



(12) 发明专利申请

(10) 申请公布号 CN 104503765 A

(43) 申请公布日 2015.04.08

(21) 申请号 201410851749.7

(22) 申请日 2014.12.31

(71) 申请人 北京纵横机电技术开发公司
地址 100081 北京市海淀区大柳树路2号
申请人 中国铁道科学研究院机车车辆研究所

(72) 发明人 郑雪洋 夏菲 高吉磊 谢冰若
马驰 殷振环 马志文 李伟

(74) 专利代理机构 北京市铸成律师事务所
11313

代理人 孟锐

(51) Int. Cl.
G06F 9/44(2006.01)

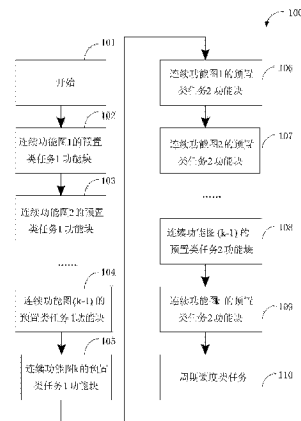
权利要求书1页 说明书10页 附图8页

(54) 发明名称

一种连续功能图编程方法

(57) 摘要

本发明涉及一种连续功能图编程方法,包括:在建立连续功能图CFC工程时,确定周期性任务的数量;为各个周期性任务指定调度周期;以及在使用连续功能图编程时,为每个功能块FB指定任务类型和执行顺序编号,其中所述任务类型是唯一的,相同的周期性任务具有相同的任务类型。



1. 一种连续功能图编程方法,包括:
在建立连续功能图 CFC 工程时,确定周期性任务的数量;
为各个周期性任务指定调度周期;以及
在使用连续功能图编程时,为每个功能块 FB 指定任务类型和执行顺序编号,其中所述任务类型是唯一的,相同的周期性任务具有相同的任务类型。
2. 根据权利要求 1 所述的方法,进一步包括:在编译连续功能图程序时,按照任务型生成任务函数,并且在每个任务函数中按执行顺序编号放置功能块代码。
3. 根据权利要求 2 所述的方法,进一步包括:通过连续功能图编译器将连续功能图编译成可被目标机调度核识别的可执行代码。
4. 根据权利要求 3 所述的方法,进一步包括:在目标机上按照所述任务类型建立对应的线程。
5. 根据权利要求 4 所述的方法,其中,包括目标机上包括多个线程,且所述多个线程具有各自的优先级,其中高优先级的线程可以中断低优先级的线程而先执行。
6. 根据权利要求 5 所述的方法,其中,调度周期越短的周期性任务对应的线程的优先级越高。
7. 根据权利要求 1 所述的方法,其中,所述周期性任务的数量为 8 或 8 的幂次个。
8. 根据权利要求 1 所述的方法,其中,所述任务类型与调度周期相对应。
9. 根据权利要求 4 所述的方法,其中,各个线程会在调度时间到时调用相应的周期任务函数。
10. 根据权利要求 4 所述的方法,其中,各个线程在其触发条件成立时启动。

一种连续功能图编程方法

技术领域

[0001] 本发明涉及一种工业控制编程方法,特别地涉及连续功能图编程方法。

背景技术

[0002] 连续功能图(CFC-Continuous Function Chart)是一种图形化编程语言。连续功能图编程具有以下特点:(1)通过绘制功能图表来自动生成程序;(2)拥有强大的预制程序库;(3)通过简单的连线来降低开发成本并减少错误。连续功能图不仅仅具有可编程逻辑控制器PLC(Programmable Logic Controller)典型的元素,例如:输入/输出、定时器、计数器、符号表等,而且具有图形化编程语言语言的特性,其非常适合于过程控制和系统工程。CFC源于IEC 61131-3标准规定,是五种编程语言之一的功能块图(Function Block Diagram)。连续功能图与功能块图的主要特点在于其展示出了其隶属的资源与任务。虽然连续功能图并没有相关国际标准,但在工业控制领域被广泛使用。

[0003] 然而,连续功能图CFC编程不够灵活,无法满足用户在工业控制方面的实际需要,因此,非常有必要提供一种更为灵活的连续功能图CFC编程方法。

发明内容

[0004] 针对以上技术问题,本申请提出了一种连续功能图编程方法,包括:在建立连续功能图CFC工程时,确定周期性任务的数量;为各个周期性任务指定调度周期;以及在使用连续功能图编程时,为每个功能块FB指定任务类型和执行顺序编号,其中所述任务类型是唯一的,相同的周期性任务具有相同的任务类型。

[0005] 如上所述的方法,进一步包括:在编译连续功能图程序时,按照任务型生成任务函数,并且在每个任务函数中按执行顺序编号放置功能块代码。

[0006] 如上所述的方法,进一步包括:通过连续功能图编译器将连续功能图编译成可被目标机调度核识别的可执行代码。

[0007] 如上所述的方法,进一步包括:在目标机上按照所述任务类型建立对应的线程。

[0008] 如上所述的方法,其中,包括目标机上包括多个线程,且所述多个线程具有各自的优先级,其中高优先级的线程可以中断低优先级的线程而先执行。

[0009] 如上所述的方法,其中,调度周期越短的周期性任务对应的线程的优先级越高。

[0010] 如上所述的方法,其中,所述周期性任务的数量为8或8的幂次个。

[0011] 如上所述的方法,其中,所述任务类型与调度周期相对应。

[0012] 如上所述的方法,其中,各个线程会在调度时间到时调用相应的周期任务函数。

[0013] 如上所述的方法,其中,各个线程在其触发条件成立时启动。

附图说明

[0014] 下面,将结合附图对本发明的优选实施方式进行进一步详细的说明,其中:

[0015] 图1是根据本发明的一个实施例的连续功能图CFC编程的方法示意图;

- [0016] 图 2 是根据本发明的一个实施例的 CFC 编程中周期任务的示意图；
- [0017] 图 3 是根据本发明的一个实施例的 CFC 编程中创建周期任务的示意图；
- [0018] 图 4a 是根据本发明的一个实施例的用于连续功能图编程的修改功能块输出连接器初始值方法；
- [0019] 图 4b 是根据本发明的一个实施例的多输出变量的实现方法的示意图；
- [0020] 图 5a 是根据本发明的一个实施例在 CFC 编程中设置物理量 M 的流程图；
- [0021] 图 5b 是根据本发明的一个实施例在 CFC 编程中对物理量 M 进行观测时的流程图；
- [0022] 图 6 是根据本发明的一个实施例的连续功能图文件比较的差异处理流程图；以及
- [0023] 图 7 是根据本发明的一个实施例的连续功能图文件差异的图形化展示示意图。

具体实施方式

[0024] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0025] 连续功能图 CFC 包括多个功能块。每个功能块能够实现一个简单的逻辑功能。通过将不同的功能块的输入连接器和输出连接器进行连接，连续功能图 CFC 编程可以实现复杂的逻辑功能控制。

[0026] 具体而言，连续功能图 CFC 的编程界面为图形界面。在该图形界面中包含了若干个图表 (Chart)。用户在图表 Chart 中以图形的方式插入已经编制好的块，指定这些块之间的连接关系，或者为这些块分配输入 / 输出地址。在这种通过鼠标点击拖拽的方式完成项目编写后，用户即可进行编译。接下来，CFC 程序将被整体编译，编译的结果以 SCL 代码的形式出现，最终将生成 STL 代码。用户将这些代码下载到 PLC 中即可执行。

