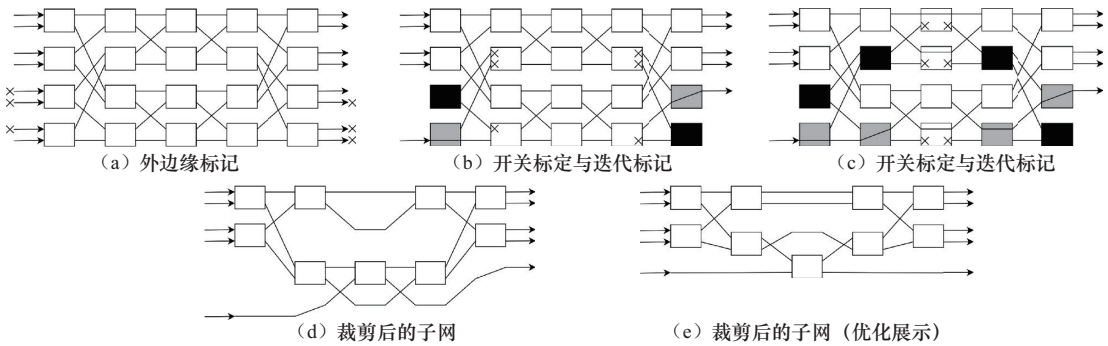


图 5　非平衡 Benes 网络的裁剪法构建方式

失开关功能，视为其所在网络损失2条链路；后者则是在特定条件下仍能保持链路功能，但交换功能丧失，视为其所在网络只损失1条链路。

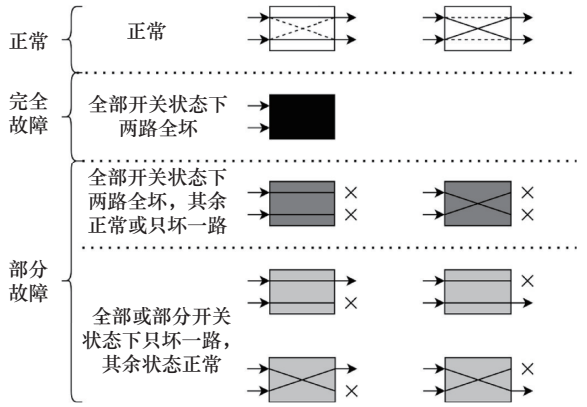


图6　开关故障状态分类

在定位故障开关单元后，完全故障的开关单元意味着其对应的全部2条链路均无法承担数据传输功能，2个端口均应被裁剪；部分故障的开关单元意味着开关单元本身失去了交换能力，每个开关将裁剪掉其中一条链路，并锁定其开关状态为固定值，相当于将开关退化为单一固定链路。

从最核心的子网开始，逐层向外迭代。每轮迭代中，统计当前网络剩余可用交换端口数量，并观察同一张Benes网络2张子网的可用交换端口数量差是否满足非平衡Benes网络可重排无阻塞条件，如不满足，则削减端口数较多的子网内可用端口数量直至二者差值为1。同时选择可用端口，令两张子网输入输出端口对应的链路尽可能多地在边缘开关中交汇，而不是每个边缘开关中仅承载其中一个子网的链路。重复上述迭代过程直至最外层，裁剪结束，得到相应的非平衡Benes网络。单一递归层级的裁剪算法如算法2所示。

算法2： Benes网络容错裁剪（输入侧，递归）

输入： 外层边缘开关故障状态$S=\{S_i\}$

输入： 当前上、下路子网裁剪状态

输出： 本层网络裁剪状态

For i in 0 to “port count of Subnetwork” **do**:

$EP_i^U \leftarrow P_i^U$ /*EP为输入侧边缘开关对应的输出端口故障状态，i为开关序号，U为其上路输出端口，L为其下路输出端口；P为子网输入端口故障状态，i为序号，U为上路子网，L为下路子网*/

$EP_i^L \leftarrow P_i^L$

End For

For i in 0 to “port count of Subnetwork” **do**:

Assign EP_i according to S_i

End For

Let C^U = valid port count of EP^U

Let C^L = valid port count of EP^L

Let $\delta \leftarrow |C^U - C^L|$

If $\delta \geqslant 2$ **do**:

$\delta \leftarrow \delta - 1$

End If

If $\delta > 0$ **do**:

For i in 0 to “port count of Subnetwork” **do**:

If EP_i^U is different with EP_i^L **do**:

$\delta \leftarrow \delta - 1$

$EP_i^U \leftarrow$invalid

$EP_i^L \leftarrow$invalid

End If

If $\delta = 0$ **do**:

break

End If

End For

End If

For i in 0 to “port count of Subnetwork” **do**:

