一种基于可重构片上互连的多核处理器架构

刘 强[1,2],苏嘉玮[1,2],罗 宇[1,2],王 滔[1,2],孙泽渝[1,2]
(1.上海航天电子技术研究所,上海201109;2.上海航天智能计算技术重点实验室,上海201109)

摘 要: 面向宇航应用,基于可重构片上互连,本文提出了一种新型的多核处理器架构。该架构在锁步技术的基础上进行了改进,利用可重构片上互连使得锁步技术引入的冗余核可以分时复用,以充分发挥多核性能。基于该架构的处理器软件编程,需对任务进行关键等级划分,使得非关键任务可以线程级并行执行,关键任务锁步执行,满足宇航应用对高可靠和高性能的需求。本文给出了该架构基于第五代精简指令集(RISC-V)的一种实现,并进行验证。结果表明:该架构实现了多核处理器在多核锁步和多核并行两种状态之间切换,分别获得高可靠或高性能的特性。

关键词: 宇航; 可重构片上互连; 处理器; 锁步; RISC-V
中图分类号: TP 332 **文献标志码:** A **DOI:** 10.19328/j.cnki.2096-8655.2021.04.009

A Multi-core Processor Architecture Based on Reconfigurable On-chip Interconnection

LIU Qiang[1,2], SU Jiawei[1,2], LUO Yu[1,2], WANG Tao[1,2], SUN Zeyu[1,2]
(1.Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China;
2.Key Laboratory of Intelligent Computing Technology (SAST), Shanghai 201109, China)

Abstract: A new type of multi-core processor architecture based on reconfigurable on-chip interconnection is proposed for aerospace applications. The architecture is improved on the basis of the lock-step technology. With reconfigurable on-chip interconnection, the redundant cores introduced by the lock-step technology can be reused by time sharing so as to give full play to the multi-core performance. In the processor software programming based on this architecture, in order to meet the high reliability and high performance requirements of aerospace applications, the tasks should to be classified into critical levels, so that the non-critical tasks can be executed in parallel at the thread level and the key tasks can be executed in lock-step. An implementation of the architecture based on the fifth generation of reduced instruction set computing (RISC-V) is proposed and verified. The results show that the architecture can realize the switching of the multi-core processor between the state of multi-core lock-step and the state of multi-core parallelism with high reliability or high performance characteristics.

Key words: aerospace; reconfigurable on-chip interconnection; processor; lock-step; the fifth generation of reduced instruction set computing (RISC-V)

0 引言

电子设备在太空中面临着恶劣的运行环境,如单粒子效应、总剂量效应等,非常容易造成电子设备运行错误。因此,在宇航产品设计中常使用多种方法来提供产品容错能力,如专为宇航设计的处理器芯片等。这些芯片为了应对空间单粒子效应,为内部易发生单粒子翻转的基础电路部件做了三模冗余[1-2],由此带来的后果是更差的时序,导致芯片相比于未做冗余的处理器,只能在更低的主频下工作。

收稿日期: 2021-03-15; **修回日期:** 2021-07-07
基金项目: 科技部国家重点研发计划(2016YFB0501004)
作者简介: 刘 强(1995—),男,硕士,助理工程师,主要研究方向为SoC设计与验证。

对于多核处理器架构，可以将多核处理器内部分为3类部件：处理器核心、片上总线、片内集成外设。多核处理器内部通常是一个相对独立的处理器核心，这些处理器核心通常包含了多级流水线，有些还会采用硬件超线程技术。这些处理器核心通过某种片上总线集成到一起，如AMBA总线、TileLink总线等。处理器核心在片上总线进行核间通信、地址译码，访问存储器和外设。

为宇航应用设计处理器的通常思路是对处理器核心内部进行冗余。在汽车电子行业中，采用了一种称为锁步的技术，提供另一个粒度的冗余[3]。锁步核是实现高诊断覆盖度（检测错误发生的能力）的一种传统方法。实现方法是：两个核运行同样的程序，将结果输入一个比较逻辑中，周期性比较两个核的输出结果是否相同。如果相同，则继续运行；否则，则需要采取一定的措施。如果一段时间后错误仍然存在，可能需要重启或重新检查，如图1所示[4]。

