

(12) 发明专利申请

(10) 申请公布号 CN 102006142 A

(43) 申请公布日 2011.04.06

(21) 申请号 201010269329.X

(22) 申请日 2010.08.31

(30) 优先权数据

2125/CHE/2009 2009.09.02 IN

(71) 申请人 艾格瑞系统有限公司

地址 美国宾夕法尼亚

(72) 发明人 A · K · 西达奥

R · K · 拉伊卡尔

(74) 专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

代理人 陈华成

(51) Int. Cl.

H04L 1/00 (2006.01)

H04B 1/06 (2006.01)

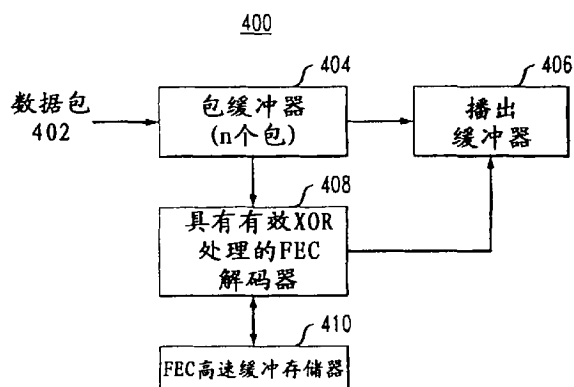
权利要求书 1 页 说明书 8 页 附图 6 页

(54) 发明名称

用于基于误差保护包的帧的接收器

(57) 摘要

本发明公开了一种用于基于误差保护包的帧的接收器。在一个实施例中，公开了一种采用实时传输协议 (RTP) 和前向纠错 (FEC) 的针对媒体包的帧的接收器。该接收器包括包缓冲器和 FEC 解码器。在该包缓冲器接收到包之后，该 FEC 解码器读取包，并且作为 FEC 处理的一部分，对包执行 XOR 运算，而不等待接收整个帧，或者实际上不等待接收该帧的任何后续包。累积 XOR 运算结果，直到接收到足够的包为止，以重构该帧中的丢失包。因为紧接在接收到包之后执行 XOR 运算，而没有等待后续包导致的任何延迟，所以，该接收器具有非常低的等待时间，并且包缓冲器可以相对较小。



1. 一种用于在接收器中执行纠错 (EC) 处理的方法,该方法包括:
 - (a) 顺次接收包括三个或更多个数据包和一 EC 包的第一帧的包的子集;以及
 - (b) 对所述第一帧的包的所述子集执行 EC 处理,以重构所述第一帧的至少一个包,其中,所述 EC 处理在接收到包的整个所述子集之前启动。
2. 根据权利要求 1 所述的方法,其中,所述 EC 处理包括在接收到所述子集中的每一个包时对其进行处理,而不等待接收所述子集中的后续包。
3. 根据权利要求 2 所述的方法,其中,对于所述子集中的至少一个已接收包,所述 EC 处理包括:
 - (b1) 对所述已接收包和存储在存储器中的值执行数学运算;以及
 - (b2) 存储所述数学运算的结果,作为存储在所述存储器中的值,以用于随后的数学运算。
4. 根据权利要求 3 所述的方法,其中,对于所述子集中的最后的接收包,所述 EC 处理包括:
 - (b3) 对最后的接收包和存储在所述存储器中的值执行最后的数学运算;和
 - (b4) 基于所述最后的数学运算的结果重构所述第一帧的丢失数据包。
5. 根据权利要求 1 所述的方法,还包括:
 - (c) 顺次接收包括三个或更多个数据包和一 EC 包的第二帧的包的子集;以及
 - (d) 对所述第二帧的包的所述子集执行 EC 处理,以重构所述第二帧的至少一个包,其中,对所述第二帧的 EC 处理在完成对所述第一帧的 EC 处理之前启动。
6. 根据权利要求 5 所述的方法,其中,对所述第一帧的 EC 处理在启动对所述第二帧的 EC 处理之前启动。
7. 根据权利要求 1 所述的方法,还包括:
 - (a1) 在包缓冲器中存储所述第一帧的包的所述子集中的每一个包;
 - (a2) 在执行对每一个包的 EC 处理之前从所述包缓冲器读取该包;以及
 - (c) 在完成对所述第一帧的包的所述子集中的每一个包的 EC 处理之后,将重构的包存储在所述包缓冲器中。
8. 一种接收器,包括:

包缓冲器,适于顺次接收包括三个或更多个数据包和一纠错 (EC) 包的第一帧的包的子集;以及

EC 解码器,适于对所述第一帧的包的所述子集执行 EC 处理,以重构所述第一帧的至少一个包,其中,所述 EC 处理在接收到包的整个所述子集之前启动。
9. 一种用于在接收器中执行纠错 (EC) 处理的设备,该设备包括:
 - (a) 用于顺次接收包括三个或更多个数据包和一 EC 包的第一帧的包的子集的装置;以及
 - (b) 用于对所述第一帧的包的所述子集执行 EC 处理以重构所述第一帧的至少一个包的装置,其中,所述 EC 处理在接收到包的整个所述子集之前启动。

用于基于误差保护包的帧的接收器

技术领域

[0001] 本发明涉及具有纠错解码器的接收器中的缓冲存储器管理,并且具体来说,涉及前向纠错 (FEC) 解码器。

背景技术

[0002] FEC 技术通常用于基于包的网路 (如因特网),以补偿包损失。在美国专利 No. 6141788 中以及在 1999 年 12 月 Internet Society 中公布的 J. Rosenberg 等人的题名为 “An RTP Payload Format for Generic Forward Error Correction” 的因特网标准路径协议 (internet standards track protocol) no. RFC-2733 中描述了一种适用于利用采用实时传输协议 (RTP) 的实时媒体包的 FEC 技术,所述文献的全部内容通过引用包含于此。根据该技术,在传输之前,将称为校验和 (checksum) 或 FEC 包的附加包添加至媒体帧,或数据包的块。在接收器处,帧中的丢失数据包可以通过将校验和包与成功地接收到的那些数据包组合来恢复和重构。更具体地说,丢失的数据包通过对校验和包与成功地接收到的数据包执行数学运算 (例如,异或 (XOR) 逻辑运算) 来恢复。接着,可以根据该数学运算的结果来重构丢失的数据包。

[0003] 图 1 描绘了根据因特网标准路径协议 no. RFC-2733 的采用 FEC 解码的常规接收器 100。接收器 100 包括连接至 FEC 解码器 108 并且连接至播出 (play-out) 缓冲器 106 的包缓冲器 104。FEC 解码器 108 还连接至播出缓冲器 106。

