



(12) 发明专利

(10) 授权公告号 CN 102662847 B

(45) 授权公告日 2015. 01. 21

(21) 申请号 201210121035. 1

(22) 申请日 2012. 04. 23

(73) 专利权人 中颖电子股份有限公司
地址 200335 上海市长宁区金钟路 767 弄 3 号

(72) 发明人 龚静 张钦

(74) 专利代理机构 上海专利商标事务所有限公
司 31100

代理人 陈亮

(51) Int. Cl.

G06F 11/36(2006. 01)

(56) 对比文件

CN 102117243 A, 2011. 07. 06, 参见说明书第 5 页第 29-32 段, 第 6 页 36-38 段, 第 8 页第 77 段,

图 1, 图 4, 图 5, 图 6.

US 5664159 A, 1997. 09. 02, 参见说明书第 5-6 栏, 图 1, 图 5, 图 10.

US 2009/0031289 A1, 2009. 01. 29, 全文.
CN 102193860 A, 2011. 09. 21, 全文.

审查员 鲍旭恒

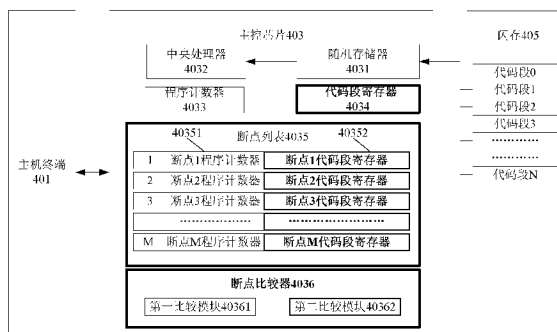
权利要求书2页 说明书8页 附图5页

(54) 发明名称

基于闪存应用的嵌入式系统的程序调试系统及方法

(57) 摘要

本发明提供一种基于闪存应用的嵌入式系统的程序调试系统及方法, 该系统包括: 主机终端, 其上具有程序调试工具; 主控芯片, 与主机终端相连接; 闪存, 与主控芯片相连接, 其上存储有待调试程序的多段代码, 分别位于不同段编号的代码段内; 主控芯片包括: 随机存储器, 暂存从闪存中加载进主控芯片的代码; 中央处理器, 与随机存储器相连接, 从随机存储器中取出代码执行; 程序计数器, 记录中央处理器从随机存储器中取代码的地址; 代码段寄存器, 记录当前从内存中加载到随机存储器中的代码所在的代码段的段编号。本发明能定位切分为多个代码段的程序执行的准确位置, 实现完整的源代码调试, 还能定位出断点的位置, 方便断点调试, 极大地提高程序开发效率。



1. 一种基于闪存应用的嵌入式系统的程序调试系统 (400), 包括:
 - 主机终端 (401), 其上具有程序调试工具;
 - 主控芯片 (403), 与所述主机终端 (401) 相连接; 以及
 - 闪存 (405), 与所述主控芯片 (403) 相连接, 其上存储有待调试程序的多段代码, 每段代码位于不同代码段内, 每个代码段具有不同的段编号;
 - 其中, 所述主控芯片 (403) 包括:
 - 随机存储器 (4031), 用于暂存从所述闪存 (405) 中加载进所述主控芯片 (403) 的代码;
 - 中央处理器 (4032), 与所述随机存储器 (4031) 相连接, 用于从所述随机存储器 (4031) 中取出代码执行;
 - 程序计数器 (4033), 用于记录所述中央处理器 (4032) 从所述随机存储器 (4031) 中取代码的地址; 以及
 - 代码段寄存器 (4034), 用于记录当前从所述闪存 (405) 中加载到所述随机存储器 (4031) 中的代码所在的代码段的段编号;
 - 所述主控芯片 (403) 还包括断点列表 (4035) 和断点比较器 (4036), 其中:
 - 所述断点列表 (4035) 用于记录所述程序调试系统 (400) 执行的所有断点信息, 其包括:
 - 断点代码段寄存器 (40352), 用于记录各个断点所在的代码段的段编号;
 - 断点程序计数器 (40351), 用于记录各个断点在其代码段内的段内偏移地址;
 - 所述断点比较器 (4036) 用于将所述断点列表 (4035) 中所有的断点信息依次取出, 每一所述断点信息包括一个断点所在的代码段的段编号以及所述断点在其代码段内的段内偏移地址, 分别与所述代码段寄存器 (4034) 以及所述程序计数器 (4033) 中的信息进行比较, 其包括:
 - 第二比较模块 (40362), 用于比较所述代码段寄存器 (4034) 与所述断点代码段寄存器 (40352) 记录的信息;
 - 第一比较模块 (40361), 用于比较所述程序计数器 (4033) 与所述断点程序计数器 (40351) 记录的信息。
2. 根据权利要求 1 所述的程序调试系统 (400), 其特征在于, 所述主机终端 (401) 为个人计算机。
3. 根据权利要求 1 所述的程序调试系统 (400), 其特征在于, 所述闪存 (405) 为 NAND 闪存。
4. 根据权利要求 3 所述的程序调试系统 (400), 其特征在于, 所述 NAND 闪存包括 SD 卡、U 盘和 MP3 播放器。
5. 一种采用权利要求 1 至 4 中任一项所述的基于闪存应用的嵌入式系统的程序调试系统 (400) 进行程序调试的方法, 包括程序定位步骤:
 - I. 从闪存 (405) 中加载一代码段的代码到随机存储器 (4031);
 - II. 代码段寄存器 (4034) 记录所述代码段所在的段编号;
 - III. 程序计数器 (4033) 记录所述中央处理器 (4032) 从所述随机存储器 (4031) 中取代码的地址;

