

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04L 12/56

H04Q 3/64



[12] 发明专利说明书

[21] ZL 专利号 00100918.4

[45] 授权公告日 2003 年 10 月 22 日

[11] 授权公告号 CN 1125548C

[22] 申请日 2000.1.7 [21] 申请号 00100918.4

[71] 专利权人 威盛电子股份有限公司

地址 台湾省台北县新店市

[72] 发明人 陈任凯 林郁如 刘建华

审查员 孙玉芳

[74] 专利代理机构 北京市柳沈律师事务所

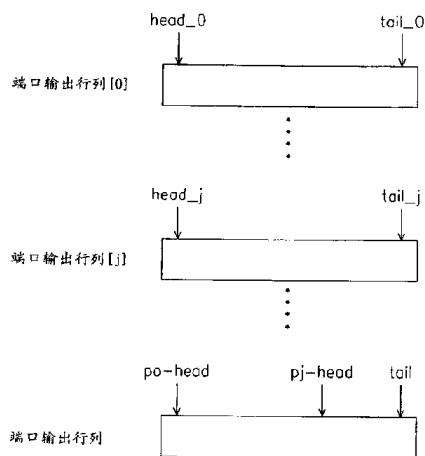
代理人 杨 梧 朱 勤

权利要求书 2 页 说明书 12 页 附图 4 页

[54] 发明名称 依序转发包的输出排队方法

[57] 摘要

一种网路包输出排队方法，可应用于交换式包传输装置。各输出端口是对应至一端口输出排队且共享一整体输出排队。各端口输出排队与整体输出排队是配置先入先出(FIFO)方块，FIFO 方块具有多个 FIFO 节点。当判断所接收包的形式与目的端口，根据包形式而设定端口输出排队或整体输出排队中的 FIFO 节点的对应字段。根据所对应字段而决定将包从输出排队传送出去或略过。



ISSN 1008-4274

1. 一种网路包的输出排队方法，应用于具有多个端口端的一交换式包传输装置，各个端口是对应至一端口输出排队且共享一整体输出排队，各个
5 端口输出排队与该整体输出排队是配置一先入先出块，该先入先出块具有 Nb 个先入先出节点与一个指向字段，Nb 为正整数，各个先入先出块节点具有一第一字段与一第二字段，其特征是该网路包的输出排队方法包括：

接收一包，并且判断该包的形式与其目的端口；

10 当该包是一单向传播包时，设定其目的端口的该端口输出排队的一第一先入先出节点；

当该包是一多向传播包时，根据该多向传播包的目的端口设定该广播输出排队的一第二先入先出节点的该第二字段，并设定各个输出排队中的最后一该第一先入先出节点中的该第一字段与该第二字段；

15 读取对应该单向传播包的该第一先入先出节点中的该第一字段与该第二字段的值，并将该单向传播包传送出去；以及

根据该第一先入先出节点中的该第一字段与该第二字段的值，以及该第二先入先出节点中的该第二字段，决定将该多向传播包送出或略过。

2. 如权利要求 1 所述的输出排队方法，其特征是各个先入先出节点还具有一第三字段，当该包为一单向传播包时，该第三字段的值为 0，当该包
20 为一多向传播包时，该第三字段的值为 1。

3. 如权利要求 1 所述的输出排队方法，其特征是设定该第一先入先出节点中的该第一字段与该第二字段的步骤包括：

当该输出排队不为该多向传播的目的端口之一时，且该第二字段的值为 0，将该第一字段的值加 1；

25 当该输出排队不为该多向传播的目的端口之一时，且该第二字段的值不为 0，将该第二字段的值加 1；以及

当该输出排队为该多向传播的目的端口之一时，且该第二字段的值加 1

。

30 4. 如权利要求 1 所述的输出排队方法，其特征是读取该第一先入先出节点中的该第一字段与该第二字段的值的步骤包括：将该第一字段与该第二字段的值读取并写入至一第一寄存器与一第二寄存器。

5. 如权利要求4所述的输出排队方法,其特征是将该多向传播包送出或略过的步骤包括:

根据该第一寄存器的值 m , m 为大于或等于 0 的整数,该端口输出排队从该整体输出排队略过 m 个多向传播包; 以及

5 根据该第二寄存器的值 n , n 为大于或等于 0 的整数,该端口输出排队从该整体输出排队略过或送出 n 个多向传播包。

6. 如权利要求5所述的输出排队方法,其特征是当该第二先入先出节点的该第二字段显示该端口输出排队为该多向传播包的目的端口之一时,是将该 n 个多向传播包之一送出,否则将此多向传播包略过。

10 7. 如权利要求6所述的输出排队方法,其特征是将该多向传播包送出的步骤还包括:

当该第二先入先出节点的该第二字段只有一个位为 1 时,释放该第二先入先出节点;

15 当该第二先入先出节点的该第二字段有多个位为 1 时,将该端口的相关位清除; 以及

当该整体输出排队的该先入先出块内的所有先入先出块节点都被释放后,将该先入先出块释放至一自由列中。

8. 如权利要求1所述的输出排队方法,其特征是将该单向传播包传送出去的步骤还包括:

20 将该第一先入先出节点释放; 以及

当该端口输出排队的该先入先出块内的所有先入先出节点都被释放后,将该先入先出块释放至一自由列中。

9. 如权利要求1所述的输出排队方法,其中各个端口输出排队包括一首指向字段,其是指向各个端口输出排队中的这些先入先出节点之一。

25 10. 如权利要求1所述的输出排队方法,其特征是各个端口输出排队包括一尾指向字段,其是指向各个端口输出排队中的这些先入先出节点之一。

11. 如权利要求1所述的输出排队方法,其特征是该整体输出排队包括多个首指向字段,与尾指向字段,各个首指向字段是一一对应至这些端口输出排队,这些首指向字段是指向各个端口输出排队在该整体输出排队中所对应的这些先入先出节点之一,该尾指向字段是指向该整体输出排队的这些先入先出节点之一。

30

依序转发包的输出排队方法

5 本发明是有关于一种支援依序(in-sequence)转发(forwarding)网路包(packet)的输出排队方法(output queuing scheme), 且特别是有关于一种在乙太交换网路(Ethernet switch network)中支援依次转发包的输出排队方法。

在一般的乙太交换网路(Ethernet switch network)中, 内部是可能包括数十个端口(port)甚至数百个端口。而交换网路中的端口包括乙太端口(Ethernet
10 port)与中央处理器端口(CPU port)。

在端口彼此传输数据的过程(如乙太端口传送数据给乙太端口, 或中央处理器端口传送数据给乙太端口), 数据是以包(packet)形式输送。所谓包是指独立的讯息, 讯息中的成分包含地址、控制与数据信号。包能作为一个单元在通讯网路中传输。

15 包可分为三种: (1)单向传播包(unicast packet): 其目的端口为单一目的端口, 这种包称为单向传播包; (2)多向传播包(multicast packet): 其目的端口为多个目的端口, 但未包括所有端口, 这种包称为多向传播包; (3)广播包(broadcast packet): 其目的端口为所有端口, 这种包称为广播包。比如, 在一个17端口的交换网路中, 一个广播包具有16个目的端口。

20 在乙太交换网路中, 处理包输入的控制模块称为“输入控制(input control)”。而处理包输出的控制模块则称之为“输出控制(output control)”。各端口具有其相关的输入控制与输出控制。输入控制与输出控制间是彼此独立运作。

在包转发至某一端口后, 该包可视为放置于该端口的端口输出排队中
25 (port output queue)中。实际上, 输出排队是将包缓冲单元(packet buffer)串链而成。各包缓冲单元可包括最大为1518字节(byte)的包。一般常将包缓冲单元与串链节点(link node)分隔开。包缓冲单元如上所述, 是用以储存包。串链节点储存其他数据, 如包大小(packet size)、目的端口位屏蔽(destination-port bit mask)、指向字段(pointer)等。

30 比如说, 某一端口的输入控制在处理该端口的所有输入包时, 包括下列

步骤:

- (1)利用查表法得知该包的目的端口屏蔽(destination port mask);
- (2)配置(allocate)空白包缓冲单元以储存此输入包; 以及
- (3)将此输入包加入(enqueue)至此端口的输出排队。

5 而某一端口的输出控制在处理该端口的所有输出包时, 包括下列步骤:

- (1)从其输出排队将一输出包退出排队(dequeue);
- (2)将此输出包送出; 以及
- (3)将此输出包的包缓冲单元释放(release)出。

10 对包括16个以太网端口(Ethernet port)与1个中央处理器端口(CPU port)的交换集成电路(IC)而言, 目的端口位屏蔽包括17位, 其用以指出哪些端口为此包的目的端口。

15 下一指向字段是指出在输出排队或自由列(free list)中的下一节点为何。对具有共享存储器架构的交换IC而言, 开始时, 所有自由包缓冲单元是位于整体自由列(global free list)内, 而所有输出排队是空的。所谓自由列是指, 由所有自由节点串列而成, 而整体自由列是指将所有自由包缓冲单元串列而成。

然而, 在将包(如单向传播包/多向传播包/广播包)转发至目的端口时, 需采用特殊演算法, 依照先入先出(First In First Out, FIFO)法来实施“依序转发”。对于支援依序转发的交换IC而言, 其输入控制包括下列步骤:

- 20 (1)借由查表以得知包的目的端口位屏蔽;
- (2)配置(allocate)自由包缓冲单元, 以储存输入包; 以及
- (3)根据所设定的目的端口位屏蔽, 将此输入包加入至具有最小端口辨认码(ID)的目的端口输出排队。

而输出控制则执行下列步骤:

- 25 (1)将一输出包从其输出排队退出排队;
- (2)送出此输出包;
- (3)清除该目的端口位屏蔽的相关位; 以及
- (4)如果该目的端口位屏蔽全为0, 将此输出包的包缓冲单元释放至自由列。否则, 根据最新的目的端口位屏蔽, 将此输出包加入至具最小端口辨认
- 30 码的目的端口的输出排队。