[0004] 图 2 描绘了常规接收器 100 的操作。在框 202 操作开始。在框 204,将已接收包 102 存储在包缓冲器 104 中。在框 206,将包 102 的副本传送至播出缓冲器 106;但是,包 102 保留在包缓冲器 104 中,供 FEC 解码器 108 随后处理。在框 208,FEC 解码器 108 确定包 102 所属于的媒体帧 F 是否准备好 FEC 解码 (例如,当已经接收到针对指定媒体帧 F 的 FEC 包和除了一个数据包以外的所有数据包时)。如果没有作好准备,则操作返回至框 204,在框 204,接收并存储后续包。然而,如果帧 F 准备好 FEC 解码,则在框 210,FEC 解码器 108 从包缓冲器 104 中读取媒体帧 F 的已接收包 (即,除了一个以外的所有数据包和 FEC 包) 并对这些已接收包执行数学运算 (例如,XOR 逻辑运算)。在框 212,FEC 解码器 108 根据公知技术,基于数学运算的结果重构丢失包。在框 214,FEC 解码器 108 将恢复的包传送至播出缓冲器 106。

[0005] 然而,常规接收器 100 的缺点是包缓冲器 104 的大小相对较大。MPEG 媒体帧例如可以具有多达 24 个数据包加一个 FEC 包,总计达 25 个包。这样,包缓冲器 104 中的需要存储 MPEG 媒体帧的缓冲存储器的量 MemPerFrame 可以如下定义:

$$[0006] \quad MemPerFrame = \sum_{j=1}^{25} SizeOf(Packet(j))$$

[0007]

$$(MemPerFrame = \sum_{j=1}^{25} (\text{包}(j)\text{的大小}))$$

[0008] 而且,每一个包最大可以为用户数据报协议(UDP)的最大包大小(例如,1.5KB,常规以太网最大传输单元(MTU)的大小)。因而,具有25个大小为1.5KB的包的MPEG媒体帧将需要37.5KB的存储器。

[0009] 为了存储m个这种帧,所需总缓冲存储器为(mxMemPerFrame)。必须存储在包缓冲器104中的帧的数量m是多个变量的函数,这些变量包括(i)FEC解码器108的等待时间(latency)(即,接收帧的第一个包与对该帧开始FEC解码之间的时段),(ii)FEC解码延迟(即,从对该帧开始FEC解码至完成FEC解码的时段,例如,包括对该帧执行24组XOR逻辑运算),以及(iii)输入包的包抖动(packet jitter)。在常规接收器100中,要存储在包缓冲器104中的帧的数量m典型处于12与30之间。因而,如果将常规接收器100设计成接收MPEG媒体包,则包缓冲器104将通常需要处于450KB与1.125MB之间的存储器。

[0010] 为了例示,图3以图形方式描绘了包缓冲器104的示范性实现方式,其中,可以存储12个媒体帧302₁-302₁₂(每一个都具有25个包)。

发明内容

[0011] 本发明的示范性实施例提供了一种用于FEC解码的、显著缩减了FEC解码的等待时间和包缓冲器大小的接收器和方法。

[0012] 因而,在第一实施例中,本发明是一种用于在接收器中执行纠错(EC)处理的方法。顺次地接收包括三个或更多个数据包和一EC包的第一帧的包的子集。针对所述第一帧的包的所述子集执行EC处理,以重构所述第一帧的至少一个包。所述EC处理在接收到包的整个所述子集之前启动。

[0013] 在另一实施例中,本发明是一种接收器。该接收器包括包缓冲器,该包缓冲器适于顺次地接收包括三个或更多个数据包和一纠错(EC)包的第一帧的包的子集。该接收器还包括EC解码器,该EC解码器适于针对所述第一帧的包的所述子集执行EC处理,以重构所述第一帧的至少一个包,其中,所述EC处理在接收到包的整个所述子集之前启动。

附图说明

[0014] 根据以下详细描述、所附权利要求书以及附图,本发明的其它方面、特征以及优点将变得更完全清楚,在附图中,相同标号标识相似或相同部件。

[0015] 图1是采用FEC解码的常规接收器的简化框图;

[0016] 图2是例示图1的常规接收器的操作的流程图;

[0017] 图3是图1的常规接收器中的包缓冲器的图形例示;

[0018] 图4是根据本发明的采用FEC解码的接收器的实施例的简化框图;

[0019] 图5是例示图4中所示接收器的操作的流程图;

[0020] 图6是图4中所示接收器中的包缓冲器的图形例示;

[0021] 图7是图4中所示接收器的包括FEC数据结构的高速缓冲存储器的一部分的图形例示;

[0022] 图8是图5中所示帧确定框的更详细流程图;以及

[0023] 图 9 是图 4 中所示接收器的更详细框图。

具体实施方式

[0024] 在此引用的“一个实施例”或“一实施例”意指结合该实施例描述的特定特征、结构或特性可以包括在本发明的至少一个实施例中。本说明书中不同位置中出现的短语“在一个实施例中”不必全部指代同一实施例，也不是必须与其它实施例相互排除的单独或另选实施例。这同样适用于术语“实现方式”。

[0025] 图 4 描绘了根据本发明的接收器 400 的实施例。如图 4 所示，接收器 400 包括：包缓冲器 404、播出缓冲器 (play-out buffer) 406、FEC 解码器 408 以及 FEC 高速缓冲存储器 410。在接收器 400 中，FEC 解码器 408 基本上实时地（例如，紧挨在每一个包到达并存储在包缓冲器 404 中之后）处理每一个已接收包。即使该包所属于的帧不完整，FEC 处理也对于每一个包启动。为此，FEC 解码器 408 生成并保持被存储在 FEC 高速缓冲存储器 410 中的中间 FEC 处理结果。FEC 解码器 408 在每一个输入包被接收到并且进行 FEC 处理之后更新存储在 FEC 高速缓冲存储器 410 中的中间结果。因此，FEC 解码器 408 不同于图 1 的现有技术 FEC 解码器 108，所述现有技术 FEC 解码器 108 在开始 FEC 处理之前等待直到接收到媒体帧中的除了一个数据包以外的所有数据包为止。

[0026] 因为 FEC 解码器 408 基本上实时处理已接收包（不需要等待接收帧中的其它的包），包缓冲器 404 可以显著小于图 1 的现有技术包缓冲器 104。包缓冲器 404 仅存储已接收包，只需要 (i) 将已接收包传送至播出缓冲器 406 和 (ii) 将已接收包传送至 FEC 解码器 408 即可。在完成这些传送之后，包缓冲器 404 中的被已接收包消耗的存储器可以被清空并且用于另一输入包。这样，不再需要包缓冲器 404 在认为帧为 FEC 处理作好准备之前存储该帧的所有包。因而，播出缓冲器 406 可以显著小于（例如，以范围从大约 2 到大约 10 的因子）播出缓冲器 106。