IV. 中央处理器 (4032) 根据所述程序计数器 (4033) 记录的所述地址, 从所述随机存储器 (4031) 中取出所述代码并执行;

V. 在所述代码执行的过程中, 所述程序计数器 (4033) 更新为下一次取代码的地址;

VI. 在所述代码执行的过程中, 当从所述闪存 (405) 中加载另外段编号的代码段的代码到所述随机存储器 (4031) 时, 所述代码段寄存器更新为所述代码段所在的段编号; 以及

VII. 循环执行上述步骤 I ~ VI, 直至待调试程序的所有代码均执行调试完毕。

6. 根据权利要求 5 所述的程序调试的方法, 其特征在于, 还包括断点调试步骤:

A. 主机终端 (401) 在所述待调试程序的一个或多个位置设置断点, 将所述断点所在的代码段的段编号以及代码段内偏移地址通知主控芯片 (403);

B. 主控芯片 (403) 内的断点代码段寄存器 (40352) 记录所述断点所在的代码段的段编号, 主控芯片 (403) 内的断点程序计数器 (40351) 记录所述断点在其代码段内的段内偏移地址;

C. 在代码执行过程中, 断点比较器 (4036) 将断点列表 (4035) 中所有的断点信息依次取出, 每一所述断点信息包括一个断点所在的代码段的段编号以及所述断点在其代码段内的段内偏移地址, 分别与所述代码段寄存器 (4034) 以及所述程序计数器 (4033) 中的信息进行比较, 若所述断点列表 (4035) 中存在着某个断点, 其对应的断点代码段寄存器 (40352) 中的信息与所述代码段寄存器 (4034) 中的信息一致, 并且该断点对应的所述断点程序计数器 (40351) 中的信息与所述程序计数器 (4033) 中的信息一致, 则表示程序运行到了某个预先设定的断点位置, 则暂停程序执行, 否则表示程序未执行到断点位置, 程序持续执行。

基于闪存应用的嵌入式系统的程序调试系统及方法

技术领域

[0001] 本发明涉及嵌入式系统技术领域,具体来说,本发明涉及一种基于闪存应用的嵌入式系统的程序调试系统及方法。

背景技术

[0002] 对于嵌入式系统的设计,程序的大小、程序存储的方式是设计者非常关心的问题。

[0003] 当系统比较简单,程序代码比较小的时候,比如只有数十 K 字节甚至更小,通常选择内置程序存储器的主控芯片,程序代码直接存储在内置的存储器中,整个系统成本比较经济;而当系统比较复杂,程序代码比较大,比如达到数百 K 字节甚至以兆 (M) 为单位时,主控芯片已经无法内置这么大的存储器,此时整个系统则采用外挂存储器来存储程序代码的方式。

[0004] 后来,随着 Nand Flash(以下简称闪存)的发明,出现了一系列的应用比如 SD 卡, U 盘, MP3 播放器等。在这些应用中,程序代码量比较大,考虑到已经使用闪存来存储歌曲或其它用户资料,设计者可以将程序代码也存储在闪存中,这样系统就不需要外挂另外的程序存储器来存储程序。

[0005] 但是由于闪存是页模式读取数据,主控芯片无法对内存中的程序代码直接寻址访问,因此在系统的实际设计中,会在主控芯片内部设计一块 RAM(随机存储器)作为程序暂存空间。系统在工作时,先将程序从闪存中读入主控芯片内部的 RAM 中,再执行即可。考虑到主控芯片内部无法设计太大的 RAM,因此往往采用分段加载程序代码段 1 ~ N 的方式实现,如图 1 所示。

[0006] 上述系统架构的特点包括:

[0007] 1) 主控芯片内不用设计很大的内部存储器来存储程序代码。

[0008] 2) 程序代码被分割为很多个小的代码段(代码段 0,代码段 1,...),存放在外部的闪存(Nand Flash Memory)中。

[0009] 3) 主控芯片内部设计一个较小的 RAM,缓存程序。

[0010] 4) 分段加载程序到 RAM,CPU 从 RAM 中取指令执行;如图 2 所示为上述系统架构的一个示例性的程序执行流程图。

[0011] 不可否认,该系统架构有非常明显的优势:

[0012] 1) 程序代码存储在闪存中,在主控芯片中不用设计很大的内部 RAM 来存储程序代码,因此整个主控芯片成本可以控制得较小。

[0013] 2) 程序代码分段存储,分段加载到 RAM 中执行,这种分段加载、分段执行的方法,可以使 RAM 设计得很小,因为某时刻,只需要加载某一小段程序代码进 RAM 执行,而不是加载所有的程序。

[0014] 例如,程序代码总长度为 200K 字节,可以将程序代码切分为 25 小段,每段长度 8K 字节。

[0015] 1) 200K 字节程序代码存储在闪存中,不用在主控芯片内部设计一个 200K 字节大

规模存储器。

[0016] 2) 芯片内部只用设计 8K 字节的 RAM,任何时刻只用将某一小段程序加载进 RAM 来执行,这样 RAM 可以设计得很小。

[0017] 虽然这种程序代码存储在外部闪存中、程序分段加载到 RAM 执行的系统设计具有很好的成本优势,但是也存在一些问题。例如,因为程序不是一次性地加载到 RAM 中执行,而是分段加载,所以在调试程序时,如何定位程序执行的位置,以及如何定位断点的位置都会很麻烦,无法实现完整的源代码调试,导致程序调试极其不便。

发明内容

[0018] 本发明所要解决的技术问题是提供一种基于闪存应用的嵌入式系统的程序调试的系统及方法,能够定位出程序执行的准确位置和断点的准确位置,实现完整的源代码调试,方便程序调试,极大地提高程序开发的效率。

[0019] 为解决上述技术问题,本发明提供一种基于闪存应用的嵌入式系统的程序调试系统,包括:

[0020] 主机终端,其上具有程序调试工具;

[0021] 主控芯片,与所述主机终端相连接;以及

[0022] 闪存,与所述主控芯片相连接,其上存储有待调试程序的多段代码,分别位于不同段编号的代码段内;

[0023] 其中,所述主控芯片包括:

[0024] 随机存储器,用于暂存从所述闪存中加载进所述主控芯片的代码;

[0025] 中央处理器,与所述随机存储器相连接,用于从所述随机存储器中取出代码执行;

