[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 11/30 (2006.01)

[12] 发明专利说明书

专利号 ZL 03157874.8

[45] 授权公告日 2008 年 1 月 2 日

[11] 授权公告号 CN 100359481C

[22] 申请日 2003.9.13 [21] 申请号 03157874.8

[73] 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

[72] 发明人 陈志强

[56] 参考文献

JP5346878A 1993.12.27

CN1400529A 2003.3.5

审查员 王 骞

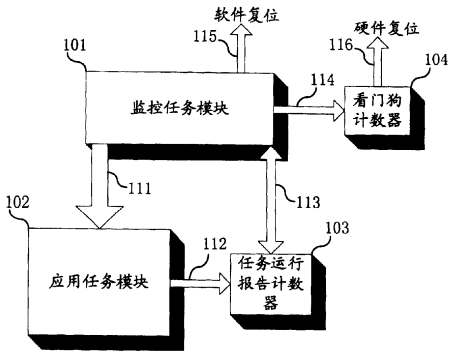
权利要求书 3 页 说明书 15 页 附图 3 页

[54] 发明名称

多任务系统的异常监控装置及其方法

[57] 摘要

本发明涉及电子系统的异常检测和处理，公开了一种多任务系统的异常监控装置及其方法，使得多任务系统的异常监控和自动复位到正常运行状态的功能得到实现。 这种多任务系统的异常监控装置包含一个用于在计数溢出时触发所述多任务系统复位的看门狗计数器，一个监控任务模块以及至少一个应用任务模块，所述异常监控装置还包括至少一个与所述应用任务模块相应的任务状态报告计数器，其中：所述应用任务模块用于通过清空相应的任务状态报告计数器向所述监控任务模块报告运行状态，监控任务模块根据应用任务模块报告的运行状态以及任务状态报告计数器的计数值对所述应用任务模块进行相应的处理。



1. 一种多任务系统的异常监控装置，包含一个用于在计数溢出时触发所述多任务系统复位的看门狗计数器，一个监控任务模块以及至少一个应用任务模块，其特征在于，所述异常监控装置还包括至少一个与所述应用任务模块相应的任务状态报告计数器，其中：

所述应用任务模块用于通过清空相应的任务状态报告计数器向所述监控任务模块报告运行状态；

所述监控任务模块用于清空所述看门狗计数器并对所述任务状态报告计数器进行计数操作，并根据应用任务模块报告的运行状态以及任务状态报告计数器的计数值对所述应用任务模块进行相应的处理。

2. 根据权利要求 1 所述的多任务系统的异常监控装置，其特征在于，所述监控任务模块中任务运行的优先级高于所述应用任务模块中任务运行的优先级。

3. 一种多任务系统的异常监控方法，其特征在于，包含以下步骤：

A、监控任务模块清空看门狗计数器，同时对与应用任务模块相对应的任务状态报告计数器进行计数操作；

B、所述应用任务模块通过清空相应的任务状态报告计数器向所述监控任务模块报告运行状态；

C、所述监控任务模块根据所述应用任务模块报告的运行状态以及任务状态报告计数器的计数值对所述应用任务模块进行相应的处理。

4. 根据权利要求 3 所述的多任务系统的异常监控方法，其特征在于，在步骤 C 中监控任务模块对所述应用任务模块进行相应的处理步骤具体包括：

C1、所述监控任务模块判断所述应用任务模块相应的任务状态报告计数器的计数值是否超过第一门限，如果是，则进入步骤 C2，否则进入步骤 C3;

C2、所述监控任务模块对所述应用任务模块进行异常处理;

C3、所述监控任务模块完成对所述应用任务模块的监控。

5. 根据权利要求 4 所述的多任务系统的异常监控方法，其特征在于，所述步骤 C2 进一步包括:

C21、所述监控任务模块判断所述应用任务模块相应的所述任务状态报告计数器的计数值是否超过第二门限，如果是，则进入步骤 C22，否则进入步骤 C23;

C22、所述监控任务模块对所述应用任务模块进行进一步的异常处理;

C23、所述监控任务模块调用所述应用任务模块的任务唤醒函数，并完成对所述应用任务模块的监控。

6. 根据权利要求 5 所述的多任务系统的异常监控方法，其特征在于，步骤 C22 进一步包含以下步骤:

C221、所述监控任务模块判断所述应用任务模块相应的所述任务状态报告计数器的计数值是否超过第三门限，如果是，则进入步骤 C222，否则进入步骤 C223;

C222、所述监控任务模块保存现场并执行系统复位;