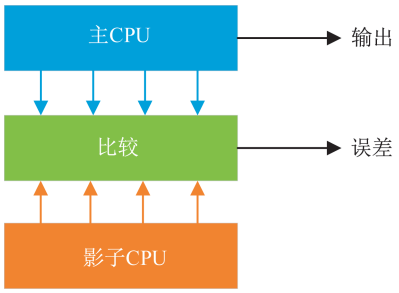

图1　双核锁步结构

Fig.1　Dual-core lock-step architecture

锁步技术采用整个处理器核心作为冗余部件，而不用在处理器核心内部做冗余。因此，对于处理器核心而言，可以采用商业上流行的微架构，这些微架构通常具备更高的性能[5]，不必为了提供高容错能力而改变微架构，并牺牲时序。

锁步技术的缺点也同样明显。锁步核的设计是针对芯片的设计，也就意味着设计一旦确定，便没有可调整性，实际上是以处理器核心为粒度做了冗余。相比对于修改处理器核心微架构做冗余，可以获得更好的性能。然而，对于整个多核处理器而言，实际上还是牺牲了内部资源来保证电子设备的稳定运行。

对于高性能处理器来说，往往复杂度高，不确定性大，实现锁步很难。软件冗余执行就是一种可选方案。如图2所示，假设两个独立应用执行在两个不同的CPU和或者两个虚拟机上，这两个应用的结果输出到一个被称为“safety island”的核中，由这个核负责决策和执行。这个核使用单独的时钟和电源，且有很高的安全性能[6-7]。这种方案提升了系统复杂度，但是软件执行更加灵活。锁步的比较机制是固化在芯片中的，而“safety island”是一个单独的核。

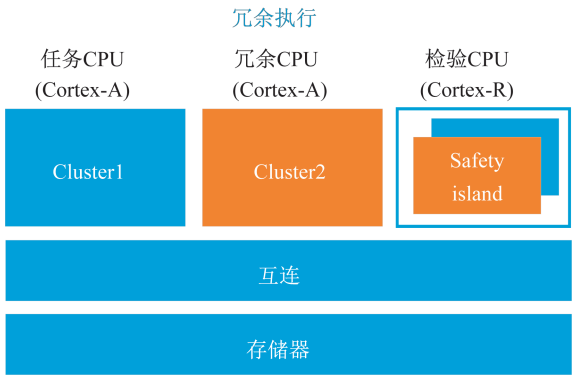

图2　Safety island 结构

Fig.2　Safety island architecture

以上两种方案各有优劣，ARM在Cortex-A76AE上集成了一种叫“Split-Lock”的技术来集成上述两种方案的优点。“Split-Lock”可以被配置成两种模式：“split mode”，两个CPU可以独立执行不同的程序或任务；“lock mode”，两个CPU执行锁步模式[8-9]。这种技术可以支持：如果两个CPU中有一个损坏，可以在降级模式下继续运行，即只运行好的那个CPU。这种技术大大提高了安全性，在汽车电子中得到了广泛应用。

航天型号任务中火箭、飞船和卫星平台与载荷大规模采用的CPU主要用于信号处理、信号和设备控制、数据计算和传输，以及综合多任务调度。考虑到星上任务，特别是平台和载荷共享计算资源的卫星上运行的任务，有的任务需要保证高可靠的运行，如平台的姿轨控和整星的遥控遥测，有的任务需要消耗高算力，如载荷中的各种算法。因此，要求卫星上的处理器既要能提供高算力，又要能够提供很好的容错能力。考虑到航天领域一般采用基于开源指令集平台的处理器，如SPARC、PowerPC等[10-11]，美国加州大学伯克利分校研发的第五代精简指令集RISC-V因其模块化、硬件实现简单并且完全开源开放的优点[12]，特别适用于航天领域处理

器设计。针对此,本文提出一种基于可重构片上互连的多核处理器架构,并基于 RISC-V 开源指令集提供了一种具体微架构实现。

1 多核可重构处理器架构

基于可重构片上互连的多核处理器架构的基本思路是:对片上总线进行特殊改造,使得多核处理器可以通过软件的配置,将处理器核心作为冗余核构成多核锁步的结构,也可以作为普通核,提供多核心本身具备的性能。