[0026] 程序计数器,用于记录所述中央处理器从所述随机存储器中取代码的地址;以及

[0027] 代码段寄存器,用于记录当前从所述闪存中加载到所述随机存储器中的代码所在的代码段的段编号。

[0028] 可选地,所述主控芯片还包括断点列表和断点比较器,其中:

[0029] 断点列表用于记录所述程序调试系统执行的所有断点信息,其包括:

[0030] 断点代码段寄存器,用于记录各个断点所在的代码段的段编号;

[0031] 断点程序计数器,用于记录各个断点在其代码段内的段内偏移地址;以及

[0032] 断点比较器用于将所述断点列表中所有的断点信息依次取出,分别与所述代码段寄存器以及所述程序计数器中的信息进行比较,其包括:

[0033] 第二比较模块,用于比较所述代码段寄存器与所述断点代码段寄存器记录的信息;

[0034] 第一比较模块,用于比较所述程序计数器与所述断点程序计数器记录的信息。

[0035] 可选地,所述主机终端为个人计算机。

[0036] 可选地,所述闪存为 NAND 闪存。

[0037] 可选地,所述 NAND 闪存包括 SD 卡、U 盘和 MP3 播放器。

[0038] 为解决上述技术问题,本发明还提供一种采用上述任一项所述的基于闪存应用的嵌入式系统的程序调试系统进行程序调试的方法,包括程序定位步骤:

- [0039] I. 从闪存中加载一代码段的代码到随机存储器；
- [0040] II. 代码段寄存器记录所述代码段所在的段编号；
- [0041] III. 程序计数器记录所述中央处理器从所述随机存储器中取代码的地址；
- [0042] V. 中央处理器根据所述程序计数器记录的所述地址，从所述随机存储器中取出所述代码并执行；
- [0043] IV. 在所述代码执行的过程中，所述程序计数器更新为下一次取代码的地址；
- [0044] VI. 在所述代码执行的过程中，当从所述闪存中加载另外段编号的代码段的代码到所述随机存储器时，所述代码段寄存器更新为所述代码段所在的段编号；以及
- [0045] VII. 循环执行上述步骤 I ~ VI，直至待调试程序的所有代码均执行调试完毕。
- [0046] 可选地，所述程序调试的方法还包括断点调试步骤：
- [0047] A. 主机终端在所述待调试程序的一个或多个位置设置断点，将所述断点所在的代码段的段编号以及代码段内偏移地址通知主控芯片；
- [0048] B. 主控芯片内的断点代码段寄存器和断点程序计数器分别记录所述断点所在的代码段的段编号和段内偏移地址；
- [0049] C. 在代码执行过程中，断点比较器将断点列表中所有的断点信息依次取出，分别与所述代码段寄存器以及所述程序计数器中的信息进行比较，若所述断点列表中存在着某个断点，其对应的断点代码段寄存器中的信息与所述代码段寄存器中的信息一致，并且该断点对应的所述断点程序计数器中的信息与所述程序计数器中的信息一致，则表示程序运行到了某个断点的位置，则暂停程序执行，否则表示程序未执行到断点位置，程序持续执行。
- [0050] 与现有技术相比，本发明具有以下优点：
- [0051] 本发明提供的设计方案，解决了基于闪存应用的嵌入式系统代码分段加载带来的程序调试不方便的问题，配合开发相应的调试工具，能够定位出断点的准确位置，可实现完整的源代码调试。另外，还可以实现断点调试，极大地方便了开发人员，提高了程序开发效率。

附图说明

- [0052] 本发明的上述的以及其他的特征、性质和优势将通过下面结合附图和实施例的描述而变得更加明显，其中：
- [0053] 图 1 为现有技术中一种嵌入式系统架构的示意图；
- [0054] 图 2 为图 1 中所示的系统架构的一个示例性的程序执行流程图；
- [0055] 图 3 为现有技术中一种基于闪存应用的嵌入式系统的程序调试系统的模块示意图；
- [0056] 图 4 为本发明一个实施例的基于闪存应用的嵌入式系统的程序调试系统的模块示意图；
- [0057] 图 5 为现有技术中另一种基于闪存应用的嵌入式系统的程序调试系统的模块示意图；
- [0058] 图 6 为本发明另一个实施例的基于闪存应用的嵌入式系统的程序调试系统的模块示意图；

[0059] 图 7 为本发明一个实施例的基于闪存应用的嵌入式系统的程序调试系统的断点列表示意图；

[0060] 图 8 为本发明一个实施例的基于闪存应用的嵌入式系统的程序调试方法的流程图示意图；

[0061] 图 9 为在图 8 基础上本发明另一个实施例的基于闪存应用的嵌入式系统的程序调试方法的流程图示意图。

具体实施方式

[0062] 下面结合具体实施例和附图对本发明作进一步说明，在以下的描述中阐述了更多的细节以便于充分理解本发明，但是本发明显然能够以多种不同于此描述的其它方式来实施，本领域技术人员可以在不违背本发明内涵的情况下根据实际应用情况作类似推广、演绎，因此不应以此具体实施例的内容限制本发明的保护范围。

[0063] I. 解决程序定位问题

[0064] 程序定位：在程序调试过程中，安装在主机终端（也叫 Host 终端，例如个人计算机）上的调试工具需要知道主控芯片中程序执行的确切位置，并将对应位置的程序的源程序代码显示在主机终端，供程序员调试。显然，如果无法将程序执行所在的确切位置，与源代码对应的位置匹配起来，则会对程序的调试带来非常大的困扰。

[0065] 在未有本发明之前的一种基于闪存应用的原嵌入式系统的程序调试系统的模块示意图可以如图 3 所示。该程序调试系统的主要模块包括主机终端、主控芯片和闪存。其中，主机终端上具有程序调试工具；主控芯片与主机终端相连接；闪存与主控芯片相连接，其上存储有待调试的程序，该程序已被切分为多个代码段 0 ~ N。

