Journal of Computer Applications
计算机应用,2018,38(5):1470 - 1475
ISSN 1001-9081
CODEN JYIIDU
2018-05-10
http://www.joca.cn

文章编号:1001-9081(2018)05-1470-06
DOI:10.11772/j.issn.1001-9081.2017102411

面向非全互连3D NoC的自适应单播路由算法

孙美东*,刘勤让,刘冬培,燕昺昊
(国家数字交换系统工程技术研究中心,郑州 450002)
(*通信作者电子邮箱137070845@qq.com)

摘　要:针对在非全互连三维片上网络(3D NoC)架构中的硅通孔(TSV)表只存储TSV地址信息,导致网络拥塞的问题,提出了记录表结构。该表不仅可以存储距离路由器最近的4个TSV地址,也可存储相应路由器输入缓存的占用和故障信息。在此基础上,又提出最短传输路径的自适应单播路由算法。首先,计算当前节点与目的节点的坐标确定数据包的传输方式;其次,检测传输路径是否故障,同时获取端口缓存占用信息;最后,确定最佳的传输端口,传输数据包到邻近路由器。两种网络规模下的实验结果表明,与Elevator-First算法相比,所提算法在平均延时和吞吐率性能指标上有明显的优势,且在网络故障率为50%时,Random和Shuffle流量模型下的丢包率分别为25.5%和29.5%。

关键词:非全互连三维片上网络;记录表;自适应单播;平均延时;吞吐率
中图分类号:TP393.02　　**文献标志码**:A

Adaptive unicast routing algorithm for vertically partially connected 3D NoC

SUN Meidong*, LIU Qinrang, LIU Dongpei, YAN Binghao
(*National Digital Switching System Engineering & Technological R&D Center, Zhengzhou Henan* 450002, *China*)

Abstract: Traditional TSV (Through Silicon Via) table in vertically partially connected three-Dimensional Network-on-Chip (3D NoC) only stores TSV address information, which easily causes network congestion. In order to solve this problem, a record table architecture was proposed. The record table stored not only the nearest four TSV addresses to the router, but also the input-buffer occupancy and fault information of the corresponding router. Based on the record table, a novel adaptive unicast routing algorithm for the shortest transmission path was proposed. Firstly, the coordinates of current node and destination node were calculated to determine the transmission mode of packets. Secondly, by using the proposed algorithm, whether the transmission path was faulty and got information of buffer occupancy was obtained simultaneously. Finally, the optimal transmission port was determined and the packets were transmitted to the neighboring router. The experimental results under two network sizes show that the proposed algorithm has obvious advantages in average delay and throughput compared with Elevator-First algorithm. Additionally, the rates of losing packet under Random model and Shuffle traffic model are 25.5% and 29.5% respectively when the network fault rate is 50%.

Key words: vertically partially connected three-Dimension Network-on-Chip (3D NoC); record table; adaptive unicast; average delay; throughput

0　引言

在3D集成系统中,使用垂直互连技术将多层晶片堆叠在一起[1]。因为硅通孔(Through Silicon Via, TSV)技术能提供最大的垂直互连密度和最小的层间距离,所以它是目前使用最普及的垂直互连技术[2]。虽然在集成系统中使用三维片上网络(three-Dimensional Network-on-Chip, 3D NoC)架构可以提高网络性能,但TSV的使用仍面临3个问题:1)TSV的制造工艺还不够成熟,使得3D集成电路成品率较低[3];2)在3D集成电路封装以及焊接TSV过程中,TSV可能出现短路导致故障[4];3)相对于普通的2D水平链路,TSV有较大的面积消耗[5]。所以在保证正常通信的前提下,应使TSV的数量尽可能减少,这使得非全互连3D NoC架构得到广泛研究。

对于3D NoC,路由算法是十分关键的问题之一,前人在这一问题上已经作出相关的研究。如文献[6]提出LAFT(Look Ahead Fault Tolerant)算法,此算法分为三个阶段:第一阶段,读取微片(flit),计算下一节点的地址;第二阶段,得到由故障模块发送的下一节点的故障信息并且获取其他三个方向的最短路径;第三阶段,做出路由方向的决定,其中最短路径的方向具有最高优先级。由于LAFT规定路径选择性为第二优先级,所以如果选取路径选择性多的方向,但此方向上的下一节点出现多处故障,此时会导致路径堵塞,使传输不能选择最短路径。针对上述现象,文献[7]提出HLAFT(Hybrid Look Ahead Fault Tolerant)算法,对于每一个到来的flit,该算法判断计算的方向是否导致路径堵塞。如果因为路径堵塞而选择非最短路径,则会根据当前节点和邻居节点的状态重新