然而, 此“依序转发”法的缺点在于其转发期较长。因而, 在已知技术

中，已发展出转发期较短的依序转发法。下面将简述这已知技术。

此已知技术称之为“串链节点复制法”。在此已知技术中，所用到的串链节点共有两种—一种是单向传播串链节点(unicast link node)，另一种是多向传播串链节点(multicast link node)。在下面，是以64位的串链节点做说明。

5 图1a与图1b是分别绘示应用于此已知技术中的单向传播串链节点与多向传播串链节点的架构图。

一个64位的单向传播串链节点是一对一关联至一个约1.5k大小byte的包缓冲单元。单向传播串链节点是固定地关联至包缓冲单元。也就是，第j个单向传播串链节点可视为固定地关联至第j个包缓冲单元。多向传播串链节点则动态地关联至储存多向传播包或广播包的包缓冲单元。

10 在此种已知技术中，当转发一个单向传播包时，需要一个包缓冲单元以及一个单向传播串链节点。包缓冲单元当然是用于储存待转发的单向传播包。

而当要转发一个多向传播包时，假设其目的端口为k个(此多向传播包亦可称为具有k个目的端口的多向传播包)，则需要一个包缓冲单元、一个单向传播串链节点以及k个多向传播串链节点。至于在具有N个端口的交换IC内，转发一个广播包则类似于转发一个多向传播包。转发一个多向传播包/广播包时需要一个包缓冲单元，一个单向传播串链节点，以及N-1个多向传播串链节点。至于这些包缓冲单元、单向传播串链节点以及多向传播串链节点的作用将于底下作说明。

欲转发一个K个目的端口的多向传播包时，输入控制是执行下列步骤：

- (1)查表来得知目的端口的位屏蔽；
- (2)配置一个原本于自由列中的包缓冲单元，以储存此待转发包；
- (3)设定单向传播串链节点内的目的端口位屏蔽。如前述般，此单向传播串链节点是一对一固定关联至此包缓冲单元；
- (4)配置动态关联至此包缓冲单元的k个自由多向传播串链节点；以及
- (5)分别将此K个自由多向传播串链节点加入(enqueue)至k个目的端口的输出排队内。

至于欲转发一个单向传播包时，输入控制是执行下列步骤：

- 30 (1)查表来得知目的端口的位屏蔽；
- (2)配置一个原本于自由列中的包缓冲单元，以储存此待转发包；

(3)配置关联至此包缓冲单元的一个单向传播串链节点；以及

(4)将此单向传播串链节点加入目的端口的输出排队内。

而输出排队有节点要释放出时，输出控制则包括：

(1)将一个输出包从其输出排队退出(dequeue)；

5 (2)将此输出包送出；

(3)如果是为单向传播串链节点，则将此输出包的包缓冲单元释放至自由列中；以及

(4)如果是为多向传播串链节点，且在相关的单向传播串链节点中的目的端口位屏蔽只有一个位为1，则将此包缓冲单元释放(release)至自由列内，并
10 释放此多向传播串链节点；以及

(5)如果是为多向传播串链节点，且在相关的单向传播串链节点中的目的端口位屏蔽不只有一个位为1，则将相关的单向传播串链节点内的目的端口位屏蔽内的相关位清除(就是设成0)，并释放此多向传播串链节点。

在输出控制的步骤(4)中，当目的端口位屏蔽只有一个位为1时，代表着
15 此多向传播包已经完成转发至其余目的端口的转发。所以，可将此包缓冲单元与单向传播串链节点释放至自由列内。

接着，利用上述来说明图1a与图1b的架构。在图1a中，单向传播串链节点的架构是包括三个字段(Three fields)，此三个字段分别包括：指向字段(pointer)；端口屏蔽(port mask)；以及其他数据。

20 当此单向传播串链节点是应用于转发一单向传播包时，其指向字段是用以指向其目的端口的下一单向传播包。

而当单向传播串链节点是应用于转发一多向传播包时，其端口屏蔽的位是用以指出该多向传播包的目的端口。当某一目的端口已接收到此包，且要将此包退出其输出排队时，相关于此目的端口的位是被重设。

25 图1b中的多向传播串链节点是包括第一指向字段、第二指向字段以及其他字段。类似地，第一指向字段是用以指向输出排队中的下一串链节点。而第二指向字段是用以指向相关的单向传播串链节点。由于单向传播串链节点与多向传播节点的类别区分是以其所在的地址，故不需额外类别字段来做记录。

30 一般而言，在衡量此种依序转发输出排队法时，其衡量标准是包括：配置次数、释放次数、加入排队次数以及退出排队次数。下表1列出此已知技

术各操作次数。

表 1

k 个目的端口的多向传播包	次数
配置次数	1+k 次
释放次数	1+k 次
加入排队次数	k
改变次数	读取: k 次 写入: k-1
退出排队次数	k 次

从表 1 可看出, 如果多向传播包是要传送至 k 个目的端口, 则此已知技术总共需要对存储器进行(6k+1)个存取操作才能完成转发操作。如此多的存取次数将使得存储器频宽需求提高。

有鉴于此, 本发明的主要目的就是在提供一种能达成依次包转发的输出排队法。本发明是提出先入先出(FIFO)节点共享法(sharing scheme), 可支援在乙太交换网路中的依次包转发。