[0066] 在图 3 所示的例子中，主控芯片包括：随机存储器（RAM）、中央处理器（CPU）和程序计数器（Program Counter，简称 PC）。随机存储器用于加载并暂存闪存中的某一段代码。中央处理器与随机存储器相连接，用于执行随机存储器中的代码。程序计数器（PC）用于记录中央处理器将要执行的下一条指令代码在随机存储器中的地址。CPU 从 RAM 中 PC 记录的地址中取指令代码执行，然后 PC 更新，记录下一条指令代码的地址。主机终端可以通过 PC 知道 RAM 中程序执行的位置。

[0067] 但是，该现有的程序调试系统存在的问题包括：主机终端可以知道 RAM 中程序执行的位置，却不知道 RAM 中程序是哪段代码的程序，即代码段 / 程序段的编号无法知道。

[0068] 例如：程序在执行过程中，先加载代码段 0 的代码执行，再加载代码段 3 的代码执行，RAM 的内容先是代码段 0 的程序，后是代码段 3 的程序。但是 RAM 内容从代码段 0 更新为代码段 3 的过程，对于主机终端是不可知的。主机终端不能区分出哪段程序在 RAM 中执行，所以无法知道整个程序执行的准确位置，也就不能显示出对应的程序代码，或者显示其他的代码，这显然会对程序员的调试工作带来极大不便。

[0069] 本发明针对上述问题提出了解决方案，图 4 为本发明一个实施例的基于闪存应用的嵌入式系统的程序调试系统的模块示意图。如图 4 所示，该程序调试系统 400 可以包括主机终端 401、主控芯片 403 和闪存 405。其中，主机终端 401 可以为个人计算机，其上具有程序调试工具（未图示）；主控芯片 403 与主机终端 401 相连接；闪存 405 可以为 NAND 闪存（例如 SD 卡、U 盘或者 MP3 播放器），与主控芯片 403 相连接，其上存储有待调试程序的

多个代码段 0 ~ N。

[0070] 在本实施例中,主控芯片 403 可以包括随机存储器 (RAM) 4031、中央处理器 (CPU) 4032、程序计数器 (Program Counter, 简称 PC) 4033 和一个新增的代码段寄存器 (Code Segment) 4034。其中,随机存储器 4031,用于暂存从闪存 405 中加载进主控芯片 403 的代码。中央处理器 4032 与随机存储器 4031 相连接,用于从随机存储器 4031 中取出代码来执行。程序计数器 4033 用于记录中央处理器 4032 从随机存储器 4031 中取代码的地址。新增的代码段寄存器 4034 用于记录当前从内存 405 中加载到随机存储器 4031 中的代码所在的代码段的段编号 0 ~ N。

[0071] 例如,当代码段 0 被加载到 RAM 4031 中,代码段寄存器 4034 就被设置为 0,代码段 0 程序执行中;当加载了代码段 3 程序到 RAM 4031 时,代码段寄存器 4034 就被设置为 3,以此类推。

[0072] 于是,主机终端 401 通过读代码段寄存器 4034,就可以知道在 RAM 4031 中正执行的程序是从哪一个程序代码段 0 ~ N 加载的。另外,主机终端 401 通过读程序计数器 4033,就可以知道 RAM 4031 中的程序执行到了哪个位置。

[0073] 结合这两点,主机终端 401 就可以定位程序执行的确切位置,并显示出当前位置对应的程序代码段。

[0074] II. 解决断点调试问题

[0075] 断点调试:通过主机终端设置断点位置,当程序执行到断点位置时,暂停执行,方便调试人员调试。

[0076] 图 5 为现有技术中另一种基于闪存应用的嵌入式系统的程序调试系统的模块示意图。如图 5 所示,其与图 3 中所示的现有例子的区别在于:该主控芯片还包括断点列表和断点比较器。其中,该断点列表用于记录程序调试时的断点信息。断点列表仅仅包括了断点程序计数器 (Breakpoint PC),只能记录下断点在代码段内部的偏移地址。该断点比较器用于对程序计数器 (PC) 与断点列表中的所有断点程序计数器 (Breakpoint PC) 记录的信息进行比较。

[0077] 适用上述程序调试系统进行程序调试的思路是:

[0078] 1) 主机终端在某程序代码段内部某偏移位置,设置断点;

[0079] 2) 主控芯片断点列表的断点程序计数器记录下该偏移位置;

[0080] 3) 主机终端通知主控芯片执行程序,在程序执行过程中程序计数器实时更新;

[0081] 4) 主控芯片的断点比较器在程序执行过程中对程序计数器和断点列表中的断点计数器进行实时比较,如果程序计数器与断点程序计数器相等,就认为程序已经执行到断点位置,然后暂停。

[0082] 但是,这种断点调试系统存在的问题是:断点信息不完整,因为主控芯片断点列表中不包含程序代码的段信息。因此,尽管之前解决了源代码与程序具体执行位置的匹配,但还存在断点不匹配的问题。

[0083] 例如,程序员通过主机终端在代码段 0 程序段,偏移 0x100 处设置了断点,希望当代码段 0 段程序加载到 RAM 中,且执行到地址为 0x100 的位置时暂停。但是,因为主控芯片的断点列表中只记录下了 0x100 这个信息,没有代码段 0 这个程序段信息,结果是在调试过程中,其他程序段(代码段 1,代码段 2,....)的程序在 RAM 中执行时,执行到 0x100 的地

址时都会暂停。这与程序员最初的设计思想不符。

[0084] 针对这个问题,本发明也进行了以下改进。如图 6 所示,其为本发明另一个实施例的基于闪存应用的嵌入式系统的程序调试系统的模块示意图。在图 6 中,主控芯片 403 在图 4 所示的实施例的基础上,进一步包括断点列表 4035 和断点比较器 4036。其中,断点列表 4035 用于记录程序调试系统 400 执行的所有断点信息,其包括 M 个断点程序计数器 40351 和相应的断点代码段寄存器 40352。断点程序计数器 40351 用于记录各个断点在其代码段 0 ~ N 内的段内偏移地址。断点代码段寄存器 40352 用于记录各个断点所在的代码段的段编号。断点比较器 4036 用于将断点列表 4035 中所有的断点信息依次取出,分别与代码段寄存器 4034 以及程序计数器 4033 中的信息进行比较,其包括第一比较模块 40361 和第二比较模块 40362。第一比较模块 40361 用于比较程序计数器 4033 与断点程序计数器 40351 记录的信息。第二比较模块 40362 用于比较代码段寄存器 4034 与断点代码段寄存器 40352 记录的信息。

