

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G06F 9/46

G06F 12/08



[12] 发明专利申请公开说明书

[21] 申请号 02813473.7

[43] 公开日 2004 年 8 月 18 日

[11] 公开号 CN 1522406A

[22] 申请日 2002.6.20 [21] 申请号 02813473.7

[30] 优先权

[32] 2001.7.6 [33] EP [31] 01202610.0

[86] 国际申请 PCT/IB2002/002546 2002.6.20

[87] 国际公布 WO2003/005197 英 2003.1.16

[85] 进入国家阶段日期 2004.1.2

[71] 申请人 皇家飞利浦电子有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 A·V·维梅尔斯费尔德

[74] 专利代理机构 中国专利代理(香港)有限公司

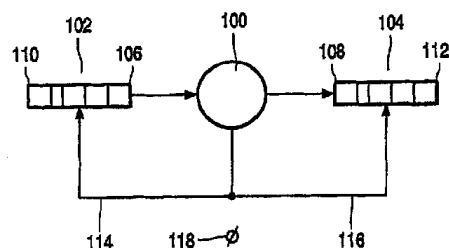
代理人 程天正 王 勇

权利要求书 2 页 说明书 8 页 附图 5 页

[54] 发明名称 处理媒体文件的方法和系统

[57] 摘要

流式应用程序经常能被划分成能被多处理器系统处理的多个任务。这些任务通过包含要被任务(100)处理的数据的队列(102、104)互相联系。为了使不同处理器对数据的访问能同步,提供一种获得信号标(118)的方法,该信号标保护任务本身以及改变包含在队列内的数据包的状态。