收稿日期:2017-10-11;**修回日期**:2017-12-18;**录用日期**:2017-12-20。
基金项目:国家科技重大专项(2016ZX01012101);国家自然科学基金资助项目(61572520, 61521003)。
作者简介:孙美东(1993—),男,黑龙江哈尔滨人,硕士研究生,主要研究方向:三维片上网络路由算法;　刘勤让(1975—),男,河南商丘人,研究员,博士,主要研究方向:宽带信息网络、片上网络设计;　刘冬培(1985—),男,湖南永州人,博士,主要研究方向:芯片验证与测试;燕昺昊(1994—),男,山西吕梁人,硕士研究生,主要研究方向:流量识别、入侵检测。

计算输出端口。HLAFT 算法可以避免路径堵塞,然而只适用于全互连的 3D NoC。DyXYZ(Dynamic XYZ)路由算法在 X、Y、Z 方向分别使用 4、4、2 个虚通道,将网络分成 2 个主网络,每个主网络又包含 4 个次网络[8]。因为每个次网络采用不同的虚通道,所以此算法不会死锁。当选择输出通道时,DyXYZ 算法考虑输出方向上的缓存信息,进一步提高网络性能,但此算法同样也只适用于全互连的 3D NoC 架构且算法共采用 10 个虚通道,增加了网络的硬件开销。

文献[9]和文献[3]分别提出适用于非全互连不规则 3D NoC 架构的 Elevator-First 路由算法和路由器结构,该算法采用 2 个虚通道避免死锁。由于虚通道的使用过于耗费资源,文献[10]对 Elevator-First 算法进行改进,当选择“电梯”节点时,增加限制规则,使得不用虚通道就可以避免死锁。基于 TSV 上/下表的容错路由算法通过在每个路由器中添加 TSV 上/下表寻找最优的 TSV 进行层间传输;此外,为了高效地更新路由表,每个路由器会额外存储一张连接表[11],但由于该算法中的 TSV 上/下表需要存储每层所有 TSV 的地址,且还要额外存储一张连接表,使得表开销过大,并且当扩大网络规模时,表的开销呈剧烈的增长。针对 TSV 上/下表开销过大的问题,低开销容错路由算法提出新的表策略,但此算法并未考虑垂直方向的拥塞信息[12]。因此,本文针对文献[11]表开销过大以及文献[12]出现的垂直拥塞问题,提出新的表结构和路由算法,该算法不仅能避让故障和拥塞节点,而且可以进行死锁恢复和免活锁,保证了有效的数据传输。

1　非全互连 3D 拓扑结构

非全互连 3D 拓扑结构层内采用 2D mesh 结构,且每层的规模相同,其与 3D mesh 结构不同的是 TSV 的分布是部分且随机的。该结构中具有两类路由器:三维路由器和平面路由器。含有 TSV 的路由器被称为三维路由器,具有 7 个端口,分别为本地、东、西、南、北、上、下方向的端口。其余的路由器称为平面路由器,具有五个端口,分别为本地、东、西、南、北方向的端口。上述的拓扑结构模型如图 1 所示,每个路由器的位置由三维笛卡尔坐标表示。

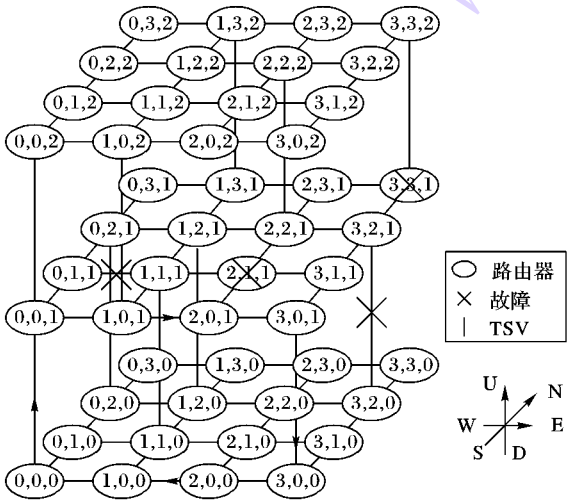

图 1　非全互连 3D 拓扑模型

Fig. 1　Vertically partially connected 3D topology model

2　记录表的设计

在非全互连的 3D NoC 中,上/下层传输的数据包要选取最合适的垂直连接,所以记录垂直连接的位置信息显得尤为重要。本文与文献[11]和文献[12]中设计的表结构不同,此表将 TSV 上/下表整合到一个表中,不仅记录与当前路由器 4 个方向距离最近的 TSV 位置信息,也记录相邻节点的缓存占用和故障信息。在传输数据包时可以选择无故障且缓存占用少的端口,减少拥塞的情况和传输时间,提高网络性能。图 1 中(1,1,1)节点的记录表如表 1 所示。如果当前路由器连接的链路或路由器故障,在表中用信号 0 表示;如果可用,用信号 1 表示。本文设置路由器输入端口的缓存深度为 8 flits。