Assign S_i according to EP_i^U and EP_i^L

End For

故障标定示例如图 7 所示。其中，图 7（a）表示初始故障开关标定，图 7（b）表示人为标定正常开关至特定故障状态，图 7（c）表示裁剪后的网络（优化展示）。图 7（a）表示一张 8×8 Benes 网络的故障状态。右侧黑色填充的开关单元，其 2 种开关状态均无法按预期工作；中心灰色填充的开关单元，在“交叉”状态下无法按预期工作，而在“平行”状态下两路链路均可正常工作。从最核心的 2×2 子网开始迭代，发现#3 号子网可用链路数为 1。传递至 4×4 子网，上路 4×4 子网由于存在故障边缘开关，可用链路数为 2，下路 4×4 子网则由于其内部子网链路数少 1 条，可用链路数为 3，2 张子网链路数差值等于 1，满足可重排无阻塞交换条件。传递至 8×8 网络，共计可用链路数为 5，需要在输入输出端口中各选择 5 个作为后续可用端口，并弃用剩余 3 个。

如图 7（b）所示，令弃用的端口尽量集中在最少的开关单元中，人为标定图中左侧#1 和右侧#2 号开关单元为“完全故障”状态，左侧#3 和右侧#4 号开关单元为“部分故障”，锁定其为“平行”状态并弃用其上路链路。向内迭代至 4×4 子网，由于从#1 和#3 开关各自传递了 1 条故障链路至#5 开关，故人为标定#5 为“完全故障”状态；#6 和#7 各自接收了由#1 和#2 传来的 1 条故障链路，故人为标定为“部分故障”并锁定为“平行”状态。核心子网#8、#9、#10 同理。经过归纳，网络等效于图 7（c）所示的非平衡 Benes 网络。

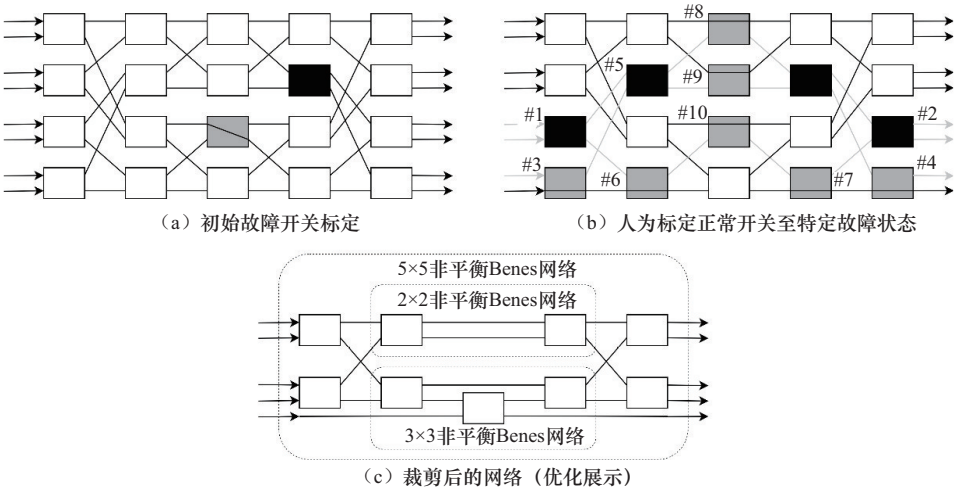
2　非平衡 Benes 快速路由求解

2.1　路由求解算法

非平衡 Benes 网络的构建仍遵循 Clos 网络构建规则，其基础构建单元与 Benes 网络一样为 2×2 开关单元，故可以基于 Benes 网络的二分法[17]实现路由求解。

对于 Benes 网络存在 2 类约束条件：（1）位于同一边缘开关的两个输入输出端口，其对应链路必然路由至不同的子网；（2）若指定输入端口 A 至输出端口 P 的路由，则端口 A 和端口 P 必然路由至相同的子网。故可对全部输入输出端口进行二元染色，其中约束条件 1 中的两个端口染色结果不同，而约束条件 2 中的 2 个端口染色结果相同。根据这 2 类约束条件，给定起始染色端口后交替使用约束条件 1 和 2 依次递推可得到其余端口的染色状态。按文献[17]中的方法将染色状态映射为将要路由的子网，即可快速确定边缘开关的开关状态与子网的路由请求。递归求解直至子网为 2×2 开关单元，即可完成全部开关单元的求解。