[0027] 在连续功能图 CFC 编程中，对于不同调度时间的功能块，将其分配在不同的周期调度任务中，每个周期调度任务设置为固定的调度周期，实现功能块的多周期调度任务控制。在同一张功能图中，相同调度周期的功能块被分配固定的执行顺序。在多张连续功能图中，相同调度周期的功能块按照连续功能图的顺序进行执行。

[0028] 在现有的连续功能图 CFC 编程中，功能块都是按照周期任务进行调度的。这显然不够灵活，因为某些应用中部分功能块只需要在周期任务调度之前被执行一次，并不要进行周期调度。

[0029] 为了实现单次调度，并保证不同连续功能图之间的相互配合，根据本发明的一个实施例，提出了一种带多级预置类任务的连续功能图。举例而言，对于两级预置任务的功能图，在开始执行周期任务之前，先执行两级预置类任务，从而使得连续功能图 CFC 编程更具有灵活性。

[0030] 图 1 是根据本发明的一个实施例的连续功能图 CFC 编程的方法示意图。如图 1 所述，方法 100 包括：在程序开始 101 后，在步骤 102，执行连续功能图 1 中的预置值类任务 1 功能块，然后在步骤 103 依次执行连续功能图 2 中的预置值类任务 1 的功能块、……、以此类推，直到步骤 104，执行连续功能图 (k-1) 中的预置值类任务 1 的功能块以及在步骤 105 执行连续功能图 k 中的预置值类任务 1 的功能块，其中 k 是连续功能图的数量。在预置值

类任务 1 全部执行完之后,在步骤 106,执行连续功能图 1 中的预置值类任务 2 功能块、在步骤 107 执行连续功能图 2 中的预置值类任务 2 功能块、……、以此类推,直到步骤 108 执行连续功能图 (k-1) 中的预置值类任务 2 的功能块以及步骤 109 连续功能图 k 中的预置值类任务 2 的功能块,其中 k 为连续功能图工程中连续功能图的数量。预置值类任务 2 全部执行完之后,在步骤 110 开始执行周期类任务。

[0031] 本发明以上的实例提出了一种带两级预置类任务的连续功能图编程方法。在周期任务开始执行之前,执行两级预置类任务,可以大大节省目标控制器的硬件资源,并且极大地提高连续功能图编程的灵活性。

[0032] 根据本发明的另一个实施例,对于周期任务而言,本发明的连续功能图编程可变数量的周期任务。现有的支持连续功能图编程的产品都将周期任务数量都是固定的,一般在 5 个以内。然而,这种对于周期任务的限制极大地制约了软件开发的灵活性,而太多的周期任务数又会浪费系统资源。

[0033] 在本发明的一个实例中,连续功能图编程中的周期性任务的数量是可变的,而且是有优先级的。高优先级任务可以占先执行,即可以打断低优先级任务而先执行。通常,调度周期越短的任务优先级越高。

[0034] 图 2 是根据本发明的一个实施例的 CFC 编程中周期任务的示意图。如图 2 所示,对于每一个功能模块 FB(Function Block)而言,其包括以下几个参数:实例名、类型 FBTYPE、输入 IN 和输出 OUT。多个 FB 的输入 IN 和输出 OUT 相互连接形成周期任务。例如,在图 2 中连续功能图中共有 8 个独立的 FB 串,每个 FB 串代表一个周期任务,因此,图 2 的实例中周期任务数为 8 个。

[0035] 进一步地,每个周期任务可以包括不同数量的 FB。例如,在第一个周期任务中包括 n 个 FB,分别为 FB1-FBn;第二个周期任务中包括 3 个 FB,分别为 FBA-FBC。其中,每个 FB 具有一个执行顺序编号,其为正整数,代表在该 FB 在该周期任务中的执行顺序。其中,每个 FB 还具有一个任务类型,其代表了其所在的周期任务的编号。在本发明的实例中,以上的各个参数和编号可以灵活设置,以根据实际需求,获得 CFC 编程的灵活性。在图 2 的实例中,在连续功能图编辑器中可为每个功能块指定任务类型 (T1-T8) 和执行顺序编号。例如图中 1 功能块 FB1、FB2……FBn 隶属于 T1 任务,它们的执行顺序编号依次为 1、2……n。其他 T2-T8 任务与 T1 任务类似。

[0036] 图 3 是根据本发明的一个实施例的 CFC 编程中创建周期任务的示意图。首先,在步骤 301 中,在建立连续功能图 CFC 工程时,确定周期任务的数量,并且确定任务类型。以图 2 所示的实施例为例,即创建 T1-T8 的 8 个任务类型。

[0037] 接下来,在步骤 302 中,为 T1-T8 任务指定调度周期。在步骤 303 中,在使用连续功能图编程时,为每个功能块 FB 指定 T1-T8 中的一个任务类型和执行顺序编号。在步骤 304 中,在编译连续功能图程序时,按照任务类别生成 T1-T8 任务函数,每个任务函数中按执行顺序编号放置功能块代码。

[0038] 以下是图 2 的实施例的程序实例,仅用以说明本发明的技术方案。

[0039]

```
T1() /* T1 任务函数 */
{
    FB1(); /* T1 任务第一个功能块的代码 */
    FB2(); /* T1 任务第二个功能块的代码 */
    ...
    FBn(); /* T1 任务第最后一个功能块的代码 */
}
T2() /* T2 任务函数 */
{
    FBA(); /* T2 任务第一个功能块的代码 */
    FBB(); /* T2 任务第二个功能块的代码 */
    ...
    FBC(); /* T2 任务第最后一个功能块的代码 */
}
...
T8() /* T8 任务函数 */
{
    FBX(); /* T8 任务第一个功能块的代码 */
    FBY(); /* T8 任务第二个功能块的代码 */
    ...
    FBZ(); /* T8 任务第最后一个功能块的代码 */
}
```

[0040] 接下来,在步骤 305 中,通过连续功能图编译器将连续功能图编译成可被目标机调度核识别的可执行代码。在步骤 306,在目标机上,基于实时调度内核(例如,参考 POSIX. 1b),按优先级从高到低建立 T1-T8 的 8 个线程。在步骤 307,按工程指定调度周期激活相应线程,在线程中执行对应的 T1-T8 任务函数。为了方便理解,线程可理解为无限循环。目标机的实时调度内核可启动或终止任务线程执行。

[0041] 具体而言,每个周期任务被编译器编译成一个周期任务函数, T1()、T2()……T8()。每个周期任务函数内按执行顺序编号,调用隶属于该任务的所有功能块对应函数。例如周期任务函数 T1() 内会依次调用 FB1()、FB2()……FBn()。其他 T2-T8 任务函数与 T1() 任务函数类似。

[0042] 如图 3 所示,在编译好的连续功能图程序下载到目标机后,目标机调度核会启动 8 个周期任务线程,如图 3 所示,这 8 个线程会在调度时间到时调用相应的周期任务函数。具体而言, T1-T8 的 8 个线程在目标机上运行。各个线程在触发条件成立,即:条件 1 等于 1 时,启动。然后在相应的各个调度时间 T1-T8 到时后,会调用 T1-T8 的任务函数 T1()-T8(), 完成调用各个 T1-T8 的周期任务,实现相应的控制功能。以 T1 任务线程为例,在 T1 调度时间到后,会调用 T1 任务函数 T1()。其他 T2-T8 任务线程与 T1 任务线程类似。

[0043] 根据本发明的一个实施例,当前流行控制器硬件的存储器最小单位多为 8 或 8 的幂次位 (bit)。采用 8 个周期任务的方案,在编写任务调度算法和任务上下文数据结构时可达到存储器资源与调度的最佳平衡。例如,如果周期任务数少于 8 个,会浪费存储器资源,而如果多于 8 个会增加系统任务调度负荷。而且,周期任务设定为 8 个具有更高的编程灵活性。