根据本发明的目的, 提出一种支援依序转发包的输出排队方法, 是应用于具有多端口的交换式包传输装置。各端口是对应至一端口输出排队且共享整体输出排队。各端口输出排队与该整体输出排队是配置一先入先出(FIFO)方块, 该 FIFO 方块(FIFO block)内有多个 FIFO 节点, 各 FIFO 节点具有略过字段与多向传播计数字段(端口屏蔽字段)。该输出排队方法包括: 判断所接收包的形式与目的端口; 根据包形式设定目的端口输出排队/整体输出排队的对应 FIFO 节点中的略过字段与多向传播计数字段(端口屏蔽字段); 以及根据略过字段与多向传播计数字段(端口屏蔽字段)的值, 决定将包从端口输出排队/整体输出排队传送出去或略过。

为了让本发明的目的、特征、和优点能更明显易懂, 下文特举一较佳实施例, 并配合附图, 作详细说明如下:

附图简要说明:

图 1a 是已知技术的单向传播串链节点的数据结构图;

图 1b 是已知技术的多向传播串链节点的数据结构图;

图 2 是绘示本实施例的 FIFO 方块的数据结构图;

图 3 是绘示本实施例的 FIFO 节点的数据结构图;

图 4 是绘示本实施例的输出排队的数据结构；
图 5 是绘示本实施例的输入控制流程图；以及
图 6 是绘示本实施例的输出控制流程图。

较佳实施例

5 本实施例是将先入先出(First In First Out, FIFO)方块来做为输出排队的构成元件。如图 2 所示, 一个 FIFO 方块包括 Nb 个 FIFO 节点以及一个指向下一 FIFO 方块的指向字段。为简化起见, 将第 k 个 FIFO 方块的第 j 个 FIFO 节点对应至一包参数(k, j)。如此一来, 各 FIFO 节点可隐藏性对应一个包缓冲单元。比如, 具有包参数(k, j)的 FIFO 节点是对应至第(k×Nb+j)个包
10 缓冲单元。

开机时, 所有 FIFO 方块是串链成一个自由列。在系统开机后, 排队控制(queue control)将配置各一个 FIFO 方块给各端口输出排队(port output queue)与一个整体输出排队(global output queue), 使得各端口输出排队与该整体输出排队具有 Nb 个自由包缓冲单元。顾名思义, 各端口有自己的端口输出排队, 存放于端口输出排队内的是单向传播包。而整体输出排队则是用以存放多向传播包/广播包。这些端口输出排队是共享一个整体输出排队, 所以本实施例可称为 FIFO 节点共享法。
15

当所配置的 FIFO 方块内的所有 FIFO 节点都被占去(也就是用以储存包)后, 排队控制是立刻配置一个空的 FIFO 方块。而当所配置的 FIFO 方块内的所有 FIFO 节点都被释放(或退出排队)后, 排队控制是立刻将此 FIFO 方块释放至自由列中。
20

此外, 本实施例引入排队控制(queue control)的观念。排队控制与输入控制、输出控制间的关系类似于主从(server/client)架构。当输出控制提出退出排队的要求, 排队控制是提供退出排队的操作给输出控制。同样地, 当输入控制提出加入排队的要求时, 排队控制是提供加入排队的操作给输入控制。
25

对每个端口输出排队, 其在排队控制中具有一个“输出略过计数(output skip count)”寄存器、一个“输出多向传播计数(output multicast count)”寄存器、一个“输入略过计数(input skip count)”寄存器与一个“输入多向传播计数(input multicast count)”寄存器。

30 “输出略过计数”寄存器是用以代表在目前此一单向传播包送出后要有几个多向传播包被略过(skip)。而“输出多向计数”寄存器则代表在该单向传

播包之后有几个多向传播包要进行退出排队处理,来确认是否需要退出排队。要注意的是,比如,当输出略过计数寄存器内的值为3,而输出多向传播计数寄存器内的数据为6时,代表在此单向传播包与下一单向传播包间有9个多向传播包。而此9个多向传播包中,前3个多向传播包是要被略过,而后6个多向传播包则是要进行退出排队处理。

“输入略过计数”寄存器是用以代表在先前输入的单向传播包之后要有几个多向传播包要被略过。而“输入多向计数”寄存器则代表在该单向传播包之后有几个多向传播包要被退出排队。要注意的是,比如,当输入略过计数寄存器的值为3,而输入多向传播计数寄存器内的值为6时,代表在此单向传播包与下一单向传播包间有9个多向传播包。而此9个多向传播包中,前3个多向传播包是要被略过,而后6个多向传播包则是要被退出排队。

图3绘示一个64位FIFO节点的数据结构图。位0-1的类别字段(cast type)是用以指出该FIFO节点所对应的包缓冲单元内所储存的是否为单向传播包。如果该类别字段为0则代表该包为单向传播包,为1则为多向传播包。