I S S N 1 0 0 8 - 4 2 7 4

1. 一种处理媒体文件的方法，该方法包含至少一个被多个处理器执行的处理步骤，该至少一个处理步骤包含处理数据，特征在于，该至少一个处理步骤还包含：

5 第一步骤，由多个处理器的其中之一规定一个保护该至少一个处理步骤的信号标；

第二步骤，从第一队列获得第一处理包，其中第一处理包包含所述处理数据；

10 第三步骤，进行等待，一直等到第一处理包具有多个预定状态中的第一预定状态；

第四步骤，将第一处理包的第一预定输入状态改变成多个预定状态中的第二预定状态；

第五步骤，由多个处理器的其中之一释放该信号标；

第六步骤，处理该媒体文件，包含对处理数据的读和/或写；

15 第七步骤，将第一处理包的第二预定状态改变成多个预定状态中的第三预定状态。

2. 按照权利要求1的处理媒体文件的方法，其中，

第二步骤进一步包含从第二队列获得第二处理包；

20 第三步骤进一步包括进行等待，直到第二处理包具有多个预定状态的第三预定状态；

第四步骤进一步包含将第二处理包的第三预定状态改变成多个预定状态中的第四预定状态；和

第七步骤进一步包含将第二处理包的第四预定状态改变成多个预定状态中的第一预定状态。

25 3. 按照权利要求1的处理媒体文件的方法，其中，第三步骤进一步包含一个更新共享存储器的内容的子步骤，该共享存储器是由多个处理器共享的。

4. 一种用于处理媒体文件的系统，该系统（500）包含：

30 多个处理器装置（502，504），用来执行对处理数据操作的至少一个处理步骤，该处理步骤包含读和/或写处理数据；

第一队列装置（506），用来包含第一处理包，其中第一处理包包含处理数据；

其特征在于该系统进一步包含：

信号标装置（510），用于保护由多个处理器其中之一执行的处理步骤；

5 状态装置（508，514），用于包含多个预定状态中的第一预定状态；

等待装置（516），用于等待输入包的第一预定状态；

变更装置（518），用于将第一预定状态改变成多个预定状态中的第二预定状态。

10 5. 按照权利要求4的处理媒体文件的系统，其中，该系统进一步包含：

第二队列装置（508），用来包含一个第二包，其中该第二包包含第二处理数据；

状态装置（508，514）进一步用来包含多个状态中的第三预定状态；和

15 变更装置（518）进一步用来将第三预定状态改变成多个状态中的第四预定状态。

6. 按照权利要求4的处理媒体文件的系统，其中，该系统（500）进一步包含用来包含多个处理器共享内容的存储装置（526）。

20 7. 用来执行按照权利要求1至3的任何一个的方法的计算机程序产品。

8. 包含按照权利要求7的计算机程序产品的存储装置（524）。

9. 包含按照权利要求4至6的任何一个的系统的电视机（600）。

10. 包含按照权利要求4至6的任何一个的系统的机顶盒（700）。

处理媒体文件的方法和系统

发明领域

- 5 本发明涉及处理媒体文件的方法和系统，该方法包含至少一个由多个处理器执行的处理步骤，并且该至少一个处理步骤包含处理数据。

此外，本发明还涉及处理媒体文件的系统，该系统包含：

- 多个处理器装置，用于执行至少一个对处理数据所执行的处
10 理步骤，该处理步骤包含读和/或写该处理的数据；

从用来包含一个第一处理包的第一队列装置中读，其中该第一处理包包含处理数据。

背景技术

- 从 W0 99/22296 中可知上述方法的一个实施例。这里，在一个第
15 一站和一个或多个第二站之间执行基于信号标 (semaphore-based) 的同步。为每个站提供单独一个二价 (bivalent) 信号。第一站检查所有的第二站信号标具有与其自己的信号标的第一状态相对的第二状态。它然后执行一个第一访问操作并翻转第一状态。否则它就放弃第一访问操作。第二站检查第一站信号标具有与其自己的信号标的第二
20 状态相对的第一状态。它然后执行一个第二访问操作并翻转第二状态。否则它就放弃第二访问操作。然而，每个站在决定执行之前都必须检查另一个站的信号标。因此，每个站必须知道处理步骤的网络内的下一站或下一步骤，这就产生在各处理步骤之间的依赖关系，导致不灵活的体系结构。

25 发明内容

本发明目的是提供一种以改进的方式获得包的方法并允许体系结构更加灵活。为了实现这个目的，按照本文前序部分的处理媒体文件的方法的特征在于该至少一个处理步骤还包含：

- 30 第一步骤，由多个处理器的其中一个处理器规定一个信号，它保护 (guard) 该至少一个处理步骤；

第二步骤，从第一队列获取第一处理包，其中第一处理包包含处理数据；

第三步骤, 进行等待, 到第一处理包具有多个预定状态中的第一预定状态;

第四步骤, 将第一处理包的第一预定输入状态改变成多个预定状态中的第二预定状态;

5 第五步骤, 由多个处理器的其中一个处理器释放该信号标;

第六步骤, 处理该媒体文件, 包含对处理数据进行读和/或写;

第七步骤, 将第一处理包的第二预定状态改变成多个预定状态中的第三预定状态。

10 通过在从一个队列获得第一包之前规定一个保护处理步骤的信号标, 就能在一个原子操作 (atomic action) 内规定该第一包和指示下一个要获得的包的指针的对应更新。这就防止其它处理器同时获得同一个包, 而如果其它处理器同时获得同一个包, 会导致队列中的包不能保持要被处理的顺序。此外, 通过等待第一包的一个预定状态, 对第一包的处理是利用第一包的正确内容来执行的。由于通过队列解除
15 每个处理步骤的联系, 各处理步骤彼此互不知道, 这导致更加灵活的体系结构, 在该体系结构中, 队列和第一包能起到处理步骤的输入或输出队列的作用。

权利要求 2 中描述了一个按照本发明方法的实施例。通过在所要求信号标的保护内获取除了第一包之外还获取第二包, 从处理第一包
20 的内容得到的结果被存储在第二包内的顺序被加以保留。此外, 在允许重写第二包之前, 第二包的可能的以前内容中不会写入新的内容。由于处理数据是在信号标的保护之外执行的, 直到第一处理器结束处理所有的数据为止其它处理器不被阻塞。在流式 (streaming) 体系结构内, 可以将第一队列和第一包视为该处理步骤的输入队列和输入
25 包, 而将第二队列和第二包视为该处理步骤的输出队列和输出包。

权利要求 3 中描述了一个按照本发明的方法的实施例。通过在信号标的保护内更新共享存储器, 共享存储器的被更新的内容变得可
同时用于所有相关的处理器, 并且与所获得的输入和输出包的内容一致。由于更新是在信号标的保护内进行的, 由处理步骤进行的对共享
30 存储器的读和写是在一个原子操作内进行的。