现代多核处理器最常见的架构是利用片上总线将多个处理器核连接起来,配以某种核间通信机制,如共享内存、核间消息等,实现线程级并行[13-14]。相比于单核处理器,多核处理器只要搭配合适的软件,性能可成倍提高。多核可重构处理器架构,即利用软硬件的协同设计,实现可重构片上互连,使得多核处理器可以在多核锁步和多核并行之间切换,从而在高容错能力和高性能之间动态取舍。在硬件上,需要通过片上总线将多个处理器核连接起来,设计专门的可重构控制器模块控制核间总线连接关系;在软件上,需要将采用此架构的多核处理器上运行的软件划分为一个个任务,以任务为基本粒度进行调度和切换。软件开发人员可以根据任务对运行可靠性的要求定义每个任务的关键等级:有的任务(如姿轨控)对运行可靠性要求很高,则将其定义为高关键等级;有的任务(如载荷数据处理)对处理速度的要求高而对运行可靠性要求相对更低,则可以将其定义为低关键等级任务。当需要运行高关键等级任务时,多个处理器核可形成一种锁步结构,利用多个核执行相同的指令流,对每一条指令的执行结果都进行实时比对检查,当发生不一致时触发相应的处理,以此确保每一条指令都执行正确。面对低关键等级的任务时,多个处理器核可以线程级并行执行这些任务,提高处理速度。

多核可重构处理器架构一种基于 RISC-V 的实现方式如图 3 所示。在多核处理器内部,集成了 4 个带紧耦合存储器(Tightly Coupled Memory, TCM)的 RISC-V 处理器核、高速/低速的片上总线及相应的总线桥、直接存储访问(Direct Memory Access, DMA)部件、串行输入输出(Serial Input and Output, SIO)、定时器/软件中断控制器、外部中断控制器,以及用于实现可重构片上互连的可重构控制器。片上总线采用 AMBA 总线[15],多核可重构处理器内部的各个处理器核对于片上总线的访问都需要经过可重构控制器的中继,因此,在可重构控制器中可以对处理器核每一条指令的执行结果都进行检验,以此决定将哪个处理器核的结果作为最终的输出。

