



(12)发明专利申请

(10)申请公布号 CN 107729050 A

(43)申请公布日 2018.02.23

(21)申请号 201710864010.3

(22)申请日 2017.09.22

(71)申请人 中国科学技术大学苏州研究院  
地址 215123 江苏省苏州市工业园区独墅湖高教区仁爱路166号

(72)发明人 李曦 陈香兰 王超 罗海钊  
万波

(74)专利代理机构 苏州创元专利商标事务所有  
限公司 32103  
代理人 范晴 丁浩秋

(51)Int.Cl.  
G06F 9/30(2006.01)  
G06F 9/48(2006.01)

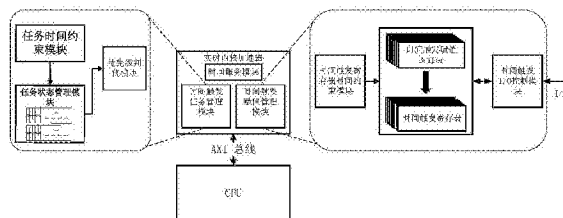
权利要求书2页 说明书5页 附图2页

(54)发明名称

基于LET编程模型的实时系统及任务构建方法

(57)摘要

本发明公开了一种基于LET编程模型的实时系统,包括实时内核加速器和实时执行器;实时内核加速器包括,时间服务模块,用于提供机器周期精确的时间信息;时间触发任务管理模块,用于对时间触发的任务管理;时间触发赋值管理模块,用于对时间触发的赋值;实时执行器包括,硬件抽象层,提供中断管理和硬件加速器功能接口;任务TCB结构数据,由内核的任务初始化API生成;内核API,提供任务调度、任务管理接口。通过对时间触发的任务管理、时间触发的赋值和周期精确的时间服务的支持,达到方便快捷地开发出具有时间语义且周期精确的基于LET任务的实时系统的目的。



CN 107729050 A

1. 一种基于LET编程模型的实时系统,其特征在于,包括实时内核加速器和实时执行器;所述实时内核加速器包括,

时间服务模块,用于提供机器周期精确的时间信息;  
时间触发任务管理模块,用于对时间触发的任务管理;  
时间触发赋值管理模块,用于对时间触发的赋值;  
所述实时执行器包括,  
硬件抽象层,提供中断管理和硬件加速器功能接口;  
任务TCB结构数据,由内核的任务初始化API生成;  
内核API,提供任务调度和任务管理的接口。

2. 根据权利要求1所述的基于LET编程模型的实时系统,其特征在于,所述时间服务模块包括一加法器和一寄存器,所述寄存器保存系统复位后的CPU周期数,每个CPU周期到来时,加法器对该寄存器加一,记录CPU启动后的Cycle数。

3. 根据权利要求1所述的基于LET编程模型的实时系统,其特征在于,所述时间触发任务管理模块包括,

任务时间约束模块,包括多个任务时间约束子模块,分别对应每个任务,所述任务时间约束子模块包括一相位寄存器和一周周期掩码寄存器,用于接收时间服务模块提供的时间信息,并使用相位寄存器和周期掩码寄存器根据时间信息判断时间触发任务是否满足时间约束;

任务状态模块,用于管理任务的任务状态,包括一任务就绪寄存器和一任务存在标识寄存器;

优先级判优模块,接收任务状态模块的任务状态信息并输出当前处于就绪状态的最高优先级任务的优先级,当最高优先级任务产生变化时,向CPU发送中断信号,使其能够进行上下文切换来切换任务。

4. 根据权利要求1所述的基于LET编程模型的实时系统,其特征在于,所述时间触发赋值管理模块包括,

时间约束模块,包括多个时间约束子模块,用于对普通时间触发赋值寄存器模块和时间触发I/O赋值寄存器模块进行约束,所述时间约束子模块包括一相位寄存器和一周周期掩码寄存器,用于接收时间服务模块提供的时间信息,根据时间信息判断时间触发赋值是否满足时间约束;

时间触发寄存器模块,包括多个普通时间触发赋值寄存器子模块和I/O时间触发赋值子模块,所述普通时间触发赋值寄存器子模块和I/O时间触发赋值子模块均包括一时间触发赋值Buffer寄存器和一时间触发寄存器,对普通时间触发赋值寄存器子模块和I/O时间触发赋值子模块的写操作为对时间触发赋值Buffer寄存器写,读操作为对时间触发寄存器的读,当且仅当从相应的时间约束子模块传过来满足时间约束信号时,时间触发寄存器的值更新为时间触发赋值Buffer寄存器的值;