此外, 本发明的目的是提供一种以改善的方式获得包并允许体系结构更加灵活的系统。为了实现这个目的, 按照本文篇首部分的处理

媒体文件的系统的特征在于该系统进一步包含：

信号标装置，用于保护多个处理器中一个处理器的处理步骤；

状态装置，用于包含多个预定状态中的第一预定状态；

等待装置，用于等待输入包的第一预定状态；

- 5 变更装置，用于将第一预定状态改变成多个预定状态中的第二预定状态；

权利要求 5 中描述了一个按照本发明的用于解译媒体文件的系统。

附图说明

- 10 通过由下述附图表示的实施例说明本发明：

图 1 表示一个从输入队列读取数据并将处理过的数据写到输出队列的任务（task）；

图 2 表示每个包的状态；

图 3 表示任务的信号标的状态；

- 15 图 4 表示按照本发明的方法的主要步骤；

图 5 示意性地表示按照本发明的系统的主要部分；

图 6 示意性地表示包含按照本发明的实施例的电视机；

图 7 示意性地表示包含按照本发明的实施例的机顶盒的最重要的部分；

- 20 图 8 示意性地用图表示 MPEG 视频解码器的处理任务。

具体实施方式

- 现今的连续媒体数据处理越来越多地由可编程部件、专用硬件部件与可编程部件的组合进行，而不是由专用的单一功能的部件进行。可编程部件引入这样的需要，即能够实现这种从硬件向软件驱动的处理的转变的模块化、灵活、可维护和功能强大的体系结构。例如在数据由多处理器系统处理的实时 MPEG 视频处理领域内，情况就是这样。本发明涉及这种流式体系结构，使得能够执行对诸如 MPEG 编码的数据、宏块（macroblock）数据和 422 视频数据之类进行视频解译和译码（transcoding）的任务或者多路转换和去多路转换（de-multiplexing）。此外，本发明可以被应用于视频处理、VBI、IP 和传输流（transport stream）处理、视频基本流或被打包的基本流（PES）处理。总之，本发明涉及其中的不同处理步骤能被多个处理器
- 25 30

执行的流式体系结构。在本说明书的其余部分，这些处理器被称作协处理器，以表明多个处理器能执行一个任务。

在本发明中，将每个能被协处理器板 (board) 执行的应用程序划分为各个任务，这些任务与相同域 (domain) 内的其它应用程序的其它任务有尽可能多的共同点。这些任务最好以允许数据流入和流出的流式方式操作。要进行若干顺序处理步骤的数据需要尽量像在流水线内那样按互相跟随的任务的方式来处理。能独立地被平行处理的数据最好按并行的任务处理或者按一个能在例如多个处理器上的多个数据块并行操作的任务处理。

图 1 表示的一个从输入队列 102 读取数据 106 并将处理过的数据 108 写到输出队列 104 的任务 100。输入队列 102 被来自在先任务的数据 110 填充，后继任务从输出队列 104 读取数据 112。该任务通过递增其读指针 114 和写指针 116 以跟踪其应当处理的数据。也可以使用其它能顺序从输入队列读和顺序向输出队列写的技术。本发明不通过比较读指针 114 和写指针 116 来在读输入队列与写输出队列写之间进行同步，而是以下述方式执行该方法。任务例如可以是：符合 MPEG 规范的前向或逆向分立余弦转换 (DCT-Discrete Cosine Transform)、符合 MPEG 规范的量化或解量化 (de-quantization)、将帧块存入帧存储器等等。

多个处理器执行该任务，任务有零个或多个输入队列以及零个或多个输出队列，每个队列含有一个或多个包括数据的包。一般来说，包要被发往特定的队列，包的大小和内容可以不一样。预定任务在特定队列内的包的大小要使该预定任务的所有类型的数据能被精确地表示。此外，数据是按值传送的。仅当有特别要求时，数据才按基准被传送。另外，数据可以是特定类型的，并且任务可以改变数据类型并生成另一个类型的数据。在大多数目前的系统中，指针指向数据区。在有些目前的系统中，数据区的实际位置 (因此，指针的值) 可以在处理数据的同时改变。于是系统必须注意位置与数据之间一致性。由于多个处理器能执行一个任务，这些处理器必须被同步，以防止数据的不一致的使用。这个同步是通过使用一个保护一个任务的执行的子部分的信号标 118 而实现的。

图 2 表示每个包的状态。为了在从输入队列 102 读与向输出队列

104 写之间进行同步，每个包保持一个状态。这个状态能从写 200 变成满 202，从满 202 变成读 204，从读 204 变成空 206。最后这个状态 206 能再次变成第一状态 200。写状态表明数据正被写入包，该数据还没一致到能被读取；满状态表明数据已经被写，并且是一致的，能被读取；读状态表明数据正在被从包中读取；空状态表明所有数据已经被从包中读取，允许再次向包中写入新数据。