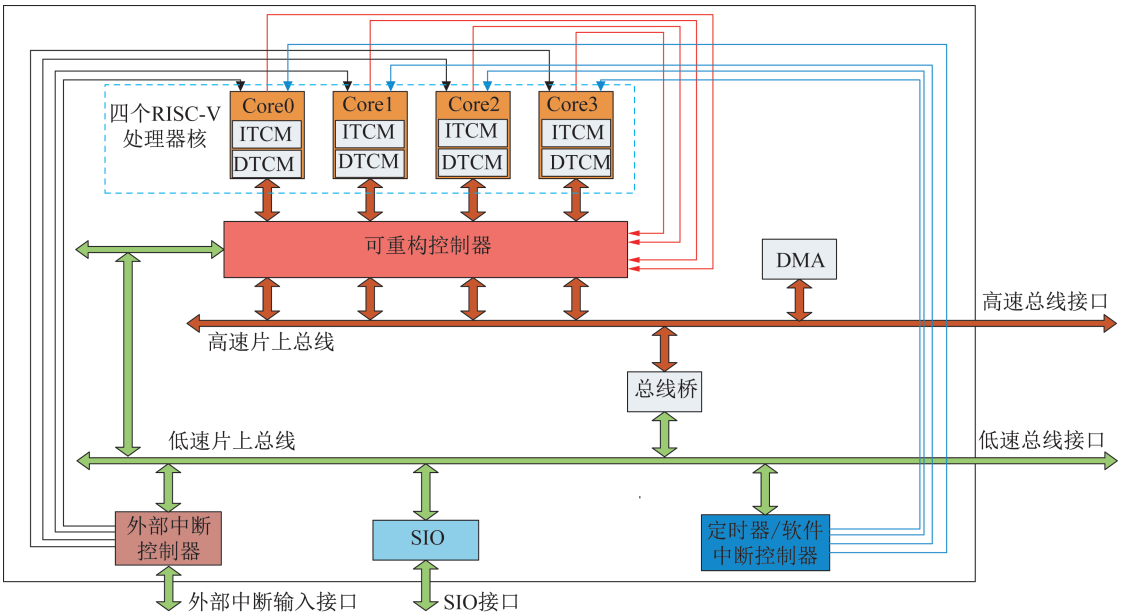

图 3 多核可重构处理器架构

Fig.3 Multi-core reconfigurable processor architecture

2　多核可重构硬件设计

在硬件上，因为要在多核并行和多核锁步这两种核间关系实现动态重构，因此，需要一个称为“可重构控制器”的外部模块来控制核间总线连接关系。可重构控制器的内部结构如图 4 所示，其内部包含了多路数据分配器、多路数据选择器、三取二逻辑，以及用于和处理器核交互的寄存器。可重构控制器要能够根据软件的状态来改变核间连接关系的状态，需要定义一组寄存器实现软硬件之间的交互。对于一个经典的 5 级流水线的处理器而言，一条指令的执行过程可以分为取指、译码、执行、访存、回写 5 个步骤。其中，访存步骤可以看作是指令的输入输出交互过程。当整个多核处理器处在多核锁步的状态去运行关键任务时，需要对指令输出进行三取二表决，这个表决动作在可重构控制器内完成，因此，可重构控制器内部包含三取二逻辑。当整个处理器在多核锁步和多核并行状态之间切换时，总线连接关系会发生变化，这个功能通过可重构控制器内部的多路数据选择器和多路数据分配器实现。

通过这种可重构片上互连，多核处理器可以在多核并行状态和多核锁步状态之间进行动态切换。如图 4(a)所示，此种工作模式下所有的处理器核的地址/数据线实际上直接连到了片上总线上，可重构控制器相当于不存在。这种状态下的多核可重构处理器便是常规的具有多核性能增长的多核处理器。多核处理器中的核心越多，带来的性能增长越高。而当软件载入一个关键任务需要执行时，多核可重构处理器需要切换到多核锁步状态去执行这个关键任务，如图 4(b)所示，此时指令的输出结果经过三取二之后才会到达片上总线，因此，对于单个核内发生的执行错误不会造成整个多核处理器的功能中断。

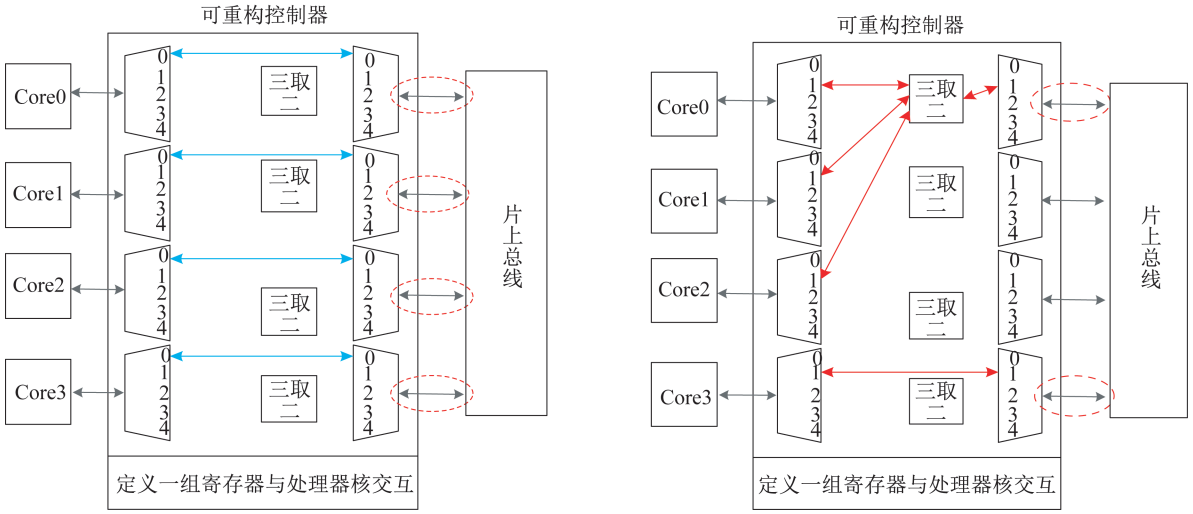

图 4　可重构控制器内部结构

Fig.4　Internal structure of the reconfigurable controller

对于具有 4 个处理器核的多核可重构处理器，从其中取 3 个处理器核进行组合，选中的 3 个处理器核构成锁步结构，剩下的 1 个核独立运行，因此，能产生 5 种可能的工作模式，见表 1。

基于此种多核可重构处理器架构，4 种不同锁步形式表征着可重构控制器内部不同的总线互连关系。因此，一种锁步形式要向另一种锁步形式转变，必须经由多核并行的状态过渡，如图 5 所示。

表 1　多核可重构处理器工作模式

Tab.1　Working modes of the multi-core reconfigurable processor

工作模式	含义
0-1-2-3	Core0、Core1、Core2、Core3 四核并行执行(线程级并行)
012-3	Core0、Core1、Core2 锁步执行，Core3 单独执行
013-2	Core0、Core1、Core3 锁步执行，Core2 单独执行
023-1	Core0、Core2、Core3 锁步执行，Core1 单独执行
123-0	Core1、Core2、Core3 锁步执行，Core0 单独执行