C223、所述监控任务模块对所述应用任务模块进行任务恢复，如果任务恢复成功，则完成对本应用任务模块的监控，否则进入步骤 C222。

7. 根据权利要求 3 至 6 中任意一条所述的多任务系统的异常监控方法，其特征在于，在步骤 A 之前进一步包含以下步骤:

应用任务模块向所述监控任务模块进行任务注册;

所述监控任务模块为请求注册的应用任务模块分配一个任务运行状态报告计数器；

所述监控任务模块记录由请求注册的应用任务模块提供的任务唤醒函数地址。

8. 根据权利要求 7 所述的多任务系统的异常监控方法，其特征在于，所述任务唤醒函数执行的操作是为处于等待状态的被监控应用任务模块分配一个虚拟任务，使该应用任务模块由等待状态变为就绪状态。

多任务系统的异常监控装置及其方法

技术领域

本发明涉及电子系统的异常检测和处理，特别涉及多任务电子系统的异常检测和处理。

背景技术

目前，无论在通信、计算机、自动控制领域的软件或硬件系统中，多任务并行的设计方法由于其对资源高效使用的特点而被广泛采用。随之而来的一些新问题也有待更好的解决，而多任务系统的异常监控问题就是其中一个。这个问题在当前新兴的多任务嵌入式系统技术中尤为突出。

在许多情况下，设计人员会用软件实现以往由硬件才能完成的电路功能，其中部分原因是低成本的微处理器(Micro Processor，简称"μP")为大家提供了广泛的选择。软件常常是解决问题成本最低、灵活性最高的方案，但它也迫使设计人员进行一些额外的测试以确保系统的可靠性。软件实现的系统特别是当前流行的嵌入式系统由于程序代码设计出现的没有被发现的错误，容易引起系统运行的异常，同时系统的硬件部分的不稳定也经常会导致系统运行的异常。在台式机系统中出现导致系统瘫痪的软件错误并不可怕，因为用户只需重新启动系统即可，它只会造成少量数据的丢失。然而，对于应用在工控系统中的嵌入式系统，则必须能够在没有人为干预的条件下处理异常、恢复故障。这一特性在两种情况下非常关键：一种是高有效性系统，如服务器、电话系统以及生产线等；另一种是高可靠性系统，因为这种系统一旦出现错误将造成伤害，如汽车、医疗设备、工业控制、机器人、自动门等。即使不考虑这些要求严格的应用，系统在无需用户干预的条件下自动(按下复位键或重新上电)从故障状态下恢复也是很有益处的，这种设备的好处是显而易见的，因为用户不希望设备内部出现问题。改善这类系统可靠性的

一种简单、有效的措施是采用异常监控机制，在普通的嵌入式系统中被称为看门狗（Watching Dog）。看门狗是一种用于监控系统运行情况的机制，当系统运行出现不能自我控制的异常时，为系统提供强制复位功能，用以恢复系统正常运行。

在单任务系统中看门狗技术已经比较成熟，基本的实现方法是在硬件上实现一个看门狗计数器，系统上电后看门狗计数器开始计数，当看门狗计数器发生溢出时就会导致系统复位。在软件上，把清空看门狗计数器的功能与系统正常功能捆绑到一起（一般是在任务循环体中插入清空看门狗计数器的操作），这样一方面系统正常运行时，软件能够周期的将看门狗计数器清空，看门狗计数器不会发生溢出；另一方面，如果系统运行出现异常，程序无法实现正常功能时，由于清空看门狗计数器的功能与系统正常功能捆绑在一起，程序也就无法清空看门狗计数器，超过预定时间后，看门狗计数器必然发生溢出，导致系统复位。这样就实现了系统故障情况下的自动复位恢复。

在实际应用中，上述方案存在以下问题：无法应用于多任务系统中，实现多任务系统中的异常监控。

造成这种情况的一个主要原因在于，目前的技术都是针对单任务系统的，只能保证单个任务的正常运行与异常监控，无法应用于多任务系统。例如，一个系统中存在三个任务，每一个任务的循环体中都有清空看门狗计数器的操作，则当其中两个任务发生出现异常时，因为还有一个任务依然在周期性地将看门狗计数器清空，因此系统无法自动复位，导致看门狗失去意义。

发明内容

本发明要解决的技术问题是提供一种多任务系统的异常监控装置及其方法，使得多任务系统的异常监控和自动复位到正常运行状态的功能得到实现。

为了解决上述技术问题，本发明提供了一种多任务系统的异常监控装

置，包含一个用于在计数溢出时触发所述多任务系统复位的看门狗计数器，一个监控任务模块以及至少一个应用任务模块，其特征在于，所述异常监控装置还包括至少一个与所述应用任务模块相应的任务状态报告计数器，其中：

