

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04B 7/26 (2006.01)

H04Q 7/20 (2006.01)

G06F 17/00 (2006.01)



[12] 发明专利说明书

专利号 ZL 200410076677. X

[45] 授权公告日 2007 年 12 月 26 日

[11] 授权公告号 CN 100358266C

[22] 申请日 2004.8.6

[21] 申请号 200410076677. X

[30] 优先权

[32] 2003.8.8 [33] EP [31] 03291999.5

[73] 专利权人 三菱电机株式会社

地址 日本东京都

[72] 发明人 C·穆伊 A·孔托里斯

A·比西奥克斯

[56] 参考文献

WO0190890A1 2001.11.29

WO0190891A1 2001.11.29

Reconfigurable terminals: An Overview of Architectural Solutions. Mehul Mehta et al. IEEE Communications Magazine, Vol. 39 No. 8. 2001

审查员 陈姗姗

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 李亚非 刘杰

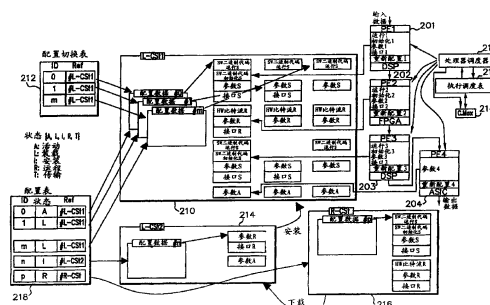
权利要求书 2 页 说明书 8 页 附图 3 页

[54] 发明名称

具有软件组件的电信设备

[57] 摘要

一种电信系统包括：处理单元(201、202、203、204)；存储器(L-Cst1)；至少一个在处理单元上实现的和使用所述存储器的软件组件(201、202、203、204)，每个软件组件包括：一组属性，每个属性是对存储配置数据的存储器中特定存储单元的指针，这些配置数据定义软件组件的特征，至少一个运行属性定义用于处理该软件组件中数据的运行时间处理；重新配置方法，适用于将属性的链接从所述存储器中的最初存储单元切换到所述存储器中的最终存储单元；基于对该软件组件或每个软件组件的重新配置方法的请求，用于触发的装置；根据预定调度，用于激活由该软件组件或每个软件组件的运行属性所定义的运行时间处理的装置。



1. 一种电信系统, 包括:

-至少一个处理单元 (201、202、203、204);

-至少一个存储器 (L-Cst1);

-至少一个在处理单元上实现并使用所述存储器的软件组件, 每个软件组件包括:

o 一组属性, 每个属性是对存储配置数据的存储器中特定存储单元的指针, 这些配置数据定义软件组件的特征, 至少一个运行属性定义用于处理该软件组件中数据的运行时间处理;

o 重新配置方法, 适用于将属性的链接从所述存储器中的最初存储单元切换到所述存储器中的最终存储单元;

-基于请求用于触发该软件组件或每个软件组件的重新配置方法的装置;

-根据预定调度, 用于激活由该软件组件或每个软件组件的运行属性所定义的运行时间处理的装置。

2. 根据权利要求 1 的电信系统, 其特征在于: 所述或每个软件组件包括初始化属性, 该初始化属性适用于为进行由运行属性所定义的运行时间处理而准备软件组件。

3. 根据任一前述权利要求的电信系统, 其特征在于: 所述一组或每组属性包括定义由一组连续步骤组成的算法的处理属性, 此算法由处理数据的处理单元实现。

4. 根据权利要求 1 或 2 的电信系统, 其特征在于: 所述一组或每组属性包括对至少一个由处理数据的处理单元所实现的处理的特定特征进行定义的参数属性。

5. 根据权利要求 1 或 2 的电信系统, 其特征在于: 所述一组或每组属性包括对即将处理数据的来源和必须输出处理数据的目的地进行定义的接口属性。

6. 根据权利要求 1 或 2 的电信系统, 其特征在于: 它包括用于修改预定调度的装置 (210, 212)。

7. 根据权利要求 1 或 2 的电信系统, 其特征在于: 所述存储器包括为所

述或每个软件组件保留的存储器存储单元，所述或每个软件组件的所述属性存储于所述保留的存储器存储单元。

8. 根据权利要求 1 或 2 的电信系统，其特征在于：它包括用于在所述存储器中存储至少两组所述或每个软件组件的配置数据的装置；所述运行属性方法只链接到所述或每个软件组件的一组配置数据。