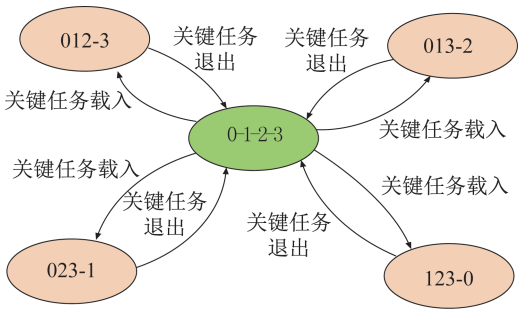

图 5 多核可重构处理器工作模式切换

Fig.5 Working mode switching of the multi-core reconfigurable processor

3 多核可重构软件设计

3.1 软件部署

多核可重构处理器架构中的软件部署采取分布式的部署方式。每个处理器核各自带有自己私有的紧耦合存储器,在紧耦合存储器中分别存放一份软件私有副本,而在公共可访问的主存中进行分块,分别存放各自私有的数据和公共的数据,如图 6 所示。这种部署方案一个明显的优点是,当多个 CPU 线程级并行执行多个任务时,每个 CPU 实际是以独立计算机的形式运行,各个 CPU 之间实现相当大程度的解耦,一个处理机出错不会影响其他处理机。

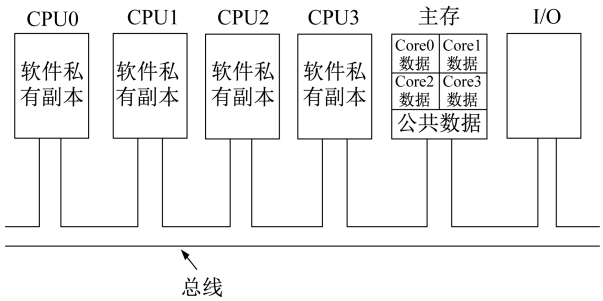

图 6 多核可重构处理器架构软件部署

Fig.6 Software deployment of the multi-core reconfigurable processor architecture

3.2 任务切换

高级编程语言编译出的软件运行时的内存结构如图 7 所示,里面预留了堆区和栈区来保存软件运行过程中产生的动态数据。任务切换过程包括将当前任务的现场保存到栈区,然后恢复新任务的现场开始执行。其中,任务的现场是通过任务栈来描述,它包含了要恢复该任务原来状态所需的所有信息,及任务上下文和任务包含的函数队列。

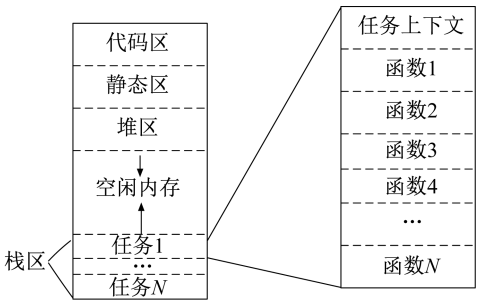

图 7 运行时内存结构图

Fig.7 Runtime memory structure

对于单独一个处理器核而言,任务切换的一般过程如下:1) 保存当前任务现场;2) 切换任务栈;3) 从任务栈中载入新任务现场开始执行。

多核可重构处理器架构下的多任务切换,特点在于,切换任务的过程中,往往伴随着多核模式的切换,如图 8 所示。对于单核系统而言,任务切换只是纯软件的操作,而对于采用多核可重构处理器架构的计算机系统而言,当不同关键等级的任务之间切换时,任务切换需要软硬件协同工作才能实现。在硬件上,处理器核和可重构控制器之间进行了一系列的交互;在软件上,进行了任务栈选择和切换。