表 1　节点(1,1,1)存储的记录表

Tab. 1　Record table stored by node (1,1,1)

端口	故障	缓存	上 TSV			下 TSV		
			地址	跳数	缓存	地址	跳数	缓存
北	1	4	(1,3)	2	6	(1,2)	1	5
南	1	6	(1,0)	1	3	(3,0)	3	3
东	0	1	(2,2)	2	2	(3,0)	3	7
西	0	4	(0.0)	2	2	(0,2)	2	4

2.1　记录表的建立

记录表是由路由器发送广播数据包建立的,其中每个广播包会存储自身在网络中传输的跳数。表建立的过程分为两步:第一步是层间广播,由偶数层三维路由器向上/下层广播带有自身输入缓存占用信息的数据包,奇数层三维路由器收到数据包后会记录缓存信息同时向偶数层返回自身的输入缓存占用信息,此时三维路由器都具有邻近层的输入缓存占用信息。第二步是层内广播,三维路由器将缓存信息加入到数据包中向层内 4 个方向发送广播包,接收到的路由器会向其返回输入缓存信息。如果接收到的路由器记录表中没有 TSV 地址,则将包中的 TSV 地址和相应的缓存信息加入到记录表中;如果已有 TSV 地址,选择跳数比表中记录更小的广播包进行更新 TSV 地址和相应的缓存。为了减少数据包的重传,当收到两个相同的数据包会选择一个丢掉。

```
1)  even layer 3D routers broadcast packets;
2)  if ( routers ← packets)                    /* 收到广播包 */
3)    record and return buffer;
4)    plane routers broadcast packets;
5)    if ( routers ← packets)
6)      record and return buffer;
7)      if ( record table is blank || TSV address is smaller)
        /* 记录表的 TSV 地址为空或数据包中携带的 TSV 地址较小 */
8)        update table;
9)      end if
10)     hop + 1,  broadcast packets;
          /* 广播包的跳数加 1,继续向层内其他方向广播 */
11)   end if
12)  end if
```

2.2　记录表的更新

通常,三维片上网络系统中出现的故障分为短暂性故障和永久性故障[13],当发生故障时,永久性故障相比短暂性故障会更大程度地影响系统性能,所以本文考虑的故障是永久性故障。在三维片上网络系统中出现的故障模型主要有以下两种:

1)路由器故障。三维路由器故障,如图 1 中的(3,3,1)路由器故障;平面路由器故障,如图 1 中的(2,1,1)路由器故障。

2)链路故障。层内链路故障,如图 1 中的(1,1,1)路由器西方向故障;层间链路故障,如图 1 中的(3,2,1)路由器下垂直连接故障。

当链路或路由器故障时,由检测到故障的路由器发送与建立记录表时相同的广播包进行记录表的更新。为达到增量更新减少更新开销,此过程只修改需要更新的表项。表更新过程与建立过程相似,这里不再赘述。

3 自适应单播路由算法

3D NoC 中的路由算法是将数据包按照一定的路径从源节点准确无误地传输到目的节点,本文提出的层内和层间路由算法中主要符号和含义如表 2 所示。

表 2 路由算法中主要符号及含义

Tab. 2 Main symbols and meanings in routing algorithm

符号	含义	符号	含义
(X_c,Y_c,Z_c)	当前节点 C 坐标	E	东方向
(X_d,Y_d,Z_c)	目的节点 D 坐标	W	西方向
(X_m,Y_m,Z_m)	映射节点 M 坐标	S	南方向
ΔX	节点在 X 维的坐标差	N	北方向
ΔY	节点在 Y 维的坐标差		