时间触发赋值控制模块,包括一数据方向寄存器和与外界相连的外界引脚,外界引脚根据数据方向寄存器的值决定其数据方向。

5. 根据权利要求1所述的基于LET编程模型的实时系统,其特征在于,所述实时内核加速器通过AXI总线接口与处理器相连,实时内核加速器的各个寄存器都映射到相应的内存

地址中,处理器通过访问映射的内存地址与内核加速器进行通信和同步。

6. 根据权利要求1所述的基于LET编程模型的实时系统,其特征在于,所述内核API包括,

任务初始化API,接收用户的任务初始化结构体,初始化相应任务的TCB和栈结构,同时在硬件加速器的时间触发任务管理器中注册任务;

启动调度器API,用于启动硬件加速器中时间触发任务管理;

任务挂起API,用于挂起时间触发任务管理器中处于就绪状态的任务;

任务恢复API,用于把处于挂起状态的任务转变成就绪状态;

任务删除API,用于删除任务的TCB和反注册时间触发任务管理器中的任务。

7. 一种基于LET编程模型的实时系统的任务构建方法,其特征在于,包括以下步骤:

S01: 用户创建LET任务,定义LET任务的周期、相位、输入和输出,通过内核API的任务初始化API初始化任务,将任务属性初始化为时间触发;

S02: 使用启动调度器API,启动硬件加速器中时间触发任务管理模块;

S03: 到达任务的输入时刻和释放时刻,实时内核加速器的时间触发赋值模块触发赋值行为,完成任务的输入,时间触发任务模块把任务从等待状态修改为就绪状态,并向CPU发送中断信号;

S04: CPU接收到中断信号,完成上下文切换后执行任务,CPU执行完任务,将任务的输出放到时间触发赋值的Buffer,然后执行其他优先级比较低的任务或者事件触发的任务;

S05: 任务的输出时刻到达,实时内核加速器的时间触发赋值模块触发赋值行为,完成任务的输出。

## 基于LET编程模型的实时系统及任务构建方法

### 技术领域

[0001] 本发明属于实时系统、计算机硬件加速技术领域,具体地涉及一种基于LET编程模型的实时系统及任务构建方法。

### 背景技术

[0002] 实时性是实时系统最重要的特点。实时性是指系统的正确运行不仅要求输出结果正确,而且要求计算结果输出结果及时产生。在硬实时系统中,任务错过了预定的截止时间,有可能引发灾难性的后果。

[0003] 但是,传统的体系结构如X86、MIPS为了追求平均性能,抛弃了对时间语义的支持;以及所有的编程抽象如汇编语言、C语言,都从程序逻辑上把时间的属性抽象掉。这些技术在一定程度上简化了编程难度,但是,当系统必须保证其时间属性(特别是硬实时系统)时,就无法简单地做到对时间语义的表达。

[0004] 逻辑运行时间(LET)是支持显式描述系统时间行为的一种编程模型,该模型主要将任务抽象为三个部分分别调度执行,包括输入、计算和输出。输入和输出动作分别发生在逻辑时间的开始和结束时刻,且认为其执行过程在逻辑上瞬间完成,不消耗执行时间。任务的计算过程实现为可抢占的线程模型,在输入时刻被释放,且要求在输出时刻之前完成计算过程。在LET中,因任务的输入和输出不受任务实际执行过程的影响,能有效减少任务抖动,具有输入输出的可组合性。任务输入和输出之间的时间长度源自于系统的时间需求。

[0005] 然而,现有的支持LET编程模型的系统大都基于传统的软硬件方式实现定时输入输出的语义。软件层面,由操作系统提供时钟服务。硬件方面,则需要专门能提供周期性的时钟中断的平台提供精确的时钟信号。此类系统的时间的精确度会受到时钟中断频率、时钟中断延迟以及时钟中断服务例程的实现效率的影响。因此大多数实时系统中,时间精度成为了影响系统满足时限要求的障碍。尤其在支持LET编程模型的系统中,时间精度对系统执行效率以及时间行为可预测性的影响尤为明显。

### 发明内容

[0006] 针对上述存在的技术问题,本发明目的是:提供了一种基于LET编程模型的实时系统及任务构建方法,包括硬件部分的实时内核加速器结构和软件部分的实时执行器结构。通过对时间触发的任务管理、时间触发的赋值和周期精确的时间服务的支持,达到方便快捷地开发出具有时间语义且周期精确的基于LET任务的实时系统的目的。