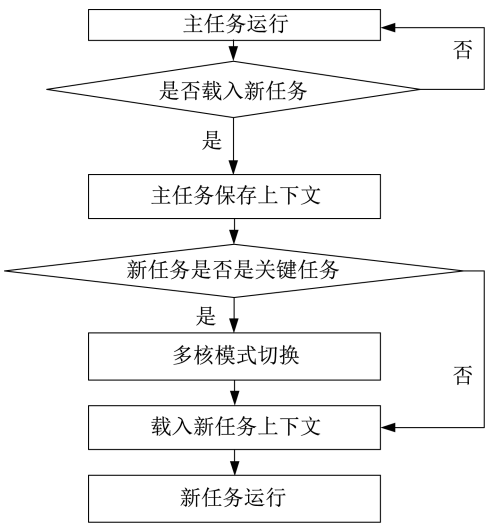

图 8 任务切换过程

Fig.8 Task switching process

3.3 应用程序框架

软硬件的紧密耦合使得多核可重构处理器架

构上应用程序的编写必须遵循一定的程序框架，如图9所示。这个程序框架最显著的特点是需要发生两次中断处理。在第一次中断里保存当前任务现场，然后使处理器核进入等待中断（Wait For Interrupt，WFI）状态。当处于WFI状态时，指令流会停住，处理器核会一直等待中断。可重构控制器在3个处理器核都进入WFI状态时切换总线连接关系，使得多个处理器核形成锁步或者从锁步状态脱离；在第二次中断里，各个处理器核从任务栈中载入新任务的现场。

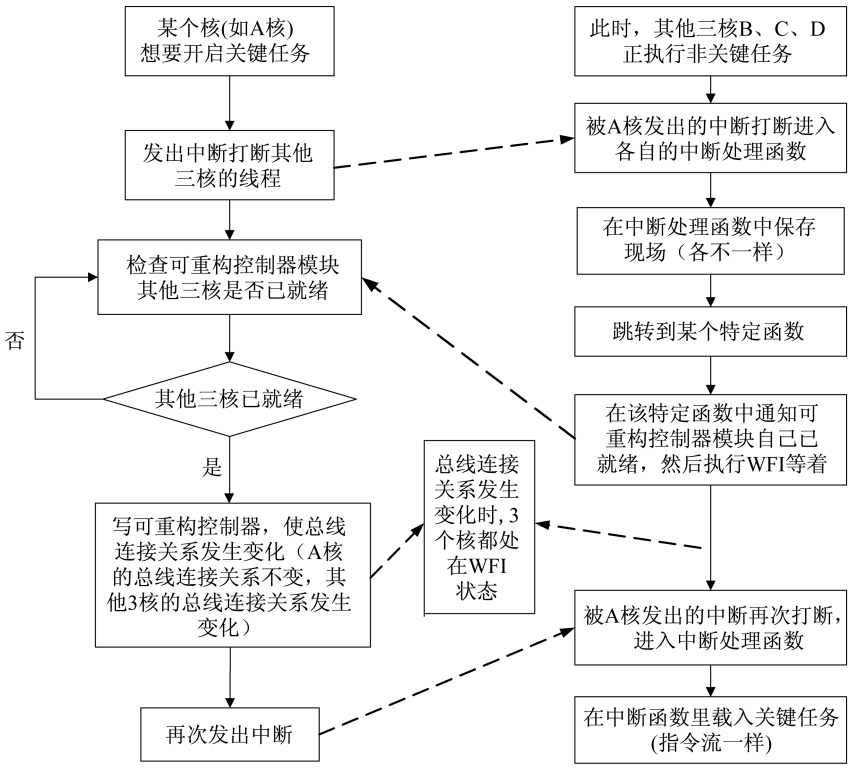

图9　多核可重构架构应用程序框架

Fig.9　Application framework of the multi-core reconfigurable architecture

4　多核可重构验证

4.1　多核并行性能验证

针对多核可重构处理器的应用场景，目前解决多核并行执行不同进程的功能，不考虑多核协同处理同一任务的情况，因此，考察单核性能来表征多核可重构处理器在多核并行状态下的性能。

测试处理器性能，业界通常是采用一些基准测试软件来跑分，如Dhrystone、WhetStone、CoreMark等，用以测量处理器在执行某种特定类型计算的硬件最高实际运行性能，如用Dhrystone测量整数运算性能，用Whetstone测量浮点运算性能。

本文使用的是一种基于RV32I指令集实现的处理器核xmcu_core_m0，使用Dhrystone来测试其整数运算性能，其性能测试结果见表2。

表2　xmcu_core_m0性能测试结果

Tab.2　Performance test results of xmcu_core_m0

主频/MHz	编译选项	Dhrystones/s	DMIPS/MHz
50	gcc-8.3.0-O2	57 415.2	0.7
50	gcc-8.3.0-O3	105 954.7	1.2