[0027] 例如，在一个实施例中，可以将播出缓冲器 406 的大小可以确定成允许存储大约 12 个包到大约 24 个包（这些包可以属于 12 个不同的连续发送帧），并且使 FEC 解码器 408 适于针对 12 个连续发送帧的包的 FEC 处理。然而，应当明白，包缓冲器 404 被设计成处理的包的数量和 FEC 解码器 408 被设计成处理的包的数量可以基于在基于包的网路中的包抖动、FEC 解码器 408 的等待时间以及 FEC 解码器 408 的处理延迟来改变。

[0028] 为了简化，下面假定，在一个实施例中，接收器 400 被设计在具有这样的包抖动的通信系统中使用，即，该包抖动足够小以准许 FEC 解码器 408 一次针对属于三个连续发送媒体帧的包进行操作。在这样的实施例中，FEC 高速缓冲存储器 410 尤其包括三个存储器位置 fec_str_1 、 fec_str_2 以及 fec_str_3 （图 4 中未示出），用于存储针对三个连续发送媒体帧（下面称为前一帧 F_1 、当前帧 F_2 以及下一帧 F_3 ）的中间数学结果（例如，XOR 逻辑运算结果），并且包括用于存储当前已接收包的存储器位置 $current_packet$ 。下面还假定，根据本领域普通技术人员已知的技术，在接收器 400 与发送器（未示出）之间预定或者商定被 FEC 帧保护的包的数量。

[0029] 图 5 描绘了接收器 400 的操作。在框 502 操作开始。在框 504，清空存储器位置 fec_str_1 、 fec_str_2 以及 fec_str_3 （例如，设置成零）。在框 506，将已接收包 402 存储在包缓冲器 404 中。在框 508，将已接收包 402 从包缓冲器 404 传送至 FEC 高速缓冲存储器 410

中的存储器位置 `current_packet` 并且将其传送至播出缓冲器 406, 以及将包缓冲器 404 中的、被已接收包 402 所占用的存储器位置释放, 以供其它包使用。在框 510, FEC 解码器 408 读取包 402 的首部信息 (head information), 并且确定包 402 所属于的媒体帧 (例如, 前一帧 F_1)。

[0030] 在框 512, FEC 解码器 408 确定包 402 是否为针对包 402 所属于的媒体帧 (例如, 前一帧 F_1) 接收器 400 已经接收到的第一个包。如果是, 则在框 514, FEC 解码器将包 402 存储在与包 402 所属于的媒体帧相对应的存储器位置 (`fec_str1`、`fec_str2` 或 `fec_str3`) 中, 并且操作返回至框 506, 以处理另一已接收包。如果包 402 不是针对包 402 所属于的媒体帧 (例如, 前一帧 F_1) 接收器 400 已经接收到的第一个包, 则在框 516, FEC 解码器 408 在包 402 的数据部分与和该包所属于的媒体帧相对应的存储器位置 (例如, `fec_str1`) 的内容之间执行数学运算 (例如, XOR 逻辑运算)。接着, 将数学运算的结果存储在另一存储器位置 (例如, `fec_str1`) 中, 由此, 替换该存储器位置中的先前内容。

[0031] 在框 518, FEC 解码器确定包 402 所属于的媒体帧是否为 FEC 解码作好准备 (例如, 当对于该媒体帧已经接收到 FEC 包和除了一个以外的所有其它数据包时)。(下面参照图 8 对帧确定框 518 进行更详细说明。) 如果包 402 所属于的帧没有为 FEC 解码作好准备, 则操作返回至框 506, 以处理另一已接收包。

[0032] 如果该媒体帧为 FEC 解码作好准备, 则在框 520, FEC 解码器 408 使用存储在与包 402 所属于的媒体帧 (例如, 前一帧 F_1) 相对应的存储器位置 (例如, `fec_str1`) 中的数学运算结果, 以根据诸如美国专利 No. 6141788 和因特网标准路径协议 no. RFC-2733 中所描述技术的公知技术重构丢失的包。最后, 在框 522 中, FEC 解码器 408 将恢复的包存储在播出缓冲器 406 中, 并且将存储器位置 (例如, `fec_str1`) 清空, 以供下一个输入帧使用。

[0033] 图 6 是 FEC 高速缓冲存储器 410 的图形例示。在一个实施例中, FEC 高速缓冲存储器 410 包括三个数据结构 (或定义的存储器部分) 602_1 、 602_2 以及 602_3 (分别被标识为 `FEC_RX_DATA1`、`FEC_RX_DATA2`, 以及 `FEC_RX_DATA3`), 和用于存储当前已接收包的 `current_packet` 存储器位置 604。在数据结构 (或定义的存储器部分) 602_1 、 602_2 以及 602_3 中, FEC 解码器 408 分别存储分别对 FEC 解码媒体帧 F_1 、 F_2 以及 F_3 有用的信息。

[0034] 在一个实施例中, 数据结构 602_1 、 602_2 , 以及 602_3 分别定义如下:

[0035] `typedef struct FEC_RX_DATA`

[0036] `{`

[0037] `uint16_t snmin;`

[0038] `uint16_t snmax;`

[0039] `uint8_t fec_str[MAX_RTP_PACKET_SIZE];`

[0040] `uint16_t fstrlen;`

[0041] `uint8_t fec_rec[MAX_RTP_PACKET_SIZE];`

[0042] `uint16_t frecflen;`

[0043] `struct FEC_RX_DATA*prev;`

[0044] `struct FEC_RX_DATA*next;`

[0045] `}FEC_RX_DATA_t;`

[0046] 其中, 数据字段定义如下:

[0047]

Snmin	当前帧 F_2 内 RTP 媒体包的最小序列号
Snmax	当前帧 F_2 内 RTP 媒体包的最大序列号
fec_str	针对属于当前帧 F_2 的已接收 RTP 媒体包的 XOR 逻辑运算的当前结果
Fstrlen	属于当前帧 F_2 的 RTP 媒体包的有效载荷长度 在一个实施例中, 可以采用 XOR 格式存储已接收包的有效载荷长度 (例如, 如果包 1 具有 9 个字节的有效载荷长度, 而包 2 具有 8 个字节的有效载荷长度, 则 $fstrlen = 8 \text{ XOR } 9 = 1$)。
fec_rec	属于当前帧 F_2 的已接收 FEC 包
Freclen	FEC 包长度
Prev	针对前一帧 F_1 的数据结构的指针
Next	针对下一帧 F_3 的数据结构的指针