[0007] 本发明的技术方案是:

一种基于LET编程模型的实时系统,包括实时内核加速器和实时执行器;所述实时内核加速器包括,

时间服务模块,用于提供机器周期精确的时间信息;

时间触发任务管理模块,用于对时间触发的任务管理;

时间触发赋值管理模块,用于对时间触发的赋值;

所述实时执行器包括，  
硬件抽象层，提供中断管理和硬件加速器功能接口；  
任务TCB结构数据，由内核的任务初始化API生成；  
内核API，提供任务调度和任务管理的接口。

[0008] 优选的，所述时间服务模块包括一加法器和一寄存器，所述寄存器保存系统复位后的CPU周期数，每个CPU周期到来时，加法器对该寄存器加一，记录CPU启动后的Cycle数。

[0009] 优选的，所述时间触发任务管理模块包括，

任务时间约束模块，包括多个任务时间约束子模块，分别对应每个任务，所述任务时间约束子模块包括一相位寄存器和一周期掩码寄存器，用于接收时间服务模块提供的时间信息，根据时间信息判断时间触发任务是否满足时间约束；

任务状态模块，用于管理任务的任務状态，包括一任务就绪寄存器和一任务存在标识寄存器；

优先级判优模块，接收任务状态模块的任务状态信息并输出当前处于就绪状态的最高优先级任务的优先级，当最高优先级任务产生变化时，向CPU发送中断信号，使其能够进行上下文切换来切换任务。

[0010] 优选的，所述时间触发赋值管理模块包括，

时间约束模块，包括多个时间约束子模块，用于对普通时间触发赋值寄存器模块和时间触发I/O赋值寄存器模块进行约束，所述时间约束子模块包括一相位寄存器和一周期掩码寄存器，用于接收时间服务模块提供的时间信息，根据时间信息判断时间触发赋值是否满足时间约束；

时间触发寄存器模块，包括多个普通时间触发赋值寄存器子模块和I/O时间触发赋值子模块，所述普通时间触发赋值寄存器子模块和I/O时间触发赋值子模块均包括一时间触发赋值Buffer寄存器和一时间触发寄存器，对普通时间触发赋值寄存器子模块和I/O时间触发赋值子模块的写操作为对时间触发赋值Buffer寄存器写，读操作为对时间触发寄存器的读，当且仅当从相应的时间约束子模块传过来满足时间约束信号时，时间触发寄存器的值更新为时间触发赋值Buffer寄存器的值；

时间触发赋值控制模块，包括一数据方向寄存器和与外界相连的外界引脚，外界引脚根据数据方向寄存器的值决定其数据方向。

[0011] 优选的，所述实时内核加速器通过AXI总线接口与处理器相连，实时内核加速器的各个寄存器都映射到相应的内存地址中，处理器通过访问映射的内存地址与内核加速器进行通信和同步。

[0012] 优选的，所述内核API包括，

任务初始化API，接收用户的任务初始化结构体，初始化相应任务的TCB和栈结构，同时在硬件加速器的时间触发任务管理器中注册任务；

启动调度器API，用于启动硬件加速器中时间触发任务管理；

任务挂起API，用于挂起时间触发任务管理器中处于就绪状态的任务；

任务恢复API，用于把处于挂起状态的任务转变成就绪状态；

任务删除API，用于删除任务的TCB和反注册时间触发任务管理器中的任务。

[0013] 本发明还公开了一种基于LET编程模型的实时系统的任务构建方法，包括以下步

骤:

S01:用户创建LET任务,定义LET任务的周期、相位、输入和输出,通过内核API的任务初始化API初始化任务,将任务属性初始化为时间触发;

S02:使用启动调度器API启动硬件加速器中时间触发任务管理模块;

S03:到达任务的输入时刻和释放时刻,实时内核加速器的时间触发赋值模块触发赋值行为,完成任务的输入,时间触发任务模块把任务从等待状态修改为就绪状态,并向CPU发送中断信号;

S04:CPU接收到中断信号,完成上下文切换后执行任务,CPU执行完任务,将任务的输出放到时间触发赋值的Buffer,然后执行其他优先级比较低的任务或者事件触发的任务;

S05:任务的输出时刻到达,实时内核加速器的时间触发赋值模块触发赋值行为,完成任务的输出。

[0014] 与现有技术相比,本发明的优点是:

可以有效地建立LET任务模型,针对LET任务模型,提供时间触发赋值和时间触发任务管理,从软件和硬件进行优化,具有时间语义且周期精确,相对通用的CPU的软件实现有更高的性能和更好的实时性。

## 附图说明

[0015] 下面结合附图及实施例对本发明作进一步描述:

图1为本发明实时内核加速器结构图;

图2为本发明实时执行器结构图;

图3为本发明实现的LET任务的运行流程。

## 具体实施方式

[0016] 以下结合具体实施例对上述方案做进一步说明。应理解,这些实施例是用于说明本发明而并不限于限制本发明的范围。实施例中采用的实施条件可以根据具体厂家的条件做进一步调整,未注明的实施条件通常为常规实验中的条件。

[0017] 实施例:

一种基于LET编程模型的实时系统,包括由硬件构成的实时内核加速器和由软件构成的实时执行器。

[0018] 实时内核加速器,如图1所示,包括时间服务模块、时间触发任务管理模块和时间触发赋值管理模块。

[0019] 时间服务模块用于提供机器周期精确的时间服务,向时间触发赋值管理模块、时间触发任务管理模块和用户提供时间信息。时间服务模块由一个加法器和一个64位寄存器组成。该64位寄存器保存系统复位后的CPU周期数,每个CPU周期到来时,加法器对该寄存器加一,以记录CPU启动后的Cycle数,从而完成对时间的记录。

[0020] 时间触发赋值管理模块分为三部分,分别为时间约束模块、时间触发寄存器模块和时间触发赋值控制模块。

[0021] 时间约束模块包括12个时间约束子模块,分别对应时间触发寄存器模块的8个32位的普通时间触发赋值寄存器和4个8位的时间触发I/O赋值寄存器,时间触发赋值寄存器

模块和时间触发I/O赋值寄存器模块进行约束。每个子模块中有一个32位的相位寄存器和一个32位的周期掩码寄存器,子模块接收来自时间服务模块提供的时间信息,使用该时间信息与周期掩码寄存器的值做与操作,该操作相当于一次求余操作,用于得到与该周期相关的当前时间的相位,然后把当前时间的相位与相位寄存器的值做比较,输出一个布尔值。

[0022] 时间触发寄存器模块包括9个子模块,其中8个普通时间触发赋值寄存器子模块和1个时间触发I/O赋值子模块。普通时间触发寄存器子模块中包含一个32位的时间触发赋值Buffer寄存器和一个32位的时间触发寄存器,用户对该子模块进行配置时,写操作是对时间触发赋值Buffer寄存器进行写,读操作是对时间触发寄存器进行读。时间触发赋值Buffer寄存器和时间触发寄存器通过一个双路选择器相连,其作用是,当从相应的时间约束子模块传过来的布尔值为1时时间触发寄存器的值更新为时间触发赋值Buffer寄存器的值否则保持不变。时间触发I/O赋值子模块其同样有着一个32位的时间触发赋值Buffer寄存器和一个32位的时间触发I/O寄存器,两个寄存器每8位为一组,分别通过4个双路选择器相连,当从相应的时间约束子模块传过来的布尔值为1时时间触发I/O寄存器的相对应的组的8位值更新为相对应的时间触发赋值Buffer值否则保持不变。

[0023] 时间触发赋值控制模块包括一个32位的数据方向寄存器,和32个与外界相连的引脚,其根据寄存器的值控制着时间触发I/O寄存器与外部32个I/O引脚的数据方向。

[0024] 时间触发任务管理模块总共支持64个任务的管理,其分为三部分,分别为任务时间约束模块、任务状态模块和优先级判优模块。

[0025] 任务时间约束模块有着64个子模块,分别对应64个任务。每个子模块中有一个32位的相位寄存器和一个32位的周期掩码寄存器,子模块接收来自时间服务模块提供的时间信息,根据时间信息判断时间触发任务是否满足时间约束。具体为使用该时间信息与周期掩码寄存器的值做与操作,得到与该周期相关的当前时间的相位,然后把当前时间的相位与相位寄存器的值做比较,输出一个布尔值给任务状态模块对应的寄存器相应的位。

[0026] 任务状态模块管理着每个任务的任务状态,任务状态包括但不限于睡眠状态、等待状态、就绪状态和中断状态。包括一个64位的任务就绪寄存器和一个64位的任务存在标识寄存器,分别对应64个任务。任务存在标识寄存器的每一位表示一个任务的存在标识,若当前位为1,表示该对应位的优先级的任务存在,任务就绪寄存器的值的每一位表示一个任务的就绪状态,若当前位为1,表示该任务为就绪状态否则为等待状态,当任务时间约束子模块传来值为1的布尔值时,相对应的就绪寄存器位修改为1。