[0044] 根据本发明的一个实施例,为了增加 CFC 编程的灵活性,本发明提出一种用于连续功能图 CFC 编程的可修改初始值的功能块输出的方法。

[0045] 在现有的连续功能 CFC 编程框图中,功能块的包括默认输入连接器和默认输出连接器,其中初始值是 0,且是不可以修改的。这种设置是针对连续功能图中的周期调度任务设置的。因为在周期调度任务下,功能块输出连接器的值是周期更新的,所以设置或修改输出连接器的初值没有意义。

[0046] 然而,在本发明的框架下,连续功能编程中除了周期调度任务还有预置任务。根据本发明的一个实施例,CFC 编程中还包括条件调用任务。当条件满足时,执行调用任务。当条件没有满足时,条件调用任务内的各功能块将不会被执行。这时,未被执行的功能块的输出连接器的值将会维持初值较长的时间,直至执行了该功能块才会被更新。针对这种情况,功能块的输出连接器的初值如果可以修改将是很有意义的。

[0047] 图 4a 是根据本发明的一个实施例的用于连续功能图编程的修改功能块输出连接器初始值方法。如图 4a 所示,在所述方法 400 中,连续功能图中任意一个功能块的执行包括:

[0048] 在步骤 410,在初始化阶段加载输入连接器初值,其中所述初值是在编程时和运行时可修改的。在步骤 420,在初始化阶段加载输出连接器的初值,其中所述初值是在编程时和运行时都可修改的。在步骤 430,在任务调度阶段,读输入连接器的值,执行功能块 FB 的代码,在功能块 FB 代码中更新功能块输出连接器的值。但由于功能块 FB 可能隶属于条件任务,因此在条件未满足时,所述功能块 FB 代码可能从未被执行,则所述功能块 FB 输出连接器将一直保持步骤 420 所述初始值。最后,在步骤 440,使用功能块 FB 输出连接器的值更新与该输出连接器相连接的输出变量的值。

[0049] 本发明的实例中所提出的用于连续功能图编程的可修改功能块输出连接器初始值的方法,改变了此前连续功能图 CFC 不能修改输出连接器初值弊端,可以极大地提高编程的灵活性。

[0050] 根据本发明的另一个实施例,为增加连续功能图编程的灵活性,本发明提出一种多输出变量。该多输出变量为在连续功能图中定义的全局变量,可以被不同的功能块的输出变量重复赋值。

[0051] 如所周知的,连续功能图编程中,最基本的程序单元是功能块(Function Block, FB),其与外界的接口就是其输入变量和输出变量。连续功能图还定义了全局变量,只要保证数据类型一致,该全局变量可以由任何任务的任何功能块的输出变量赋值,同时该全局变量还可以给任何任务的任何功能块的输入变量赋值,以此实现不同任务、不同功能块之间的数据传递。

[0052] 但是,目前的连续功能图中,全局变量只能被功能块的输出变量赋值一次,不可以被多个功能块的输出变量重复赋值,也就是说目前的连续功能图编程只支持单输出变量。

[0053] 针对上述问题,为了进一步提供编程的灵活性,本发明提出了一种用于连续功能图编程方法的多输出变量。该多输出变量是一种全局变量,可以定义为任意的数据类型,只要连续功能图的任意功能块的输出变量与其数据类型一致,就可以给该多输出变量重复赋值。根据本发明的一个实施例,该多输出变量可以接受相同或不同任务的多个功能块输出变量的赋值,其当前的值由最新一次的赋值操作决定。另外,该多输出变量还可以给相同或

不同任务的多个功能块的输入变量赋值。

[0054] 图 4b 是根据本发明的一个实施例的多输出变量的实现方法的示意图。如图 4b 所示,连续功能图编程的工程中,含有 T1, T2, T3...等任务。T1 任务中功能块 X1 的输出变量 2, T2 任务中功能块 Y1 的输出变量 1, T3 任务中功能块 Z1 的输出变量 2, 都与多输出变量 M 一样是数据类型 W。因此,它们都可以赋值给多输出变量 M。

[0055] 同时,在保证输出类型相同的前提下,多输出变量 M 还可以赋值给不同任务的不同功能块的输入变量,在图 4b 中, T1 任务中功能块 X2 的输入变量 1, T2 任务中功能块 Y2 的输入变量 2, T3 任务中功能块 Z2 的输入变量 1, 都与多输出变量 M 一样是数据类型甲,它们都可以被多输出变量 M 赋值。

[0056] 本领域技术人员应当了解,图 4b 仅仅给出的是一种具体实现方式,但不限于此。在连续功能图编程环境中,只要保证数据类型与多输出变量一致,不同任务的不同功能块都可以向多输出变量重复赋值,同时,多输出变量也可以向相同数据类型的不同任务不同功能块的输入变量赋值,这样可以更灵活地实现不同任务不同功能块的数据传递。

[0057] 根据本发明的一个实施例,在本发明的框架下,本发明提出了一种新的可定标的连续功能图数据类型。该数据类型可以表示任何的物理量:在编程时可设置物理量的基值和单位及其初始物理值。在观测调试时可直接读取其物理值。这种新的数据类型,可以更加连续功能图 CFC 编程的灵活性,减少编程人员的工作量。

[0058] 目前的连续功能图编程中,数据类型有布尔型、无符号单字节整型、有符号单字节整型、无符号双字节整型、有符号双字节整型、无符号四字节整型、有符号四字节整型、时间类型等数据类型。

[0059] 然而,在连续功能图 CFC 编程和调试过程中,通常会存在许多的物理量。如果强行将这些物理量定标转化为上述数据类型才能进行使用,会带来非常大的不便,限制了编程的灵活性。在进行观测调试时,还必须将上述数据类型手动转化为物理量。这极大地增加了连续功能图编程和调试的繁琐程度。

[0060] 根据本发明的一个实施例,提出一种可定标的连续功能图数据类型,物理量类型 M。对于该新的连续功能图可定标数据类型,物理量类型 M 的基值 M_j 、单位 MD 和物理量初始值 M_0 需要在连续功能图编程时设置。而在具体的功能块 FB 执行过程中,根据基值、物理量值和定标值,自动将物理量转化为二进制数据,其中定标值表示二进制数据的长度。定标值越大,所能表示的数的范围越大,精度也越高。

[0061] 根据本发明的一个实施例,对物理量进行设置时,其公式为:

[0062] 二进制数据 = 物理量值 / 基值 $\times 2^{\text{定标值}}$ 。

[0063] 该二进制数据由目标机获取后进行运算。而在通过上位机进行观测时,目标机将二进制数据发送给上位机,上位机根据其基值和定标值进行运算得到物理值进行显示,其公式为:

[0064] 物理值 = 二进制数据 / $2^{\text{定标值}} \times$ 基值。

[0065] 图 5a 是根据本发明的一个实施例在 CFC 编程中设置物理量 M 的流程图和图 5b 是根据本发明的一个实施例在 CFC 编程中对物理量 M 进行观测时的流程图。如图 5a 所示,设置方法 510 中,在步骤 511,用户在上位机上输入物理量类型的一个实例 N 的基值 N_j 和单位 ND。在步骤 512,上位机计算该实例 N 的转换系数,即 $2^{\text{定标值}} /$ 基值。在步骤 513,上位机根