9. 根据权利要求 1 或 2 的电信系统，其特征在于：所述一组或每组存储在所述存储器中的配置数据位于在设计时所定义的特定存储器单元。

10. 根据权利要求 1 或 2 的电信系统，其特征在于：它包括一个表 ExecutionSchedulingTable，该表包含系统的所有处理元件，该处理元件包括处理数据的所述或每个软件组件，并且根据预定调度用于激活由所述或每个软件组件的运行属性所定义的运行时间处理的装置适用于激活由该表内软件组件的运行属性所定义的运行时间处理。

11. 根据权利要求 1 或 2 的电信系统，其特征在于：它包括软件组件链，并且为了在系统正在通信时实时地重新配置，用于触发软件组件重新配置方法的装置以及激活软件组件运行时间处理的装置被应用在相同的软件组件链上。

具有软件组件的电信设备

技术领域

本发明涉及一种可重新配置的电信单元。

背景技术

由于技术和成本的约束，目前的无线电信系统专门使用一种通信标准或者通信方式。这就是为什么目前系统物理层处理的大部分仍然保留为由专用的预设电路功能（ASIC）组成。为了保持一定的灵活性（经常用以补偿 ASIC 的不足），有时可以引入某些可编程的功能以便于设计。

这意味着在系统设计和使用中存在一定僵化。

一种新的方法正在凸现：软件无线电(Software Radio)。最初的动机在于设计成本和上市时间。这可以被看作目前在电信系统设计中趋势的正常演变。在过去数年中，比物理层更高的层已经越来越多地以数字和软件的方式实现。软件无线电意欲向物理层推广此趋势。但是由于物理层很强的实时嵌入的约束，这使新的范例和方法成为必要。由于数字还意味着可容易改变，它们原则上是基于数字技术（尽可能多的）应用。

发明内容

本发明目的在于实时重新配置无线移动站或者基站收发信机。

“实时”重新配置在此必须被理解为这样的能力：也就是当通信系统正在通信时能够重新配置该系统且不中断该通信所支持的服务。

某些情况使无线收发信机的实时重新配置成为必要：性能增强、错误修复、环境适应、自动优化……。

将要解决的问题在于定义软件结构，其允许在通信系统自己数据处理期间实时重新配置处理数据的通信系统。

这里所关心的可重新配置的处理元件具有这样的特性：在通信工作期间，信息传输装置自己由所带来的信息内容重新配置。

为了实现上述目的，本发明提供了一种电信系统，包括：

- 至少一个处理单元；
- 至少一个存储器；
- 至少一个在处理单元上实现并使用所述存储器的软件组件，每个软件组件包括：
 - o 一组属性，每个属性是对存储配置数据的存储器中特定存储单元的指针，这些配置数据定义软件组件的特征，至少一个运行属性定义用于处理该软件组件中数据的运行时间处理；
 - o 重新配置方法，适用于将属性的链接从所述存储器中的最初存储单元切换到所述存储器中的最终存储单元；
- 基于请求用于触发该软件组件或每个软件组件的重新配置方法的装置；
- 根据预定调度，用于激活由该软件组件或每个软件组件的运行属性所定义的运行时间处理的装置。。

根据具体实施例，本发明包含一个或多个从属权利要求的特征。

附图说明

在阅读下列仅由示例方式给出的描述的基础上，并且同时参照附图可以更好地理解本发明，其中：

图1是根据本发明的通信设备的功能图；

图2是用于重新配置的数据结构，和用于实现具有最初配置的配置功能元件的示意图；

图3是重新配置通信设备的系统操作流程图中；和

图4是根据本发明，具有第二配置的通信设备的功能图。

具体实施方式

根据本发明的通信单元例如是无线移动站或基站收发信机。它可以仅仅是发射机或接收机。本发明还可以应用于定位系统、智能传输系统或广播系统。

如图1所示，通信系统硬件在发射端或接收端或在两端包含以下元件，有时包含其中一部分，或者有时包含其中至少一个：

一模拟输入 10，例如包含天线（或多个天线）、模拟滤波器（或多个模拟滤波

器)、功率放大器(或多个功率放大器)、双工器(或多个双工器)、混频器(或多个混频器)、振荡器(或多个振荡器)……;