图 3 表示任务的信号标的状态。信号标 118 可处于两个状态中的一个。开始时，信号标处于未规定 (not-claimed) 状态 300。这个状态可变成规定 (claimed) 状态 302，规定状态能再次变成未规定状态 300。信号标的状态的改变一般是在读-改-写操作中进行的。在本发明中，状态改变以及相应的操作最好在一个原子操作中进行，以防止因其它处理器的干预更新而引起的不一致状态。

图 4 表示按照本发明的方法的主要步骤。这些主要步骤对媒体文件的解译和编码二者都是要进行的。第一步骤 S400 是初始化步骤，在初始化步骤中，各队列被分配给适当的任务，把各任务分配给适当的协处理器。此外，将任务细分为不同的线程，不同的处理器能执行每个线程。在下一个步骤 S402 内，通过取得任务的信号标 118 在有资格执行任务的不同协处理器之间实现同步。当得不到任务的信号标时，特定的协处理器可在一个外加步骤内（未予示出）等待正确的状态。然后在步骤 S402 内，信号标 118 的状态从未规定 300 变成规定 302。当信号标是规定时，其它处理器不能规定该信号标。只有规定信号标 118 的协处理器才被允许更新任务的读和写指针，将它们指向其中的数据要被读和写的下一个包。该包内包含的数据不受该信号标保护。在步骤 S404 内，该任务访问输入队列和输出队列，以便读取输入队列中由任务的读指针所指示的输入包的状态。此外，该任务还读取输出队列中由任务的写指针所指示的输出包的状态。任务的读指针和写指针都被递增，这样在下一次，任务就读取要被处理的下一个包的状态。在步骤 S406 内，检查各包的状态。任务只能使用输入队列中的满 202 包和输出队列中的空 206 包。这样，就能防止该任务因为在先任务没有完成写数据而读取不一致的数据并且防止该任务重写输出队列上的数据，因为输出队列还包含必须被后继任务读取的数据。注意，其它任务仍然能将该包的状态从读变成空和从写变成满，因为信号标并不只

是保护这些包从空到写和从满到读的状态变化。当输入包的状态不是满 202 或者输出包的状态不是空 206 时, 已经对该信号标作了规定的协处理器就必须在步骤 S408 内等待, 直到两个状态都变成正确的状态: 在输入是满 202, 在输出是空 206。然后, 在步骤 S410, 输入包的满状态 202 被改变成写状态 204, 输出包的空状态 206 被变成写状态 200。此外, 当该数据包被执行该任务的协处理器共享的数据时, 该共享数据在可选的步骤 S412 中被传送到共享存储器中。这种共享数据能影响所有各相关协处理器对该任务的执行, 它可能包含要被该任务处理的其它参数集合, 或者它可能包含只有特殊的任务能解释的特定数据。为了防止对共享数据的不一致的更新和/或使用, 该数据还被拷贝到一个处理器的自己的地址空间。在将共享数据传送到共享存储器后, 信号标在步骤 S414 中被解除。如果不执行步骤 S412, 则步骤 S414 跟在步骤 S410 之后。其它协处理器一旦准备好要去处理队列中后面的包时, 就能开始规定该信号标。在步骤 S416 内, 该数据被实际处理, 处理该数据的协处理器能占用其需要的所有时间来进行处理。协处理器结束处理数据时, 在步骤 S418 内, 包的状态改变。输入包的状态从读 204 变成空 206, 输出包的状态从写 204 变成满 202。至此, 不需要规定一个信号标, 但是同一个协处理器不能立即访问这些包, 因为在其它协处理器上运行的任务可能会访问这些包。

一般来说, 所述的方法与分配给一个任务的处理器器的数量无关, 因此, 通过把同一个任务分配给更多处理器能透明地提高一个完整应用程序的速度, 而不需要对该应用程序重新设计, 细分成不同的任务。重新分配可以实时地进行, 使协处理器向任务的分配实质上不是一个设计问题。此外, 所述的方法从输入队列到输出队列保留了包的顺序, 而与多个运行其它包上相同任务的协处理器的处理时间的差别无关。当为每个队列单独引入一个信号标时, 如果不采取适当措施, 例如通过向能被平等地比较的包引入额外的参数, 则包的顺序可能不能维持。为每个包使用一个单独的信号标可以实现一个更灵活的输入包与输出包的比率, 例如读输入队列中的一个包, 而写输出队列中的两个或更多的包。