当类别字段为0时,其下一字段是储存“多向传播计数”。多向传播计数字段是代表紧邻着这些被略过多向传播包后而要被退出排队的多向传播包数量。比如说,当略过字段为3,而多向传播计数字段为6时,代表在此单向传播包与下一单向传播包间有9个多向传播包。而此9个多向传播包中,前3个多向传播包是要被略过,而后6个多向传播包则是要被退出排队。

当类别字段为1时,其下一字段是储存“目的端口屏蔽”。此目的端口屏蔽字段意思类似于已知技术的目的端口屏蔽字段,其用以指出该多向传播包的目的端口。在本实施例中,转发一个单向传播包/多向传播包时只需要一个包缓冲单元以及一个FIFO节点。

第18-34位的略过计数(skip count)字段则是代表在此单向传播包前有几个多向传播包要被略过,非被退出排队。结合此略过计数字段与FIFO节点的连续性,要将一个多向传播包于整体输出排队中退出排队的次数将可少于N次(N为全部端口个数)。

开始时,“输入略过计数”寄存器与“输入多向传播计数”寄存器的值皆为0。对于一 $k(k=2\sim N)$ 个目的端口的多向传播包而言,排队控制是执行下列步骤:

(1)如果第 j 个端口不为该多向传播包的目的端口,则排队控制将视该第

j 个端口所对应的输入多向传播计数寄存器(可简称为第 j 个输出多向传播计数寄存器)的值而有不同操作。若第 j 个输出多向传播计数寄存器的值为 0, 将该第 j 个所对应的输入略过计数寄存器(可简称为第 j 个输出略过计数寄存器)的值增加 1。若该第 j 个输出多向传播计数寄存器的值不为 0, 将该第 j 个输出多向传播计数寄存器的值增加 1。

(2)如果第 j 个端口为该多向传播包的目的地端口, 则将第 j 个输入多向传播计数寄存器的值增加 1。

当有一个单向传播包要加入某一端口的输出排队中时, 该单向传播包后面可能会跟随几个多向传播包。直到有下一个单向传播包要加入该端口的输出排队后, 便可确认前一个单向传播包是有几个多向传播包跟着。

当此端口输出排队中的下一个单向传播包出现时, “输入略过计数”寄存器的值是设定到此单向传播包的“略过计数”字段内。此外, “输入多向传播计数”寄存器的值是设定到此单向传播包的“多向传播计数”字段内, 接着“输入略过计数”寄存器与“输入多向传播计数”寄存器是皆重设为 0。

而当要将一个 FIFO 节点从端口输出排队退出时, 其“略过计数”字段内的值是写入至该端口所对应的“输出略过计数”寄存器内。而“多向传播计数”字段内的值是写入至该端口所对应的“输出多向传播计数”寄存器内。

图 4 是绘示本实施例的各输出排队的数据结构, 其绘示各输出排队的首指向字段(head pointer)与尾指向字段(tail pointer)。在图 4 中, 端口输出排队 $j(j=0\sim 15)$ 是代表第 j 个端口的输出排队, $head_j$ 与 $tail_j$ 是分别代表端口输出排队 j 的首指向字段与尾指向字段。首指向字段 $head_j$ 是用以指出端口输出排队 j 的下一个待退出排队的 FIFO 节点。尾指向字段 $tail_j$ 则用以指出端口输出排队 j 的下一个待加入排队的 FIFO 节点。而整体输出排队中的各首指向字段 pj_head 是一一对一对应至各端口输出排队。各首指向字段 pj_head 是指出各端口输出排队在整体输出排队中的下一待退出排队的 FIFO 节点。这些指向字段 pj_head 的位置与操作是互相独立的。而整体输出排队中的尾指向字段 $tail$ 是由所有端口输出排队所共享, 以将多向传播包/广播包加入至整体输出排队中。

请参考图 5 的输入控制流程图。首先, 查表得到该包的形式与目的地端口

屏蔽，如步骤 501 所示。

向排队控制要求锁定(lock)一个 FIFO 节点，如步骤 502 所示。如果该包是为单向传播包，则是锁定其目的端口的端口输出排队中的 FIFO 节点。如此包为多向传播包，则是锁定整体输出排队中的 FIFO 节点。

5 接着，设定被锁定 FIFO 节点的相关字段，如步骤 503 所示。当此包为单向传播包时，则将相关 FIFO 节点的类别字段设为 0，并设定排队控制是将“输入多向传播计数”寄存器的值与“输入略过计数”寄存器的值设定至前一 FIFO 节点的“多向传播计数”字段与“略过计数”字段内。如果此包为多向传播包，则将其类别字段设为 1，并设定其端口屏蔽字段。

10 将此 FIFO 节点加入至端口输出排队或整体输出排队中，如步骤 504 所示。此 FIFO 节点加入的位置是由尾指向字段(tail 或 tail_j)所指定。各排队的尾端 FIFO 节点皆存于 IC 内部，以减少存储器存取次数。当一 FIFO 节点新加入至此排队时，原尾端 FIFO 节点须写入其对应的存储器位置。