- 接收端的模-数转换级 12, 或发射端的数-模转换级; 以及
- 处理设备 14, 例如包含处理器(或多个处理器)和/或可重新配置硬件和/或 ASIC。

根据本发明, 通信系统的全局处理功能被分为个别的处理(功能)元件。其中至少一个可由软件组件实现, 它可以由本方法重新配置。这类可重新配置的软件组件被称作代表软件定义的无线电(Software Defined Radio)组件的 SDR 组件。

每个 SDR 组件可以在以下设备上执行:

- 类似于支持非可变性的 ASIC 的完全专用硬件;
- 类似于数字或模拟 ASIC 的可参数化的硬件, 该 ASIC 行为可通过改变参数值而相对改变;
- 类似于通过使用比特流来支持重新配置的 FPGA 或 CPLD 的可参数化硬件;
- 类似于任何种类的处理器可编程软件, 处理器包括: GPP、 μ C 或支持可执行代码形式的软件的 DSP。

根据本发明, 每个 SDR 组件由属性和一种方法组成。

SDR 组件的属性是对配置数据的引用, 该配置数据存储在存储器中且定义由处理数据的软件组件实现的处理。有利地, 它们有下列四种类型:

- 运行
- 初始化
- 参数, 和
- 接口。

运行和初始化属性是定义 SDR 组件的运行时间处理或初始化处理的属性。每个运行或初始化属性分别包含对运行时间处理或初始化处理的代码存储单元的引用。它们包含处理器的代码、可重新配置硬件的比特流。例如, 对代码存储单元的引用是指针。

参数属性包含对可以影响代码工作的数据的引用、对比特流的引用或对取决于处理单元的电缆连接前(pre-cabled)行为的引用。

接口属性包含在全局处理链路中 SDR 组件连接的引用。接口属性定义输入数据的源和必须输出的已处理数据的目的地。

运行属性指向运行时间处理，并与对应的参数有关，以便通过对应的接口消耗和生成将被处理的数据。

初始化属性也工作在处理属性及其相应参数的初始化处理上，且具有与运行属性相同的接口。如果需要，有可能从剩余的 SDR 组件中分离出初始化以形成具有不同接口的独立 SDR 组件。

SDR 组件的功能特性通过运行和初始化属性与参数和接口属性的指示地址得以确定。

其运行和初始化属性的指示地址未定义的 SDR 组件被称作白组件，由于它们仅仅保留了不属于任何当前 SDR 组件的存储空间，但却用于未来扩展。

图 2 示出了包括 SDR 组件的通信单元的软件排列的示例。

为了清晰起见，假定每个 SDR 组件将在特定的处理单元上实现。

由于 SDR 组件 201、202、203、204 被连接以形成处理链路的子集，SDR 组件 201 和 203 在 DSP 上实现，SDR 组件 202 和 204 分别在 FPGA 和 ASIC 上实现。

如前所述，每个 SDR 组件包括运行和初始化处理属性、参数属性和接口属性。这些属性根据它们所从属的 SDR 组件在附图上被编上号码。

每个 SDR 组件还包括它自有的重新配置方法。

因为 SDR 组件在 ASIC 上实现，所以只显示了参数属性。实际上，由于对应的特征不可重新配置，其它的属性也不存在。

为每个 SDR 组件保留一组存储器的存储单元以支持多种配置。这些存储器的存储单元位于被称作 L-Cst 1 的第一级存储器 210 中。这些存储单元的数量取决于系统同时保持的配置数量。这些存储单元中的每个存储单元包含每个 SDR 组件的处理属性、参数和接口中的一个或多个的新版本。

每个保留的存储器存储单元包含通信单元特定配置的配置数据。由配置数据 (configDATA) 表定义每个存储器存储单元的排列。

配置数据 (configDATA) 是一种结构，其包含对存储器区域的引用，该存储器区域包含允许重新配置一个或一组 SDR 组件的实际数据。该实际数据是组件的初始化与运行时间处理属性、参数、接口的二进制信息或代码。在图