[0085] 由此,程序员在进行断点调试时,系统工作过程可以包括:

[0086] 1) 主机终端 401 在程序的某位置设置断点,将该断点所在的程序段,以及段内偏移地址通知主控芯片 403;

[0087] 2) 主控芯片 403 中的断点代码段寄存器 40352 记录下断点的所在的代码段编号;

[0088] 3) 主控芯片 403 中的断点程序计数器 40351 记录下断点在其所在的代码段内偏移地址;

[0089] 4) 主控芯片 403 在程序执行过程中,程序计数器 4033 及代码段寄存器 4034 会实时改变,指示当前程序执行的位置;

[0090] 5) 程序执行过程中,断点比较器 4036 实时进行断点比较动作,将断点列表 4035 中所有的断点信息依次取出,分别对程序计数器 4033 与断点列表 4035 中的所有断点程序计数器 40351 记录的信息进行比较,并对代码段寄存器 4034 与断点列表 4035 中的所有断点代码段寄存器 40352 进行比较。若断点列表 4035 中存在着某个断点,其对应的断点代码段寄存器 40352 中的信息与代码段寄存器 4034 中的信息一致,并且该断点对应的断点程序计数器 40351 中的信息与所述程序计数器 4033 中的信息一致,则表示程序运行到了某个断点的位置,则暂停程序执行,否则表示程序未执行到断点位置,程序持续执行。

[0091] 例如:断点调试时,设置 2 个断点:

[0092] 代码段 0,段内偏移 0x100 地址

[0093] 代码段 3,段内偏移 0x200 地址

[0094] 图 7 为本发明一个实施例的基于闪存应用的嵌入式系统的程序调试系统的断点列表示意图,则具体过程如下:

[0095] 1) 主机终端 401 在代码段 0 程序段,段内偏移 0x100 设置断点;主机终端 401 在代码段 3 程序段,段内偏移 0x200 设置断点,并将该信息通知主控芯片 403;

[0096] 2) 主控芯片 403 记录下断点的程序代码段号(代码段 0,代码段 3);

[0097] 3) 主控芯片 403 记录下断点的程序代码段内部偏移地址(0x100,0x200),如图 7 所示;

[0098] 4) 程序执行过程中,程序计数器 4033 及代码段寄存器 4034 会实时改变;

[0099] 5) 主控芯片 403 的断点比较器 4036 进行断点比较动作:

[0100] 当程序计数器 4033 等于 0x100,且代码段寄存器 4034 等于 0,认为执行到了断点处,暂停执行;

[0101] 或者

[0102] 当程序计数器 4033 等于 0x200,且代码段寄存器 4034 等于 3,认为执行到了断点处,暂停执行。

[0103] 显然,采用这种方式修改断点系统之后,可以解决之前程序调试系统存在的断点调试问题。

[0104] 图 8 为本发明一个实施例的基于闪存应用的嵌入式系统的程序调试方法的流程示意图。如图所示,该程序调试方法可以包括程序定位步骤:

[0105] 执行步骤 S801,从闪存 405 中加载一段代码的代码到随机存储器 4031;

[0106] 执行步骤 S802,代码段寄存器 4034 记录代码段所在的段编号;

[0107] 执行步骤 S803,程序计数器 4033 记录中央处理器 4032 从随机存储器 4031 中取代码的地址;

[0108] 执行步骤 S804,中央处理器 4032 根据程序计数器 4033 记录的地址,从随机存储器 4031 中取出代码并执行;

[0109] 执行步骤 S805,在代码执行的过程中,程序计数器 4033 更新为下一次取代码的地址;

[0110] 执行步骤 S806,在代码执行的过程中,当从闪存 405 中加载另外段编号的代码段的代码到随机存储器 4031 时,代码段寄存器更新为代码段所在的段编号;以及

[0111] 执行步骤 S807,循环执行上述步骤 S801 ~ S806,直至待调试程序的所有代码均执行调试完毕。

[0112] 图 9 为在图 8 基础上本发明另一个实施例的基于闪存应用的嵌入式系统的程序调试方法的流程示意图。如图所示,该程序调试方法还可以包括断点调试步骤:

[0113] 执行步骤 S901,主机终端 401 在待调试程序的一个或多个位置设置断点,将断点所在的代码段的段编号以及代码段内偏移地址通知主控芯片 403;

[0114] 执行步骤 S902,主控芯片 403 内的断点代码段寄存器 40352 和断点程序计数器 40351 分别记录断点所在的代码段的段编号和段内偏移地址;

[0115] 执行步骤 S903,在代码执行过程中,断点比较器 4036 将断点列表 4035 中所有的断点信息依次取出,分别与代码段寄存器 4034 以及程序计数器 4033 中的信息进行比较,若断点列表 4035 中存在着某个断点,其对应的断点代码段寄存器 40352 中的信息与代码段寄存器 4034 中的信息一致,并且该断点对应的断点程序计数器 40351 中的信息与程序计数器 4033 中的信息一致,则表示程序运行到了某个断点的位置,则暂停程序执行,否则表示程序未执行到断点位置,程序持续执行。

[0116] 本发明提供的设计方案,解决了基于闪存应用的嵌入式系统代码分段加载带来的程序调试不方便的问题,配合开发相应的调试工具,能够定位出断点的准确位置,可实现完整的源代码调试。另外,还可以实现断点调试,极大地方便了开发人员,提高了程序开发效率。

[0117] 本发明虽然以较佳实施例公开如上,但其并不是用来限定本发明,任何本领域技术人员在不脱离本发明的精神和范围内,都可以做出可能的变动和修改。因此,凡是未脱离