所述应用任务模块用于通过清空相应的任务状态报告计数器向所述监控任务模块报告运行状态；

所述监控任务模块用于清空所述看门狗计数器并对所述任务状态报告计数器进行计数操作，并根据应用任务模块报告的运行状态以及任务状态报告计数器的计数值对所述应用任务模块进行相应的处理。

本发明还提供了一种多任务系统的异常监控方法，包含以下步骤：

监控任务模块清空看门狗计数器，同时对与应用任务模块相对应的任务状态报告计数器进行计数操作；

所述应用任务模块通过清空相应的任务状态报告计数器向所述监控任务模块报告运行状态；

所述监控任务模块根据所述应用任务模块报告的运行状态以及任务状态报告计数器的计数值对所述应用任务模块进行相应的处理。

本发明的技术方案与现有技术的区别在于，本发明为每个应用任务模块设置相应的任务状态报告计数器，所述应用任务模块用于通过清空相应的任务状态报告计数器向所述监控任务模块报告运行状态，所述监控任务模块根据应用任务模块报告的运行状态以及任务状态报告计数器的计数值对所述应用任务模块进行相应的处理。

这种技术方案上的区别，带来了较为明显的有益效果，即同时监控多任务系统中多个应用任务模块的运行状态，实现每一个应用任务模块的异常处理和自动恢复，实现多任务系统的异常监控，提高多任务系统的可靠性。

所述监控任务定期检查应用任务对应的计数器，确定应用任务的状态，并进行相应的处理。

在所述应用任务模块向所述监控任务模块报告运行状态的步骤中，进一步包含以下步骤：

所述应用任务模块在自身主循环体中清空任务状态报告计数器。

在所述监控任务模块监控所有应用任务模块的异常情况的步骤中，进一步包含以下步骤：

A 所述监控任务模块对所有应用任务模块相应的任务状态报告计数器进行计数操作；

B 所述监控任务模块判断所述应用任务模块相应的任务状态报告计数器的计数值是否超过第一门限，如果是，则进入步骤 C，否则进入步骤 D；

C 所述监控任务模块对所述应用任务模块进行异常处理；

D 所述监控任务模块完成对所述应用任务模块的监控。

所述步骤 C 进一步包含以下子步骤：

C1 所述监控任务模块判断所述应用任务模块相应的所述任务状态报告计数器的计数值是否超过第二门限，如果是，则进入步骤 C2，否则进入步骤 C3；

C2 所述监控任务模块对所述应用任务模块进行进一步的异常处理；

C3 所述监控任务模块调用所述应用任务模块的任务唤醒函数，并完成对所述应用任务模块的监控。

步骤 C2 进一步包含以下子步骤：

C21 所述监控任务模块判断所述应用任务模块相应的所述任务状态报告计数器的计数值是否超过第三门限，如果是，则进入步骤 C22，否则进入

步骤 C23;

C22 所述监控任务模块保存现场并执行系统复位;

C23 所述监控任务模块对所述应用任务模块进行任务恢复，如果任务恢复成功，则完成对本应用任务模块的监控，否则进入步骤 C22。

在所述应用任务模块向所述监控任务模块进行的任务注册的步骤中，进一步包含以下子步骤:

所述监控任务模块为请求注册的应用任务模块分配一个任务运行状态报告计数器;

所述监控任务模块记录由请求注册的应用任务模块提供的任务唤醒函数地址。

所述任务唤醒函数执行的操作是为处于等待状态的被监控应用任务模块分配一个虚拟任务，使该应用任务模块由等待状态变为就绪状态。

通过比较可以发现，本发明的技术方案与现有技术的区别在于，通过设置专门的监控任务模块来监控多个应用任务模块的运行状态;通过原看门狗技术实现对监控任务模块的异常监控;通过任务状态报告计数器实现监控任务模块对应用任务模块的监控;通过任务唤醒函数的调用来唤醒等待状态的应用任务模块;通过任务注册的方法，实现任务状态计数器的分配和任务唤醒函数调用地址的传递。

这种技术方案上的区别，带来了较为明显的有益效果，即同时监控多任务系统中多个应用任务模块的运行状态，实现每一个应用任务模块的异常处理和自动恢复，实现多任务系统的异常监控，提高多任务系统的可靠性。当系统中的某一个应用任务模块长时间没有反应时，系统会先后采用唤醒、恢复直至重新复位整个系统的手段，只用一个看门狗硬件就确保了多任务系统中每一个任务都能处在正常的工作状态。

附图说明

图 1 是根据本发明的一个实施例的多任务系统的异常监控装置示意图;