2 中, 最初的重新配置操作 (CongfigDATA #0) 涉及在三种设备上可改变的所有领域, 而第二和第三重新配置操作只是部分的领域。

配置切换表 (CongfigSwitch Table) 212 被定义用于内部的簿记 (bookkeeping)。对于这些配置中的每一个配置, 该表指示包含对配置数据的结构的引用。

在每个 SDR 组件中, 每个属性是到 LCst-1 中存储器存储单元的指针, 该存储单元包括运行和初始化处理方法、参数或接口的实现。

在图 2 上, 假设 CongfigDATA #0 将被使用, 因此, 所有的属性指向对应的 CongfigDATA #0 的存储单元。

通信单元中的存储装置被编组为多级高速缓冲存储器。在此定义了三级。

第一级 (L-Cst1) 对应于存储在执行存储器中的配置, 即, 例如处理器内部存储器。预安装这些配置并为在某些初始化之后的执行做准备。仅通过向有关组件扩散参数值并解析软件组件函数的指针引用, 执行在这类配置间的切换。

第二级 (L-Cst2) 对应于存储在第二级存储器中的配置, 即, 例如处理器外部存储器。通过首先使用某些数据传输装置 (例如后台 (background) DMA) 将配置数据带入执行存储器 (第一级), 执行对这些配置的切换。已传输的数据替代某些其它的第一级配置或被存储于自由的空间。然后继续如前所述进行切换。

第三级 (R-CST) 对应于存储在一些远程地点的配置。如果请求重新配置, 这类配置可以直接向 L-Cst1 传输, 或者如果只请求更新本地存储的配置, 则这类配置可以向 L-Cst2 传输。通过建立基于某些保证无差错数据递送的传输协议的无线通信链路, 执行该传输。

在 L-Cst1、L-Cst2 和 R-CST 中的配置对应于系统支持的配置。为了在高速缓冲存储器中追踪配置的位置, 实现涉及数据库组织的某种簿记。通过序列表或者其它的由配置标识符提供索引的随机存取数据结构, 可以有效地实现该簿记。

在 L-Cst1 存储器中存储的配置数据可以被来自通信单元中表示为 L-Cst2 的第二级存储器的其它配置数据替换。

配置表 218 指示 ConfigTable, 被提供用于指示配置信息的存储单元。

ConfigTable 包含每个支持的配置的入口; 安装在执行存储器中的配置以

及存储于外部存储器中的配置。配置的标识符（configID）是访问配置入口并最终访问配置数据的索引关键字。在一个简单的实现中，ConfigTable 被实现为序列表，并且 configID 用于给 ConfigTable 编索引。

ConfigTable 包含每个支持的配置的状态。该状态指示配置在哪里存储以及它们是否被使用。

可能的状态是：

- A 代表活动（Active），意指配置数据由 SDR 组件实现；
- L 代表装载（Loaded），意指配置数据存储在 L-Cst2；
- I 代表安装（Installed），意指配置数据存储在 L-Cst1；
- R 代表远程装载（RemotedLoaded），意指配置数据存储在 R-Cst2；
- T 代表传输（Transfert），意指它们正在被传输，且因此不可用。

提供处理器调度器 210 和执行调度表（ExecutionSchedulingTable）212 用于管理通信单元的全部处理。处理器调度器适用于根据无限循环触发可以是 SDR 组件或者不是 SDR 组件的每个处理元件。将参照图 3 公开特定 SDR 组件的循环步骤。

执行调度表 212 包含处于执行顺序中的将被执行的 SDR 组件列表。处理器调度器覆盖执行调度表以计算此处理。

由与应用有关的数据相关性确定执行调度表。

为了有效处理当通过重新配置而删除(suppress)SDR 组件时留下的漏洞，或者当通过重新配置添加 SDR 组件时，有效处理由插入新 SDR 组件所引起的移位，执行调度表被实现为链接表，但是其它的选项也是可能的。