3.1 层内路由算法

基于记录表中的信息,本文提出一种新的层内路由算法。首先根据当前节点和目的节点的地址确定最优方向,然后判断当前节点的故障信息,如最优方向故障则选取非最优方向,如果两个非最优方向的故障信息相同则进一步考虑端口的缓存信息,使得数据包可以选择无故障输出端口且避免拥塞。由于发生故障的连接会比发生拥塞的连接更易影响网络的性能,所以本文将判断故障信息的优先级设为最高,拥塞的优先级次之。

```
1)  if (ΔX = 0 && ΔY = 0)
2)    local port ← packet;
3)  else if (ΔX = 0)
4)    N or S is available, N or S ← packet; otherwise, check W and
      E; check W and E adjacent routers;   /* N 或 S 可用,选择
      相应端口;否则,检查 W 和 E 方向的故障,同时检查 W 和
      E 可用方向邻近路由器的故障信息 */
5)  if (one of W and E is faulty)        /* 两个方向其一故障 */
6)    W or E ← packet;                   /* 选取无故障端口 */
7)  else if (W and E are faulty)
8)    N or S ← packet;          /* 选择目标方向相反的端口 */
9)  else (W and E aren' t faulty)
10)   adjacent routers are available in target direction, less buffer W
      or E ← packet; otherwise, less faulty port ← packet;
      /* 两个邻近路由器在所需方向上可用,选择 W 或 E 缓存
      占用少的端口;两个邻近路由器在所需方向上故障不同,
      选择故障少的方向 */
11)  else if (ΔY = 0)
12)   处理方式同上;
13) else (ΔX≠0 && ΔY≠0)
14)   check 2 target direction and adiacent routers;   /* 检查与目
      标方向距离最近两个方向的故障信息,同时检查这两个方
      向邻近路由器的故障信息 */
15) if (one of them is faulty)
16)   no faulty port ← packet;
17) else if (they are faulty)
18)   opposite port ← packet;
19) else (they aren' t faulty)
20)   adjacent routers are available, less buffer port ← packet;
      otherwise, less faulty port ← packet;
```

3.2 层间路由算法

当目的节点与源节点不在同层时,要先将数据包传输到目的节点层,然后再传输到目的节点。在层间传输时,选择最优的垂直连接是关键,所以在寻找最优垂直连接时,层间路由算法将拥塞和故障信息同时考虑。本文将目的节点映射到源节点所在层的节点称为映射节点。如果映射节点自身含有需要的 TSV,那么选择此 TSV 进行传输;否则,根据映射节点的记录表,计算源节点与各个 TSV 的信息进而选择出最优的 TSV,将数据包传输到邻近层,然后根据目的节点地址继续在层内传输。如果此层仍不是目的层,则采用同样的方法继续传输。源节点与各个 TSV 的信息计算方法为:

$$Info = dis + interbuff \tag{1}$$

其中:dis 表示源节点与映射节点记录表中各个 TSV 的距离,选择最小距离的 TSV 可以减少传输延时;$interbuff$ 表示 TSV 连接的邻近层路由器上/下输入端口的缓存占用信息,选择缓存占用最少的端口可以缓解网络的拥塞;$Info$ 值越小表示选择的 TSV 地址越优,可以作为最优的传输路径进行传输。

```
1)  if (Z_d > Z_c)
2)    X_m = X_d; Y_m = Y_d; Z_m = Z_c;
3)    if (M has TSV)          /* 映射节点含有需要的 TSV */
4)      temporary address ← TSV, inner-layer routing;
        /* 选择此 TSV,采用层内路由进行传输 */
5)    else
6)      calculate TSV, temporary address ← TSV, inner-layer
        routing;
7)    else if (Z_d < Z_c)
8)      处理方式同上;
9)  else
10)   intra-layer routing;
```

由上所述,本文提出的自适应单播路由算法在当前节点选择最佳的输出端口后,在接下来的路由器中循环执行判断逻辑,所以该算法的时间复杂度为 $O(N)$,N 为网络中每一维的节点数目。

3.3 死锁恢复与免活锁

在三维片上网络架构中采用自适应路由算法很可能会引起死锁。如图 1 所示,当节点(1,0,1)向节点(1,0,0)发送数据包,同时节点(2,0,0)向节点(2,0,1)发送数据包,两条传输路径相互占用资源,形成死锁。本文采用 RAB(Random-Access-Buffer)[7]进行死锁恢复。在没有死锁的情况下采用 FIFO 缓存管理机制,一旦检测到死锁,RAB 会放弃缓存中正在执行的请求,寻找允许的请求进行缓存释放,进而打破死锁恢复网络状态。

因为本文考虑网络中的拥塞和故障信息,使得数据包在传输时,由于多次避免拥塞和故障的节点导致活锁现象发生。所以本文采用跳数限制避免活锁,具体方法是如果数据包记录的跳数超过设定的阈值,本文算法对路径的选择将只进行故障的判断,从而避免了活锁。