4.2　多核锁步功能验证

本文描述的多核锁步，以4核处理器为例，是指其中3核锁步运行，剩余1核独立运行。对于锁步运行的3个核而言，应实现“纠一检二”的功能，即其中1个核执行错乱，但是3个核最终的输出，应该依

然是准确的;当2个核都执行错乱的时候,能够给出信息到共享内存中表征此错误。

为验证此功能,本文使用软件方法进行故障注入。无论是锁步运行还是并行运行,多核可重构处理器中的核都是从TCM中取指令。正常情况下,锁步运行的3个核执行相同的指令流,是因为各自的TCM中存放着相同的指令流,利用软件方法修改TCM中的内容,进入锁步状态后,个别核取出来的指令流与其他核不一致,以此来验证"纠一检二"的功能。修改Core0的指令流,如图10所示,模拟3核中有1核指令错乱的情况;修改Core0、Core1的指令流,如图11所示,锁步的3个核指令流各不一样,用以模拟3核中有2核指令错乱的情况。

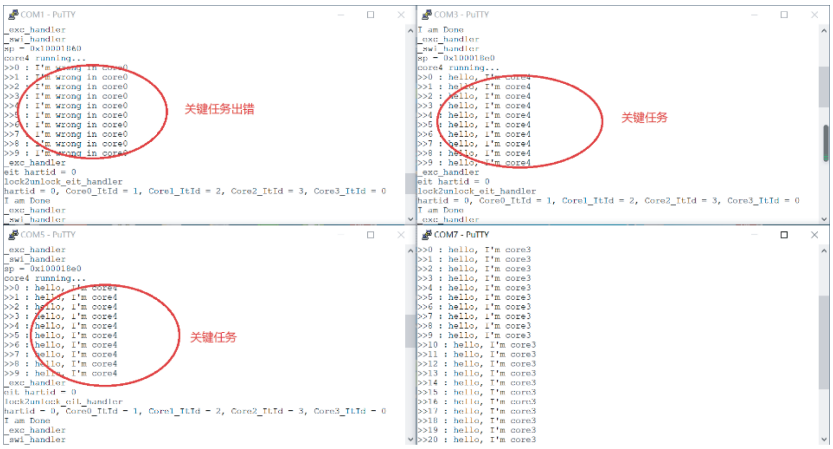

图10 三份中有一份错的情况

Fig.10 A situation where one of the three copies is wrong

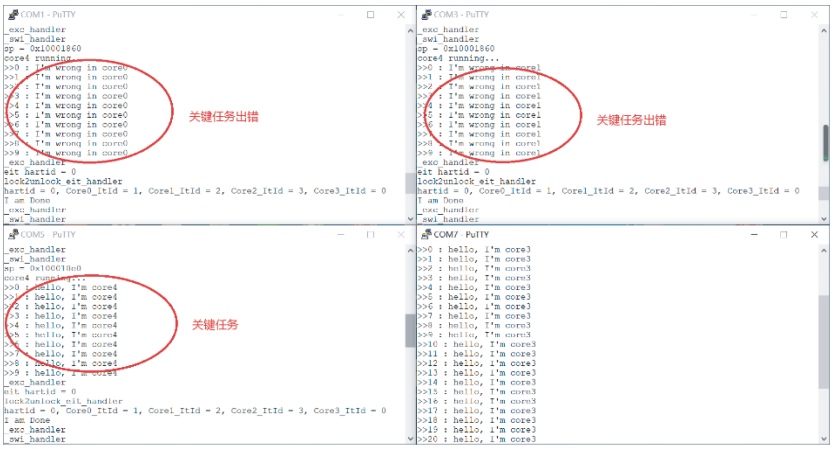

图11 三份中有两份错的情况

Fig.11 A situation where two of the three copies are wrong

结果表明,当三份中有一份错时,虽然在关键任务中,有一个处理器核的打印输出与其他两个相异,但是最终在共享存储器DATA RAM中写下的数据是正确的。当三份中有两份错时,在关键任务中,3个处理器核的打印输出各不相同,最终在共享存储器DATA RAM中写下的数据也是错误的,但是处理器在另一个变量里提示了这个数据错误。

4.3 多核可重构功能验证

多核可重构功能,是指多核可重构处理器能够在多核并行和多核锁步这两种状态之间自如切换。反映在指令流上,当多核并行状态转向多核锁步状态时,3个核的指令流由不同步转向同步;当多核锁步状态转向多核并行状态时,原本同步的指令流转向无明显同步关系。