图 2 是根据本发明的一个实施例的多任务系统的异常监控装置的监控任务模块的流程图;

图 3 是根据本发明的一个实施例的多任务系统的异常监控装置的应用任务模块的流程图。

具体实施方式

为使本发明的目的、技术方案和优点更加清楚，下面将结合附图对本发明作进一步地详细描述。

本发明涉及一种多任务系统的异常监控装置及其方法。所述多任务系统可以是应用于各个领域的软件、硬件系统，比如通信、计算机、自动控制领域的软件系统、硬件系统、嵌入式系统。所述多任务是指，系统中同时存在多个任务并行处理，不同任务之间协调占用 CPU 处理时间，或者由系统按照某种原则统一分配 CPU 处理时间给各个任务。所述多任务系统的硬件组成可以是现场可编程门阵列(Field Programmable Gate Array，简称"FPGA")，专用集成电路(Application Specific Integrated Circuit，简称"ASIC")，中央处理器(Central Processing Unit，简称"CPU")，数字信号处理(Digital Signal Processing，简称"DSP")芯片，复杂可编程器件(Complex Programmable Logic Device，简称"CPLD")等。

下面详细描述本发明的一个较佳实施例，该实施例是针对多任务嵌入式系统的。

在单任务系统看门狗技术的基础上，通过添加监控任务模块以及各模块之间的关系组成多任务嵌入式系统的异常监控装置。本实施例对硬件支撑的要求是能实现基本的看门狗计数器功能，即系统上电后软件必须在规定时间

内周期性清空看门狗计数器，如果任何两次清空看门狗计数器的时间间隔超出了规定时间，看门狗计数器就会产生溢出，看门狗计数器溢出后会使该系统复位。这一点与单任务系统的看门狗对硬件支撑的要求一致。

在系统启动过程中清空看门狗计数器。由于系统启动过程中系统还处于单任务运行状态，该情况下系统执行的功能于与现有的单任务嵌入式系统看门狗技术相同，在正常运行出循环体中，捆绑清空看门狗计数器的操作，使得在正常运行时，看门狗计数器不会溢出；而出现异常时看门狗计数器不能及时清空，终将溢出，导致硬件复位，使系统恢复正常运行。

图 1 示出了根据本发明的一个较佳实施例的多任务系统的异常监控装置的结构及其组成。整个多任务系统的异常监控装置包含以下部分：监控任务模块 101、应用任务模块 102、看门狗计数器 104、任务运行状态报告计数器 103。

监控任务模块 101 是一个周期性执行的任务，该任务的功能就是清空看门狗计数器 104 和监控系统其它应用任务模块 102 的运行情况。清空看门狗计数器 104 的功能设在该任务循环体中，与监控其它应用任务模块运行状况的功能捆绑在一起。为保证监控任务模块 101 能够在规定时间间隔内，周期清空看门狗计数器 104，该任务优先级设为低于操作系统任务优先级，而高于任何应用任务模块 102 优先级。

看门狗计数器 104 的功能是自动计数，并在监控任务模块 101 执行清狗操作 114 的时候置回初值重新计数，如果因为没有被及时清空而计数溢出时，自动执行硬件复位操作 116。

应用任务模块 102 是普通的在系统中可以并行处理的任务，该任务在处理系统分配任务的同时在自身主循环体中执行清空相应的任务运行状态报告计数器 103 的操作 112。也就是说清空任务运行状态报告计数器 103 的操作 112 设在该任务的循环体中，与正常运行的操作捆绑在一起。

任务运行状态报告计数器103由系统在创建应用任务模块102的同时被相应地创建，一般来说任务运行状态报告计数器103和应用任务模块102是一一对应的，有多少应用任务模块102就有多少任务运行状态报告计数器103。它功能是接收监控任务模块101的计数读数操作113，并在应用任务模块102执清空操作112的时候置回初值重新计数。

多任务的异常监控装置的基本思想是，在系统创建每个应用任务模块102的同时，向监控任务模块101进行任务注册；应用任务模块102开始运行后，定期通过清空操作112向监控任务模块101报告运行状态；如果超过预定时间后应用任务模块102未主动向监控任务模块101报告运行状态，监控任务模块101通过操作111核查该任务，并根据核查结果做相应的处理，比如软件复位115等。

所述应用任务模块102和相应的任务状态报告计数器103应该有多个，与其相关的操作也同时增加，图1为了简洁并未全部示出。

下面详细描述在本发明的一个较佳实施例中，各部分功能的实现方法。

应用任务模块的注册：在设计监控任务模块时，需要同时提供应用任务模块登记注册函数。该函数的功能是为请求注册的应用任务模块分配一个任务运行状态报告计数器，同时记录应用任务模块注册时提供的唤醒函数地址。唤醒函数用于将应用任务模块从等待状态转变为就绪状态，以防止正常等待状态的任务因为长时间没有执行清空相应任务运行状态报告计数器的操作而被监控任务模块误认为发生了异常。唤醒函数在应用任务模块设计时提供，在监控任务模块请求核查该应用任务模块时被监控任务模块调用。