4　实验仿真

4.1　数据包设计

根据记录表中的内容和路由算法的特性,本文设计出对应的数据包结构,如图 2 所示。其中数据包分为两类:1)单播包,用来进行数据传输;2)广播包,用来建立和更新记录表。本文采用虫孔路由器,数据传输的最小单位是微片,微片大小为 32 b。路由器根据前 2 b 进行头微片、体微片和尾微片的类型区分,其中头微片带有跳数和目的地址等信息,体微片和尾微片携带传输的数据。当网络出现故障时,及时更新记录表可以减少传输延时和功耗开销,所以用于更新记录表的广播包具有比单播包更高的优先级,因此用第 3 b 代表包的优先级(Pri):1 代表高优先级;0 代表低优先级。Temp 字段为 1 代表数据包包含临时目的地址,Temp 为 0 代表数据包包含真实目的地址。Hop 字段记录数据包在网络中传输的跳数,大小为 7 b。X_d、Y_d、Z_d 分别代表目的地址的 X、Y、Z 轴坐标,因为设定拓扑结构为 32 × 32 × 32,所以每一维的坐标需要用 5 b 表示。其中 Hop 和坐标地址所需的位数根据拓扑结构大小而相应地改变。Ext 字段可供以后进行扩展来使用。

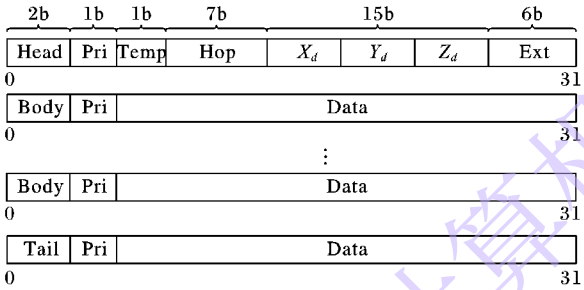

图 2　数据包结构

Fig. 2　Packet structure

4.2　性能仿真分析

本文采用台湾大学开发的 Access Noxim 开源仿真器进行网络性能仿真,它将 Noxim 和 HopSpot 结合,使得 Access Noxim 支持网络模型、功率模型和热模型的 3D NoC 仿真。

为了说明算法的性能,实验将算法应用到不同规模的网络结构,并且采用了两种流量模型进行目的地址的选择。由于文献[9]和文献[11]使用两个虚通道避免死锁,为了实验的公平起见,本实验使用 RAB 死锁恢复机制代替虚通道。

表 3　实验参数设置

Tab. 3　Setting of experimental parameters

实验参数	参数设置	实验参数	参数设置
网络结构	4 × 4 × 4、6 × 6 × 6	数据包大小	最大 8 flits、最小 4 flits
流量模型	Random、Shuffle	微片大小	32 b

4.2.1　网络平均延时

在不同的数据包注入率下,4 种算法在不同网络规模下的平均延时如图 3 和图 4 所示。单个数据包的延时是指数据包的头微片进入网络到目的节点接收尾微片的这段时间。平均延时是指多个数据包延时的平均值。数据包注入率是指每个 IP 核在每个时钟周期下向网络中注入数据包的速度。

在 Random 流量模型下,由于本文考虑到各输入端口的缓存占用信息,在传输数据包时会判断选择最优路径,所以在注入率较低时,平均延时高于文献[9]和文献[11]的算法。但在两种网络规模下,当注入率分别大于 0.028 和 0.02 时,由于文献[9]和文献[11]算法只考虑水平和垂直连接的故障,导致拥塞的情况发生,所以平均延时高于本文算法。文献[12]和本文的平均延时情况很接近,原因是它也同样考虑了层内缓存占用信息。