据物理量 N 的实际物理值 N_i ($i = 0, 1, \dots$) 与 N 的转换系数, 计算 N 的二进制值, 即 $N_i \times 2^{\text{定标值}} / \text{基值}$ 。在步骤 514 中, 将上述步骤中得到的 N 的二进制值下载到目标机, 在目标机中进行运算。同样地, 如图 5b 所示, 在观测方法 520 中, 在步骤 521, 将物理量 N 的二进制数据从目标机上传到上位机。在步骤 522 中, 上位机计算该实例 N 的转换系数, 即 $2^{\text{定标值}} / \text{基值}$ 。在步骤 523, 上位机根据 N 的二进制值和转换系数计算 N 的物理值, 即 N 的二进制值 / 转换系数。在步骤 524 中, 用户就可以很方便地从上位机观测到 N 的物理值。

[0066] 本发明的实例提出了一种可定标的连续功能图数据类型, 物理量 M。该数据类型可以表示任何的物理量。物理量类型的数据在使用时非常方便, 仅需要在编程时可设置物理量的基值和单位。初始物理值即可以默认也可以在编程时设置。在观测调试时, 本发明的实例中可直接读取其物理值。如上面实施例所说明的, 新的数据类型, 物理量 M, 的加入方便了连续功能图编程和调试中物理量的设置和观测, 使其更加易于操作, 大大提高了连续功能图的编程和调试效率。

[0067] 在本发明的框架下, 对现有的连续功能图 CFC 编程所作出的另一个调整是增加了比较功能。这一功能对于 CFC 编程人员非常重要。因为当前的连续功能图编程产品都不具备连续功能图文件图形化比较功能。然而, 通过软件设计文档很难准确全面地反应软件设计上的差异。这对软件版本管理和软件维护造成很大不便, 增加了软件开发人员在这方面的工作量。这也限制了 CFC 编程的灵活性。

[0068] 根据本发明的一个实施例所提供的连续功能图比较的方法自动识别两个 CFC 文件的设计差异, 并将差异以易于对比的文本和图形化方式输出。这在对比软件版本或软件分支差异性时尤为必要, 可减轻软件设计、测试、维护人员的工作量, 提高工作效率。

[0069] 具体而言, 连续功能图差异分两类:

[0070] (1) 非功能性差异, 包括: 注释的内容、功能块的摆放位置、连接线的起点 / 终点 / 拐点坐标。此类差异不影响最终的软件功能, 仅影响连续功能图可读性, 简称 N 差异; 以及

[0071] (2) 功能性差异, 包括: 功能块实例的增减、同名功能块类型的变化、同名功能块所属任务类别和执行顺序编号的差异, 功能块连接器或全局信号的连接关系, 功能块连接器与全局信号初始值。此类差异会影响软件功能, 简称 F 差异。

[0072] 本发明的实例的连续功能图文件之间的比较对象包括: 功能块、信号、注释块、连接。既包括了 N 差异, 也包括了 F 差异。

[0073] 本发明的实例的连续功能图比较原理如下:

[0074] 采集两个连续功能图文件中所有编程对象的信息, 包括功能块、信号、注释块与连接信息, 分别将两个连续功能图的对象信息组织到两个排序表中。表中每个对象都带有特征字符串。功能块与注释块的特征字符串为块名, 信号的特征字符串为信号名, 连接的特征字符串为连接起点对象 (功能块或信号) 名加终点对象 (连接器或信号) 名。在两个表中搜索特征字符串相同的对象, 并进一步比较其对象信息。特征字符串仅在一个表中存在的对象, 直接判定该对象为差异对象。

[0075] 根据以上比较原理, 本发明的实例做出如下设定:

[0076] 1. 连续功能图文件中的功能块名具有唯一性, 即无重名的功能块;

[0077] 2. 连续功能图中的信号名具有唯一性, 即无重名的信号;

[0078] 3. 连续功能图中每个任务里的功能块执行顺序号具有唯一性, 即在同一个任务中

无序号相同的两个功能块。

[0079] 根据本发明的一个实例,参与比较的功能块信息包括:

- [0080] 1. 功能块所属任务类型;
- [0081] 2. 功能块的执行顺序编号;
- [0082] 3. 功能块版本号;
- [0083] 4. 功能块所有连接器;
- [0084] 5. 连接器初始值;
- [0085] 6. 布尔类型输入连接器是否求反;
- [0086] 7. 输出连接器是否带有非易失属性;
- [0087] 8. 连接器注释;
- [0088] 9. 功能块注释;以及
- [0089] 10. 功能块的坐标

[0090] 根据本发明的一个实例,参与比较的信号信息包括:

- [0091] 1. 信号的类型;
- [0092] 2. 信号的初始值;以及
- [0093] 3. 信号的描述。

[0094] 根据本发明的一个实例,参与比较的注释块信息包括:

- [0095] 1. 注释内容文本;以及
- [0096] 2. 注释块的位置坐标;

[0097] 根据本发明的一个实例,参与比较的连接信息包括:

- [0098] 1. 连接起点坐标;
- [0099] 2. 连接终点坐标;以及
- [0100] 3. 连接拐点数量与每个拐点坐标。

[0101] 根据本发明的一个实例,对于对象数据表中的每个对象,记录该对象所在页码与坐标信息,以便实现图形化对比。

[0102] 根据本发明的一个实例,所有差异信息都被记录到一个差异数据表中。在对比结束后,以文本方式按条目输出所有差异信息。在一个更具体的实例中,用户可以通过双击差异文本条目,在两个并列的窗格中图形化地、高亮显示差异对象。

[0103] 根据本发明的一个实例,差异数据表中每个差异条目包含以下信息:

- [0104] 1. 差异类别:非功能性差异(N)或功能性差异(F);
- [0105] 2. 差异索引;以及
- [0106] 3. 差异描述文本。

[0107] 其中,所谓差异索引用于定位差异的页码和坐标。根据本发明的一个实例,差异索引可以为差异编号,根据该差异编号可以确定该差异所对应的对象。如果某个差异为不存在对应的对象,则差异索引定义为-1。

[0108] 根据本发明的一个实例,当用户选择或双击差异文本条目时,可依据差异索引快速确定该差异所对应的对象;然后,根据对应对象的页码和坐标定位出该差异的页码和坐标,从而确定需要向用户显示的内容,并突出显示该差异对应的对象。

[0109] 在连续功能图编写的应用软件升级、建立分支、除错时经常需要比较两个连续功

能图文件的差异。在本发明之前,这部分工作主要依靠程序员编写的软件修改日志和版本说明,非常具有局限性。例如,在软件修改过程中因程序员失误而引入的设计错误是不可能记录到文档中的。本发明的连续功能图文件比较方法可以全面直观地反映出连续功能图软件的设计差异,并且这种自动差异分析比手工比较或二进制比较的效率更高,也更准确。

[0110] 图6是根据本发明的一个实施例的连续功能图文件比较的差异处理流程图。如图6所示,该方法包括:

[0111] 首先,提取第一个连续功能图文件CFC1中所有的程序页、功能块、信号、注释块、连接信息,组织成易于快速搜索的、已排序的内部数据表(简称表1);

[0112] 其次,提取第二个连续功能图文件CFC2中所有的程序页、功能块、信号、注释块、连接信息,组织成易于快速搜索的、已排序的内部数据表(简称表2);并且,初始时表2中所有对象都标记为未比较对象;

[0113] 接下来,遍历表1。依次提取表1中的对象(简称表1对象),根据表1对象的特征在表2中搜索特征相同的对象。若不存在,则标记该表一对象为有差异,对象写入差异表(对于相关表1对象在表1中的索引,若不存在则为-1)。若存在,则进一步比较这两个对象的数据信息,若相同则各自标记为无差异对象;若不相同则各自标记为有差异对象,并将差异信息写入差异信息表。