[0048] 为了例示, 图 7 以图形方式描绘了数据结构 602₁。

[0049] 图 8 是更详细例示图 5 的帧确定框 510 的流程图, 其中, FEC 解码器 408 确定已接收包 402 属于哪个媒体帧 (例如, 前一帧 F_1)。

[0050] 在框 802, FEC 解码器 408 读取针对存储在图 4 所示包缓冲器 404 中的已接收包 402 的有效载荷类型描述符 PT 和序列号 SNRcvd。在框 804, FEC 解码器 408 检查已接收包 402 是否为 FEC 包 (即, 有效载荷类型描述符 PT 等于预定值 FEC_TYPE)。如果已接收包 402 不是 FEC 包, 则在框 806 继续操作。在框 806, FEC 解码器 408 检查序列号 SNRcvd 是否大于或等于当前帧 F_2 内的媒体包的最小序列号 snmin。如果不是, 则在框 808, FEC 解码器 408 将已接收包 402 标识为属于前一帧 F_1 , 并且将指针设置到数据结构 602₁ (FEC_RX_DATA1), 用于存储与前一帧 F_1 有关的 FEC 解码信息的定义的存储器位置。在框 11 继续操作, 其中, 操作返回至图 5 中的框 516。在框 516, FEC 解码器 408 使用该指针以标识具有要在针对包 402 的数据部分执行数学运算中使用的存储器位置 (例如, fec_str₁) 的数据结构 (例如, FEC_RX_DATA1)。

[0051] 然而, 如果在框 806, FEC 解码器 408 确定序列号 SNRcvd 大于或等于当前帧 F_2 内的媒体包的最小序列号 snmin, 则在框 810 继续进行操作。FEC 解码器 408 检查已接收包的序列号 SNRcvd 与当前帧 F_2 内 RTP 媒体包的最小序列号 snmin 的差是否大于或等于可以受 FEC 包保护的包的预定最大数量 MAX_FEC_PKT_PER_FEC。例如, 对于 MPEG RTP 媒体帧来说, 可以受 FEC 包保护的包的预定最大数量 MAX_FEC_PKT_PER_FEC 为 24。

[0052] 如果已接收包的序列号 SNRcvd 与当前帧 F_2 内 RTP 媒体包的最小序列号 snmin 的差不大于或等于 (即, 小于) 预定最大数量 MAX_FEC_PKT_PER_FEC, 则在框 814, FEC 解码器 408 将已接收包 402 标识为属于当前帧 F_2 。具体来说, 将指针设置至数据结构 602₂ (FEC_RX_

DATA2), 用于存储与当前帧 F_2 有关的 FEC 解码信息的定义的存储器位置。在框 1i 继续操作, 其中, 操作返回至图 5 中的框 516。在框 516, FEC 解码器 408 使用指针来标识具有要在对数据包 402 的数据部分执行数学运算中使用的存储器位置 (例如, fec_str_2) 的数据结构 (例如, FEC_RX_DATA2)。

[0053] 返回至框 810, 如果 FEC 解码器 408 确定已接收包的序列号 SNRcvd 与当前帧 F_2 内 RTP 媒体包的最小序列号 snmin 的差大于或等于 (即, 不小于) 预定最大数量 MAX_FEC_PKT_PER_FEC, 则在框 812 继续操作。FEC 解码器 408 将已接收包 402 标识为属于下一帧 F_3 。具体来说, 将指针设置至数据结构 602₃ (FEC_RX_DATA3), 用于存储与下一帧 F_3 有关的 FEC 解码信息的定义的存储器位置。在框 1i 继续操作, 其中, 操作返回至图 5 中的框 516。在框 516, FEC 解码器 408 使用指针来标识具有要在对包 402 的数据部分执行数学运算中使用的存储器位置 (例如, fec_str_3) 的数据结构 (例如, FEC_RX_DATA3)。

[0054] 如果在框 804 中, FEC 解码器 408 确定已接收包 402 是 FEC 包 (即, 有效载荷类型描述符 PT 等于预定值 FEC_TYPE), 则在框 816 中继续操作。在框 816, 将包 402 的掩码字段和 snbase 字段用于确定包 402 属于哪一个帧 (例如, 前一帧 F_1 、当前帧 F_2 或下一帧 F_3)。在框 818, 接着, 在 FEC 高速缓冲存储器 410 中将包 402 以恰当的数据结构 (FEC_RX_DATA1、FEC_RX_DATA2, 或 FEC_RX_DATA3) 存储在 fec_rec 存储器位置中, 并且在框 820, 操作返回至图 5 中的框 518。

[0055] 图 9 是图 4 中所示接收器 400 的更详细框图。基于可从位于英国剑桥的 ARM Holdings plc 获得的内核设计, 将 FEC 解码器 408 优选地实施为精简指令集计算 (RISC) 处理器 (如 ARM1176J-S 处理器)。RISC 处理器优选地以至少 250MHz 的速度运算。

[0056] 将包缓冲器 404 和 FEC 高速缓冲存储器 410 优选地实施在 RISC 处理器内的 64KB 数据紧密耦合存储器 (D-TCM) 910 内。RISC 处理器还可以包括 8KB 数据高速缓冲存储器 (D-cache)、8KB 指令高速缓冲存储器 (I-cache) 以及 64KB 指令紧密耦合存储器 (I-TCM)。

[0057] 如图 9 所示, 播出缓冲器 406 和 FEC 解码器 408 经由 64 位基于高级可扩展接口 (AXI) 的总线矩阵 914 连接。播出缓冲器 406 优选为外部双倍数据速率 (DDR) 同步动态随机存取存储器 (SDRAM), 其 (i) 具有每秒传送 533 兆数据的数据传送速率, (ii) 遵循题为 “DDR2SDRAM 规范” 的 JEDEC 标准 No. JESD79-2E, 以及 (iii) 以 “x16” 构造 (即, DDR2-533x16 存储器) 实施。播出缓冲器 406 通过 DDR2 外部存储器接口 (EMI) 938 连接至总线矩阵 914。

[0058] 接收器 400 还包括供 FEC 解码器 408 使用的 256KB 通用 SRAM 存储器 (“PPBMEM”) 912。

[0059] 接收器 400 还包括用于提供以太网支持的介质接入控制器 (MAC) 924、930。MAC 924、930 通过包分类引擎 (PCE) 协处理器 920、928 以及传送 MAC DMA (TXD) 协处理器连接至总线矩阵 914。PCE 协处理器 920、928 在 MAC 924、930 与接收包的包缓冲器 404 之间提供 L2/L3/L4IP 和 UDP 包分类和直接存储器存取 (DMA) 支持。TXD 协处理器 922、926 在 MAC 924、930 与用于传送包的传送存储器 (未示出) 之间提供 DMA 支持。48KB 通用 SRAM 存储器 902 还被设置用于 PCE 协处理器 920、928, 例如, 用于存储用于因特网协议版本 6 (IPv6) 支持的表。