[0027] 任务就绪寄存器和任务存在标识寄存器做一个按位与操作,生成当前存在的任务就绪集合数据输出给优先级判优模块,优先级判优模块接收该信号并输出当前就绪任务集合中的最高优先级任务的优先级,当最高优先级任务产生变化时,向CPU发送中断信号,使其能够进行上下文切换来切换任务。

[0028] 本实施例中,实时内核加速器结构通过AXI总线接口与处理器相连,实时内核加速器的各个寄存器都映射到相应的内存地址中,处理器通过访问内存地址与内核加速器进行通信和同步。同时,加速器与CPU的Int中断有连接,用于传输中断信号。实时内核加速器中的各个寄存器,都映射到对应的不同的内存地址中,但是对于时间触发寄存器模块的子模块的Buffer寄存器和时间触发寄存器,两个寄存器都映射到相同的地址:当对该地址进行写时,是对Buffer寄存器进行写;当对该地址进行读时,是对时间触发寄存器进行读。

[0029] 软件构成的实时执行器如图2所示,包括硬件抽象层、任务TCB和内核API。

[0030] 硬件抽象层提供了中断管理和与硬件加速器功能接口。中断管理包括对外部事件的响应、用户软中断的响应和加速器的任务调度的响应。硬件加速器的功能接口包括时间触发赋值模块接口、时间触发任务模块的接口和时间服务模块的接口。

[0031] 任务TCB结构存放于内存中,包括但不仅限于任务的入口地址、优先级、周期、相位、时间触发输入端口、时间触发输出端口、栈指针、栈底指针和栈长度。该数据结构由内核API的任务初始化API生成。

[0032] 内核API包括但不仅限于任务初始化、启动调度器、任务挂起、任务恢复和任务删除。任务初始化API接收一个用户的任务初始化结构体,然后初始化相应任务的TCB和栈结构,同时在硬件加速器的时间触发任务管理器中注册任务;启动调度器API用于启动硬件加速器中时间触发任务管理;任务挂起API用于挂起时间触发任务管理器中处于就绪状态的任务;任务恢复API用于把处于挂起状态的任务转变成就绪状态;任务删除API用于删除任务的TCB和反注册时间触发任务管理器中的任务。

[0033] 如图3所示,使用本软硬件实现的LET任务的运行流程包括以下步骤:

S1、使用内核API初始化任务,该步骤中用户创建LET任务,定义LET任务的周期、相位、输入和输出,把任务属性初始化为时间触发;

S2、使用内核API启动加速器;

S3、到达任务的输入时刻和释放时刻,实时内核加速器的时间触发赋值模块触发赋值行为,完成任务的输入,时间触发任务模块把任务从等待状态修改为就绪状态,并向CPU发送中断信号;

S4、CPU接收到中断信号,完成上下文切换,然后执行任务;

S5、CPU执行完任务,把任务的输出放到时间触发赋值的Buffer,然后去执行其他任务,如优先级比较低的任务或者事件触发的任务;

S6、任务的输出时刻到达,实时内核加速器的时间触发赋值模块触发赋值行为,完成任务的输出。

[0034] 上述实例只为说明本发明的技术构思及特点,其目的在于让熟悉此项技术的人是能够了解本发明的内容并据以实施,并不能以此限制本发明的保护范围。凡根据本发明精神实质所做的等效变换或修饰,都应涵盖在本发明的保护范围之内。