本发明技术方案的内容,依据本发明的技术实质对以上实施例所作的任何修改、等同变化及修饰,均落入本发明权利要求所界定的保护范围之内。

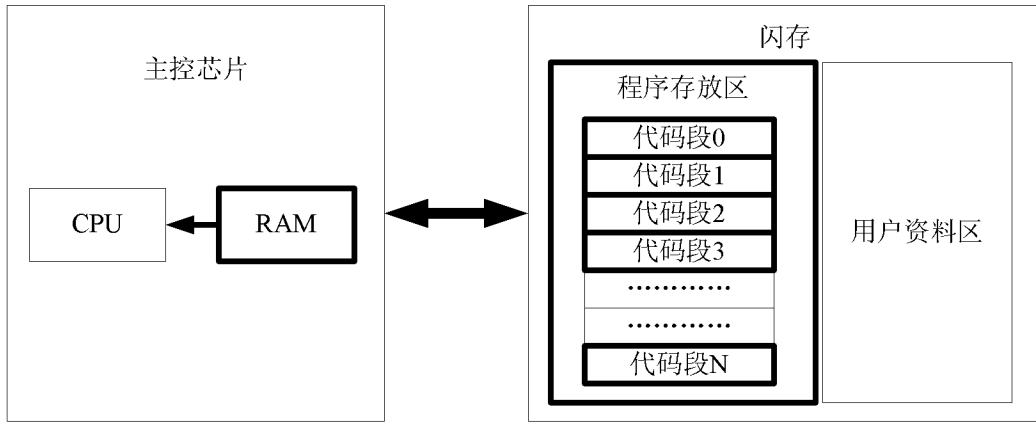


图 1

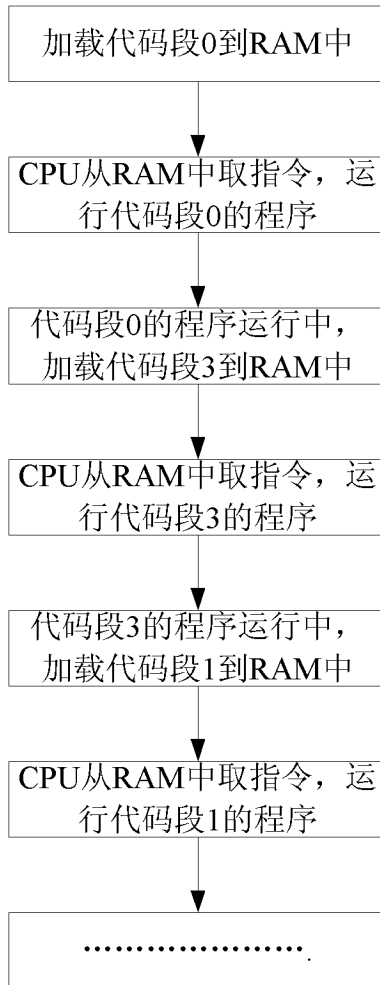


图 2