本发明方法的所述实施例中的顺序不是强制性的, 本领域的熟练人员在不偏离本发明的构思的情况下, 可以利用线程模型、多处理器

系统或多处理器，改变步骤的顺序或同时执行各步骤。

图 5 示意性地表示按照本发明的系统的主要部分。系统 500 既能解译又能编码 MPEG 视频和音频流之类的媒体流。系统包含两个协处理器 502 和 504，它们被分配去执行一个包含多个能被不同的协处理器执行的线程的任务。此外，系统还包含另一个协处理器 520，它能执行由系统执行的其它任务。存储器 506 包含输入队列，输入队列含有带有要被任务处理的数据的包。存储器 508 包含输出队列，输出队列含有带有已被任务处理过的数据的包。存储器 510 包含如前面所述的信号标。任务能在一个原子操作中改变存储器 510 的内容。此外，存储器 512 包含输入包的状态，而存储器 514 包含输出包的状态。存储器 516 包含处理器 502 和 504 的等待状态。这个存储器还包含用来确定处理器的状态的软件，而存储器 518 包含能改变包含输入和输出包的状态的存储器的状态的软件。协处理器之间的共享参数被包含在能被任务更新的存储器 526 内。所述各存储器和协处理器通过软件总线 530 互相通信连接。此外，所述的任务是能播放音频、视频、显示图像等的多媒体应用的一部分。系统 500 以软件方式实现，旨在作为由计算机或由任何能运行软件的其它标准体系结构运行的应用软件而运行。系统 500 也可以以硬件方式实现，并且系统能被用来操作数字或模拟电视机 522。软件也能根据从存储设备 524 中得到更新，该存储设备包含被安排得能执行按照本发明的方法的计算机程序产品。该存储器件由与系统 500 相连的适当的读取装置读取，例如由 CD 读取装置 528 读取。

图 6 示意性地表示包含按照本发明的系统的实施例的电视机 610。这里，天线 600 接收电视信号。任何能接收或再现电视信号的装置，例如卫星碟式天线、电缆、存储装置、互联网或以太网等等也能代替天线 600。接收机 602 接收信号。信号可以是例如数字的、模拟的、RGB 或 YUV 的。除了接收机 602，电视机还含有可编程部件 604，例如可编程集成电路。这个可编程部件含有按照本发明的系统 606。电视屏幕 608 显示被接收机 602 接收并被可编程部件 604 处理的图像。在另一个例子中，当用户想要录制所接收的信号例如电影时，按照本发明的系统 606 就在 DVD+RW、光盘或硬盘之类的记录装置上记录所接收的信号。在另一个例子中，当用户想要播放一个录制的电影时，按照本

发明的系统 606 就从记录装置中检索出适当的数据。

图 7 表示 MPEG 视频解码器的处理任务的示意图。视频流由流源 702 提供。任务 704 执行报头检测并将该流分成单独的片 (slices)。这个任务是一个不可分割的任务, 最好不要被划分给多个处理器。通过
5 缓冲器 706 将一个片传送到任务 708。708 执行对该片的实际的霍夫曼解译 (Huffman decoding)。被解译的数据现在已经转变成宏块流, 然后分成多个任务。宏块流的系数部分被执行逆量化 (inverse quantisation) 的任务 710 使用, 并被传送到执行逆 DCT 的任务 712。宏块流的向量部分被分别对前向和后向帧执行运动补偿的任务 722 和
10 726 使用。这两个预测然后被任务 724 相加和平均, 形成最后的预测。这个最后预测然后被任务 714 加到预测误差信号上。结果是最终解译的图像, 并被任务 716 存储在帧存储器中。任务 716 从存储自由帧的显示任务 720 得到帧缓冲器标引。一旦一个图形的所有宏块都已经被写到该帧, 就将帧标引传送到任务 718, 即帧管理器。这个帧管理器决
15 定哪些帧要被用作锚帧 (anchor frame), 哪些帧要被显示。锚帧被任务 722 和 726 用于对后面各帧的运动预测。要被显示的帧被传送到任务 720, 任务 720 进一步显示新帧并将旧帧送回到自由列表。在这个体系结构中, 所有可应用的任务都能被多个协处理器处理, 其中, 同步是按照前面所述的按照本发明的方法进行。