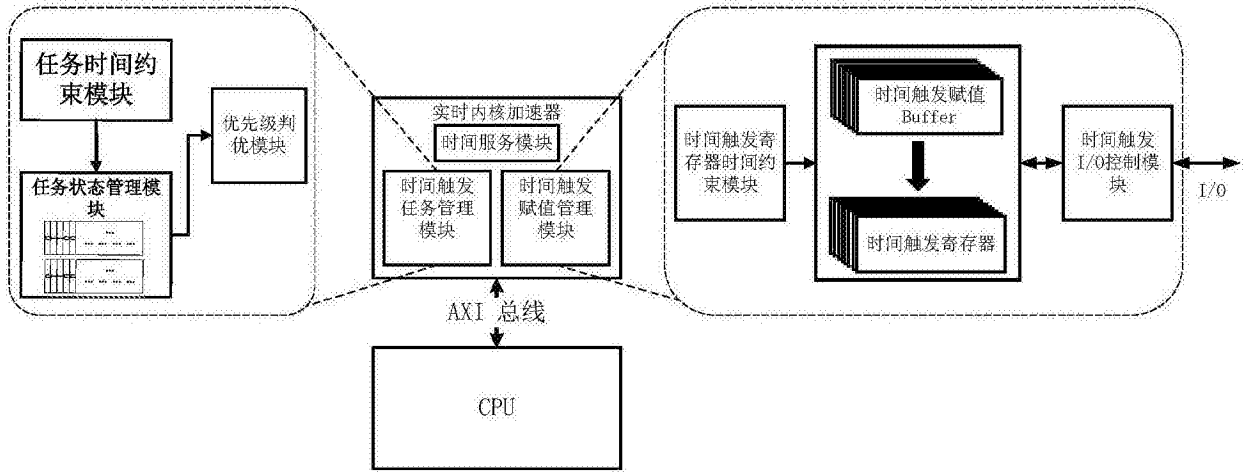


图1

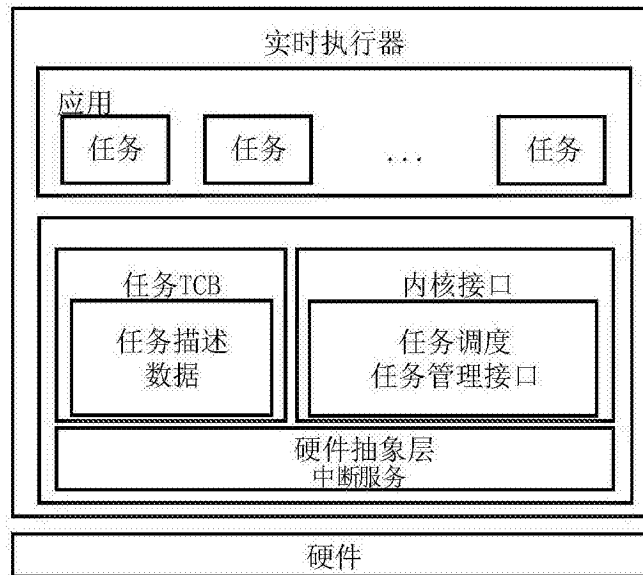


图2

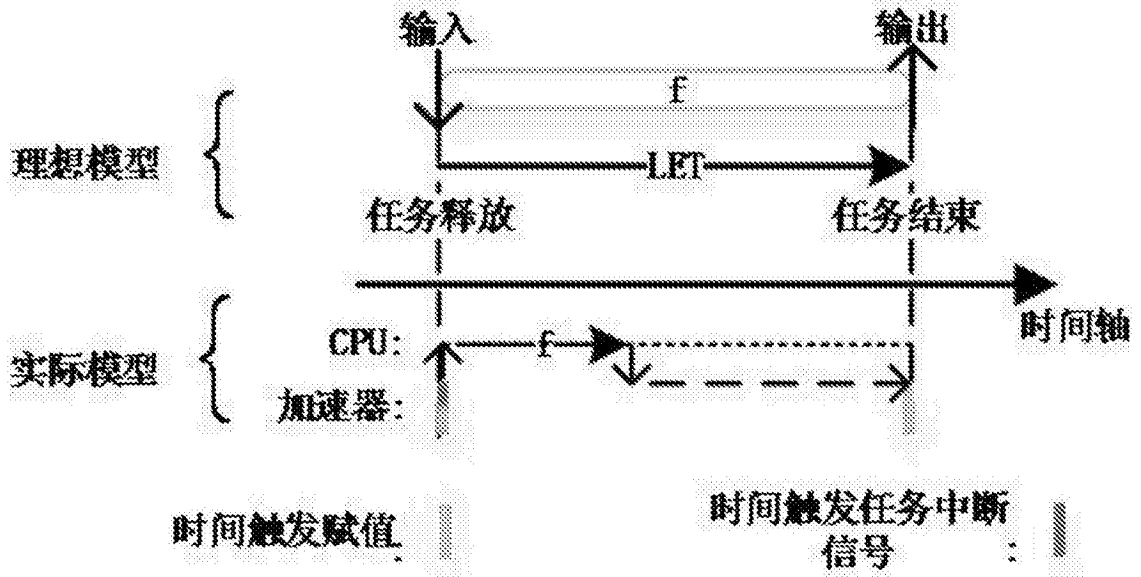


图3