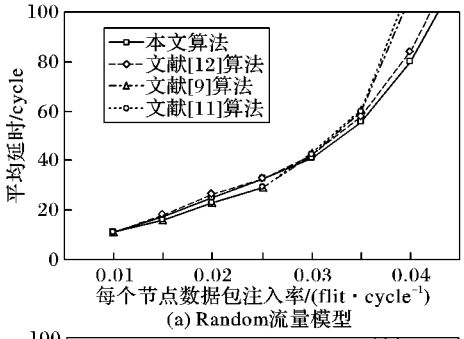

图 3　4 × 4 × 4 网络规模下的平均延时

Fig. 3　Average delay under 4 × 4 × 4 network size

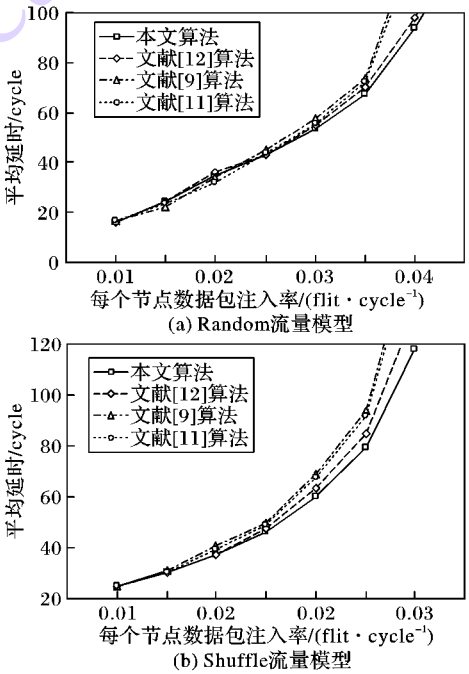

图 4　6 × 6 × 6 网络规模下的平均延时

Fig. 4　Average delay under 6 × 6 × 6 network size

在 Shuffle 流量模型下,文献[9]和文献[11]算法在 4 × 4 × 4网络规模并且注入率大于 0.015 时,平均延时均高于本文算法。文献[12]算法在注入率小于 0.016 时依旧与本文有着接近的平均延时,但在注入率大于 0.016 后,平均延时高于本文。这是因为 Shuffle 流量模型以层间传输数据包为主,本文额外考虑了垂直方向的缓存占用信息,当注入过多的数

据包时,可以减少层间传输的拥塞情况,使得平均延时低于文献[12]算法。6×6×6 网络规模下的情况与上述类似,但因为网络规模的增加,所有算法的平均延时都有所增加。

4.2.2 网络吞吐率

吞吐率是指网络接收或发送消息的速率,实验以 flit 为网络中基本传输单位,所以吞吐率的大小可以用每个节点在每个时钟周期下传输的 flit 数量来衡量。吞吐率的理论计算公式如下:

$$Throughput = \frac{\sum_{i=1}^{P_N} Length_i}{N_{router} \times T} \tag{2}$$

其中:*Throughput* 指网络吞吐率,吞吐率越高表明网络可以处理的数据包越多;N_{router} 指网络中节点的数量,与网络规模有关;T 指实验的仿真时间;P_N 指在 T 时间内成功到达目的节点的数据包总数;$Length_i$ 指第 i 个数据包的大小(包含 flit 的数量)。

图 5 和图 6 为在两种网络规模下 4 种算法的吞吐率比较。

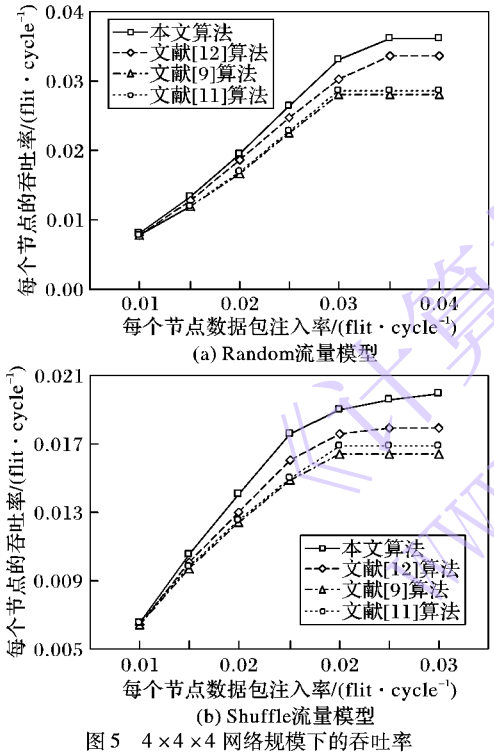

图 5 4×4×4 网络规模下的吞吐率

Fig. 5 Throughput under 4×4×4 network size

可以看出随着注入率的增加,本文算法比文献[9]、文献[11]和文献[12]算法都有优势,在 Random 流量模型下本文算法的优势并不明显,但在 Shuffle 流量模型下本文的吞吐率较其他三个算法有明显的优势。其原因是各节点的缓存使用率随着数据包注入率的增加而逐渐提高,对于以层间数据包传输为主的 Shuffle 流量模型,本文算法不会因为避让故障而导致某一层或某一节点发生拥塞。

另外由于文献[9]算法选择的 TSV 地址直接从寄存器中读取,当存储的 TSV 发生故障,即便有可用的 TSV 也无法进行层间传输,导致网络提前趋于饱和,在两种流量模型下的吞吐率均低于其他三个算法。