为了管理该重新配置，通信单元包括表示为 CMan 的配置管理器 214。

配置管理器（CMan）负责启动和管理该重新配置。Cman 可以自己决定或基于网络其它元件的请求而决定启动重新配置。通过由 CMan 控制的重新配置处理（RP），展开正确的配置。该处理设置硬件参数、设置软件参数并以指定的顺序执行必要的软件更新。它还保证在重新配置处理期间内及其后进行一致的系统操作。通过每个系统组件（硬件或软件）给出的良好定义的控制接口，CMan 可以控制该处理。如果配置在 L-CSt 中不是本地可获得的，CMan 还负责启动传输处理，以从远程配置存储（R-CSt）获得所需的数据。通常，该重新配置处理至少包括下列阶段中的一个或其子集的组合：下载-安装-切换阶段。

确切的组合取决于所需重新配置的可用性（本地可用的或非本地可用，其可通过在 ConfigTable 中检查状态字段得以确定）以及可限制系统（重新）配置能力的特定布置。

通信单元如下展开工作。

在运行时间，处理元件通过它们的接口（输入、输出等）通信，并且整个处理在处理器调度器 210 的驱动下循环进行。在无限循环的每个迭代之际，输入抽样块被处理以生成输出抽样块。在每个迭代开始之际，执行 CMan 处理。

对于每个迭代，图 3 的算法被应用于全局处理链路，该链路包含如执行调度表所示为活动状态的 SDR 组件。

CMan 处理对在 ConfigDATA 中存在的所有 SDR 组件进行重新配置。

只有下一个配置数据已经被预先存储在 L-Cst1 中时，才可以做出从先前的配置到下一个配置的切换。

在下文，我们假定下一个配置的数据是 L-Cst1，接着将公开如何管理配置在 L-Cst1、LCst-2 以及 R-Cst 之间的移动。

在左手侧示出了在处理单元上实现的 SDR 组件的循环处理。它表示所执行的通信单元无限循环的每个迭代，以及每次处理输入抽样块以生成输出抽样块。更确切地，在步骤 302 读取输入抽样，并且为了根据由如通过运行属性定义的 SDR 组件当前使用的运行时间处理来生成处理过的抽样，由处理单元在步骤 304 处理输入抽样。在步骤 306 输出该处理过的抽样。

在每个迭代开始之际，在步骤 310 执行 CMan 处理。它的细节在图 3 右手侧的流程图中示出。CMan 处理是基本的。在步骤 314，根据例如被用户或本地管理员激活的重新配置标志，CMan 确定是否需要重新配置。

在不要求重新配置的情况下，Cman 开销仅仅在于对重新配置条件的评价。所以处理单元的发射机/接收机的工作不被中断。

如果要求重新配置，遭受的开销延迟的长度主要取决于对硬件级的接口有多快。R/T 的中断取决于富裕时间的数量，此富裕时间被定义为输入块变得可用的时间与完成先前块处理的时间之间的差值。

在步骤 318 由 CMan 在 L-Cst1 中确定 configDATA 的存储单元。

在已经发现 configDATA 之后，应 CMan 的处理请求，其属性处于 ConfigDATA 中的 SDR 组件重新配置的方法被触发。它在步骤 320 实现以便

重新配置 SDR 组件。为了重新配置，每个属性的指针由在新的配置数据中对这些属性所定义的指针所代替。

另外，当重新配置发生时，为了与新的配置数据匹配，由 Cman 更新执行调度表。

如果重新配置未在数据相关性和 SDR 组件的互连中引入变化，在执行调度表中，SDR 组件在执行调度表中停留在相同的位置。如果重新配置在数据相关性和 SDR 组件的互连中引入了变化，在执行调度表中，SDR 组件被移动。

如果必要，在执行调度表中插入新的 SDR 组件。

从执行调度表中删除已取消的 SDR 组件。在这种情况下，未做出代码的插入或删除。它只是在平级上改变的一条链路。

图 4 示出一个重新配置的示例。

假设通信单元从 ConfigDATA#0 被切换到 ConfigDATA#1。因此，如果在 ConfigDATA#1 中声明了属性，SDR 组件保留所有来自 ConfigDATA#1 的属性而不是先前的属性。如果在 ConfigDATA#1 中未存有属性，仍旧使用由 ConfigDATA#0 所定义的先前对应属性。