15 如果端口输出排队中有包需要被退出，则输出控制将会执行如图 6 的流程图所示步骤。

首先，输出控制将一单向传播包从端口输出排队中退出排队，如步骤 601。接着，将此单向传播包送出，如步骤 602。

20 接着，将储存此单向传播包的包缓冲单元释放至自由列，并且释放对应到此包缓冲单元的 FIFO 节点，如步骤 603。要注意的是，虽然在此步骤有释放此 FIFO 节点的操作，但是排队控制是等到整个 FIFO 方块中的所有 FIFO 节点都被释放后，该 FIFO 方块才会被释放。

接着，继续进行有关于多向传播包的转发/退出排队。输出控制先略过所有待略过的多向传播包。其过程如下叙般。

25 排队控制是判断输出略过计数寄存器的值是否为 0，如步骤 604 所示。“输出略过计数”寄存器的值代表此输出控制要略过几个多向传播包。当输出略过计数寄存器的值不为 0 时，下一步骤是为步骤 605。当输出略过计数寄存器的值为 0 时，下一步骤是为步骤 607。

30 在步骤 605 中，从整体输出排队中略过一个包缓冲单元。接着，将输出略过计数寄存器的值减 1，如步骤 606 所示。如流程图所示，重复步骤 604、605 与 606，直到输出略过计数寄存器的值变成 0，才接着执行下一步骤 607。

在步骤 607 中，排队控制判断“输出多向传播计数”寄存器的值是否为 0。如果为 0，则输出控制到此告一段落，接着进行处理下一包组；如不为 0，则执行步骤 608。多向传播计数寄存器的值不为 0 代表，输出控制可能要从整体输出排队将数个多向传播包退出排队。如果多向传播计数寄存器的值为 0 代表在略过数个多向传播包后，后续已无多向传播包。所以输出控制对此包组已告一段落，等待下一包组。

在步骤 608 中，排队控制是检查整体输出排队的该 FIFO 节点内的目的端口屏蔽的相关位，来确认输出控制是否要将此多向传播包从端口输出排队中退出排队。当该相关位为 1 时(此端口为该多向传播包的目的端口)，下一步骤是为 609。若当该相关位为 0 时(此端口不为该多向传播包的目的端口)，下一步骤是为 614。

在步骤 609 中，排队控制是告知输出控制要将此多向传播包从该端口输出排队中退出排队。

接着，在步骤 610 中，将被退出排队的此多向传播包送出。

在步骤 611 中，要判断在整体输出排队中的 FIFO 节点的目的端口屏蔽是否只有一个位值为 1。如果不只有一个位值为 1，则接着进行步骤 612。反之则进行步骤 613。

在步骤 612 中，清除(重设)此 FIFO 节点中的目的端口屏蔽中的相关位。此操作代表着，此目的端口输出排队已完成将此多向传播包退出排队/转发。

在步骤 613 中，释放储存此多向传播包的包缓冲单元至自由列中，并释放整体输出排队中的 FIFO 节点。由于当目的端口屏蔽字段只有一个位值为 1 时，代表其余的目的端口皆已完成将此多向传播包退出排队或已无此多向传播包的目的端口。所以，可以将该包缓冲单元释放至自由列中，并将此 FIFO 节点释放。同样地，虽然此步骤 613 中有释放 FIFO 节点的操作。然而，在整体输出排队中，是等到整个 FIFO 方块中的所有 FIFO 节点都被释放后，该 FIFO 方块才会被释放至自由列中。

在步骤 614 中，将该端口的输出多向传播计数寄存器的值减 1。如果步骤 608 后是进行步骤 614，代表此端口不为该多向传播包的目的端口，所以排队控制并不告知输出控制要做退出排队的操作。亦即可视为输出控制略过此多向传播包。

输出控制是重复步骤 607-614，直到将所有多向传播包完成退出排队(或略过)操作。

在图 6 的流程图中，为方便解释，是将这些步骤显示成依序进行。实际上，也可将某些步骤以，比如流水线(pipeline)方式进行，以增快进度。比如，
5 当一个单向/多向传播包长度很长时，这时要将此长包转发便需较长时间。故可在送出一包同时进行下一个包的退出排队操作来加快速度。

表 2 列出本发明与已知技术在性能上的比较。

表 2

k 个目的端口的多向传播包	已知技术	本发明
配置次数	1+k 次	1/Nb
释放次数	1+k 次	1/Nb
加入排队次数	k 次	1
改变次数	读取: k 次 写入: k-1	读取: k 次 写入: k-1 次
退出排队次数	k 次	≤N

在表 2 中，k 是代表多向传播包的端口数量，而 Nb 则是一个 FIFO
10 方块中的 FIFO 节点数量，而 N 则是一个交换式 IC 内的所有端口的数量。

由表 2 可看出，本发明的最大优点在于配置与释放次数上的改善，由已知的(1+k)次变成(1/Nb)次。此外，本发明还利用“略过计数”来减少部分退出排队次数。

因此，本发明特征之一是利用共享观念来导入一个整体输出排队，该整体
15 输出排队是为所有端口输出排队共享。所有多向传播包/广播传播包是储存于该整体输出排队中。各端口输出排队若有需要可至整体输出排队中拿取多向传播包。