监控任务模块的设计：下面参照图2，详细描述监控任务模块各个步骤及其功能。

当系统启动并运行监控任务模块后，进入循环体的第一个步骤201，监控任务模块清空看门狗计数器。当监控任务模块出现异常而不能清空看门狗

计数器时，看门狗计数器将溢出，系统自动执行硬件复位操作，实现了监控任务模块本身的异常监控。

接着进入步骤 202，监控任务模块释放一定长度的 CPU 处理时间。使得其他优先级较低的应用任务模块得以运行。

接着进入步骤 203，监控任务模块对下一应用任务模块的任务运行状态报告计数器进行计数操作。

接着进入步骤 204，判断该任务运行状态报告计数器是否超过门限 1，如果是，说明该应用任务模块未在规定的时间内清空计数器操作，则进入步骤 205；否则说明该任务运行正常，进入步骤 211。在本发明的一个较佳实施例中，门限 1 定义为应用任务模块正常运行的周期门限。

在步骤 205 中，判断该任务运行状态报告计数器是否超过门限 2，如果是，说明该应用任务模块未被唤醒，则进入步骤 206；否则进入 210。在本发明的一个较佳实施例中，门限 2 定义为应用任务模块唤醒时间 + 门限 1。

在步骤 206 中，判断该任务运行状态报告计数器是否超过门限 3，如果是，说明该对该应用任务模块恢复操作不起作用，进入步骤 207；否则进入步骤 208。在本发明的一个较佳实施例中，门限 3 定义为应用任务模块恢复时间 + 门限 2。

在步骤 207 中，监控任务模块保存现场并进行软件复位，终止本流程。监控任务模块在发现某应用任务模块出现不能处理的异常而无法正常运行时，执行软件复位操作，强制复位系统。

在步骤 208 中，监控任务模块对该任务进行任务恢复，然后进入步骤 209。监控任务模块发现某应用任务模块无法唤醒时，执行任务恢复操作，即删除该应用任务模块后重新启动该应用任务模块。

在步骤 209 中，判断任务恢复是否成功，如果是，进入 211；否则，说明已无法通过任务恢复解决该应用任务模块的异常，进入步骤 207。

在步骤 210 中，监控任务模块调用该任务的任务唤醒函数，然后进入步骤 211。监控任务模块调用唤醒函数使得因为处在等待状态的时间较长而无法及时清空任务状态报告计数器的应用任务模块被唤醒，而得以运行并及时清空任务状态报告计数器。

在步骤 211 中，判断是否所有应用任务模块都已处理完毕，如果是，则返回步骤 201；否则返回步骤 203。监控任务模块保证每次循环对每个注册的应用任务模块进行一次核查，并根据核查结果做相应处理，核查完毕则重新执行循环体。

熟悉本领域的技术人员可以理解，所述监控任务模块的各个门限值、核查步骤及相应的处理方法可以根据具体系统的实际问题适当改变或增减，而不影响本发明的实质和范围。

应用任务模块设计：下面参照图 3，详细描述在本发明的一个较佳实施例中应用任务模块的各个步骤及其功能。

当系统创建好该任务，并向监控任务模块进行了任务注册之后，该应用任务模块便进入就绪状态，等待系统运行。首先进入步骤 301，应用任务模块进行初始化。

接着进入步骤 302，应用任务模块清空相应的任务运行状态报告计数器。这是应用任务模块主循环体的第一个操作，保证应用任务模块在正常运行的情况下能及时清空相应的任务运行状态报告计数器，这一操作即向监控任务模块报告本应用任务模块运行在正常状态。

接着进入步骤 303，判断系统是否有分配任务，如果是，则进入步骤 304；否则说明该应用任务模块无需处理，则返回步骤 303 继续等待。

在步骤 304 中应用任务模块处理系统分配的任务，返回步骤 302。应用任务模块处理完一次系统分配的任务后，重新执行主循环体。

熟知本领域的技术人员可以理解，在满足将正常运行的功能和清空任务

状态报告计数器的操作捆绑在主循环体中的要求下，所述应用任务模块的设计可以根据实际需要做相应的变化，而不影响本发明的实质和范围。

任务唤醒函数设计：任务唤醒函数在设计各应用任务模块时同时提供，该函数执行的操作是为处于等待状态的被监控应用任务模块分配一个虚拟任务，使本应用任务模块由等待状态变为就绪状态。例如某应用任务模块等待一条消息进行处理，唤醒函数可以发送一条特殊消息给该任务；某应用任务模块等待一个信号量，唤醒函数可以释放一次该信号量；某应用任务模块是一个周期执行的查询任务，唤醒函数则不需要做任何处理。