不同于 Benes 网络在染色开始时可任意指定



（a）初始故障开关标定　（b）人为标定正常开关至特定故障状态

（c）裁剪后的网络（优化展示）

图 7　故障标定示例

一个未染色的节点作为起始节点，非平衡Benes网络在上下路子网规模不一致时，染色起点必须为直接从子网中引出的链路所对应的端口，并将其染色定义为对应的路由子网。其后续的染色步骤则与文献[17]相同。

2.2 求解器设计

非平衡Benes网络的硬件求解器基于文献[17]的二分法实现，除了接收文献[17]定义的路由关系以外，还会接收当前每个开关单元的裁剪状态（0代表未被裁剪，1代表被裁剪）与被裁剪开关单元的预设开关单元状态。

文献[17]提出的数据依赖解耦方法也适用于非平衡Benes网络，即可通过对约束条件的提前推导得到任意输入端口执行3次推导后对应的输入端口，即文献[17]提到的NB序列。但由于在给定路由关系请求时，未被使用的输入输出端口，其路由终点并不需要给出，因此需要将未使用端口对应的路由起点和终点标记为−1，并在推导NB序列时，一旦推导路径存在−1，则最终的值也为−1。如此，在全部输入端口染色完毕后，即可与文献[17]一样，使用正向映射序列对全部输出端口进行并行染色。NB求解算法如算法3所示。

算法3：NB序列求解

输入：正向映射序列MP

输出：NB

Let MN={MN_i}

For i in 0 to MP.count **do**:

$MN_j \leftarrow_{i,j} = MP_i$

End For

Let NB={NB_i}

For i in 0 to MN.count **do**:

$NB_i \leftarrow MN_{j,j} = \mathrm{Mutex}(i)$

End For

Return NB

染色开始前，需要确定染色的起始端口。由非平衡Benes网络的定义可知，当存在由子网直接引出的端口A，即仅一个端口接入信号的2×2边缘开关时，选取端口A，查询其NB序列中的对应值，若值为−1，则说明存在链路$A \rightarrow P$，其中P为输出侧由子网直接引出的端口。将其排除后，染色过程退化为常规Benes网络的情形，即任意选择尚未被染色的端口即可；若其对应项的值不为−1，则选取其值对应的输入端口A'，该端口对应的链路将路由至与A不同的子网，以A'为起始端口，并以其路由子网对应的染色状态为起始染色状态。染色结束后，利用路由关系将输入侧端口的染色状态并行迁移至输出侧，即可完成两侧的染色。具体如算法4所示。

算法4：输入侧端口染色

交换规模：N

If exists 子网直接引出的输入端口P_a^{in} **do**:

Let $P_{\mathrm{current}} \leftarrow P_a^{\mathrm{in}}$

$C^{\mathrm{in}}[P_{\mathrm{current}}] \leftarrow$ “U” if 子网是上路子网 else “L”

Let $C_{\mathrm{prev}} \leftarrow \boldsymbol{C}^{\mathrm{in}}[P_{\mathrm{current}}]$

$P_{\mathrm{current}} \leftarrow \mathrm{Neighbor}(P_{\mathrm{current}})$

Else **do**:

Let $P_{\mathrm{current}} \leftarrow P_0^{\mathrm{in}}$

Let $C_{\mathrm{prev}} \leftarrow$ “U”

For i in 1 to $\lceil (N-1)/2 \rceil$ **do**:

$\boldsymbol{C}^{\mathrm{in}}[P_{\mathrm{current}}] \leftarrow$ “L” **if** $C_{\mathrm{prev}} =$ “U” else “U”

$C_{\mathrm{prev}} \leftarrow \boldsymbol{C}^{\mathrm{in}}[P_{\mathrm{current}}]$

Let $P_{\mathrm{next}} \leftarrow \mathrm{Mutex}(P_{\mathrm{current}})$ /*定位下一个互斥性端口*/

$P_{\mathrm{current}} \leftarrow P_{\mathrm{next}}$

$\boldsymbol{C}^{\mathrm{in}}[P_{\mathrm{next}}] \leftarrow$ “L” if $C_{\mathrm{prev}} =$ “U” else “U”

$C_{\mathrm{prev}} \leftarrow \boldsymbol{C}^{\mathrm{in}}[P_{\mathrm{next}}]$

$P_{\text{next}} \leftarrow \text{Neighbor}(P_{\text{current}})$/* 定位下一个邻居端口*/

If $\boldsymbol{C}^{\text{in}}[P_{\text{next}}]$ is already colored **do**:

$P_{\text{current}} \leftarrow P_{\text{u}} \in \boldsymbol{P}^{\text{in}}$ and $\boldsymbol{C}^{\text{in}}[P_{\text{u}}]$ is not colored

Else **do**:

$P_{\text{current}} \leftarrow P_{\text{next}}$

End If

End For

一次边缘开关状态求解流程如图8所示，展示了对一张非平衡Benes网络的边缘开关状态求解过程。其中，图8（a）表示接收路由请求，图8（b）表示预先标记故障开关，图8（c）表示计算NB序列，图8（d）表示基于NB序列的染色迭代，图8（e）表示确定全部边缘开关状态。

当得到如图8（d）所示的端口染色状态后，可如文献[17]描述那样，根据边缘开关内信号输入端口位置和其期望路由的子网关系，并行对每个边缘开关进行开关状态求解，并得到全部边缘开关的开关状态如图8（e）所示。子网的路由关系同样可像文献[17]那样进行处理，直至全部开关单元均被遍历并得到开关状态。收集全部开关状态即可完成求解。

3　Benes网络的可控局部重构方法

在进行非平衡Benes网络相关研究时，发现合理利用非平衡Benes网络的裁剪构建方法，可实现传统Benes网络的可控局部重构。

传统Benes网络路由求解算法存在一定的使用局限性，其只保证给定的全局路由能够实现，而不保证相邻2次全局路由请求中相同的路由对应相同的内部动态链路，导致每次路由求解后，实施重构操作前，需要将全部接入终端的流量进行排空，否则可能产生致命的数据丢失或损坏错误。注意到网络高频发生的路由变更为局部路由变更，即只有一部分接入终端$u \in \boldsymbol{U}$会改变其通信目标，而另一部分接入终端$v \in \boldsymbol{V}$仍会保持与当前通信目标的通信，若采用传统路由求解算法，会导致2种可能的情况：（1）通信终端$u \in \boldsymbol{U}$的通信过程被频繁打断，通信效率大幅降低；（2）多条局部路由变更请求会被合并为全局路由变更，以减少重构频率，但会降低网

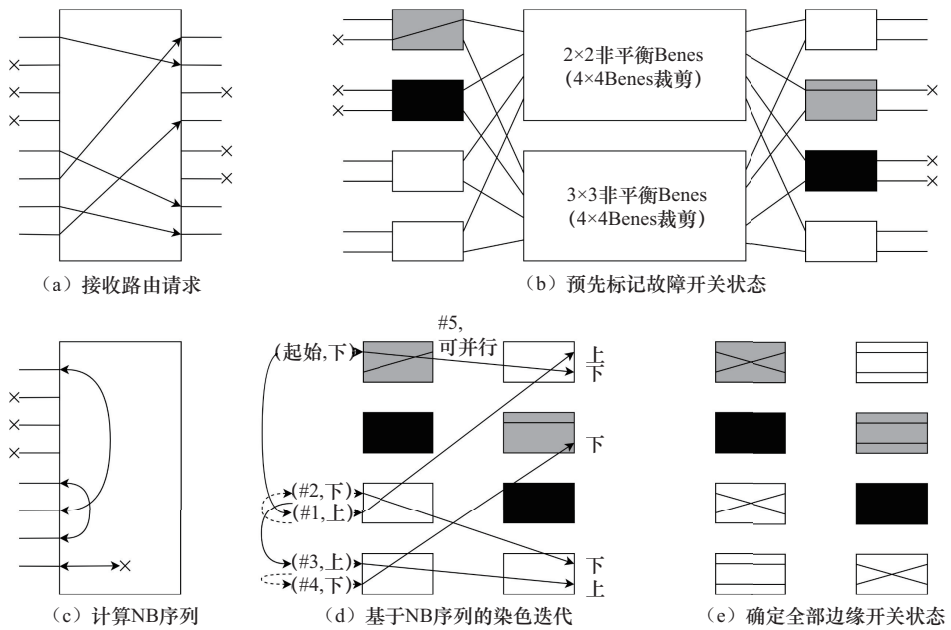


图8　一次边缘开关状态求解流程

络的交换灵活性，大幅增加对通信终端$v \in V$的负面影响。

在进行Benes网络的容错路由研究时，注意到所使用的故障标记手段会锁定被标记开关单元的开关状态，使其在重构过程中不发生改变，因而实现了对其承载链路的部分锁定。由此可知，若一条链路经过的全部开关单元均通过上述故障标记方法进行锁定，即可实现对链路的锁定。此时，将开关阵列剩余部分按非平衡Benes网络可重排无阻塞交换约束进行裁剪，得到非平衡Benes网络，并对其进行路由求解，则可保证其对应可重排无阻塞交换子集的任意重构操作不会影响被锁定的链路，从而避免了这些被锁定链路的流量排空操作，提升网络整体性能。