[0114] 如图6所示,赋初值 $i = 0$,进入循环, $i = i + 1, i < \text{表1索引范围}$ 。提取表1对象 $[i]$,根据表1对象 $[i]$ 的特征搜索表2中是否有对象 $[i]$ 的特征。如果没有找到,则在差异表中记录差异内容,并且该差异的索引为-1。如果找到,例如是表2中的对象 $[j]$,则将表2中的对象 $[j]$ 标记为已经比较过。下一步确定,确定表1对象 $[i]$ 和表2中对象 $[j]$ 的信息是否一致。如果一致,则返回循环, $i = i + 1$,否则,在差异表中记录差异内容,并且该差异的索引为正整数的顺序编号,例如差异1,差异2,以此类推。差异编号是唯一的,在本实施例中也可以作为差异索引。根据差异编号能够确定表1对象 $[i]$ 和表2中对象 $[j]$ 。返回循环, $i = i + 1$ 。

[0115] 接下来,遍历表2。依次提取表2中仍标记为未比较的对象,标记为差异对象,并写入差异信息表(对于相关表2对象在表2中的索引,若不存在则为-1)。

[0116] 如图6所示,赋初值 $i = 0$,进入循环, $i = i + 1, i < \text{表1索引范围}$ 。提取表2中的仍标记为未比较的对象 $[k]$,判断表2中的对象 $[k]$ 是否已经比较过,如果仍为未比较过,则将其标记为已经比较过,并且在差异表中记录差异内容,并且该差异的索引为-1。返回循环, $i = i + 1$ 。

[0117] 最后,以文本格式输出差异信息表的内容。

[0118] 图7是根据本发明的一个实施例的连续功能图文件差异的图形化展示示意图。如图7所示,所有差异信息都以文本方式输出到差异文本输出窗格中。每项差异文本都以差异类别和编号开头,该编号也是差异表项的索引值。

[0119] 如图7所示,当用户双击差异文本,则此条差异文本被高亮显示。同时自动提取差异编号,并且根据此项差异编号可查表索引出表一和表二中的相应差异对象。根据表一和表二中存储的差异对象页码与坐标来确定两个连续功能图窗格内的显示内容,同时把相应差异对象高亮。

[0120] 在以上的实施例中介绍了本发明为了提高连续功能图(CFC)编程的灵活性。必须

说明的是,虽然上述实施例分别介绍了其各自对连续功能图(CFC)编程做出的改进;但是,这些改进之间并不是无法结合的。所有以上的改进中的一个或多个之间可以相互组合以进一步增加CFC编程的灵活性。这些组合或相互结合也同样在本发明的范围内。

[0121] 上述实施例仅供说明本发明之用,而并非是对本发明的限制,有关技术领域的普通技术人员,在不脱离本发明范围的情况下,还可以做出各种变化和变型,因此,所有等同的技术方案也应属于本发明公开的范畴。