4.2.3 可靠性

数据包的丢弃是因为传输路径中的 TSV 故障或网络中的资源不足,因此可以用丢包率衡量算法的可靠性。本文计算了两种网络规模下算法的丢包率,结果如图 7 所示。

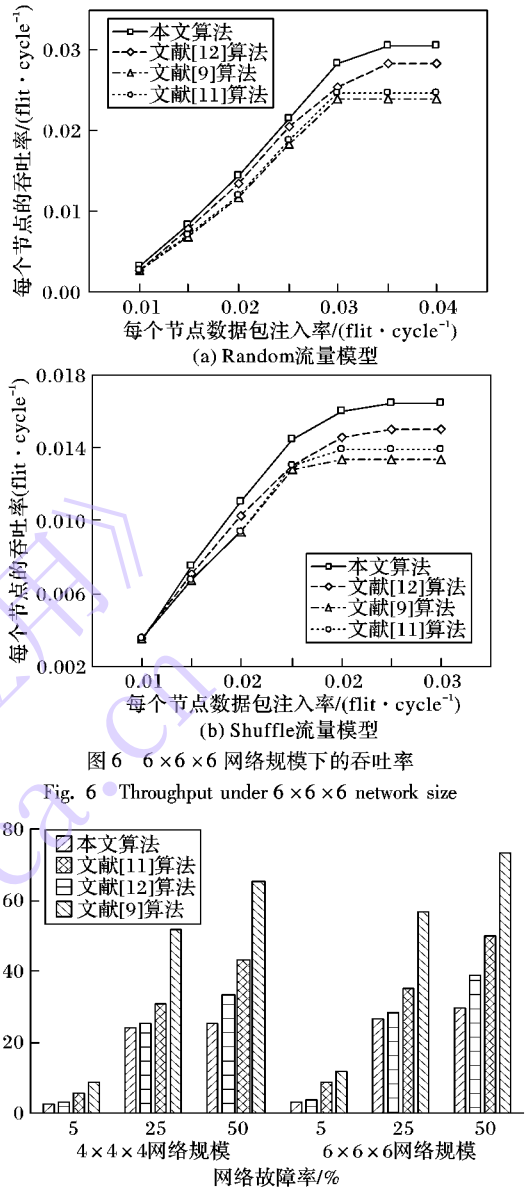

图 6 6×6×6 网络规模下的吞吐率

Fig. 6 Throughput under 6×6×6 network size

图 7 两种网络规模下的丢包率

Fig. 7 Rate of losing packet under two network sizes

文献[12]和文献[11]算法由于没考虑垂直方向的缓存,当层间传输发生拥塞时,便会丢弃数据包。文献[9]算法在传输数据包时静态地选择 TSV,所以当 TSV 发生故障,文献[9]算法不得不丢弃数据包,而且随着故障率的增加,数据包的丢弃变得越来越频繁。本文算法通过综合考虑记录表中的信息,躲避了故障的连接并缓解了拥塞的情况,在故障率为 5% 时,两种网络模型下的丢包率分别为 2.4% 和 2.8%;在故障率为 50% 时,丢包率分别为 25.5% 和 29.5%。

4.3 记录表的开销分析

根据本文记录表的结构,保存记录表所需寄存器大小的计算公式如下:

$$Regbit = \{fault + intrabuff + (pc \times 2 + hop + interbuff) \times 2\} \times 4 \tag{3}$$

其中:*Regbit* 表示寄存器的大小;*fault* 表示链路或路由器是否故障,需要 1 b;*intrabuff* 表示层内邻节点缓存占用信息;

interbuff 已在式(1)中介绍;本文设置缓存深度为 8 flit,所以 *intrabuff* 和 *interbuff* 均需要 3 b;*pc* 表示节点的坐标信息;*hop* 表示当前节点到 TSV 的跳数,其中根据网络规模的不同,*pc* 和 *hop* 需要的位数也不同。

表 4 给出了具体网络规模下保存记录表所需的硬件开销。在层内网络规模为 $N \times N$ 时,文献[11]的硬件开销为 $2N^2$;文献[12]的硬件开销为 24 lb N + 8;本文的硬件开销为 24 lb N + 48。由上述可知,随着网络规模的增加,文献[11]的表开销呈剧烈的增长,而文献[12]和本文的表开销增长缓慢。

表 4　记录表的硬件开销

Tab. 4　Hardware overhead of record table

网络规模	记录表的硬件开销/b		
	文献[11]算法	文献[12]算法	本文算法
4×4	32	56	96
8×8	128	80	120
16×16	512	104	144
32×32	2048	128	168

5　结语

本文主要针对非全互连 3D NoC 的拓扑结构提出基于记录表的自适应单播路由算法。该表不仅记录了 TSV 的位置信息也记录了缓存占用和故障信息,而且在较大的网络规模下比文献[11]算法的表开销要小得多;仿真实验结果表明,本文算法的可靠性优于其他三个算法,并且在平均延时和吞吐率性能指标上均有一定的优势。尤其在 Shuffle 流量模型下,当注入率高于 0.02 时,本文算法的网络还没有达到饱和,而其他两个算法的网络趋于饱和状态。当层内网络规模小于 8×8 时,本文算法的表开销过大,这是本文算法的不足。为了进一步提高非全互连 3D NoC 的网络性能,表结构的优化以及非全互连 3D NoC 的多播路由算法都是以后需要解决的问题。

参考文献(References)

[1] RAHMANI A M, VADDINA K R, LILJEBERG P, et al. Power and area optimization of 3D networks-on-chip using smart and efficient vertical channels[C]// PATMOS 2011: International Conference on Integrated Circuit and System Design: Power and Timing Modeling, Optimization, and Simulation. Berlin: Springer-Verlag, 2011: 278 - 287.