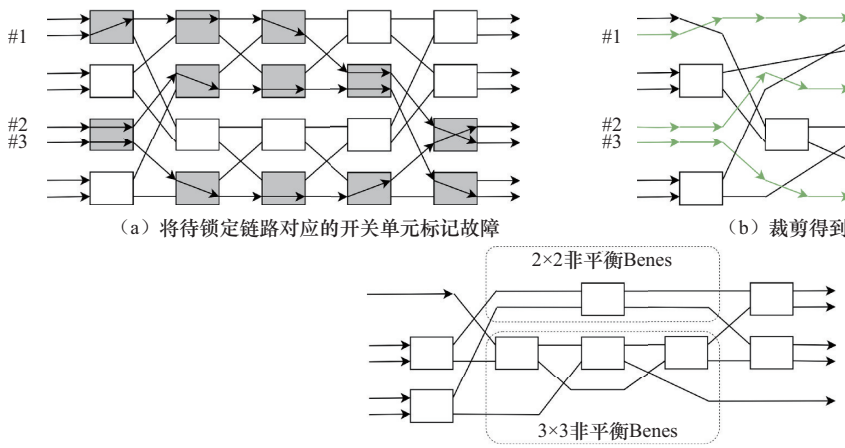
非平衡Benes网络用于局部重构的示例如图9（a）所示，其中，图9（a）表示将待锁定链路对应的开关单元标记为故障，图9（b）表示裁剪得到不含待锁定链路的子网，图9（c）表示裁剪得到不含待锁定链路的子网（优化展示）。在一次Benes网络重构前，若不希望既有链路#1、#2和#3被破坏，则可将链路对应的输入输出端口、开关单元临时标定为“故障”（图中着色链路和开关单元），然后使用动态故障冗余方案，将网络剩余部分裁剪为符合可重排无阻塞交换要求的平衡Benes网络，如图9（b）和图9（c）所示。裁剪完毕后，剩余输入输出端口仍保有可重排无阻塞特性，且重构过程不会改变标定为故障的开关单元的状态，因此，该方案在实现了局部重构的同时，指定链路#1、#2、#3不受该次重构的影响，在重构全程无须排空流量，可保持正常通信。

4 实验及分析

4.1 容错路由交换规模对比

以往的容错方案通过大量冗余开关与链路实现了少量开关故障下的可重排无阻塞交换，但其支持的交换规模不会随开关单元故障数量动态变化，故存在大量开关单元的浪费；同时不能给出稳定可靠的可重排无阻塞交换容量下限，在故障开关单元达到一定数量时可能会失去可重排无阻塞特性。基于非平衡Benes网络的容错方案则可根据故障开关单元的位置与数量规划出当前支持的最大可重排无阻塞交换规模，既能保证阵列的可重排无阻塞交换特性，也能减少开关单元的过度冗余，降低容错成本。

不同冗余方案消耗的开关单元数与可用可重排无阻塞交换规模对比见表1，总结了非平衡



（a）将待锁定链路对应的开关单元标记故障

（b）裁剪得到不含待锁定链路的子网

（c）裁剪得到不含待锁定链路的子网（优化展示）

图9 非平衡Benes网络用于局部重构的示例

表 1　不同冗余方案消耗的开关单元数与可用可重排无阻塞交换规模对比

网络类型	2×2 开关单元消耗数	可用可重排无阻塞交换规模			
		无开关故障	1 开关故障	2 开关故障	3 开关故障
Benes	56	16	—	—	—
双路冗余	56	8	8	8	8
扩张型	48	8	8	8	8
非平衡	56	16	11/12.42/14	9/11.47/14	9/10.93/13
Benes	144	32	—	—	—
双路冗余	144	16	16	16	16
扩张型	128	16	16	16	16
非平衡	144	32	23/26.77/30	19/24.99/30	19/23.94/29
Benes	352	64	—	—	—
双路冗余	352	32	32	32	32
扩张型	320	32	32	32	32
非平衡	352	64	47/55.92/62	39/52.44/62	39/50.37/61

Benes 网络与双路冗余、扩张型 Benes 网络在容错环境下的互连规模表现，表格正文项为其可用的可重排无阻塞交换规模，对照组为常规的 Benes 网络。由于非平衡 Benes 网络的容错规模随故障开关的数量和位置动态变化，故采用蒙特卡洛方法，随机选取 Benes 网络内的开关单元将其标记为故障，实施裁剪法统计可用可重排无阻塞交换规模，进行大量重复实验（超过 100 000 次）统计其分布，用“最低/平均/最高”的方式记述。因采用蒙特卡洛方法得到统计数据，在不同的实验环境中可能存在数据的差异。$N\times N$ 双路冗余方案本质上为 $2N\times 2N$ Benes 网络，但仅接入一半的链路；与扩张型 Benes 网络相比，双路冗余方案减少了一级交换所需的开关级数，故总开关单元消耗数比双路冗余方案略有降低。相同或相似的开关单元消耗数时，基于非平衡 Benes 网络的动态冗余方案使网络的容错效率得到显著提升，开关阵列的可重排无阻塞交换规模在少量开关单元故障时相比传统容错方案可提升最高达 93.75%，平均也能达到 56.05%。