20 图 8 示意性地表示包含按照本发明的实施例的机顶盒的最重要部分。这里, 天线 800 接收电视信号。该天线也可以是例如卫星碟式天线、存储装置、互联网、以太网或其它任何能接收电视信号的装置。机顶盒 802 接收该信号。信号可以是例如数字的、模拟的、RGB 或 YUV 的。除了机顶盒中通常含有但在这里没有显示的部分外, 机顶盒还含
25 有一个按照本发明的系统 804。当用户想要录制所接收的信号例如电影时, 按照本发明的系统 804 就在 DVD+RW、光盘或硬盘之类的记录装置上记录所接收的信号。当用户想要播放一个录制的电影时, 按照本发明的系统 804 就从记录装置中检索出适当的数据。电视机 806 能显示被机顶盒 802 从接收的信号中生成的输出信号。

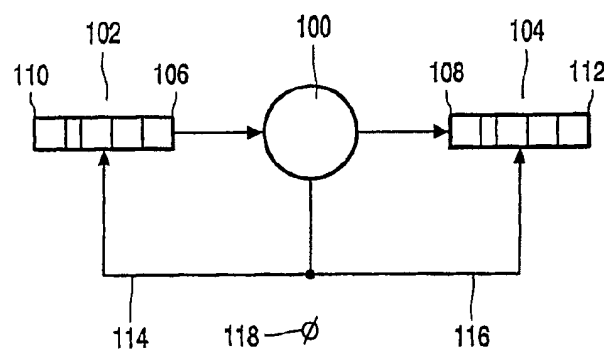


图 1

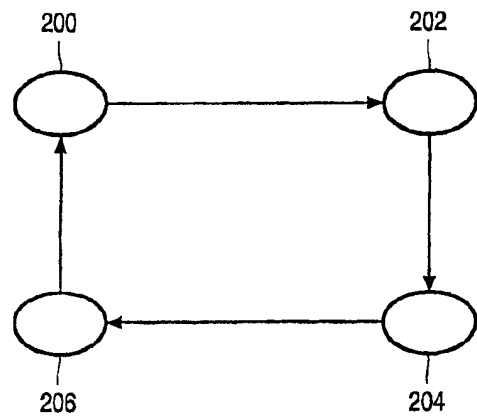


图 2

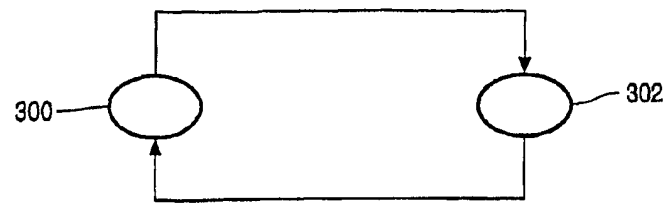


图 3