[0060] 接收器 400 还包括向播出缓冲器 406 提供 DMA 支持 (通过外部存储器接口 938) 的 DMA 控制器 932。

[0061] FEC 解码器 408 还优选地 (i) 通过经由 AXI 至 APB 桥 916 连接的高级外设总线 (APB) 918 与 MAC 924、930 和 PCE 协处理器 920、928 通信, 以及 (ii) 通过经由 AXI 至 AHB 桥 934 连接的高级高性能总线 (AHB) 936 与 DMA 控制器 932 和 DDR2 外部存储器接口 938 通信。

[0062] 还可以将一个或多个数字信号处理器 (DSP) (未示出) 连接至总线矩阵 914, 以解码存储在播出缓冲器 406 中的媒体包。

[0063] 由此, 已经描述了一种用于接收具有多个数据包和纠错包的媒体帧的新颖且创新的系统和方法。可以注意到, 在上述图 4 所示接收器 400 中, FEC 解码器 408 在已经接收到对应媒体帧中所有或大部分剩余包之前直接针对每一个已接收包执行数学运算 (例如, XOR 逻辑运算), 而没有延迟。与此相反, 在图 1 所示常规接收器 100 中, FEC 解码器 108 延迟执行任何数学运算直到确定被缓冲帧为 FEC 解码作好准备为止 (例如, 直到接收到针对该帧的 FEC 包和除了一个以外的其余所有数据包为止)。这样, FEC 解码器 408 具有比常规 FEC 解码器 108 显著小的等待时间。而且, 因为 FEC 解码器 408 直接针对每一个已接收包执行数学运算, 所以不存在对于要能够存储每一个帧的几乎所有包的包缓冲器 404 的需要。结果, 包缓冲器 404 在大小上可以比图 1 的包缓冲器 104 更小。

[0064] 本发明可以被实施为基于全数字、全模拟或混合模拟和数字两者的电路的处理, 包括作为单一集成电路 (如 ASIC 或 FPGA)、多芯片模块、单一卡或多卡电路封装的可能实现方式。如本领域技术人员应当清楚的, 电路部件的各种功能还可以被实施为采用软件程序的处理模块。这种软件例如可以在数字信号处理器、微控制器或通用计算机中采用。

[0065] 还应明白, 在不脱离如下权利要求书中所表达的本发明的范围的情况下, 本领域技术人员可以在为了说明本发明的性质而描述并例示的部件的细节、材料以及配置方面进行各种改变。

[0066] 因而, 尽管上面已经关于利用 XOR 逻辑运算的纠错对本发明进行了描述, 但本发明并不因而受限, 而是可以使用其它逻辑运算 (例如, 异或非 (NXOR) 逻辑运算)。而且, 本发明可以利用基于除了逻辑运算以外的其它数学运算的其它纠错 (EC) 算法来实践, 只要这种 EC 算法允许 FEC 处理在没有延迟的情况下针对每一个已接收包发生即可, 例如, 不需要一帧中的全部或大致全部包在启动 EC 处理之前存在。

[0067] 而且, 尽管上面关于特定长度和量对本发明进行了描述, 但本发明并不因而受限, 而是可以使用其它长度和量。例如, FEC 解码器 408 和包缓冲器 404 被设计成处理的帧的数量以及 FEC 高速缓冲存储器 410 中的存储器位置的数量可以基于在基于包的网络中的包抖动和 / 或基于 FEC 解码器 408 的等待时间来增加或减小。例如, 在本发明的一个实施例中, 可以采用双帧包缓冲器, 而非如上所述三帧包缓冲器。在这种实施例中, 在 FEC 高速缓冲存储器 410 中仅需要两个数据结构 602₁、602₂, 而非如上所述三个。

[0068] 在另一实施例中, 如果网络包抖动非常小, 则可以采用一帧包缓冲器, 并且 FEC 高速缓冲存储器 410 可以包括单一数据结构 602₁, 或者由 602₁ 组成。另一方面, 如果包抖动较大, 则 FEC 解码器 408 和包缓冲器 404 可以被设计成处理四个 (或更多个) 帧, 并且在 FEC 高速缓冲存储器 410 中将需要对应数量的数据结构。例如, 预期的是, 如果包抖动和上述接收器 100 的情况 (其中, 包缓冲器 104 被设计成能够存储 12 个帧) 一样大, 则 FEC 解码器 408 和包缓冲器 404 同样可以被设计成处理 12 个帧, 并且在 FEC 高速缓冲存储器 410 中具有 12 个数据结构。

[0069] 还应明白,在此阐述的示范性方法的步骤不必要求按所述次序执行,而应当将这种方法步骤的次序理解成仅是示范性的。同样,在这种方法中可以包括附加步骤,并且可以在与本发明的各种实施例相一致的方法中省略或组合特定步骤。