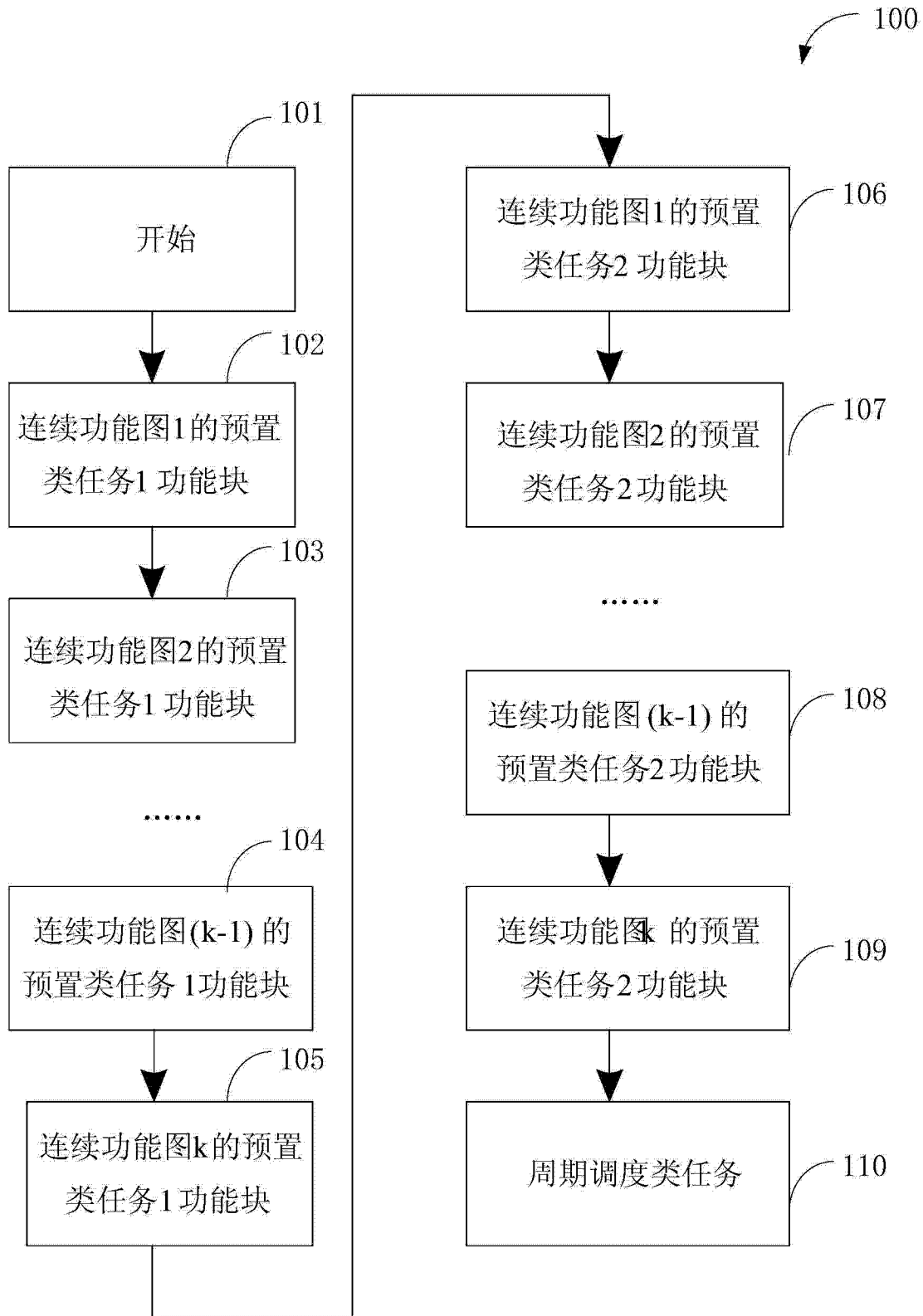


图 1

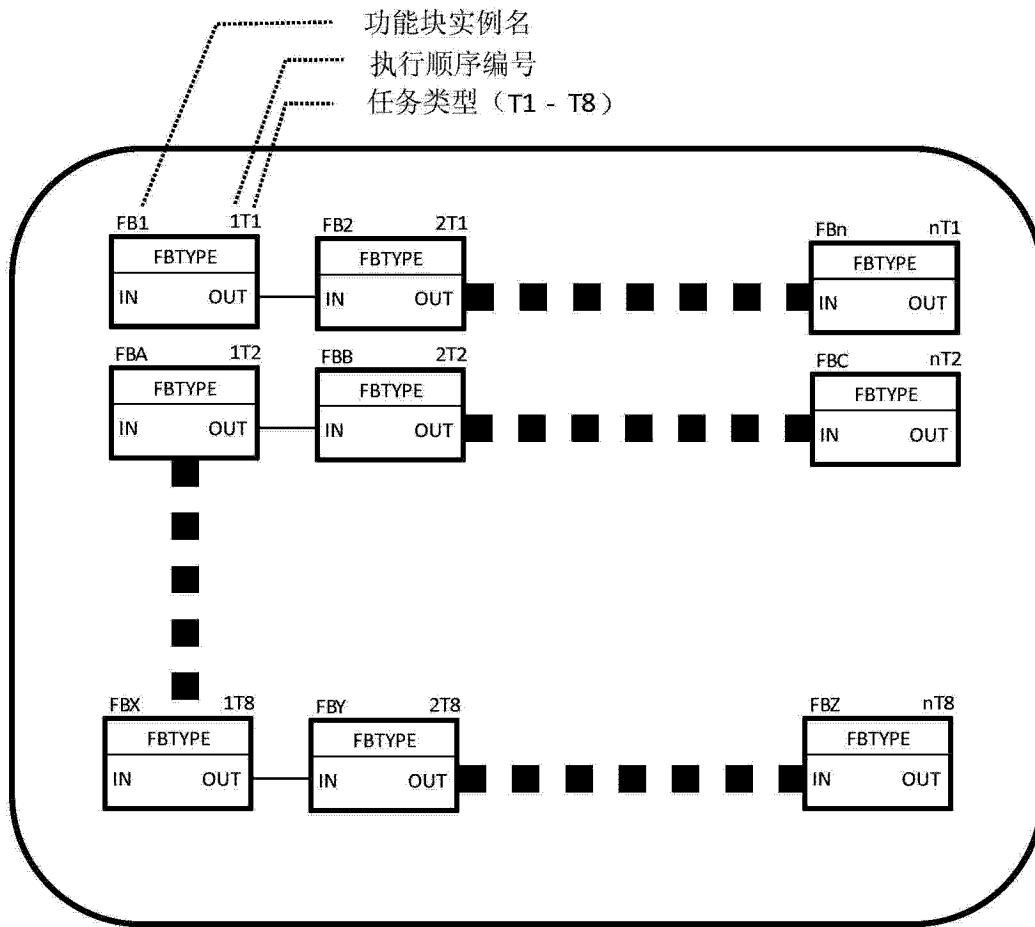


图 2

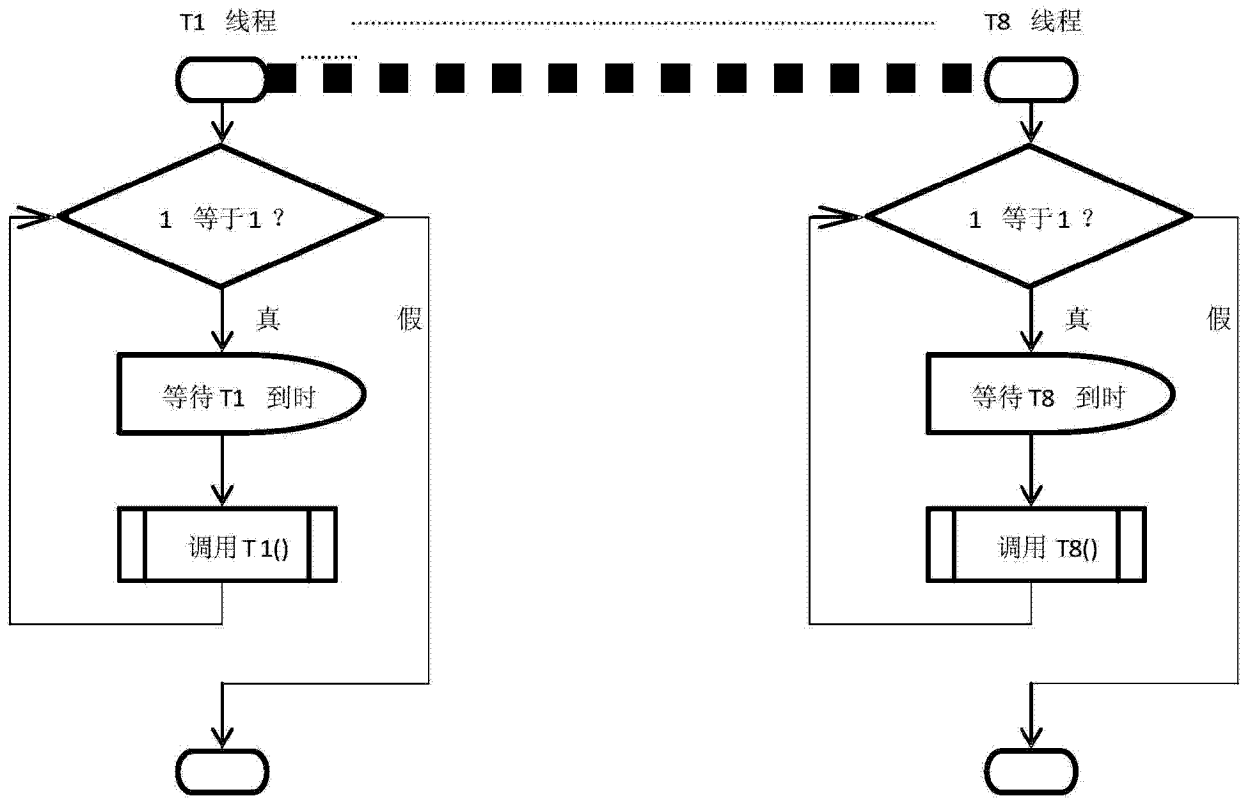