注意到使用互补金属氧化物半导体（CMOS）硅基集成工艺时，开关单元的良率实际上维持在一个较高水平，单一开关阵列超过 3 个开关单元故障的概率极低，开关单元因工艺问题产生插损过大的情形也较为有限，因此使用非平衡 Benes 网络进行动态容错的方法可有效提升开关阵列的整体良率，并能减少因容错产生的链路和开关单元浪费，提高容错效率。以最多容许阵列中 3 个开关故障为设计指标，则相同开关单元消耗数时，非平衡 Benes 网络方案总能提供 10%~20% 的额外交换容量而不破坏可重排无阻塞交换特性，容许 3 个开关故障时的容错交换能力对比如图 10 所示。

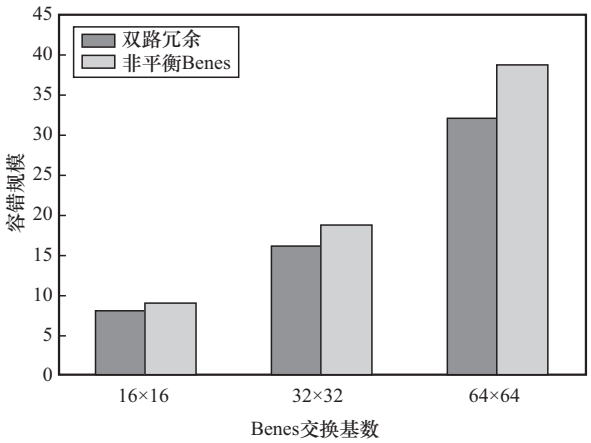


图 10　容许 3 个开关故障时的容错交换能力对比

非平衡Benes裁剪方案会产生较多交换容量为奇数的情形，在现有数据报文网络为全双工通信、需要回传如应答（acknowledgement，ACK）报文等情形下，通信终端总是两两配对，此时奇数交换容量将有一个终端无法与其他终端配对而是自回环，进而无法产生有效通信，造成带宽浪费。但光交换由于不缓存光信号，因此必然存在一张辅助电互连网络用于配置路由等作用。复用该网络传输诸如ACK报文，将全双工通信解耦并简化为单向传输，此时将不再强制要求通信终端的两两配对，而是只要求单向通信链路的建立即可。在网络误码率极低时，复用电互连网络传输ACK产生的时延不会对吞吐量产生影响。因此，此时的奇数交换容量不会因自回环问题产生带宽浪费。若考虑兼容全双工通信场景，亦可屏蔽其中一个输入输出端口使其变为偶数交换容量，从而避免自回环。

4.2 非平衡Benes网络硬件求解加速器性能评估

使用FPGA验证非平衡Benes网络硬件求解器。由同一张Benes网络经裁剪得到的不同的非平衡Benes网络，均可由同一个求解器完成求解。求解器功能模块构成如图11所示，其中预处理模块用于处理Benes网络的预先裁剪结果（进而实现任意非平衡Benes网络配置）、生成NB序列；动态染色模块用于对边缘开关进行端口染色；开关状态求解模块用于将染色结果转化为开关状态，并生成子网的求解初始条件；开关状态汇总模块用于汇总自身和子网的开关状态结果并返回上级。

选用Xilinx ZYNQ ZCU104 FPGA，使用Vivado 2020.1软件生成硬件求解器逻辑的固件并烧录至FPGA进行验证。对应8×8、16×16、32×32规模Benes网络求解器的硬件开销，求解器性能与成本概览见表2，其中LUT为硬件查找表（look-up table），FF为寄存器（flip-flop register），在不同的实验环境中可能存在数据的差异。求解器能够实现较高的时钟频率，较高的并行度也保证了较短的求解周期数消耗，因此求解耗时较短。