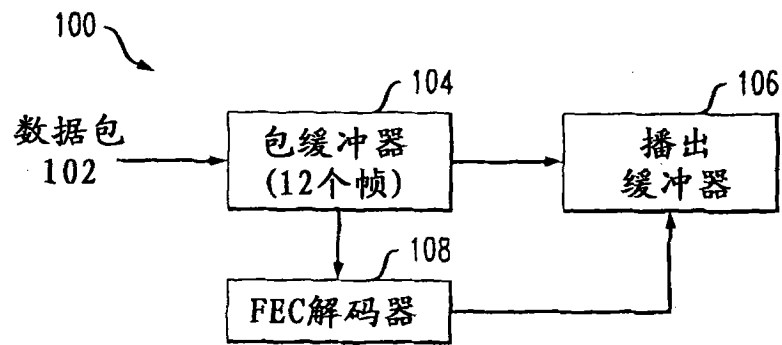


图 1

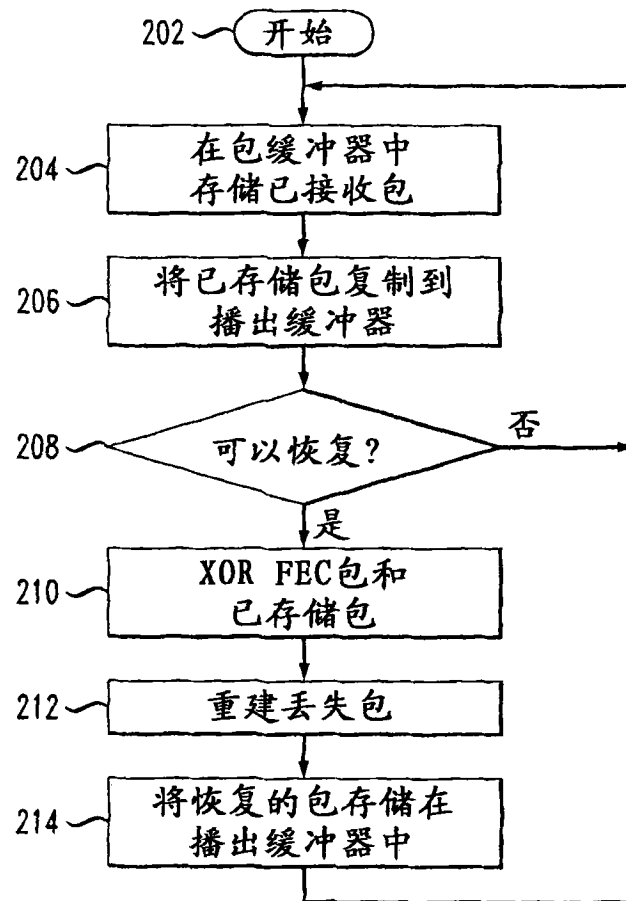


图 2

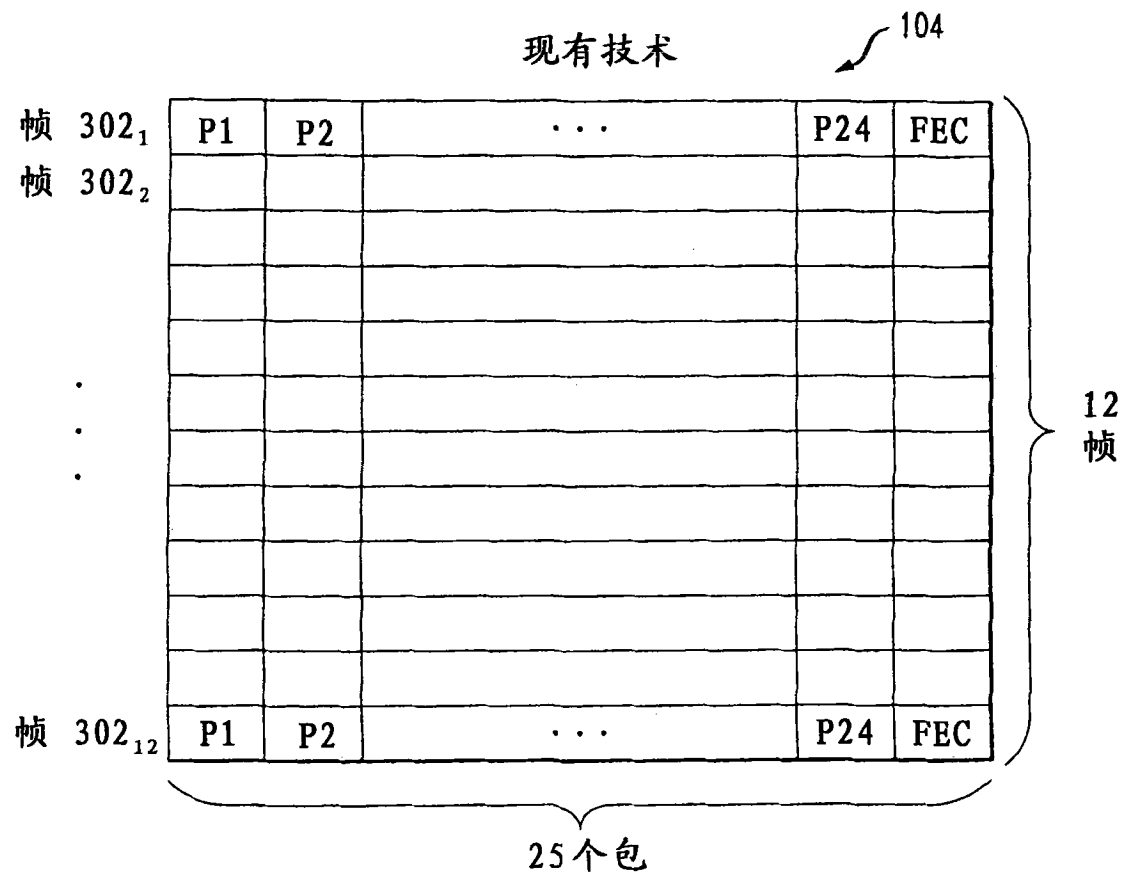


图 3