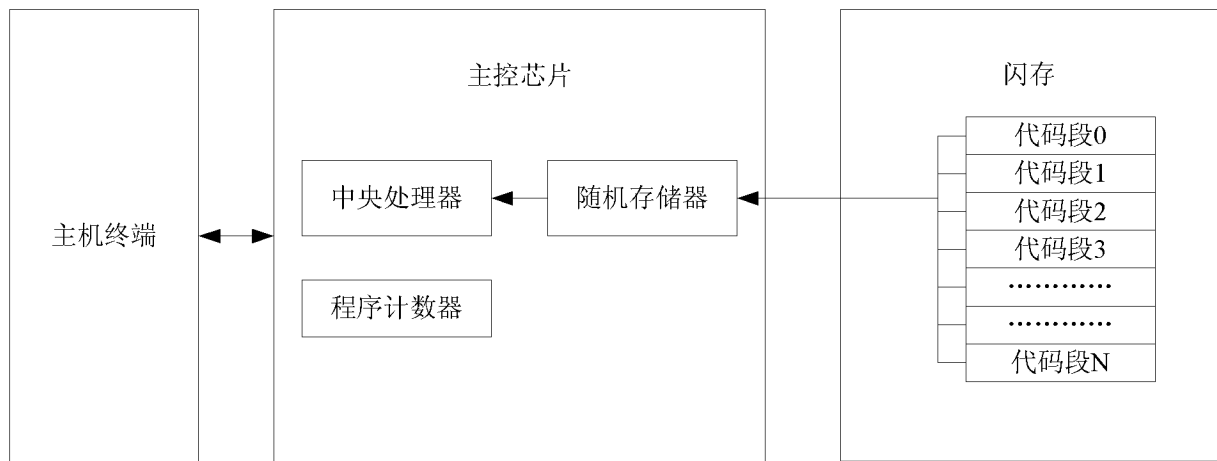


图 3

400

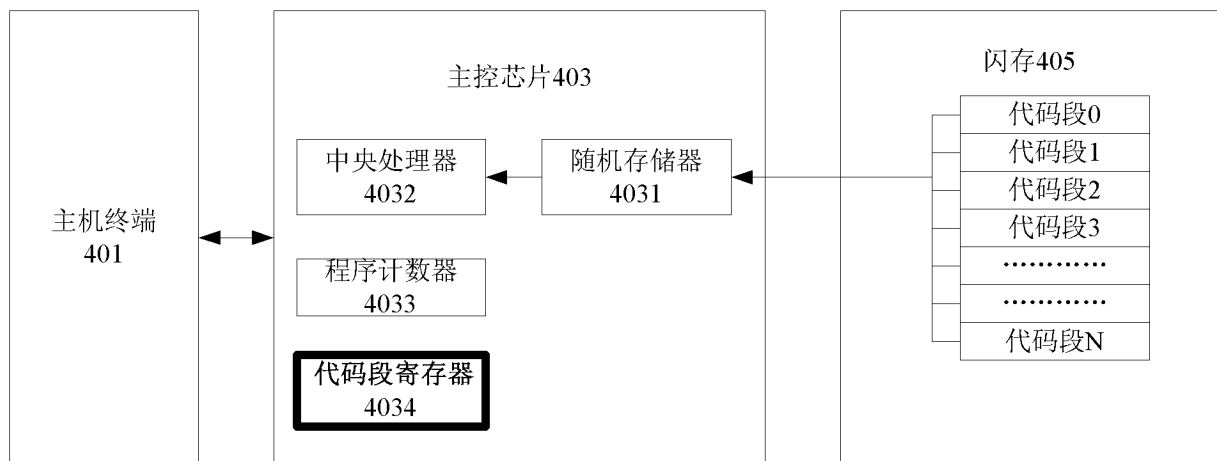


图 4

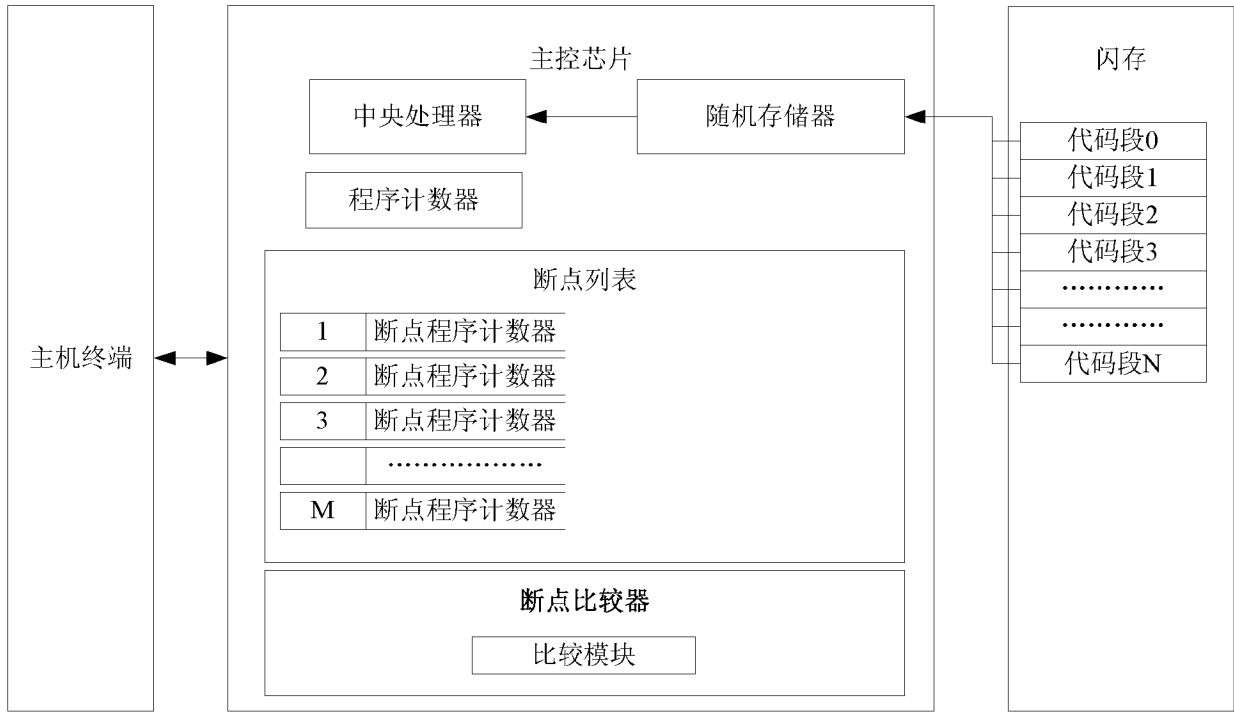


图 5

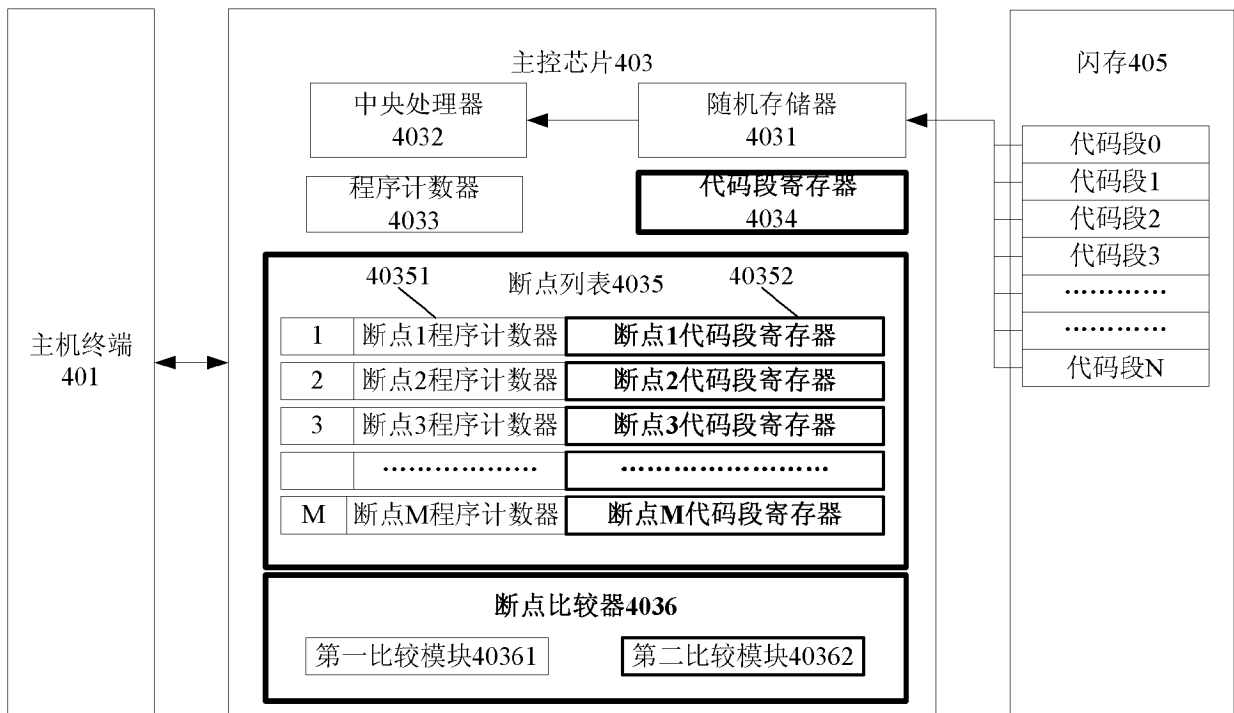