图 3

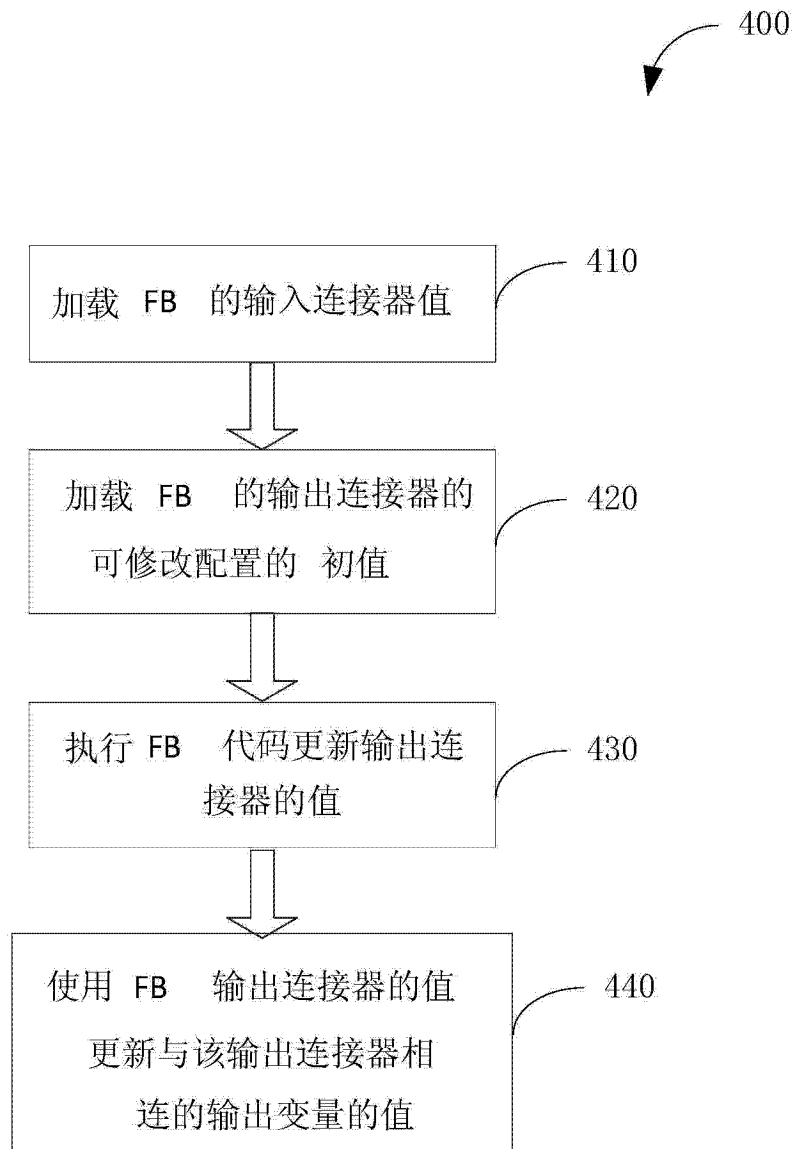


图 4a

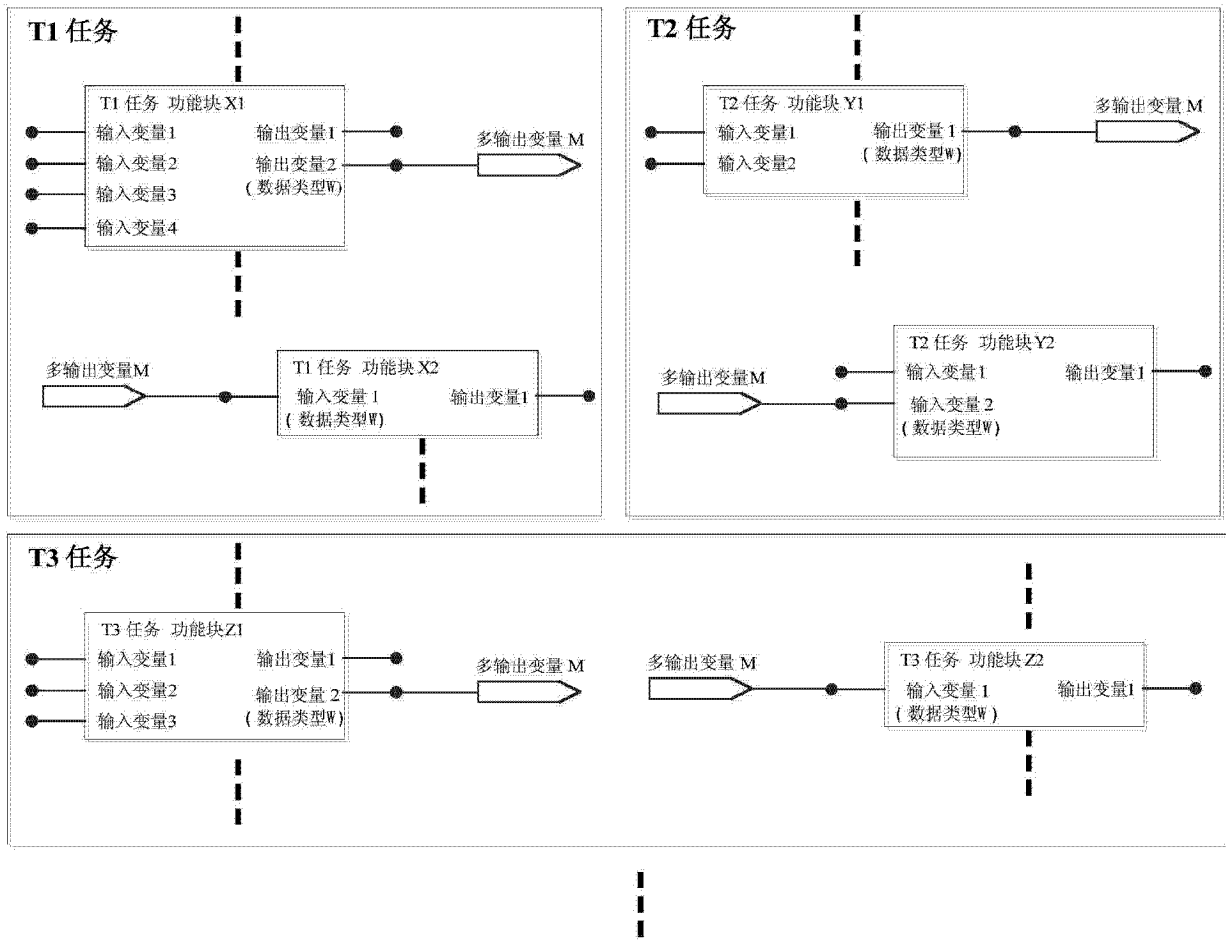


图 4b