所述任务唤醒操作以及任务唤醒函数的设计，主要是针对多任务系统中可能出现的，由于某应用任务模块处于等待状态的时间过长而无法清空相应的任务状态报告计数器，不能及时向监控任务模块包括运行状态，而实际上却是处于正常状态的情况，可以让监控任务模块通过任务唤醒操作进行判断，而避免误认为是应用任务模块出现异常。

熟悉本领域的技术人员可以理解，所述监控任务模块监控应用任务模块运行情况的方法也可以是监控任务模块定期向应用任务模块发送核查消息，应用任务模块收到核查消息后给出响应，同样能达到监控应用任务模块的目的，而不影响本发明的实质和范围。

最后给出本发明的一个较佳实施例中，用类似于 C 语言的伪代码编写的各个任务、函数、及其数据结构。熟知本领域的技术人员可以理解，以下伪代码中各个部分的功能可以根据以上所述方案而做适当变化或增减并具体实现，而不影响本发明的实质和范围。

```
/**********************************************************
*每个被监控的应用任务模块对应的数据结构，用于记录任务运行报告
*计数器和任务唤醒函数。
*/
```

```
struct {
unsigned int timer; /*任务运行报告计数器*/
FUCPTR pwakeuptask;    /*记录任务唤醒函数指针*/
}
/*************************************************************
*任务注册函数：记录任务唤醒函数，分配任务运行状态报告计数器
*/
unsigned int *TaskRegister(FUCPTR pWakeup)
{
  if（分配记录项成功）
   {
      记录任务唤醒函数为 pWakeUp 函数;
      任务运行状态报告计数器设为初始值;
      返回任务运行状态报告计数器地址;
  }
  else
   {
      返回空指针;
  }
}
/*************************************************************
*监控任务模块设置：监控所有已注册应用任务模块的运行情况
*/
```

```
void monitor(void)
{
  while(1)  /*任务循环体*/
  {
     执行一次清狗操作;
     释放 CPU 一定长度的执行时间;
     for（任务号 = 0 ~ MAX）
      {
                        对应任务运行状态报告计数器增加一次计数;
        if（对应任务运行状态报告计数器未超过门限 1）
         {
         }
        elseif（对应任务运行状态报告计数器尚未超过门限 2）
         {
           调用该任务唤醒函数
         }
        elseif（对应任务运行状态报告计数器尚未超过门限 3）
         {
           if（任务恢复失败）
            {
              进行现场记录;
              软件恢复;
```

```
            }
        }
        else
        {
            进行现场记录;
            软件恢复;
        }
    }
  } /*任务循环体*/
}
/*************************************************************
*应用任务模块设计：向监控任务模块主动报告任务运行情况
*/
void TaskX(unsigned int* 对应任务运行状态报告计数器地址)
{
  任务相关初始化;
  while（1）  /*任务循环体*/
  {
    清空任务运行状态报告计数器;
    等待系统分配任务;
    系统任务处理;
  ]
```

```
}
/*****************************************************************
*任务唤醒函数设计:为本任务分配虚拟任务，唤醒任务执行;
*/
static void wakeup(void)
{
  为本任务分配虚拟任务，唤醒任务执行;
}
```

虽然通过参照本发明的某些优选实施例，已经对本发明进行了图示和描述，但本领域的普通技术人员应该明白，可以在形式上和细节上对其作各种各样的改变，而不偏离所附权利要求书所限定的本发明的精神和范围。