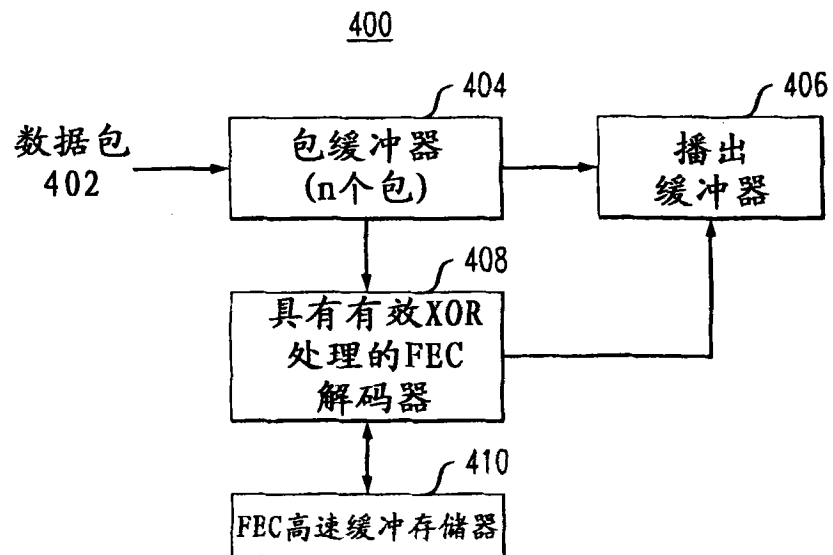


图 4

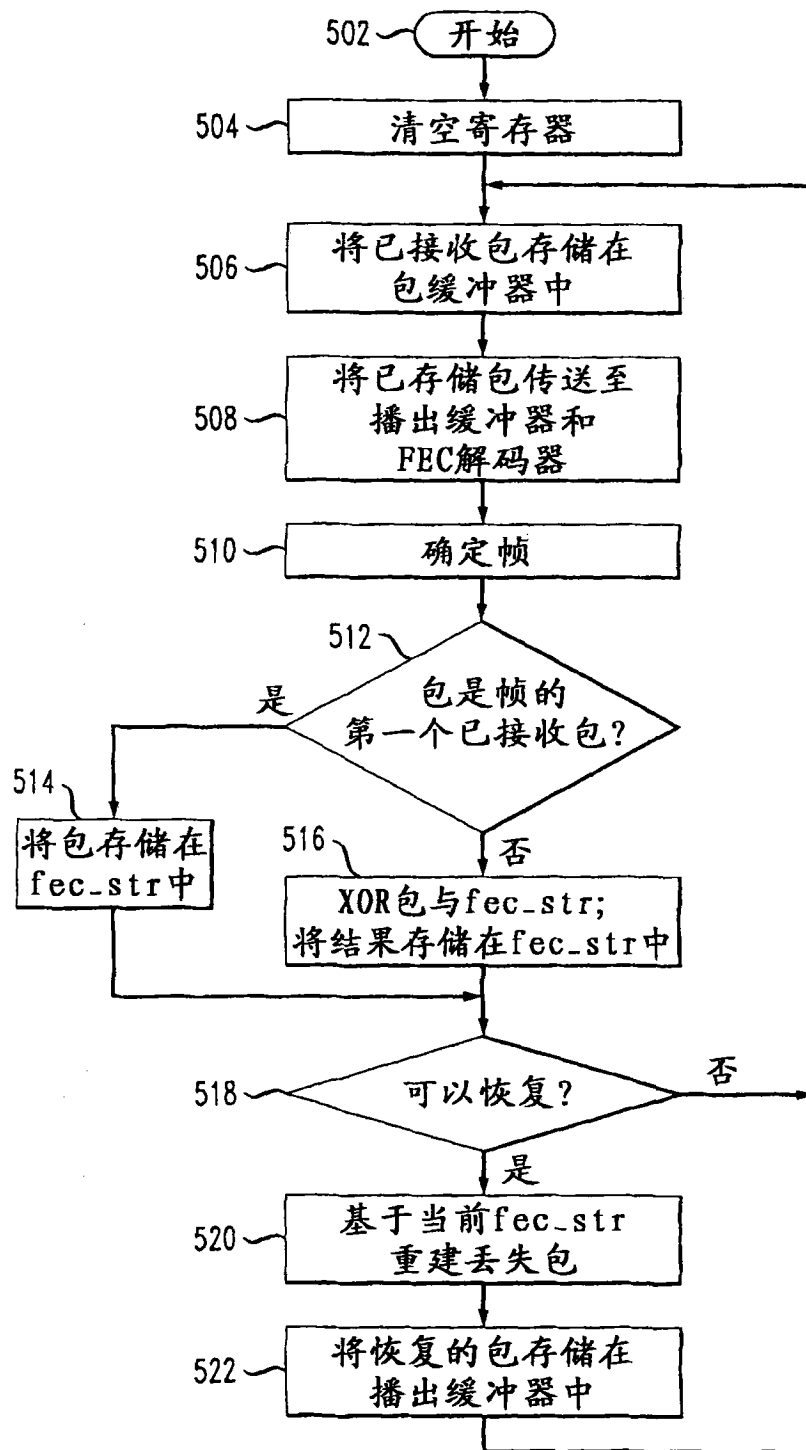


图 5

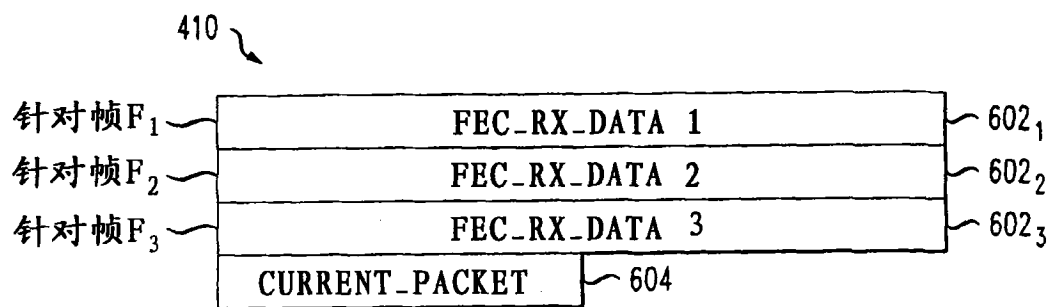


图 6