如图 4 所示，来自 SDR 组件的指针被指向对应 ConfigDATA#1 的第二列参数而不再指向第一列参数。

在图 4 所考虑的示例中，ConfigDATA#1 改变 SDR 组件 201 的使用、参数和接口属性。对于 SDR 组件 202，除接口属性之外的所有属性被改变。对于 SDR 组件 203，修改了参数和接口属性。因此，由于由 SDR 组件 201 处理的数据直接被 SDR 组件 203 采用，所以新的联系得以在 SDR 组件 201 和 203 之间建立。

对于 SDR 组件 204，参数属性被改变。因为 SDR 组件 204 是 ASIC，所以只有该属性可以被修改。

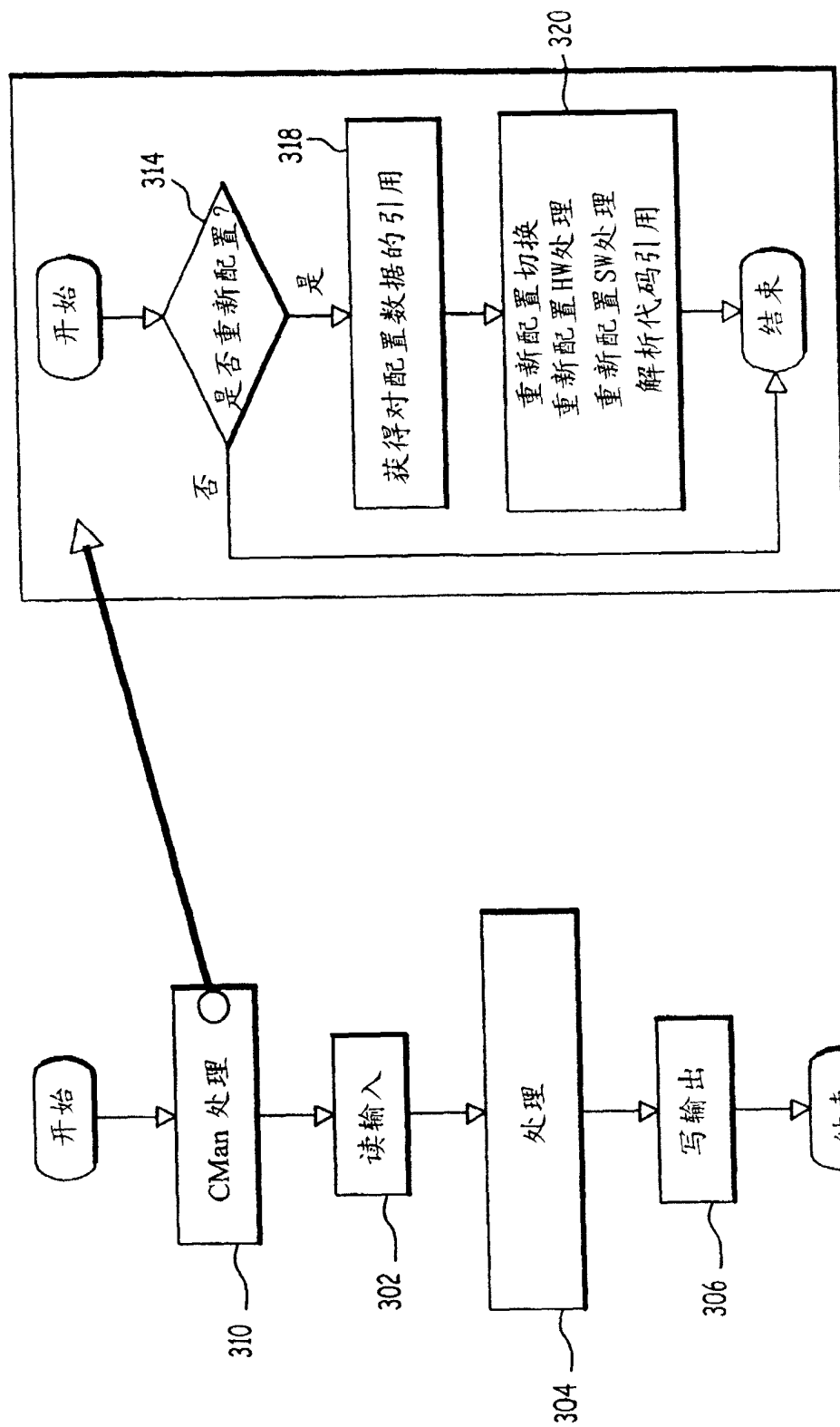


图 3

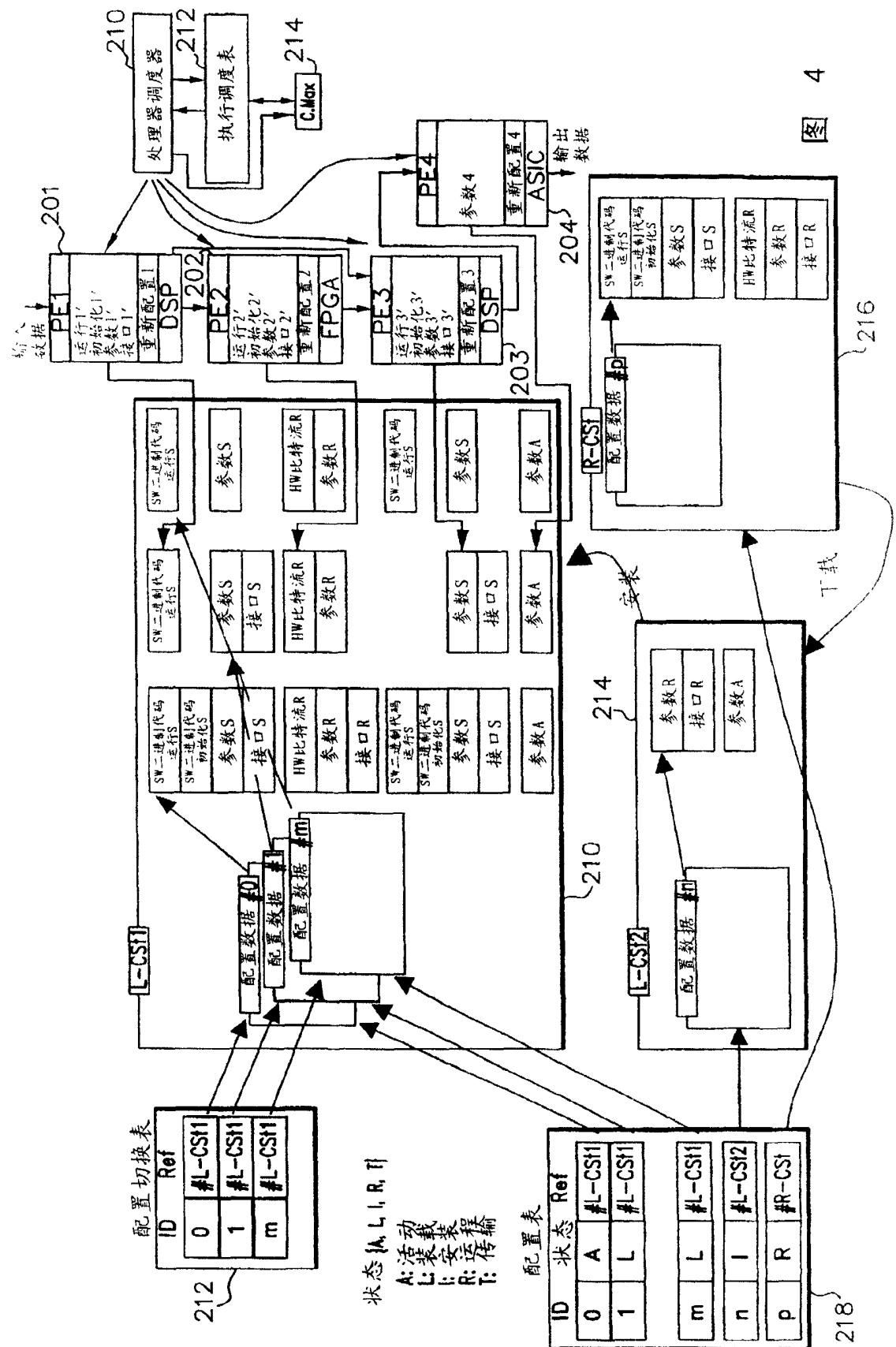


图 4