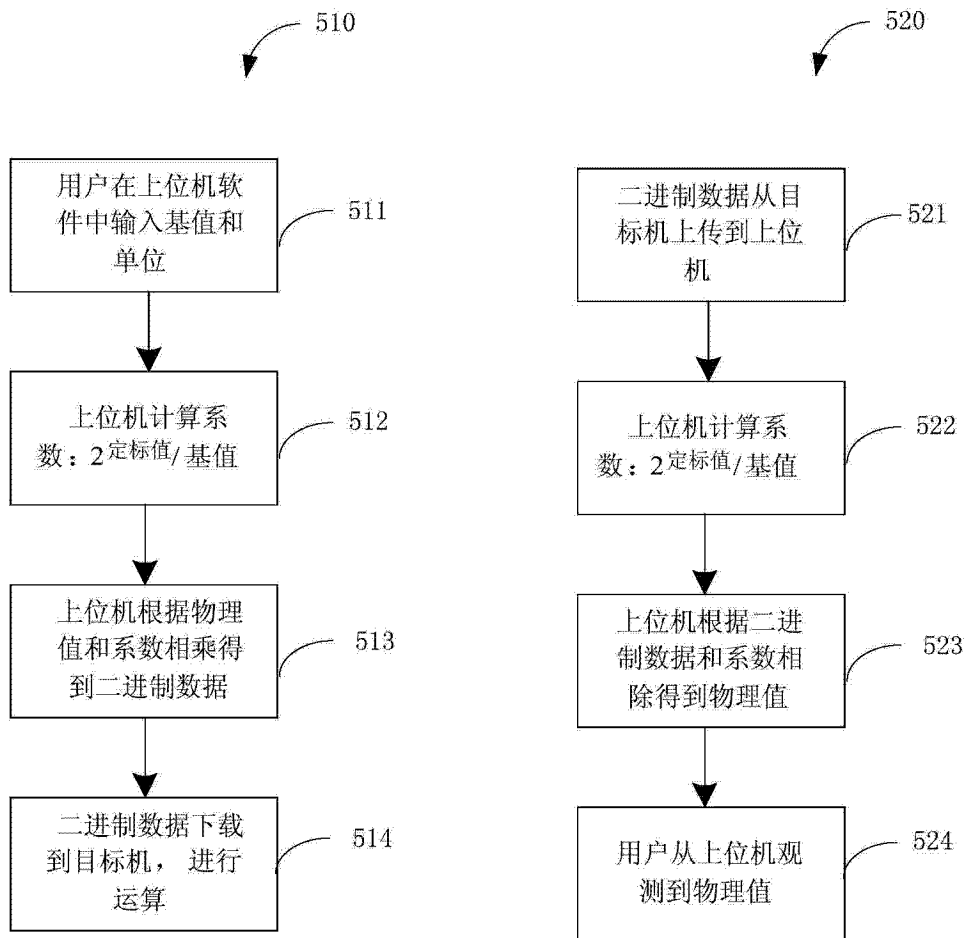


图 5a

图 5b

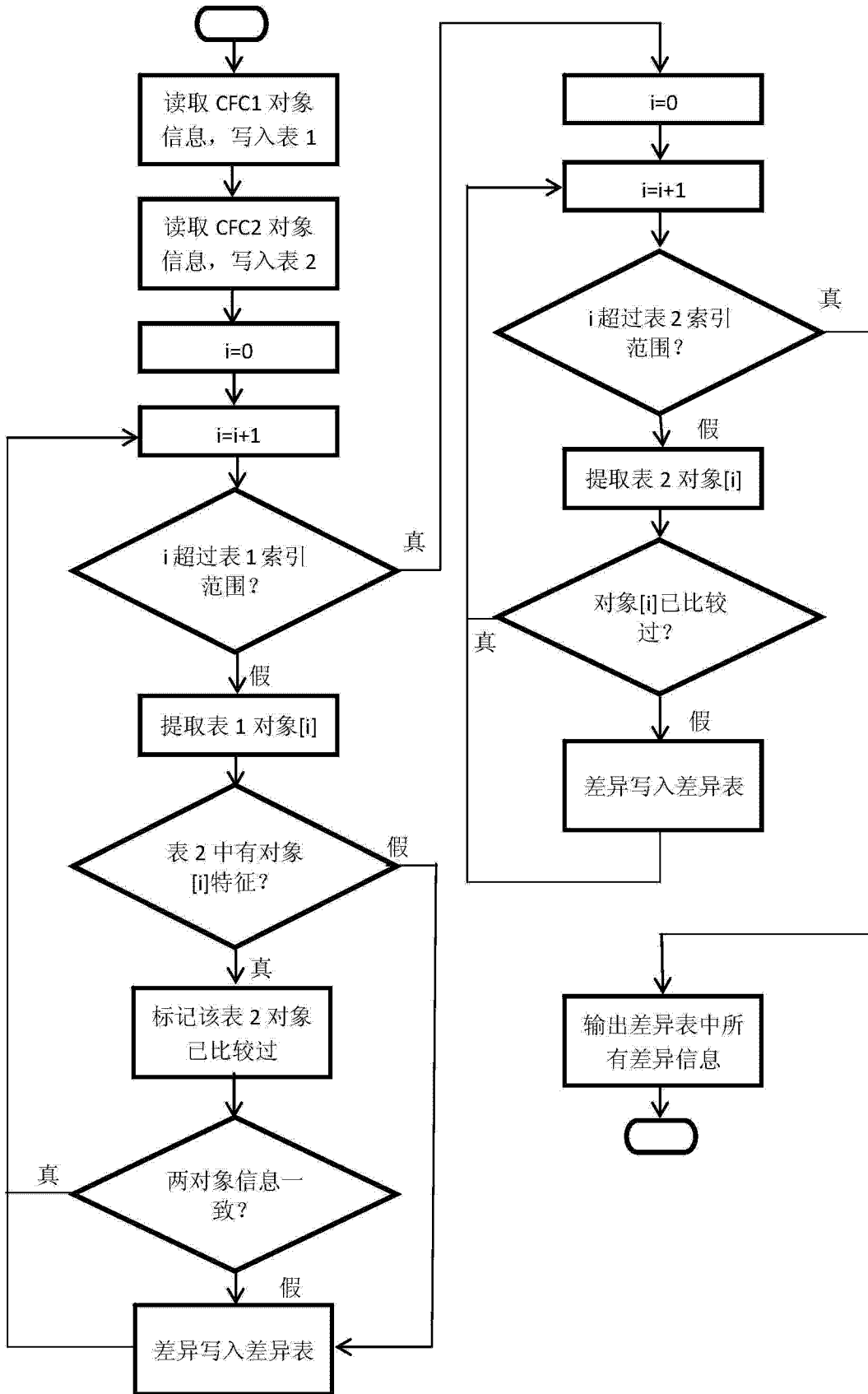


图 6

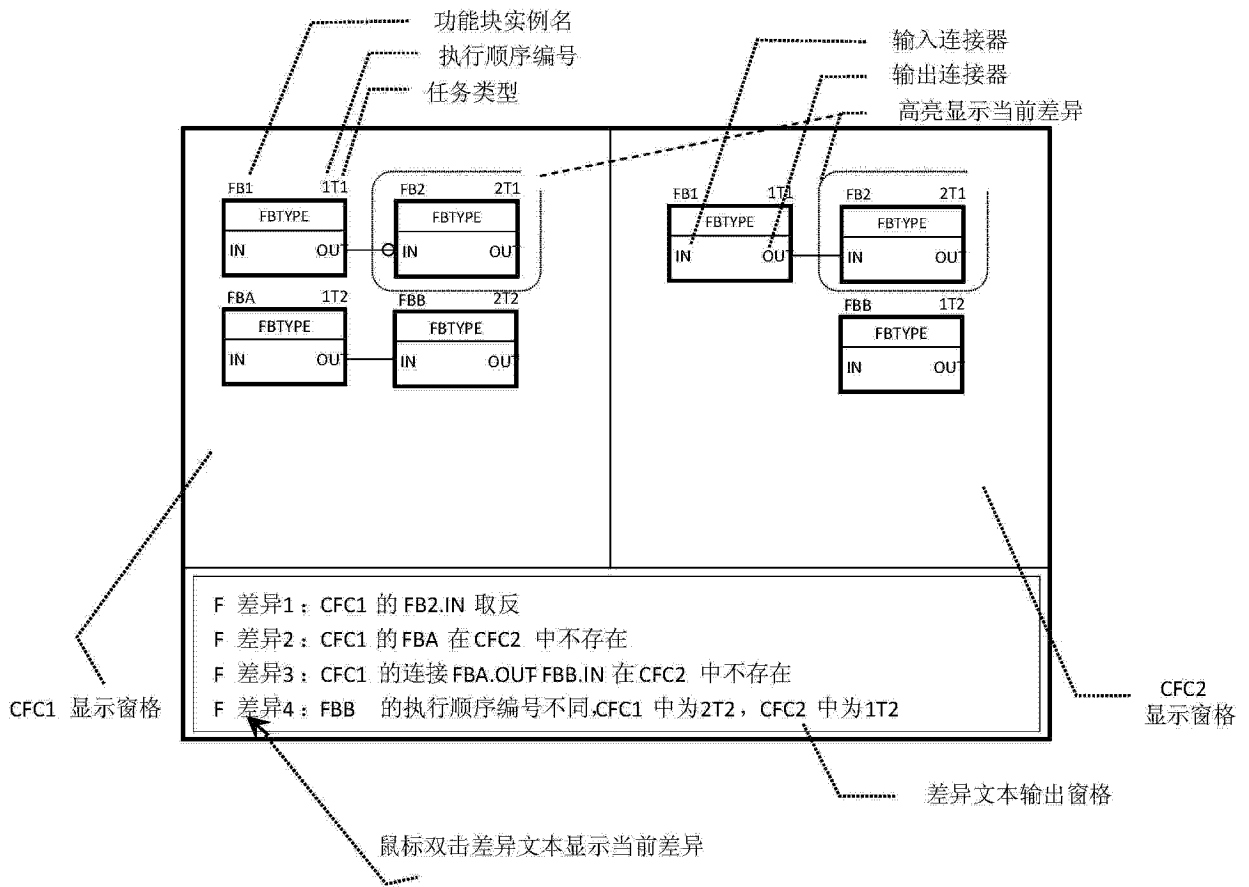


图 7