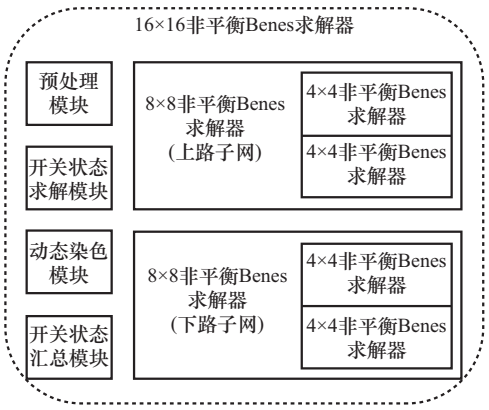


图11 求解器功能模块构成

表2 求解器性能与成本概览

求解规模	LUT	FF	时钟频率/MHz	求解耗时/ns
8×8	585	310	650	≤20
16×16	2 405	861	600	≤40
32×32	9 662	2 155	420	≤102
64×64	40 896	5 694	250	≤312
128×128	168 042	15 177	90	≤1 595

求解器性能对比见表3，表3展示了与其他Benes求解器的求解耗时对比结果。显而易见，非平衡Benes求解器的求解耗时仍能做到与性能最优的Benes求解器处于相同水平，因此对非平衡Benes网络的路由求解不会成为性能瓶颈。

5 结束语

本文在Clos与Benes网络构建原理的基础上，提出了非平衡Benes网络，并证明了其可重排无阻塞特性。依托于非平衡Benes网络的可重排无阻塞特性，提出一种将常规Benes网络通过屏蔽部分开关单元的方式转化为非平衡Benes网络的方法，并在开关单元故障这一场景下进行了深度适配，使其可有效处理Benes网络在部分开关单元故障时会失去可重排无阻塞交换特性的问题。相比于传统Benes网络容错解决方案，本文提出的非平衡Benes网络可实现基于故障开关单元数的动态容错调节，在网络故障开关数相同时，支

表3　求解器性能对比

求解规模	本文	文献[16]	文献[14]	文献[17]	文献[15]
8×8	≤20 ns	100 ns	96 ns	12 ns	13 ns
16×16	≤40 ns	150 ns	250 ns	26 ns	73 ns
32×32	≤102 ns	200 ns	960 ns	85 ns	216 ns
64×64	≤312 ns	—	3 072 ns	—	540 ns
128×128	≤1 595 ns	—	—	—	—

持的可重排无阻塞交换规模显著高于常规容错方案平均达 56.05%，最高达 93.75%，且能给出确定的可重排无阻塞交换子集。本文提出的非平衡 Benes 网络硬件求解器同样拥有与当前性能最优的 Benes 硬件求解器相当的求解性能，对基于 16×16 Benes 网络裁剪的非平衡 Benes 网络单次路由求解耗时小于或等于 40 ns，确保了应用非平衡 Benes 网络的系统不会因路由求解产生性能瓶颈。

使用非平衡 Benes 网络模型，生产制造过程中出现的轻微瑕疵品可降级为更低交换基数的开关阵列，而不必进入报废流程，可大幅提高生产良率；在使用过程中，随着阵列内部分开关单元的老化与意外故障，阵列仍可进行二次标定降级使用而无须淘汰，且仍保有可重排无阻塞交换的特性。在对可靠性要求较高的场景，非平衡 Benes 网络模型亦可强化传统双路冗余技术，指导其在内部故障积累到何种程度时会丧失可重排无阻塞交换特性，进而降低开关阵列的替换频率，降低成本。利用基于非平衡 Benes 网络的动态容错方法，实现了针对常规 Benes 网络的可控局部路由重构方案，可令特定链路不受重构影响，在重构全程可正常通信。该重构方法部分实现了诸如 Crossbar 等严格无阻塞网络的交换特性，使应用 Benes 网络的互连系统支持更加灵活的路由调度而不会损失交换性能。

本文初步提出了非平衡 Benes 网络，并以此为基础给出常规 Benes 网络的容错可重排无阻塞交换规模下限。尽管该下限目前显著高于已有的双路冗余方案，但仍偏保守，在部分容错场景存在无阻塞交换规模更大的容错方案，因此后续研究将针对这类场景进一步完善 Benes 网络容错路由方法，使其网络内开关单元利用率进一步提升，支持更大的容错交换规模。

参考文献：

[1] Beneš V E. Permutation groups, complexes, and rearrangeable connecting networks[J]. Bell System Technical Journal, 1964, 43(4): 1619-1640.

[2] Clos C. A study of non-blocking switching networks[J]. Bell System Technical Journal, 1953, 32(2): 406-424.

[3] Shen L, Lu L J, Guo Z Z, et al. Silicon optical filters reconfigured from a 16 × 16 Benes switch matrix[J]. Optics Express, 2019, 27(12): 16945-16957.

[4] Tunesi L. Integrated Benes optical switches: an automated bottom-up design implementation[D]. Torino: Politecnico di Torino, 2021.