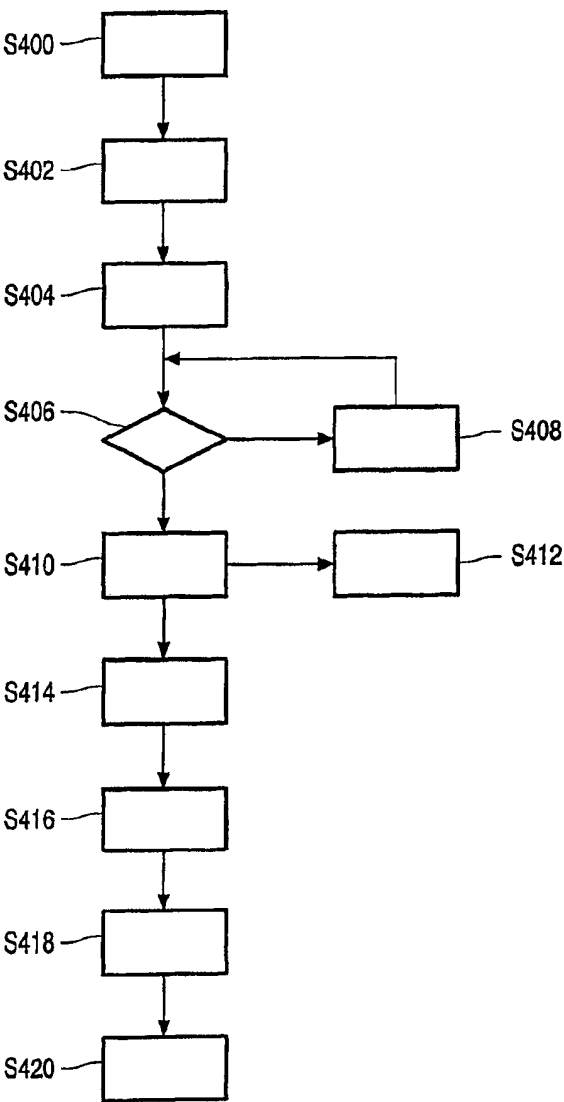


图 4

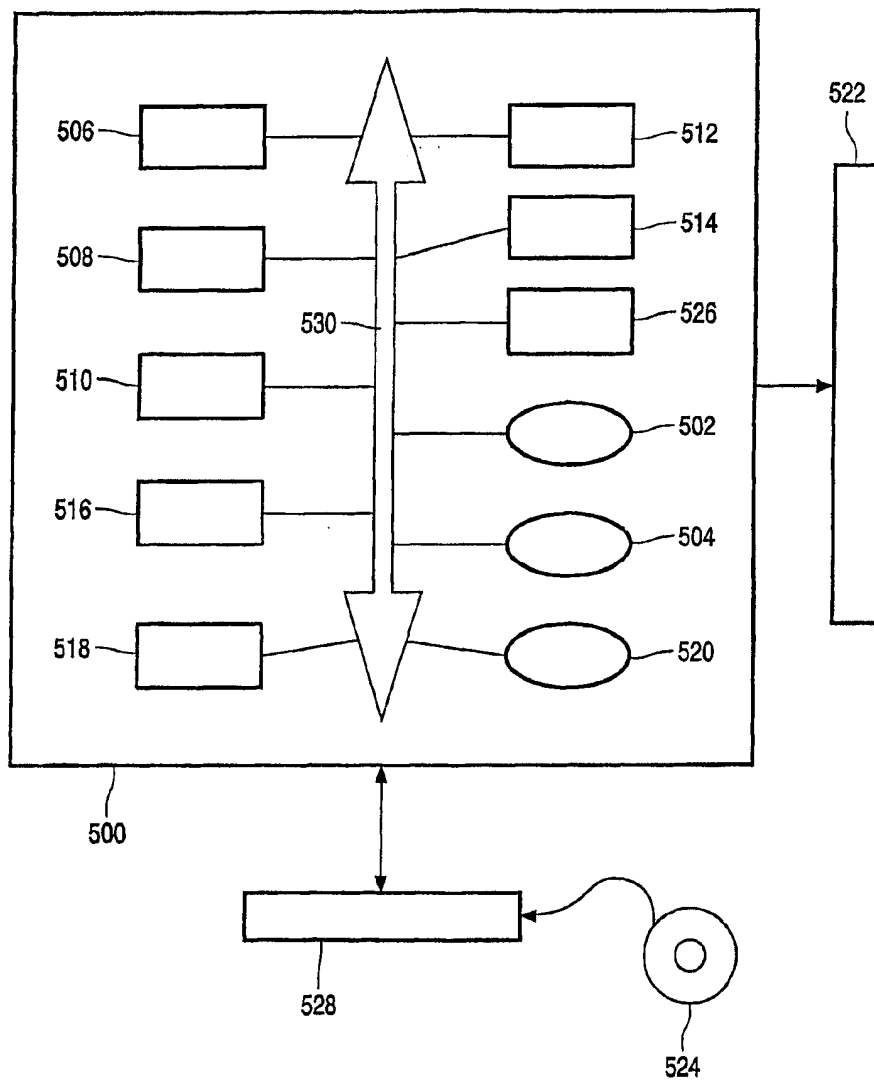


图 5

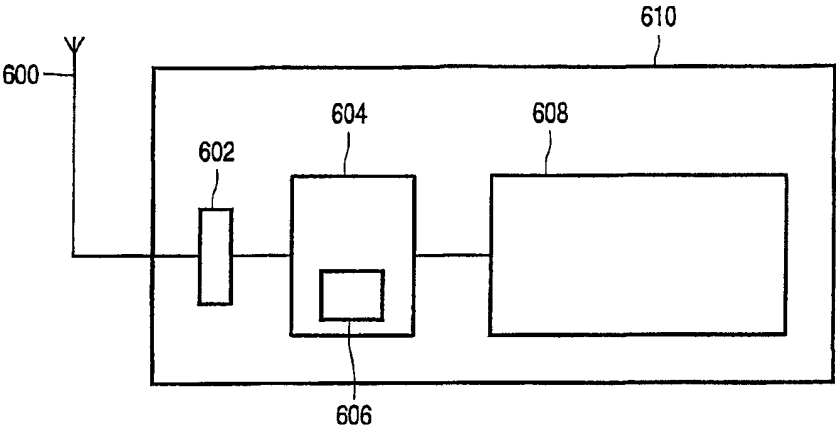


图 6

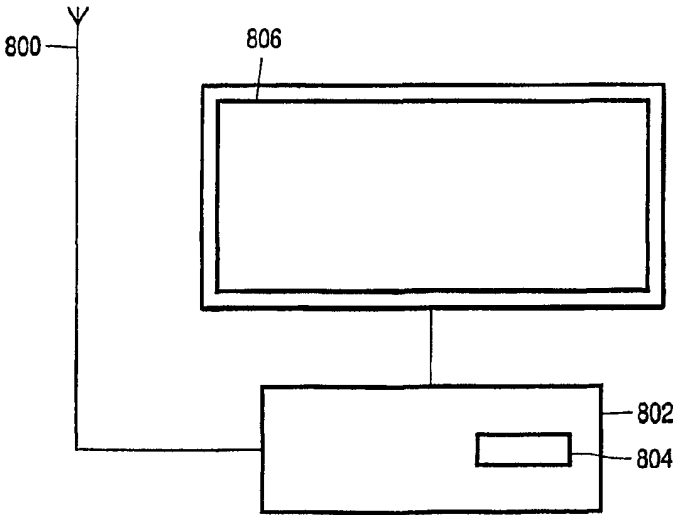


图 8

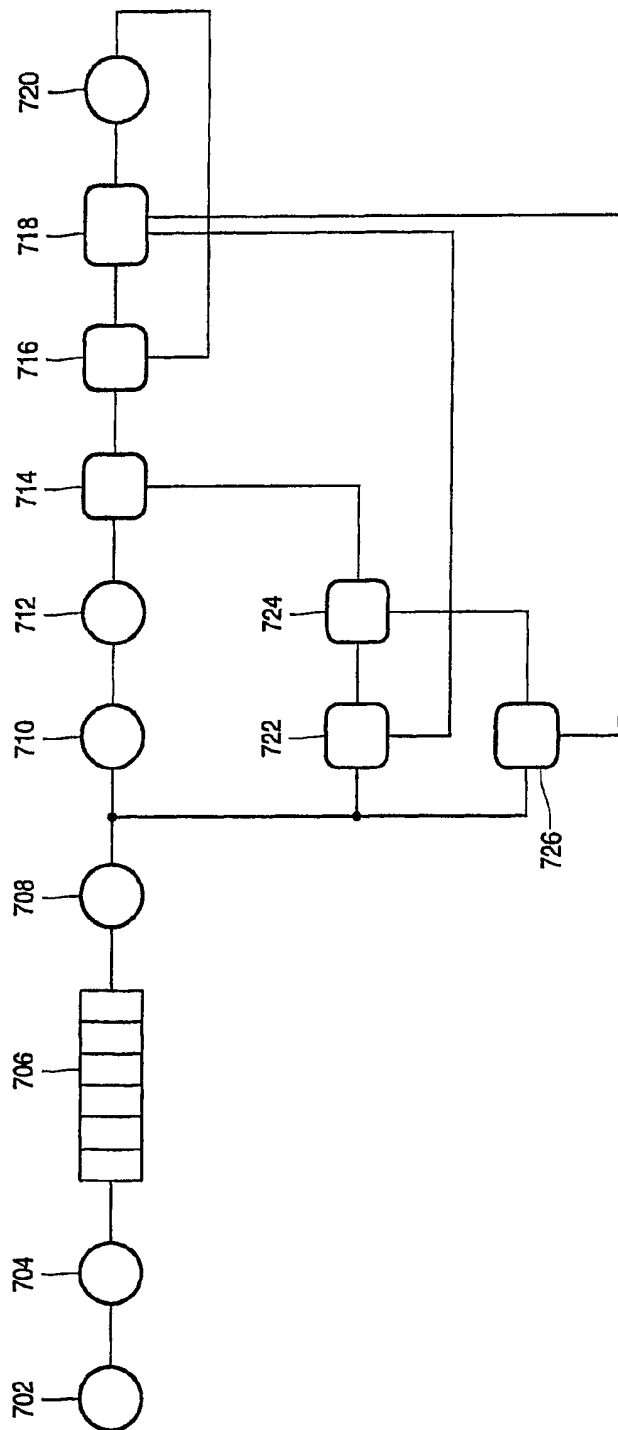


图 7