图 6

断点列表4035		
1	0x100	0
2	0x200	3
3	无效值	无效值
	无效值	无效值
M	无效值	无效值

图 7

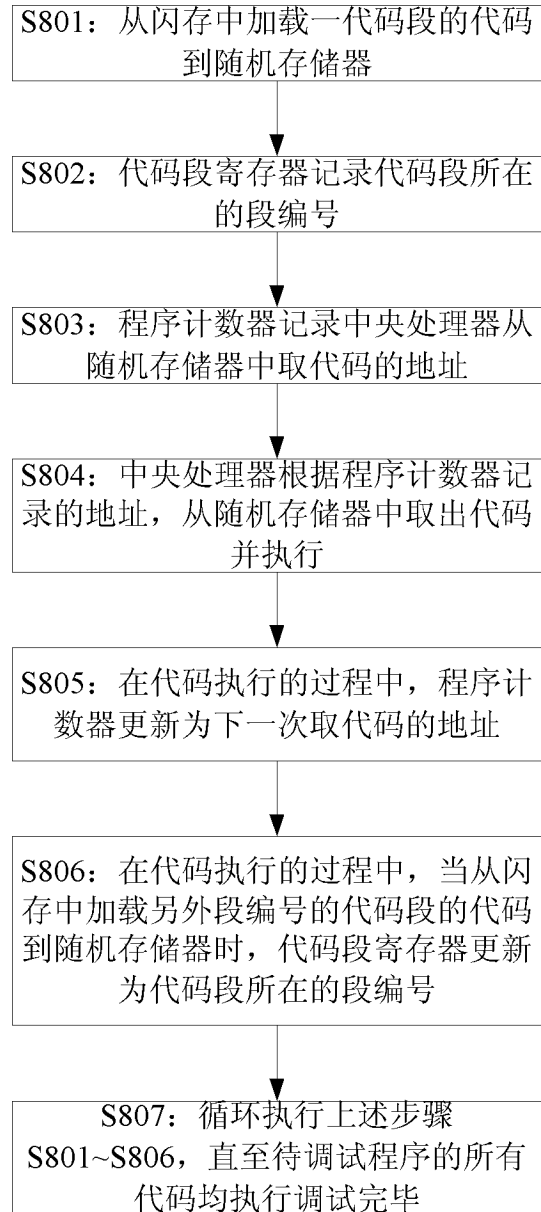


图 8

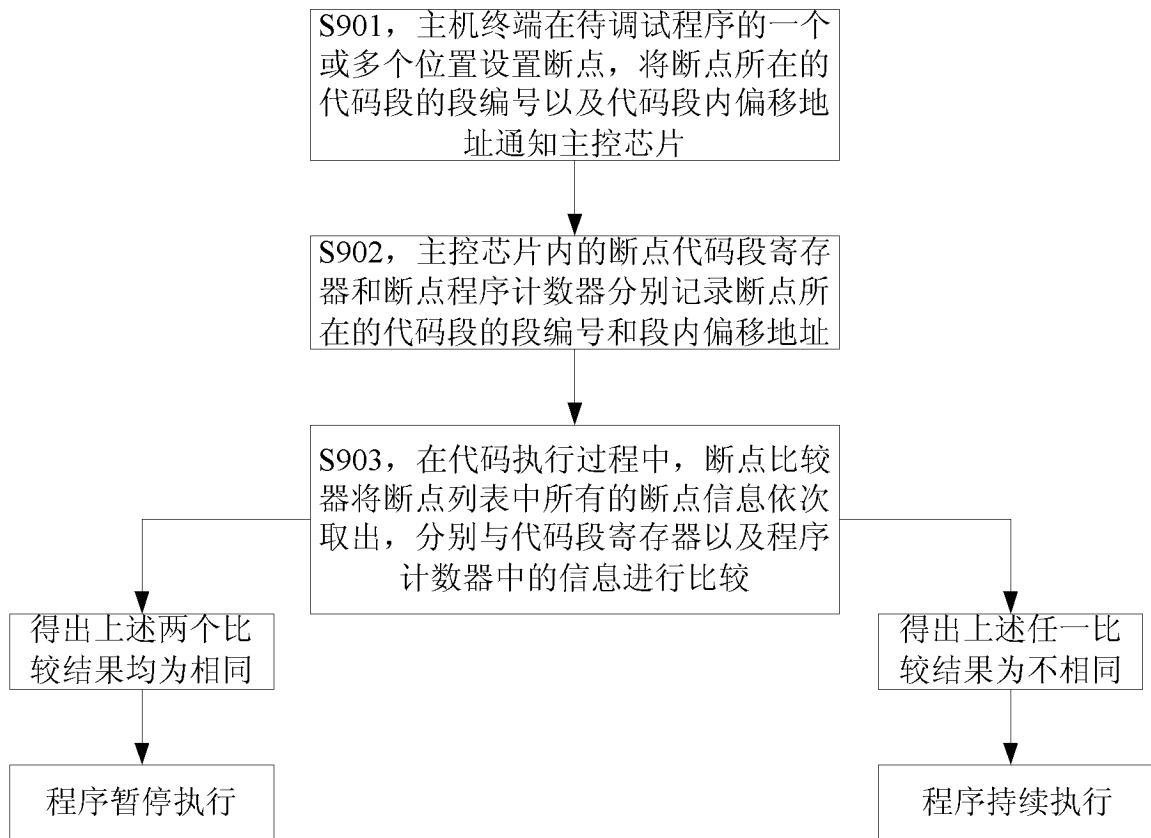


图 9