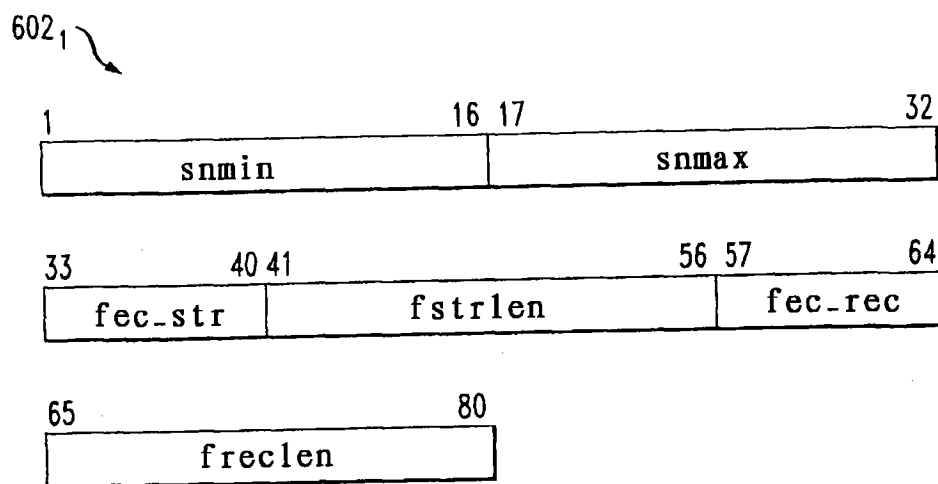


图 7

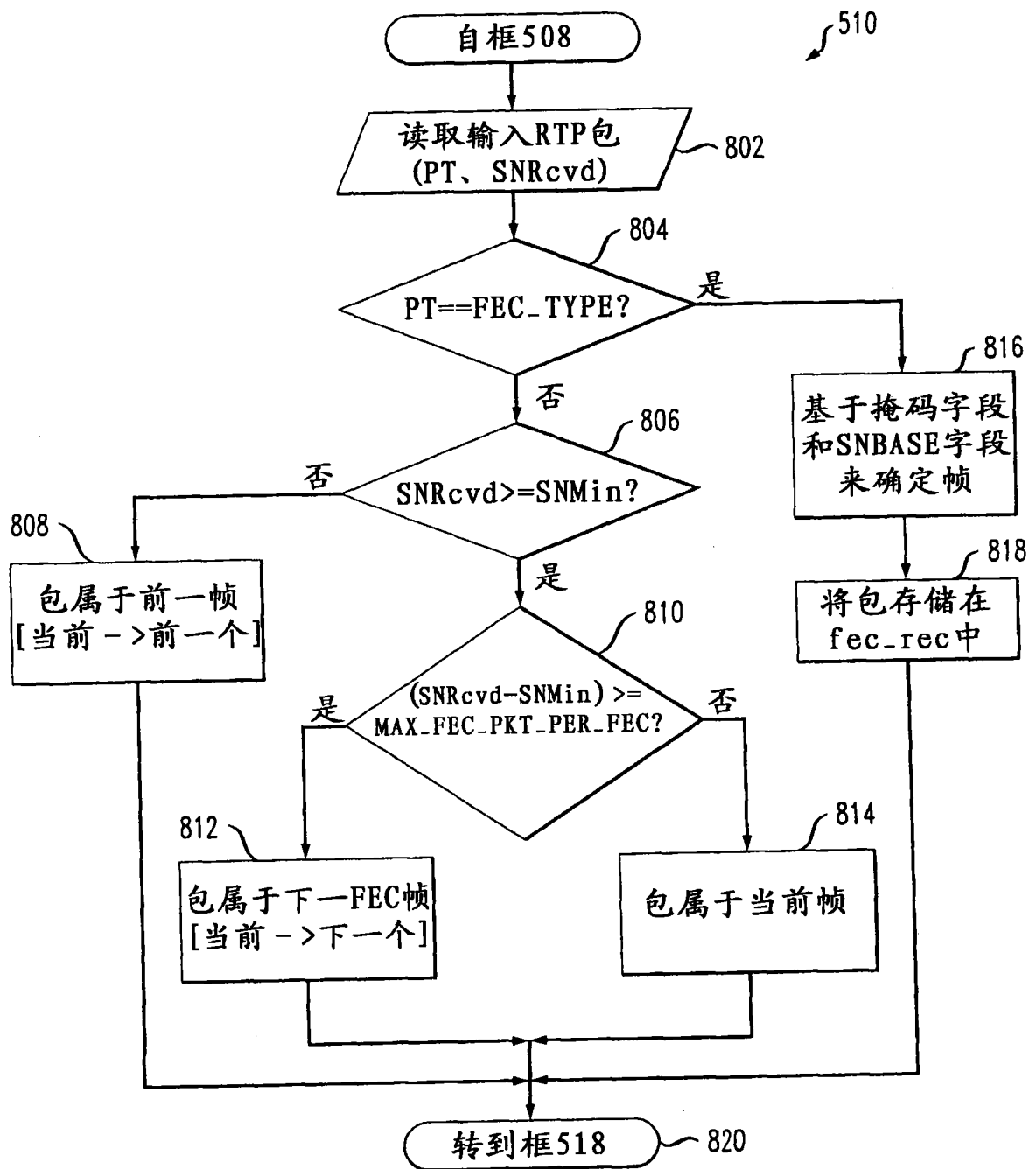


图 8

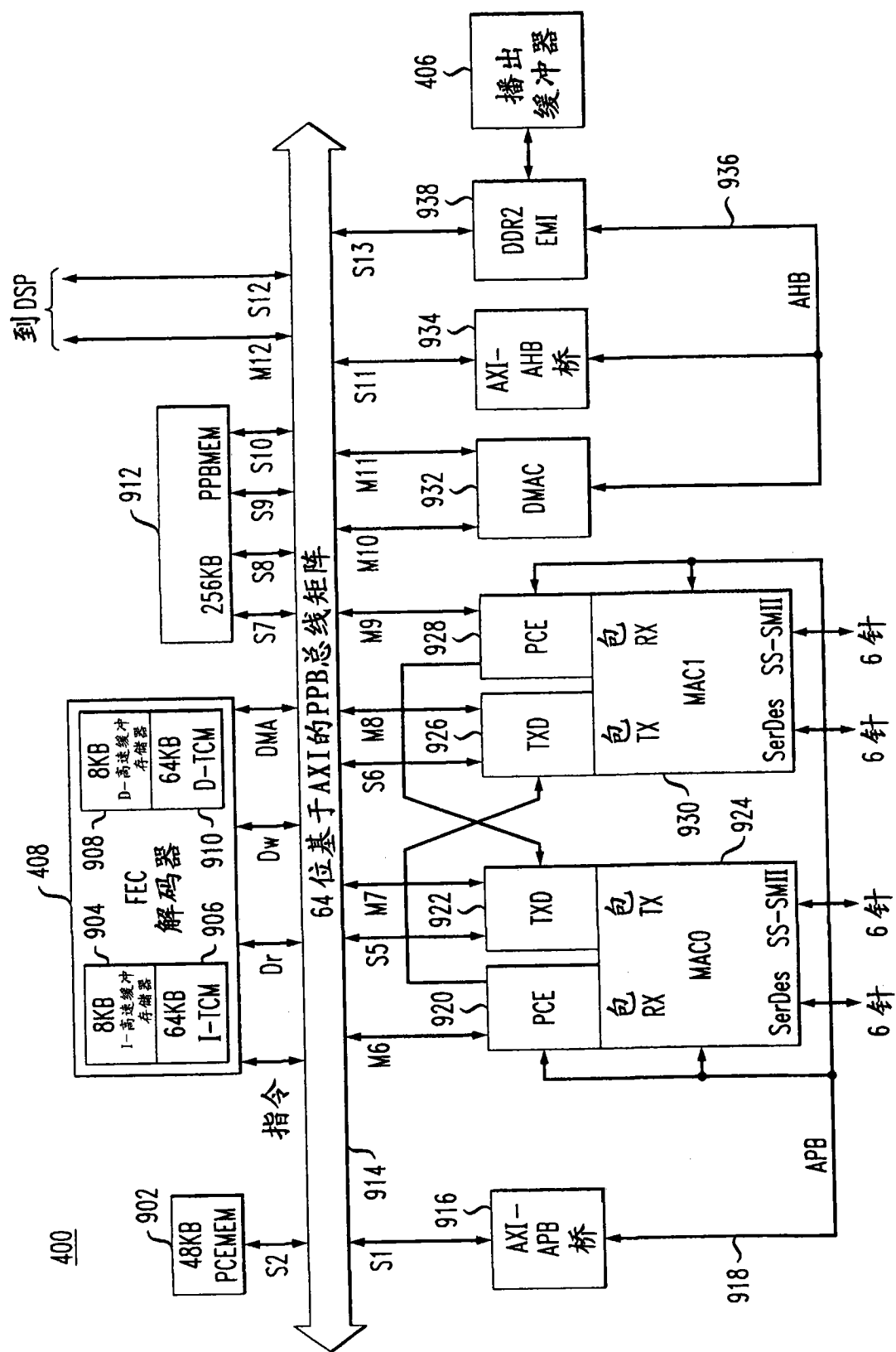


图 9