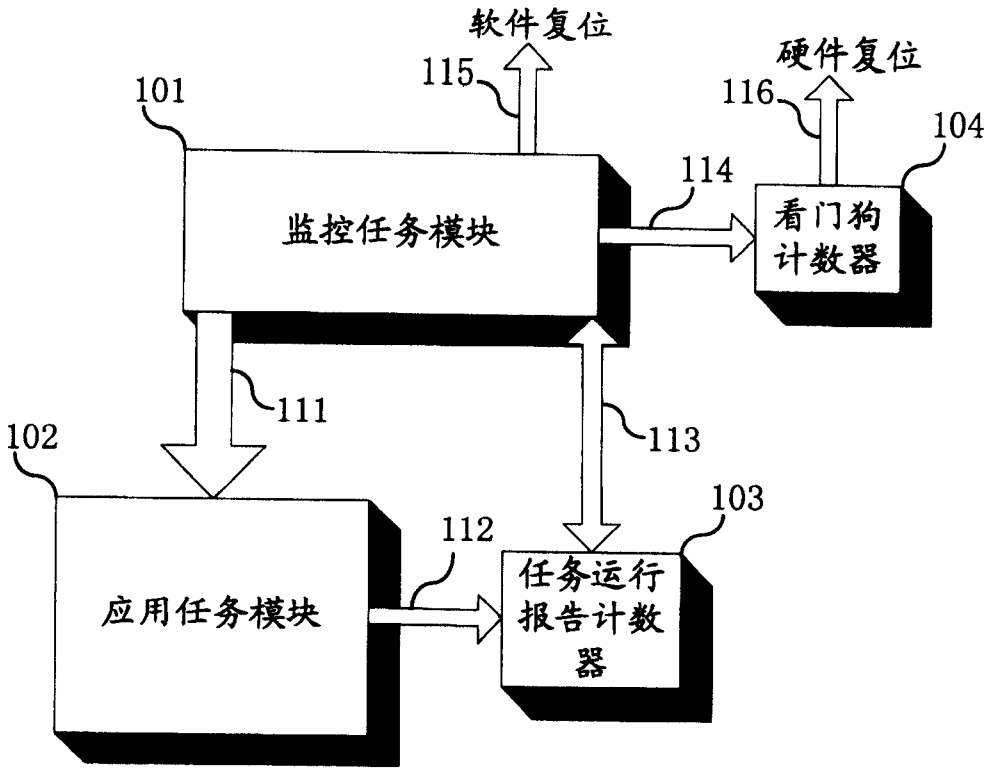

软件复位
硬件复位
115
116
101
104
114
监控任务模块
看门狗
计数器
111
113
102
103
112
应用任务模块
任务运行
报告计数
器