本发明的另一特征是利用主从架构导入一个排队控制。该排队控制是提供加入排队操作给输入控制，以及提供退出排队操作给输出控制。

20 本发明的另一特征是利用 FIFO 方块来做为组成端口输出排队与整体输出排队的单位。一个 FIFO 方块是包括 Nb 个 FIFO 节点与一个指向字段。利用同一 FIFO 方块内的 FIFO 节点间的连续性，本发明在存取次数可有良好的改善。

综上所述，虽然本发明已以较佳实施例揭露如上，然其并非用以限定本

发有，本领域的技术人员，在不脱离本发明的精神和范围内，当可作各种的更动与润饰，因此本发明的保护范围以本发明的权利要求范围为准。

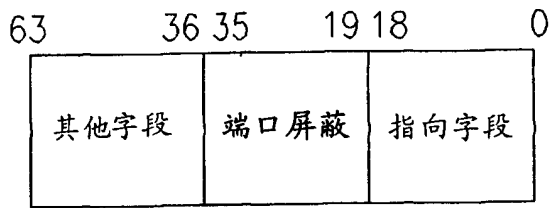


图 1A

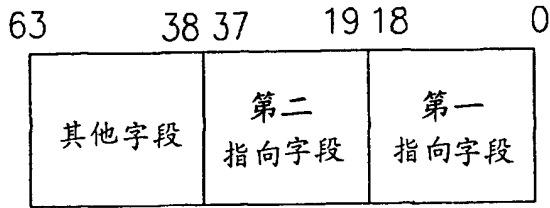


图 1B

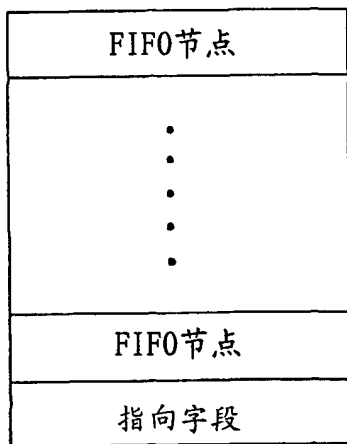


图 2

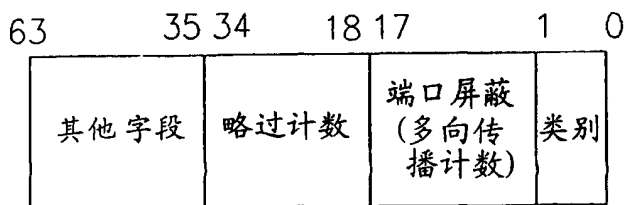


图 3

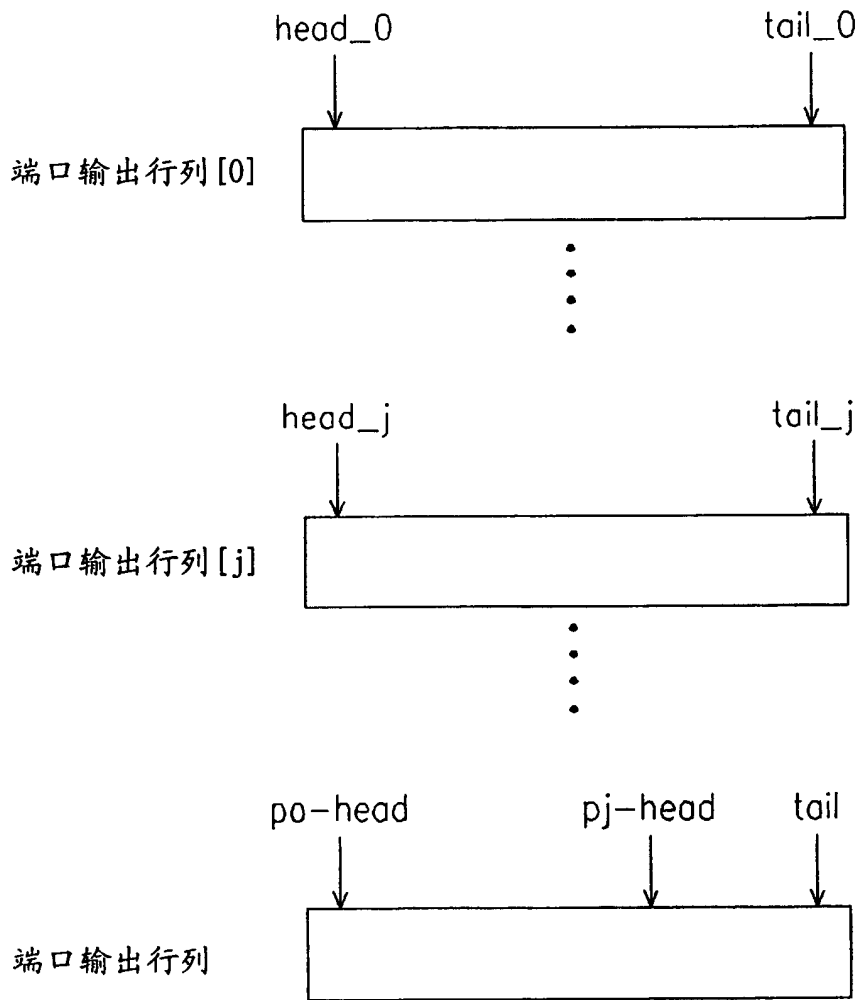


图 4

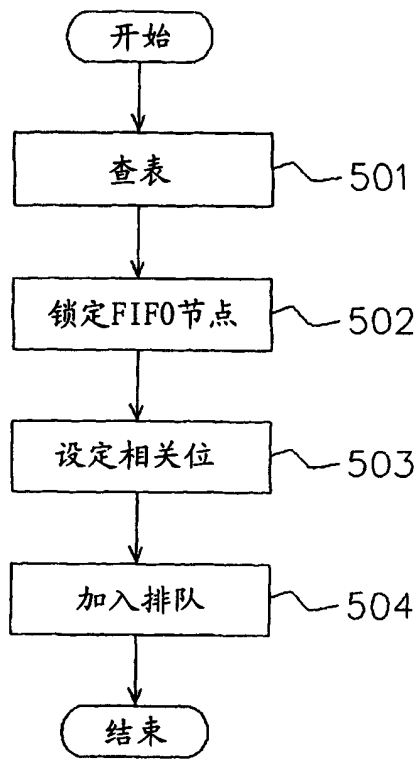


图 5

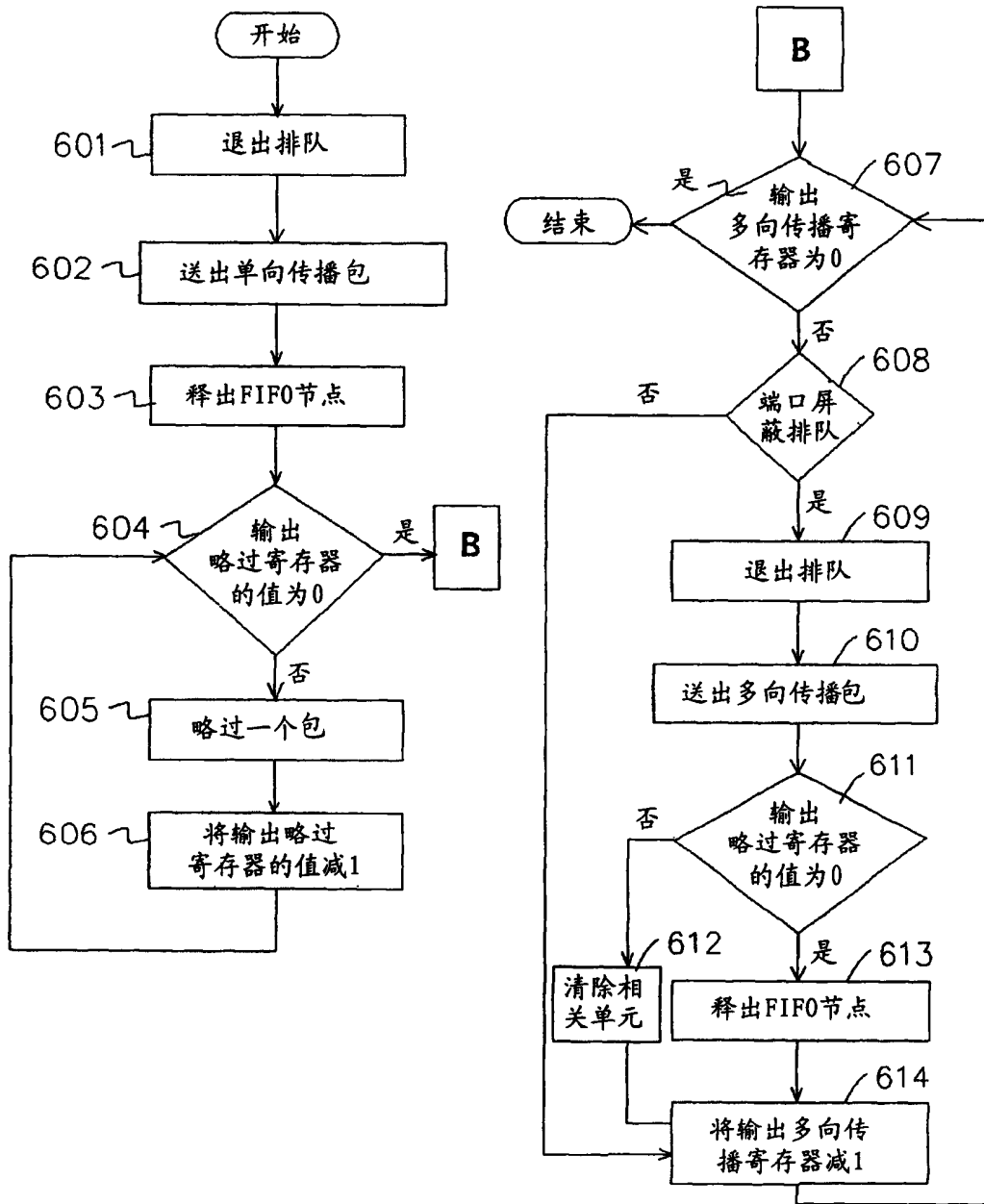


图 6