[2] SALAMAT R, EBRAHIMI M, BAGHERZADEH N. An adaptive, low restrictive and fault resilient routing algorithm for 3D network-on-chip[C]// PDP 2015: Euromicro International Conference on Parallel, Distributed and Network-Based Processing. Piscataway, NJ: IEEE, 2015: 392 - 395.

[3] BAHMANI M, SHEIBANYRAD A, PETROT F, et al. A 3D-NoC router implementation exploiting vertically-partially-connected topologies[C]// Proceedings of the 2012 IEEE Computer Society Symposium on VLSI. Washington, DC: IEEE Computer Society, 2012: 9 - 14.

[4] JIANG L, XU Q, EKLOW B. On effective TSV repair for 3D-stacked ICs[C]// DATE 2012: Proceedings of the Conference on Design, Automation and Test in Europe. San Jose, CA: EDA Consortium, 2012: 793 - 798.

[5] DAVIS W R, WILSON J, MICK S, et al. Demystifying 3D ICs: the pros and cons of going vertical[J]. IEEE Design & Test of Computers, 2005, 22(6): 498 - 510.

[6] AHMED A B, ABDALLAH A B. Architecture and design of high-throughput, low-latency, and fault-tolerant routing algorithm for 3D-Network-on-Chip (3D-NoC)[J]. The Journal of Supercomputing, 2013, 66(3): 1507 - 1532.

[7] AHMED A B, ABDALLAH A B. Graceful deadlock-free fault-tolerant routing algorithm for 3D network-on-chip architectures[J]. Journal of Parallel & Distributed Computing, 2014, 74(4): 2229 - 2240.

[8] EBRAHIMI M, CHANG X, DANESHTALAB M, et al. DyXYZ: fully adaptive routing algorithm for 3D NoCs[C]// Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing. Washington, DC: IEEE Computer Society, 2013: 499 - 503.

[9] DUBOIS F, SHEIBANYRAD A, BAHMANI M. Elevator-First: a deadlock-free distributed routing algorithm for vertically partially connected 3D-NoCs[J]. IEEE Transactions on Computers, 2013, 62(3): 609 - 615.

[10] LEE J, CHOI K. A deadlock-free routing algorithm requiring no virtual channel on 3D-NoCs with partial vertical connections[C]// Proceedings of the 7th IEEE/ACM International Symposium on Networks on Chip. Piscataway, NJ: IEEE, 2013: 1 - 2.

[11] 欧阳一鸣, 韩倩倩, 梁华国, 等. 面向非全互连3D NoC可靠通信的分布式路由算法[J]. 计算机辅助设计与图形学学报, 2014, 26(3): 502 - 510. (OUYANG Y M, HAN Q Q, LIANG H G, et al. A distributed routing algorithm for reliable communication in vertically partially connected 3D NoC[J]. Journal of Computer-Aided Design & Computer Graphics, 2014, 26(3): 502 - 510.)

[12] 赵俊宇, 朱珂, 沈剑良, 等. 面向非全互连3D NoC的低开销容错路由算法[J]. 小型微型计算机系统, 2017, 38(4): 791 - 796. (ZHAO J Y, ZHU K, SHEN J L, et al. Low hardware overhead, fault-tolerant routing algorithm in vertically partially connected 3D NoC[J]. Journal of Chinese Computer Systems, 2017, 38(4): 791 - 796.)

[13] AHMED A B, ABDALLAH A B. Adaptive fault-tolerant architecture and routing algorithm for reliable many-core 3D-NoC systems[J]. Journal of Parallel & Distributed Computing, 2016, 93/94(C): 30 - 43.

This work is partially supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China (2016ZX01012101), the National Natural Science Foundation of China (61572520, 61521003).

SUN Meidong, born in 1993, M. S. candidate. His research interests include routing algorithm for 3D NoC.

LIU Qinrang, born in 1975, Ph. D., research fellow. His research interests include broadband information network, NoC design.

LIU Dongpei, born in 1985, Ph. D. His research interests include chip verification and testing.

YAN Binghao, born in 1994, M. S. candidate. His research interests include traffic identification, intrusion detection.