图 1

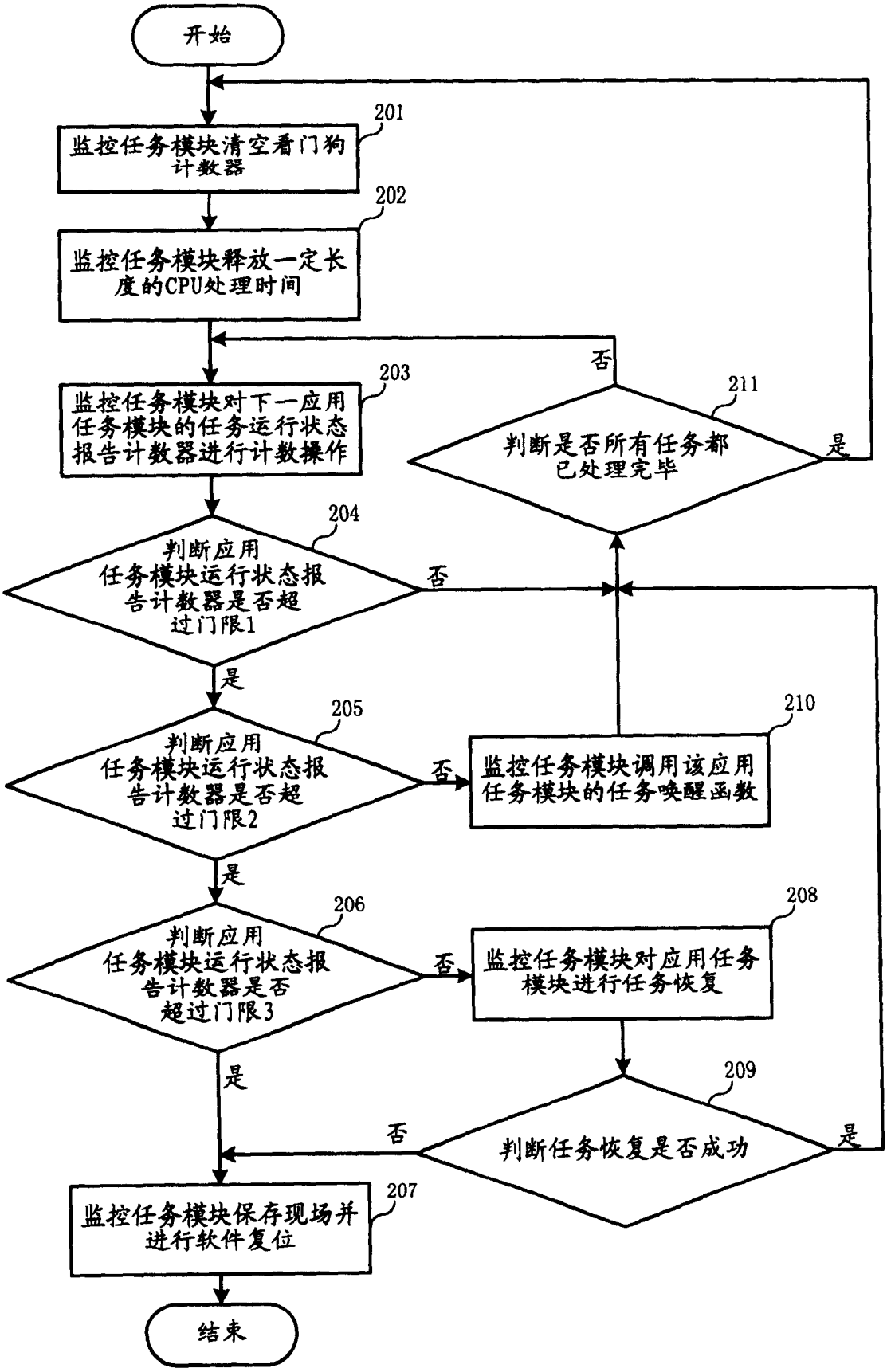

开始
201
监控任务模块清空看门狗计数器
202
监控任务模块释放一定长度的CPU处理时间
203
监控任务模块对下一应用任务模块的任务运行状态报告计数器进行计数操作
204
判断应用任务模块运行状态报告计数器是否超过门限1
否
是
205
判断应用任务模块运行状态报告计数器是否超过门限2
否
是
210
监控任务模块调用该应用任务模块的任务唤醒函数
206
判断应用任务模块运行状态报告计数器是否超过门限3
否
是
208
监控任务模块对应用任务模块进行任务恢复
209
判断任务恢复是否成功
否
是
211
判断是否所有任务都已处理完毕
否
是
207
监控任务模块保存现场并进行软件复位
结束

图 2

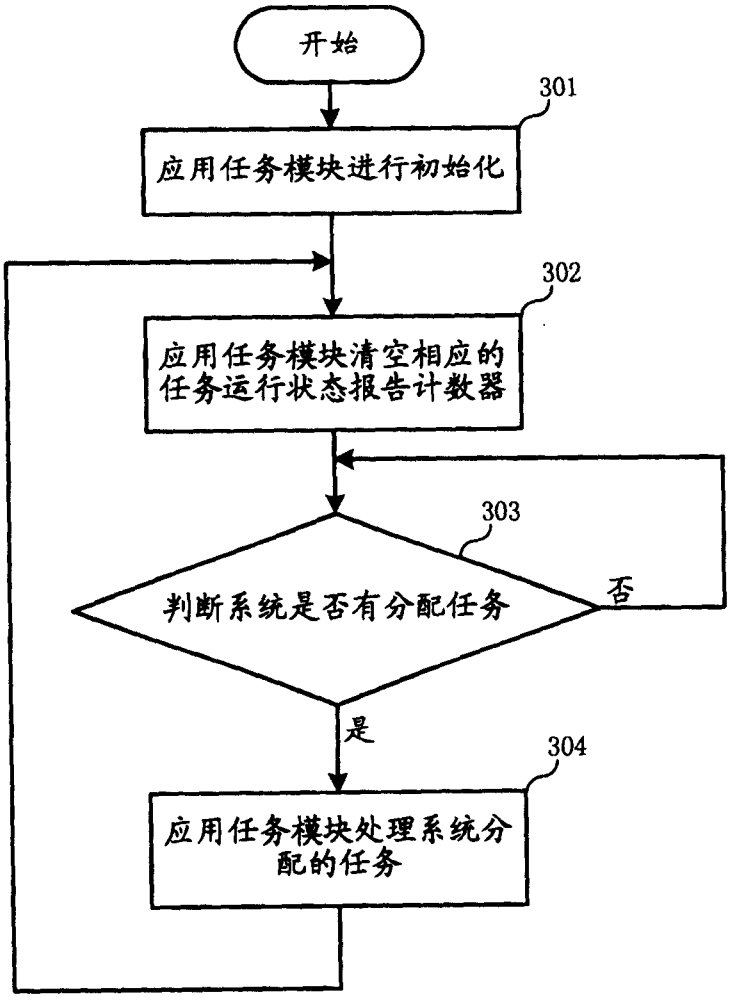

开始
301
应用任务模块进行初始化
302
应用任务模块清空相应的任务运行状态报告计数器
303
判断系统是否有分配任务
否
是
304
应用任务模块处理系统分配的任务

图 3