[5] Chu T, Qiao L, Tang W J, et al. Fast, high-radix silicon photonic switches[C]//Proceedings of the 2018 Optical Fiber Communications Conference and Exposition (OFC). Piscataway: IEEE Press, 2018: 1-3.

[6] Wang B, Liu L B, Deng C C, et al. Exploration of benes network in cryptographic processors: a random infection countermeasure for block ciphers against fault attacks[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(2): 309-322.

[7] Runge A, Kolla R. Using benes networks at fault-tolerant and deflection routing based network-on-chips[C]//Proceedings of the 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS). Piscataway: IEEE Press, 2016: 1-8.

[8] Wang Y, Qin Y B, Deng D Z, et al. A 28nm 27.5TOPS/W approximate-computing-based transformer processor with asymptotic sparsity speculating and out-of-order computing[C]//

Proceedings of the 2022 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE Press, 2022: 1-3.

[9] Ghandriz Z S, Zeinali E K. A new routing algorithm for a three-stage clos interconnection networks[J]. International Journal of Computer Science Issues (IJCSI), 2011, 8(5): 309.

[10] Wang L K, Ye T, Lee T T. A parallel route assignment algorithm for fault-tolerant clos networks in OTN switches[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(5): 977-989.

[11] Waksman A. A permutation network[J]. Journal of the ACM, 1968, 15(1): 159-163.

[12] Nassimi, Sahni. A self-routing benes network and parallel permutation algorithms[J]. IEEE Transactions on Computers, 1981, C-30(5): 332-340.

[13] Lee K Y. A new benes network control algorithm[J]. IEEE Transactions on Computers, 1987, C-36(6): 768-772.

[14] Nikolaidis D, Groumas P, Kouloumentas C, et al. Novel Benes network routing algorithm and hardware implementation[J]. Technologies, 2022, 10(1): 16.

[15] Jiang Y, Yang M. Hardware implementation of parallel algorithm for setting up Benes networks[C]//Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016: 10.

[16] Koloko L, Matsumoto T, Obara H. Design and implementation of fast and hardware-efficient parallel processing elements to set full and partial permutations in Beneš networks[J]. The Journal of Engineering, 2021(6): 312-320.

[17] 秦梦远, 刘宏伟, 郝沁汾 . 高性能Benes网络路由求解算法及硬件加速器[J]. 计算机工程与应用, 2025, 61(14): 163-175.
Qin M Y, Liu H W, Hao Q F. High-performance route-resolving algorithm and hardware accelerator for benes network[J]. Computer Engineering and Applications, 2025, 61(14): 163-175.

[18] Jahanshahi M, Bistouni F. Improving the reliability of the Benes network for use in large-scale systems[J]. Microelectronics Reliability, 2015, 55(3/4): 679-695.

[19] Dong Y, Wang J. Fault tolerance study on large scale Benes switches[C]//Proceedings of the HPSR. 2005 Workshop on High Performance Switching and Routing. Piscataway: IEEE Press, 2005: 197-201.

[20] Dong Y, Wang J. Fault tolerance design for large-scale optical switches[J]. Optical Switching and Networking, 2008, 5(1): 51-58.

[21] Hwang I S, Tseng W D, Huang I F. Integrated fault tolerant connections-scheduling for dilated Benes network[J]. IEE Proceedings - Communications, 2005, 152(3): 343.

[22] Tseng W, Hwang I, Lee L, et al. Clique-partitioning connections-scheduling with faulty switches in dilated benes network[J]. Journal of the Chinese Institute of Engineers, 2009, 32(6): 853-860.

[23] 张金花, 武保剑, 邱昆 . 扩张型Benes光交换芯片未满配置情形下的约束链路路由算法[J]. 激光与光电子学进展, 2019, 56(21): 211301.
Zhang J H, Wu B J, Qiu K. Constrained link routing algorithm for dilated benes optical switching chips under non-full configuration[J]. Laser & Optoelectronics Progress, 2019, 56(21): 211301.

[24] 冯斐玲 . Benes网的寻径控制及容错分析[J]. 计算机学报, 1994, 17(S): 26-34.
Feng F L. Routing control and fault-tolerance analysis of benes network[J]. Chinese Journal of Computers, 1994, 17(S): 26-34.

[作者简介]

秦梦远（1994– ），男，中国科学院计算技术研究所博士生，主要研究方向为新型计算机系统结构、Benes网络和可重构光互连系统。

刘宏伟（1984– ），男，博士，中国科学院计算技术研究所高级工程师，主要研究方向为高性能处理器设计、异构计算、硬件安全与加密芯片和软件定义芯片。

郝沁汾（1969– ），男，中国科学院计算技术研究所正高级工程师、博士生导师，主要研究方向为高性能计算机、高端SMP服务器和CPU等。