由图12和图13可知,执行不同指令流的3个

核，在一系列动作后，实现了指令流同步，即由多核并行的状态进入了多核锁步的状态。

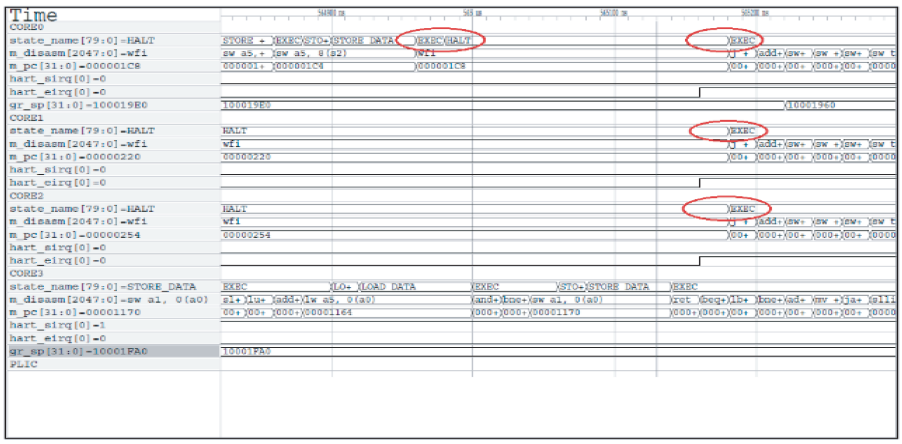

图 12　多核处理器状态变化

Fig.12　Status changes of the multi-core processor

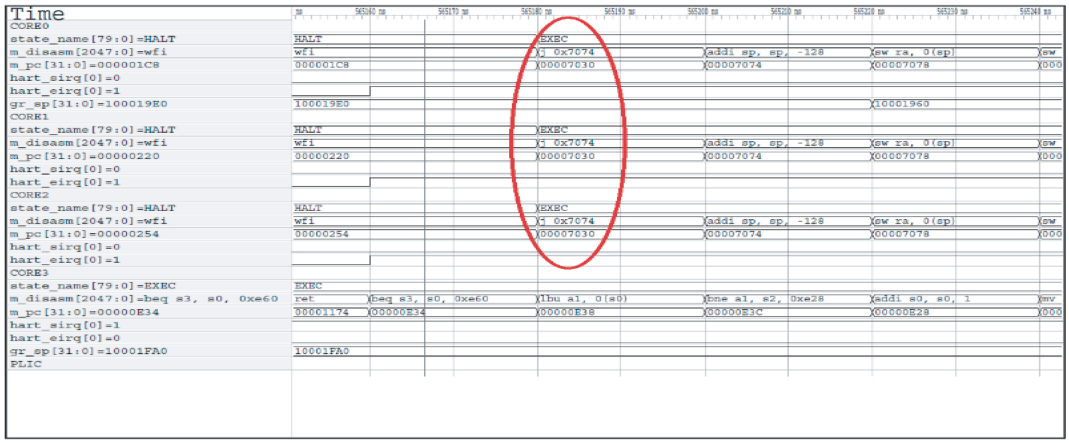

图 13　指令流同步

Fig.13　Instruction stream synchronization

5　结束语

多核可重构处理器架构集成了锁步技术和可重构片上互连技术，结合了两者的优点，使得多核处理器既能以多个核冗余的形式提高对内部软错误的容错能力，又能获得多核带来的性能增长，解决了锁步技术不能发挥多核性能的问题。从提高容错能力的角度而言，多核可重构处理器架构是在锁步技术基础上的改良，保留了锁步技术的优点，即锁步技术不依靠对每个时序逻辑单元进行三模冗余来提高容错能力，而是将完整的处理器核作为冗余部件，使得处理器核内部微架构设计可以不受冗余约束，为航天领域引用商业上的处理器微架构提供了可能。

参考文献

[1] 颜军，蒋晓华，唐芳福，等．面向宇航应用的高性能多核处理器 S698PM 芯片的设计[J]．航天控制，2016，34(4)：89-94.

[2] 李伟．基于 TSC695F 的星载计算机设计与实现[D]．西安：西安电子科技大学，2011.

[3] KROL T. Interconnecting lock-step synchronous fault-tolerant system based on voting and error-correcting codes[C]//1999 Pacific Rim International Symposium on Dependable Computing. Washington D. C., USA: IEEE Press, 1999: 62-69.

（下转第 74 页）